



**HAL**  
open science

# Tackling heterogeneity in federated learning systems

Othmane Marfoq

► **To cite this version:**

Othmane Marfoq. Tackling heterogeneity in federated learning systems. Artificial Intelligence [cs.AI]. Université Côte d'Azur, 2023. English. NNT : 2023COAZ4104 . tel-04498083

**HAL Id: tel-04498083**

**<https://theses.hal.science/tel-04498083v1>**

Submitted on 11 Mar 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THÈSE DE DOCTORAT

## Surmonter l'Hétérogénéité dans les Systèmes d'Apprentissage Fédéré

**Othmane MARFOQ**

Centre Inria d'Université Côte d'Azur, Équipe NEO

**Présentée en vue de l'obtention du grade de docteur en Informatique de  
l'Université Côte d'Azur**

**Dirigée par :** Giovanni NEGLIA, Directeur de Recherche, Centre Inria  
d'Université Côte d'Azur

**Devant le jury, composé de :**

Frédéric GIROIRE, Directeur de Recherche, CNRS

Peter RICHTARIK, Full Professor, King Abdullah University of Science and Technology

Marc TOMMASI, Professeur des Universités, Université de Lille

Martin JAGGI, Associate Professor, École Polytechnique Fédérale de Lausanne

Gauri JOSHI, Associate Professor, Carnegie Mellon University

Laetitia KAMENI, AI R&D Lead, Accenture Labs



**SURMONTER L'HÉTÉROGÉNÉITÉ DANS LES SYSTÈMES  
D'APPRENTISSAGE FÉDÉRÉ**

---

*Tackling Heterogeneity in Federated Learning Systems*

**Othmane MARFOQ**



**Jury :**

**Président du jury**

Frédéric GIROIRE, Directeur de Recherche, CNRS

**Rapporteurs**

Peter RICHTARIK, Full Professor, King Abdullah University of Science and Technology

Marc TOMMASI, Professeur des Universités, Université de Lille

**Examineurs**

Martin JAGGI, Associate Professor, École Polytechnique Fédérale de Lausanne

Gauri JOSHI, Associate Professor, Carnegie Mellon University

**Directeur de thèse**

Giovanni NEGLIA, Directeur de Recherche, Centre Inria d'Université Côte d'Azur

**Membres invités**

Laetitia KAMENI, AI R&D Lead, Accenture Labs

Othmane MARFOQ

*Surmonter l'Hétérogénéité dans les Systèmes d'Apprentissage Fédéré*

xiv+332 p.

*To my mentor, who illuminated the path, and to my family, whose unwavering support made this journey possible.*



# Surmonter l'Hétérogénéité dans les Systèmes d'Apprentissage Fédéré

## Résumé

L'apprentissage fédéré, qui provient de l'anglais "Federated Learning" (FL), se présente comme un cadre facilitant l'apprentissage collaboratif de modèles d'apprentissage automatique par des clients géographiquement répartis sans divulguer leurs données locales. Cette thèse se concentre sur la prise en charge de l'hétérogénéité, un défi majeur dans le domaine de l'apprentissage fédéré. L'hétérogénéité se manifeste par des variations entre les ensembles de données locaux des clients (hétérogénéité statistique), des disparités dans les capacités de stockage et de calcul (hétérogénéité système), et des fluctuations dans les ensembles de données locaux au fil du temps (hétérogénéité temporelle). Cette thèse explore différentes sources d'hétérogénéité dans le contexte de l'apprentissage fédéré et propose des algorithmes pratiques pour atténuer l'impact de l'hétérogénéité.

La première partie de la thèse se concentre sur la résolution des défis associés à l'hétérogénéité du système dans deux scénarios distincts : inter-silos et inter-appareils. Dans les environnements inter-silos, nous exploitons la théorie des systèmes linéaires dans l'algèbre max-plus pour modéliser le débit, c'est-à-dire le nombre de cycles complets par unité de temps, dans un système d'apprentissage fédéré entièrement décentralisé en inter-silos. Ensuite, nous proposons des algorithmes pratiques qui, en utilisant les caractéristiques mesurables du réseau, trouvent une topologie avec le débit le plus élevé ou avec des garanties de débit vérifiables. Dans les environnements inter-appareils, où les contraintes du système influencent la disponibilité et l'activité des clients, nous explorons différents niveaux de participation des clients, souvent présentant une corrélation au fil du temps et avec d'autres clients. Dans ce contexte, nous analysons un algorithme similaire à FedAvg sous une disponibilité hétérogène et corrélée des clients. L'analyse met en évidence comment la corrélation affecte négativement le taux de convergence de l'algorithme et comment la stratégie d'agrégation peut atténuer cet effet, même au prix de diriger l'entraînement vers un modèle biaisé. Guidé par l'analyse théorique, nous proposons "Correlation-Aware FL" (CA-Fed), un nouvel algorithme FL qui tente d'équilibrer les objectifs contradictoires de maximiser la vitesse de convergence et de minimiser le biais du modèle. À cette fin, CA-Fed ajuste dynamiquement le poids attribué à chaque client et peut ignorer les clients avec une faible disponibilité et une forte corrélation.

La deuxième partie traite de l'hétérogénéité statistique grâce à deux algorithmes de personnalisation. Le premier algorithme, appelé FedEM, repose sur une hypothèse souple selon laquelle l'ensemble de données de chaque client est généré à partir d'un mélange de distributions sous-jacentes communes inconnues. Le deuxième algorithme, appelé kNN-Per, combine un modèle global entraîné collectivement avec un modèle local de plus proches voisins (kNN) pour la personnalisation. Des garanties théoriques, notamment des bornes de convergence et de généralisation, sont fournies pour les deux algorithmes.

La troisième partie explore l'apprentissage fédéré pour les flux de données, en considérant deux scénarios : des échantillons indépendants tirés d'une distribution inconnue et des distributions de données composées de mélanges de distributions sous-jacentes inconnues. Pour le premier scénario, un meta-algorithme est proposé, offrant des informations sur la configuration et le compromis entre le temps d'entraînement et le biais du modèle appris. Pour le deuxième scénario, une variante fédérée de la descente du miroir séquentielle appelée Fed-OMD est introduite, avec un regret asymptotiquement sous-linéaire dans le cas des modèles de mélange Gaussien.

**Mots-clés :** Apprentissage fédéré, Personnalisation, Apprentissage séquentiel, Optimisation distribuée.



# Tackling Heterogeneity in Federated Learning Systems

## Abstract

Federated Learning (FL) stands as a framework facilitating geographically distributed clients to collaboratively learn machine learning models without divulging their local data. This thesis focuses on addressing heterogeneity, a major challenge in federated learning. Heterogeneity manifests in variations across clients' local datasets (statistical heterogeneity), disparities in storage and computational capabilities (system heterogeneity), and fluctuations in local datasets over time (temporal heterogeneity). This thesis investigates different sources of heterogeneity in the context of federated learning, and proposes practical algorithms to mitigate the impact of heterogeneity.

The first part of the thesis focuses on tackling challenges associated with system heterogeneity in two distinct scenarios: cross-silo and cross-device settings. In the cross-silo environments, we leverage the theory of linear systems in the max-plus algebra to model the throughput—the number of completed rounds per time unit—in a fully decentralized cross-silo federated learning system. Subsequently, we proffer practical algorithms that, under the knowledge of measurable network characteristics, find a topology with the largest throughput or with provable throughput guarantees. In the cross-device settings, where system constraints influence the availability and activity of clients, we explore varying degrees of client participation, often exhibiting correlation over time and with other clients. Within this context, we analyze a FedAvg-like algorithm under heterogeneous and correlated client availability. The analysis highlights how correlation adversely affects the algorithm's convergence rate and how the aggregation strategy can alleviate this effect at the cost of steering training toward a biased model. Guided by the theoretical analysis, we propose Correlation-Aware FL (CA-Fed), a new FL algorithm that tries to balance the conflicting goals of maximizing convergence speed and minimizing model bias. To this purpose, CA-Fed dynamically adapts the weight given to each client and may ignore clients with low availability and large correlation.

The second part of the thesis proposes two personalized federated learning algorithms. The first algorithm, FedEM, is a federated EM-like algorithm based on the flexible assumption that the dataset of each client is generated according to a mixture of unknown common underlying distributions. The second algorithm, kNN-Per, is based on local memorization, achieving personalization by interpolating a collectively trained global model with a local k-nearest neighbors (kNN) model based on the shared representation provided by the global model. Theoretical guarantees for both algorithms are provided. Convergence bounds for FedEM are established through a novel federated surrogate optimization framework; generalization bounds for kNN-Per are also presented.

The third part investigates federated learning for data streams. Two scenarios corresponding to different assumptions about the data process are considered. In the first scenario, a general FL algorithm is proposed for learning from data streams through weighted empirical risk minimization. The theoretical analysis provides insights into configuring such an algorithm and reveals a bias-optimization trade-off. In the second scenario, assuming client data distributions are mixtures of a finite number of unknown common underlying distributions with varying mixing weights, a federated variant of online mirror descent, named FEM-OMD, is proposed. In the case of Gaussian mixture models, it is shown that the regret of FEM-OMD is asymptotically sub-linear in the sample size.

**Keywords:** Federated learning, Personalization, Online learning, Distributed optimization.

# Acknowledgements

---

I extend my deepest gratitude to those who have been instrumental in the completion of this Ph.D. journey, contributing to both the challenges and successes that define this academic endeavor.

First and foremost, I wish to convey my heartfelt appreciation to my thesis advisor, Giovanni Neglia. Your steadfast guidance, profound expertise, unwavering trust, and continuous encouragement have served as the cornerstone of this thesis. Your emphasis on rigor has not only elevated my research skills but has also instilled in me a dedication to consistently deliver work of the highest standards. Throughout this thesis, I gained valuable insights from you, both academically and on a personal level. Collaborating with you proved to be a truly enriching experience, surpassing the typical expectations of a Ph.D. advisor.

I express my sincere gratitude to my esteemed thesis committee members: Frédéric Giroire, Martin Jaggi, Gauri Joshi, Peter Richtarik, and Marc Tommasi. I am profoundly thankful for their willingness to be part of my Ph.D. defense jury. Their dedication, as demonstrated by the time they invested in reviewing my thesis, participating in the defense, and offering invaluable comments and advice, is truly appreciated. The opportunity to defend my work and engage in discussions with experts in the field has been both challenging and rewarding.

I am grateful to collaborate with exceptional individuals. Chuan Xu brought an infectious optimism to our research, playing a pivotal role in driving the initial contributions of this thesis. Aurélien Bellet provided invaluable insights, and his positive outlook proved instrumental in shaping and enhancing our work. Additionally, working with Aryan Mokhtari during my visit to UT Austin was a privilege. I extend my thanks for the warm hospitality at UT Austin and for the fruitful discussions that significantly influenced the final chapter of this thesis. I am grateful to collaborate with talented researchers, including Angelo Rodio, Caelin Kaplan, Francescomaria Faticanti, and Emilio Leonardi. The exchanges with these individuals have tremendously benefited this thesis, enriching it with diverse perspectives and insights.

I am exceedingly grateful to have conducted my research at the rich and supportive learning environments of Inria Sophia Antipolis research laboratory and Côte d'Azur University. I am thankful to work along side the superb NEO team members, Younes Ben Mazziane, Tareq Si Salem, Angelo Rodio, Caelin Kaplan, Mikhail Kamalov, Olga Chuchuk, Jose Francisco Daunas Torre, Louis Hauseux, Xufeng Zhang, Xinying Zou, Ibtihal El Mimouni, Jacopo Talpini, Lucas Gamertsfelder, Haleh Dizaji, Rahul Misra, Maximilien Drevet, Andrei Bobu, Kishor Patil, Guilherme Iecker Ricardo, Ashok Krishnan Komalan Sindhu, Vijith Kumar, Ke Sun, Sadaf Ul-Zuhra, Emmanouil Athanasakos, Gabriele Castellano, Vinay Kumar B.R., Jake Clarkson, Francescomaria Faticanti, Suhail Mohamad Shah, Sara Alouf, Samir M. Perlaza, Konstantin Avrachenkov, Eitan Altman, and Alain Jean-Marie with whom I had enjoyable lunch discussions, coffee pauses, and social activities.

Special thanks are due to my family, in particular to my parents Mohamed and Touria, to my brother Ali, and to my sisters Fatima Zahra and Malak, for their unconditional love, encouragement, and understanding throughout this challenging yet rewarding pursuit. Their unwavering support and belief in my abilities have been a constant source of motivation.

This work would not have been possible without the support and encouragement of all those mentioned above, as well as many others who have played a part, however small, in this endeavor.



# Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	A Typical Federated Learning System . . . . .	3
1.2.1	Problem Formulation . . . . .	3
1.2.2	Federated Averaging: A Typical Federated Training Process . . . . .	4
1.2.3	Review of Theoretical Results of Federated Learning . . . . .	4
1.3	Fully-Decentralized Federated Learning . . . . .	9
1.4	Challenges and Open Problems in Federated Learning . . . . .	9
1.4.1	Statistical, System, and Temporal Heterogeneity . . . . .	10
1.4.2	Other Challenges . . . . .	12
1.5	Summary of the Main Contributions of the Thesis . . . . .	13
1.6	Additional Contributions . . . . .	14
1.6.1	The Role of Reference Data in Empirical Privacy Defenses . . . . .	14
1.6.2	FLamby: Datasets and Benchmarks for Cross-Silo FL . . . . .	18
1.7	Publications . . . . .	20
1.7.1	Published . . . . .	20
1.7.2	Submitted . . . . .	21
<b>2</b>	<b>System Considerations in Heterogeneous Federated Learning</b>	<b>23</b>
2.1	Throughput-Optimal Topology Design for Cross-Silo Federated Learning . . . . .	23
2.1.1	Introduction . . . . .	24
2.1.2	Problem Formulation . . . . .	25
2.1.3	Theoretical Results and Algorithms . . . . .	28
2.1.4	Numerical Experiments . . . . .	32
2.1.5	Conclusion . . . . .	41
2.2	Federated Learning under Heterogeneous and Correlated Client Availability . . . . .	42
2.2.1	Introduction . . . . .	42
2.2.2	Background and Related Works . . . . .	43
2.2.3	Analysis . . . . .	46
2.2.4	Proposed Algorithm . . . . .	50
2.2.5	Fairness, and Computational Cost of CA-Fed . . . . .	52
2.2.6	Experimental Evaluation . . . . .	53
2.2.7	Conclusion . . . . .	58
<b>3</b>	<b>Personalized Federated Learning</b>	<b>59</b>
3.1	Introduction . . . . .	59
3.1.1	Contributions . . . . .	60
3.1.2	Organization . . . . .	61
3.2	Related Work . . . . .	62
3.2.1	Statistical Heterogeneity . . . . .	62

3.2.2	System Heterogeneity . . . . .	63
3.3	Problem Formulation . . . . .	64
3.4	An Impossibility Result . . . . .	64
3.5	Personalized Federated Learning under a Mixture of Distributions . . . . .	65
3.5.1	The Mixture Assumption . . . . .	65
3.5.2	Relation with Other Personalized Federated Learning Frameworks . . . . .	66
3.5.3	Federated Expectation-Maximization . . . . .	68
3.5.4	Federated Surrogate Optimization . . . . .	73
3.5.5	Distributed Surrogate Optimization with Black-Box Solver . . . . .	76
3.5.6	Numerical Experiments . . . . .	78
3.5.7	Conclusion . . . . .	82
3.6	Personalized Federated Learning through Local Memorization . . . . .	82
3.6.1	kNN-Per Algorithm . . . . .	83
3.6.2	Generalization Bound . . . . .	84
3.6.3	Numerical Experiments . . . . .	86
3.6.4	Conclusion . . . . .	94
3.7	A Comparison between FedEM and kNN-Per . . . . .	96
<b>4</b>	<b>Federated Learning in Dynamic Environments</b>	<b>97</b>
4.1	Introduction . . . . .	97
4.1.1	Contributions . . . . .	99
4.1.2	Organization . . . . .	99
4.2	Related Work . . . . .	100
4.3	Federated Learning for Data Streams . . . . .	101
4.3.1	Problem Formulation . . . . .	102
4.3.2	Federated Learning Meta-Algorithm for Data Streams . . . . .	103
4.3.3	Case Study . . . . .	108
4.3.4	Numerical Experiments . . . . .	110
4.3.5	Conclusion . . . . .	115
4.4	Online Federated Learning with Mixture Models . . . . .	116
4.4.1	Problem Formulation . . . . .	116
4.4.2	FEM-OMD Algorithm . . . . .	118
4.4.3	Federated Online Learning with Gaussian Mixture Models . . . . .	119
4.4.4	FEM-OMD for Discriminative Models . . . . .	124
4.4.5	Experimental Results . . . . .	125
4.4.6	Conclusion and Perspectives . . . . .	128
<b>5</b>	<b>Conclusion</b>	<b>129</b>
5.1	Summary of the Main Contributions . . . . .	129
5.2	Perspectives and Future Research Directions . . . . .	133
5.3	Concluding Reflections . . . . .	135
	<b>Bibliography</b>	<b>137</b>
	<b>List of Figures</b>	<b>173</b>

## Appendix

A	Background on Numeric Optimization . . . . .	183
A.1	Differentiability . . . . .	183
A.2	Lipschitzianity and Smoothness . . . . .	183
A.3	Convexity . . . . .	184
B	Background on Graph Theory . . . . .	185
C	Throughput-Optimal Topology Design for Cross-Silo Federated Learning . . . . .	187
C.1	Proofs . . . . .	187
C.2	Additional Experiments . . . . .	194
D	Federated Learning under Heterogeneous and Correlated Client Availability . . . . .	202
D.1	Proof of Theorem 2.2.2 . . . . .	202
D.2	Proof of Theorem 2.2.3 . . . . .	204
D.3	Proof of Theorem 2.2.4 . . . . .	226
D.4	Convexity of $\bar{e}_{\text{opt}} + \bar{e}_{\text{bias}}$ . . . . .	227
D.5	Minimizing $\bar{e}_{\text{opt}}$ . . . . .	229
D.6	Background on Markov Chains . . . . .	231
D.7	Details on Experimental Setup . . . . .	234
E	Personalized Federated Learning under a Mixture of Distributions . . . . .	237
E.1	Proof of Proposition 3.5.1 . . . . .	237
E.2	Proofs for Centralized Expectation Maximization . . . . .	241
E.3	Proofs for Client-Server Setting . . . . .	244
E.4	Proofs for Fully Decentralized Setting . . . . .	260
E.5	Proof of Theorem 3.5.5' . . . . .	275
E.6	Proof of Theorem 3.5.5 . . . . .	277
E.7	Supporting Lemmas . . . . .	277
E.8	Additional Experiments . . . . .	285
E.9	Fully Decentralized Federated Expectation-Maximization . . . . .	285
E.10	Comparison with MOCHA . . . . .	286
E.11	Generalization to Unseen Clients . . . . .	286
E.12	FedEM and Clustering . . . . .	287
E.13	Effect of $M$ in Time-Constrained Setting . . . . .	288
E.14	Additional Results under Client Sampling . . . . .	289
E.15	Convergence Plots . . . . .	289
F	Personalized Federated Learning through Local Memorization . . . . .	296
F.1	Proof of Theorem 3.6.1 . . . . .	296
F.2	Intermediate Lemmas . . . . .	297
G	Federated Learning for Data Streams . . . . .	300
G.1	Proofs . . . . .	300
G.2	Proof of Lemma 4.3.2 . . . . .	305
G.3	Bound $\bar{\sigma}^2(\lambda)$ . . . . .	312
G.4	Case Study . . . . .	315
H	Online Federated Learning with Mixture Models . . . . .	323
H.1	Proof of Theorem 4.4.2 . . . . .	323

H.2	Proof of Theorem 4.4.3 . . . . .	323
H.3	Proof of Theorem 4.4.5 . . . . .	324
H.4	Supporting Lemmas . . . . .	327

# CHAPTER 1

---

## Introduction

### 1.1 Motivation

The increasing size of data generated by smartphones and IoT devices motivated the development of *Federated Learning* (FL) [Kon+17a; McM+17], a framework allowing geographically distributed clients to jointly learn machine learning (ML) models, without the need to share their own local data.

The term *federated learning* was introduced in 2016\* in the seminal works Konečný et al. [Kon+17a] and McMahan et al. [McM+17]: “We investigate a learning technique that allows users to collectively reap the benefits of shared models trained from this rich data, without the need to centrally store it. We term our approach *Federated Learning*, since the learning task is solved by a loose federation of participating devices (which we refer to as *clients*) which are coordinated by a central *server*.”

Federated learning distinguishes itself through a crucial feature: it ensures that the data held by each client remains securely stored locally, avoiding any exchange or transfer of sensitive information. This approach to local data storage brings forth two compelling promises, each with its unique set of advantages. Firstly, the emphasis on local data storage significantly bolsters privacy protection. By keeping data confined to its originating device, federated learning reduces the potential attack surface of the system. This, in turn, minimizes the risk of data breaches and unauthorized access, aligning with the fundamental principle of data minimization in privacy and security practices [Par16]. Secondly, the local data storage approach also leads to notable efficiencies in communication resources. Since data does not need to be replicated in a centralized cloud server, the burden on network bandwidth is substantially reduced. This reduction in data transmission requirements not only alleviates congestion and latency issues but also conserves valuable network resources, ultimately resulting in faster and more cost-effective data processing.

The concept of federated learning, as elucidated previously, has attracted a surge of attention from both the academic and industrial spheres. This field has witnessed a remarkable evolution, transitioning from a mere handful of papers in 2016 to an impressive influx of over 17,000 new publications incorporating the term “federated learning” in the year 2022 (as illustrated in Figure 1.1). Concurrently, in the industrial landscape, federated learning is gaining significant momentum as organizations increasingly recognize its potential to revolutionize data-driven decision-making processes. Major tech giants like Google have embraced federated learning in various applications, such as Gboard’s next word prediction [Har+19], emoji suggestion [Ram+19], and out-of-vocabulary

---

\*Reference [McM+17] was initially disseminated on arXiv on the 17<sup>th</sup> of February in 2016, preceding its appearance in the Proceedings of the 20<sup>th</sup> International Conference on Artificial Intelligence and Statistics (AISTATS) in 2017. On the other hand, Reference [Kon+17a] was first made public on arXiv on the 18<sup>th</sup> of October in 2016, with a subsequent version uploaded on the 30<sup>th</sup> of October in 2017



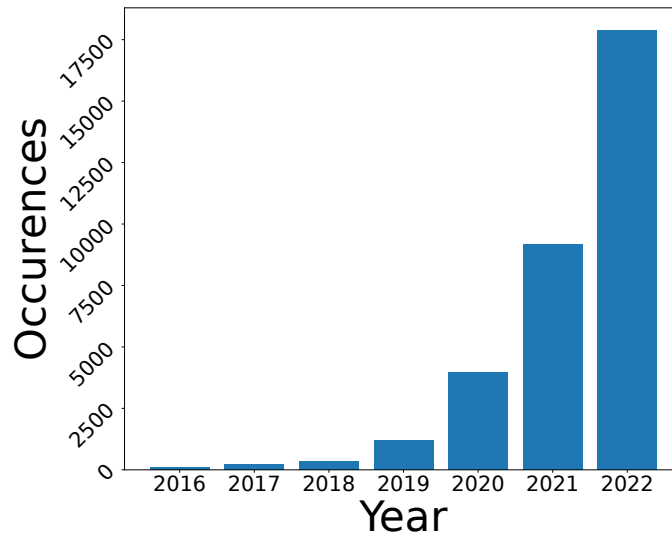


Figure 1.1: Occurrences of the key word “federated learning” over time in academic papers (from Google Scholar). The results were obtained using the code from <https://github.com/Pold87/academic-keyword-occurrence> [Str18]

word discovery [Che+19]. Apple has also integrated federated learning into iOS 13 for key functions like the QuickType keyboard and the voice classifier for "Hey Siri" [App19]. Beyond the realm of mobile applications, federated learning finds application in diverse sectors, such as the banking industry, where institutions like WeBank employ it for money laundering detection [WeB19], and in healthcare, where it is harnessed to tackle critical and poorly understood diseases like triple-negative breast cancer [Ter+21] and the ongoing challenges posed by COVID-19 [Day+21].

Heterogeneity is a core and fundamental challenge in federated learning [Li+20a]. Indeed, clients highly differ both in size and distribution of their local datasets (*statistical heterogeneity*), and in their storage, computational, and communication capabilities (*system heterogeneity*). Moreover, local datasets may vary over time, an aspect that has been neglected until now and we call *temporal heterogeneity*. Heterogeneity often hinders the performance of federated learning. Statistical heterogeneity in general slows convergence down and may lead to unfair models, unsuited for minorities. System heterogeneity, if ignored, may lead to intolerably long training period or to models too simple for the most powerful clients, and, if naively addressed, to biased final models. Similarly, ignoring temporal data-access heterogeneity limits the possibility to use federated learning in online and dynamic settings, but simple solutions introduce bias in the final model.

Given the set of challenges related to the heterogeneity in federated learning, this thesis aims at providing theoretical understanding of distributed and federated learning systems. Inspired by the theoretical insights, we seek to design large-scale distributed/federated learning algorithms that can efficiently exploit data and system resources. Ultimately, this thesis delves into the multifaceted sources of heterogeneity within FL and introduces practical algorithms endowed with provable theoretical guarantees, aimed at mitigating the adverse effects of heterogeneity on the learning process.

In the rest of this chapter, we describe a typical (centralized) federated learning system in

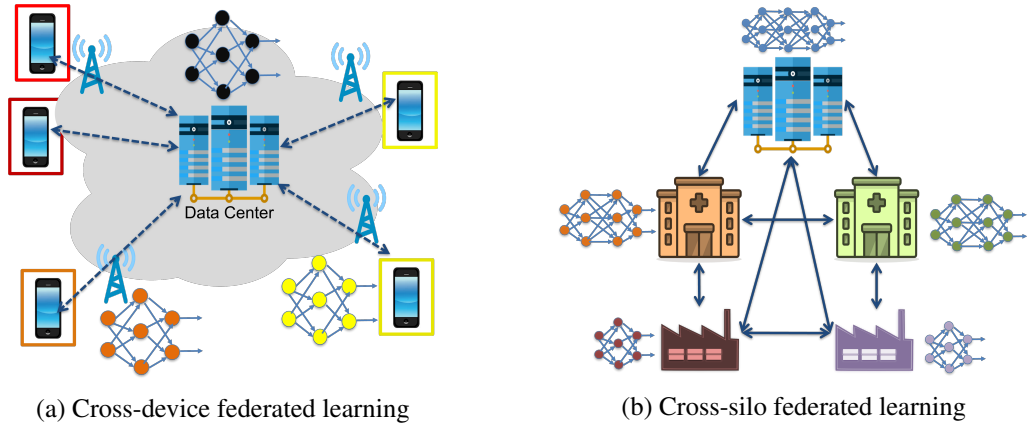


Figure 1.2: Federated learning system. **Left:** the cross-device scenario includes a large number of unreliable mobile devices with limited computing resources and slow Internet connections; it requires a client server architecture where mobiles communicate only with the server. **Right:** the cross-silo scenario includes at most a few hundred reliable data silos with powerful computing resources and high-speed access links; it may take advantage of peer-to-peer communications.

Section 1.2, and a fully-decentralized one in Section 1.3, before providing an overview of the main challenges and open problems in federated learning, Section 1.4. To provide a roadmap for readers, we summarize the principal contributions of this manuscript and outline the forthcoming chapters in Section 1.5. Additionally, we present an overview of two supplementary contributions not exhaustively expounded upon in this manuscript in Section 1.6. Lastly, we compile a list of the related publications in Section 1.7.

## 1.2 A Typical Federated Learning System

In this section, we describe the original federated learning system introduced in [McM+17], usually referred to as *cross-device* federated learning (as depicted in Figure 1.2a). The system includes a large number of unreliable mobile devices with limited computing resources and slow Internet connections; it requires a client server architecture where mobiles communicate only with the server.

### 1.2.1 Problem Formulation

The canonical federated learning formulation in [McM+17] involves learning a *single, global* statistical model from data stored on a finite number  $C > 0$  of remote clients. In particular, the goal is typically to minimize the following objective function:

$$\min_{\mathbf{w} \in \mathbb{R}^d} F(\mathbf{w}) = \sum_{c=1}^C \frac{N_c}{N} F_c(\mathbf{w}). \quad (1.1)$$

Here  $F_c$  is the local objective function for the  $c$ -th device,  $N_c$  is the number of samples available locally at client  $c \in [C]$ , and  $N = \sum_{c=1}^C N_c$ . The local objective function is often defined as the

empirical risk over the local dataset, i.e.,

$$F_c(\mathbf{w}) = \frac{\sum_{(\mathbf{x}, y) \in \mathcal{S}_c} \ell(\mathbf{w}; \mathbf{x}, y)}{N_c}, \quad (1.2)$$

where  $\ell(\mathbf{w}; \mathbf{x}, y)$  is the loss induced by model  $\mathbf{w}$  on the example  $(\mathbf{x}, y)$ , and  $\mathcal{S}_c$  is the local data of client  $c$ .

**Remark 1.** *The global objective  $F$  in (1.1) could be rewritten as follows:*

$$F(\mathbf{w}) = \sum_{c=1}^C \frac{N_c}{N} F_c(\mathbf{w}) = \sum_{c=1}^C \frac{N_c}{N} \times \frac{\sum_{(\mathbf{x}, y) \in \mathcal{S}_c} \ell(\mathbf{w}; \mathbf{x}, y)}{N_c} = \frac{\sum_{(\mathbf{x}, y) \in \mathcal{S}} \ell(\mathbf{w}; \mathbf{x}, y)}{N}, \quad (1.3)$$

Therefore,  $F$  could be interpreted as the empirical loss associated with the aggregated dataset  $\mathcal{S} \triangleq \bigcup_{c=1}^C \mathcal{S}_c$ .

## 1.2.2 Federated Averaging: A Typical Federated Training Process

A typical algorithm to solve (1.1) is federated averaging (FedAvg), first proposed in [McM+17]. FedAvg (Algorithm 1) is an iterative algorithm that divides the training process into  $T > 0$  communication rounds. At the beginning of the  $t$ -th communication round, the server selects a set of clients meeting eligibility requirements (Line 3). Specifically, for mobile phones, a device is typically considered eligible if it is currently plugged in, connected to an unmetered WiFi network, and idle, as described in [Kai+21, Section 1.1.2]. Then, the server broadcasts the current model  $\mathbf{w}_t$  to the selected clients  $\mathcal{C}_t$  (Line 4). Upon reception of the model  $\mathbf{w}_t$ , each client  $c \in \mathcal{C}_t$  locally updates the model, usually through a finite number of local stochastic gradient descent (SGD) updates, using its local dataset  $\mathcal{S}_c$  (Line 6). Afterwards, the client sends-back the resulting model  $\mathbf{w}_{t+1}^{(c)}$  to the server (Line 7). Finally, the server aggregates the local update models  $\mathbf{w}_{t+1}^{(c)}$  in order to produce a new global model  $\mathbf{w}_{t+1}$  (Line 9).

The FedAvg algorithm can be extended to a versatile framework known as FedOpt [Red+21] (refer to Algorithm 2), which grants the algorithm designer the flexibility to modify the client selection protocol, the client local update rule, the aggregation method, or the server global update rule. FedOpt maintains the same fundamental structure as FedAvg, but it incorporates two significant distinctions. Firstly, each client transmits the local model change  $\Delta_t^{(k)}$  to the server, as opposed to sending the model itself (see Line 7). Secondly, the server leverages the negative of the aggregated local changes, denoted as  $-\Delta_t$ , as a pseudo-gradient and applies it to the global model, rather than aggregating the gradients (see Line 10). In the original FedAvg algorithm, the default settings implicitly configure `ServerUpdate` and `ClientUpdate` to be Stochastic Gradient Descent (SGD), with a fixed server learning rate of  $\eta_s = 1.0$ . FedOpt is a widely used framework for describing and analyzing federated training processes, as illustrated in a recent survey by Wang et al. [Wan+21a].

## 1.2.3 Review of Theoretical Results of Federated Learning

The purpose of this manuscript is to provide theoretical understanding of distributed and federated learning systems. In particular, we are interested in two types of results: *optimization results*, which focus on the behavior and convergence of the learning algorithms, and *generalization results*, which assess the model’s performance on unseen data. In this section, we give an overview of some known convergence and generalization results for federated learning.

---

**Algorithm 1:** FedAvg: Federated Averaging [McM+17, Algorithm 1].

---

**Input** : Data  $\mathcal{S}_{1:C}$ ; number of communication rounds  $T$ ; number of local epochs  $E$ ;  
learning rate  $\eta$

- 1 **server** randomly initialize  $\mathbf{w}_1$ ;
- 2 **for**  $t = 1, \dots, T$  **do**
- 3     **server** selects a subset  $\mathcal{C}_t$  of clients ;
- 4     **server** broadcast  $\mathbf{w}_t$  to the selected clients  $\mathcal{C}_t$ ;
- 5     **for each client**  $c \in \mathcal{C}_t$  **in parallel do**
- 6          $\mathbf{w}_{t+1}^{(c)} \leftarrow \text{ClientUpdate}(\mathbf{w}_t, \mathcal{S}_c, E)$  ;
- 7         **client** sends  $\mathbf{w}_{t+1}^{(c)}$  to the **server** ;
- 8     **end**
- 9     **server** aggregates clients' updates:  $\mathbf{w}_{t+1} \leftarrow \sum_{c \in \mathcal{C}_t} (N_c/N) \cdot \mathbf{w}_{t+1}^{(c)}$ ;
- 10 **end**
- 11 **Function**  $\text{ClientUpdate}(\mathbf{w}, \mathcal{S}, E)$  :
- 12     **for**  $e = 1, \dots, E$  **do**
- 13         Sample indexes  $\mathcal{I}$  uniformly from  $1, \dots, |\mathcal{S}|$ ;
- 14          $\mathbf{w} \leftarrow \mathbf{w} - \eta \sum_{i \in \mathcal{I}} \ell(\mathbf{w}; \mathbf{x}_i, y_i)$ ;
- 15     **end**
- 16     **return**  $\mathbf{w}$ ;

---

### 1.2.3.1 An Optimization Result

Federated optimization algorithms have been extensively studied, with a substantial body of literature dedicated to their analysis, as highlighted in several notable works [Li+20c; Sti19; KMR20; WJ21]. For a comprehensive overview of this research landscape, readers can refer to the recent surveys by Wang et al. [Wan+21a]. In the subsequent section, we will leverage the FedOpt framework to explore and discuss key theoretical tools frequently employed in the convergence analysis of the vanilla FedAvg algorithm.

For simplicity, in this section, we suppose that each client contributes the same number of samples, i.e.,  $N_c \equiv N/C$ , and participates at every round, i.e.,  $\mathcal{C}_t \equiv [C]$ . Therefore, the aggregation step (Line 9 in Algorithm 2) becomes  $\Delta_t = \Delta_t^{(c)}/C$ . Moreover, we assume that the local objective function  $F_c$  are convex and  $L$ -smooth, as defined in Appendix A. Additionally, we consider a simplified instance of FedOpt, where the server-update takes a unit descent step, i.e.,  $\mathbf{w}_{t+1} = \mathbf{w}_t + \Delta_t$ , and the client update consists in  $E$  local steps of SGD with constant learning rate  $\eta > 0$ , i.e.,

$$\mathbf{w}_{t,1}^{(c)} = \mathbf{w}_t, \quad (1.4)$$

$$\mathbf{w}_{t,e+1}^{(c)} = \mathbf{w}_{t,e}^{(c)} - \eta \cdot \mathbf{g}_c(\mathbf{w}_{t,e}^{(c)}); \quad e = 1, \dots, E-1, \quad (1.5)$$

$$\mathbf{w}_{t+1}^{(c)} = \mathbf{w}_{t,E}^{(c)}. \quad (1.6)$$

Here  $\mathbf{g}_c$  is the stochastic gradient of  $F_c$ , that is usually assumed to verify the following unbiasedness and bounded variance property:

**Assumption 1.** (Unbiased gradients and bounded variance) Each client  $c \in [C]$  can compute an

---

**Algorithm 2:** FedOpt Algorithm [Red+21, Algorithm 1].

---

**Input** : Data  $\mathcal{S}_{1:C}$ ; ClientUpdate (); ServerUpdate (); Aggregate ()

- 1 **server** randomly initialize  $\mathbf{w}_1$ ;
- 2 **for**  $t = 1, \dots, T$  **do**
- 3     **server** selects a subset  $\mathcal{C}_t$  of clients ;
- 4     **server** broadcast  $\mathbf{w}_t$  to the selected clients  $\mathcal{C}_t$ ;
- 5     **for each client**  $c \in \mathcal{C}_t$  **in parallel do**
- 6          $\mathbf{w}_{t+1}^{(c)} \leftarrow \text{ClientUpdate}(\mathbf{w}_t, \mathcal{S}_c)$  ;
- 7         **client** computes local model change  $\Delta_t^{(c)} = \mathbf{w}_{t+1}^{(c)} - \mathbf{w}_t$  and sends it to the **server** ;
- 8     **end**
- 9     **server** aggregates clients' updates:  $\Delta_t \leftarrow \text{Aggregate}(\{\Delta_t^{(c)}, c \in \mathcal{C}_t\})$  ;
- 10    **server** updates global model:  $\mathbf{w}_{t+1} \leftarrow \text{ServerUpdate}(\mathbf{w}_t, -\Delta_t)$  ;
- 11 **end**

---

unbiased estimator  $\mathbf{g}_c$  of the local gradient with bounded variance, i.e.,

$$\mathbb{E} \left[ \mathbf{g}_c(\mathbf{w}_{t,e}^{(c)}) \mid \mathbf{w}_{t,e}^{(c)} \right] = \nabla F_c(\mathbf{w}_{t,e}^{(c)}), \quad \mathbb{E} \left[ \left\| \mathbf{g}_c(\mathbf{w}_{t,e}^{(c)}) - \nabla F_c(\mathbf{w}_{t,e}^{(c)}) \right\|^2 \mid \mathbf{w}_{t,e}^{(c)} \right] \leq \sigma^2. \quad (1.7)$$

In contrast to the conventional SGD algorithm, which updates model parameters by iteratively moving them in the direction opposite to the current stochastic gradient, vanilla FedAvg updates the global model parameters with a single unitary step in the direction of the aggregated local changes  $\Delta_t$ . The vector  $\Delta_t$  is the result of multiple local SGD iterates performed by each client. To handle iterates from multiple clients, a concept of shadow sequence is introduced [Lia+17; YLY16; Wan+21a; Sti19], and defined as:  $\bar{\mathbf{w}}_{t,e} \triangleq \sum_{c=1}^C \mathbf{w}_{t,e}^{(c)} / C$ . Given this notation, we have for  $e \in \{1, \dots, E-1\}$

$$\bar{\mathbf{w}}_{t,e+1} = \bar{\mathbf{w}}_{t,e} - \frac{\eta}{K} \sum_{c=1}^C \mathbf{g}_c(\mathbf{w}_{t,e}^{(c)}). \quad (1.8)$$

In light of (1.8), we observe that the average iterate  $\bar{\mathbf{w}}_{t,e+1}$  performs a perturbed stochastic gradient descent, where the gradient is evaluate at  $\mathbf{w}_{t,e}^{(c)}$  instead of  $\bar{\mathbf{w}}_{t,e}$ . If the distance between  $\mathbf{w}_{t,e}^{(c)}$  and  $\bar{\mathbf{w}}_{t,e+1}$  is uniformly bounded, then one proves that the vanilla-FedAvg algorithm is in expectation making progress at each round, as quantified by Lemma 1.2.1

**Lemma 1.2.1.** (Per round progress [Wan+21a, Lemma 1]) *If the learning rate satisfies  $\eta \leq 1/4L$ , then*

$$\begin{aligned} \mathbb{E} \left[ \frac{1}{E} \sum_{e=1}^E F(\bar{\mathbf{w}}_{t,e}) - F(\mathbf{w}^*) \right] &\leq \frac{1}{2\eta E} (\|\mathbf{w}_t - \mathbf{w}^*\| - \|\mathbf{w}_{t+1} - \mathbf{w}^*\|) + \frac{\eta\sigma^2}{C} \\ &\quad + \frac{L}{CE} \sum_{c=1}^C \sum_{e=1}^E \mathbb{E} \left\| \mathbf{w}_{t,e}^{(c)} - \bar{\mathbf{w}}_{t,e+1} \right\|^2, \end{aligned} \quad (1.9)$$

where  $\mathbf{w}^*$  is a minimizer of  $F$ .

The result presented in Lemma 1.2.1 closely resembles the convergence outcome of the standard (centralized) SGD algorithm. The key distinction lies in the additional term on the right-hand side

of equation (1.9), which introduces  $L/CE \sum_{c=1}^C \sum_{e=1}^E \mathbb{E} \|\mathbf{w}_{t,e}^{(c)} - \bar{\mathbf{w}}_{t,e}\|^2$ . This term captures the divergence between each client's local iterate  $\mathbf{w}_{t,e}^{(c)}$  and the shadow iterate  $\bar{\mathbf{w}}_{t,e}$ . Fortunately, under the assumption that the local objective functions exhibit limited dissimilarity (as stipulated by Assumption 2), all client iterates tend to remain in close proximity to the global average.

**Assumption 2.** (*Bounded dissimilarity*) *The difference of local gradient  $\nabla F_c$  and the global gradient  $\nabla F$  is  $B$ -uniformly bounded, i.e.,*

$$\max_c \sup_{\mathbf{w}} \|\nabla F_c(\mathbf{w}) - \nabla F(\mathbf{w})\| \leq B. \quad (1.10)$$

When Assumption 2 holds, the client drift is bounded, as shown by Lemma 1.2.2.

**Lemma 1.2.2.** (*Bounded client drift [Wan+21a, Lemma 2]*) *Assuming the client learning rate satisfies  $\eta \leq 1/4L$ ,*

$$\forall c \in [C], \quad \max_e \mathbb{E} \|\mathbf{w}_{t,e}^{(c)} - \bar{\mathbf{w}}_{t,e}\|^2 \leq 18E^2\eta^2B^2 + 4E\eta^2\sigma^2. \quad (1.11)$$

Combining Lemmas 1.2.1 and 1.2.2 and telescoping  $t$  for 1 to  $T$ , we obtain the main convergence theorem for vanilla FedAvg:

**Theorem 1.2.3.** [*Wan+21a, Theorem 1*] *Assuming the client learning rate satisfies  $\eta \leq 1/4L$ ,*

$$\mathbb{E} \left[ \frac{1}{ET} \sum_{t=1}^T \sum_{e=1}^E F(\bar{\mathbf{w}}_{t,e}) - F(\mathbf{w}^*) \right] \leq \frac{\|\mathbf{w}_1 - \mathbf{w}^*\|^2}{2\eta ET} + \frac{\eta\sigma^2}{C} + 4E\eta^2L\sigma^2 + 18E^2\eta^2LB^2. \quad (1.12)$$

Furthermore, when the client learning rate is chosen as

$$\eta = \min \left\{ \frac{1}{4L}, \frac{\sqrt{C} \|\mathbf{w}_1 - \mathbf{w}^*\|}{\sqrt{ET}\sigma}, \frac{\|\mathbf{w}_1 - \mathbf{w}^*\|^{2/3}}{E^{2/3}T^{1/3}L^{1/3}\sigma^{2/3}}, \frac{\|\mathbf{w}_1 - \mathbf{w}^*\|^{2/3}}{E^{2/3}T^{1/3}L^{1/3}B^{2/3}} \right\}, \quad (1.13)$$

we have

$$\mathbb{E} \left[ \frac{1}{ET} \sum_{t=1}^T \sum_{e=1}^E F(\bar{\mathbf{w}}_{t,e}) - F(\mathbf{w}^*) \right] = \mathcal{O} \left( \frac{L}{ET} + \frac{\sigma}{\sqrt{CET}} + \frac{L^{1/3}\sigma^{2/3}}{C^{1/3}T^{2/3}} + \frac{L^{1/3}B^{2/3}}{T^{2/3}} \right). \quad (1.14)$$

### 1.2.3.2 A Generalization Result

As previously discussed, the fundamental aim of federated learning is to enable clients to collectively harness the advantages of shared models trained on the abundant data amassed collectively [McM+17]. The underlying promise is that through collaborative efforts, each client can derive benefits from the data holdings of all other participating clients. This naturally prompts the question of whether it is indeed advantageous for a client to engage in global model training, a pursuit pursued by many federated learning algorithms, including FedAvg [McM+17], FedProx [Li+20b], and SCAFFOLD [Kar+20a]. To address this question, our aim is to conduct a comparative analysis of the generalization capabilities of the global model, trained on data from all clients, and of the local model trained exclusively on a specific client's data. A similar discussion on this topic can be found in [Man+20, Section 2].

We start with some general notation used throughout this section. Let  $\mathcal{X}$  and  $\mathcal{Y}$  denote the input and output space, respectively. Let  $\mathcal{H}$  be a hypothesis class of functions mapping  $\mathcal{X}$  to

$\mathcal{Y}$ , and let  $d_{\mathcal{H}}$  is the pseudo-dimension of the hypothesis class  $\mathcal{H}$  [MRT18, Chapter 11]. Note that pseudo-dimension coincides with the VC dimension for the 0 – 1 loss. A given client, say  $k \in [K]$ , has access to a dataset  $\mathcal{S}_c$  of examples *independently and identically distributed* (i.i.d.) according to a probability distribution  $\mathcal{P}_c$  over  $\mathcal{X} \times \mathcal{Y}$ , i.e.,  $\mathcal{S}_c \stackrel{\text{i.i.d.}}{\sim} \mathcal{P}_c^{N_c}$ . The client  $c$  wants to learn a hypothesis/model minimizing its own population risk, defined as

$$\mathcal{L}_c(h) \triangleq \mathcal{L}_{\mathcal{P}_c}(h) \triangleq \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{P}_c} [\ell(h(\mathbf{x}), y)], \quad (1.15)$$

where  $\ell(h(\mathbf{x}), y)$  is the loss induced by the hypothesis  $h \in \mathcal{H}$  on the example  $(\mathbf{x}, y)$ . Note however that this expectation cannot be computed by a given client since it requires full knowledge of the data distribution  $\mathcal{P}_c$ , which is usually unknown. Instead the client can compute an empirical estimation defined as

$$\hat{\mathcal{L}}_c(h) \triangleq \mathcal{L}_{\mathcal{S}_c}(h) \triangleq \frac{1}{N_c} \sum_{i=1}^{N_c} \ell(h(\mathbf{x}_i), y_i). \quad (1.16)$$

The purely local model of client  $c$  is obtained by minimizing the empirical risk  $\hat{\mathcal{L}}_c(h)$  associated with client  $c$ . We use  $\hat{h}_c$  to denote such model. By standard statistical learning theoretic tools [MRT18], the generalization performance of this model can be bounded as shown in Proposition 1.2.4.

**Proposition 1.2.4.** *Let  $\delta \in (0, 1)$ . With probability at least  $1 - \delta$ , the following holds:*

$$\mathcal{L}_c(\hat{h}_c) - \min_{h \in \mathcal{H}} \mathcal{L}_c(h) = \mathcal{O} \left( \sqrt{\frac{d_{\mathcal{H}} + \log 1/\delta}{N_c}} \right). \quad (1.17)$$

Proposition 1.2.4 underscores that the purely local model  $\hat{h}_c$  demonstrates strong generalization when the sample size  $N_c$  is sufficiently large. However, it's important to note that this favorable outcome is not always guaranteed. To address this challenge, conventional federated learning approaches adopt a different strategy: they train a global model  $\bar{h}$  by minimizing the empirical loss associated with the aggregated dataset  $\mathcal{S} = \bigcup_{c=1}^C \mathcal{S}_c$ , as elaborated in Remark 1. Proposition 1.2.5 bounds the generalization performance of the global model.

**Proposition 1.2.5.** [Man+20, Eq. 2] *Let  $\delta \in (0, 1)$ . With probability at least  $1 - \delta$ , the following holds:*

$$\mathcal{L}_c(\bar{h}) - \min_{h \in \mathcal{H}} \mathcal{L}_c(h) = \mathcal{O} \left( \sqrt{\frac{d_{\mathcal{H}} + \log 1/\delta}{N}} \right) + \text{disc}_{\mathcal{H}} \left( \mathcal{P}_c, \sum_{c'} \frac{N_{c'}}{N} \mathcal{P}_{c'} \right), \quad (1.18)$$

where  $\text{disc}_{\mathcal{H}}$  is the label discrepancy [MRT18] associated to the hypothesis class  $\mathcal{H}$  defined for two distributions over  $\mathcal{X} \times \mathcal{Y}$ ,  $\mathcal{P}_1$  and  $\mathcal{P}_2$  as:

$$\text{disc}_{\mathcal{H}}(\mathcal{P}_1, \mathcal{P}_2) = \max_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{P}_1}(h) - \mathcal{L}_{\mathcal{P}_2}(h)|. \quad (1.19)$$

Since the global model is trained on the concatenation of all users' data, it generalizes well. However, due to the distribution mismatch, the model may not perform well for a specific user. In particular if  $\mathcal{P}_c$  is substantially different from  $\bar{\mathcal{P}} (\equiv \sum_{c'} N_{c'}/N \mathcal{P}_{c'})$ , the second term in the RHS of (1.18) will be large.

### 1.3 Fully-Decentralized Federated Learning

As we have seen in Section 1.2, in federated learning, clients usually train the model through an iterative procedure under the supervision of a central orchestrator, which, for example, decides to launch the training process and coordinates training advances. Often—e.g., in FedAvg [McM+17], SCAFFOLD [Kar+20a], and FedProx [Li+20b]—the orchestrator directly participates to the training, by aggregating clients’ updates, generating a new model, and pushing it back to the clients. Hence, clients only communicate with a potentially far-away (e.g., in another continent) orchestrator and do not exploit communication opportunities with close-by clients. This choice is justified in the cross-device setting, where inter-device communication is unreliable (devices may drop-out from training at any time) and slow (a message needs to traverse two slow access links). But in the cross-silo setting (depicted in Figure 1.2b), data silos (e.g., data centers) are almost always available, enjoy high-speed connectivity comparable to the orchestrator’s one, and may exchange information faster with some other silos than with the orchestrator. An orchestrator-centered communication topology is then potentially inefficient, because it ignores fast inter-silo communication opportunities and makes the orchestrator a candidate for congestion. A current trend [Wan+19a; VBT17; Tan+18; Bel+18; Lia+17; Lia+18] is then to replace communication with the orchestrator by peer-to-peer communications between individual silos, which perform local partial aggregations of model updates. The approach of replacing communication with the server by peer-to-peer communication between individual clients is commonly referred to as *fully-decentralized federated learning* [Kai+21, Section 2.1].

In fully-decentralized federated learning, the clients are represented as vertices of a (connected) graph, usually referred to as the *communication topology*. In this decentralized learning approach, each client maintains a local copy of the model, iteratively updating it through one or a few local stochastic gradient steps. Subsequently, the client transmits its updated model to its out-neighboring nodes within the communication topology. Afterward, the client aggregates its model with those received from its in-neighboring nodes. In Section 2.1, we give more details on fully-decentralized federated learning, and we show that this approach has the potential to significantly speed-up the training in comparison to the server-client architecture.

### 1.4 Challenges and Open Problems in Federated Learning

As highlighted in Section 1.1, federated learning emerges as a promising solution to address critical concerns such as *privacy* [McM+18] and *environmental impact* [Qiu+23] associated with traditional centralized model training, which necessitates aggregating all data at powerful data centers for computation. Additionally, we recognize heterogeneity as a central challenge in federated learning, with statistical, system, and temporal data-access heterogeneity being the primary sources. While addressing heterogeneity stands as the primary focus of this manuscript, federated learning presents various other challenges. This section is dedicated to providing a concise overview of the core challenges encountered in federated learning and discussing ongoing efforts to overcome them. Several recent surveys, including [Kai+21; Wan+21a; Li+20a], offer comprehensive insights into the open challenges and unresolved issues within the field of federated learning.



## 1.4.1 Statistical, System, and Temporal Heterogeneity

Within this section, our attention is directed towards elucidating the challenges imposed by statistical, system, and temporal heterogeneity within the framework of federated learning. Our objective here is to underscore the noticeable gaps in research when it comes to addressing these heterogeneity sources and to offer clarity on how this manuscript adeptly addresses and bridges these gaps. In essence, this section serves as an exploration of the intricacies and shortcomings in current approaches to handling statistical, system, and temporal heterogeneity, while simultaneously highlighting the novel contributions and solutions provided by our manuscript to address these challenges effectively.

### 1.4.1.1 Statistical Heterogeneity

In federated learning, data is generated and gathered by clients with varying behaviors and preferences. Therefore, the local data of a particular client will not be representative of the population distribution. Further, the number of data points across devices may vary significantly. This imbalance and the fact that the data is not identically and independently distributed (non-IID) were introduced as fundamental challenges in federated learning. These challenges set it apart from traditional distributed optimization methods, as highlighted by the work of McMahan et al. [McM+17].

Statistical heterogeneity presents a dual challenge within the context of a federated learning system, manifesting in both convergence hurdles and the viability of a shared model for diverse clients.

On one hand, statistical heterogeneity hinders and slows down the convergence of federated learning algorithms, such as `FedAvg`, as shown by [Kar+20a; Li+19; Red+21; Li+20b]. In particular, [Kar+20a] shows that `FedAvg` suffers from “client drift” when the data is heterogeneous—i.e., when performing local updates from the same global model, clients will drift towards the minima of local objectives and end up with different local models, resulting in unstable and slow convergence. As a solution, Karimireddy et al. propose the `SCAFFOLD` algorithm which uses control variates (variance reduction) to correct for the client drift in its local updates. Li et al. [Li+20b] propose adding a proximal term to the objective in order to improve the stability of federated optimization, they name the approach `FedProx`. Both methods provide a principled way for the server to account for heterogeneity.

On the other hand, statistical heterogeneity challenges the assumption that clients should train a common model. In fact, as discussed in [SMS20], the existence of such a global model suited for all clients is at odds with the statistical heterogeneity observed across different clients. Consider for example a language modeling task: given the sequence of tokens “I love eating,” the next word can be arbitrarily different from one client to another. Moreover, in presence of statistical heterogeneity, a global model may be arbitrarily bad for some clients raising important fairness concerns [Li+21]. Thus, having personalized models for each client is a necessity in many FL applications. The few recent years have seen the development of a plethora of personalized federated learning techniques. We dedicate Chapter 3 to the discussion of personalized federated learning, and we introduce two novel personalized federated learning algorithm, namely `FedEM` (Section 3.5) and `kNN-Per` (Section 3.6).

### 1.4.1.2 System Heterogeneity

Within the domain of federated learning, clients exhibit a diverse range of characteristics, encompassing disparities in storage capacity, computational resources, and communication capabilities. These disparities arise from variations in hardware specifications (CPU power, memory capacity), network connectivity types (3G, 4G, 5G, WiFi), and power availability (battery levels) [Li+20a]. In this section, we delineate the challenges associated with system heterogeneity across three distinct scenarios: cross-silo, cross-device, and heterogeneous hardware environments.

In cross-silo scenarios, the communication capabilities between each data silo (e.g., data center) and the training orchestrator, as well as among different data silos, can exhibit significant variations from one silo to another. In this scenario, an orchestrator-centered communication topology is then potentially inefficient, because it ignores fast inter-silo communication opportunities and makes the orchestrator a candidate for congestion. As a response to this challenge, a current trend is to replace the conventional communication with the orchestrator with peer-to-peer communications among individual silos, as evidenced by recent works such as [Tan+18; Bel+18; Wan+19a; Yua+21; Yua+23], as we have discussed in Section 1.3. In this context, an important question arises: *How can we design a communication setup that allows for the fastest convergence, considering that different silos have different communication capabilities?* This question is the focus of Section 2.1. In Section 2.1, we formally define the problem of topology design for cross-silo federated learning using the theory of max-plus linear systems to compute the system throughput—number of communication rounds per time unit. We also propose practical algorithms that, under the knowledge of measurable network characteristics, find a topology with the largest throughput or with provable throughput guarantees.

In the cross-device settings, the system constraints affects the availability/activity of the clients. For instance, only smartphones that are idle, under charge, and connected to broadband networks are commonly allowed to participate in the training process [McM+17]. These system characteristics dramatically exacerbate challenges such as straggler mitigation and fault tolerance. The techniques to mitigate the stragglers problem could be grouped into three groups: *asynchronous communication* [Lia+18], *active device sampling* [NY19], and *fault tolerance* [XNS21]. Moreover, the heterogeneous clients participation patterns may introduce statistical bias if the less active clients have specific data characteristic. Previous effort on federated learning [Tan+22c; RVd23; Tan+22a; CHR22; Fra+21] considered this problem and, under different assumptions on the clients' availability, designed aggregation strategies that unbiased the federated updates through an appropriate choice of the aggregation weights. However, most of the aforementioned works ignore the temporal and spatial correlation in the clients' availability patterns. The *temporal correlation* may originate from a smartphone being under charge for a few consecutive hours and then ineligible for the rest of the day. The *spatial correlation* refers instead to correlation across different clients, which often emerges as consequence of users' different geographical distribution. For instance, clients in the same time zone often exhibit similar availability patterns, e.g., due to time-of-day effects. We dedicate Section 2.2 to the analysis of a FedAvg-like under heterogeneous and correlated client availability. Our analysis leads to the development of a Correlation-Aware FL (CA-Fed) algorithm.

In FL scenarios with highly heterogeneous hardware (like smartphones, IoT devices, edge computing servers, and the cloud), each client would ideally learn a different model architecture, suited to its capabilities. Common approaches to achieve this goal include knowledge distillation, sub-model training [Hor+21; DDT20], and collaboration through prototypes communication [Tan+22b]. Our  $k\text{NN-Per}$  algorithm (discussed in Section 3.6) offers a simple and efficient way to achieve

this goal by partially relieving the most powerful clients from the need to align their model to the weakest ones.

### 1.4.1.3 Temporal Heterogeneity

Federated learning [McM+17] usually involves the minimization of an objective function, which is only available through unbiased estimates of its gradients [BCN18]. The objective function is either the expected risk, when clients can sample new data points at every iteration, or the empirical risk, when they rely on a fixed dataset.

Most previous works on federated learning, e.g., [McM+17; Kon+17a], focus on the second case, i.e., the minimization of the empirical risk. They assume that clients operate in static environments and have access to identically distributed examples collected before training starts. Learning on static datasets can be sub-optimal (or even impossible) in many cases, because (1) new samples collected during training are ignored, and (2) clients may have limited memory capacities, and cannot store a large number of data samples. For example, nodes in a sensor network may continuously collect new measurements, but may be able to store only a few of them in the local memory [De +16]. Moreover, in many real-world applications, clients' underlying data distributions are non-stationary and constantly evolve. For instance, user sentiment and preference change drastically due to external environments such as the pandemic and macroeconomics [Koh+21; Gar+21].

Regrettably, there is a scarcity of work that systematically formalizes the intricacies of federated learning in the context of data streams, along with offering a comprehensive theoretical analysis. To the best of our knowledge, this deficiency is only addressed by a handful of exceptions, specifically, the studies conducted in [Che+20b], [Yoo+21], [OZ21], and [Jot+23]. In response to this research gap, we dedicate Chapter 4 to the investigation of federated learning within dynamic environments, where clients engage in collaborative learning from distributed data streams, characterized by the continuous generation of data. Our inquiry is particularly focused on two distinct scenarios, each corresponding to different assumptions regarding the data process. The first scenario delves into the case where samples within the data stream are independently drawn from an undisclosed fixed distribution. In the second scenario, we assume that client data distributions are mixtures of a finite number of undisclosed common underlying distributions, each varying over time in terms of mixing coefficients.

## 1.4.2 Other Challenges

In this section, we present a comprehensive overview of additional challenges in federated learning that extend beyond the realms of statistical, system, and temporal heterogeneity. Although these challenges are not the primary focal point of this manuscript, their significance cannot be understated. Effectively addressing these issues is imperative for the development of efficient federated learning systems. It is worth emphasizing that the challenges we delve into in this section often exhibit a degree of independence from the techniques we propose to mitigate the effects of statistical, system, and temporal heterogeneity. In essence, tackling these challenges operates in parallel with our primary focus, enhancing the overall robustness and effectiveness of federated learning frameworks.

### 1.4.2.1 Privacy

Federated learning, with its distributed model training across numerous edge devices, brings forth a set of intricate privacy challenges. The core principle of federated learning is to keep data localized

and secure, minimizing the need for centralized data aggregation. However, this very advantage introduces concerns regarding the privacy of sensitive information at the individual device level. One of the foremost concerns is the potential leakage of private data through model updates and gradients exchanged between the central server and participating devices [McM+17]. Federated learning systems may be vulnerable to various forms of attacks, including model inversion and membership inference attacks, which aim to extract confidential information about individual data contributors [NSH19a].

In order to limit these vulnerabilities, ongoing efforts are made to anonymize the model updates and gradients exchanged between the central server and participating devices. Addressing the privacy challenges in federated learning has spurred a significant body of research, reflecting the community’s commitment to preserving user confidentiality while harnessing the potential of decentralized machine learning. Prominent efforts include the development of advanced cryptographic techniques, such as Secure Multi-Party Computation (SMPC) [Yao86; Lap+16; Ara+16] and Homomorphic Encryption (HE) [Gen09; Bra12; CLT14], which allow model updates to be aggregated without exposing individual data points. While theoretically promising, their practical adoption faces limitations due to their computational intensity and the potential susceptibility to malicious attacks. Differential privacy mechanisms have also gained attention, offering a mathematical framework to quantify and control information leakage during federated learning iterations. Comprehensive surveys on these techniques and their application in federated learning can be found in recent surveys by [Kai+21] and [Yan+20a].

As federated learning continues to evolve, addressing these privacy challenges remains a pivotal research area, necessitating robust privacy-preserving mechanisms and encryption techniques to safeguard user data while reaping the benefits of decentralized machine learning [Kai+21].

#### 1.4.2.2 Expensive Communication

In federated learning, communication cost is often a critical bottleneck to scale up distributed optimization algorithms to collaboratively learn a model from millions of devices with potentially unreliable or limited communication and heterogeneous data distributions. The expensive communications in federated learning have sparked considerable interest in the development of compression schemes aimed at reducing the communication overhead [Had+21; Bez+22; PD21; KSJ19].

## 1.5 Summary of the Main Contributions of the Thesis

This thesis investigates different sources of heterogeneity in the context of federated learning, and proposes practical algorithms to mitigate the impact of heterogeneity.

In Chapter 2, we focus on tackling challenges associated with system heterogeneity in two scenarios: cross-silo and cross-device settings. In the cross-silo settings, Section 2.1 uses the theory of linear systems in the max-plus algebra to model the throughput, i.e., the number of completed rounds per time unit, of a fully decentralized cross-silo federated learning system. Afterwards, Section 2.1 proposes practical algorithms that, under the knowledge of measurable network characteristics, find a topology with the largest throughput or with provable throughput guarantees. In the cross-device settings, where the system constraints affects the availability/activity of the clients, clients may exhibit different levels of participation, often correlated over time and with other clients. In Section 2.2, we analyze a FedAvg-like algorithm under heterogeneous and correlated client availability. Our analysis highlights how correlation adversely affects the

algorithm’s convergence rate and how the aggregation strategy can alleviate this effect at the cost of steering training toward a biased model. Guided by the theoretical analysis, we propose Correlation-Aware FL (CA-Fed), a new FL algorithm that tries to balance the conflicting goals of maximizing convergence speed and minimizing model bias. To this purpose, CA-Fed dynamically adapts the weight given to each client and may ignore clients with low availability and large correlation.

In Chapter 3, we propose two personalized federated learning algorithms. The first algorithm, named FedEM, is a federated EM-like algorithm based on the flexible assumption that the dataset of each client is generated according to mixture of unknown common underlying distributions. The second algorithm, named kNN-Per, is based on local memorization; personalization is obtained by interpolating a collectively trained global model with a local k-nearest neighbors (kNN) model based on the shared representation provided by the global model. We provide theoretical guarantees for both algorithm; we provide convergence bounds for FedEM through a novel federated surrogate optimization framework, which can be of general interest, and we provide generalization bounds for kNN-Per.

In Chapter 4 we investigate federated learning for data streams (continuously generated data). Specifically, we focus on two scenarios corresponding to two different assumptions about the data process. The first focuses on the case where samples in the data stream are drawn independently from some fixed unknown distribution. In this case, we propose a general FL algorithm to learn from data streams through an opportune weighted empirical risk minimization. Our theoretical analysis provides insights to configure such an algorithm and shows a bias-optimization trade-off: by controlling the relative importance of older samples in comparison to newer ones; one can speed training up at the cost of a larger bias of the learned model or reduce the bias at the cost of a longer training time. In the second scenario, we assume that client’s data distributions are mixtures of a finite number of unknown common underlying distributions with varying mixing weights. In this case, we propose FEM-OMD, a federated variant of online mirror descent, where the gradient of the cost function is estimated through an EM-like algorithm at each time step. In the case of Gaussian mixture models, we show that the regret of FEM-OMD is asymptotically (in the sample size) sub-linear.

## 1.6 Additional Contributions

In addition to the contributions outlined in Section 1.5, which address the challenges of statistical, system, and temporal heterogeneity in federated learning, this thesis also encompasses two additional noteworthy contributions: a comprehensive analysis highlighting the role of reference data in empirical privacy defenses, and the creation of a novel cross-silo dataset suite tailored for healthcare applications. While these contributions are substantial, we have chosen not to dedicate separate chapters to them. This decision is based on the desire to maintain a streamlined and cohesive structure within the thesis, allowing for a more focused presentation of the main contributions while still providing essential insights into these supplementary aspects. Instead of dedicating separate chapters, we provide concise overviews of each in this section.

### 1.6.1 The Role of Reference Data in Empirical Privacy Defenses

*This Section is based on our work [Kap+24b], published at the 24th Privacy Enhancing Technologies Symposium (PETS’24).*

Data-driven applications, often using machine learning models, are proliferating throughout industry and society. Consequently, concerns about the use of data relating to individual persons has led to a growing body of legislation, most notably the European Union’s General Data Protection Regulation (GDPR) [Par16]. According to the GDPR principle of data minimization, it is necessary to reduce the degree to which data can be connected to individuals, even when that data is used for the purposes of training a statistical model [Par20]. It has therefore become important to ensure that a machine learning model is not leaking private information about its training data.

Membership inference attacks, which seek to discern whether or not a given data point has been used during training, have emerged as the de-facto standard for empirically measuring a machine learning model’s privacy leakage [Sho+17]. Indeed, inferring training dataset membership can be thought of as the most fundamental privacy violation. Although other attacks exist, such as model inversion [FJR15], property inference [Gan+18], dataset reconstruction [Sal+20], and model extraction [He+21; Kri+19; Tra+16], they all require a stronger adversary than is necessary to execute a membership inference attack.

Many methods have been proposed to defend against membership inference attacks. The use of differential privacy [DR+14] has emerged as a leading candidate for two reasons. First, it provides mathematically rigorous guarantees that upper-bound the influence a given data point can exert on the final machine learning model. Second, it is straightforward to integrate differential privacy into a machine learning model’s training procedure with algorithms such as differentially private gradient descent (DP-SGD) [Aba+16] or PATE [Pap+16]. Despite the many advantages associated with differential privacy, there are several key drawbacks that include: the significant degradation of model utility when using differential privacy during training [TB20], the difficulty of translating differential privacy’s theoretical privacy guarantees to real-world privacy leakage [Nas+21; Ber+19], and the fact that the decrease in accuracy resulting from differentially private training methods has been shown to more adversely affect underrepresented groups [BPS19; Uni+21; GOD22].

To address these issues, empirical privacy defenses (i.e., without theoretical privacy guarantees) have been developed to protect the privacy of training data against membership inference attacks. Existing empirical privacy defenses may be categorized by their method of protecting the training data (e.g., regularization [LLR21; NSH18], confidence-vector masking [Jia+19; Yan+20b], knowledge distillation [Tan+21]). Alternatively, one can group defenses by whether they use only the private training data [Tan+21] or require access to reference data [NSH18; LLR21; SH21; Jia+19; Yan+20b; Wan+20d], defined as additional data from the same underlying distribution [NSH18]. The two most prominent differentially private defenses can also be distinguished according to this distinction, where PATE [Pap+16] requires access to (unlabeled) reference data but DP-SGD [Aba+16] does not.

There are several problems with the current evaluation strategy of empirical privacy defenses. First, today’s best practice is to produce a utility-privacy curve that compares a model’s classification accuracy with its training data privacy for different values of a given defense parameter. Although this approach appears valid in the general case, assuming access to reference data makes the situation more complicated. This additional dataset may have its own privacy requirements, especially since it is assumed to come from the same underlying distribution as private training data. As gains in model utility and/or training data privacy may come at the expense of reference data privacy, it is only possible to meaningfully compare defenses when the relative level of privacy considerations between these two datasets is made explicit. In Figure 1.3, we present an example evaluation where

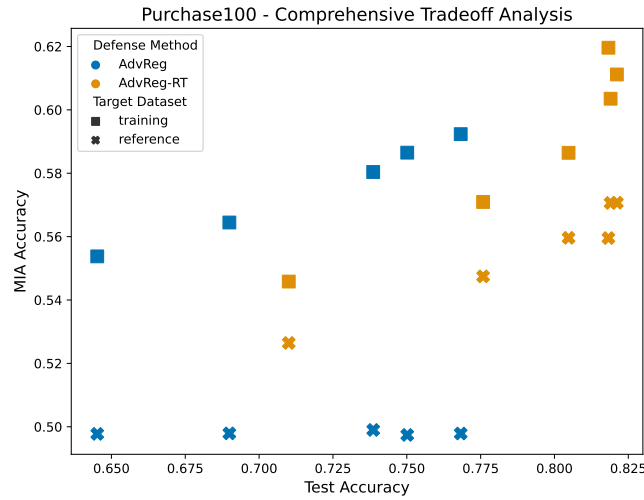


Figure 1.3: Tradeoff between a defended classifier’s prediction accuracy on test data (i.e., its model utility), membership inference attack accuracy on training data (i.e., training data privacy leakage), membership inference attack accuracy on reference data (i.e., reference data privacy leakage) for Purchase100 dataset. “AdvReg” corresponds to the original formulation of adversarial regularization [NSH18] and “AdvReg-RT” corresponds to a revisited version that we propose.

this issue becomes apparent. Looking only at the utility-privacy curves\* with respect to training data, it seems that AdvReg-RT is strictly better than AdvReg: for a target value of membership inference attack accuracy on the training data (resp. of test accuracy), AdvReg-RT is able to achieve a higher test accuracy (resp. lower membership inference attack accuracy). Alternatively, when the utility-privacy curves are examined with respect to both training and reference data, one cannot determine the better method without knowing their relative privacy considerations. Despite the necessity of measuring reference data privacy leakage to conduct a complete evaluation, as can be seen in Table 1.1, existing empirical privacy defenses are surprisingly reticent about this aspect.

A second problem with the current evaluation methodology of empirical privacy defenses is the lack of a well-understood and simple baseline. The literature contains several examples where proposed empirical privacy defenses have been later shown to leak significantly more training data privacy than originally reported and sometimes to even perform worse than simpler defenses [Cho+21; LLR21; SM21]. A well-established baseline could have provided more accurate expectations about such defenses. While early stopping [CLG00] has recently been suggested as a candidate to play such role [SM21], it does not utilize reference data and therefore fails to allow for a fair comparison with privacy defense techniques that exploit reference data. *Thus, there is a strong need for the development of a baseline designed to operate in the same assumption setting as the vast majority of existing empirical privacy defenses.*

**Contributions.** We introduce the notion of a training-reference data privacy tradeoff and conduct the first comprehensive investigation into how empirical privacy defenses perform with respect to all three relevant metrics: model utility, training data privacy leakage, and reference data privacy leakage. Given this evaluation setting, we propose a well-motivated baseline for

\*For the AdvReg and AdvReg-RT, the curves are obtained by changing the relative importance of the classification loss and the attacker loss [NSH18].

Table 1.1: Comparison of existing privacy defenses by reference data treatment. In the second column, “relative level unspecified” means the target level of relative privacy requirements between training and reference data is not stated. In the third column, “single privacy level” means the reference data privacy leakage is evaluated at a single point on the utility-privacy curve. We use a dashed line (—) to convey that the defense either does not use reference data (column 2) or does not need to evaluate reference data privacy leakage (column 3).

defense	reference data privacy setting	reference data privacy evaluation
Adversarial Regularization [NSH18]	not mentioned	no evaluation
MemGuard [Jia+19]	not mentioned	no evaluation
Model Pruning [Wan+20d]	not mentioned	no evaluation
MMD-based Regularization [LLR21]	private (relative level unspecified)	yes (single privacy level)
Distillation for Membership Privacy [SH21]	private (relative level unspecified)	yes (single privacy level)
Prediction Purification [Yan+20b]	private (relative level unspecified)	yes (single privacy level)
Self-Distillation [Tan+21]	—	—
PATE [Pap+16]	public	—
DP-SGD [Aba+16]	—	—

empirical privacy defenses that has guarantees on the resulting classifier’s generalization bound and, when coupled with DP-SGD, on the privacy leakage of training and reference data. Our method introduces the privacy requirement as a constraint on the generalization capability of the learned model. By evaluating the generalization error [SB14] as we will describe, the formulation leads to a convenient weighted empirical risk minimization (WERM) over the training and reference data. We evaluate our WERM baseline using three standard datasets in the field of privacy-preserving machine learning (Purchase100, Texas100, CIFAR100).

Our results show that, surprisingly, compared to state-of-the-art empirical privacy defenses using reference data, WERM is the best-performing method in nearly all privacy regimes. Additionally, we demonstrate that existing methods are only capable of extracting limited information from reference data during training and fail to effectively trade off reference data privacy for model utility and/or training data privacy. Our analysis reveals that the mechanisms provided by these defenses to control the utility-privacy tradeoff with respect to the three aforementioned factors do not function as expected, since they are only able to operate in the high reference data privacy case. By contrast, WERM is interpretable, straightforward to train, and highly effective. These traits enable it to serve as a baseline for evaluating future empirical privacy defenses using reference data. Importantly, comparing against our method requires selecting relative weights for the loss on the training data explicit any underlying assumption about their relative privacy.

In summary:

- We highlight the importance of clearly specifying the privacy requirements for reference data when training empirical privacy defenses.
- We propose a baseline that yields a better and more comprehensive utility-privacy landscape than state-of-the-art defenses using reference data.
- We provide an extensive theoretical analysis of our method, demonstrating how the weight term that governs the balance between training and reference data has a direct impact on the generalization bound and privacy leakage.



- We reveal that existing empirical privacy defenses do not function as expected, as they are unable to operate outside of the high reference data privacy regime.

In this work, we have analyzed the role of reference data in empirical privacy defenses and identified the issue that reference data privacy leakage must be explicitly considered to conduct a meaningful evaluation. We advanced the current state-of-the-art by proposing a generalization error constrained ERM, which can in practice be evaluated as a weighted ERM over the training and reference datasets. As WERM is intended to function as a baseline, we derive theoretical guarantees about its utility and privacy to ensure that its results will be well-understood in all utility-privacy settings. We present experimental results showing that our principled baseline outperforms the most well-studied and current state-of-the-art empirical privacy defenses in nearly all privacy regimes (i.e., independent of the nature of reference data and its level of privacy). Our experiments also reveal that existing methods are unable to trade off reference data privacy for model utility and/or training data privacy, and thus cannot operate outside of the highly private reference data case.

Regarding ethical concerns, our proposed baseline operates on the defense side of machine learning privacy; no novel attack has been proposed. Nevertheless, our experiments have analyzed the average privacy leakage over the whole dataset, but privacy protection is not always fair across groups in a dataset [Kul+19; Oli+23]. Future work can evaluate then the fairness of various defense mechanisms using reference data or propose the creation of privacy defenses intended to operate in use-case dependent settings. We hope that our work will continue to motivate the development of a robust evaluation framework for privacy defenses.

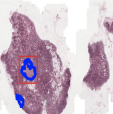

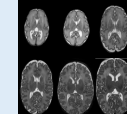
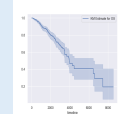
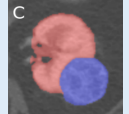
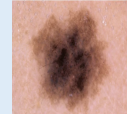
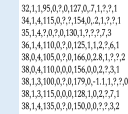
## 1.6.2 FLamby: Datasets and Benchmarks for Cross-Silo FL

*This section is based on our work [Ogi+22b], published at the 36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks.*

As previously highlighted in this chapter, statistical heterogeneity is a distinctive characteristic and a fundamental challenge in FL, and it is necessary to take it into consideration when evaluating FL algorithms. Most FL papers simulate statistical heterogeneity by artificially partitioning classic datasets, e.g., CIFAR-10/100 [Kri09], MNIST [LC10] or ImageNet [Den+09], on a given number of clients. Common approaches to produce synthetic partitions of classification datasets include associating samples from a limited number of classes to each client [McM+17], Dirichlet sampling on the class labels [HQB19; Yur+19], and using Pachinko Allocation Method (PAM) [LM06; Red+21] (which is only possible when the labels have a hierarchical structure). In the case of regression tasks, [PD22] partitions the *superconduct* dataset [CJB04] across 20 clients using Gaussian Mixture clustering based on t-SNE representations [VH08] of the features. Such synthetic partition approaches may fall short of modeling the complex statistical heterogeneity of real federated datasets. Evaluating FL strategies on datasets with natural client splits is a safer approach to ensuring that new strategies address real-world issues.

For cross-device FL, the LEAF dataset suite [Cal+19] includes five datasets with natural partition, spanning a wide range of machine learning tasks: natural language modeling (Reddit [Vol+17]), next character prediction (Shakespeare [McM+17]), sentiment analysis (Sent140 [GBH09]), image classification (CelebA [Liu+15]) and handwritten-character recognition (FEMNIST [Coh+17]). TensorFlow Federated [Bon+19] complements LEAF and provides three additional naturally split federated benchmarks, i.e., StackOverflow [Ten19], Google Landmark v2 [HQB20] and iNaturalist [Van+18]. Further, FLSim [Res12] provides cross-device examples based on LEAF and CIFAR10 [Kri09] with a synthetic split, and FedScale [Lai+22] introduces a large FL benchmark

Table 1.2: Overview of the datasets, tasks, metrics and baseline models in FLamby. For Fed-Camelyon16 the two different sizes refer to the size of the dataset before and after tiling.

Dataset	Fed-Camelyon16	Fed-LIDC-IDRI	Fed-IXI	Fed-TCGA-BRC	Fed-KITS2019	Fed-ISIC2019	Fed-Heart-Disease
<b>Input (x)</b>	Slides	CT-scans	T1WI	Patient info.	CT-scans	Dermoscopy	Patient info.
<b>Preprocessing</b>	Matter extraction + tiling	Patch Sampling	Registration	None	Patch Sampling	Various image transforms	Removing missing data
<b>Task type</b>	binary classificatio	3D segmentation	3D segmentation	survival	3D segmentation	multi-class classification	binary classification
<b>Prediction (y)</b>	Tumor on slide	Lung Nodule Mas	Brain mask	Risk of death	Kidney and tumor masks	Melanoma class	Heart disease
<b>Center extractor</b>	Hospital	Scanner Manufacturer	Hospital	Group of Hospital	Group of Hospital	Hospital	Hospital
<b>Thumbnails</b>							
<b>Original paper</b>	Litjens <i>et al.</i> 2018	Armato <i>et al.</i> 2011	Perez <i>et al.</i> 2021	Liu <i>et al.</i> 2018	Heller <i>et al.</i> 2019	Tschandl <i>et al.</i> 2018 Codella <i>et al.</i> 2017 / Combalia <i>et al.</i> 2019	Janosi <i>et al.</i> 1988
<b># clients</b>	2	5	3	5	6	5	4
<b># examples</b>	399	1,018	566	1,088	96	23,247	740
<b># examples per center</b>	239, 150	670, 205, 69, 74	311, 181, 74	311, 196, 206, 16, 51	12, 14, 12, 12, 16, 30	12413, 3954, 3363, 2259, 819, 439	303, 261, 46, 130
<b>Model</b>	DeepMIL [ITW]	Vnet [MNA16; Nik19]	3D U-net [Çiç+16]	Cox Model [Cox72]	nnU-Net [Ise+21]	efficientnet [TL19] + linear layer	Logistic Regression
<b>Metric</b>	AUC	DICE	DICE	C-index	DICE	Balanced Accuracy	Accuracy
<b>Size</b>	50G (850G total)	115G	444M	115K	54G	9G	40K
<b>Image resolution</b>	0.5 µm / pixel	~1.0 × 1.0 × 1.0 mm / voxel	~1.0 × 1.0 × 1.0 mm / voxel	NA	~1.0 × 1.0 × 1.0 mm / voxel	~0.02 mm / pixel	NA
<b>Input dimensior</b>	10,000 × 2048	128 × 128 × 128	48 × 60 × 48	39	64 × 192 × 192	200 × 200 × 3	13

focused on mobile applications. Apart from iNaturalist, the aforementioned datasets target the cross-device setting.

In contrast, publicly available datasets for the cross-silo FL setting are scarce. As a consequence, researchers usually rely on heuristics to artificially generate heterogeneous data partitions from a single dataset and assign them to hypothetical clients. Such heuristics might fall short of replicating the complexity of natural heterogeneity found in real-world datasets. The example of digital histopathology [Vet+14], a crucial data type in cancer research, illustrates the potential limitations of such synthetic partition methods. In digital histopathology, tissue samples are extracted from patients, stained, and finally digitized. In this process, known factors of data heterogeneity across hospitals include patient demographics, staining techniques, storage methodologies of the physical slides, and digitization processes [Jan+19; Fu+20; How+21]. Although staining normalization [Lah+20; Haa+21] has seen recent progress, mitigating this source of heterogeneity, the other highlighted sources of heterogeneity are difficult to replicate with synthetic partitioning [How+21] and some may be unknown, which calls for actual cross-silo cohort experiments. This observation is also valid for many other application domains, e.g. radiology [HBB06], dermatology [Bad+15], retinal images [Bad+15] and more generally computer vision [TE11].

In order to address the lack of realistic cross-silo datasets, we propose FLamby, an open source cross-silo federated dataset suite with natural partitions focused on healthcare, accompanied by code examples, and benchmarking guidelines. Table 1.2 gives an overview of FLamby. To the best of our knowledge, apart from some promising isolated works to build realistic cross-silo FL datasets, our work is the first standard benchmark allowing to systematically study healthcare cross-silo FL on different data modalities and tasks. Our contributions are threefold:

1. We build an open-source federated cross-silo healthcare dataset suite including 7 datasets. These datasets cover different tasks (classification/segmentation/survival) in multiple application domains and with different data modalities and scale. Crucially, all datasets are partitioned using natural splits.
2. We provide guidelines to help compare FL strategies in a fair and reproducible manner, and provide illustrative results for this benchmark.
3. We make open-source code accessible at <https://github.com/owkin/FLamby> for benchmark reproducibility and easy integration in different FL frameworks, but also to allow the research community to contribute to FLamby development, by adding more datasets, benchmarking types and FL strategies.

Currently, FLamby is limited to healthcare datasets. In the longer run and with the help of the FL community, it could be enriched with datasets from other application domains to better reflect the diversity of cross-silo FL applications, which is possible thanks to its modular design. Regarding machine learning back-ends, FLamby only provides PyTorch [Pas+19] code: supporting other back-ends, such as TensorFlow [Mar+15] or JAX [Bra+18], is a relevant future direction if there is such demand from the community. Further, our benchmark currently does not integrate all constraints of cross-silo FL, especially privacy aspects, which are important in this setting.

In terms of FL setting, the benchmark mainly focuses on the heterogeneity induced by natural splits. In order to make it more realistic, future developments might include in depth study of Differential Privacy (DP) training [DR+14], cryptographic protocols such as Secure Aggregation [Bon+17], Personalized FL [FMO20], or communication constraints [Sat+19] when applicable.

## 1.7 Publications

The contributions of this manuscript led to the following publications and submissions in conferences and peer-reviewed journals

### 1.7.1 Published

- [Kap+24a] Caelin Kaplan, Chuan Xu, **Othmane Marfoq**, Giovanni Neglia, and Anderson Santana de Oliveira. “A Cautionary Tale: On the Role of Reference Data in Empirical Privacy Defenses”. In: *Proceedings on Privacy Enhancing Technologies* (2024).
- [Mar+21a] **Othmane Marfoq**, Giovanni Neglia, Aurélien Bellet, Laetitia Kamani, and Richard Vidal. “Federated Multi-Task Learning under a Mixture of Distributions”. In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021.
- [Mar+23a] **Othmane Marfoq**, Giovanni Neglia, Laetitia Kamani, and Richard Vidal. “Federated Learning for Data Streams”. In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. Ed. by Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent. Vol. 206. Proceedings of Machine Learning Research. PMLR, Apr. 2023, pp. 8889–8924. URL: <https://proceedings.mlr.press/v206/marfoq23a.html>.

- [Mar+22a] **Othmane Marfoq**, Giovanni Neglia, Laetitia Kameni, and Richard Vidal. “Personalized Federated Learning through Local Memorization”. In: *Proceedings of the 39th International Conference on Machine Learning*. Proceedings of Machine Learning Research. PMLR, 2022.
- [Mar+20a] **Othmane Marfoq**, Chuan Xu, Giovanni Neglia, and Richard Vidal. “Throughput-Optimal Topology Design for Cross-Silo Federated Learning”. In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020.
- [Ogi+22a] Jean Ogier du Terrail, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold, Tanguy Marchand, **Othmane Marfoq**, Erum Mushtaq, Boris Muzellec, Constantin Philippenko, Santiago Silva, Maria Teleńczuk, Shadi Albarqouni, Salman Avestimehr, Aurélien Bellet, Aymeric Dieuleveut, Martin Jaggi, Sai Praneeth Karimireddy, Marco Lorenzi, Giovanni Neglia, Marc Tommasi, and Mathieu Andreux. “FLamby: Datasets and Benchmarks for Cross-Silo Federated Learning in Realistic Settings”. Proceedings of The 36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks. <https://openreview.net/forum?id=GgM5DiAb6A2>. 2022.
- [Rod+23a] Angelo Rodio, Francescomaria Faticanti, **Othmane Marfoq**, Giovanni Neglia, and Emilio Leonardi. “Federated Learning under Heterogeneous and Correlated Client Availability”. In: *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*. 2023, pp. 1–10. DOI: 10.1109/INFOCOM53939.2023.10228876.

### 1.7.2 Submitted

- [MMa] **Othmane Marfoq** and Aryan Mokhtari. *Online Federated Learning with Mixture Models*.



---

# System Considerations in Heterogeneous Federated Learning

As discussed in Chapter 1, clients in federated learning exhibit a diverse range of characteristics, encompassing disparities in storage capacity, computational resources, and communication capabilities. These disparities arise from variations in hardware specifications (CPU power, memory capacity), network connectivity types (3G, 4G, 5G, WiFi), and power availability (battery levels). The system heterogeneity results in different challenges depending on the learning scenario. In Section 1.4, we delineated the challenges associated with system heterogeneity across three distinct scenarios: cross-silo, cross-device, and heterogeneous hardware environments. In this chapter, we focus on the cross-silo and cross-device scenarios, and we leave the heterogeneous hardware environments to Chapter 3 (Section 3.6).

In Section 2.1, we define the problem of topology design for cross-silo federated learning using the theory of max-plus linear systems to compute the system throughput—number of communication rounds per time unit. We also propose practical algorithms that, under the knowledge of measurable network characteristics, find a topology with the largest throughput or with provable throughput guarantees.

In Section 2.2, we provide a novel analysis for a FedAvg-like algorithm under heterogeneous and correlated client availability. Our analysis highlights how correlation adversely affects the algorithm’s convergence rate and how the aggregation strategy can alleviate this effect at the cost of steering training toward a biased model. Guided by the theoretical analysis, we propose CA-Fed, a new FL algorithm that tries to balance the conflicting goals of maximizing convergence speed and minimizing model bias. To this purpose, CA-Fed dynamically adapts the weight given to each client and may ignore clients with low availability and large correlation.

This chapter is based on our works [Mar+20b], published in *Advances in Neural Information Processing Systems 2020 (NeurIPS’20)*, and [Rod+23b], published in the *IEEE/ACM Transactions on Networking*.

## 2.1 Throughput-Optimal Topology Design for Cross-Silo Federated Learning

As discussed in Chapter 1, specifically in Section 1.3 and Section 1.4.1.2, the standard federated learning approach, employing a server-client architecture where an orchestrator iteratively aggre-

gates model updates from remote clients and pushes them back a refined model, may be inefficient in cross-silo settings, as close-by data silos with high-speed access links may exchange information faster than with the orchestrator, and the orchestrator may become a communication bottleneck. In this context, an important question arises: *How can we design a communication topology that allows for the fastest convergence, considering that different silos have different communication capabilities?*

In this chapter, we define the problem of topology design for cross-silo federated learning using the theory of max-plus linear systems to compute the system throughput—number of communication rounds per time unit. We also propose practical algorithms that, under the knowledge of measurable network characteristics, find a topology with the largest throughput or with provable throughput guarantees. In realistic Internet networks with 10 Gbps access links at silos, our algorithms speed up training by a factor 9 and 1.5 in comparison to the server-client architecture and to state-of-the-art MATCHA, respectively. Speedups are even larger with slower access links.

### 2.1.1 Introduction

In federated learning, clients (e.g., mobile devices or whole organizations) usually train the model through an iterative procedure under the supervision of a central orchestrator, which, for example, decides to launch the training process and coordinates training advances. Often—e.g., in FedAvg [McM+17], SCAFFOLD [Kar+20b], and FedProx [Li+20b]—the orchestrator directly participates to the training, by aggregating clients’ updates, generating a new model, and pushing it back to the clients. Hence, clients only communicate with a potentially far-away (e.g., in another continent) orchestrator and do not exploit communication opportunities with close-by clients. This choice is justified in the cross-device setting, where inter-device communication is unreliable (devices may drop-out from training at any time) and slow (a message needs to traverse two slow access links). But in the cross-silo setting, data silos (e.g., data centers) are almost always available, enjoy high-speed connectivity comparable to the orchestrator’s one, and may exchange information faster with some other silos than with the orchestrator. An orchestrator-centered communication topology is then potentially inefficient, because it ignores fast inter-silo communication opportunities and makes the orchestrator a candidate for congestion. A current trend [Tan+18; Bel+18; Wan+19a; Yua+21; Yua+23] is then to replace communication with the orchestrator by peer-to-peer communications between individual silos, which perform local partial aggregations of model updates. We also consider this scenario and study how to design the communication topology.

The communication topology has two contrasting effects on training duration. First, a more connected topology leads to faster convergence in terms of iterations or communication rounds, as quantified by convergence bounds in terms of the spectral properties of the topology [NOR18; DAW12; Sca+17; Sca+18; WJ21; Jia+17]. Second, a more connected topology increases the duration of a communication round (e.g., it may cause network congestion), motivating the use of degree-bounded topologies where every client sends and receives a small number of messages at each round [Ass+19; Lia+17]. Recent experimental and theoretical work suggests the second effect may dominate the first one (see [Lia+17; Lia+18; Luo+19; POP20b; Ass+19] and the discussion in [Neg+20]).

Only a few studies have designed topologies taking into account the duration of a communication round. Under the simplistic assumption that the communication time is proportional to node degree, MATCHA [Wan+19a] decomposes the set of possible communications into matchings (disjoint pairs of clients) and, at each communication round, randomly selects some matchings and

allows their pairs to transmit. MATCHA chooses the matchings’ selection probabilities in order to optimize the algebraic connectivity of the expected topology. Reference [Neg+19] studies how to select the degree of a regular topology when the duration of a communication round is determined by stragglers [Kar+17; Li+18]. Apart from these corner cases, “*how to design a [decentralized] model averaging policy that achieves the fastest convergence remains an open problem*” [Kai+21].

In this chapter, we address this open problem, by using the theory of linear systems in the max-plus algebra [Bac92] to design topologies for cross-silo distributed learning. The theory holds for synchronous systems and has been successfully applied in other fields (e.g., manufacturing [CMK01], communication networks [LT01], biology [BRH12], railway systems [Gov98], and road networks [FGQ11]). Synchronous optimization algorithms are often preferred for federated learning [Bon+19], because they enjoy stronger convergence guarantees than their asynchronous counterparts and can be easily combined with cryptographic secure aggregation protocols [Bon+17], differential privacy techniques [Aba+16], and model and update compression [SL21].

This work is the first work to take explicitly in consideration all delay components contributing to the total training time including computation times, link latencies, transmission times, and queueing delays. It complements the topology design approaches listed above that only account for congestion at access links [Wan+19a] and straggler effect [Neg+19].

## 2.1.2 Problem Formulation

### 2.1.2.1 Machine Learning Training

We consider a network of  $N$  siloed data centers who collaboratively train a global machine learning model, solving the following optimization problem:

$$\underset{\mathbf{w} \in \mathbb{R}^d}{\text{minimize}} \sum_{i=1}^N \mathbb{E}_{\xi_i} [f_i(\mathbf{w}, \xi_i)], \quad (2.1)$$

where  $f_i(\mathbf{w}, \xi_i)$  is the loss of model  $\mathbf{w}$  at a sample  $\xi_i$  drawn from data distribution at silo  $i$ . (It is also possible to weight each loss with the size of the local dataset).

In order to solve Problem (2.1) in an FL scenario, silos do not share the local datasets, but periodically transmit model updates, and different distributed algorithms have been proposed [Li+20b; McM+17; Kar+20b; Wan+19a; Kon+17a; WJ21].

In this section we consider as archetype the decentralized periodic averaging stochastic gradient descent (DPASGD) [WJ21], where silos are represented as vertices of a communication graph that we call *overlay*. Each silo  $i$  maintains a local model  $\mathbf{w}_i$  and performs  $s$  mini-batch gradient updates before sending its model to a subset of silos  $\mathcal{N}_i^-$  (its out-neighbors in the overlay). It then aggregates its model with those received by a (potentially different) set of silos  $\mathcal{N}_i^+$  (its in-neighbors). Formally, the algorithm is described by the following equations:

$$\mathbf{w}_i(k+1) = \begin{cases} \sum_{j \in \mathcal{N}_i^+ \cup \{i\}} \mathbf{A}_{i,j} \mathbf{w}_j(k), & \text{if } k \equiv 0 \pmod{s+1}, \\ \mathbf{w}_i(k) - \alpha_k \frac{1}{m} \sum_{h=1}^m \nabla f_i(\mathbf{w}_i(k), \xi_i^{(h)}(k)), & \text{otherwise.} \end{cases} \quad (2.2)$$

where  $m$  is the batch size,  $\alpha_k > 0$  is a potentially varying learning rate, and  $\mathbf{A} \in \mathbb{R}^{N \times N}$  is a matrix of non-negative weights, referred to as the *consensus matrix*. For particular choices of the matrix  $\mathbf{A}$  and the number of local updates  $s$ , DPASGD reduces to other schemes previously



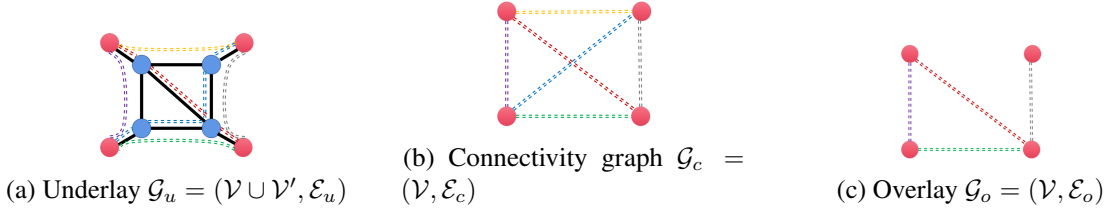


Figure 2.1: Examples for underlay, connectivity graph, and overlay, with routers (blue nodes), silos (red nodes), underlay links (solid black lines), and information exchanges (dashed lines).

proposed [Lia+17; Yua+19], including FedAvg [McM+17], where the orchestrator just performs the averaging step (this corresponds to its local loss function  $f_i(\cdot)$  being a constant). Convergence of (2.2) was proved in [WJ21].

In this chapter, we study how to design the overlay in order to minimize the training time. While we consider DPASGD, our results are applicable to any synchronous iterative algorithm where each silo alternates a local computation phase and a communication phase during which it needs to receive inputs from a given subset of silos before moving to the next computation phase. This includes the distributed algorithms already cited, as well as push-sum training schemes [Ass+19; Shi+15; RNV12; NOS17; DS16; TLR12; ZY17] and in general the *black-box optimization procedures* as defined in [Sca+17].

### 2.1.2.2 Underlay, Connectivity graph, and Overlay

FL silos are connected by a communication infrastructure (e.g., the Internet or some private network), which we call *underlay*. The underlay can be represented as a directed graph (digraph)  $\mathcal{G}_u = (\mathcal{V} \cup \mathcal{V}', \mathcal{E}_u)$ , where  $\mathcal{V}$  denotes the set of silos,  $\mathcal{V}'$  the set of other nodes (e.g., routers) in the network, and  $\mathcal{E}_u$  the set of communication links. For simplicity, we consider that each silo  $i \in \mathcal{V}$  is connected to the rest of the network through a single link  $(i, i')$ , where  $i' \in \mathcal{V}'$ , with uplink capacity  $C_{UP}(i)$  and downlink capacity  $C_{DN}(i)$ . The example in Fig. 2.1 illustrates the underlay and the other concepts we are going to define.

The *connectivity graph*  $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$  captures the possible direct communications among silos. Often the connectivity graph is fully connected, but specific NAT or firewall configurations may prevent some pairs of silos to communicate. If  $(i, j) \in \mathcal{E}_c$ ,  $i$  can transmit its updated model to  $j$ . The message experiences a delay that is the sum of two contributions: 1) an end-to-end delay  $l(i, j)$  accounting for link latencies, and queuing delays along the path, and 2) a term depending on the model size  $M$  and the *available bandwidth*\*  $A(i, j)$  of the path. Each pair of silos  $(i, j)$  can use probing packets [JD02; Pra+03; Hsi+17] to measure end-to-end delays and available bandwidths and communicate them to the orchestrator, which then designs the topology. We assume that in the stable cross-silo setting these quantities do not vary or vary slowly, so that the topology is recomputed only occasionally, if at all.

The training algorithm in (2.2) does not need to use all potential connections. The orchestrator can select a connected subgraph of  $\mathcal{G}_c$ . We call such subgraph *overlay* and denote it by  $\mathcal{G}_o = (\mathcal{V}, \mathcal{E}_o)$ , where  $\mathcal{E}_o \subset \mathcal{E}_c$ . Only nodes directly connected in  $\mathcal{G}_o$  will exchange messages. We can associate a

\*The available bandwidth of a path is the maximum rate that the path can provide to a flow, taking into account the rest of the traffic [CC96; JD02]; it is then smaller than the minimum link capacity of the path.

delay to each link  $(i, j) \in \mathcal{E}_o$ , corresponding to the time interval between the beginning of a local computation at node  $i$ , and the receiving of  $i$ 's updated model by  $j$ :

$$d_o(i, j) = s \times T_c(i) + l(i, j) + \frac{M}{A(i, j)} = s \times T_c(i) + l(i, j) + \frac{M}{\min \left( \frac{C_{\text{UP}}(i)}{|\mathcal{N}_i^-|}, \frac{C_{\text{DN}}(j)}{|\mathcal{N}_j^+|}, A(i', j') \right)}, \quad (2.3)$$

where  $T_c(i)$  denotes the time to compute one local update of the model. We also define  $d_o(i, i) = s \times T_c(i)$ . Equation (2.3) holds under the following assumptions. First, each silo  $i$  uploads its model in parallel to its out-neighbors in  $\mathcal{N}_i^-$  (with a rate at most  $C_{\text{UP}}(i)/|\mathcal{N}_i^-|$ ). Second, downloads at  $j$  happen in parallel too. While messages from different in-neighbors may not arrive at the same time at  $j$ 's downlink, their transmissions are likely to partially overlap. Finally, different messages do not interfere significantly in the core network, where they are only a minor component of the total network traffic ( $A(i', j')$  does not depend on  $\mathcal{G}_o$ ).

Our model is more general than those considered in related work: [Wan+19a] considers  $d_o(i, j) = M \times |\mathcal{N}_i^-|/C_{\text{UP}}(i)$  and [Neg+19] considers  $d_o(i, j) = T_c(i)$  (but it accounts for random computation times).

### 2.1.2.3 Time per Communication Round (Cycle Time)

Let  $t_i(k)$  denote the time at which worker  $i$  starts computing  $w_i((s+1)k+1)$  according to (2.2) with  $t_i(0) = 0$ . As  $i$  needs to wait for the inputs  $w_j((s+1)k)$  from its in-neighbors, the following recurrence relation holds

$$t_i(k+1) = \max_{j \in \mathcal{N}_i^+ \cup \{i\}} (t_j(k) + d_o(j, i)). \quad (2.4)$$

This set of relations generalizes the concept of a linear system in the max-plus algebra, where the max operator replaces the usual sum and the + operator replaces the usual product. We refer the reader to [Bac92] for the general theory of such systems and we present here only the key results for our analysis.

We call the time interval between  $t_i(k)$  and  $t_i(k+1)$  a *cycle*. The average cycle time for silo  $i$  is defined as  $\tau_i = \lim_{k \rightarrow \infty} t_i(k)/k$ . The cycle time 1) does not depend on the specific silo (i.e.,  $\tau_i = \tau_j$ ) [Bac92, Sect. 7.3.4], and 2) can be computed directly from the graph  $\mathcal{G}_o$  [Bac92, Thm. 3.23]. In fact:

$$\tau(\mathcal{G}_o) = \max_{\gamma} \frac{d_o(\gamma)}{|\gamma|}, \quad (2.5)$$

where  $\gamma$  is a generic circuit, i.e., a path  $(i_1, \dots, i_p = i_1)$  where the initial node and the final node coincide,  $|\gamma| = p$  is the length of the circuit, and  $d_o(\gamma) = \sum_{k=1}^{p-1} d_o(i_k, i_{k+1})$  is the sum of delays on  $\gamma$ . A circuit  $\gamma$  of  $\mathcal{G}_o$  is called *critical* if  $\tau(\mathcal{G}_o) = d_o(\gamma)/|\gamma|$ . There exist algorithms with different complexity to compute the cycle time [Kar78; DG98].

The cycle time is a key performance metric for the system because the difference  $|t_i(k) - \tau(\mathcal{G}_o) \times k|$  is bounded for all  $k \geq 0$  so that, for large enough  $k$ ,  $t_i(k) \approx \tau(\mathcal{G}_o) \times k$ . In particular, the inverse of the cycle time is the *throughput* of the system, i.e., the number of communication rounds per time unit. An overlay with minimal cycle time minimizes the time required for a given number of communication rounds. This observation leads to our optimization problem.

Table 2.1: Algorithms to design the overlay  $\mathcal{G}_o$  from the connectivity graph  $\mathcal{G}_c$ .

Network	Conditions	Algorithm	Complexity	Guarantees
Edge-capacitated	Undirected $\mathcal{G}_o$	Prim's Algorithm [Pri57]	$\mathcal{O}( \mathcal{E}_c  +  \mathcal{V}  \log  \mathcal{V} )$	Optimal solution (Prop. 2.1.1)
Edge/Node-capacitated	Euclidean $\mathcal{G}_c$	Christofides' Algorithm [MPT02]	$\mathcal{O}( \mathcal{V} ^2 \log  \mathcal{V} )$	3N-approximation (Proposition 2.1.3, 2.1.6)
Node-capacitated	Euclidean $\mathcal{G}_c$ and undirected $\mathcal{G}_o$	Algorithm 3	$\mathcal{O}( \mathcal{E}_c   \mathcal{V}  \log  \mathcal{V} )$	6-approximation (Prop. 2.1.5)

### 2.1.2.4 Optimization Problem

Given a connectivity graph  $\mathcal{G}_c$ , we want the overlay  $\mathcal{G}_o$  to be a strong digraph (i.e., a strongly connected directed graph) with minimal cycle time. Formally, we define the following *Minimal Cycle Time* problem:

Minimal Cycle Time (MCT)

**Input:** A strong digraph  $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$ ,  $\{C_{\text{UP}}(i), C_{\text{DN}}(j), l(i, j), A(i', j'), T_c(i), \forall (i, j) \in \mathcal{E}_c\}$

**Question:** What is the strong spanning subdigraph of  $\mathcal{G}_c$  with minimal cycle time?

Note that the input does not include detailed information about the underlay  $\mathcal{G}_u$ , but only information available or measurable at the silos (see Sect. 2.1.2.2). To the best of our knowledge, our work is the first effort to study MCT. The closest problem considered in the literature is, for a given overlay, to select the largest delays that guarantee a minimum throughput [Gau95; Dav+14].

## 2.1.3 Theoretical Results and Algorithms

In this section we present complexity results for MCT and algorithms to design the optimal topology in different settings. Table 2.1 lists these algorithms, their time-complexity, and their guarantees. We note that in some cases we adapt known algorithms to solve MCT. All proofs are in App. C.1.

Borrowing the terminology from P2P networks [Mas+07] we call a network *edge-capacitated* or *node-capacitated*, respectively, if access links delays can be neglected or not. While in cross-device FL the network is definitely node-capacitated, in cross-silo FL—the focus of our work—silos may be geo-distributed data centers or branches of a company and then have high-speed connections, so that neglecting access link delays may be an acceptable approximation.

### 2.1.3.1 Edge-capacitated networks

FL algorithms often use an *undirected* overlay with symmetric communications, i.e.,  $(i, j) \in \mathcal{E}_o \Rightarrow (j, i) \in \mathcal{E}_o$ . This is the case of centralized schemes, like FedAvg, but is also common for other consensus-based optimization schemes where the consensus matrix  $\mathbf{A}$  is required to be doubly-stochastic [NO09; RNV12; WJ21]—a condition simpler to achieve when  $\mathcal{G}_o$  is undirected.

When building an undirected overlay, we can restrict ourselves to consider trees as solutions of MCT. In fact, additional links can only increase the number of circuits and then increase the cycle time (see (2.5)). Moreover, we can prove that the overlay has simple critical circuits of the form  $\gamma = (i, j, i)$ , for which  $d_o(\gamma)/|\gamma| = (d_o(i, j) + d_o(j, i))/2$ . Intuitively, if we progressively build a tree using the links in  $\mathcal{G}_c$  with the smallest average of delays in the two directions, we obtain the overlay with minimal cycle time. This construction corresponds to finding a minimum weight spanning tree (MST) in an opportune undirected version of  $\mathcal{G}_c$ :

**Proposition 2.1.1.** Consider an undirected weighted graph  $\mathcal{G}_c^{(u)} = (\mathcal{V}, \mathcal{E}_c^{(u)})$ , where  $(i, j) \in \mathcal{E}_c^{(u)}$  iff  $(i, j) \in \mathcal{E}_c$  and  $(j, i) \in \mathcal{E}_c$  and where  $(i, j) \in \mathcal{E}_c^{(u)}$  has weight  $d^{(u)}(i, j) = (d_o(i, j) + d_o(j, i))/2$ . A minimum weight spanning tree of  $\mathcal{G}_c^{(u)}$  is a solution of MCT when  $\mathcal{G}_c$  is edge-capacitated and  $\mathcal{G}_o$  is required to be undirected.

Prim’s algorithm [Pri57] is an efficient algorithm to find an MST with complexity  $\mathcal{O}(|\mathcal{E}_c| + |\mathcal{V}| \log |\mathcal{V}|)$  and then suited for the usual cross-silo scenarios with at most a few hundred nodes [Kai+21].

We have pointed out a simple algorithm when the overlay is undirected, but directed overlays can have arbitrarily shorter cycle times than undirected ones even in simple settings where all links in the underlay are bidirectional with identical delays in the two directions (see Example 1).

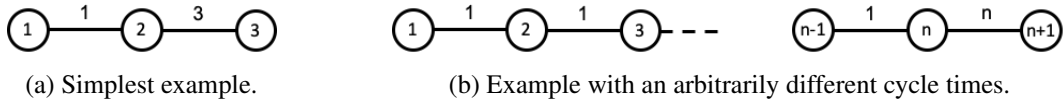


Figure 2.2: Networks where a directed topology outperforms an undirected one.

**Example 1.** We provide two examples where the underlay network is undirected and still a directed overlay can have shorter cycle time than directed overlays. Examples are in Fig. 2.2, where numbers associated to links are the corresponding delays (in the two directions).

The network in Fig. 2.2a has only three nodes,  $\mathcal{V} = \{1, 2, 3\}$ . We have  $d_c(1, 2) = d_c(2, 1) = 1$ ,  $d_c(2, 3) = d_c(3, 2) = 3$ , and  $d_c(1, 3) = d_c(3, 1) = 4$ . The fastest undirected overlay is  $\mathcal{G}_o^{(u)} = (\mathcal{V}, \{(1, 2), (2, 3)\})$ . Consider the directed ring  $\mathcal{G}_o = (\mathcal{V}, \{(1, 2), (2, 3), (3, 1)\})$ . We have:

$$\tau(\mathcal{G}_o^{(u)}) = \max \left\{ \frac{1+1}{2}, \frac{3+3}{2}, \frac{1+3+1+3}{4} \right\} = 3, \quad (2.6)$$

$$\tau(\mathcal{G}_o) = \frac{1+3+(3+1)}{3} = \frac{8}{3} < 3. \quad (2.7)$$

The network in Fig. 2.2b shows that a directed ring can be arbitrarily faster than an undirected one. Similarly to above, the fastest undirected overlay is  $\mathcal{G}_o^{(u)}$  coincides with the underlay. The directed overlay is the ring  $(1 \rightarrow 2 \rightarrow 3 \rightarrow \dots \rightarrow n \rightarrow n+1 \rightarrow 1)$ . We have

$$\tau(\mathcal{G}_o^{(u)}) = n, \quad (2.8)$$

$$\tau(\mathcal{G}_o) = \frac{(n-1) \times 1 + n + (n + (n-1) \times 1)}{n+1} = \frac{4n-2}{n+1} < 4. \quad (2.9)$$

The ratio of the two cycle times can be made arbitrarily large.

Despite these advantageous features of directed topologies, it’s worth noting that the computation of optimal directed overlays poses a challenge due to its NP-hard nature (Proposition 2.1.2). Consequently, the pursuit of finding optimal solutions for such overlays remains a complex task that demands innovative approaches.

**Proposition 2.1.2.** MCT is NP-hard even when  $\mathcal{G}_c$  is a complete Euclidean edge-capacitated graph.

We call a connectivity graph  $\mathcal{G}_c$  *Euclidean* if its delays  $d_c(i, j) \triangleq s \times T_c(i) + l(i, j) + M/A(i', j')$  are symmetric ( $d_c(i, j) = d_c(j, i), \forall i, j \in \mathcal{V}$ ) and satisfy the triangle inequality ( $d_c(i, j) \leq d_c(i, k) + d_c(k, j), \forall i, j, k \in \mathcal{V}$ ). These assumptions are roughly satisfied for geographically distant computing clusters with similar computation times, as the delay to transmit a message between two silos is roughly an affine function of the geodesic distance between them [Gue+04]. Under this condition MCT can be approximated:

**Proposition 2.1.3.** *Christofides' algorithm [MPT02] is a  $3N$ -approximation algorithm for MCT when  $\mathcal{G}_c$  is edge-capacitated and Euclidean.*

The result follows from Christofides' algorithm being a 1.5-approximation algorithm for the Traveling Salesman Problem [MPT02], and our proof shows that a solution of the Traveling Salesman Problem provides a  $2N$ -approximation of MCT. Note that Christofides' algorithm finds *ring* topologies. Note that the obtained approximation factor is exact (up to a multiplicative constant), we provide Example 2 where the TSP solution is an  $\Omega(N)$  of the optimal solution of MCT.

**Example 2.** *We provide an example of an euclidean underlay where using a ring as overlay is  $N$  times worse than the optimal overlay. We consider a complete connectivity graph  $\mathcal{G}_c = (\mathcal{V}, \mathcal{V} \times \mathcal{V})$  to which we associate a delay function  $d_u$  verifying*

$$\forall (i, j) \in \mathcal{V} \times \mathcal{V}; \quad d(i, j) = \begin{cases} 0 & i, j \in \{1, \dots, N\} \\ 1 & i \in \{N+1, \dots, 2N\} \text{ or } j \in \{N+1, \dots, 2N\} \end{cases} \quad (2.10)$$

$\mathcal{G}_c$  is clearly an Euclidean graph.

A Hamiltonian cycle  $\mathcal{H}$  of  $\mathcal{G}_c$  needs to use exactly  $2N$  different edges, thus it has a cost at least  $N \times 0 + N \times 1 = N$ , and a cycle time  $\tau(\mathcal{H}) \geq \frac{N}{2N} = \frac{1}{2}$ .

Consider a directed overlay  $\mathcal{G}_o = (\mathcal{V}, \mathcal{E}_o)$ , with

$$\mathcal{E}_o = \{(i, i+1); i \in \{1, \dots, N-1\}\} \cup \bigcup_{K \in \{N+1, \dots, 2N\}} \{(N, K), (K, 1)\} \quad (2.11)$$

The set of elementary circuits of  $\mathcal{E}_o$  is exactly the set  $\mathcal{C} = \{C_K = (1, \dots, N, K, 1); K \in \{N+1, 2N\}\}$ . For any circuit  $C_K \in \mathcal{C}$ ,  $\tau(C_K) = \frac{0 \times N + 2 \times 1}{N+2} = \frac{2}{N+2}$ .

It follows that the minimal cycle time  $\tau_{OPT}$  that a strong spanning subdigraph of  $\mathcal{G}_c$  can achieve is such that  $\tau_{OPT} \leq \frac{2}{N+2}$ . Thus  $\tau(\mathcal{H}) \geq \frac{N+2}{4} \tau_{OPT}$  for any Hamiltonian cycle  $\mathcal{H}$  of  $\mathcal{G}_c$ .

### 2.1.3.2 Node-capacitated networks

When silos do not enjoy high-speed connectivity, congestion at access links can become the dominant contribution to network delays, especially when one silo communicates with many others. Intuitively, in this setting, good overlays will exhibit small degrees.

If  $\mathcal{G}_o$  is required to be undirected, MCT can be reduced from the problem of finding the minimum bottleneck spanning tree with bounded degree  $\delta > 1$  ( $\delta$ -MBST for short),\* which is NP-hard.

**Proposition 2.1.4.** *In node-capacitated networks MCT is NP-hard even when the overlay is required to be undirected.*

\*A  $\delta$ -MBST is a spanning tree with degree at most  $\delta$  in which the largest edge delay is as small as possible.

We propose Algorithm 3, which combines existing approximation algorithms for  $\delta$ -MBST on a particular undirected graph built from  $\mathcal{G}_c$  and denoted by  $\mathcal{G}_c^{(u)}$  (lines 1-3). Lemma C.4 establishes a connection between the bottleneck of the MBST of  $\mathcal{G}_c^{(u)}$  and the cycle time of MCT on  $\mathcal{G}_c$  when the overlay is required to be undirected. To get an approximated 2-MBST on  $\mathcal{G}_c^{(u)}$ , we apply the best known 3-approximation algorithm from [AR16, Sect. 3.2.1] (lines 6-8) which requires  $\mathcal{G}_c^{(u)}$  to be Euclidean (Lemma C.5), and take its result as one candidate for our solution (line 9). The cube of a graph  $\mathcal{G}$ , denoted by  $\mathcal{G}^3$ , is the super-graph of  $\mathcal{G}$  such that the edge  $(u, v)$  is in  $\mathcal{G}^3$  if and only if there is a path between  $u$  and  $v$  in  $\mathcal{G}$  with three or fewer edges. It has been proved that the cube of a connected graph is Hamiltonian and to find a Hamiltonian path in such a cube can be done in polynomial time.\* Other  $\delta$ -BSTs built by Algorithm 4 for  $3 \leq \delta \leq N$  are considered as candidates (lines 10-11) and we finally provide as solution the overlay with the smallest cycle time (line 13).

---

**Algorithm 3:** Approximation algorithm for MCT on node-capacitated networks.

---

**Input:**  $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$ , uplink capacity  $C_{\text{UP}}(i)$ , end-to-end delay  $l(i, j)$ , computation time  $T_c(i)$  and model size  $M$ .

**Result:** Undirected overlay  $\mathcal{G}_o$ .

```

1 Create  $\mathcal{G}_c^{(u)} = (\mathcal{V}, \mathcal{E}_c^{(u)})$  where  $(i, j) \in \mathcal{E}_c^{(u)}$  iff  $(i, j) \in \mathcal{E}_c$  and  $(j, i) \in \mathcal{E}_c$ ;
2 for  $(i, j) \in \mathcal{E}_c^{(u)}$  do
3   |  $d^{(u)}(i, j) = [s \times (T_c(i) + T_c(j)) + l(i, j) + l(j, i) + \frac{M}{C_{\text{UP}}(i)} + \frac{M}{C_{\text{UP}}(j)}] / 2$ 
4 end
5  $\mathbb{S} \leftarrow \emptyset$ ; // the set of candidate solutions
   /* consider 2-MBST approximate solution on  $\mathcal{G}_c^{(u)}$  as one candidate */
6  $\mathcal{T} \leftarrow$  a minimum weight spanning tree of  $\mathcal{G}_c^{(u)}$ ;
7  $\mathcal{T}^3 \leftarrow$  the cube of  $\mathcal{T}$ ;
8  $\mathcal{H} \leftarrow$  a Hamiltonian path in  $\mathcal{T}^3$ ;
9  $\mathbb{S} \leftarrow \mathcal{H}$ ;
   /* consider other  $\delta$ -BST for  $3 \leq \delta \leq N$  as candidates */
10 for  $\delta \in \{3, 4, 5, \dots, N\}$  do
11   |  $\mathbb{S} \leftarrow \mathbb{S} \cup \delta\text{-PRIM}(\mathcal{G}_c^{(u)})$  //  $\delta\text{-PRIM}(\mathcal{G}_c^{(u)})$  gives a  $\delta$ -BST on  $\mathcal{G}_c^{(u)}$ 
12 end
   /* choose the one with the minimum cycle time as output overlay */
13  $\mathcal{G}_o \leftarrow \arg \min_{G \in \mathbb{S}} \tilde{\tau}(G)$ 

```

---

\*Jerome J.Karaganis. "On the cube of a graph," 1968.

**Algorithm 4:**  $\delta$ -PRIM[AR19]

---

```

1 Function  $\delta$ -PRIM( $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ):
2    $\mathcal{V}_T := \{v_0\}$  for some  $v_0 \in \mathcal{V}$ ;
3    $\mathcal{E}_T := \{\}$ ;
4    $T = (\mathcal{V}_T, \mathcal{E}_T)$ ;
5   while  $|\mathcal{E}_T| < |\mathcal{V}| - 1$  do
6     Find the smallest weight edge  $(u, v)$  such that  $u \in \mathcal{V}_T, v \notin \mathcal{V}_T$ , and
       DEGREE $_T(u) < \delta$ ;
7     Add  $v$  to  $\mathcal{V}_T$ ;
8     Add  $(u, v)$  to  $\mathcal{E}_T$ ;
9   end
10  return  $T$ ;

```

---

**Proposition 2.1.5.** *Algorithm 3 is a 6-approximation algorithm for MCT when  $\mathcal{G}_c$  is node-capacitated and Euclidean with  $C_{DN}(j) = A(i', j') = \infty$  for all  $j \in \mathcal{V}$ , and  $\mathcal{G}_o$  is required to be undirected.*

Finding directed overlays is obviously an NP-hard problem also for node-capacitated networks. Christofides' algorithm holds its approximation factor also in this more general case:

**Proposition 2.1.6.** *Christofides' algorithm is a  $3N$ -approximation algorithm for MCT when  $\mathcal{G}_c$  is node-capacitated and Euclidean.*

## 2.1.4 Numerical Experiments

### 2.1.4.1 Time Simulator

**Algorithm 5:** Time Simulator

---

```

Input:  $(l_{i,j})_{(i,j) \in \mathcal{G}_i}, (T_i^c)_{i \in \mathcal{V}}, (C_{DN}(i))_{i \in \mathcal{V}}$  and  $(C_{UP}(i))_{i \in \mathcal{V}}$ 
Result:  $t \in \mathbb{R}^{N \times K}$ 
1 for  $i \in \mathcal{V}$  do
2    $t_i(0) = 0$ 
3 end
4 for  $k \in \{1, \dots, K\}$  do
5    $t_i(k) = \max_{j \in \mathcal{N}_i^+} \left( t_j(k-1) + l(i, j) + \frac{M}{\min\left(\frac{C_{UP}(i)}{|\mathcal{N}_i^-|}, \frac{C_{DN}(j)}{|\mathcal{N}_j^+|}, A(i', j')\right)} \right)$ .
6    $t_i(k) = t_i(k) + s \times T_c(i)$ 
7 end

```

---

We adapted PyTorch with the MPI backend to run DPASGD (see (2.2)) on a GPU cluster. We also developed a separate network simulator that takes as input an arbitrary underlay topology described in the Graph Modeling Language [Him97] and silos' computation times and calculates the time instants at which local models  $w_i(k)$  are computed according to (2.2). While PyTorch

trains the model as fast as the cluster permits, the network simulator reconstructs the real timeline on the considered underlay. The code is available at <https://github.com/omarfoq/communication-in-cross-silo-fl>

The time simulator reconstructs the wall-clock time. We suppose that we have complete knowledge about the underlay topology, i.e., we know the capacities of all physical links, and the upload and download capacities for each silo. For a given overlay topology  $\mathcal{G}_o = (\mathcal{V}, \mathcal{E}_o)$ , the purpose of the proposed time simulator (Alg. 7) is to compute  $t(k) = (t_i(k))_{1 \leq i \leq N}$ , i.e., the time at which each silo starts computing for the  $k$ -th time. The simulator needs to compute the delay required to send a message with a known size on each physical link of the underlay. This delay is the sum of two terms [Lia+04]:

- **Latency:** it is the time required by the first transmitted bit to travel from the source to the destination. The latency of a link  $(i, j)$  essentially depends on the length of the link and the speed of the light in the link’s transmission medium. We have estimated the latency using the formula proposed in [Gue+04]:  $0.0085 \times \text{distance}(i, j) + 4$ , where the distance is expressed in kilometers and the latency in milliseconds. The latency of a path is the sum of the link latencies.
- **Transmission Delay:** it is the time between the reception of the first bit of the message and the reception of the last bit. It depends on the capacities of each link along the path and the other traffic. We compute it as  $M / \min \left( \frac{C_{\text{UP}}(i)}{|\mathcal{N}_i^-|}, \frac{C_{\text{DN}}(j)}{|\mathcal{N}_j^+|}, A(i', j') \right)$ .

Finally, the simulator also accounts for the total time spent in computation by each node, that is the the product of the number of local steps  $s$  and the time needed to perform one local step (in milliseconds), i.e.,  $s \times T_c(i)$ .

#### 2.1.4.2 Networks and Communication model

We considered three real topologies from *Rocketfuel engine* [Spr+04] (Exodus and Ebone) and from *The Internet Topology Zoo* [Kni+11] (Géant), and two synthetic topologies (AWS North-America and Gaia) built from the geographical locations of AWS data centers [Hsi+17; AWS20] (Table 2.5).

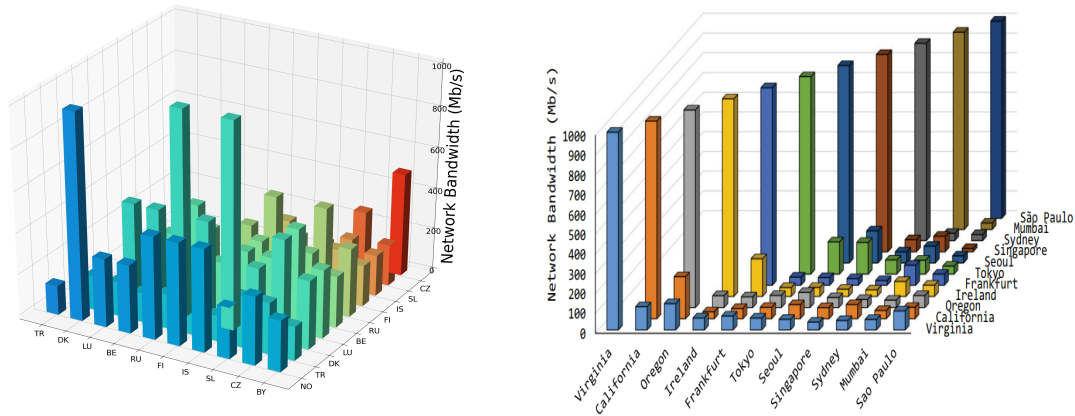
For the synthetic topologies, we consider a full-meshed underlay. We assume all underlays support a shortest path routing with metric the geographical distance (or equivalently the latency). These topologies have between 11 and 87 nodes located in the same continent with the exception of Gaia, which spans four continents. The Géant and Ebone network consist of European cities and Exodus network consist of American cities. We considered that each node is connected to a geographically close silo by a symmetric access link.

Some underlays and examples of overlays are shown in Figures 2.5, 2.4, and 2.6.

#### 2.1.4.3 Datasets and Models

We provide full details on datasets and models used in our experiments. We use multiple datasets spanning a wide range of machine learning tasks (sentiment analysis, language modeling, image classification, handwritten character recognition), including those used in prior work on federated learning [McM+17], and in LEAF [Cal+19] benchmark, and a cross-silo specific dataset based on iNaturalist [Hor+18].





(a) Available bandwidth between some pairs of silos in Géant as computed through our model. (b) Available bandwidth measurements between Gaia sites [Hsi+17, Fig. 2].

Figure 2.3: Our simulator with 1 Gbps capacity links generates a distribution of available bandwidths with the same variability observed in real networks.

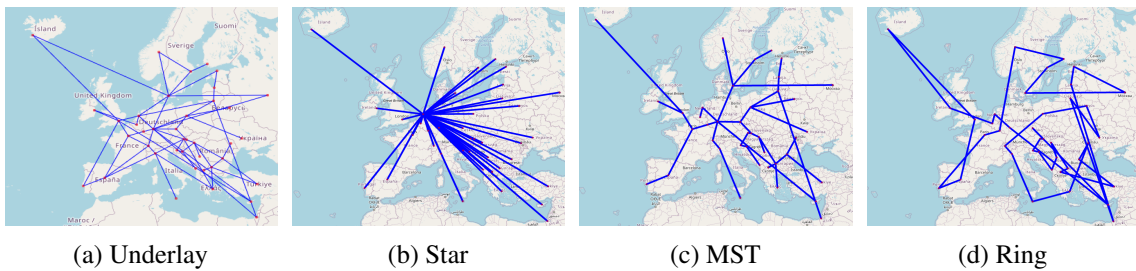


Figure 2.4: Géant Network: the underlay (a) and selected overlays computed when core links have 1 Gbps capacity and access links have 10 Gbps capacity (b-d).

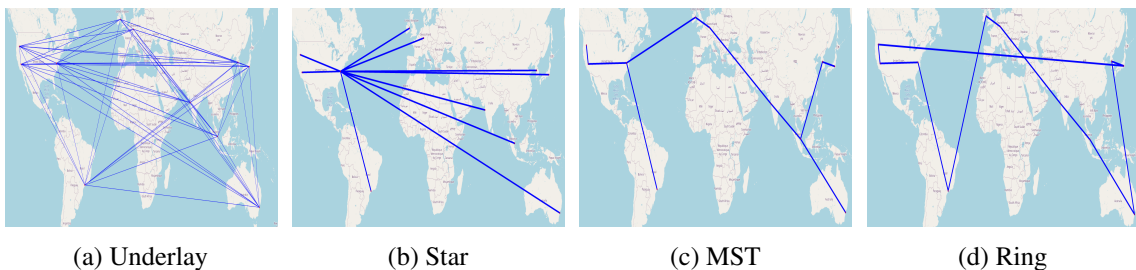


Figure 2.5: Gaia Network: the underlay (a) and selected overlays computed when core links have 1 Gbps capacity and access links have 10 Gbps capacity (b-d).

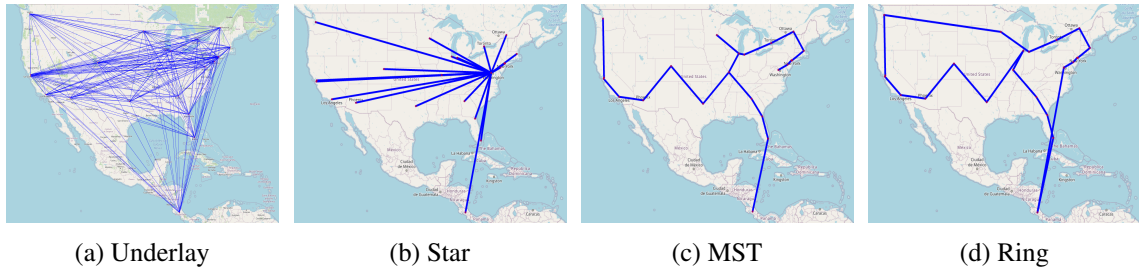


Figure 2.6: AWS-North America Network: the underlay (a) and selected overlays computed when core links have 1 Gbps capacity and access links have 10 Gbps capacity (b-d).

**iNaturalist dataset.** iNaturalist [Hor+18] consists of images from over 5,000 different species of plants and animals. We train the full dataset from iNaturalist 2018 competition <sup>\*</sup> where the geo-locations of these images are provided. For the sub-iNaturalist training, we use a subset of the original iNaturalist dataset, by selecting images containing the 80 most popular species. <sup>†</sup>

In order to simulate a realistic cross-silo environment with non-iid local datasets, one can assign the images to the geographically closest silo obtaining local datasets different in size and in the species represented. This distribution would lead some silos to have no point. We decided then to assign half of the images uniformly at random and half to the closest silo. Moreover, since most of the images in iNaturalist are from North America, for European networks such as Ebone and Géant, we mapped the European cities 90 degrees to the west. Table 2.2 shows that our method generates quite unbalanced data distribution (e.g., for Ebone, one silo can have up to 43 times more images than another one). Moreover Figure 2.7 shows pairwise Jensen-Shanon (JS) divergence [Lin91] across workers labels distributions for different networks both using our method and when the samples are distributed uniformly across the workers. The JS divergence across workers is larger when the samples are distributed following our method in comparison the the uniform case, suggesting that our data is non-iid.

To classify iNaturalist images we fine-tuned pretrained ResNet-18 and ResNet-50 on ImageNet [Den+09] in particular we used the Torchvision [MR10] implementation of ResNet-18 and ResNet-50.

**LEAF datasets.** LEAF [Cal+19] is a benchmark framework for learning in federated settings. We used three LEAF datasets in our experiments on AWS North America network where we took 20% of the samples randomly as our dataset. <sup>‡</sup> Statistics for the corresponding data distributions are in Table 2.3.

- **FEMNIST** (*Federated Extended MNIST*): A 62-class image classification dataset built by partitioning the data of Extended MNIST based on the writer of the digits/characters. In our experiments, we associate each silo with a random number of writers following a lognormal

<sup>\*</sup>iNaturalist 2018 competition is part of the *FGVC*<sup>5</sup> workshop at CVPR ([https://github.com/visipedia/inat\\_comp/blob/master/2018/README.md](https://github.com/visipedia/inat_comp/blob/master/2018/README.md)). This dataset (120GB) contains around 450,000 images of 8142 different classes.

<sup>†</sup>The dataset size is reduced from 120 GB to 18 GB containing 67,000 images. We sub-sample then 20% from this dataset for training.

<sup>‡</sup>Actually, the amount of data we considered is comparable to the federated learning paper [Li+20a]: we considered 10 times more data for FEMNIST and the same amount of data for Sentiment140 and Shakespeare.

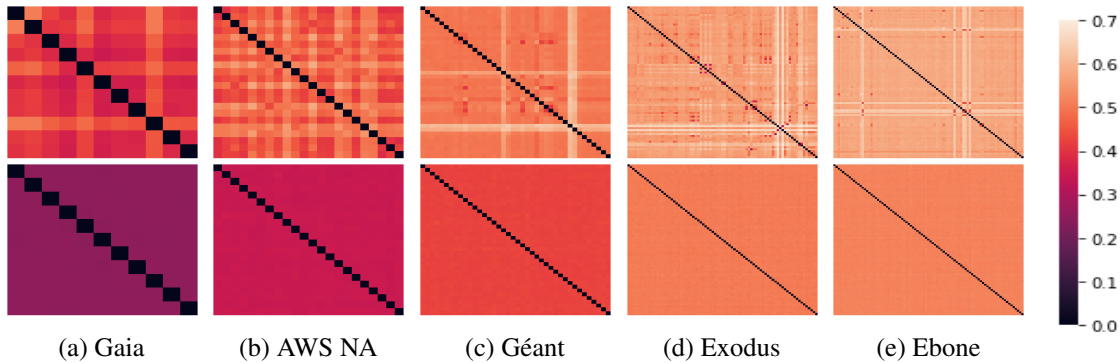


Figure 2.7: Pairwise Jensen-Shannon divergence across workers labels distributions for iNaturalist dataset on different networks. First row is for data distributed with our method and second row is for data uniformly distributed at random.

Table 2.2: Statistics of iNaturalist dataset distribution for different networks.

Network name	Silos	Samples/silo			
		Mean	Stdev	Min	Max
Gaia	11	37795	29986	19344	112745
AWS North America	22	18897	9915	10502	50727
Géant	40	10393	17535	5102	116498
Exodus	79	5262	3368	2710	18454
Ebone	87	4778	11222	2264	98886

distribution with mean equal to 5 and standard deviation equal to 1.5. We train a convolutional neural network, similar to LeNet, with two convolutional layers followed by a max-pooling layer and two fully connected layers.

- **Shakespeare**: A dataset built from *The Complete Works of William Shakespeare*, which is partitioned by the speaking roles [McM+17]. In our experiment, we associate each silo with a random number of speaking roles following a lognormal distribution with mean equal to 5 and standard deviation equal to 1.5. We consider character-level based language modeling on this dataset. The model takes as input a sequence of 200 English characters and predicts the next character. The model embeds the 200 characters into a learnable 16D embedding space, and uses two stacked-GRU layers with 256 hidden units, followed by a densely-connected layer.
- **Sentiment140** [GBH09]: An automatically generated sentiment analysis dataset that annotates tweets based on their emoticons. In our experiment, we associate each silo with a random number of twitter accounts following a lognormal distribution with mean equal to 5 and standard deviation equal to 1.5. We use a two layer bi-directional LSTM binary classifier containing 256 hidden units with pretrained 100D GloVe embedding [PSM14].

Table 2.3: Statistics of LEAF dataset distribution for AWS North America network (22 silos).

Dataset	Samples/silo			
	Mean	Stdev	Min	Max
Shakespeare	36359	6837	24207	50736
FEMNIST	6847	7473	196	26469
Sentiment140	13101	14273	424	50562

#### 2.1.4.4 Implementation Details

**Machines.** The experiments have been run on a CPU/GPU cluster, with different GPUs available (e.g., Nvidia Tesla V100, GeForce GTX 1080 Ti, and Titan X).

**Libraries.** All code is implemented in PyTorch Version 1.4.0. We offer two possibilities for running the code: *sequential* (using only one GPU) and *parallel* (using multiple GPUs). In the parallel setting MPI backend is used for inter-GPU communications.

**Hyperparameters.** The dataset is randomly split into an 80% training set and a 20% testing set. For LEAF and sub-iNaturalist datasets, when training on Gaia, AWS North America, and Géant networks, the initial learning rate is set to 0.001 with Adam optimizer. When training on Exodus and Ebone networks, the initial learning rate is set to 0.1 with SGD optimizer. We decay the learning rate based on the inverse square root of the communication rounds. For iNaturalist dataset, when training on Gaia, AWS North America and Géant networks, the initial learning rate is set to 5e-5 with Adam optimizer. When training on Exodus and Ebone networks, the initial learning rate is set to 0.1 with SGD optimizer. We decay the learning rate by half every epoch.

The batch size is set to 512 for Sentiment140 and Shakespeare datasets, to 128 for Femnist dataset, to 16 for sub-iNaturalist dataset and to 96 for iNaturalist dataset.

**Consensus Matrix.** For a given overlay  $\mathcal{G}_o = (\mathcal{V}, \mathcal{E}_o)$ , the consensus matrix  $\mathbf{A}$  is selected according to the local-degree rule [LB03]. The weight on an arc is based on the larger in-degree of its two incident nodes:

$$\mathbf{A}_{i,j} = \frac{1}{1 + \max\{|\mathcal{N}_i^-|, |\mathcal{N}_j^-|\}}, \quad \forall (i, j) \in \mathcal{E}_o \quad (2.12)$$

$$\mathbf{A}_{i,i} = 1 - \sum_{j \in \mathcal{N}_i^-} \mathbf{A}_{i,j}, \quad \forall i \in \mathcal{V}. \quad (2.13)$$

The matrix  $\mathbf{A}$  so-built is doubly stochastic. The weights can be determined in a fully-distributed way: every node just needs to exchange degree information with its neighbours.

**MATCHA.** We implemented MATCHA as described in [Wan+19a] but for one difference. In MATCHA each matching is selected independently with some probability. With some probability no matching is selected and then no communication occurs. This is equivalent to perform a random number of local steps  $s$  between two communication rounds. In order to compare fairly the different

Table 2.4: Datasets and Models. Mini-batch gradient computation time with NVIDIA Tesla P100.

Dataset	Task	Samples ( $\times 10^3$ )	Batch Size	Model	Parameters ( $\times 10^3$ )	Model Size (Mbits)	Computation Time (ms)
Shakespeare	Next-Character Prediction	4,226	512	Stacked-GRU	840	3.23	389.6
FEMNIST	Image classification	805	128	2-layers CNN	1,207	4.62	4.6
Sentiment140	Sentiment analysis	1,600	512	GloVe + LSTM	4,810	18.38	9.8
sub-iNaturalist	Image classification	13	16	ResNet-18	11,217	42.88	25.4
iNaturalist	Image classification	450	96	ResNet-50	25,557	161.06	946.7

approaches and isolate the effect of  $s$ , we fixed  $s$  also for MATCHA as follows. Silos perform a given number of local steps  $s$  and then, when a communication should occur, matchings are independent sampled until at least one of them is selected. In practice, in our experiments there was always a matching selected with probability almost one, so that the two approaches are not practically distinguishable. Finally, we observe that MATCHA computes the matchings coloring an initial topology, but it is not explained how this initial topology is selected. MATCHA and MATCHA<sup>+</sup> operate exactly in the same way but starting from two different initial topologies: the connectivity graph  $\mathcal{G}_c$  and the underlay  $\mathcal{G}_u$ , respectively. The silos can easily discover the connectivity graph  $\mathcal{G}_c$ ; reconstructing the underlay is much more complicated. Nevertheless, as MATCHA<sup>+</sup> was in general outperforming MATCHA, we showed the results for MATCHA<sup>+</sup>.

We evaluated our solutions on three standard federated datasets from LEAF [Cal+19] and on iNaturalist dataset [Hor+18] with geolocalized images from over 8,000 different species of plants and animals (Table 2.4). We trained ResNet-18 and ResNet-50 on a sub-set and a full-set of iNaturalist dataset respectively to simulate different training environments in silos (i.e., different computation times and model sizes). For LEAF datasets, we generated non-iid data distributions following the procedure in [Li+20a]. For iNaturalist we assigned half of the images uniformly at random and half to the closest silo obtaining local datasets different in size and in the species represented.

### 2.1.4.5 Main Results

Table 2.5 shows the effect of 6 different overlays when training ResNet-18 over sub-iNaturalist in networks with capacities equal to 1 Gbps and 10 Gbps for core links and access links, respectively.\* These overlays are (1) the STAR, corresponding to the usual master-slave setting, where the orchestrator (located at the node with the highest load centrality [Bra08]) averages all models at each communication round, (2) a dynamic topology built from MATCHA starting from the connectivity graph, (3) one built starting from the underlay and denoted as MATCHA<sup>+</sup> (in both cases MATCHA’s parameter  $C_b$  equals 0.5 as in experiments in [Wan+19a]), (4) the minimum spanning tree (MST) from Prop. 2.1.1, (5) the  $\delta$ -minimum bottleneck tree ( $\delta$ -MBST) from Prop. 2.1.5, and (6) the directed RING from Prop. 2.1.6. In this particular setting,  $\delta$ -MBST selects the same overlay as MST. The consensus matrix  $A$  is selected according to the local-degree rule [LB03].

The overlays found by our algorithms achieve a higher throughput (smaller cycle time) than the STAR (the master-slave architecture) and, in most cases, than state-of-the-art MATCHA<sup>(+)</sup>.<sup>†</sup>

\*The delay in the core network is determined by the available bandwidth as in (2.3). Available bandwidths are often limited to tens or hundreds of Mbps even over inter-datacenter links with capacities between 100 Gbps and 1 Tbps [Hsi+17; LL17; Per+17; Kat+18]. By selecting 1 Gbps core links in our simulator, which ignores other traffic, we obtain available bandwidth distributions comparable to those observed in experimental studies like [Hsi+17].

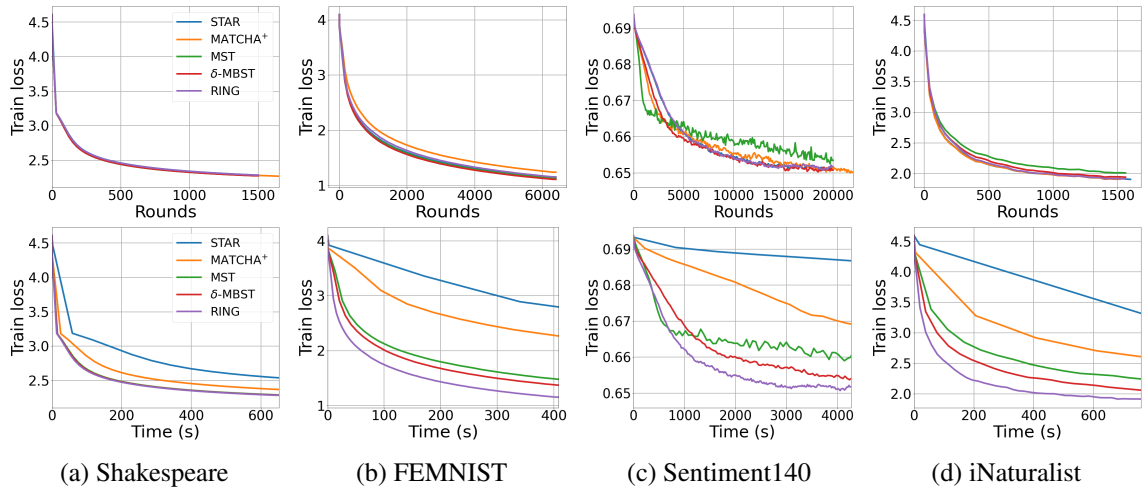
<sup>†</sup>As MATCHA<sup>(+)</sup> overlays are random, we compute their empirical cycle time.

Table 2.5: Sub-iNaturalist training over different networks. 1 Gbps core links capacities, 10 Gbps access links capacities. One local computation step ( $s = 1$ ).

Network name	Silos	Links	Cycle time (ms)					Ring's training speed-up	
			STAR	MATCHA(+)	MST	$\delta$ -MBST	RING	vs STAR	vs MATCHA(+)
Gaia [Hsi+17]	11	55	391	228 (228)	138	138	<b>118</b>	2.65	1.54 (1.54)
AWS North America [AWS20]	22	231	288	124 (124)	90	90	<b>81</b>	3.41	1.47 (1.47)
Géant [20a]	40	61	634	452 (106)	<b>101</b>	<b>101</b>	109	4.85	3.46 (0.81)
Exodus [Mah+02]	79	147	912	593 (142)	145	145	<b>103</b>	8.78	5.71 (1.37)
Ebone [Mah+02]	87	161	902	580 (123)	122	122	<b>95</b>	8.83	6.09 (1.29)

Table 2.6: iNaturalist training over different networks. 1 Gbps core links capacities, 1 Gbps access links capacities. One local computation step ( $s = 1$ ).

Network name	Silos	Links	Cycle time (ms)					Ring's training speed-up	
			STAR	MATCHA(+)	MST	$\delta$ -MBST	RING	vs STAR	vs MATCHA(+)
Gaia [Hsi+17]	11	55	4444	2721 (2721)	1498	1363	<b>1156</b>	3.84	12.10 (12.10)
AWS North America [AWS20]	22	231	7785	4384 (4384)	1441	1297	<b>1119</b>	6.96	23.50 (23.50)
Géant [20a]	40	61	13585	4912 (1894)	1944	1464	<b>1196</b>	11.35	4.10 (1.58)
Exodus [Mah+02]	79	147	26258	6180 (1825)	2078	1481	<b>1194</b>	13.74	2.59 (0.96)
Ebone [Mah+02]	87	161	28753	8045 (1933)	2448	1481	<b>1178</b>	19.52	5.80 (1.39)

Figure 2.8: Effect of overlays on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row) when training four different datasets on AWS North America underlay. 1 Gbps core links capacities, 100 Mbps access links capacities,  $s = 1$ .

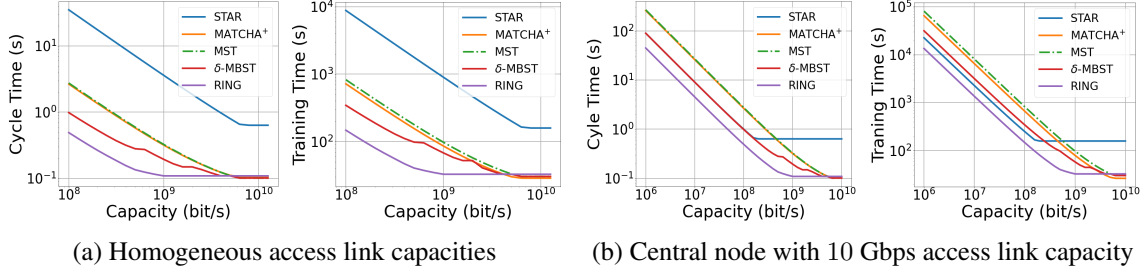


Figure 2.9: Effect of access link capacities on the cycle time and the training time when training iNaturalist on Géant network. 1 Gbps core links capacities,  $s = 1$ . (2.9a): All access links have the same capacity. (2.9b): One node (the center of the star) has a fixed 10 Gbps access link capacity. The training time is the time when training accuracy reaches 55%.

In particular, the RING is between 3.3 ( $\approx 391/118$  on Gaia) and 9.4 ( $\approx 902/95$  on Ebone) times faster than the STAR and between 1.5 and 6 times faster than MATCHA. MATCHA<sup>+</sup> relies on the knowledge of the underlay—probably an unrealistic assumption in an Internet setting—while our algorithms only require information about the connectivity graph. Still, the RING is also faster than MATCHA<sup>+</sup> but on Géant network (where MST is the fastest overlay). From now on, we show only the results for MATCHA<sup>+</sup>, as it outperforms MATCHA.

The final training time is the product of the cycle time and the number of communication rounds required to converge. The overlay also influences the number of communication rounds, with sparser overlays demanding more rounds [NOR18; DAW12]. The last two columns in the table show that this is a second order effect: the RING requires at most 20% more communication rounds than the STAR and then maintains almost the same relative performance in terms of the training time. \* These results (and those in Fig. 2.8) confirm that the number of communication rounds to converge is weakly sensitive to the topology (as already observed in [Lia+17; Lia+18; KSJ19; Luo+19] and partially explained in [POP20a; Ass+19; Neg+20]): overlays should indeed be designed for throughput improvement—as our algorithms do.

Table 2.6 shows the effect of 6 different overlays when training ResNet-50 over (full) iNaturalist in networks with capacities equal to 1 Gbps for core links and access links. †

The same qualitative results hold for other datasets and Fig. 2.8 shows the training loss versus the number of communication rounds (top row) and versus time (bottom row) when training on AWS North America with 100 times slower access links. The advantage of designing the topology on the basis of the underlay characteristics is evident also in this setting.

Figure 2.9 illustrates the effect of access link speeds on the cycle time and the training time. When all silos have the same access link capacity (Fig. 2.9a), for capacity values smaller than 6 Gbps, the RING has the largest throughput followed by  $\delta$ -MBST, MST and MATCHA<sup>+</sup> almost paired, and finally the STAR. In fact, Eq. (2.5) shows that, with  $N$  silos, the RING is up to  $2N$  ( $=80$  for Géant) times faster than the STAR and  $C_b \times \max(\text{degree}(\mathcal{G}_u))$  ( $= 5$  for Géant) times faster than MATCHA<sup>(+)</sup> for slow access links as confirmed in Fig. 2.9a (left plot). RING’s throughput speedups

\*Training time is evaluated as the time to reach a training accuracy equal to 65%, 55%, 55%, 50% and 50% for Gaia, AWS North America, Géant, Exodus, and Ebone networks, respectively. Note that data distribution is different in each networks, so that a different global model is learned when solving Problem (2.1).

†Training time is evaluated as the time to reach a Top 5 training accuracy equal to 18% for Gaia and to 13% for other networks. Full iNaturalist contains 400,000 training images covering 8142 classes. The top 5 training accuracy reached by centralized training ResNet-50 after 50 epochs is up to around 20%.

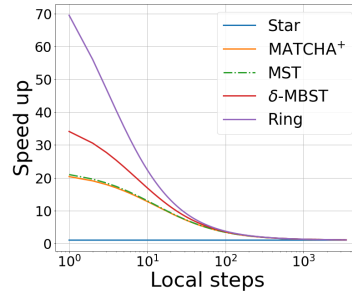


Figure 2.10: Throughput speedup in comparison to the STAR, when training iNaturalist over Exodus network. All links with 1 Gbps capacity.

lead to almost as large training time speedups, even larger than those in Table 2.5: e.g.  $72\times$  in comparison to the STAR and  $5.6\times$  in comparison to MATCHA<sup>+</sup> for 100 Mbps access link capacities. When the most central node (which is also the center of the STAR) maintains a fixed capacity value equal to 10 Gbps (Fig. 2.9b), the STAR performs better, but still is twice slower than the RING and only as fast as  $\delta$ -MBST. This result may appear surprising at first, but it is another consequence of Eq. (2.5). Again the relative performance of different overlays in terms of throughput is essentially maintained when looking at the final training time, with differences across topologies emerging only for those with very close throughputs, i.e., MST and MATCHA<sup>+</sup>, and STAR and  $\delta$ -MBST in the heterogeneous setting.

When local computation requires less time than transmission of model updates, the silo may perform  $s$  local computation steps before a communication round. As  $s$  increases, the total computation time ( $s \times T_c(i)$ ) becomes dominant in (2.3) and the throughput of different overlays become more and more similar (Fig. 2.10). \* Too many local steps may degrade the quality of the final model, and how to tune  $s$  is still an open research area [Wan+20c; WJ19; Lin+20b; Woo+20; Kol+20]. Our next research goal is to study this aspect in conjunction with topology design. Intuitively, a faster overlay reduces the number of local steps needed to amortize the communication cost and may lead to better models given the available time budget for training.

### 2.1.5 Conclusion

We used the theory of max-plus linear systems to propose topology design algorithms that can significantly speed-up federated learning training by maximizing the system throughput. Our results show that this approach is more promising than targeting topologies with the best algebraic connectivity, as MATCHA<sup>(+)</sup> does.

Expanding upon the concept of targeting topologies with high throughput, rather than exclusively aiming for optimal consensus rates, recent research conducted by Takezawa et al. [Tak+23] introduces topologies that boast both rapid consensus rates and minimal maximum degree. Unlike existing topologies, their design, known as Base- $(k+1)$  Graph, guarantees that all nodes reach exact consensus after a finite number of iterations, regardless of the number and maximum degree  $k$ .

Related to the problem of topology design, several recent works [Dan+22; Le +23] have started to consider data heterogeneity when crafting fully-decentralized learning topologies.

\*In Appendix C.2 we show tables similar to Table 2.5 for different values of  $s$ .



## 2.2 Federated Learning under Heterogeneous and Correlated Client Availability

In Section 1.4.1.2, we have seen that clients exhibit a diverse range of characteristics, encompassing disparities in storage capacity, computational resources, and communication capabilities. These disparities arise from variations in hardware specifications (CPU power, memory capacity), network connectivity types (3G, 4G, 5G, WiFi), and power availability (battery levels). In the cross-device settings, these system constraints affect the availability/activity of the clients. In general, the clients exhibit heterogeneous availability patterns, often correlated over time and with other clients. This chapter addresses the problem of heterogeneous and correlated client availability in FL. Our theoretical analysis is the first to demonstrate the negative impact of correlation on FL algorithms' convergence rate and highlights a trade-off between optimization error (related to convergence speed) and bias error (indicative of model quality). To optimize this trade-off, we propose Correlation-Aware FL ( $CA\text{-Fed}$ ), a novel algorithm that dynamically balances the competing objectives of fast convergence and minimal model bias.  $CA\text{-Fed}$  achieves this by dynamically adjusting the aggregation weight assigned to each client and selectively excluding clients with high temporal correlation and low availability. Experimental evaluations on diverse datasets demonstrate the effectiveness of  $CA\text{-Fed}$  compared to state-of-the-art methods. Specifically,  $CA\text{-Fed}$  achieves the best trade-off between training time and test accuracy. By dynamically handling clients with high temporal correlation and low availability,  $CA\text{-Fed}$  emerges as a promising solution to mitigate the detrimental impact of correlated client availability in FL.

### 2.2.1 Introduction

In the original  $FedAvg$  algorithm [McM+17], described in Section 1.2, a central server selects a random subset of clients from the set of available clients and broadcasts them the shared model. The sampled clients perform a number of independent Stochastic Gradient Descent (SGD) steps over their local datasets and send their local model updates back to the server. Then, the server aggregates all the received client updates to produce a new global model, and a new training round begins. In each iteration of  $FedAvg$ , typically a few hundred devices are chosen randomly by the server to participate [Eic+19; Wan+21a].

In real-world scenarios, the availability/activity of clients is dictated by exogenous factors that are beyond the control of the orchestrating server and hard to predict, as previously elucidated in Section 1.4.1.2. For example, only smartphones that are idle, under charge, and connected to broadband networks are commonly allowed to participate in the training process [McM+17; Bon+19]. These eligibility requirements can make the availability of devices correlated over time and space [Eic+19; Din+20; Zhu+22; Doa20]. For example, *temporal correlation* may originate from a smartphone being under charge for a few consecutive hours and then ineligible for the rest of the day. Similarly, the activity of a sensor powered by renewable energy may depend on natural phenomena intrinsically correlated over time (e.g., solar light). *Spatial correlation* refers instead to correlation across different clients, which often emerges as consequence of users' geographical distribution. For example, clients in the same time zone often exhibit similar availability patterns, e.g., due to time-of-day effects.

Temporal correlation in the data sampling procedure is known to negatively affect the performance of ML training even in the centralized setting [Doa+20a; SSY18] and can potentially lead to *catastrophic forgetting*: the data used during the final training phases can have a disproportionate

effect on the final model, “erasing” the memory of previously learned information [MC89; Kir+17]. Catastrophic forgetting has also been observed in FL, where clients in the same geographical area have more similar local data distributions and clients’ participation follows a cyclic daily pattern (leading to spatial correlation) [Eic+19; Din+20; Zhu+22; Tan+22c]. Despite this evidence, a theoretical study of the convergence of FL algorithms under temporally and spatially correlated client participation is still missing.

This section provides a convergence analysis of `FedAvg` [McM+17] under heterogeneous and correlated client availability. We assume that clients’ temporal and spatial availability follows an arbitrary finite-state Markov process: this assumption models a realistic scenario in which the activity of clients is correlated and, at the same time, still allows the analytical tractability of the system. Our theoretical analysis (i) quantifies the negative effect of correlation on the algorithm’s convergence rate through an additional term depending on the spectral properties of the Markov chain; (ii) points out a trade-off between two conflicting objectives: slow convergence to the optimal model, or fast convergence to a biased model, i.e., a model that minimizes an objective function different from the initial target. Guided by insights from the theoretical analysis, we propose `CA-Fed`, a federated learning algorithm which dynamically assigns weights to clients and balances the trade-off between maximizing convergence speed and minimizing model bias. Interesting that `CA-Fed` can decide to ignore clients with low availability and large time-correlation. Our experimental results demonstrate that excluding clients with high temporal correlation and low availability is an effective approach to handle the heterogeneous and correlated client availability in federated learning. Indeed, while `CA-Fed` achieves a comparable maximum test accuracy as the state-of-the-art methods `F3AST` [RVd23] and `AdaFed` [Tan+22a], it achieves a higher time-average and a lower standard deviation of the test accuracy.

The remainder of this section is organized as follows. The next section describes the problem of correlated device availability in FL and discusses the main related works. Section 2.2.3 provides a convergence analysis of `FedAvg` under heterogeneous and correlated device participation. `CA-Fed`, our correlation-aware FL algorithm, is presented in Section 2.2.4. We evaluate `CA-Fed` in Section 2.2.6, comparing it with other state-of-the-art methods. Section 2.2.7 concludes the section.

## 2.2.2 Background and Related Works

We consider a finite set  $\mathcal{K}$  of  $N$  clients. Each client  $k \in \mathcal{K}$  holds a local dataset  $D_k$ . Clients aim to jointly learn the parameters  $\mathbf{w} \in W \subseteq \mathbb{R}^d$  of a global ML model (e.g., the weights of a neural network architecture). During training, the quality of the model with parameters  $\mathbf{w}$  on a data sample  $\xi \in D_k$  is measured by a loss function  $f(\mathbf{w}; \xi)$ . The clients solve, under the orchestration of a central server, the following optimization problem:

$$\min_{\mathbf{w} \in W \subseteq \mathbb{R}^d} \left[ F(\mathbf{w}) := \sum_{k \in \mathcal{K}} \alpha_k F_k(\mathbf{w}) \right], \quad (2.14)$$

where  $F_k(\mathbf{w}) := \frac{1}{|D_k|} \sum_{\xi \in D_k} f(\mathbf{w}; \xi)$  is the average loss computed on client  $k$ ’s local dataset, and  $\boldsymbol{\alpha} = (\alpha_k)_{k \in \mathcal{K}}$  are positive coefficients such that  $\sum_k \alpha_k = 1$ . They represent the *target importance* assigned by the central server to each client  $k$ . Typically  $(\alpha_k)_{k \in \mathcal{K}}$  are set proportional to the clients’ dataset size  $|D_k|$ , such that the objective function  $F$  in (2.14) coincides with the average loss computed on the union of the clients’ local datasets  $D = \cup_{k \in \mathcal{K}} D_k$ .

Under proper assumptions, precised in Section 4.4.3.2, Problem (2.14) admits a unique solution. We use  $\mathbf{w}^*$  (resp.  $F^*$ ) to denote the minimizer (resp. the minimum value) of  $F$ . Moreover, for  $k \in \mathcal{K}$ ,  $F_k$  admits a unique minimizer. We use  $\mathbf{w}_k^*$  (resp.  $F_k^*$ ) to denote the minimizer (resp. the minimum value) of  $F_k$ .

Problem (2.14) is commonly solved through iterative algorithms [McM+17; Wan+21a] requiring multiple communication rounds between the server and the clients. At round  $t > 0$ , the server broadcasts the latest estimate of the global model  $\mathbf{w}_{t,0}$  to the set of available clients ( $\mathcal{A}_t$ ). Client  $k \in \mathcal{A}_t$  updates the global model with its local data through  $E \geq 1$  steps of local Stochastic Gradient Descent (SGD):

$$\mathbf{w}_{t,j+1}^k = \mathbf{w}_{t,j}^k - \eta_t \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) \quad j = 0, \dots, E-1, \quad (2.15)$$

where  $\eta_t > 0$  is an appropriately chosen learning rate, referred to as *local learning rate*;  $\mathcal{B}_{t,j}^k$  is a random batch sampled from client- $k$ 's local dataset at round  $t$  and step  $j$ ;  $\nabla F_k(\cdot, \mathcal{B}) := \frac{1}{|\mathcal{B}|} \sum_{\xi \in \mathcal{B}} \nabla f(\cdot, \xi)$  is an unbiased estimator of the local gradient  $\nabla F_k$ . Then, each client sends its local model update  $\Delta_t^k := \mathbf{w}_{t,E}^k - \mathbf{w}_{t,0}^k$  to the server. The server computes  $\Delta_t := \sum_{k \in \mathcal{A}_t} q_k \cdot \Delta_t^k$ , a weighted average of the clients' local updates with non-negative *aggregation weights*  $\mathbf{q} = (q_k)_{k \in \mathcal{K}}$ . The choice of the aggregation weights defines an aggregation strategy (we will discuss different aggregation strategies later). The aggregated update  $\Delta_t$  can be interpreted as a proxy for  $-\nabla F(\mathbf{w}_{t,0})$ ; the server applies it to the global model:

$$\mathbf{w}_{t+1,0} = \Pi_W(\mathbf{w}_{t,0} + \bar{\eta} \cdot \Delta_t), \quad (2.16)$$

where  $\Pi_W(\cdot)$  denotes the projection over the set  $W$ , and  $\bar{\eta} > 0$  is an appropriately chosen learning rate, referred to as the *server learning rate*.\*

The aggregate update  $\Delta_t$  is generally a biased estimator of the pseudo-gradient  $-\nabla F(\mathbf{w}_{t,0})$ , to which each client  $k$  contributes proportionally to its frequency of appearance in the set  $\mathcal{A}_t$  and its aggregation weight  $q_k$ . More specifically, under proper assumptions specified in Section 2.2.3, we will prove in Theorem 2.2.3 that the update rule described by (2.15) and (2.16) converges to the unique minimizer of a biased global objective  $F_B$ . This objective function depends both on the clients' availability (i.e., on the sequence  $(\mathcal{A}_t)_{t>0}$ ) and on the aggregation strategy (i.e., on  $\mathbf{q} = (q_k)_{k \in \mathcal{K}}$ ):

$$F_B(\mathbf{w}) := \sum_{k=1}^N p_k F_k(\mathbf{w}), \quad \text{with } p_k := \frac{\pi_k q_k}{\sum_{h=1}^N \pi_h q_h}, \quad (2.17)$$

where  $\pi_k$  represents the asymptotic availability of client  $k$ , defined as  $\pi_k := \lim_{t \rightarrow +\infty} \mathbb{P}(k \in \mathcal{A}_t)$ . We denote  $\boldsymbol{\pi} = (\pi_k)_{k \in \mathcal{K}}$ . Moreover, the coefficients  $\mathbf{p} = (p_k)_{k \in \mathcal{K}}$  in (2.17) can be interpreted as the *biased importance* the server is giving to each client  $k$  during training, in general different from the *target importance*  $\boldsymbol{\alpha}$ . In what follows,  $\mathbf{w}_B^*$  (resp.  $F_B^*$ ) denotes the minimizer (resp. the minimum value) of  $F_B$ .

In some large-scale FL applications, like training Google keyboard next-word prediction models, each client participates in training at most for one round. The orchestrator usually selects a few hundred clients at each round for a few thousand rounds (e.g., see [Kai+21, Table 2]), but the available set of clients may include hundreds of millions of Android devices. In this scenario, it is

\*The aggregation rule (2.16) has been considered also in other works, e.g., [NAS18; Red+21; Wan+21a]. In other FL algorithms, the server computes an average of clients' local models. This aggregation rule can be obtained with minor changes to (2.16).

difficult to address the potential bias unless there is some a-priori information about each client’s availability. Anyway, FL can be used by service providers with access to a much smaller set of clients (e.g., smartphone users that have installed a specific app). In this case, a client participates multiple times in training: the orchestrating server may keep track of each client’s availability and try to compensate for the potentially dangerous heterogeneity in their participation.

Much previous effort on federated learning [McM+17; Li+19; Li+20a; CHR22; Fra+21; Tan+22c; Tan+22a; RVd23] considered this problem and, under different assumptions on the clients’ availability (i.e., on  $(\mathcal{A}_t)_{t>0}$ ), designed aggregation strategies that unbiased  $\Delta_t$  through an appropriate choice of  $\mathbf{q}$ . Reference [Li+19] provides the first analysis of  $\text{FedAvg}$  on non-iid data under clients’ partial participation. Their analysis covers both the case when active clients are sampled uniformly at random without replacement from  $\mathcal{K}$  and assigned aggregation weights equal to their target importance (as assumed in [McM+17]), and the case when active clients are sampled iid with replacement from  $\mathcal{K}$  with probabilities  $\alpha$  and assigned equal weights (as assumed in [Li+20a]). However, references [McM+17; Li+19; Li+20a] ignore the variance induced by the clients stochastic availability. The authors of [CHR22] reduce such variance by considering only the clients with important updates, as measured by the value of their norm. References [Tan+22c] and [Fra+21] reduce the aggregation variance through clustered and soft-clustered sampling, respectively.

Some recent works [Tan+22a; RVd23; JWJ22] do not actively pursue the optimization of the unbiased objective. Instead, they derive bounds for the convergence error and propose heuristics to minimize those bounds, potentially introducing some bias. Our work follows a similar development: we compare our algorithm with  $\text{F3AST}$  from [RVd23] and  $\text{AdaFed}$  from [Tan+22a].

The novelty of our study is in considering the spatial and temporal correlation in clients’ availability dynamics. As discussed in Section 2.2.1, such correlations are also introduced by clients’ eligibility criteria, e.g., smartphones being under charge and connected to broadband networks. The effect of correlation has been ignored until now, probably due to the additional complexity in studying FL algorithms’ convergence. To the best of our knowledge, the only exception is [RVd23], which scratches the issue of spatial correlation by proposing two different algorithms for the case when clients’ availabilities are uncorrelated and for the case when they are positively correlated (there is no smooth transition from one algorithm to the other as a function of the degree of correlation).

The effect of temporal correlation on *centralized* stochastic gradient methods has been addressed in [SSY18; Doa+20a; Doa+20b; Doa20]: these works study a variant of stochastic gradient descent where samples are drawn according to a Markov chain. Reference [Doa20] extends its analysis to a FL setting where each client draws samples according to a Markov chain. In contrast, our work does not assume a correlation in the data sampling but rather in the client’s availability. Nevertheless, some of our proof techniques are similar to those used in this line of work and, in particular, we rely on some results in [SSY18].

## 2.2.3 Analysis

### 2.2.3.1 Main assumptions

We consider a time-slotted system where a slot corresponds to a single FL communication round. We assume that clients' availability over the timeslots  $t \in \mathbb{N}$  follows a discrete-time Markov chain  $(\mathcal{A}_t)_{t \geq 0}$ .\*

**Assumption 3.** *The Markov chain  $(\mathcal{A}_t)_{t \geq 0}$  on the  $M$ -finite state space  $\mathcal{M}$  is time-homogeneous, irreducible, and aperiodic. It has transition matrix  $\mathbf{P}$ , stationary distribution  $\rho$ , and has state distribution  $\rho$  at time  $t = 0$ .*

Markov chains have already been used in the literature to model the dynamics of stochastic networks where some nodes or edges in the graph can switch between active and inactive states [MY21; OYJ97]. The previous Markovian assumption, while allowing a great degree of flexibility, still guarantees the analytical tractability of the system. The distance dynamics between the current and the stationary distributions of the Markov process can be characterized in terms of the spectral properties of its transition matrix  $\mathbf{P}$  [LP17]. Let  $\bar{\lambda}_2(\mathbf{P})$  denote the the second largest module of the eigenvalues of  $\mathbf{P}$ . Previous work [SSY18] has shown that:

$$\max_{i,j \in [M]} |[\mathbf{P}^t]_{i,j} - \rho_j| \leq C_P \cdot \lambda(\mathbf{P})^t, \quad \text{for } t \geq T_P, \quad (2.18)$$

where the parameters  $\lambda(\mathbf{P}) := (\bar{\lambda}_2(\mathbf{P}) + 1)/2$ ,  $C_P$ , and  $T_P$  are positive constants whose values are defined in [SSY18, Lemma 1] and reported for completeness in Appendix D.2.2, Lemma D.17.<sup>†</sup> Note that  $\lambda(\mathbf{P})$  quantifies the correlation of the Markov process  $(\mathcal{A}_t)_{t \geq 0}$ : the closer  $\lambda(\mathbf{P})$  is to one, the slower the Markov chain converges to its stationary distribution.

In our analysis, we make the following additional assumptions.

**Assumption 4.** *The hypothesis class  $W$  is convex and compact with diameter  $\text{diam}(W)$ , and contains the minimizers  $\mathbf{w}^*$ ,  $\mathbf{w}_B^*$ ,  $\mathbf{w}_k^*$  in its interior.*

The following assumptions concern clients' local objective functions  $\{F_k\}_{k \in \mathcal{K}}$ . Assumptions 5 and 6 are standard in the literature on convex optimization [BCN18, Sections 4.1, 4.2]. Assumption 7 is a standard hypothesis in the analysis of federated optimization algorithms [Wan+21a, Section 6.1].

**Assumption 5 (L-smoothness).** *The local functions  $\{F_k\}_{k=1}^N$  have  $L$ -Lipschitz continuous gradients:  $F_k(\mathbf{v}) \leq F_k(\mathbf{w}) + \langle \nabla F_k(\mathbf{w}), \mathbf{v} - \mathbf{w} \rangle + \frac{L}{2} \|\mathbf{v} - \mathbf{w}\|_2^2, \forall \mathbf{v}, \mathbf{w} \in W$ .*

**Assumption 6 (Strong convexity).** *The local functions  $\{F_k\}_{k=1}^N$  are  $\mu$ -strongly convex:  $F_k(\mathbf{v}) \geq F_k(\mathbf{w}) + \langle \nabla F_k(\mathbf{w}), \mathbf{v} - \mathbf{w} \rangle + \frac{\mu}{2} \|\mathbf{v} - \mathbf{w}\|_2^2, \forall \mathbf{v}, \mathbf{w} \in W$ .*

**Assumption 7 (Bounded variance).** *The variance of stochastic gradients in each device is bounded:  $\mathbb{E} \|\nabla F_k(\mathbf{w}, \mathcal{B}) - \nabla F_k(\mathbf{w})\|^2 \leq \sigma_k^2, k = 1, \dots, N$ .*

Assumptions 4–7 imply the following properties for the local functions, described by Lemma 2.2.1 (proof in Appendix D.2).

\*In Section 2.2.3.4 we will focus on the case where this chain is the superposition of  $N$  independent Markov chains, one for each client.

<sup>†</sup>Note that (2.18) holds for different definitions of  $\lambda(\mathbf{P})$  as long as  $\lambda(\mathbf{P}) \in (\bar{\lambda}_2(\mathbf{P}), 1)$ . The specific choice for  $\lambda(\mathbf{P})$  changes the values of  $C_P$  and  $T_P$ .

**Lemma 2.2.1.** *Under Assumptions 4–7, there exist constants  $D$ ,  $G$ , and  $H > 0$ , such that, for all  $\mathbf{w} \in W$  and  $k \in \mathcal{K}$ , we have:*

$$\|\nabla F_k(\mathbf{w})\| \leq D, \quad (2.19)$$

$$\mathbb{E}\|\nabla F_k(\mathbf{w}, \mathcal{B})\|^2 \leq G^2, \quad (2.20)$$

$$|F_k(\mathbf{w}) - F_k(\mathbf{w}_B^*)| \leq H. \quad (2.21)$$

Similarly to other works [Li+19; Li+20a; Wan+20b; Wan+21a], we introduce a metric to quantify the heterogeneity of clients' local datasets, typically referred to as *statistical heterogeneity*:

$$\Gamma := \max_{k \in \mathcal{K}} \{F_k(\mathbf{w}^*) - F_k^*\}. \quad (2.22)$$

If the local datasets are identical, the local functions  $\{F_k\}_{k \in \mathcal{K}}$  coincide among them and with  $F$ ,  $\mathbf{w}^*$  is a minimizer of each local function, and  $\Gamma = 0$ . In general,  $\Gamma$  is smaller the closer the distributions the local datasets are drawn from.

### 2.2.3.2 Main theorems

**Theorem 2.2.2** (Decomposing the total error). *Let  $\kappa := L/\mu$ . Under Assumptions 4–6, the optimization error of the target global objective  $\epsilon = F(\mathbf{w}) - F^*$  can be bounded as follows:*

$$\epsilon \leq 2\kappa^2 \left( \underbrace{F_B(\mathbf{w}) - F_B^*}_{:=\epsilon_{\text{opt}}} + \underbrace{F(\mathbf{w}_B^*) - F^*}_{:=\epsilon_{\text{bias}}} \right). \quad (2.23)$$

Moreover, let  $\chi_{\alpha\|\mathbf{p}}^2 := \sum_{k=1}^N (\alpha_k - p_k)^2 / p_k$ . Then:

$$\epsilon_{\text{bias}} \leq \kappa^2 \cdot \underbrace{\chi_{\alpha\|\mathbf{p}}^2}_{:=\epsilon_{\text{bias}}} \cdot \Gamma. \quad (2.24)$$

Theorem 2.2.2 (proof in Appendix D.1) decomposes the error of the target objective ( $\epsilon$ ) as the sum of an optimization error for the biased objective ( $\epsilon_{\text{opt}}$ ) and a bias error ( $\epsilon_{\text{bias}}$ ). The term  $\epsilon_{\text{opt}}$ , evaluated on the trajectory determined by scheme (2.16), quantifies the optimization error associated with the biased objective  $F_B$  and asymptotically vanishes (see Theorem 2.2.3 below). The non-vanishing bias error  $\epsilon_{\text{bias}}$  captures the discrepancy between  $F(\mathbf{w}_B^*)$  and  $F^*$ . This term is bounded by the chi-square divergence  $\chi_{\alpha\|\mathbf{p}}^2$  between the target and biased probability distributions  $\alpha = (\alpha_k)_{k \in \mathcal{K}}$  and  $\mathbf{p} = (p_k)_{k \in \mathcal{K}}$ , and by  $\Gamma$ , that quantifies the degree of heterogeneity of the local functions. When all local functions are identical ( $\Gamma = 0$ ), the bias term  $\epsilon_{\text{bias}}$  also vanishes. For  $\Gamma > 0$ , the bias error can still be controlled by the aggregation weights assigned to the devices. In particular, the bias term vanishes when  $q_k \propto \alpha_k / \pi_k, \forall k \in \mathcal{K}$ . Since it asymptotically cancels the bias error, we refer to this choice as *unbiased aggregation strategy*.

However, in practice, FL training is limited to a finite number of iterations  $T$  (typically a few hundreds [Eic+19; Kai+21]), and the previous asymptotic considerations may not apply. In this regime, the unbiased aggregation strategy can be sub-optimal, since the minimization of  $\epsilon_{\text{bias}}$  not necessarily leads to the minimization of the total error  $\epsilon \leq 2\kappa^2(\epsilon_{\text{opt}} + \epsilon_{\text{bias}})$ . This motivates the analysis of the optimization error  $\epsilon_{\text{opt}}$ .

**Theorem 2.2.3** (Convergence of the optimization error  $\epsilon_{\text{opt}}$ ). *Let Assumptions 3–7 hold and the constants  $M, L, D, G, H, \Gamma, \sigma_k, C_P, T_P$ , and  $\lambda(\mathbf{P})$  defined above. Let  $Q := \sum_{k \in \mathcal{K}} q_k$ . We require a diminishing step-size  $\eta_t > 0$  satisfying:*

$$\eta_1 \leq \frac{1}{2L(1+2EQ)}, \quad \sum_{t=1}^{+\infty} \eta_t = +\infty, \quad \sum_{t=1}^{+\infty} \ln(t) \cdot \eta_t^2 < +\infty. \quad (2.25)$$

Let  $T$  denote the total communication rounds. For  $T \geq T_P$ , the expected optimization error can be bounded as follows:

$$\mathbb{E}[F_B(\bar{\mathbf{w}}_{T,0}) - F_B^*] \leq \underbrace{\frac{\frac{1}{2} \mathbf{q}^\top \boldsymbol{\Sigma} \mathbf{q} + v}{\boldsymbol{\pi}^\top \mathbf{q}} + \psi + \frac{\phi}{\ln(1/\lambda(\mathbf{P}))}}_{:= \bar{\epsilon}_{\text{opt}}}, \quad (2.26)$$

where  $\bar{\mathbf{w}}_{T,0} := \frac{\sum_{t=1}^T \eta_t \mathbf{w}_{t,0}}{\sum_{t=1}^T \eta_t}$ , and

$$\begin{aligned} \boldsymbol{\Sigma} &:= \text{diag}(2(E+1)\sigma_k^2 \pi_k \sum_{t=1}^{+\infty} \eta_t^2), \\ v &:= \frac{2}{E} \text{diam}(W)^2 + \frac{1}{4} M Q \sum_{t=1}^{+\infty} (\eta_t^2 + \frac{1}{t^2}), \\ \psi &:= (4L(1+EQ)\Gamma + 2E^2 G^2) \sum_{t=1}^{+\infty} \eta_t^2 + H(\sum_{t=1}^{T_P-1} \eta_t), \\ \mathcal{J}_t &:= \min \{ \max \{ \lceil \ln(2C_P H t) / \ln(1/\lambda(\mathbf{P})) \rceil, T_P \}, t \}, \\ \phi &:= 2EDGQ \sum_{t=1}^{+\infty} \ln(2C_P H t) \eta_{t-\mathcal{J}_t}^2. \end{aligned}$$

Theorem 2.2.3 (proof in Appendix D.2) proves convergence of the expected biased objective  $F_B$  to its minimum  $F_B^*$  under correlated client participation. Our bound (2.26) captures the effect of correlation through the factor  $\ln(1/\lambda(\mathbf{P}))$ : a high correlation worsens the convergence rate. In particular, we found that the numerator of (2.26) has a quadratic-over-linear fractional dependence on  $\mathbf{q}$ . Minimizing  $\bar{\epsilon}_{\text{opt}}$  leads, in general, to a different choice of  $\mathbf{q}$  than minimizing  $\bar{\epsilon}_{\text{bias}}$ .

### 2.2.3.3 Minimizing the total error $\epsilon \leq 2\kappa^2(\bar{\epsilon}_{\text{opt}} + \bar{\epsilon}_{\text{bias}})$

Our analysis points out a trade-off between minimizing  $\bar{\epsilon}_{\text{opt}}$  or  $\bar{\epsilon}_{\text{bias}}$ . Our goal is to find the optimal aggregation weights  $\mathbf{q}^*$  that minimize the upper bound on total error  $\epsilon(\mathbf{q})$  in (2.23):

$$\begin{aligned} &\underset{\mathbf{q}}{\text{minimize}} && \bar{\epsilon}_{\text{opt}}(\mathbf{q}) + \bar{\epsilon}_{\text{bias}}(\mathbf{q}); \\ &\text{subject to} && \mathbf{q} \geq 0, \\ &&& \|\mathbf{q}\|_1 = Q. \end{aligned} \quad (2.27)$$

In Appendix D.4 we prove that (2.27) is a convex optimization problem, which can be solved with the method of Lagrange multipliers. However, its solution lacks practical utility because the constants in (2.23) and (2.26) (e.g.,  $L, \mu, \Gamma, C_P$ ) are in general problem-dependent and difficult to estimate during training. In particular,  $\Gamma$  poses particular difficulties as it is defined in terms of the minimizer of the target objective  $F$ , but the FL algorithm generally minimizes the biased function  $F_B$ . Moreover, the bound in (2.23), as well as the bound in [Wan+20b], diverges when setting some  $q_k$  values equal to 0, but this divergence is merely an artifact of the proof technique. For more practical considerations, we present the following result (proof in Appendix D.3):

**Theorem 2.2.4** (An alternative bound on the bias error  $\epsilon_{\text{bias}}$ ). *Under the same assumptions of Theorem 2.2.2, define  $\Gamma' := \max_k \{F_k(\mathbf{w}_B^*) - F_k^*\}$ . The following result holds:*

$$\epsilon_{\text{bias}} \leq 4\kappa^2 \cdot \underbrace{d_{TV}^2(\boldsymbol{\alpha}, \mathbf{p})}_{:= \bar{\epsilon}'_{\text{bias}}} \cdot \Gamma', \quad (2.28)$$

where  $d_{TV}(\boldsymbol{\alpha}, \mathbf{p}) := \frac{1}{2} \sum_{k=1}^N |\alpha_k - p_k|$  is the total variation distance between the probability distributions  $\boldsymbol{\alpha}$  and  $\mathbf{p}$ .

The new constant  $\Gamma'$  is defined in terms of  $\mathbf{w}_B^*$ , and then it is easier to evaluate during training. However,  $\Gamma'$  depends on  $\mathbf{q}$ , because it is evaluated at the point of minimum of  $F_B$ . This dependence makes the minimization of the right-hand side of (2.28) more challenging (for example, the corresponding problem is not convex). We study the minimization of the two terms  $\bar{\epsilon}_{\text{opt}}$  and  $\bar{\epsilon}'_{\text{bias}}$  separately and learn some insights, which we use to design the new FL algorithm CA-Feed.

### 2.2.3.4 Minimizing $\bar{\epsilon}_{\text{opt}}$

The minimization of  $\bar{\epsilon}_{\text{opt}}$  is still a convex optimization problem (Appendix D.5). In particular, at the optimum, non-negative weights are set accordingly to  $q_k^* = a(\iota^* \pi_k - \theta^*)$  with  $a$  and  $\iota^*$  positive constants (Appendix D.5.2). It follows that clients with smaller availability get smaller weights in the aggregation. In particular, this suggests that clients with the smallest availability can be excluded from the aggregation, leading to the following guideline:

*Guideline A: to accelerate convergence, we can exclude clients with low availability  $\pi_k$  by setting  $q_k^* = 0$ .*

This guideline can be justified intuitively: updates from clients with low participation may be too sporadic to allow the FL algorithm to keep track of their local objectives. Their updates act as a noise slowing down the algorithm's convergence. It may then be advantageous to exclude these clients.

We observe that the choice of the aggregation weights  $\mathbf{q}$  does not affect the clients' availability process and, in particular,  $\lambda(\mathbf{P})$ . However, if the algorithm excludes some clients, it is possible to consider the state space of the Markov chain that only specifies the availability state of the remaining clients, and this Markov chain may have different spectral properties. For the sake of concreteness, unless otherwise specified, we consider from now on the particular case when the availability of each client  $k$  evolves according to a Markov chain  $(\mathcal{A}_t^k)_{t \geq 0}$  with transition probability matrix  $\mathbf{P}_k$  and these Markov chains are all independent [LP17, Exercise 12.6]. In this case, the aggregate process is described by the product Markov chain  $(\mathcal{A}_t)_{t \geq 0}$  with transition matrix  $\mathbf{P} = \bigotimes_{k \in \mathcal{K}} \mathbf{P}_k$  and  $\lambda(\mathbf{P}) = \max_{k \in \mathcal{K}} \lambda(\mathbf{P}_k)$ , where  $\mathbf{P}_i \otimes \mathbf{P}_j$  denotes the Kronecker product between matrices  $\mathbf{P}_i$  and  $\mathbf{P}_j$  (Appendix D.6.2). In this setting, it is possible to redefine the Markov chain  $(\mathcal{A}_t)_{t \geq 0}$  by taking into account the reduced state space defined by the clients with a non-null aggregation weight, i.e.,  $\mathbf{P}' = \bigotimes_{k' \in \mathcal{K}|q_{k'} > 0} \mathbf{P}_{k'}$  and  $\lambda(\mathbf{P}') = \max_{k' \in \mathcal{K}|q_{k'} > 0} \lambda(\mathbf{P}_{k'})$ , which is potentially smaller w.r.t. the case when all clients participate to the aggregation. These considerations lead to the following guideline:

*Guideline B: to accelerate convergence, we can exclude clients with high correlation (high  $\lambda(\mathbf{P}_k)$ ) by setting their  $q_k^* = 0$ .*

Intuition also supports this guideline. Clients with large  $\lambda(\mathbf{P}_k)$  tend to be available or unavailable for long periods of time. Due to the well-known catastrophic forgetting problem affecting gradient methods [Goo+15; Kem+18], these clients may unfairly steer the algorithm toward their local



objective when they appear at the final stages of the training period. Moreover, their participation in the early stages may be useless, as their contribution will be forgotten during their long absence. The FL algorithm may benefit from directly neglecting such clients.

We observe that Guideline B strictly applies to this specific setting where clients' dynamics are independent (and there is no spatial correlation). We do not provide a corresponding guideline for the case when clients are spatially correlated (we leave this task for future research). However, in this more general setting, it is possible to ignore Guideline B but still draw on Guidelines A and C, or still consider Guideline B if the spatially correlated clients can be grouped in clusters, each cluster evolving as an independent Markov chain (see Section 2.2.6.2, Paragraph 2.2.6.2).

### 2.2.3.5 Minimizing $\bar{\epsilon}'_{\text{bias}}$

The bias error  $\bar{\epsilon}'_{\text{bias}}$  in (2.28) vanishes when the total variation distance between the target importance  $\boldsymbol{\alpha}$  and the biased importance  $\boldsymbol{p}$  is zero, i.e., when  $q_k \propto \alpha_k/\pi_k, \forall k \in \mathcal{K}$ . Then, after excluding the clients that contribute the most to the optimization error and particularly slow down the convergence (Guidelines A and B), we can assign to the remaining clients an aggregation weight inversely proportional to their availability, such that the bias error  $\bar{\epsilon}'_{\text{bias}}$  is minimized.

*Guideline C:* to minimize the bias error, we assign  $q_k^* \propto \alpha_k/\pi_k$  to the clients not excluded by the previous guidelines.

## 2.2.4 Proposed Algorithm

Guidelines A and B in Section 4.4.3.2 suggest that minimizing  $\bar{\epsilon}_{\text{opt}}$  can lead to the exclusion of some available clients from the aggregation step (2.16), in particular those with low availability and/or high correlation. For the remaining clients, Guideline C proposes setting their aggregation weight inversely proportional to their availability to reduce the bias error  $\bar{\epsilon}'_{\text{bias}}$ . Motivated by these insights, we propose **CA-Fed**, a client aggregation strategy that considers the problem of correlated client availability in FL, described in Algorithm 6. **CA-Fed** learns during training which clients to exclude and how to set the aggregation weights of the remaining clients to achieve a good trade-off between  $\bar{\epsilon}_{\text{opt}}$  and  $\bar{\epsilon}'_{\text{bias}}$ . While Guidelines A and B indicate which clients to remove, the exact number of clients to remove at round  $t$  is identified by minimizing  $\epsilon^{(t)}$  as a proxy for the bounds in (2.23) and (2.28):

$$\epsilon^{(t)} := \underbrace{F_B(\boldsymbol{w}_{t,0}) - F_B^*}_{\epsilon_{\text{opt}}} + 4\bar{\kappa}^2 \cdot \underbrace{d_{TV}^2(\boldsymbol{\alpha}, \boldsymbol{p})\Gamma'}_{\bar{\epsilon}'_{\text{bias}}}, \quad (2.29)$$

where  $\bar{\kappa}^2 \geq 0$  is a hyper-parameter that weights the relative importance of the optimization and bias error (see Section 2.2.4.3).

### 2.2.4.1 CA-Fed's core steps

At each communication round  $t$ , the server sends the current model  $\boldsymbol{w}_{t,0}$  to all active clients and each client  $k$  sends back a noisy estimate  $F_k^{(t)}$  of the current loss computed on a batch of samples  $\mathcal{B}_{t,0}^k$ , i.e.,  $F_k^{(t)} = \frac{1}{|\mathcal{B}_{t,0}^k|} \sum_{\xi \in \mathcal{B}_{t,0}^k} f(\boldsymbol{w}_{t,0}, \xi)$  (line 3). The server uses these values and the information about the current set of available clients  $\mathcal{A}_t$  to refine its own estimates of each client's loss ( $\hat{\boldsymbol{F}}^{(t)} = (\hat{F}_k^{(t)})_{k \in \mathcal{K}}$ ), and each client's loss minimum value ( $\hat{\boldsymbol{F}}^* = (\hat{F}_k^*)_{k \in \mathcal{K}}$ ), as well as

**Algorithm 6:** CA-Fed (Correlation-Aware FL)

---

**Input** :  $\mathbf{w}_{0,0}, \boldsymbol{\alpha}, \mathbf{q}^{(0)}, \{\eta_t\}_{t=1}^T, \bar{\eta}, E, \bar{\kappa}^2, \beta, \tau$

- 1 Initialize  $\hat{\mathbf{F}}^{(0)}, \hat{\mathbf{F}}^*, \hat{\Gamma}^{(0)}, \hat{\boldsymbol{\pi}}^{(0)}$ , and  $\hat{\boldsymbol{\lambda}}^{(0)}$ ;
- 2 **for**  $t = 1, \dots, T$  **do**
- 3     Receive set of active client  $A_t$ , loss vector  $\mathbf{F}^{(t)}$ ;
- 4     Update  $\hat{\mathbf{F}}^{(t)}, \hat{\Gamma}^{(t)}, \hat{\boldsymbol{\pi}}^{(t)}$ , and  $\hat{\boldsymbol{\lambda}}^{(t)}$ ;
- 5     Initialize  $\mathbf{q}^{(t)} = \frac{\boldsymbol{\alpha}}{\hat{\boldsymbol{\pi}}^{(t)}}$ ;
- 6      $\mathbf{q}^{(t)} \leftarrow \text{get}(\mathbf{q}^{(t)}, \boldsymbol{\alpha}, \hat{\mathbf{F}}^{(t)}, \hat{\mathbf{F}}^*, \hat{\Gamma}^{(t)}, \hat{\boldsymbol{\pi}}^{(t)}, \hat{\boldsymbol{\lambda}}^{(t)})$ ;
- 7      $\mathbf{q}^{(t)} \leftarrow \text{get}(\mathbf{q}^{(t)}, \boldsymbol{\alpha}, \hat{\mathbf{F}}^{(t)}, \hat{\mathbf{F}}^*, \hat{\Gamma}^{(t)}, \hat{\boldsymbol{\pi}}^{(t)}, -\hat{\boldsymbol{\pi}}^{(t)})$ ;
- 8     **for** client  $\{k \in A_t; q_k^{(t)} > 0\}$ , *in parallel* **do**
- 9         **for**  $j = 0, \dots, E - 1$  **do**
- 10              $\mathbf{w}_{t,j+1}^k = \mathbf{w}_{t,j}^k - \eta_t \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k)$ ;
- 11             **end**
- 12              $\Delta_t^k \leftarrow \mathbf{w}_{t,E}^k - \mathbf{w}_{t,0}^k$ ;
- 13         **end**
- 14          $\mathbf{w}_{t+1,0} \leftarrow \Pi W \mathbf{w}_{t,0} + \bar{\eta} \sum_{k \in A_t} q_k^{(t)} \cdot \Delta_t^k$ ;
- 15     **end**
- 16 **Function**  $\text{get}(\mathbf{q}, \boldsymbol{\alpha}, \mathbf{F}, \mathbf{F}^*, \Gamma, \boldsymbol{\pi}, \boldsymbol{\rho})$  :
  - 17     Sort  $\mathcal{K}$  by descending order in  $\boldsymbol{\rho}$ ;
  - 18      $\hat{\epsilon} \leftarrow \langle \mathbf{F} - \mathbf{F}^*, \boldsymbol{\pi} \odot \mathbf{q} \rangle + 4\bar{\kappa}^2 \cdot d_{TV}^2(\boldsymbol{\alpha}, \boldsymbol{\pi} \odot \mathbf{q})\Gamma$ ;
  - 19     **for**  $k \in \mathcal{K}$  **do**
  - 20          $q_k^+ \leftarrow 0$ ;
  - 21          $\hat{\epsilon}^+ \leftarrow \langle \mathbf{F} - \mathbf{F}^*, \boldsymbol{\pi} \odot \mathbf{q}^+ \rangle + 4\bar{\kappa}^2 \cdot d_{TV}^2(\boldsymbol{\alpha}, \boldsymbol{\pi} \odot \mathbf{q}^+)\Gamma$ ;
  - 22         **if**  $\hat{\epsilon} - \hat{\epsilon}^+ \geq \tau$  **then**
  - 23              $\hat{\epsilon} \leftarrow \hat{\epsilon}^+$ ;
  - 24              $\mathbf{q} \leftarrow \mathbf{q}^+$ ;
  - 25         **end**
  - 26     **return**  $\mathbf{q}$

---

of  $\Gamma'$ ,  $\pi_k$ ,  $\lambda(\mathbf{P}_k)$ , and  $\epsilon^{(t)}$ , denoted as  $\hat{\Gamma}^{(t)}$ ,  $\hat{\pi}_k^{(t)}$ ,  $\hat{\lambda}_k^{(t)}$ , and  $\hat{\epsilon}^{(t)}$ , respectively (possible estimators are described below) (line 4).

The server decides whether excluding clients whose availability pattern exhibits high correlation (high  $\hat{\lambda}_k^{(t)}$ ) (line 6). First, the server considers all clients in descending order of  $\hat{\lambda}_k^{(t)}$  (line 17), and evaluates if, by excluding them (line 20),  $\hat{\epsilon}^{(t)}$  appears to be decreasing by more than a threshold  $\tau \geq 0$  (line 22). Then, the server considers clients in ascending order of  $\hat{\pi}_k^{(t)}$ , and repeats the same procedure to possibly exclude some of the clients with low availability (low  $\hat{\pi}_k^{(t)}$ ) (lines 7).

Once the participating clients (those with  $q_k > 0$ ) have been selected, the server notifies them to proceed updating the current models (lines 9–10) according to (2.15), while the other available clients stay idle. Finally, model's updates are aggregated according to (2.16) (line 14).

### 2.2.4.2 Estimators

We now briefly discuss possible implementation of the estimators  $\hat{F}_k^{(t)}$ ,  $\hat{F}_k^*$ ,  $\hat{\Gamma}^{(t)}$ ,  $\hat{\pi}_k^{(t)}$ , and  $\hat{\lambda}_k^{(t)}$ . Server's estimates for the clients' local losses ( $\hat{\mathbf{F}}^{(t)} = (\hat{F}_k^{(t)})_{k \in \mathcal{K}}$ ) can be obtained from the received active clients' losses ( $\mathbf{F}^{(t)} = (F_k^{(t)})_{k \in \mathcal{A}_t}$ ) through an auto-regressive filter with parameter  $\beta \in (0, 1]$ :

$$\hat{\mathbf{F}}^{(t)} = (\mathbf{1} - \beta \mathbb{1}_{\mathcal{A}_t}) \odot \hat{\mathbf{F}}^{(t-1)} + \beta \mathbb{1}_{\mathcal{A}_t} \odot \mathbf{F}^{(t)}, \quad (2.30)$$

where  $\odot$  denotes the component-wise multiplication between vectors, and  $\mathbb{1}_{\mathcal{A}_t}$  is a  $N$ -dimensions binary vector whose  $k$ -th component equals 1 if and only if client  $k$  is active at round  $t$ , i.e.,  $k \in \mathcal{A}_t$ . The server can estimate client- $k$ 's loss minimum value  $F_k^*$  as  $\hat{F}_k^* = \min_{s \in [0, t]} \hat{F}_k^{(s)}$ . The values of  $F_B(\mathbf{w}_{t,0})$ ,  $F_B^*$ ,  $\Gamma'$ , and  $\epsilon^{(t)}$  can be estimated as follows:

$$\hat{F}_B^{(t)} - \hat{F}_B^* = \langle \hat{\mathbf{F}}^{(t)} - \hat{\mathbf{F}}^*, \hat{\boldsymbol{\pi}}^{(t)} \tilde{\odot} \mathbf{q}^{(t)} \rangle, \quad (2.31)$$

$$\hat{\Gamma}'^{(t)} = \max_{k \in \mathcal{K}} (\hat{F}_k^{(t)} - \hat{F}_k^*), \quad (2.32)$$

$$\hat{\epsilon}^{(t)} = \hat{F}_B^{(t)} - \hat{F}_B^* + 4\bar{\kappa}^2 \cdot d_{TV}^2(\boldsymbol{\alpha}, \hat{\boldsymbol{\pi}}^{(t)} \tilde{\odot} \mathbf{q}^{(t)}) \hat{\Gamma}'^{(t)}. \quad (2.33)$$

where  $\boldsymbol{\pi} \tilde{\odot} \mathbf{q} \in \mathbb{R}^N$ , such that  $(\boldsymbol{\pi} \tilde{\odot} \mathbf{q})_k := \frac{\pi_k q_k}{\sum_{h=1}^N \pi_h q_h}$ ,  $k \in \mathcal{K}$ .

For  $\hat{\pi}_k^{(t)}$ , the server can simply keep track of the total number of times client  $k$  was available up to time  $t$  and compute  $\hat{\pi}_k^{(t)}$  using a Bayesian estimator with beta prior, i.e.,  $\hat{\pi}_k^{(t)} = (\sum_{s \leq t} \mathbb{1}_{k \in \mathcal{A}_s} + n_k) / (t + n_k + m_k)$ , where  $n_k$  and  $m_k$  are the initial parameters of the beta prior.

For  $\hat{\lambda}_k^{(t)}$ , the server can assume the client's availability evolves according to a Markov chain with two states (active and inactive), track the corresponding number of state transitions, and estimate the transition matrix  $\hat{\mathbf{P}}_k^{(t)}$  through a Bayesian estimator similarly to what done for  $\hat{\pi}_k^{(t)}$ . Finally,  $\hat{\lambda}_k^{(t)}$  is obtained computing the eigenvalues of  $\hat{\mathbf{P}}_k^{(t)}$ .

### 2.2.4.3 The role of the hyper-parameter $\bar{\kappa}^2$

Theorems 2.2.2 and 2.2.4 suggest that the condition number  $\kappa^2$  has a significant impact on the minimization of the total error  $\epsilon$ . Our algorithm uses a proxy ( $\epsilon^{(t)}$ ) for the total error (see (2.29)). To account for the effect of  $\kappa^2$ , we introduced the hyper-parameter  $\bar{\kappa}^2 \geq 0$ , which weights the relative importance of the optimization and bias error in (2.29). In practice,  $\bar{\kappa}^2$  controls the number of excluded clients by CA-Fed. A small value of  $\bar{\kappa}^2$  penalizes the bias term in favor of the optimization error, resulting in a larger number of excluded clients. Conversely, the bias term dominates for large values of  $\bar{\kappa}^2$ , and CA-Fed tends to include more clients. Asymptotically, for  $\bar{\kappa}^2 \rightarrow \infty$ , CA-Fed reduces to the *unbiased aggregation strategy*.

## 2.2.5 Fairness, and Computational Cost of CA-Fed

### 2.2.5.1 CA-Fed's computation/communication cost

CA-Fed aims to improve training convergence and not to reduce its computation and communication overhead. Nevertheless, excluding some available clients reduces the overall training cost, as we will discuss in this section referring, for the sake of concreteness, to neural networks' training.

In terms of computation, the available clients not selected for training are only requested to evaluate their local loss on the current model once on a single batch instead than performing  $E$

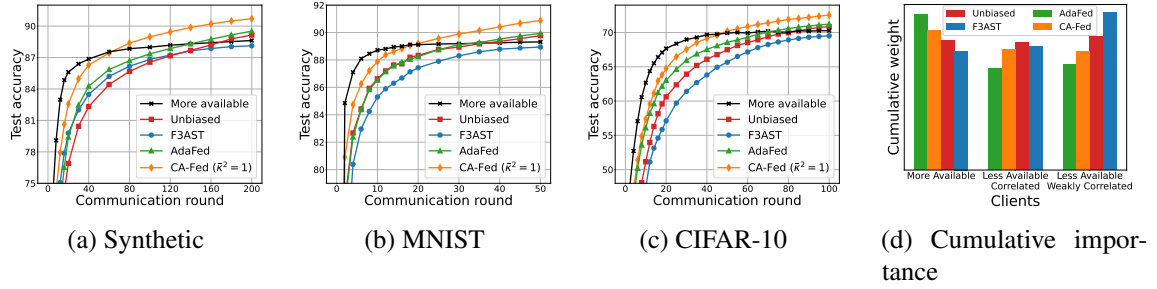


Figure 2.11: Average test accuracy among  $N = 100$  clients achieved by the algorithms on the Synthetic, MNIST, and CIFAR-10 datasets. Cumulative importance assigned by the algorithms to the clients after  $T = 200$  rounds on the Synthetic dataset.

gradient updates, which would require roughly  $2 \times E - 1$  more calculations (because of the forward and backward pass). The selected clients have no extra computation cost as computing the loss corresponds to the forward pass they should, in any case, perform during the first local gradient update.

In terms of communication, the excluded clients only transmit the loss, a single scalar, much smaller than the model update. Conversely, participating clients transmit the local loss and the model update. Still, this additional overhead is negligible and likely fully compensated by the communication savings for the excluded clients.

### 2.2.5.2 About CA-Fed’s fairness

Strategies that exclude clients from the training phase, such as CA-Fed, may raise concerns about fairness. The concept of *fairness* in federated learning does not have a unified definition in the literature [LB22, Chapter 8]. Fairness goals can be established by appropriately selecting the target weights  $\alpha = \{\alpha_k\}_{k \in \mathcal{K}}$  in the definition of the global target objective (2.14). For instance, *per-client fairness* can be achieved by setting  $\alpha_k$  to be equal for every client (i.e.,  $\alpha_k = 1/N$ ), while *per-sample fairness* can be accomplished by setting  $\alpha_k$  proportional to the local dataset size  $|D_k|$  (i.e.,  $\alpha_k = |D_k|/|D|$ ).

Assuming that the global objective in (2.14) truly reflects fairness concerns, then CA-Fed can be considered intrinsically fair. This is because CA-Fed continually focuses on minimizing the total error  $\epsilon := F(\mathbf{w}_T) - F^*$ , which guarantees that the performance objective of the learned model is as close as possible to its optimal value at every time. Although CA-Fed occasionally excludes clients with low availability and high temporal correlation, the optimization problem (2.14) is carefully designed to ensure that the learned model performs well for these clients. As a result, CA-Fed effectively learns a model that is consistently accurate and fair across all clients, regardless of their availability or temporal correlation.

## 2.2.6 Experimental Evaluation

### 2.2.6.1 Experimental Setup

**Federated system simulator** In our experiments, we consider a population of  $N = |\mathcal{K}| = 100$  clients. We model the activity of each client  $k \in \mathcal{K}$  as a two-state homogeneous Markov process with state space  $\mathcal{S} = \{\text{“active”}, \text{“inactive”}\}$ , characterized by a transition matrix  $\mathbf{P}_k$ , a stationary

distribution  $\pi^{(k)}$ , and a second largest absolute eigenvalue  $\bar{\lambda}_2(\mathbf{P}_k)$  (see Appendix D.6.3 for details). Our goal is to simulate realistic dynamics of federated systems featuring varying levels of clients’ availability and correlation. To introduce heterogeneity in clients’ availability patterns, we divide the population in two equally-sized classes: the “more available” clients with a steady-state probability of being active  $\pi_{k,\text{active}} = 1/2 + g$ , and the “less available” clients with  $\pi_{k,\text{active}} = 1/2 - g$ . Here, the parameter  $g \in (0, 1/2)$  controls the degree of heterogeneity in clients’ availability. We furthermore divide each class of clients in two equally-sized sub-classes: clients exhibiting a largely correlated time behavior (in the following referred to as “correlated” clients) that tend to persist in the same state for rather long periods ( $\lambda_k = \nu$  with values of  $\nu$  close to 1), and clients exhibiting a weakly correlated time behavior (referred to as “weakly correlated” clients) that are almost as likely to keep as to change their state at every  $t$  ( $\lambda_k \sim \mathcal{N}(0, \varepsilon^2)$ , with  $\varepsilon$  close to 0). We use  $g = 0.4$ ,  $\nu = 0.9$ , and  $\varepsilon = 10^{-2}$ .

**Datasets and models** We conduct experiments on the LEAF Synthetic dataset [Cal+19], a benchmark for multinomial classification tasks, and on the real-world MNIST [LC10] and CIFAR-10 [Kri09] datasets, respectively for handwritten digits and image recognition tasks. To simulate the statistical heterogeneity present in the federated learning system, we use common approaches in the literature. For the Synthetic dataset, we tune the parameters  $(\gamma, \delta)$ , which control data heterogeneity among clients [Li+19]. For MNIST and CIFAR-10, we distribute samples from the same class across the clients according to a symmetric Dirichlet distribution with parameter  $\varsigma$ , following the same approach as [Wan+20a]. Unless otherwise indicated, we set  $\gamma = \delta = \varsigma = 0.5$ . We use the original training/test data split of MNIST and reserve 20% of the training dataset as the validation dataset. For Synthetic and MNIST, we use a linear classifier with a ridge penalization of parameter  $10^{-2}$ , which corresponds to a strongly convex objective function. For CIFAR-10, we use a neural network with two convolutional and one fully connected layers.

**Benchmarks** We compare CA-Fed, defined in Algorithm 6, with four baselines including two state-of-the-art FL algorithms discussed in Section 2.2.2: 1) *Unbiased*, which aggregates the active clients  $k \in \mathcal{A}_t$  with weights  $q_k = \alpha_k / \pi_k$ ; 2) *More available*, which considers only the “more available” clients and always excludes the “less available” ones; 3) *AdaFed* [Tan+22a], which, similarly to *Unbiased*, aggregates all active clients, but normalizes their aggregation weights (i.e., it considers  $q_k = \frac{\alpha_k / \pi_k}{\sum_{k \in \mathcal{A}_t} \alpha_k / \pi_k}$ ); 4) *F3AST* [RVd23], which, oppositely to *More available*, favors the “less available” clients. For all algorithms, we tuned the learning rates  $\eta, \bar{\eta}$  via grid search. For CA-Fed, we use  $\beta = \tau = 0$ . Unless otherwise specified, we assume that the algorithms can access an oracle providing the true availability parameters for each client: in practice, all the algorithms rely on the exact knowledge of  $\pi_{k,\text{active}}$ ; in addition, CA-Fed also receives  $\lambda(\mathbf{P}_k)$ . In Section 2.2.6.2, Paragraph 2.2.6.2, we will relax this assumption by considering the estimators  $\hat{\pi}_k^{(t)}$  and  $\hat{\lambda}_k^{(t)}$ . The code for this section is available at: <https://github.com/arodio/CA-Fed>.

### 2.2.6.2 Experimental Results

**CA-Fed vs. baselines** Figure 2.11 compares the test accuracy achieved by CA-Fed ( $\bar{\kappa}^2 = 1$ ) and the baselines on the Synthetic (Fig. 2.11a), MNIST (Fig. 2.11b), and CIFAR-10 (Fig. 2.11c) datasets over 10 different runs. Across all three datasets, CA-Fed consistently outperforms the baselines, achieving higher test accuracy (+1.56 pp on Synthetic; +0.94 pp on MNIST; +1.32 pp on

CIFAR-10) compared to the second best performing method, `AdaFed`. These results demonstrate that `CA-Fed` achieves the best balance between convergence speed and test accuracy. For deeper insights into the algorithms’ behavior, Figure 2.11d illustrates the cumulative aggregation weights  $\{\frac{1}{T} \sum_{t=1}^T q_k^{(t)}\}_{k \in \mathcal{K}}$ , representing the cumulative importance that the algorithms assigned to the clients at the end of the training. In Figure 2.11d, we grouped the clients into three categories: “more available”, “less available, weakly correlated”, and “less available, correlated”. By setting the aggregation weights inversely proportional to the clients’ availabilities, `Unbiased` equalizes the importance for all clients (see Fig. 2.11d), but achieves a slower convergence (as shown in Figs. 2.11a, 2.11b, and 2.11c). On the contrary, by excluding all the “less available” clients, `More available` achieves a faster convergence but introduces a non-vanishing bias error  $\epsilon_{\text{bias}}$ , which, in practice, leads to poor accuracy performance. The state-of-the-art algorithm `AdaFed`, similarly to `Unbiased`, considers all the active clients, but normalizes their aggregation weights at each communication round. As a result, similarly to `CA-Fed`, `AdaFed` indeed prioritizes the “more available” clients (as shown in Fig. 2.11d), and then a convergence speed-up could be expected. However, `AdaFed` does not exclude the “less available and correlated” clients, and therefore their presence causes a convergence slowdown. Finally, `F3AST` favors the “less available, correlated” clients and achieves a slower convergence with a non-vanishing bias error, which corresponds to lower accuracy performance. By opportunely excluding some of the “less available and correlated” clients, `CA-Fed` achieves the best test accuracy by the end of the training time.

**Convergence speed vs. Bias error** The trade-off between  $\epsilon_{\text{opt}}$  or  $\epsilon_{\text{bias}}$  discussed in Section 4.4.3.2 is visible in our experiments. In particular, Figure 2.12a compares the test accuracy achieved by `More available`, `Unbiased`, and `CA-Fed` on the Synthetic dataset for  $T = 500$  communication rounds. As expected, by targeting the minimization of  $\epsilon_{\text{opt}}$  and thus excluding the “less available” clients, `More available` achieves the fastest convergence at the expense of a large non-vanishing bias error  $\epsilon_{\text{bias}}$ . On the other hand, by targeting the minimization of  $\epsilon_{\text{bias}}$  and thus equalizing the clients’ importance, `Unbiased` asymptotically removes this error and ultimately achieves the highest test accuracy at communication round  $T = 500$ , but suffers from slower convergence due to the presence of the “correlated” clients. Our algorithm, `CA-Fed`, leverages the trade-off between convergence speed and model bias and achieves fast convergence to the neighborhood of the target objective. To explore this trade-off, in Figure 2.12a, we varied the value of the hyper-parameter  $\bar{\kappa}^2$  in the range  $\{10^{-2}, 10^{-1}, 10^0, 10^1, 10^2\}$ . `CA-Fed` tends to exclude more clients for low values of  $\bar{\kappa}^2$  and achieves a similar convergence rate as `More available` for  $\bar{\kappa}^2 = 10^{-2}$ . For intermediate values of  $\bar{\kappa}^2$ , `CA-Fed` trades a small accuracy decrease for faster convergence (refer, for example, to the curves  $\bar{\kappa}^2 = 10^0, 10^1$ ). For  $\bar{\kappa}^2 = 10^2$ , `CA-Fed` reduces to `Unbiased` (their curves overlap in Fig. 2.12a). Moreover, we observe that the optimal value of  $\bar{\kappa}^2$  depends on the available time for training. Low values of  $\bar{\kappa}^2$  speed-up convergence and then they can be beneficial for short training durations (e.g., `CA-Fed` ( $\bar{\kappa} = 10^{-1}$ ) achieves a higher test accuracy of +2.8 pp with respect to `Unbiased` at communication round  $t = 40$ ). For longer training periods, a larger value of  $\bar{\kappa}^2$  may be preferable as it reduces the bias error and increases the test accuracy (e.g., `CA-Fed` ( $\bar{\kappa} = 10^2$ ) improves of +3.8 pp with respect to `More available` at communication round  $t = 500$ ). Figure 2.12b illustrates the optimal value of  $\bar{\kappa}^2$  for different durations of the training period  $T$ .

**Effect of statistical heterogeneity** The bias error bounds  $\bar{\epsilon}_{\text{bias}}$  and  $\bar{\epsilon}'_{\text{bias}}$  in Theorems 2.2.2 and 2.2.4 are influenced by the degree of heterogeneity among local functions, commonly known as

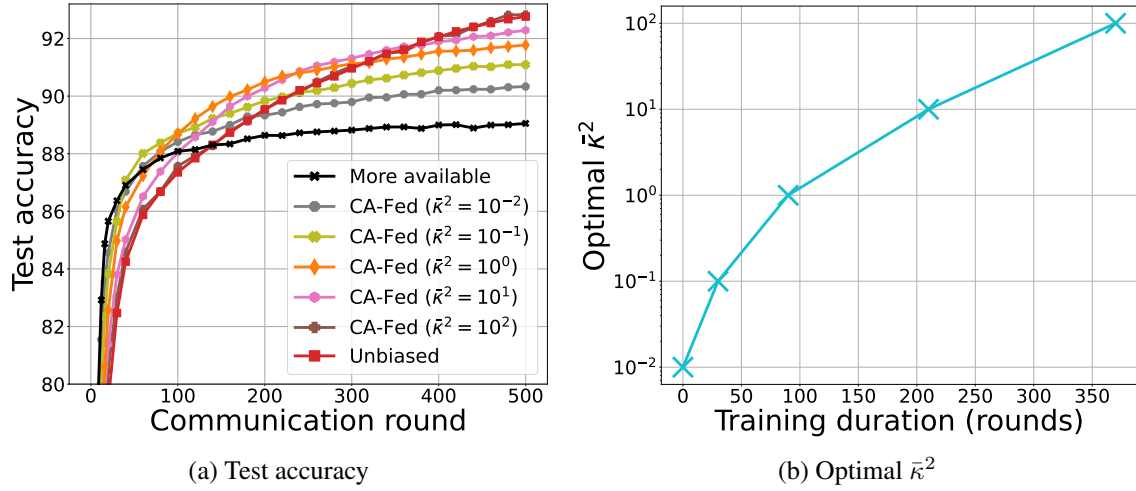


Figure 2.12: *Convergence speed vs. Model bias trade-off for different values of  $\bar{\kappa}^2$  on the Synthetic dataset, for  $\gamma = \delta = 0.5$ .*

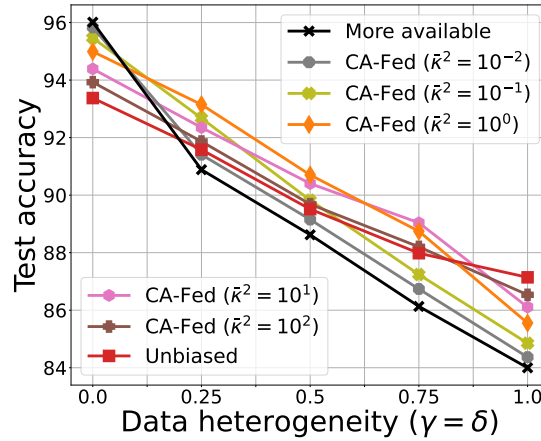


Figure 2.13: *Effects of data heterogeneity on the Synthetic dataset after  $T = 200$  rounds.*

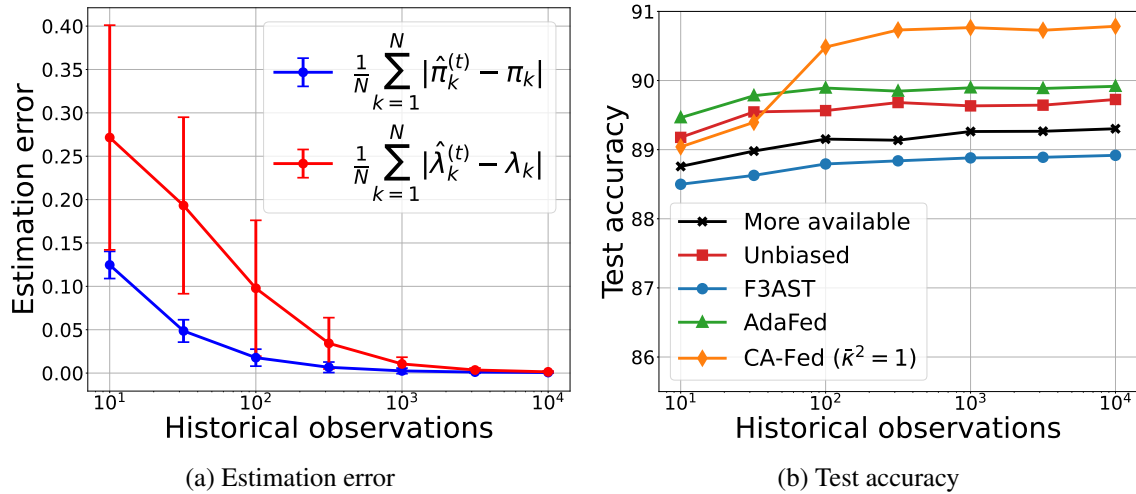


Figure 2.14: *Estimation of the clients' activities  $(\hat{\pi}_k^{(t)}, \hat{\lambda}_k^{(t)})$  for different priors  $t \in \{10^1, 10^{1.5}, 10^2, 10^{2.5}, 10^3, 10^{3.5}, 10^4\}$  and test accuracy after  $T = 50$  rounds on the MNIST dataset.*

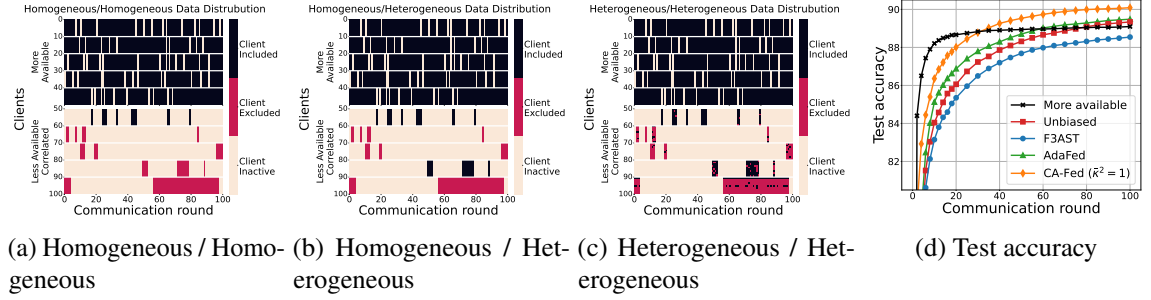


Figure 2.15: Clients’ activities and CA-Fed’s inclusion/exclusion decisions in the presence of *spatial correlation* for different degrees of *intra-cluster/inter-cluster* data distributions. Average test accuracy after  $T = 100$  rounds on the MNIST dataset.

*statistical heterogeneity*, characterized by the constants  $\Gamma$  and  $\Gamma'$  in (2.24) and (2.28), respectively. To control statistical heterogeneity, we manipulate the dissimilarity among the clients’ local datasets, specifically through the parameters  $\gamma$  and  $\delta$  in the case of the Synthetic dataset, as explained in Section 2.2.6.1. Figure 2.13 illustrates the impact of  $\gamma$  and  $\delta$  on the test accuracy achieved by CA-Fed after  $T = 200$  communication rounds on the Synthetic dataset. As expected, in the extreme IID setting (when  $\gamma = \delta = 0$ ),  $\Gamma$  and  $\Gamma'$  are small, and the bias error  $\epsilon_{\text{bias}}$  is negligible. As a result, More available and CA-Fed ( $\bar{\kappa}^2 = 10^{-2}$ ) reach the highest test accuracy, whereas CA-Fed ( $\bar{\kappa}^2 = 10^2$ ) and Unbiased present slow convergence. Nevertheless, More available and CA-Fed ( $\bar{\kappa}^2 = 10^{-2}$ ) perform poorly as the statistical heterogeneity increases (i.e.,  $\gamma = \delta \geq 0.25$ ). In the extreme non-IID setting (when  $\gamma = \delta = 1$ ),  $\Gamma$  and  $\Gamma'$  are large, and  $\epsilon_{\text{bias}}$  dominates. In this case, CA-Fed ( $\bar{\kappa}^2 = 10^2$ ) and Unbiased should be preferred. For  $\gamma = \delta = \{0.25, 0.5, 0.75\}$ , CA-Fed (with  $\bar{\kappa}^2 = 1$  or  $\bar{\kappa}^2 = 10$ ) achieves the highest test accuracy (+1.6 pp, +1.2 pp, and +1.0 pp with respect to Unbiased).

**Estimation of the clients’ availability and correlation** In this experiment, CA-Fed utilizes estimators  $\hat{\pi}_k^{(t)}$  and  $\hat{\lambda}_k^{(t)}$  to estimate the clients’  $\pi_k$  and  $\lambda_k$  values. We employ a Bayesian estimator with a beta prior to estimate  $\hat{P}_k^{(t)}$ , which we generate by observing the evolution of the Markov chain defined by  $P_k$  over  $t'$  time-steps. We compute  $\hat{\pi}_k^{(t)}$  and  $\hat{\lambda}_k^{(t)}$  analytically, following the methodology explained in Section 2.2.4.2 and described in detail in Appendix D.6.3. Figure 2.14a shows the estimation errors  $\frac{1}{N} \sum_{k \in \mathcal{K}} |\hat{\pi}_k^{(t)} - \pi_k|$  and  $\frac{1}{N} \sum_{k \in \mathcal{K}} |\hat{\lambda}_k^{(t)} - \lambda_k|$  as a function of the number of historical observations  $t'$ . As expected, both errors decrease with an increasing number of observations, and the estimation error for  $\lambda_k$  is larger than that for  $\pi_k$ . Furthermore, Figure 2.14b compares the final test accuracy obtained by CA-Fed and the baselines for varying numbers of historical observations  $t' \in \{10^1, 10^{1.5}, 10^2, 10^{2.5}, 10^3, 10^{3.5}, 10^4\}$  when training for  $T = 50$  rounds on the MNIST dataset. In this setting, CA-Fed outperforms the baselines for  $t' \geq 100$ . This value is reasonable, because estimating  $\lambda_k$  requires a number of observations comparable to the expected hitting time for the slowest Markov chain, which is given by  $\max_{k \in \mathcal{K}} \frac{1}{(1-\lambda_k)\pi_k} = 100$ .

**CA-Fed with Spatial Correlation** Although CA-Fed is primarily designed to handle temporal correlation (as discussed in Section 2.2.3.4), we also evaluate its performance in the presence of spatial correlation. In the considered spatially correlated scenario, clients are grouped into clusters, and each cluster  $c \in \mathcal{C}$  is characterized by an underlying Markov chain that determines when



all clients in the cluster are available or unavailable. The Markov chains of different clusters are independent. Let  $\lambda_c$  denote the second-largest eigenvalue in magnitude of cluster  $c$ 's Markov chain. To reduce the eigenvalue of the aggregate Markov chain, CA-Fed needs to exclude all clients in the cluster  $\bar{c} = \arg \max_{c \in \mathcal{C}} \lambda_c$ . In this experiment, we consider a population of  $N = 100$  clients grouped into  $|\mathcal{C}| = 10$  clusters. We equally split the clients, or equivalently, the clusters, into two categories: “more available” with  $\pi_c = 0.9$  and  $\lambda_c = 0$  for  $c = 0, \dots, 4$ , and “less available, correlated” with  $\pi_c = 0.1$  and  $\lambda_c = c/10$  for  $c = 5, \dots, 9$ . In Figures 2.15a, 2.15b, and 2.15c, each pixel represents, for each client  $k \in \mathcal{K}$  and for each communication round, the client's activity (active/inactive) and CA-Fed's decision (included/excluded in training). From the experiments, we observe that CA-Fed's decisions depend on the degree of statistical heterogeneity among clients within a cluster (i.e., *intra-cluster*) and among clusters (i.e., *inter-cluster*). When both the intra-cluster and inter-cluster clients' data distributions are homogeneous, CA-Fed starts considering the clients in cluster  $\bar{c} = 9$  with  $\lambda_{\bar{c}} = 0.9$ , and sequentially excludes, in order, all clients from clusters  $\{9, 8, 7, 6\}$  (as shown in Fig. 2.15a). When the clients' data distributions are homogeneous within clusters, but heterogeneous among clusters (Fig. 2.15b), CA-Fed still excludes all clients from clusters  $c = \{9, 7, 6\}$ , but decides to include clients from cluster  $c = 8$ . This is because these clients happen to have a lower value of  $\hat{F}_k^{(t)} - \hat{F}_k^*$ , and despite having a large  $\lambda_c$ , CA-Fed decides to include them. Finally, when both the intra-cluster and inter-cluster clients' data distributions are heterogeneous (Fig. 2.15c), CA-Fed can partially include clients from the more correlated clusters, even though their  $\lambda_c$  is large. Figure 2.15d compares the test accuracy achieved by CA-Fed and the baselines with spatial correlation in the same setting as in Figure 2.15c. The experimental results show that CA-Fed can operate correctly in the presence of spatial correlation and still outperforms the baselines (+0.6 pp w.r.t. AdaFed).

### 2.2.7 Conclusion

This section presents the first convergence analysis of a FedAvg-like federated learning (FL) algorithm in presence of heterogeneous and correlated client availability. The analysis reveals the detrimental effect of correlation on the convergence rate and highlights a fundamental trade-off between convergence speed and model bias. To navigate this tradeoff, we introduce CA-Fed, a novel FL algorithm, which adaptively manages the conflicting aims of enhancing convergence speed and reducing model bias, with the ultimate objective of maximizing model quality within the constraints of the training time available. CA-Fed achieves this goal by dynamically excluding clients who exhibit high temporal correlation and limited availability, contingent on their data distributions. Indeed, model updates from such clients may act as noise, increasing variance and slowing down the algorithm's convergence. CA-Fed disregards such clients unless their local datasets notably enhance the quality of the final model. The experimental results validate the effectiveness of our strategy, demonstrating that CA-Fed is a versatile and resilient FL algorithm, well-suited to address real-world scenarios characterized by heterogeneous and correlated client availability.

# CHAPTER 3

---

## Personalized Federated Learning

In Chapter 1, we established that in federated learning, data is sourced from clients with varying behaviors and preferences. Consequently, the local data of any single client fails to capture the complete population distribution, resulting in a phenomenon known as *statistical heterogeneity*. The presence of statistical heterogeneity challenges the conventional assumption that clients should collectively train a common model. Therefore, the adoption of personalized models becomes a necessity in federated learning.

This chapter is dedicated to an in-depth exploration of personalized federated learning, wherein we introduce two novel personalization algorithms: `FedEM` (Section 3.5) and `kNN-Per` (Section 3.6). While `FedEM` primarily targets the resolution of statistical heterogeneity, `kNN-Per` goes a step further. In addition to addressing statistical heterogeneity, `kNN-Per` provides a straightforward and efficient approach to tackle *system heterogeneity* and *temporal heterogeneity*. It achieves this by 1) freeing the most powerful clients from the requirement to fully align their model with the weakest ones, and 2) enabling learning in dynamic environments where client data distributions change post-training. This triple focus on statistical, system, and temporal heterogeneity paves the way for more adaptable and efficient personalized federated learning models.

This chapter is based on our works [Mar+21b], published in *Advances in Neural Information Processing Systems 2021 (NeurIPS'21)*, and [Mar+22b], published in the proceedings of the 39<sup>th</sup> *International Conference on Machine Learning (ICML'22)*.

### 3.1 Introduction

As elucidated in Section 1.4, heterogeneity is a core and fundamental challenge in federated learning. Within federated learning ecosystems, clients highly differ both in size and distribution of their local datasets (*statistical heterogeneity*), and in their storage and computational capabilities (*system heterogeneity*). These dual facets pose a challenge to the conventional assumption that all clients should collaborate in training a single, global model—an approach often advocated in numerous seminal papers on federated learning [McM+17; Kon+17b; MSS19]. In fact, the pursuit of training a single global model encounters a fundamental limitation: all clients should be content with a model's architecture constrained by the minimum common capabilities. Even when clients have similar hardware (e.g., they are all smartphones), in presence of statistical heterogeneity, a global model may be arbitrarily bad for some clients, raising important fairness concerns [Li+21]. To illustrate this point, consider a language modeling task where the input sequence is "I love eating." The prediction of the next word can exhibit significant divergence from one client to another due

to their individual datasets and usage patterns. An alternative approach, in lieu of disseminating a single global model to all clients, is to serve each client with a personalized model, potentially tailored to their unique requirements, including a personalized model architecture.

We recall the essential findings presented in Propositions 1.2.4 and 1.2.5, which expound on the generalization aspects of both purely local and global models. Proposition 1.2.4 underscores a key limitation of purely local models, demonstrating their suboptimal performance in scenarios where the local sample size remains notably limited. This scenario frequently arises in federated learning applications, where individual clients possess access to only a small subset of data samples. In contrast, Proposition 1.2.5 shows that the global model generalizes well, by increasing the number of samples linearly in the number of clients, at the cost of a non-vanishing additive bias term resulting from distribution mismatches among clients. Consequently, the global model suffers from a dramatically poor generalization error on local dataset. Addressing this intricate generalization challenge, a potential solution emerges in the form of personalized models—a middle ground between the extremes of global and purely local modeling paradigms. This raises the fundamental question: *what is the optimal tradeoff between personalization and coordination, and how can this delicate equilibrium be achieved?*

Numerous studies within the literature have undertaken the task of providing a theoretical framework to address this critical question. Notably, [SMS20; Man+20] propose cluster users into groups and train a model for each group. Collins et al. [Col+21] study personalized federated learning (PFL) when clients share a global feature representation. [EMS22; DW22] introduce a personalized federated learning approach founded on the detection of collaboration patterns. This method leverages prior knowledge regarding some measure of distance between local data distributions, often relying on Integral Probability Metrics. Clients utilize this information to identify potential collaboration partners and design aggregation schemes that strike a balance between bias and variance. For an in-depth exploration of the personalized federated learning literature, please refer to Section 3.2, where we provide a comprehensive overview.

### 3.1.1 Contributions

The above-mentioned lines of work all stipulate some assumption on the underlying clients' distribution; clustered FL approaches assume the existence of a cluster structure, Collins et al. [Col+21] assumes that clients share a global feature representation, and [EMS22; DW22] require the prior knowledge on some notion of distance between local data distributions. However, it is unclear if an assumption on the data distributions is necessary for collaboration to be provably beneficial. In Section 3.4, we answer this question, and we show that *federated learning is impossible without assumptions on local data distributions*.

Motivated by this negative result, we formulate two general and flexible assumption. The first, named the *mixture assumption* (Assumption 8), stipulates that the data distribution of each client is a mixture of  $M$  underlying distributions. The second, named the *representation assumption* (Assumption 21), stipulates that if two samples have close representations, then their labels are likely to be the same. This is all the more so, the more suitable the global model is for the local distribution.

The mixture assumption is a flexible and generic assumption that encompasses most of the personalized FL approaches previously proposed in the literature (as we show in Section 3.5.2). The proposed formulation has the advantage that each client can benefit from knowledge distilled from all other clients' datasets (even if any two clients can be arbitrarily different from each

other). All clients jointly learn the  $M$  components, while each client learns its personalized mixture weights. We show that federated EM-like algorithms can be used for training under the mixture assumption. In particular, we propose  $\text{FedEM}$  and  $\text{D-FedEM}$  for the client-server and the fully decentralized settings, respectively, and we prove convergence guarantees. Our approach also provides a principled and efficient way to infer personalized models for clients unseen at training time. Our algorithms can easily be adapted to solve more general problems in a novel framework, which can be seen as a federated extension of the centralized surrogate optimization approach in [Mai13]. To the best of our knowledge, our work is the first to propose federated surrogate optimization algorithms with convergence guarantees.

The representation assumption leads to the development of  $\text{kNN-Per}$ , a PFL algorithm based on local memorization.  $\text{kNN-Per}$  combines a global model trained collectively with a  $\text{kNN}$  model on a client’s local datastore. The global model also provides the shared representation used by the local  $\text{kNN}$ . Local memorization at each FL client can capture the client’s local distribution shift with respect to the global distribution. In addition to addressing statistical heterogeneity,  $\text{kNN-Per}$  provides a straightforward and efficient approach to tackle system heterogeneity by relieving the most powerful clients from the obligation to align their model entirely with the weakest ones. Furthermore,  $\text{kNN-Per}$  offers a simple and effective way to address statistical heterogeneity even in a dynamic environment where client’s data distributions change after training. It is indeed sufficient to update the local datastore with new data without the need to retrain the global model. As such, it presents a valuable solution to cope with temporal heterogeneity.

Through extensive experiments on FL benchmark datasets, we show that both algorithms ( $\text{FedEM}$  and  $\text{kNN-Per}$ ) generally yields models that 1) are on average more accurate, 2) are fairer across clients, and 3) generalize better to unseen clients than state-of-the-art personalized and non-personalized FL approaches. Moreover, we demonstrate the ability of  $\text{kNN-Per}$  to address both statistical and system heterogeneity even in a dynamic environment where client’s data distributions change after training.

### 3.1.2 Organization

The rest of this chapter is organized as follows. In Section 3.2, we provide an overview of related work. In Section 3.3, we formalize the problem of personalized federated learning. In Section 3.4, we provide our impossibility result, showing that federated learning is impossible without assumptions on local data distributions. Section 3.5 introduces and analyses the  $\text{FedEM}$  algorithm. In Section 3.5.1, we introduce the underlying assumption of  $\text{FedEM}$ : *the mixture assumption* (Assumption 8), and show that several popular personalization approaches can be obtained as special cases of our mixture-based framework. In Section 3.5.3, we introduce the  $\text{FedEM}$  algorithm and its fully-decentralized version  $\text{D-FedEM}$ , and we state their convergence results. In Section 3.5.4, we present our general federated surrogate optimization framework, used to establish the convergence of  $\text{FedEM}$  and  $\text{D-FedEM}$ . Finally, we provide  $\text{FedEM}$ ’s experimental results in Section 3.5.6. Section 3.6 is dedicated to the presentation and analysis of the  $\text{kNN-Per}$  algorithm. After motivating the use of the local memorization techniques for personalization, we present  $\text{kNN-Per}$  in Section 3.6.1 and provide its generalization bound in Section 3.6.2.  $\text{kNN-Per}$ ’s experimental setup and results are described in Section 3.6.3. Finally, Section 3.7 provides a comparison between  $\text{FedEM}$  and  $\text{kNN-Per}$ , and concluding remarks.

## 3.2 Related Work

We discuss personalized FL approaches for addressing statistical heterogeneity and system heterogeneity.

### 3.2.1 Statistical Heterogeneity

This body of work considers that all clients have the same model architecture but potentially different parameters.

A simple approach to FL personalization is learning first a global model and then fine-tuning its parameters at each client through stochastic gradient descent for a few epochs [Jia+23; YBS22]; we refer later to this approach as `FedAvg+`. `FedAvg+` was later studied by [CC22] and [CCD22]. The global model can then be considered as a meta-model to be used as initialization for a few-shot adaptation at each client. Later work [KBT19; FMO20; Aca+21] has formally established the connection with Model Agnostic Meta Learning (MAML) [Jia+23] and proposed different algorithms to train a more suitable meta-model for local personalization. However, if local distributions are far from the average distribution, a relevant global model does not exist and this approach boils down to every client learning only on its own local data. This issue is formally captured by the generalization bound in [DKM20, Theorem 1].

`ClusteredFL` [SMS20; Gho+20; Man+20] addresses the potential lack of a global model by assuming that clients can be partitioned into several clusters. Clients belonging to the same cluster share the same optimal model, but those models can be arbitrarily different across clusters (see [SMS20, Assumption 2] for a rigorous formulation). During training, clients learn the cluster to which they belong as well as the cluster model. The Clustered FL assumption is also quite limiting, as no knowledge transfer is possible across clusters. In the extreme case where each client has its own optimal local model (recall the example on language modeling), the number of clusters coincides with the number of clients and no federated learning is possible. Our `FedEM` [Mar+21b] can be considered as a soft clustering algorithm, as clients learn personalized models as mixtures of a limited number of component models.

Multi-Task Learning (MTL) has recently emerged as an alternative approach to learn personalized models in the federated setting and allows for more nuanced relations among clients' models [Smi+17; VBT17; ZBT20; HR21; TTN20]. The authors of [Smi+17; VBT17] were the first to frame FL personalization as a MTL problem. In particular, they defined federated MTL as a penalized optimization problem, where the penalization term models relationships among tasks (clients). The work [Smi+17] proposed the MOCHA algorithm for the client-server scenario, while [VBT17; ZBT20] presented decentralized algorithms for the same problem. Unfortunately, these algorithms can only learn simple models (linear models or linear combination of pre-trained models), because of the complex penalization term. Other MTL-based approaches [HR21; Han+20b; TTN20] are able to train more general models at the cost of considering simpler penalization terms (e.g., the distance to the average model), thereby losing the capability to capture complex relations among tasks. Moreover, a general limitation of this line of work is that the penalization term is justified qualitatively and not on the basis of clear statistical assumptions on local data distributions.

An alternative approach is to interpolate a global model and one local model per client [DKM20; CBB21; Man+20]. [Zha+21] extended this idea by letting each client interpolate the local models of other clients with opportune weights learned during training. Our algorithm, `kNN-Per`, also

interpolates a global and a local model, but the global model plays a double role as it is also used to provide a useful representation for the local kNN.

A recent research direction has cast personalization as a stochastic optimization problem involving biased gradients, exemplified by works such as [Cha+22; Gri+21; Bea+21]. These studies revolve around the training of a single client while incorporating biased gradient information from another group of clients. Their results are articulated in terms of quantifying the distance between a client’s objective function and the average objective function of all clients. Similarly, Even et al. [EMS22] study personalized federated learning under the lens of stochastic optimization. They introduce both lower and upper bounds on the number of samples needed from all clients to approximate the generalization error of a specific client and provide corresponding strategies that align with the lower bounds. However, their approach, which relies on gradient filtering, necessitates prior knowledge about the divergence between local data distributions. This divergence is quantified using specific Integral Probability Metrics (IPMs). Similarly, Ding et al. [DW22], concurrently with [EMS22], presents a personalization approach hinging on the detection of collaboration partners. These partners are selected based on an optimization problem that depends on clients’ sample sizes and the divergence between their data distributions, quantified using a specific IPM notion. Distinct from these research lines, our methods do not demand prior quantified knowledge of the divergence between clients’ local distributions.

Overall, although current personalization approaches can lead to superior empirical performance in comparison to a shared global model or individually trained local models, it is still not well understood whether and under which conditions clients are guaranteed to benefit from collaboration.

### 3.2.2 System Heterogeneity

Some FL application scenarios envision clients with highly heterogeneous hardware, like smartphones, IoT devices, edge computing servers, and the cloud. Ideally, each client could learn a potentially different model architecture, suited to its capabilities. Such system heterogeneity has been studied much less than statistical heterogeneity. Some work [Lin+20a; LW19; ZHZ21; ZWY22] proposed to address system heterogeneity by distilling the knowledge from a global teacher to clients’ student models with different architectures. While early methods [LW19; Lin+20a] required the access to an extra (unlabeled) public dataset, more recent ones [ZHZ21; ZWY22] eliminated this requirement.

Some papers [DDT20; Hor+21; PFT21] propose that each client only trains a sub-model of a global model. The sub-model size is determined by the client’s computational capabilities. The approach appears particularly advantageous for convolutional neural networks with clients selecting only a limited subset of channels.

Reference [Tan+22b] followed another approach where devices and server communicate prototypes, i.e., average representations for all samples in a given class, instead of communicating model’s gradients or parameters, allowing each client to have a different model architecture and input space.

To the best of our knowledge, the only existing method that takes into account both system and statistical heterogeneity is  $p\text{FedHN}$  [Sha+21].  $p\text{FedHN}$  feeds local clients representations to a global (across clients) hypernetwork, which can output personalized heterogeneous models. Unfortunately, the hypernetwork has a large memory footprint already for small clients’ models (e.g., the hypernetwork in the experiments in [Sha+21] has 100 more parameters than the output model): it is not clear if  $p\text{FedHN}$  can scale to complex models.

### 3.3 Problem Formulation

We consider a (countable) set  $\mathcal{T}$  of classification (or regression) tasks which represent the set of possible clients. We will use the terms task and client interchangeably. Data at client  $t \in \mathcal{T}$  is generated according to a local distribution  $\mathcal{D}_t$  over  $\mathcal{X} \times \mathcal{Y}$ . Local data distributions  $\{\mathcal{D}_t\}_{t \in \mathcal{T}}$  are in general different, thus it is natural to fit a separate model (hypothesis)  $h_t \in \mathcal{H}$  to each data distribution  $\mathcal{D}_t$ . The goal is thus to solve (in parallel) the following optimization problems

$$\forall t \in \mathcal{T}, \quad \underset{h_t \in \mathcal{H}}{\text{minimize}} \mathcal{L}_{\mathcal{D}_t}(h_t), \quad (3.1)$$

where  $h_t : \mathcal{X} \mapsto \Delta^{|\mathcal{Y}|}$  ( $\Delta^D$  denoting the unitary simplex of dimension  $D$ ),  $l : \Delta^{|\mathcal{Y}|} \times \mathcal{Y} \mapsto \mathbb{R}^+$  is a loss function,\* and  $\mathcal{L}_{\mathcal{D}_t}(h_t) = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} [l(h_t(\mathbf{x}), y)]$  is the true risk of a model  $h_t$  under data distribution  $\mathcal{D}_t$ . For  $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ , we will denote the joint distribution density associated to  $\mathcal{D}_t$  by  $p_t(\mathbf{x}, y)$ , and the marginal densities by  $p_t(\mathbf{x})$  and  $p_t(y)$ .

A set of  $T$  clients  $[T] \triangleq \{1, 2, \dots, T\} \subseteq \mathcal{T}$  participate to the initial training phase; other clients may join the system in a later stage. We denote by  $\mathcal{S}_t = \{s_t^{(i)} = (\mathbf{x}_t^{(i)}, y_t^{(i)})\}_{i=1}^{n_t}$  the dataset at client  $t \in [T]$  drawn i.i.d. from  $\mathcal{D}_t$ , and by  $n = \sum_{t=1}^T n_t$  the total dataset size.

The idea of federated learning is to enable each client to benefit from data samples available at other clients in order to get a better estimation of  $\mathcal{L}_{\mathcal{D}_t}$ , and therefore get a model with a better generalization ability to unseen examples.

### 3.4 An Impossibility Result

We start by showing that some assumptions on the local distributions  $p_t(\mathbf{x}, y)$ ,  $t \in \mathcal{T}$  are needed for federated learning to be possible, i.e., for each client to be able to take advantage of the data at other clients. This holds even if all clients are observed during the initial training phase (i.e.,  $\mathcal{T} = [T]$ ).

Our argument relies on a reduction to an impossibility result for semi-supervised learning (SSL). If clients have arbitrarily different label distributions, the information carried by  $p_{t'}(y|\mathbf{x})$ ,  $t' \in [T] \setminus \{t\}$  is not relevant for client  $t$ , and client  $t$  can only use the information carried by the marginals  $p_{t'}(\mathbf{x})$ . Assuming that these marginals are identical for all clients, federated learning with  $T$  clients is then equivalent to  $T$  SSL problems, where the SSL problem associated with client  $t$  relies on labeled samples in  $\mathcal{S}_t$  and unlabeled samples in  $\mathcal{U}_t = \cup_{t' \in [T] \setminus \{t\}} \{\mathbf{x} : (\mathbf{x}, y) \in \mathcal{S}_{t'}\}$ .<sup>†</sup>

The authors of [BLP08] conjectured that even when the quantity of unlabeled data goes to infinity, the worst-case sample complexity of SSL improves over supervised learning at most by a constant factor that only depends on the hypothesis class [BLP08, Conjecture 4]. Later work has shown the conjecture to hold for the realizable case and hypothesis classes of finite VC dimension [DSS13, Theorem 1], even when the marginal distribution is known [Göp+19, Theorem 2] (whether the conjecture in [BLP08] holds in the agnostic case is still an open problem). The main consequence for FL is that, without further assumptions, a client cannot provably benefit from larger amounts of data available at other clients.

\*In the case of (multi-output) regression, we have  $h_t : \mathcal{X} \mapsto \mathbb{R}^d$  for some  $d \geq 1$  and  $l : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}^+$ .

<sup>†</sup>Note that in FL settings, we have the extra difficulty that client  $t$  cannot have direct access to samples  $\mathcal{U}_t$ , since local data cannot be moved across clients.

## 3.5 Personalized Federated Learning under a Mixture of Distributions

In this section, we propose to study personalized federated learning under the flexible assumption that each local data distribution is a mixture of unknown underlying distributions. This assumption encompasses most of the existing personalized FL approaches and leads to federated EM-like algorithms for both client-server and fully decentralized settings. Moreover, it provides a principled way to serve personalized models to clients not seen at training time. The algorithms' convergence is analyzed through a novel federated surrogate optimization framework, which can be of general interest. Experimental results on FL benchmarks show that our approach provides models with higher accuracy and fairness than state-of-the-art methods.

### 3.5.1 The Mixture Assumption

Motivated by the above impossibility result (Section 3.4), we propose that each local data distribution  $\mathcal{D}_t$  is a mixture of  $M$  underlying distributions  $\tilde{\mathcal{D}}_m$ ,  $1 \leq m \leq M$ , as formalized below.

**Assumption 8.** *There exist  $M$  underlying (independent) distributions  $\tilde{\mathcal{D}}_m$ ,  $1 \leq m \leq M$ , such that for  $t \in \mathcal{T}$ ,  $\mathcal{D}_t$  is mixture of the distributions  $\{\tilde{\mathcal{D}}_m\}_{m=1}^M$  with weights  $\pi_t^* = [\pi_{t1}^*, \dots, \pi_{tM}^*] \in \Delta^M$ , i.e.*

$$z_t \sim \mathcal{M}(\pi_t^*), \quad ((\mathbf{x}_t, y_t) | z_t = m) \sim \tilde{\mathcal{D}}_m, \quad \forall t \in \mathcal{T}, \quad (3.2)$$

where  $\mathcal{M}(\pi)$  is a multinomial (categorical) distribution with parameters  $\pi$ .

Similarly to what was done above, we use  $p_m(\mathbf{x}, y)$ ,  $p_m(\mathbf{x})$ , and  $p_m(y)$  to denote the probability distribution densities associated to  $\tilde{\mathcal{D}}_m$ . We further assume that marginals over  $\mathcal{X}$  are identical.

**Assumption 9.** *For all  $m \in [M]$ , we have  $p_m(\mathbf{x}) = p(\mathbf{x})$ .*

Assumption 9 is not strictly required for our analysis to hold, but, in the most general case, solving Problem (3.1) requires to learn generative models. Instead, under Assumption 9 we can restrict our attention to discriminative models (e.g., neural networks). More specifically, we consider a parameterized set of models  $\tilde{\mathcal{H}}$  with the following properties.

**Assumption 10.**  *$\tilde{\mathcal{H}} = \{h_\theta\}_{\theta \in \mathbb{R}^d}$  is a set of hypotheses parameterized by  $\theta \in \mathbb{R}^d$ , whose convex hull is in  $\mathcal{H}$ . For each distribution  $\tilde{\mathcal{D}}_m$  with  $m \in [M]$ , there exists a hypothesis  $h_{\theta_m^*}$ , such that*

$$\ell(h_{\theta_m^*}(\mathbf{x}), y) = -\log p_m(y|\mathbf{x}) + c, \quad (3.3)$$

where  $c \in \mathbb{R}$ , is a normalization constant.  $\ell(\cdot, \cdot)$  is then the log loss associated to  $p_m(y|\mathbf{x})$ .

We refer to the hypotheses in  $\tilde{\mathcal{H}}$  as *component models* or simply *components*. We denote by  $\Theta^* \in \mathbb{R}^{M \times d}$  the matrix whose  $m$ -th row is  $\theta_m^*$ , and by  $\Pi^* \in \Delta^{T \times M}$  the matrix whose  $t$ -th row is  $\pi_t^* \in \Delta^M$ . Similarly, we will use  $\Theta$  and  $\Pi$  to denote arbitrary parameters.

**Remark 2.** *Assumptions 9–10 are mainly technical and are not required for our approach to work in practice. Experiments in Sec. 3.5.6 show that our algorithms perform well on standard FL benchmark datasets, for which these assumptions do not hold in general.*



Note that, under the above assumptions,  $p_t(\mathbf{x}, y)$  depends on  $\Theta^*$  and  $\pi_t^*$ . Moreover, we can prove (see App. E.1) that the optimal local model  $h_t^* \in \mathcal{H}$  for client  $t$  is a weighted average of models in  $\tilde{\mathcal{H}}$ .

**Proposition 3.5.1.** *Let  $\ell(\cdot, \cdot)$  be the mean squared error loss, the logistic loss or the cross-entropy loss, and  $\check{\Theta}$  and  $\check{\Pi}$  be a solution of the following optimization problem:*

$$\underset{\Theta, \Pi}{\text{minimize}} \mathbb{E}_{t \sim D_{\mathcal{T}}} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} [-\log p_t(\mathbf{x}, y | \Theta, \pi_t)], \quad (3.4)$$

where  $D_{\mathcal{T}}$  is any distribution with support  $\mathcal{T}$ . Under Assumptions 8, 9, and 10, the predictors

$$h_t^* = \sum_{m=1}^M \check{\pi}_{tm} h_{\check{\theta}_m}(\mathbf{x}), \quad \forall t \in \mathcal{T} \quad (3.5)$$

minimize  $\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} [\ell(h_t(\mathbf{x}), y)]$  and thus solve Problem (3.1).

Proposition 3.5.1 suggests the following approach to solve Problem (3.1). First, we estimate the parameters  $\check{\Theta}$  and  $\check{\pi}_t$ ,  $1 \leq t \leq T$ , by minimizing the empirical version of Problem (3.4) on the training data, i.e., minimizing:

$$f(\Theta, \Pi) \triangleq -\frac{\log p(\mathcal{S}_{1:T} | \Theta, \Pi)}{n} \triangleq -\frac{1}{n} \sum_{t=1}^T \sum_{i=1}^{n_t} \log p(s_t^{(i)} | \Theta, \pi_t), \quad (3.6)$$

which is the (negative) likelihood of the probabilistic model (3.2).<sup>\*</sup> Second, we use (3.5) to get the client predictor for the  $T$  clients present at training time. Finally, to deal with a client  $t_{\text{new}} \notin [T]$  not seen during training, we keep the mixture component models fixed and simply choose the weights  $\pi_{t_{\text{new}}}$  that maximize the likelihood of the client data and get the client predictor via (3.5).

### 3.5.2 Relation with Other Personalized Federated Learning Frameworks

Before presenting our federated learning algorithms in Section 3.5, we show that the generative model in Assumption 8 extends some popular multi-task/personalized FL formulations in the literature.

**Clustered Federated Learning [SMS20; Gho+20]** assumes that each client belongs to one among  $C$  clusters and proposes that all clients in the same cluster learn the same model. Our framework recovers this scenario considering  $M = C$  and  $\pi_{tc}^* = 1$  if task (client)  $t$  is in cluster  $c$  and  $\pi_{tc}^* = 0$  otherwise.

**Personalization via model interpolation [Man+20; DKM20]** relies on learning a global model  $h_{\text{glob}}$  and  $T$  local models  $h_{\text{loc},t}$ , and then using at each client the linear interpolation  $h_t = \alpha_t h_{\text{loc},t} + (1 - \alpha_t) h_{\text{glob}}$ . Each client model can thus be seen as a linear combination of  $M = T + 1$  models  $h_m = h_{\text{loc},m}$  for  $m \in [T]$  and  $h_0 = h_{\text{glob}}$  with specific weights  $\pi_{tt}^* = \alpha_t$ ,  $\pi_{t0}^* = 1 - \alpha_t$ , and  $\pi_{tt'}^* = 0$  for  $t' \in [T] \setminus \{t\}$ .

<sup>\*</sup>As the distribution  $D_{\mathcal{T}}$  over tasks in Prop. 3.5.1 is arbitrary, any positively weighted sum of clients' empirical losses could be considered.

**Alternating Structure Optimization [ZCY11].** Alternating structure optimization (ASO) is a popular MTL approach that learns a shared low-dimensional predictive structure on hypothesis spaces from multiple related tasks, i.e., all tasks are assumed to share a common feature space  $P \in \mathbb{R}^{d' \times d}$ , where  $d' \leq \min(T, d)$  is the dimensionality of the shared feature space and  $P$  has orthonormal columns ( $PP^\top = I_{d'}$ ), i.e.,  $P$  is *semi-orthogonal matrix*. ASO leads to the following formulation:

$$\underset{W, P: PP^\top = I_{d'}}{\text{minimize}} \quad \sum_{t=1}^T \sum_{i=1}^{n_t} l(h_{w_t}(\mathbf{x}_t^{(i)}), y_t^{(i)}) + \alpha (\text{tr}(WW^\top) - \text{tr}(WP^\top PW^\top)) + \beta \text{tr}(WW^\top), \quad (3.7)$$

where  $\alpha \geq 0$  is the regularization parameter for task relatedness and  $\beta \geq 0$  is an additional L2 regularization parameter.

When the hypothesis  $(h_\theta)_\theta$  are assumed to be linear, Eq. (3.5) can be written as  $W = \Pi\Theta$ . Writing the LQ decomposition\* of matrix  $\Theta$ , i.e.,  $\Theta = LQ$ , where  $L \in \mathbb{R}^{M \times M}$  is a lower triangular matrix and  $Q \in \mathbb{R}^{M \times d}$  is a semi-orthogonal matrix ( $QQ^\top = I_M$ ), (3.5) becomes  $W = \Pi LQ \in \mathbb{R}^{T \times d}$ , thus,  $W = WQ^\top Q$ , leading to the constraint  $\|W - WQ^\top Q\|_F^2 = \text{tr}(WW^\top) - \text{tr}(WQ^\top QW^\top) = 0$ . If we assume  $\|\theta_m\|_2^2$  to be bounded by a constant  $B > 0$  for all  $m \in [M]$ , we get the constraint  $\text{tr}(WW^\top) \leq TB$ . It means that minimizing  $\sum_{t=1}^T \sum_{i=1}^{n_t} l(h_{w_t}(\mathbf{x}_t^{(i)}), y_t^{(i)})$  under our Assumption 8 can be formulated as the following constrained optimization problem

$$\begin{aligned} & \underset{W, Q: QQ^\top = I_M}{\text{minimize}} \quad \sum_{t=1}^T \sum_{i=1}^{n_t} l(h_{w_t}(\mathbf{x}_t^{(i)}), y_t^{(i)}), \\ & \text{subject to} \quad \text{tr}\{WW^\top\} - \text{tr}\{WQ^\top QW^\top\} = 0, \\ & \quad \quad \quad \text{tr}(WW^\top) \leq TB. \end{aligned} \quad (3.8)$$

Thus, there exists Lagrange multipliers  $\alpha \in \mathbb{R}$  and  $\beta > 0$ , for which Problem (3.8) is equivalent to the following regularized optimization problem

$$\underset{W, Q: QQ^\top = I_M}{\text{minimize}} \quad \sum_{t=1}^T \sum_{i=1}^{n_t} l(h_{w_t}(\mathbf{x}_t^{(i)}), y_t^{(i)}) + \alpha (\text{tr}\{WW^\top\} - \text{tr}\{WQ^\top QW^\top\}) + \beta \text{tr}\{WW^\top\}, \quad (3.9)$$

which is exactly Problem (3.7).

**Federated MTL via task relationships.** The ASO formulation above motivated the authors of [Smi+17] to learn personalized models by solving the following problem

$$\min_{W, \Omega} \sum_{t=1}^T \sum_{i=1}^{n_t} l(h_{w_t}(\mathbf{x}_t^{(i)}), y_t^{(i)}) + \lambda \text{tr}(W\Omega W^\top), \quad (3.10)$$

Two alternative MTL formulations are presented in [Smi+17] to justify Problem (3.10): MTL with probabilistic priors [ZY10] and MTL with graphical models [Lau96]. Both of them can be covered using our Assumption 8 as follows:

- Considering  $T = M$  and  $\Pi = I_M$  in Assumption 8 and introducing a prior on  $\Theta$  of the form

$$\Theta \sim \left( \prod \mathcal{N}(0, \sigma^2 I_d) \right) \mathcal{MN}(I_d \otimes \Omega) \quad (3.11)$$

lead to a formulation similar to MTL with probabilistic priors [ZY10].

\*Note that when  $\Theta$  is a full rank matrix, this decomposition is unique.

- Two tasks  $t$  and  $t'$  are independent if  $\langle \pi_t, \pi_{t'} \rangle = 0$ , thus using  $\Omega_{t,t'} = \langle \pi_t, \pi_{t'} \rangle$  leads to the same graphical model as in [Lau96].

Several personalized FL formulations, e.g., pFedMe [TTN20], FedU [Din+22] and the formulation studied in [HR21] and in [Han+20b], are special cases of formulation (3.11).

### 3.5.3 Federated Expectation-Maximization

#### 3.5.3.1 Centralized Expectation-Maximization

Our goal is to estimate the optimal components' parameters  $\Theta^* = (\theta_m^*)_{1 \leq m \leq M}$  and mixture weights  $\Pi^* = (\pi_t^*)_{1 \leq t \leq T}$  by minimizing the negative log-likelihood  $f(\Theta, \Pi)$  in (3.6). A natural approach to solve such non-convex problems is the Expectation-Maximization algorithm (EM), which alternates between two steps. Expectation steps update the distribution (denoted by  $q_t$ ) over the latent variables  $z_t^{(i)}$  for every data point  $s_t^{(i)} = (\mathbf{x}_t^{(i)}, y_t^{(i)})$  given the current estimates of the parameters  $\{\Theta, \Pi\}$ . Maximization steps update the parameters  $\{\Theta, \Pi\}$  by maximizing the expected log-likelihood, where the expectation is computed according to the current latent variables' distributions.

The following proposition provides the EM updates for our problem (proof in Appendix E.2).

**Proposition 3.5.2.** *Under Assumptions 8 and 9, at the  $k$ -th iteration the EM algorithm updates parameter estimates through the following steps:*

$$\mathbf{E}\text{-step:} \quad q_t^{k+1}(z_t^{(i)} = m) \propto \pi_{tm}^k \cdot \exp\left(-l(h_{\theta_m^k}(\mathbf{x}_t^{(i)}), y_t^{(i)})\right), \quad t \in [T], m \in [M], i \in [n_t] \quad (3.12)$$

$$\mathbf{M}\text{-step:} \quad \pi_{tm}^{k+1} = \frac{\sum_{i=1}^{n_t} q_t^{k+1}(z_t^{(i)} = m)}{n_t}, \quad t \in [T], m \in [M] \quad (3.13)$$

$$\theta_m^{k+1} \in \arg \min_{\theta \in \mathbb{R}^d} \sum_{t=1}^T \sum_{i=1}^{n_t} q_t^{k+1}(z_t^{(i)} = m) l(h_{\theta}(\mathbf{x}_t^{(i)}), y_t^{(i)}), \quad m \in [M] \quad (3.14)$$

The EM updates in Proposition 3.5.2 have a natural interpretation. In the E-step, given current component models  $\Theta^k$  and mixture weights  $\Pi^k$ , (3.12) updates the a-posteriori probability  $q_t^{k+1}(z_t^{(i)} = m)$  that point  $s_t^{(i)}$  of client  $t$  was drawn from the  $m$ -th distribution based on the current mixture weight  $\pi_{tm}^k$  and on how well the corresponding component  $\theta_m^k$  classifies  $s_t^{(i)}$ . The M-step consists of two updates under fixed probabilities  $q_t^{k+1}$ . First, (3.13) updates the mixture weights  $\pi_{tm}^{k+1}$  to reflect the prominence of each distribution  $\tilde{\mathcal{D}}_m$  in  $\mathcal{S}_t$  as given by  $q_t^{k+1}$ . Finally, (3.14) updates the components' parameters  $\Theta^{k+1}$  by solving  $M$  independent, weighted empirical risk minimization problems with weights given by  $q_t^{k+1}$ . These weights aim to construct an unbiased estimate of the true risk over each underlying distribution  $\tilde{\mathcal{D}}_m$  using only points sampled from the client mixtures, similarly to importance sampling strategies used to learn from data with sample selection bias [Sug+07; Cor+08; CMM10; Vog+20].

### 3.5.3.2 Client-Server Algorithm

Federated learning aims to train machine learning models directly on the clients, without exchanging raw data, and thus we should run EM while assuming that only client  $t$  has access to dataset  $\mathcal{S}_t$ . The E-step (3.12) and the  $\Pi$  update (3.13) in the M-step operate separately on each local dataset  $\mathcal{S}_t$  and can thus be performed locally at each client  $t$ . On the contrary, the  $\Theta$  update (3.14) requires interaction with other clients, since the computation spans all data samples  $\mathcal{S}_{1:T}$ .

In this section, we consider a client-server setting, in which each client  $t$  can communicate only with a centralized server (the orchestrator) and wants to learn components' parameters  $\Theta^* = (\theta_m^*)_{1 \leq m \leq M}$  and its own mixture weights  $\pi_t^*$ .

We propose the algorithm FedEM for *Federated Expectation-Maximization* (Algorithm 7). FedEM proceeds through communication rounds similarly to most FL algorithms including FedAvg [McM+17], FedProx [Li+20b], SCAFFOLD [Kar+20a], and pFedMe [TTN20]. At each round, 1) the central server broadcasts the (shared) component models to the clients, 2) each client locally updates components and its personalized mixture weights, and 3) sends the updated components back to the server, 4) the server aggregates the updates. The local update performed at client  $t$  consists in performing the steps in (3.12) and (3.13) and updating the local estimates of  $\theta_m$  through a solver which approximates the exact minimization in (3.14) using only the local dataset  $\mathcal{S}_t$  (see line 13). FedEM can operate with different local solvers—even different across clients—as far as they satisfy some local improvement guarantees (see the discussion in Section 3.5.5). In what follows, we restrict our focus on the practically important case where the local solver performs multiple stochastic gradient descent updates (local SGD [Sti19]).

**Remark 3.** A simultaneously published work [Die+21] proposes a federated EM algorithm (also called FedEM), which does not address personalization but reduces communication requirements by compressing appropriately defined complete data sufficient statistics.

Under the following standard assumptions (see e.g., [Wan+20b]), FedEM converges to a stationary point of  $f$ . Below, we use the more compact notation  $l(\theta; s_t^{(i)}) \triangleq l(h_\theta(\mathbf{x}_t^{(i)}), y_t^{(i)})$ .

**Assumption 11.** The negative log-likelihood  $f$  is bounded below by  $f^* \in \mathbb{R}$ .

**Assumption 12.** (Smoothness) For all  $t \in [T]$  and  $i \in [n_t]$ , the function  $\theta \mapsto l(\theta; s_t^{(i)})$  is  $L$ -smooth and twice continuously differentiable.

**Assumption 13.** (Unbiased gradients and bounded variance) Each client  $t \in [T]$  can sample a random batch  $\xi$  from  $\mathcal{S}_t$  and compute an unbiased estimator  $\mathbf{g}_t(\theta, \xi)$  of the local gradient with bounded variance, i.e.,  $\mathbb{E}_\xi[\mathbf{g}_t(\theta, \xi)] = \frac{1}{n_t} \sum_{i=1}^{n_t} \nabla_\theta l(\theta; s_t^{(i)})$  and  $\mathbb{E}_\xi \|\mathbf{g}_t(\theta, \xi) - \frac{1}{n_t} \sum_{i=1}^{n_t} \nabla_\theta l(\theta; s_t^{(i)})\|^2 \leq \sigma^2$ .

**Assumption 14.** (Bounded dissimilarity) There exist  $\beta$  and  $G$  such that for any set of weights  $\alpha \in \Delta^M$ :

$$\sum_{t=1}^T \frac{n_t}{n} \left\| \frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{m=1}^M \alpha_m \cdot l(\theta; s_t^{(i)}) \right\|^2 \leq G^2 + \beta^2 \left\| \frac{1}{n} \sum_{t=1}^T \sum_{i=1}^{n_t} \sum_{m=1}^M \alpha_m \cdot l(\theta; s_t^{(i)}) \right\|^2.$$

Assumption 14 limits the level of dissimilarity of the different tasks, similarly to what is done in [Wan+20b].

**Algorithm 7:** FedEM: Federated Expectation-Maximization

---

**Input** : Data  $\mathcal{S}_{1:T}$ ; number of mixture components  $M$ ; number of communication rounds  $K$ ; number of local steps  $J$

**Output**:  $\theta_m^K$  for  $1 \in [M]$ ;  $\pi_t^K$  for  $t \in [T]$

// Initialization

- 1 **server** randomly initialize  $\theta_m^0 \in \mathbb{R}^d$  for  $1 \leq m \leq M$ ;
- 2 **for** tasks  $t = 1, \dots, T$  **in parallel over**  $T$  **clients** **do**
- 3 | Randomly initialize  $\pi_t^0 \in \Delta^M$ ;
- 4 **end**

// Main loop

- 5 **for** iterations  $k = 1, \dots, K$  **do**
- 6 | **server** broadcasts  $\theta_m^{k-1}$ ,  $1 \leq m \leq M$  **to the**  $T$  **clients** ;
- 7 | **for** tasks  $t = 1, \dots, T$  **in parallel over**  $T$  **clients** **do**
- 8 | | **for** component  $m = 1, \dots, M$  **do**
- 9 | | | // E-step
- 9 | | | **for** sample  $i = 1, \dots, n_t$  **do**
- 10 | | |  $q_t^k(z_t^{(i)} = m) \leftarrow \frac{\pi_{tm}^k \cdot \exp(-l(h_{\theta_m^k}(\mathbf{x}_t^{(i)}), y_t^{(i)}))}{\sum_{m'=1}^M \pi_{tm'}^k \cdot \exp(-l(h_{\theta_{m'}^k}(\mathbf{x}_t^{(i)}), y_t^{(i)}))}$  ;
- 11 | | | **end**
- 11 | | | // M-step
- 12 | | |  $\pi_{tm}^k \leftarrow \frac{\sum_{i=1}^{n_t} q_t^k(z_t^{(i)} = m)}{n_t}$  ;
- 13 | | |  $\theta_{m,t}^k \leftarrow \text{LocalSolver}(J, m, \theta_m^{k-1}, q_t^k, \mathcal{S}_t)$  ;
- 14 | | | **end**
- 15 | | **client**  $t$  sends  $\theta_{m,t}^k$ ,  $1 \leq m \leq M$  **to the** **server** ;
- 16 | | **end**
- 17 | | **for** component  $m = 1, \dots, M$  **do**
- 18 | | |  $\theta_m^k \leftarrow \sum_{t=1}^T \frac{n_t}{n} \cdot \theta_{m,t}^k$ ;
- 19 | | **end**
- 20 **end**

21 **Function** LocalSolver( $J, m, \theta, q, \mathcal{S}$ ) :

- 22 | **for**  $j = 0, \dots, J - 1$  **do**
- 23 | | Sample indexes  $\mathcal{I}$  uniformly from  $1, \dots, |\mathcal{S}|$ ;
- 24 | |  $\theta \leftarrow \theta - \eta_{k-1,j} \sum_{i \in \mathcal{I}} q(z^{(i)} = m) \cdot \nabla_{\theta} l(h_{\theta}(\mathbf{x}^{(i)}), y^{(i)})$ ;
- 25 | **end**
- 26 | **return**  $\theta$ ;

---

**Theorem 3.5.3.** *Under Assumptions 8–14, when clients use SGD as local solver with learning rate  $\eta = \frac{a_0}{\sqrt{K}}$ , after a large enough number of communication rounds  $K$ , FedEM’s iterates satisfy:*

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \left\| \nabla_{\Theta} f(\Theta^k, \Pi^k) \right\|_F^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{K}}\right), \quad \frac{1}{K} \sum_{k=1}^K \Delta_{\Pi} f(\Theta^k, \Pi^k) \leq \mathcal{O}\left(\frac{1}{K^{3/4}}\right), \quad (3.15)$$

where the expectation is over the random batches samples, and  $\Delta_{\Pi} f(\Theta^k, \Pi^k) \triangleq f(\Theta^k, \Pi^k) - f(\Theta^k, \Pi^{k+1}) \geq 0$ .

Theorem 3.5.3 (proof in App. E.3) expresses the convergence of both sets of parameters ( $\Theta$  and  $\Pi$ ) to a stationary point of  $f$ . Indeed, the gradient of  $f$  with respect to  $\Theta$  becomes arbitrarily small (left inequality in (3.15)) and the update in Eq. (3.13) leads to arbitrarily small improvements of  $f$  (right inequality in (3.15)).

We conclude this section observing that FedEM allows an *unseen client*, i.e., a client  $t_{\text{new}} \notin T$  arriving after the distributed training procedure, to learn its personalized model. The client simply retrieves the learned components’ parameters  $\Theta^K$  and computes its personalized weights  $\pi_{t_{\text{new}}}$  (starting for example from a uniform initialization) through one E-step (3.12) and the first update in the M-step (3.13).

### 3.5.3.3 Fully Decentralized Algorithm

In some cases, clients may want to communicate directly in a peer-to-peer fashion instead of relying on the central server mediation [see Kai+21, Section 2.1]. In fact, fully decentralized schemes may provide stronger privacy guarantees [CB22] and speed-up training as they better use communication resources [Lia+17; Mar+20b] and reduce the effect of stragglers [Neg+19]. For these reasons, they have attracted significant interest recently in the machine learning community [Lia+17; Lia+18; VBT17; Bel+18; Neg+20; Mar+20b; Kol+20]. We refer to [NOR18] for a comprehensive survey of fully decentralized optimization (also known as consensus-based optimization), and to [Kol+20] for a unified theoretical analysis of decentralized SGD.

We propose D-FedEM (Algorithm 8), a *fully decentralized version* of our federated expectation maximization algorithm. As in FedEM, the M-step for  $\Theta$  update is replaced by an approximate maximization step consisting of local updates. The global aggregation step in FedEM (Alg. 7, line 18) is replaced by a partial aggregation step, where each client computes a weighted average of its current components and those of a subset of clients (its *neighborhood*), which may vary over time. The convergence of decentralized optimization schemes requires certain assumptions to guarantee that each client can influence the estimates of other clients over time. We consider the general assumption in [Kol+20, Assumption 4], restated as Assumption 15:

**Assumption 15** ([Kol+20, Assumption 4]). *Symmetric doubly stochastic mixing matrices are drawn at each round  $k$  from (potentially different) distributions  $W^k \sim \mathcal{W}^k$  and there exists two constants  $p \in (0, 1]$ , and integer  $\tau \geq 1$  such that for all  $\Xi \in \mathbb{R}^{M \times d \times T}$  and all integers  $l \in \{0, \dots, K/\tau\}$ :*

$$\mathbb{E} \left\| \Xi W_{l,\tau} - \bar{\Xi} \right\|_{\mathcal{F}}^2 \leq (1-p) \left\| \Xi - \bar{\Xi} \right\|_{\mathcal{F}}^2, \quad (3.16)$$

where  $W_{l,\tau} \triangleq W^{(l+1)\tau-1} \dots W^{l\tau}$ ,  $\bar{\Xi} \triangleq \Xi \frac{\mathbf{1}\mathbf{1}^T}{T}$ , and the expectation is taken over the random distributions  $W^k \sim \mathcal{W}^k$ .

---

**Algorithm 8:** D-FedEM: Fully Decentralized Federated Expectation-Maximization
 

---

**Input** : Data  $\mathcal{S}_{1:T}$ ; number of mixture components  $M$ ; number of iterations  $K$ ; number of local steps  $J$ ; mixing matrix distributions  $\mathcal{W}^k$  for  $k \in [K]$

**Output** :  $\theta_{m,t}^K$  for  $m \in [M]$  and  $t \in [T]$ ;  $\pi_t$  for  $t \in [T]$

// Initialization

**1 for tasks**  $t = 1, \dots, T$  **in parallel over**  $T$  **clients do**

**2** | Randomly initialize  $\Theta_t = (\theta_{m,t})_{1 \leq m \leq M} \in \mathbb{R}^{M \times d}$ ;

**3** | Randomly initialize  $\pi_t^0 \in \Delta^M$ ;

**4 end**

// Main loop

**5 for iterations**  $k = 1, \dots, K$  **do**

    // Select the communication topology and the aggregation weights

**6** | Sample  $W^{k-1} \sim \mathcal{W}^{k-1}$ ;

**7 for tasks**  $t = 1, \dots, T$  **in parallel over**  $T$  **clients do**

**8** | **for component**  $m = 1, \dots, M$  **do**

    // E-step

**9** | **for sample**  $i = 1, \dots, n_t$  **do**

**10** | |  $q_t^k(z_t^{(i)} = m) \leftarrow \frac{\pi_{tm}^k \cdot \exp(-l(h_{\theta_m^k}(\mathbf{x}_t^{(i)}), y_t^{(i)}))}{\sum_{m'=1}^M \pi_{tm'}^k \cdot \exp(-l(h_{\theta_{m'}^k}(\mathbf{x}_t^{(i)}), y_t^{(i)}))}$ ;

**11** | | **end**

    // M-step

**12** | |  $\pi_{tm}^k \leftarrow \frac{\sum_{i=1}^{n_t} q_t^k(z_t^{(i)} = m)}{n_t}$ ;

**13** | |  $\theta_{m,t}^{k-\frac{1}{2}} \leftarrow \text{LocalSolver}(J, m, \theta_{m,t}^{k-1}, q_t^k, \mathcal{S}_t, t)$ ;

**14** | | **end**

**15** | | Send  $\theta_{m,t}^{k-\frac{1}{2}}, 1 \leq m \leq M$  to neighbors;

**16** | | Receive  $\theta_{m,s}^{k-\frac{1}{2}}, 1 \leq m \leq M$  from neighbors;

**17** | | **for component**  $m = 1, \dots, M$  **do**

**18** | | |  $\theta_{m,t}^k \leftarrow \sum_{s=1}^T w_{s,t}^{k-1} \cdot \theta_{m,s}^{k-\frac{1}{2}}$ ;

**19** | | **end**

**20** | **end**

**21 end**

**22 Function** LocalSolver( $J, m, \theta, q, \mathcal{S}, t$ ):

**23** | **for**  $j = 0, \dots, J - 1$  **do**

**24** | | Sample indexes  $\mathcal{I}$  uniformly from  $1, \dots, |\mathcal{S}|$ ;

**25** | |  $\theta \leftarrow \theta - \frac{n_t}{n} \cdot \eta_{k-1,j} \sum_{i \in \mathcal{I}} q(z^{(i)} = m) \cdot \nabla_{\theta} l(h_{\theta}(\mathbf{x}^{(i)}), y^{(i)})$ ;

**26** | **end**

**27** | **return**  $\theta$ ;

---

Assumption 15 expresses the fact that the sequence of mixing matrices, on average and every  $\tau$  communication rounds, brings the values in the columns of  $\Xi$  closer to their row-wise average (thereby mixing the clients' updates over time). For instance, the assumption is satisfied if the communication graph is strongly connected every  $\tau$  rounds, i.e., the graph  $([T], \mathcal{E})$ , where the edge  $(i, j)$  belongs to the graph if  $w_{i,j}^h > 0$  for some  $h \in \{k+1, \dots, k+\tau\}$  is connected.  $\text{D-FedEM}$  converges to a stationary point of  $f$  (proof in Appendix E.4).

**Theorem 3.5.4.** *Under Assumptions 8–15, when clients use SGD as local solver with learning rate  $\eta = \frac{a_0}{\sqrt{K}}$ ,  $\text{D-FedEM}$ 's iterates satisfy the following inequalities after a large enough number of communication rounds  $K$ :*

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \left\| \nabla_{\Theta} f(\bar{\Theta}^k, \Pi^k) \right\|_F^2 \leq \mathcal{O} \left( \frac{1}{\sqrt{K}} \right), \quad \frac{1}{K} \sum_{k=1}^K \sum_{t=1}^T \frac{n_t}{n} \mathcal{KL}(\pi_t^k, \pi_t^{k-1}) \leq \mathcal{O} \left( \frac{1}{K} \right), \quad (3.17)$$

where  $\bar{\Theta}^k = [\Theta_1^k, \dots, \Theta_T^k] \cdot \frac{\mathbf{1}\mathbf{1}^\top}{T}$ . Moreover, individual estimates  $(\Theta_t^k)_{1 \leq t \leq T}$  converge to consensus, i.e., to  $\bar{\Theta}^k$ :

$$\min_{k \in [K]} \mathbb{E} \sum_{t=1}^T \left\| \Theta_t^k - \bar{\Theta}^k \right\|_F^2 \leq \mathcal{O} \left( \frac{1}{\sqrt{K}} \right). \quad (3.18)$$

### 3.5.4 Federated Surrogate Optimization

$\text{FedEM}$  and  $\text{D-FedEM}$  can be seen as particular instances of a more general framework—of potential interest for other applications—that we call *federated surrogate optimization*.

The standard majorization-minimization principle [LHY00] iteratively minimizes, at each iteration  $k$ , a surrogate function  $g^k$  majorizing the objective function  $f$ . The work [Mai13] studied this approach when each  $g^k$  is a first-order surrogate of  $f$  (the formal definition from [Mai13] is given by Definition 3.5.1).

Our novel federated surrogate optimization framework considers that the objective function  $f$  is a weighted sum  $f = \sum_{t=1}^T \omega_t f_t$  of  $T$  functions and iteratively minimizes  $f$  in a distributed fashion using *partial* first-order surrogates  $g_t^k$  for each function  $f_t$ . “Partial” refers to the fact that  $g_t^k$  is not required to be a first order surrogate wrt the whole set of parameters, as defined formally below.

**Definition 1** (Partial first-order surrogate). *A function  $g(\mathbf{u}, \mathbf{v}) : \mathbb{R}^{d_u} \times \mathcal{V} \rightarrow \mathbb{R}$  is a partial-first-order surrogate of  $f(\mathbf{u}, \mathbf{v})$  wrt  $\mathbf{u}$  near  $(\mathbf{u}_0, \mathbf{v}_0) \in \mathbb{R}^{d_u} \times \mathcal{V}$  when the following conditions are satisfied:*

1.  $g(\mathbf{u}, \mathbf{v}) \geq f(\mathbf{u}, \mathbf{v})$  for all  $\mathbf{u} \in \mathbb{R}^{d_u}$  and  $\mathbf{v} \in \mathcal{V}$ ;
2.  $r(\mathbf{u}, \mathbf{v}) \triangleq g(\mathbf{u}, \mathbf{v}) - f(\mathbf{u}, \mathbf{v})$  is differentiable and  $L$ -smooth with respect to  $\mathbf{u}$ . Moreover, we have  $r(\mathbf{u}_0, \mathbf{v}_0) = 0$  and  $\nabla_{\mathbf{u}} r(\mathbf{u}_0, \mathbf{v}_0) = 0$ .
3.  $g(\mathbf{u}, \mathbf{v}_0) - g(\mathbf{u}, \mathbf{v}) = d_{\mathcal{V}}(\mathbf{v}_0, \mathbf{v})$  for all  $\mathbf{u} \in \mathbb{R}^{d_u}$  and  $\mathbf{v} \in \arg \min_{\mathbf{v}' \in \mathcal{V}} g(\mathbf{u}, \mathbf{v}')$ , where  $d_{\mathcal{V}}$  is non-negative and  $d_{\mathcal{V}}(\mathbf{v}, \mathbf{v}') = 0 \iff \mathbf{v} = \mathbf{v}'$ .

Under the assumption that each client  $t$  can compute a partial first-order surrogate of  $f_t$ , we propose algorithms for federated surrogate optimization in both the client-server setting (Algorithm 10) and the fully decentralized one (Algorithm 11) and prove their convergence under mild conditions (Appendix E.3 and E.4).  $\text{FedEM}$  and  $\text{D-FedEM}$  can be seen as particular instances of these algorithms and Theorem 3.5.3 and Theorem 3.5.4 follow from the more general convergence results for federated surrogate optimization.



### 3.5.4.1 Reminder on Basic (Centralized) Surrogate Optimization

In this appendix, we recall the (centralized) *first-order surrogate optimization* framework introduced in [Mai13]. In this framework, given a continuous function  $f : \mathbb{R}^d \mapsto \mathbb{R}$ , we are interested in solving

$$\min_{\theta \in \mathbb{R}^d} f(\theta)$$

using the majoration-minimization scheme presented in Alg. 9.

---

#### Algorithm 9: Basic Surrogate Optimization

---

**Input** :  $\theta^0 \in \mathbb{R}^d$ ; number of iterations  $K$ ;  
**Output** :  $\theta^K$   
**1 for iterations**  $k = 1, \dots, K$  **do**  
**2** | Compute  $g^k$ , a surrogate function of  $f$  near  $\theta^{k-1}$ ;  
**3** | Update solution:  $\theta^k \in \arg \min_{\theta} g^k(\theta)$ ;  
**4 end**

---

This procedure relies on surrogate functions, that approximate well the objective function in a neighborhood of a point. Reference [Mai13] focuses on *first-order surrogate functions* defined below.

**Definition 3.5.1** (First-Order Surrogate [Mai13]). A function  $g : \mathbb{R}^d \mapsto \mathbb{R}$  is a first order surrogate of  $f$  near  $\theta^k \in \mathbb{R}^d$  when the following is satisfied:

- **Majorization:** we have  $g(\theta') \geq f(\theta')$  for all  $\theta' \in \arg \min_{\theta \in \mathbb{R}^d} g(\theta)$ . When the more general condition  $g \geq f$  holds, we say that  $g$  is a **majorant** function.
- **Smoothness:** the approximation error  $r \triangleq g - f$  is differentiable, and its gradient is  $L$ -Lipschitz. Moreover, we have  $r(\theta^k) = 0$  and  $\nabla r(\theta^k) = 0$ .

### 3.5.4.2 Novel Federated Version

Our novel federated surrogate optimization framework minimizes an objective function  $(\mathbf{u}, \mathbf{v}_{1:T}) \mapsto f(\mathbf{u}, \mathbf{v}_{1:T})$  that can be written as a weighted sum  $f(\mathbf{u}, \mathbf{v}_{1:T}) = \sum_{t=1}^T \omega_t f_t(\mathbf{u}, \mathbf{v}_t)$  of  $T$  functions. We suppose that each client  $t \in [T]$  can compute a partial first order surrogate of  $f_t$  (Definition 1).

Under the assumption that each client  $t$  can compute a partial first order surrogate of  $f_t$ , we propose algorithms for federated surrogate optimization in both the client-server setting (Alg. 10) and the fully decentralized one (Alg. 11). Both algorithms are iterative and distributed: at each iteration  $k > 0$ , client  $t \in [T]$  computes a partial first-order surrogate  $g_t^k$  of  $f_t$  near  $\{u^{k-1}, v_t^{k-1}\}$  (resp.  $\{u_t^{k-1}, v_t^{k-1}\}$ ) for federated surrogate optimization in Alg. 10 (resp. for fully decentralized surrogate optimization in Alg 11).

The convergence of those two algorithms requires the following standard assumptions. Each of them generalizes one of the Assumptions 11–14 for our EM algorithms.

**Assumption 11'.** *The objective function  $f$  is bounded below by  $f^* \in \mathbb{R}$ .*

**Assumption 12'.** (Smoothness) *For all  $t \in [T]$  and  $k > 0$ ,  $g_t^k$  is  $L$ -smooth wrt to  $\mathbf{u}$ .*

**Algorithm 10:** Federated Surrogate Optimization

---

**Input** :  $\mathbf{u}^0 \in \mathbb{R}^{d_u}$ ;  $\mathbf{V}^0 = (\mathbf{v}_t^0)_{1 \leq t \leq T} \in \mathcal{V}^T$ ; number of iterations  $K$ ; number of local steps  $J$

**Output** :  $\mathbf{u}^K$ ;  $\mathbf{v}_t^K$

- 1 **for** iterations  $k = 1, \dots, K$  **do**
- 2     **server broadcasts**  $\mathbf{u}^{k-1}$  **to the**  $T$  **clients ;**
- 3     **for** tasks  $t = 1, \dots, T$  **in parallel over**  $T$  **clients do**
- 4         Compute partial first-order surrogate function  $g_t^k$  of  $f_t$  near  $\{\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}\}$ ;
- 5          $\mathbf{v}_t^k \leftarrow \arg \min_{\mathbf{v} \in \mathcal{V}} g_t^k(\mathbf{u}^{k-1}, \mathbf{v})$ ;
- 6          $\mathbf{u}_t^k \leftarrow \text{LocalSolver}(J, \mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1}, g_t^k, \mathcal{S}_t)$ ;
- 7         **client**  $t$  **sends**  $\mathbf{u}_t^k$  **to the server ;**
- 8     **end**
- 9      $\mathbf{u}^k \leftarrow \sum_{t=1}^T \omega_t \cdot \mathbf{u}_t^k$ ;
- 10 **end**

- 11 **Function**  $\text{LocalSolver}(J, \mathbf{u}, \mathbf{v}, g, \mathcal{S})$  :
- 12     **for**  $j = 0, \dots, J - 1$  **do**
- 13         sample  $\xi^{k-1,j}$  from  $\mathcal{S}$ ;
- 14          $\mathbf{u} \leftarrow \mathbf{u} - \eta_{k-1,j} \cdot \nabla_{\mathbf{u}} g(\mathbf{u}, \mathbf{v}; \xi^{k-1,j})$ ;
- 15     **end**
- 16     **return**  $\Theta$ ;

---

**Assumption 13'**. (Unbiased gradients and bounded variance) Each client  $t \in [T]$  can sample a random batch  $\xi$  from  $\mathcal{S}_t$  and compute an unbiased estimator  $\nabla_{\mathbf{u}} g_t^k(\mathbf{u}, \mathbf{v}; \xi)$  of the local gradient with bounded variance, i.e.,  $\mathbb{E}_{\xi}[\nabla_{\mathbf{u}} g_t^k(\mathbf{u}, \mathbf{v}; \xi)] = \nabla_{\mathbf{u}} g_t^k(\mathbf{u}, \mathbf{v})$  and  $\mathbb{E}_{\xi} \|\nabla_{\mathbf{u}} g_t^k(\mathbf{u}, \mathbf{v}; \xi) - \nabla_{\mathbf{u}} g_t^k(\mathbf{u}, \mathbf{v})\|^2 \leq \sigma^2$ .

**Assumption 14'**. (Bounded dissimilarity) There exist  $\beta$  and  $G$  such that

$$\sum_{t=1}^T \omega_t \cdot \left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}, \mathbf{v}) \right\|^2 \leq G^2 + \beta^2 \left\| \sum_{t=1}^T \omega_t \cdot \nabla_{\mathbf{u}} g_t^k(\mathbf{u}, \mathbf{v}) \right\|^2.$$

Under these assumptions a parallel result to Thm. 3.5.3 holds for the client-server setting.

**Theorem 3.5.3'**. Under Assumptions 11'–14', when clients use SGD as local solver with learning rate  $\eta = \frac{\alpha_0}{\sqrt{K}}$ , after a large enough number of communication rounds  $K$ , the iterates of federated surrogate optimization (Alg. 10) satisfy:

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\|_F^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{K}}\right), \quad \frac{1}{K} \sum_{k=1}^K \Delta_{\mathbf{v}} f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \leq \mathcal{O}\left(\frac{1}{K^{3/4}}\right), \quad (3.19)$$

where the expectation is over the random batches samples, and  $\Delta_{\mathbf{v}} f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \triangleq f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - f(\mathbf{u}^k, \mathbf{v}_{1:T}^{k+1}) \geq 0$ .

In the fully decentralized setting, if in addition to Assumptions 11'–14', we suppose that Assumption 15 holds, a parallel result to Thm. 3.5.4 holds.

**Theorem 3.5.4'.** *Under Assumptions 11'–14' and Assumption 15, when clients use SGD as local solver with learning rate  $\eta = \frac{a_0}{\sqrt{K}}$ , after a large enough number of communication rounds  $K$ , the iterates of fully decentralized federated surrogate optimization (Alg. 11) satisfy:*

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^k, v_{1:T}^k) \right\|^2 = \mathcal{O}\left(\frac{1}{\sqrt{K}}\right), \quad \frac{1}{K} \sum_{k=1}^K \sum_{t=1}^T \omega_t \cdot d_{\mathcal{V}}(\mathbf{v}_t^k, \mathbf{v}_t^{k+1}) = \mathcal{O}\left(\frac{1}{K}\right), \quad (3.20)$$

where  $\bar{\mathbf{u}}^k = \frac{1}{T} \sum_{t=1}^T \mathbf{u}_t^k$ . Moreover, local estimates  $(\mathbf{u}_t^k)_{1 \leq t \leq T}$  converge to consensus, i.e., to  $\bar{\mathbf{u}}^k$ :

$$\frac{1}{K} \sum_{k=1}^K \sum_{t=1}^T \left\| \mathbf{u}_t^k - \bar{\mathbf{u}}^k \right\|^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{K}}\right).$$

The proofs of Theorem 3.5.3' and Theorem 3.5.4' are in Appendix E.3 and Appendix 0 E.4, respectively.

### 3.5.5 Distributed Surrogate Optimization with Black-Box Solver

In this section, we cover the scenario where the local SGD solver used in our algorithms (Alg. 10 and Alg. 11) is replaced by a (possibly non-iterative) black-box solver that is guaranteed to provide a *local inexact solution* of

$$\forall m \in [M], \text{ minimize}_{\theta \in \mathbb{R}^d} \sum_{i=1}^{n_t} q^k(z_t^i = m) \cdot l(h_{\theta}(\mathbf{x}_t^{(i)}), y_t^{(i)}), \quad (3.21)$$

with the following approximation guarantee.

**Assumption 16** (Local  $\alpha$ -approximate solution). *There exists  $0 < \alpha < 1$  such that for  $t \in [T]$ ,  $m \in [M]$  and  $k > 0$ ,*

$$\sum_{i=1}^{n_t} q^k(z_t^i = m) \cdot \left\{ l(h_{\theta_{m,t}^k}(\mathbf{x}_t^{(i)}), y_t^{(i)}) - l(h_{\theta_{m,t,*}^k}(\mathbf{x}_t^{(i)}), y_t^{(i)}) \right\} \leq \alpha \cdot \sum_{i=1}^{n_t} q^k(z_t^i = m) \cdot \left\{ l(h_{\theta_m^{k-1}}(\mathbf{x}_t^{(i)}), y_t^{(i)}) - l(h_{\theta_{m,t,*}^k}(\mathbf{x}_t^{(i)}), y_t^{(i)}) \right\}, \quad (3.22)$$

where  $\theta_{m,t,*}^k \in \arg \min_{\theta \in \mathbb{R}^d} \sum_{i=1}^{n_t} q^k(z_t^i = m) \cdot l(h_{\theta}(\mathbf{x}_t^{(i)}), y_t^{(i)})$ ,  $\theta_{m,t}^k$  is the output of the local solver at client  $t$  and  $\theta_m^{k-1}$  is its starting point (see Alg. 7).

We further assume strong convexity.

**Assumption 17.** *For  $t \in [T]$  and  $i \in [n_t]$ , we suppose that  $\theta \mapsto l(h_{\theta}(\mathbf{x}_t^{(i)}), y_t^{(i)})$  is  $\mu$ -strongly convex.*

**Algorithm 11:** Fully-Decentralized Federated Surrogate Optimization

---

**Input** :  $\mathbf{u}^0 \in \mathbb{R}^{d_u}$ ;  $\mathbf{V}^0 = (\mathbf{v}_t^0)_{1 \leq t \leq T} \in \mathcal{V}^T$ ; number of iterations  $K$ ; number of local step  $J$ ; mixing matrix distributions  $\mathcal{W}^k$  for  $k \in [K]$

**Output**:  $\mathbf{u}_t^K$  for  $t \in [T]$ ;  $\mathbf{v}_t^K$  for  $t \in [T]$

```

1 for iterations  $k = 1, \dots, K$  do
    // Select the communication topology and the
    // aggregation weights
2   Sample  $W^{k-1} \sim \mathcal{W}^{k-1}$ ;
3   for tasks  $t = 1, \dots, T$  in parallel over  $T$  clients do
4     compute partial first-order surrogate function  $g_t^k$  of  $f_t$  near  $\{\mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1}\}$ ;
5      $\mathbf{v}_t^k \leftarrow \arg \min_{\mathbf{v} \in \mathcal{V}} g_t^k(\mathbf{u}_t^{k-1}, \mathbf{v})$ ;
6      $\mathbf{u}_t^{k-\frac{1}{2}} \leftarrow \text{LocalSolver}(J, \mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1}, g_t^k, t)$ ;
7     Send  $\mathbf{u}_t^{k-\frac{1}{2}}$  to neighbors;
8     Receive  $\mathbf{u}_s^{k-\frac{1}{2}}$  from neighbors;
9      $\mathbf{u}_t^k \leftarrow \sum_{s=1}^T w_{ts}^{k-1} \times \mathbf{u}_s^{k-\frac{1}{2}}$ ;
10  end
11 end

12 Function LocalSolver( $J, \mathbf{u}, \mathbf{v}, g, \mathcal{S}, t$ ):
13   for  $j = 0, \dots, J-1$  do
14     sample  $\xi^{k-1,j}$  from  $\mathcal{S}$ ;
15      $\mathbf{u} \leftarrow \mathbf{u} - \omega_t \cdot \eta_{k-1,j} \nabla_{\mathbf{u}} g(\mathbf{u}, \mathbf{v}, \xi^{k-1,j})$ ;
16   end
17   return  $\mathbf{u}$ ;
```

---

Assumption 16 is equivalent to the  $\gamma$ -inexact solution used in [Li+20a] (Lemma. E.19), when local functions  $(\Phi_t)_{1 \leq t \leq T}$  are assumed to be convex. We also need to have  $G^2 = 0$  in Assumption 14 as in [Li+20b, Definition 3], in order to ensure the convergence of Alg. 7 and Alg. 8 to a stationary point of  $f$ , as shown by [Wan+20b, Thm. 2].\*

**Theorem 3.5.5.** *Suppose that Assumptions 8–14, 16 and 17 hold with  $G^2 = 0$  and  $\alpha < \frac{1}{\beta^2 \kappa^4}$ , then the updates of federated surrogate optimization converge to a stationary point of  $f$ , i.e.,*

$$\lim_{k \rightarrow +\infty} \left\| \nabla_{\Theta} f(\Theta^k, \Pi^k) \right\|_F^2 = 0, \quad (3.23)$$

and

$$\lim_{k \rightarrow +\infty} \sum_{t=1}^T \frac{n_t}{n} \mathcal{KL}(\pi_t^k, \pi_t^{k-1}) = 0. \quad (3.24)$$

---

\*As shown by [Wan+20b, Thm. 2], the convergence is guaranteed in two scenarios: 1)  $G^2 = 0$ , 2) All clients use the same number of local steps using the same local solver. Note that we allow each client to use an arbitrary approximate local solver.

We provide the analysis for the general case of federated surrogate optimization (Algorithm 10) before showing that FedEM (Algorithm 7) is a particular case.

We suppose that, at iteration  $k > 0$ , the partial first-order surrogate functions  $g_t^k$ ,  $t \in [T]$  used in Alg. 10 verifies, in addition to Assumptions 11'–14', the following assumptions that generalize Assumptions 16 and 17,

**Assumption 16'** (Local  $\alpha$ -inexact solution). *There exists  $0 < \alpha < 1$  such that for  $t \in [T]$  and  $k > 0$ ,*

$$\forall \mathbf{v} \in \mathcal{V}, g_t^k(\mathbf{u}_t^k, \mathbf{v}) - g_t^k(\mathbf{u}_{t,*}^k, \mathbf{v}) \leq \alpha \cdot \left\{ g_t^k(\mathbf{u}^{k-1}, \mathbf{v}) - g_t^k(\mathbf{u}_{t,*}^k, \mathbf{v}) \right\}, \quad (3.25)$$

where  $\mathbf{u}_{t,*}^k \in \arg \min_{\mathbf{u} \in \mathbb{R}^{d_u}} g_t^k(\mathbf{u}, \mathbf{v}_t^k)$ .

**Assumption 17'**. *For  $t \in [T]$  and  $k > 0$ ,  $g_t^k$  is  $\mu$ -strongly convex in  $\mathbf{u}$ .*

Under these assumptions a parallel result to Thm. 3.5.5 holds.

**Theorem 3.5.5'**. *Suppose that Assumptions 11'–14', Assumptions 16' and 17' hold with  $G^2 = 0$  and  $\alpha < \frac{1}{\beta^2 \kappa^4}$ , then the updates of federated surrogate optimization converges to a stationary point of  $f$ , i.e.,*

$$\lim_{k \rightarrow +\infty} \left\| \nabla_{\mathbf{u}} f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\|^2 = 0, \quad (3.26)$$

and

$$\lim_{k \rightarrow +\infty} \sum_{t=1}^T \omega_t \cdot d_{\mathcal{V}}(\mathbf{v}_t^k, \mathbf{v}_t^{k-1}) = 0. \quad (3.27)$$

## 3.5.6 Numerical Experiments

### 3.5.6.1 Datasets and Models

In this section we provide detailed description of the datasets and models used in our experiments. We used a synthetic dataset, verifying Assumptions 8-10, and five "real" datasets (CIFAR-10/CIFAR-100 [Kri09], sub part of EMNIST [Coh+17], sub part of FEMNIST [Cal+19; McM+17] and Shakespeare [Cal+19; McM+17]) from which, two (FEMNIST and Shakespeare) has natural client partitioning. Below, we give a detailed description of the datasets and the models / tasks considered for each of them.

**CIFAR-10 / CIFAR-100** CIFAR-10 and CIFAR-100 are labeled subsets of the 80 million tiny images dataset. They both share the same 60,000 input images. CIFAR-100 has a finer labeling, with 100 unique labels, in comparison to CIFAR-10, having 10 unique label. We used Dirichlet allocation [Wan+20a], with parameter  $\alpha = 0.4$  to partition CIFAR-10 among 80 clients. We used Pachinko allocation [Red+21] with parameters  $\alpha = 0.4$  and  $\beta = 10$  to partition CIFAR-100 on 100 clients. For both of them we train MobileNet-v2 [San+18] architecture with an additional linear layer. We used TorchVision [MR10] implementation of MobileNet-v2.

**EMNIST** EMNIST (Extended MNIST) is a 62-class image classification dataset, extending the classic MNIST dataset. In our experiments, we consider 10% of the EMNIST dataset, that we partition using Dirichlet allocation of parameter  $\alpha = 0.4$  over 100 clients. We train the same convolutional network as in [Red+21]. The network has two convolutional layers (with  $3 \times 3$  kernels), max pooling, and dropout, followed by a 128 unit dense layer.

Table 3.1: Average computation time and used GPU for each dataset.

Dataset	GPU	Simulation time
Shakespeare	Quadro RTX 8000	4h42min
FEMNIST	Quadro RTX 8000	1h14min
EMNIST	GeForce GTX 1080 Ti	46min
CIFAR10	GeForce GTX 1080 Ti	2h37min
CIFAR100	GeForce GTX 1080 Ti	3h9min
Synthetic	GeForce GTX 1080 Ti	20min

**FEMNIST** FEMNIST (Federated Extended MNIST) is a 62-class image classification dataset built by partitioning the data of Extended MNIST based on the writer of the digits/characters. In our experiments, we used a subset with 15% of the total number of writers in FEMNIST. We train the same convolutional network as in [Red+21]. The network has two convolutional layers (with  $3 \times 3$  kernels), max pooling, and dropout, followed by a 128 unit dense layer.

**Shakespeare** This dataset is built from The Complete Works of William Shakespeare and is partitioned by the speaking roles [McM+17]. In our experiments, we discarded roles with less than two sentences. We consider character-level based language modeling on this dataset. The model takes as input a sequence of 200 English characters and predicts the next character. The model embeds the 80 characters into a learnable 8-dimensional embedding space, and uses two stacked-LSTM layers with 256 hidden units, followed by a densely-connected layer. We also normalized each character by its frequency of appearance.

### 3.5.6.2 Implementation Details

**Machines** We ran the experiments on a CPU/GPU cluster, with different GPUs available (e.g., Nvidia Tesla V100, GeForce GTX 1080 Ti, Titan X, Quadro RTX 6000, and Quadro RTX 8000). Most experiments with CIFAR10/CIFAR-100 and EMNIST were run on GeForce GTX 1080 Ti cards, while most experiments with Shakespeare and FEMNIST were run on the Quadro RTX 8000 cards. For each dataset, we ran around 30 experiments (not counting the development/debugging time). Table 3.1 gives the average amount of time needed to run one simulation for each dataset. The time needed per simulation was extremely long for Shakespeare dataset, because we used a batch size of 128. We remarked that increasing the batch size beyond 128 caused the model to converge to poor local minima, where the model keeps predicting a white space as next character.

**Libraries** We used PyTorch [Pas+19] to build and train our models. We also used Torchvision [MR10] implementation of MobileNet-v2 [San+18], and for image datasets preprocessing. We used LEAF [Cal+19] to build FEMNIST dataset and the federated version of Shakespeare dataset.

**Hyperparameters** For each method and each task, the learning rate was set via grid search on the set  $\{10^{-0.5}, 10^{-1}, 10^{-1.5}, 10^{-2}, 10^{-2.5}, 10^{-3}\}$ . FedProx and pFedMe’s penalization parameter  $\mu$  was tuned via grid search on  $\{10^1, 10^0, 10^{-1}, 10^{-2}, 10^{-3}\}$ . For clustered FL, we used the same

Table 3.2: Test accuracy: average across clients / bottom decile.

Dataset	Local	FedAvg	FedProx	FedAvg+	Clustered FL	pFedMe	FedEM (Ours)
FEMNIST	71.0 / 57.5	78.6 / 63.9	78.9 / 64.0	75.3 / 53.0	73.5 / 55.1	74.9 / 57.6	<b>79.9 / 64.8</b>
EMNIST	71.9 / 64.3	82.6 / 75.0	83.0 / 75.4	83.1 / 75.8	82.7 / 75.0	83.3 / 76.4	<b>83.5 / 76.6</b>
CIFAR10	70.2 / 48.7	78.2 / 72.4	78.0 / 70.8	82.3 / 70.6	78.6 / 71.2	81.7 / 73.6	<b>84.3 / 78.1</b>
CIFAR100	31.5 / 19.9	40.9 / 33.2	41.0 / 33.2	39.0 / 28.3	41.5 / 34.1	41.8 / 32.5	<b>44.1 / 35.0</b>
Shakespeare	32.0 / 16.6	<b>46.7</b> / 42.8	45.7 / 41.9	40.0 / 25.5	46.6 / 42.7	41.2 / 36.8	<b>46.7 / 43.0</b>
Synthetic	65.7 / 58.4	68.2 / 58.9	68.2 / 59.0	68.9 / 60.2	69.1 / 59.0	69.2 / 61.2	<b>74.7 / 66.7</b>

values of tolerance as the ones used in its official implementation [SMS20]. We found tuning  $\text{tol}_1$  and  $\text{tol}_2$  particularly hard: no empirical rule is provided in [SMS20], and the few random settings we tried did not show any improvement in comparison to the default ones.

### 3.5.6.3 Main Results

**Other FL approaches.** We compared our algorithms with global models trained with FedAvg [McM+17] and FedProx [Li+20b] as well as different personalization approaches: a personalized model trained only on the local dataset, FedAvg with local tuning (FedAvg+) [Jia+23], clustered FL [SMS20] and pFedMe [TTN20]. For each method and each task, the learning rate and the other hyper-parameters were tuned via grid search (details in App. 3.5.6.2). FedAvg+ updated the local model through a single pass on the local dataset. Unless otherwise stated, the number of components considered by FedEM was  $M = 3$ , training occurred over 80 communication rounds for Shakespeare and 200 rounds for all other datasets. At each round, clients train for one epoch. Results for D-FedEM are in Appendix E.9.

**Average performance of personalized models.** The performance of each personalized model (which is the same for all clients in the case of FedAvg and FedProx) is evaluated on the local test dataset (unseen at training). Table 3.2 shows the average weighted accuracy with weights proportional to local dataset sizes. We observe that FedEM obtains the best performance across all datasets.

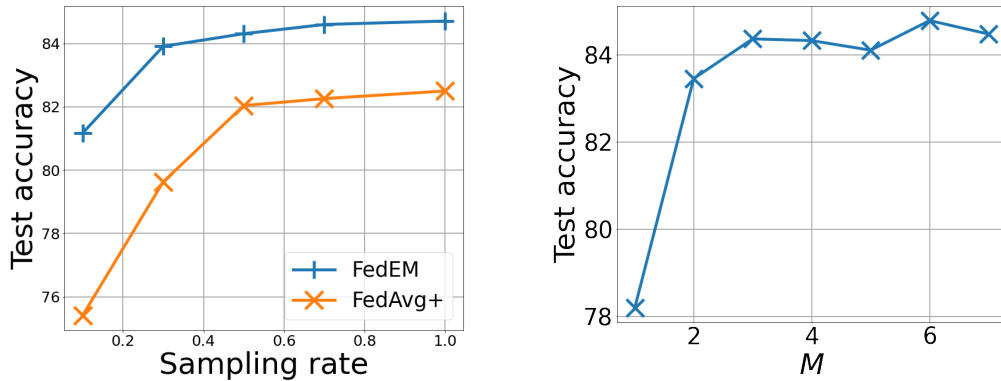
**Fairness across clients.** FedEM’s improvement in terms of average accuracy could be the result of learning particularly good models for some clients at the expense of bad models for other clients. Table 3.2 shows the bottom decile of the accuracy of local models, i.e., the  $(T/10)$ -th worst accuracy (the minimum accuracy is particularly noisy, notably because some local test datasets are very small). Even clients with the worst personalized models are still better off when FedEM is used for training.

**Clients sampling.** In cross-device federated learning, only a subset of clients may be available at each round. We ran CIFAR10 experiments with different levels of participation: at each round a given fraction of all clients were sampled uniformly without replacement. We restrict the comparison to FedEM and FedAvg+, as 1) FedAvg+ performed better than FedProx and FedAvg in the previous CIFAR10 experiments, 2) it is not clear how to extend pFedMe and clustered FL to handle client sampling. Results in Fig. 3.1 (left) show that FedEM is more robust to low clients’ participation levels.

**Generalization to unseen clients.** As discussed in Section 3.5.3.2, FedEM allows new clients arriving after the distributed training to easily learn their personalized models. With the exception of FedAvg+, it is not clear if and how the other personalized FL algorithms can tackle the same goal. In order to evaluate the quality of new clients’ personalized models, we performed an experiment

Table 3.3: Average test accuracy across clients unseen at training (train accuracy in parenthesis).

Dataset	FedAvg	FedAvg+	FedEM (Ours)
FEMNIST	78.3 (80.9)	74.2 (84.2)	<b>79.1</b> (81.5)
EMNIST	83.4 (82.7)	83.7 (92.9)	<b>84.0</b> (83.3)
CIFAR10	77.3 (77.5)	80.4 (80.5)	<b>85.9</b> (90.7)
CIFAR100	41.1 (42.1)	36.5 (55.3)	<b>47.5</b> (46.6)
Shakespeare	<b>46.7</b> (47.1)	40.2 (93.0)	<b>46.7</b> (46.6)
Synthetic	68.6 (70.0)	69.1 (72.1)	<b>73.0</b> (74.1)

Figure 3.1: Effect of client sampling rate (left) and FedEM number of mixture components  $M$  (right) on the test accuracy for CIFAR10 [Kri09].

where only 80% of the clients (“old” clients) participate to the training. The remaining 20% join the system in a second phase and use their local training datasets to learn their personalized weights. Table 3.3 shows that FedEM allows new clients to learn a personalized model at least as good as FedAvg’s global one and always better than FedAvg+’s one. Unexpectedly, new clients achieve sometimes a significantly higher test accuracy than old clients (e.g., 47.5% against 44.1% on CIFAR100). Our investigation (App. E.11) suggests that, by selecting their mixture weights on local datasets that were not used to train the components, new clients can compensate for potential overfitting in the initial training phase.

**Effect of  $M$ .** A limitation of FedEM is that each client needs to update and transmit  $M$  components at each round, requiring roughly  $M$  times more computation and  $M$  times larger messages. Nevertheless, the number of components to consider in practice is quite limited. We used  $M = 3$  in our previous experiments, and Fig. 3.1 (right) shows that larger values do not yield much improvement and  $M = 2$  already provides a significant level of personalization. In all experiments above, the number of communication rounds allowed all approaches to converge. As a consequence, even if other methods trained over  $M = 3$  times more rounds—in order to have as much computation and communication as FedEM—the conclusions would not change. As a final experiment, we considered a time-constrained setting, where FedEM is limited to run one third ( $= 1/M$ ) of the rounds (Table 6 in App. E.13). Even if FedEM does not reach its maximum accuracy, it still outperforms the other methods on 3 datasets.



### 3.5.7 Conclusion

In this section, we proposed a novel federated MTL approach based on the flexible assumption that local data distributions are mixtures of underlying distributions. Our EM-like algorithms allow clients to jointly learn shared component models and personalized mixture weights in client-server and fully decentralized settings. We proved convergence guarantees for our algorithms through a general federated surrogate optimization framework which can be used to analyze other FL formulations. Extensive empirical evaluation shows that our approach learns models with higher accuracy and fairness than state-of-the-art FL algorithms, even for clients not present at training time.

In future work, we aim to reduce the local computation and communication of our algorithms. Aside from standard compression schemes [Had+21], a promising direction is to limit the number of component models that a client updates/transmits at each step. This could be done in an adaptive manner based on the client’s current mixture weights. A second interesting research direction is to study personalized FL approaches under privacy constraints (quite unexplored until now with the notable exception of [Bel+18]). Some features of our algorithms may be beneficial for privacy (e.g., the fact that personalized weights are kept locally and that all users contribute to all shared models). We hope to design differentially private versions of our algorithms and characterize their privacy-utility trade-offs.

Since introducing the mixture assumption and the `FedEM` algorithm in [Mar+21b], several personalization approaches have emerged, expanding on this paradigm. Notable contributions to this growing field include soft-clustering (`FedSoft`) [RJ22], federated Gaussian mixture models (`FedGMM`) [Wu+23], federated modular networks (`FedMN`) [Wan+22a], and personalized federated learning with the right collaborators (`FedRiCo`) [Sui+22]. Beyond its original context, our `FedEM` approach has found applications in characterizing internal evasion attacks within federated learning [Kim+23]. Its unique capability to measure data distribution similarity among clients has been instrumental in this regard. Moreover, mixture models have been used to address the challenging problem of diverse distribution shifts in federated learning [GTL23; Jot+23; Zhu+22]. In Chapter 4, we explore online federated learning under the assumption that clients’ data distributions consist of mixtures of a finite number of unknown underlying distributions with varying mixing weights.

Apart from its theoretical impact and its adoption as a foundational element in various personalization approaches and distribution shift mitigation strategies, the `FedEM`’s accompanying code (accessible at <https://github.com/omarfoq/FedEM>) has provided a versatile framework for simulating federated learning. This resource has facilitated numerous researchers in experimenting with new algorithms, as seen in recent works such as [TH23; Sui+22].

## 3.6 Personalized Federated Learning through Local Memorization

In this section, we exploit the ability of deep neural networks to extract high quality vectorial representations (embeddings) from non-tabular data, e.g., images and text, to propose a personalization mechanism based on local memorization. Personalization is obtained by interpolating a collectively trained global model with a local  $k$ -nearest neighbors (kNN) model based on the shared representation provided by the global model.

Motivated by the recent success of memorization techniques based on nearest neighbors for natural language processing, [Kha+19; Kha+21], computer vision [PM18; Orh18], and few-shot classification [SSZ17; Wan+19b], we propose `kNN-Per`, a personalized FL algorithm based on local mem-

orization. `kNN-Per` combines a global model trained collectively (e.g., via `FedAvg` [McM+17]) with a kNN model on a client’s local datastore. The global model also provides the shared representation used by the local kNN.

`kNN-Per` offers a simple and effective way to address statistical heterogeneity even in a dynamic environment where client’s data distributions change after training. It is indeed sufficient to update the local datastore with new data without the need to retrain the global model. Moreover, each client can independently tune the local kNN to its storage and computing capabilities, partially relieving the most powerful clients from the need to align their model to the weakest ones. Finally, `kNN-Per` has a limited leakage of private information, as personalization only occurs once communication exchanges have ended, and, if needed, it can be easily combined with differential privacy techniques.

### 3.6.1 kNN-Per Algorithm

In this section, we suppose that all tasks have access to a global discriminative model  $h_S$  minimizing the empirical risk on the aggregated dataset  $\mathcal{S} \triangleq \bigcup_{t=1}^T \mathcal{S}_t$ , i.e.,

$$h_S \in \arg \min_{h \in \mathcal{H}} \mathcal{L}_S(h), \quad (3.28)$$

where  $\mathcal{L}_S(h) \triangleq \sum_{t=1}^T \frac{n_t}{n} \cdot \frac{1}{n_t} \sum_{i=1}^{n_t} l(h(\mathbf{x}_t^{(i)}), y_t^{(i)})$ , and  $n = \sum_{t=1}^T n_t$ . Typically  $h_S$  is a feed-forward neural network, jointly trained by the clients using a standard FL algorithm like `FedAvg`.

We also suppose that the global model can be used to compute a fixed-length representation for any input  $\mathbf{x} \in \mathcal{X}$ , and we use  $\phi_{h_S} : \mathcal{X} \mapsto \mathbb{R}^p$  to denote the function that maps the input  $\mathbf{x} \in \mathcal{X}$  to its representation.

The intermediate representation can be, for example, the output of the last convolutional layer in the case of CNNs, or the last hidden state in the case of recurrent networks or the output of an arbitrary self-attention layer in the case of transformers. Note that an alternative possible approach would be to separately learn an independent shared representation, e.g., using metric learning techniques [BHS15].

Our method (see 12) involves augmenting the global model with a local nearest neighbors’ retrieval mechanism at each client. The proposed method does not need any additional training; it only requires a single forward pass over the local dataset  $\mathcal{S}_t$ ,  $t \in [T]$ : client  $m$  computes the intermediate representation  $\phi_{h_S}(\mathbf{x})$  for each sample  $(\mathbf{x}, y) \in \mathcal{S}_t$ . The corresponding representation-label pairs are stored in a local key-value datastore  $(\mathcal{K}_t, \mathcal{V}_t)$  that is queried during inference. Formally,

$$(\mathcal{K}_t, \mathcal{V}_t) = \left\{ \left( \phi_{h_S}(\mathbf{x}_t^{(i)}), y_t^{(i)} \right), \forall (\mathbf{x}_t^{(i)}, y_t^{(i)}) \in \mathcal{S}_t \right\}. \quad (3.29)$$

At inference time, given input data  $\mathbf{x} \in \mathcal{X}$ , client  $t \in [T]$  computes  $h_S(\mathbf{x})$  and the intermediate representation  $\phi_{h_S}(\mathbf{x})$ . Then, it queries its local datastore  $(\mathcal{K}_t, \mathcal{V}_t)$  with  $\phi_{h_S}(\mathbf{x})$  to retrieve its  $k$ -nearest neighbors  $\mathcal{N}_t^{(k)}(\mathbf{x})$  according to a distance  $d(\cdot, \cdot)$ :

$$\mathcal{N}_t^{(k)}(\mathbf{x}) = \left( \phi_{h_S}(\mathbf{x}_{\pi_t^{(i)}(\mathbf{x})}), y_{\pi_t^{(i)}(\mathbf{x})} \right)_{1 \leq i \leq k}, \quad (3.30)$$

where  $\pi_t^{(1)}(\mathbf{x}), \dots, \pi_t^{(n_t)}(\mathbf{x})$  is a permutation of  $[n_t]$  corresponding to the distance of the samples

**Algorithm 12:** `kNN-Per` (Typical usage)

- 
- 1 Learn global model using available clients with `FedAvg`;
  - 2 **for** each client  $t \in [T]$  (in parallel) **do**
  - 3     Build datastore using  $\mathcal{S}_t$ ;
  - 4     At inference on  $\mathbf{x} \in \mathcal{X}$ , return  $h_{t,\lambda_t}(\mathbf{x})$  given by (3.33) ;
  - 5 **end**
- 

in  $\mathcal{S}_t$  from  $\mathbf{x}$ , i.e., for  $i \in [n_t - 1]$ ,

$$d(\phi_{h_S}(\mathbf{x}), \phi_{h_S}(\mathbf{x}_{\pi_t^{(i)}(\mathbf{x})})) \leq d(\phi_{h_S}(\mathbf{x}), \phi_{h_S}(\mathbf{x}_{\pi_t^{(i+1)}(\mathbf{x})})). \quad (3.31)$$

Then, the client computes a local hypothesis  $h_{\mathcal{S}_t}^{(k)}$  which estimates the conditional probability  $\mathcal{D}_t(y|\mathbf{x})$  using a kNN method, e.g., with a Gaussian kernel:

$$\left[ h_{\mathcal{S}_t}^{(k)}(\mathbf{x}) \right]_y \propto \sum_{i=1}^k \mathbb{1}_{\left\{ y=y_{\pi_t^{(i)}(\mathbf{x})} \right\}} \times \exp \left\{ -d \left( \phi_{h_S}(\mathbf{x}), \phi_{h_S}(\mathbf{x}_{\pi_t^{(i)}(\mathbf{x})}) \right) \right\}. \quad (3.32)$$

The final decision rule (hypothesis) at client  $t \in [T]$  ( $h_{t,\lambda_t}$ ) is obtained interpolating the nearest neighbor distribution  $h_{\mathcal{S}_t}^{(k)}$  with the distribution obtained from the global model  $h_S$  using a hyper-parameter  $\lambda_t \in (0, 1)$  to produce the final prediction, i.e.,

$$h_{t,\lambda_t}(\mathbf{x}) \triangleq \lambda_t \cdot h_{\mathcal{S}_t}^{(k)}(\mathbf{x}) + (1 - \lambda_t) \cdot h_S(\mathbf{x}). \quad (3.33)$$

As  $h_{t,\lambda_t}$  may not belong to  $\mathcal{H}$ , we are considering an *improper learning* setting. The parameter  $\lambda_t$  is tuned at client  $t$  through a local validation dataset or cross-validation as in [CBB21; Man+20; Zha+21; Li+21]. Clients could also use different values  $k_t$  and different distance metrics  $d_t(\cdot)$ , but, in what follows, we consider them equal across clients. Also our experiments in Section 3.6.3 show that  $k$  and  $d(\cdot)$  do not require careful tuning.

### 3.6.2 Generalization Bound

In this section we provide a generalization bound associated with the proposed approach in the case of binary classification, namely  $\mathcal{Y} = \{0, 1\}$ , when only one neighbour is used for kNN estimation, i.e.,  $k = 1$ , and  $d(\cdot, \cdot)$  is the Euclidean distance. For client  $t \in [T]$ , we denote by  $\eta_t : \mathcal{X} \mapsto \mathbb{R}$  the true conditional probability of label 1, that is

$$\eta_t(\mathbf{x}) = \mathcal{D}_t(y = 1|\mathbf{x}). \quad (3.34)$$

Our result holds under the following assumptions:

**Assumption 18** (Bounded representation).  $\phi_{h_S} : \mathcal{X} \mapsto [0, 1]^p$ .

**Assumption 19** (Bounded loss).  $l : \Delta^{|\mathcal{Y}|} \times \mathcal{Y} \mapsto [0, 1]$ . Moreover, for  $y, y' \in \{0, 1\}$ ,  $l(\mathbf{e}_y, y') = \mathbb{1}_{y \neq y'}$ , where  $\mathbf{e}_y \in \Delta^{|\mathcal{Y}|}$  is the vector having all entries equal to 0 except the entry on the  $y$ -th coordinate.

**Remark 4.** *Loss boundedness is a common assumption, e.g., [Man+20],[SB14, Ch. 4]. The second requirement is that the maximum loss is achieved when the model is fully confident about a prediction, but this is wrong. A simple transformation of common loss functions—e.g., exponentiating the logistic function—make them satisfy Assumption 19.*

**Assumption 20** (Loss convexity). *The loss function is convex on the first variable*

$$\begin{aligned} \forall y_1, y_2 \in \Delta^{|\mathcal{Y}|}, \forall y \in \mathcal{Y}, \forall \lambda_t \in [0, 1], \\ l(\lambda_t \cdot y_1 + (1 - \lambda_t) \cdot y_2, y) \leq \lambda_t \cdot l(y_1, y) + (1 - \lambda_t) \cdot l(y_2, y). \end{aligned} \quad (3.35)$$

**Remark 5.** *Assumption 20 holds for most loss functions used in supervised machine learning, including the mean squared error loss, the cross-entropy loss, and the hinge loss.*

**Assumption 21.** *There exist constants  $\gamma_1, \gamma_2 > 0$ , such that for any dataset  $\mathcal{S}$  drawn from  $\mathcal{X} \times \mathcal{Y}$  and any data points  $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ , we have*

$$|\eta_t(\mathbf{x}) - \eta_t(\mathbf{x}')| \leq d(\phi_{h_{\mathcal{S}}}(\mathbf{x}), \phi_{h_{\mathcal{S}}}(\mathbf{x}')) \times (\gamma_1 + \gamma_2 (\mathcal{L}_{\mathcal{D}_t}(h_{\mathcal{S}}) - \mathcal{L}_{\mathcal{D}_t}(h_t^*))), \quad (3.36)$$

where  $h_t^* \in \arg \min_{h \in \mathcal{H}} \mathcal{L}_{\mathcal{D}_t}(h)$ .

This assumption means that if two samples  $\mathbf{x}$  and  $\mathbf{x}'$  have close representations  $\phi_{h_{\mathcal{S}}}(\mathbf{x})$  and  $\phi_{h_{\mathcal{S}}}(\mathbf{x}')$ , then their labels are likely to be the same ( $|\eta_t(\mathbf{x}) - \eta_t(\mathbf{x}')|$  is small). This is all the more so, the better  $h_{\mathcal{S}}$  predictions are for distribution  $\mathcal{D}_t$ ,  $t \in [T]$  (the smaller  $\mathcal{L}_{\mathcal{D}_t}(h_{\mathcal{S}}) - \mathcal{L}_{\mathcal{D}_t}(h_t^*)$  is). Experimental results support Assumption 21 (see 3.4).

Our generalization bound depends, as usual, on the complexity of the hypothesis class  $\mathcal{H}$  (expressed by its VC-dimension,  $d_{\mathcal{H}}$ ) and on the size of the local and global datasets ( $n_t$  and  $n$ , respectively), but also on the distance between the local distribution  $\mathcal{D}_t$  and the average distribution  $\bar{\mathcal{D}} = \sum_{t=1}^T \frac{n_t}{n} \cdot \mathcal{D}_t$ , which is the one the global model  $h_{\mathcal{S}}$  is targeting (see (3.28)). The distance between two distributions  $\mathcal{D}$  and  $\mathcal{D}'$  associated to a hypothesis class  $\mathcal{H}$  can be quantified by the *label discrepancy* [Man+20]:

$$\text{disc}_{\mathcal{H}}(\mathcal{D}, \mathcal{D}') = \max_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{D}}(h) - \mathcal{L}_{\mathcal{D}'}(h)|. \quad (3.37)$$

**Theorem 3.6.1.** *Suppose that Assumptions 18–21 hold, and consider  $t \in [T]$  and  $\lambda_t \in (0, 1)$ , then there exist constants  $c_1, c_2, c_3, c_4$ , and  $c_5 \in \mathbb{R}$ , such that*

$$\begin{aligned} \mathbb{E}_{\mathcal{S} \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t}} [\mathcal{L}_{\mathcal{D}_t}(h_{t, \lambda_t})] &\leq (1 + \lambda_t) \cdot \mathcal{L}_{\mathcal{D}_t}(h_t^*) + c_1 (1 - \lambda_t) \cdot \text{disc}_{\mathcal{H}}(\bar{\mathcal{D}}, \mathcal{D}_t) \\ &+ c_2 \lambda_t \cdot \frac{\sqrt{p}}{p+1/\sqrt{n_t}} \cdot (\text{disc}_{\mathcal{H}}(\bar{\mathcal{D}}, \mathcal{D}_t) + 1) \\ &+ c_3 (1 - \lambda_t) \cdot \sqrt{\frac{d_{\mathcal{H}}}{n}} \cdot \sqrt{c_4 + \log\left(\frac{n}{d_{\mathcal{H}}}\right)} \\ &+ c_5 \lambda_t \cdot \sqrt{\frac{d_{\mathcal{H}}}{n}} \cdot \sqrt{c_4 + \log\left(\frac{n}{d_{\mathcal{H}}}\right)} \cdot \frac{\sqrt{p}}{p+1/\sqrt{n_t}}, \end{aligned} \quad (3.38)$$

where  $d_{\mathcal{H}}$  is the VC dimension of the hypothesis class  $\mathcal{H}$ ,  $n = \sum_{t=1}^M n_t$ ,  $\bar{\mathcal{D}} = \sum_{t=1}^T \frac{n_t}{n} \cdot \mathcal{D}_t$ ,  $p$  is the dimension of representations, and  $\text{disc}_{\mathcal{H}}$  is the label discrepancy associated to the hypothesis class  $\mathcal{H}$ .

Table 3.4: Datasets and models.

DATASET	TASK	CLIENTS	TOTAL SAMPLES	MODEL
FEMNIST	CHARACTER RECOGNITION	3,550	805,263	MOBILENET-V2
CIFAR-10	IMAGE CLASSIFICATION	200	60,000	MOBILENET-V2
CIFAR-100	IMAGE CLASSIFICATION	200	60,000	MOBILENET-V2
SHAKESPEARE	NEXT-CHARACTER PREDICTION	778	4,226,158	STACKED-LSTM

The proof of Theorem 3.6.1 is in Appendix F. Let us consider, for simplicity, the non-agnostic case, i.e.,  $\mathcal{L}_{\mathcal{D}_t}(h_t^*) = 0$ . We observe that, when clients only use the global model ( $\lambda_t = 0$ ), our generalization bound is analogous to the probabilistic bound in [Man+20, Eq. (2)]. In particular, if data is i.i.d. distributed across the clients ( $\text{disc}_{\mathcal{H}}(\bar{\mathcal{D}}, \mathcal{D}_t) = 0$ ), the difference between the expected losses of the learned model and the optimal one decreases with rate  $\tilde{O}\left(\sqrt{\frac{d_{\mathcal{H}}}{n}}\right)$ . Instead, when each client only uses the  $k$ NN model ( $\lambda_t = 1$ ),\* we recover the  $k$ NN generalization bound in [SB14, Thm 19.3].

The bound (3.38) leads to predict that client  $t$  should give a larger weight ( $\lambda_t > 1/2$ ) to its  $k$ NN model, when  $n_t$  exceeds a given threshold, even when local distributions are identical. The bound contributes then to explain why adding a memorization mechanism on top of a pretrained model can improve performance, as observed in [Kha+19] and [Kha+21]. While it is difficult to quantify the threshold analytically (also because the constants involved depend on  $\gamma_1$  and  $\gamma_2$  in Assumption 21), our experiments in Section 3.6.3 show that even clients with a few tens of samples weigh more the  $k$ NN model than the global one.

### 3.6.3 Numerical Experiments

We evaluate `kNN-PER` on four federated datasets spanning a wide range of machine learning tasks: language modeling (Shakespeare [Cal+19; McM+17]), image classification (CIFAR-10 and CIFAR-100 [Kri09]), handwritten character recognition (FEMNIST [Cal+19]). Unless otherwise said, `kNN-PER`'s global model  $h_{\mathcal{S}}$  is trained by all clients through `FedAvg`. Code is available at <https://github.com/omarfoq/knn-per>.

**Datasets.** For Shakespeare and FEMNIST datasets there is a natural way to partition data through clients (by character and by writer, respectively). We relied on common approaches in the literature to sample heterogeneous local datasets from CIFAR-10 and CIFAR-100. We created a federated version of CIFAR-10 by randomly partitioning the dataset among clients using a symmetric Dirichlet distribution, as done in [Wan+20a]. In particular, for each label  $y$  we sampled a vector  $p_y$  from a Dirichlet distribution of order  $M = 200$  and parameter  $\alpha = 0.3$  (unless otherwise specified) and allocated to client  $m$  a  $p_{y,m}$  fraction of all training instances of class  $y$ . The approach ensures that the number of data points and label distributions are unbalanced across clients. For CIFAR-100, we exploit the availability of “coarse” and “fine” label structure, to partition the dataset using Pachinko allocation method [LM06] as in [Red+21]. The method generates local datasets with heterogeneous distributions by combining a per-client Dirichlet distribution with parameter  $\alpha = 0.3$

\*Note that the  $k$ NN model still relies on the representation provided by the global model.

(unless otherwise specified) over the coarse labels and a per-coarse-label Dirichlet distribution with parameter  $\beta = 10$  over the corresponding fine labels. We also partitioned CIFAR-10 and CIFAR-100 in a different way following [Ach+21]: each client has only samples from two and ten classes for CIFAR-10 and CIFAR-100, respectively. We refer to the resulting datasets as CIFAR-10 (v2) and CIFAR-100 (v2). For FEMNIST and Shakespeare, we randomly split each local dataset into training (60%), validation (20%), and test (20%) sets. For CIFAR-10 and CIFAR-100, we maintained the original training/test data split and used 20% of the training dataset as validation dataset. Table 3.4 summarizes datasets, models and number of clients.

**Models and representations.** For CIFAR-100, CIFAR-10, and FEMNIST, we used MobileNet-v2 [San+18] as a base model with the output of the last hidden layer—a 1280-dimensional vector—as representation. For Shakespeare, the base model was a stacked LSTM model with two layers, each of them with 256 units; a 1024-dimensional representation was obtained by concatenating the hidden states and the cell states.

**Benchmarks.** We compared `kNN-Per` with locally trained models (with no collaboration across clients) and `FedAvg` [McM+17], as well as with one method for each of the personalization approaches described in Section 3.2, namely, `FedAvg+` [Jia+23],\* `ClusteredFL` [SMS20], `Ditto` [Li+21], `FedRep` [Col+21], `APFL` [DKM20], and `pFedGP` [Ach+21].† For each method, and each dataset, we tuned the learning rate via grid search on the values  $\{10^{-0.5}, 10^{-1}, 10^{-1.5}, 10^{-2}, 10^{-2.5}\}$ . `FedPer`’s learning rate for network heads’ training was separately tuned on the same grid. `Ditto`’s penalization parameter  $\lambda_t$  was selected among the values  $\{10^1, 10^0, 10^{-1}, 10^{-2}\}$  on a per-client basis. For `ClusteredFL`, we used the same values of tolerance specified in its official implementation [SMS20]. We found tuning `tol1` and `tol2` particularly hard: no empirical rule is provided in [SMS20], and the few random settings we tried did not show any improvement in comparison to the default ones. For `APFL`, the mixing parameter  $\alpha$  was tuned via grid search on the grid  $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ . For `pFedGP`, we used the same hyperparameters as in [Ach+21]. The parameter  $\lambda_t$  of `kNN-Per` was tuned for each client via grid search on the grid  $\{0.0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$ , and the number of neighbors was set to  $k = 10$ . Once the optimal hyperparameters’ values were selected, models were retrained on the concatenation of training and validation sets.

**Training details.** In all experiments with CIFAR-10 and CIFAR-100, training spanned 200 rounds with full clients’ participation at each round for all methods. The learning rate was reduced by a factor 10 at round 100 and then again at round 150. For Shakespeare, 10% of clients were sampled uniformly at random without replacement at each round, and we trained for 300 rounds with a constant learning rate. For FEMNIST, 5% of the clients participated at each round for a total 1000 rounds, with the learning rate dropping by a factor 10 at round 500 and 750. In all our experiments we employed the following aggregation scheme

$$\mathbf{w}_{k+1} = \sum_{t \notin \mathbb{S}_k} \frac{n_t}{n} \mathbf{w}_k + \sum_{t \in \mathbb{S}_k} \frac{n_t}{n} \mathbf{w}_k^t, \quad (3.39)$$

\*We also implemented the more sophisticated first-order MAML approach from [FMO20], but had worse performance than `FedAvg+`.

†We were able to run the official `pFedGP`’s code (<https://github.com/IdanAchituve/pFedGP>) only on datasets partitioned as in [Ach+21].

Table 3.5: Test accuracy: average across clients / bottom decile.

DATASET	LOCAL	FEDAVG	FEDAVG+	CLUSTEREDFL	DITTO	FEDREP	APFL	pFEDGP	kNN-PER (OURS)
FEMNIST	71.0 / 57.5	83.4 / 68.9	84.3 / 69.4	83.7 / 69.4	84.3 / 71.3	85.3 / 72.7	84.1 / 69.4	- / -	<b>88.2 / 78.8</b>
CIFAR-10	57.6 / 41.1	72.8 / 59.6	75.2 / 62.3	73.3 / 61.5	80.0 / 66.5	77.7 / 65.2	78.9 / 68.1	- / -	<b>83.0 / 71.4</b>
CIFAR-10 (v2)	82.4 / 71.3	67.9 / 60.1	85.0 / 79.6	79.9 / 72.3	86.3 / 80.6	89.1 / 85.3	82.6 / 76.4	88.9 / 84.1	<b>93.8 / 88.2</b>
CIFAR-100	31.5 / 19.8	47.4 / 36.0	51.4 / 41.1	47.2 / 36.2	52.0 / 41.4	53.2 / 41.7	51.7 / 41.1	- / -	<b>55.0 / 43.6</b>
CIFAR-100 (v2)	45.7 / 38.2	42.3 / 34.8	48.1 / 41.9	43.5 / 37.2	48.7 / 40.3	70.1 / 65.2	48.3 / 42.1	61.1 / 50.0	<b>74.6 / 67.3</b>
SHAKESPEARE	32.0 / 16.0	48.1 / 43.1	47.0 / 42.2	46.7 / 41.4	47.9 / 42.6	47.2 / 42.3	45.9 / 42.4	- / -	<b>51.4 / 45.4</b>

Table 3.6: Average test accuracy across clients unseen at training (train accuracy between parentheses).

Dataset	FedAvg	FedAvg+	ClusteredFL	Ditto	FedRep	APFL	pFedGP	kNN-Per (Ours)
FEMNIST	83.1 (83.3)	84.2 (88.5)	83.2 (86.0)	83.9 (86.9)	85.4 (88.9)	84.2 (85.5)	-	<b>88.1</b> (90.5)
CIFAR-10	72.9 (72.8)	75.3 (78.2)	73.9 (76.2)	79.7 (84.3)	76.4 (79.5)	79.2 (80.6)	-	<b>82.4</b> (87.1)
CIFAR-10 (v2)	67.5 (68.1)	85.1 (85.0)	79.6 (79.9)	85.9 (86.0)	89.0 (89.1)	82.3 (82.5)	89.0 (88.8)	<b>93.0</b> (93.1)
CIFAR-100	47.1 (47.5)	50.8 (53.4)	47.1 (48.2)	52.1 (57.3)	53.5 (58.2)	49.1 (52.7)	-	<b>56.1</b> (59.3)
CIFAR-100 (v2)	42.1 (42.2)	47.9 (48.1)	43.2 (43.4)	48.8 (48.5)	69.8 (70.0)	48.2 (48.4)	61.3 (61.0)	<b>74.3</b> (74.5)
Shakespeare	49.0 (48.3)	49.3 (48.1)	49.4 (46.7)	48.1 (49.2)	48.7 (47.8)	46.1 (52.7)	-	<b>50.7</b> (64.2)

where  $\mathbf{w}_k$ ,  $\mathbf{w}_k^t$ , and  $\mathbb{S}_k$  denote, respectively, the global model, the updated model at client  $t$ , and the set of clients participating to training at round  $k$ .

In all our experiments, local hypotheses follow Eq. (3.32) with  $d(\cdot)$  being the Euclidean distance. kNN retrieval relied on FAISS library [JDJ19].

**Average performance of personalized models.** The performance of each personalized model (which coincides with the global one in the case of FedAvg) is evaluated on the local test dataset (unseen at training). Table 3.5 shows the average weighted accuracy with weights proportional to local dataset sizes. kNN-Per consistently achieves the highest accuracy across all datasets. We observe that Local performs much worse than any other FL method as expected (e.g., 25 pp w.r.t. kNN-Per or 22 pp w.r.t. to Ditto on CIFAR-10). Local outperforms some other FL methods on CIFAR-10/100 (v2). This splitting was proposed in pFedGP’s paper—where the same result is observed [Ach+21, Table 1]. This occurs because each client only receives samples for a few classes, and then its local task is much easier than the global one.

**Fairness across clients.** Table 3.5 also shows the bottom decile of the accuracy of personalized models, i.e., the  $(T/10)$ -th worst accuracy (the minimum accuracy is particularly noisy, notably because some local test datasets are very small). We observe that even clients with the worst personalized models are still better off when kNN-Per is used for training.

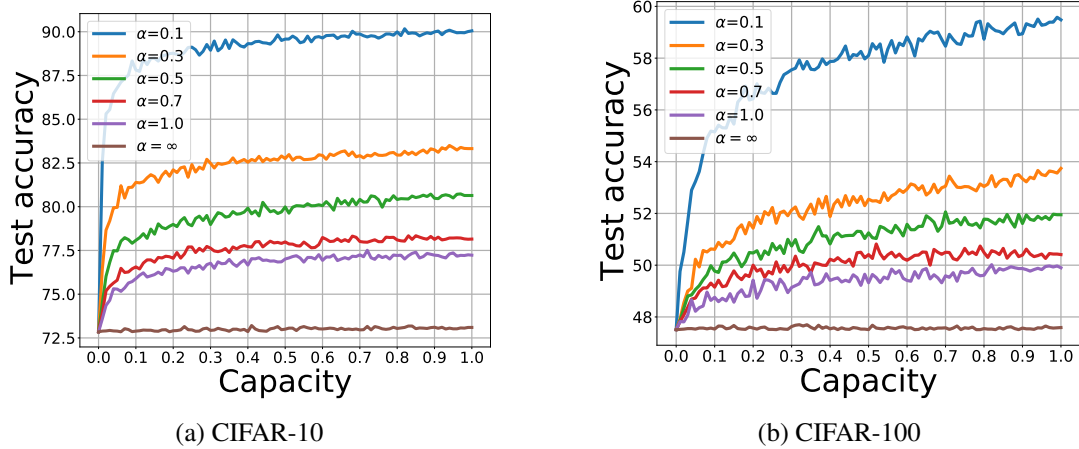


Figure 3.2: Test accuracy vs capacity (local datastore size). The capacity is normalized with respect to the initial size of the client’s dataset partition. Smaller values of  $\alpha$  correspond to more heterogeneous data distributions across clients.

**Generalization to unseen clients.** An advantage of  $kNN-Per$  is that a “new” client arriving after training may easily learn a personalized model: it may simply retrieve the global model (whose training it did not participate to) from the orchestrator and use it to build the local datastore for  $kNN$ . Even if this scenario was not explicitly considered in their original papers, other personalized FL methods can also be adapted to new clients as follows.  $FedAvg+$  personalizes the global model through stochastic gradient updates on the new client’s local dataset.  $Ditto$  operates similarly, but maintains a penalization term proportional to the distance between the personalized model and the global model.  $FedRep$  trains the network head using the local dataset, while freezing the body as in the global model. For  $pFedGP$  new clients inherit the previously trained shared network and compute their local kernel.  $ClusteredFL$  assigns the new client to one learned cluster model using a held-out validation set. In the case of  $FedAvg$ , there is no personalization and the new client uses directly the global model. We performed an experiment where only 80% of the clients participated to the training and the remaining 20% joined later. Results in Table 3.6 show that, despite its simplicity in dealing with new clients,  $kNN-Per$  still outperforms all other methods.

**Effect of local dataset’s size.** Beside its relevance for some practical scenarios, the distinction between old and new clients also helps us to evaluate how different factors contribute to the final performance of  $kNN-Per$ . For example, to understand how the size of the local dataset affects performance, we reduced proportionally the size of new clients’ local datasets, while maintaining unchanged the global model, which was trained on old clients. Figure 3.2 shows that new clients still reap most of  $kNN-Per$ ’s benefits even if their local datastore is reduced by a factor 3. Note that if we had changed the local dataset sizes also for old clients, the global model (and then the representation) would have changed too, making it difficult to isolate the effect of the local datastore size. We show the results for this experiment in Figure 3.7.

**Effect of data heterogeneity.** Figure 3.2 also shows that, as expected by Theorem 3.6.1, the benefit of the memorization mechanism is larger when data distributions are more heterogeneous (smaller  $\alpha$ ). While other methods also benefit from higher heterogeneity,  $kNN-Per$  appears to



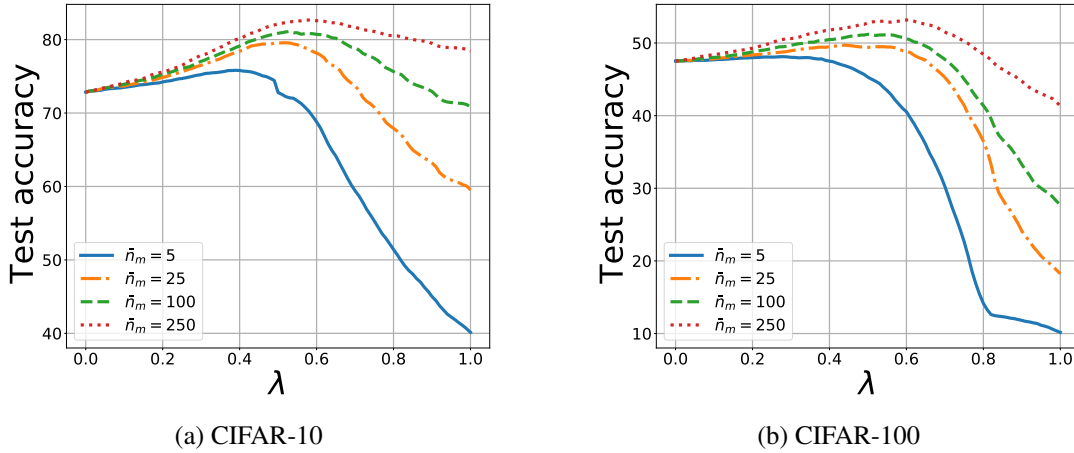


Figure 3.3: Test accuracy vs the interpolation parameter  $\lambda$  (shared across clients) for different average local dataset sizes. For  $\lambda = 1$  (resp.  $\lambda = 0$ ) the client uses only the  $k$ NN model (resp. the global model).

address statistical heterogeneity more effectively (Figure 3.11). Note that if local distributions were identical ( $\alpha \rightarrow \infty$ ), no personalization method would provide any advantage.

**Hyperparameters.**  $k$ NN-Per’s performance is not highly sensitive to the value  $k$  which can be selected between 7 and 14 for CIFAR-10 and between 5 and 12 for CIFAR-100 with less than 0.2 percentage points of accuracy variation (see Figure 3.6). Similarly, scaling the Euclidean distance by a factor  $\sigma$  has almost no effect for values of  $\sigma$  between 0.1 and 100 and between 1 and 100, respectively for CIFAR-10 and CIFAR-100 (see Figure 3.8). The interpolation parameter  $\lambda_t$  plays a more important role. Figure 3.9 shows that, as expected, the larger the local dataset, the more clients rely on the local  $k$ NN model. Interestingly, clients give a larger weight to the  $k$ NN model than to the global one ( $\lambda > 1/2$ ) for datasets with just one hundred samples (Figure 3.3).

**Effect of global model’s quality.** Assumption 21 stipulates that the smaller the expected loss of the global model, the better representations’ distances capture the variability of  $\mathbf{x} \mapsto \mathcal{D}_t(\cdot|\mathbf{x})$  and then the more accurate the  $k$ NN model. This effect is quantified by Lemma F.2, where the loss of the local memorization mechanism is upper bounded by a term that depends linearly on the loss of the global model. In order to validate this assumption, we study the relation between the test accuracies of the global model and  $k$ NN-Per. In particular, we train two global models, one for CIFAR-10 and the other for CIFAR-100, in a centralized way, and we save the weights at different stages of the training, leading to global models with different qualities. Figure 3.4 shows the test accuracy of  $k$ NN-Per with  $\lambda = 1$  (i.e., when only the  $k$ NN predictor is used) as a function of the global model’s test accuracy for different levels of heterogeneity on CIFAR-10 and CIFAR-100 datasets. We observe that, quite unexpectedly, the relation between the two accuracies is almost linear. The experiments also confirm what observed in Figure 3.2:  $k$ NN-Per performs better when local distributions are more heterogeneous (smaller  $\alpha$ ). Similar plots with  $\lambda$  optimized locally at every client are shown in Figure 3.5.

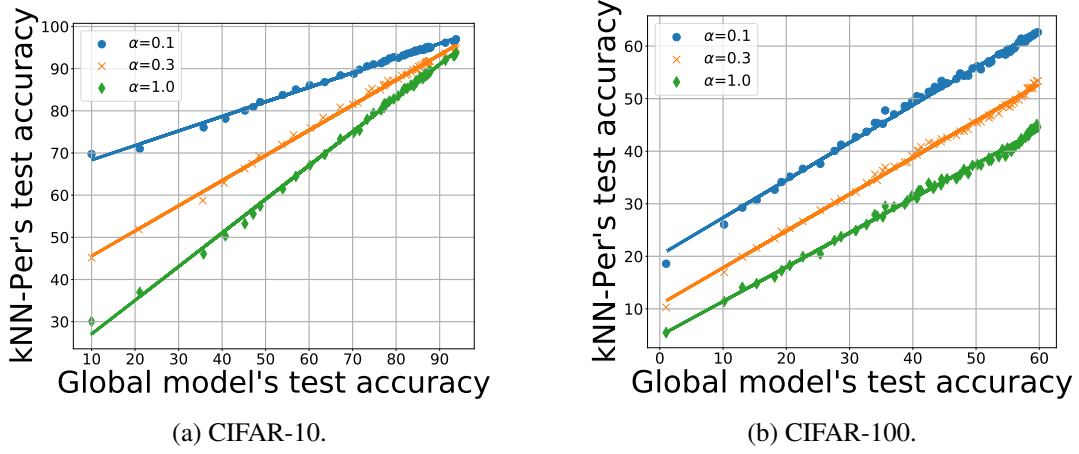


Figure 3.4: Effect of the global model quality on the test accuracy of kNN-Per with  $\lambda_t = 1$  for each  $t \in [T]$ .

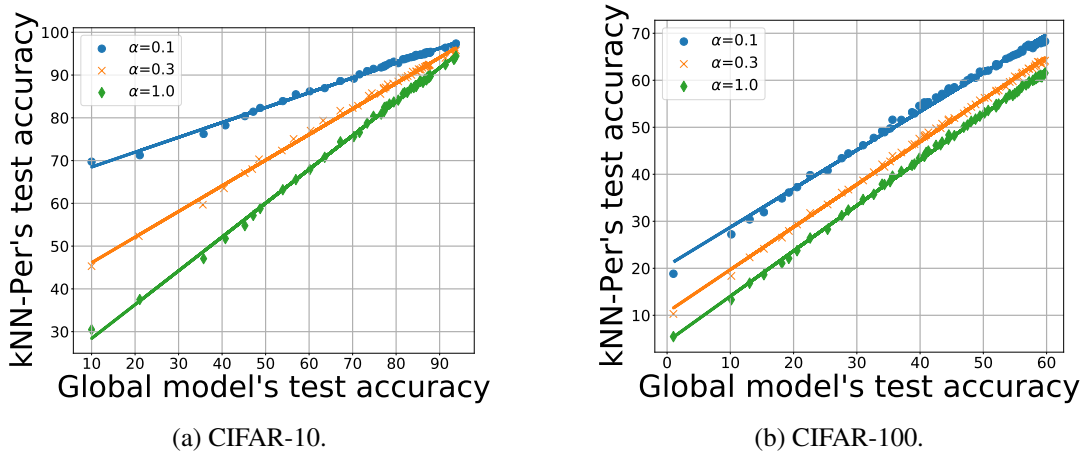


Figure 3.5: Effect of the global model quality on the test accuracy of kNN-Per with  $\lambda_t$  tuned per client.

**Effect of kernel scale parameter  $\sigma$ .** We consider distance metrics of the form

$$\forall \mathbf{z}, \mathbf{z}' \in \mathbb{R}^p; d_\sigma(\mathbf{z}, \mathbf{z}') = \frac{\|\mathbf{z} - \mathbf{z}'\|_2}{\sigma}, \quad (3.40)$$

where  $\sigma \in \mathbb{R}^+$  is a scale parameter. Figure 3.8 shows that kNN-Per's performance is not highly sensitive to the selection of the length scale parameter, as scaling the Euclidean distance by a constant factor  $\sigma$  has almost no effect for values of  $\sigma$  between 0.1 and 1000.

**Effect of datastore's size on the optimal  $\lambda$ .** Figure 3.9 shows the effect of the local number of samples  $n_t$  on the optimal mixing parameter  $\lambda_{\text{opt}}$  (evaluated on the client's test dataset). The number of samples changes across clients and, for the same client, with different values of the capacity. The figure shows a positive correlation between the local number of samples and the optimal mixing parameter and then validates the intuition that clients with more samples tend to rely more on the

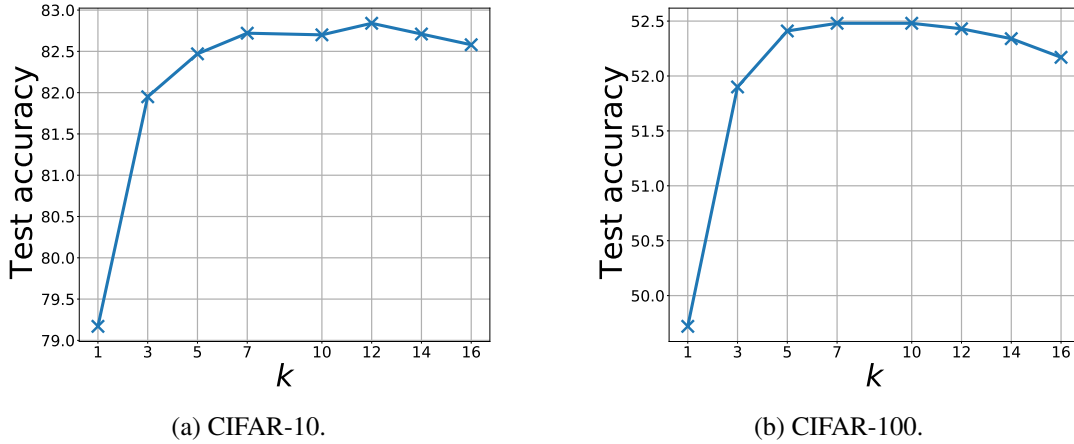
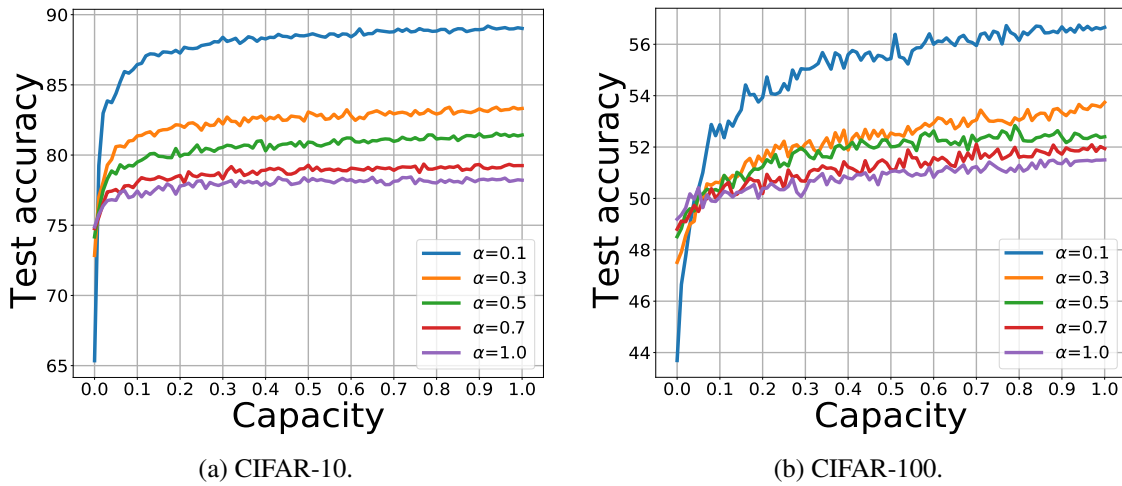
Figure 3.6: Test accuracy vs number of neighbors  $k$ .

Figure 3.7: Test accuracy vs capacity (local dataset size) when the global model is retrained for each value of  $\alpha$ . The capacity is normalized with respect to the initial size of the client’s dataset partition. Smaller values of  $\alpha$  correspond to more heterogeneous data distributions across clients. The curves start from different accuracy values for zero capacity, but are qualitatively similar to those in Figure 3.2 for large capacities. As expected, the global model performs worse the more heterogeneous the local distributions are, but the local model is able to compensate such effect (at least partially) as far as the dataset is large enough.

memorization mechanism than on the base model, as captured by the generalization bound from Theorem 3.6.1.

**Effect of hardware heterogeneity.** In our experiments above, clients’ local datasets had different size, which can also be due to different memory capabilities. In order to investigate more in depth the effect of system heterogeneity, we split the new clients in two groups: “weak” clients with normalized capacity  $1/2 - \Delta C$  and “strong” clients with normalized capacity  $1/2 + \Delta C$ , where  $\Delta C \in (0, 1/2)$  is a parameter controlling the hardware heterogeneity of the system. Note that the total amount of memory in the system is constant, but varying  $\Delta C$  changes its distribution across clients from a homogeneous scenario ( $\Delta C = 0$ ) to an extremely heterogeneous one ( $\Delta C = 0.5$ ).

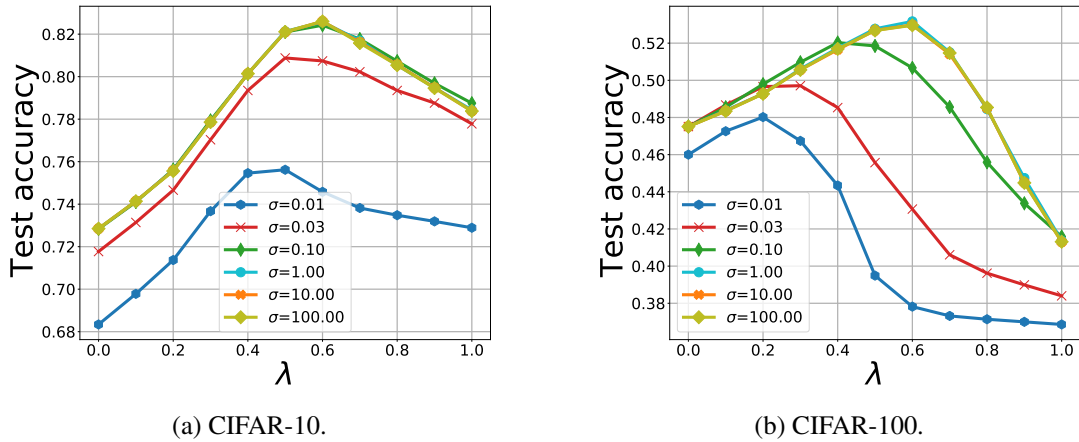


Figure 3.8: Test accuracy vs the interpolation parameter  $\lambda$  for different values of the kernel scale parameter  $\sigma$ .

Figure 3.10 shows the effect of the hardware heterogeneity, as captured by  $\Delta C$ . As the marginal improvement from additional memory is decreasing (see, e.g., Figure 3.2) the gain for strong clients does not compensate the loss for weak ones. The overall effect is then that the average test accuracy decreases as system heterogeneity increases.

**Adding compression techniques.** `kNN-Per` can be combined with nearest neighbors compression techniques as `ProtoNN` [Gup+17]. `ProtoNN` reduces the amount of memory required by jointly learning 1) a small number of prototypes to represent the entire training set and 2) a data projection into a low dimensional space. We combined `kNN-Per` and `ProtoNN` and explored both the effect of the number of prototypes and the projection dimension used in `ProtoNN`. For each client, the number of prototypes is set to a given fraction of the total number of available samples. We refer to this quantity also as capacity. We varied the capacity in the grid  $\{i \times 10^{-1}, i \in [10]\}$ , and the projection dimension in the grid  $\{i \times 100, i \in [12]\} \cup \{1280\}$ . Note that smaller projection dimension and less prototypes correspond to a smaller memory footprint, suited for more restricted hardware. Our implementation is based on `ProtoNN`'s official.\* Figure 3.12a shows that, on CIFAR-10, `ProtoNN` allows to reduce the `kNN-Per`'s memory footprint by a factor four (using  $n_t/3$  prototypes and projection dimension 1000) at the cost of a limited reduction in test accuracy (82.3% versus 83.0% in Table 3.5). Note that `kNN-Per` with `ProtoNN` still outperforms all other methods. On CIFAR-100, `ProtoNN`'s compression techniques appear less advantageous: the approach loses about 3 percentage points (52.1% versus 55.0% in Table 3.5) while only reducing memory requirement by 20%.

**Robustness to distribution shift.** As previously mentioned, `kNN-Per` offers a simple and effective way to address statistical heterogeneity in a dynamic environment where client's data distributions change after training. We simulate such a dynamic environment as follows. Client  $t$  initially has a datastore built using instances sampled from a data distribution  $\mathcal{D}_t$ . For time step  $k < k_0$ , client  $t$  receives a batch of  $n_t^{(k)}$  instances drawn from  $\mathcal{D}_t$ . At time step  $k_0$ , we suppose that a data distribution shift takes place, i.e., for  $k_0 \leq k \leq K$ , client  $t$  receives  $n_t^{(k)}$  instances drawn

\*<https://github.com/Microsoft/EdgeML>.

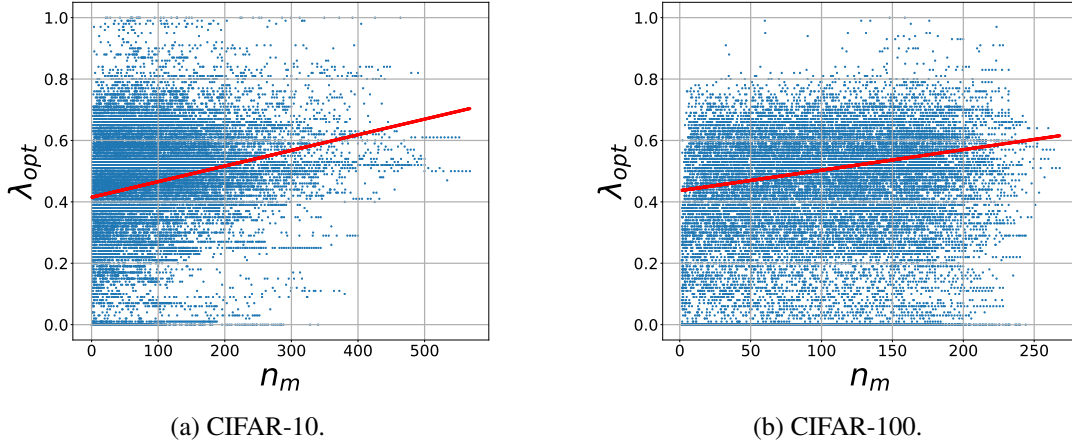


Figure 3.9:  $\lambda_{\text{opt}}$  vs local number of samples  $n_t$ .

from a data distribution  $\mathcal{D}'_t \neq \mathcal{D}_t$ . Upon receiving new instances, client  $t$  may use those instances to update its datastore. We consider 3 different strategies: (1) *first-in-first-out* (FIFO) where, at time step  $t$ , the  $n_t^{(k)}$  oldest samples are replaced by the newly obtained samples; (2) *concatenate*, where the new samples are simply added to the datastore; (3) *fixed datastore*, where the datastore is not updated at all. In our simulations, we consider CIFAR-10/100 datasets with  $T = 100$  clients. Once again, we used a symmetric Dirichlet distribution to generate two datasets for every client. In particular, for each label  $y$  we sampled two vectors  $p_y$  and  $p'_y$  from a Dirichlet distribution of order  $T = 100$  and parameter  $\alpha = 0.3$ . Then, for client  $t$ , we generated two datasets  $\mathbb{S}_t$  and  $\mathbb{S}'_t$  by allocating  $p_{y,t}$  and  $p'_{y,t}$  fraction of all training instances of class  $y$ .<sup>\*</sup> Both  $\mathbb{S}_t$  and  $\mathbb{S}'_t$  are partitioned into training and test sets following the original CIFAR training/test data split. Half of the training set obtained from  $\mathbb{S}_t$  is stored in the datastore, while the rest is further partitioned into  $k_0$  batches  $\mathbb{S}_t^{(0)}, \dots, \mathbb{S}_t^{(k_0-1)}$ . These batches are the new samples arriving at client  $t$ . Similarly,  $\mathbb{S}'_t$  is partitioned into  $K - k_0$  equally sized batches. Figure 3.13 shows the evaluation of the test accuracy across time. If clients do not update their datastores, there is a significant drop in accuracy as soon as the distribution changes at  $k_0 = 50$ . If datastores are updated using FIFO, we observe some random fluctuations for the accuracy for  $k < k_0$ , as repository changes affect the kNN predictions. While accuracy inevitably drops for  $k = k_0$ , it then increases as datastores are progressively populated by instances from the new distributions. Once all samples from the previous distributions are evicted, the accuracy settles around a new value (higher or lower than the one for  $k < k_0$  depending on the difference between the new and the old distributions). If clients keep adding new samples to their datastores (the “concatenate” strategy), results are similar, but 1) accuracy increases for  $k < k_0$  as the quality of kNN predictors improves for larger datastores, 2) accuracy increases also for  $k > k_0$ , but at a slower pace than what observed under FIFO, as samples from the old distribution are never evicted.

### 3.6.4 Conclusion

In this section, we showed that local memorization at each client is a simple and effective way to address statistical heterogeneity in federated learning. In particular, while a global model trained

<sup>\*</sup>We always make sure that  $|\mathbb{S}_t| \leq |\mathbb{S}'_t|$ .

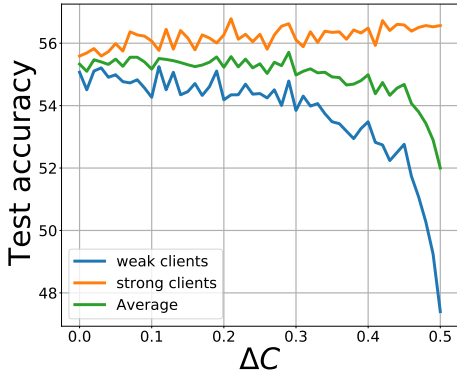


Figure 3.10: Effect of system heterogeneity across clients on CIFAR-100 dataset. The size of the local datastore increases (resp. decreases) with  $\Delta C$  for strong (resp. weak) clients.

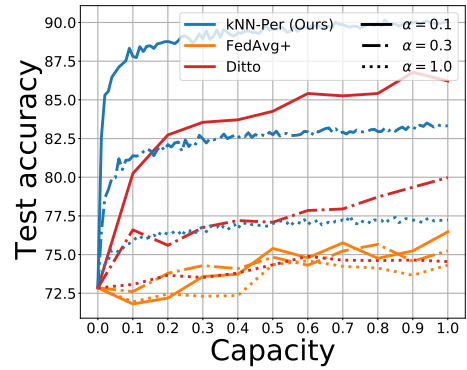
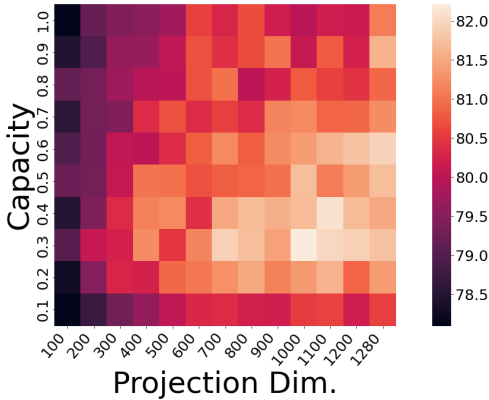
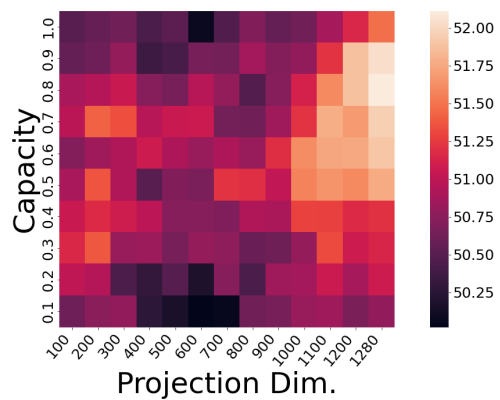


Figure 3.11: Test accuracy vs capacity (local datastore size) for different methods on CIFAR-10. The capacity is normalized with respect to the initial size of the client’s dataset partition.



(a) CIFAR-10.



(b) CIFAR-100.

Figure 3.12: Test accuracy when the  $k$ NN mechanism is implemented through `PROTONN` for different values of projection dimension and number of prototypes (expressed as a fraction of the local dataset). CIFAR-10 (left) and CIFAR-100 (right) datasets.

with classic FL techniques, like `FedAvg`, may not deliver accurate predictions at each client, it may still provide a good representation of the input, which can be advantageously used by a local  $k$ NN model. This finding suggests that combining memorization techniques with neural networks has additional benefits other than those highlighted in the seminal papers [Gre+15; JM15] and the recent applications to natural language processing [Kha+19; Kha+21].

The better performance of `kNN-Per` in comparison to `FedRep` and `pFedGP` show that jointly learning the shared representation and the local models (as `FedRep` and `pFedGP` do) may lead to potentially conflicting and interfering goals, but further study is required to understand this interaction. Semi-parametric learning [Bic+93] could be the right framework to formalize this problem, but its extension to a federated setting is still unexplored.

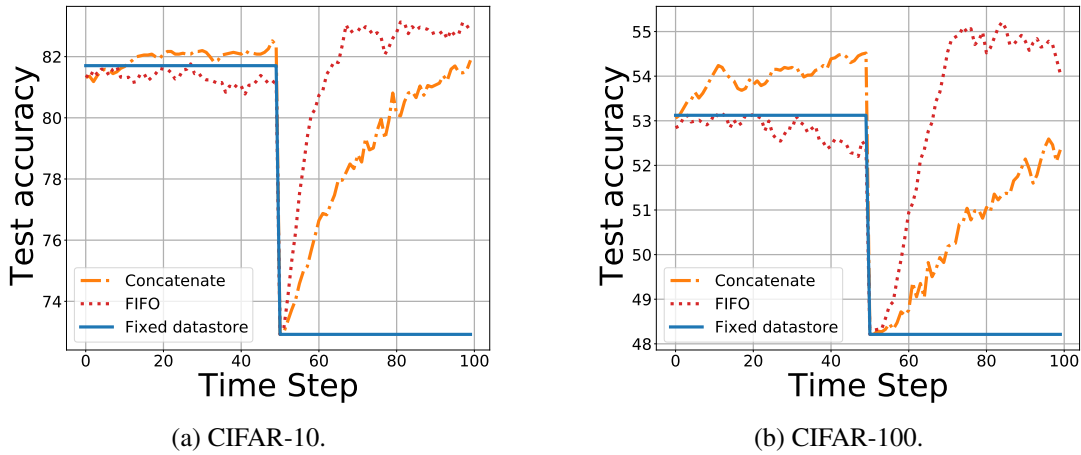


Figure 3.13: Test accuracy when a distribution shift happens at time step  $k_0 = 50$  for different datastore management strategies.

### 3.7 A Comparison between FedEM and kNN-Per

In this chapter, we have conducted an extensive exploration of personalized federated learning, demonstrating its efficacy not only in addressing statistical heterogeneity but also in mitigating system and temporal heterogeneity challenges. We have introduced two novel personalization algorithms, each grounded in distinct principles. The first approach, FedEM, leverages the mixture assumption, positing that each local data distribution is a composite of unknown underlying distributions. In contrast, the second approach, kNN-Per, operates on the representation assumption. It presupposes the existence of a global model capable of serving as a feature/representation extractor. Furthermore, it establishes that when two samples exhibit similar representations, the likelihood of their labels being the same increases.

The mixture assumption is a flexible and generic assumption that encompasses most of the personalized FL approaches previously proposed in the literature (as we show in Section 3.5.2). Notably, it provides a principled means of quantifying data distribution similarity among clients—a feature exploited by Kim et al. [Kim+23] to analyze and characterize internal evasion attacks within the federated learning context. Additionally, owing to its adaptability, the mixture assumption has been instrumental in modeling and mitigating the issue of distributed concept shift in federated learning [Zhu+22; Jot+23]. However, one significant drawback of employing a mixture model lies in the local computational and communication overhead incurred due to the maintenance and parallel training of multiple base models.

Conversely, the representation assumption, while less flexible, offers a streamlined approach that effectively addresses not only statistical heterogeneity but also system and temporal heterogeneity. The local memorization technique employed in kNN-Per serves as a straightforward and efficient solution to these challenges. Moreover, thanks to its simplicity, kNN-Per can seamlessly integrate as a lightweight module atop standard federated learning approaches, requiring minimal modification to these established methods.

---

# Federated Learning in Dynamic Environments

In the preceding chapters of this manuscript, our focus has been on scenarios wherein clients operate within static environments and possess access to identically distributed examples gathered prior to the initiation of training. However, as elucidated in Section 1.4, relying solely on static datasets can prove to be suboptimal, and in some cases, impractical. This is primarily due to the fact that, firstly, newly acquired samples during training are often disregarded, and secondly, clients may grapple with limited memory capacities, hindering the storage of an extensive number of data samples. To illustrate, nodes within a sensor network continually amass new measurements, yet their local memory may only accommodate a fraction of this influx [De +16]. Furthermore, in various real-world applications, the underlying data distributions of clients exhibit non-stationary characteristics and undergo constant evolution. For instance, user sentiments and preferences can undergo drastic changes owing to external factors such as pandemics and macroeconomic shifts [Koh+21; Gar+21].

This chapter explores federated learning within dynamic environments, where clients collaboratively learn from distributed data streams characterized by the continual generation of data. Our focus is specifically on two distinct scenarios, each rooted in differing assumptions about the data process. The first scenario (Section 4.3) addresses instances where samples within the data stream are independently drawn from an undisclosed fixed distribution. In the second scenario (Section 4.4), we posit that client data distributions are mixtures of a finite number of undisclosed common underlying distributions, each varying in terms of mixing coefficients.

This chapter builds upon our works [Mar+23b], presented in the proceedings of the 26<sup>th</sup> International Conference on Artificial Intelligence and Statistics (AISTATS'23), and [MMb], currently under review.

## 4.1 Introduction

Collaboration in dynamic environments introduces a distinct set of challenges and opportunities that go beyond those encountered in traditional federated learning scenarios with static datasets. Two distinct and orthogonal challenges come to the forefront in this context.

First, when learning from a data stream, every client only has access to samples currently present in its local memory. Due to the limited storage capacity at each client and to the variability in the number of new samples arriving across time, samples may spend different amounts of time in memory and then be used a different number of times during training (see Figure 4.1). In order to potentially compensate for such *temporal data-access heterogeneity*, one has to allow samples to be weighted differently over time and across clients. In Section 4.3, we provide a formal definition



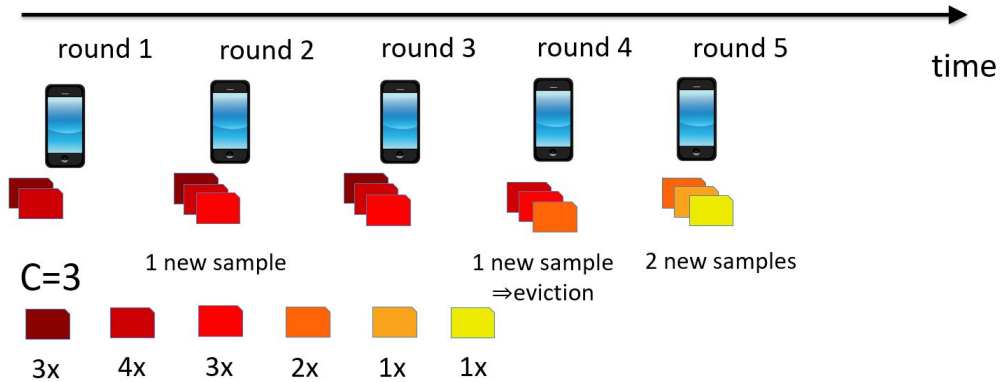


Figure 4.1: A depiction of a data stream: The client/device, with a limited storage capacity ( $C = 3$ ), updates its local memory following a FIFO (First-In-First-Out) rule. This involves evicting the oldest samples from memory to make space for the most recent ones. Consequently, various samples, represented by distinct colors, reside in memory for varying durations.

of this challenge and conduct an analysis within the specific scenario where samples within the data stream are independently drawn from an undisclosed stationary distribution that may vary across clients. Our analysis (Section 4.3.2) shows a *bias-optimization trade-off*: by controlling the relative importance of older samples in comparison to newer ones, one can speed training up at the cost of a larger bias of the learned model, or reduce the bias at the cost of a longer training time.

The second challenge emerges in uncertain environments, characterized by clients receiving data in an online fashion, with the information revealed only after the clients make their model predictions. This intriguing challenge, termed *online federated learning*, and initially presented in [MHP21], introduces learning intricacies even when samples spend the same duration in memory—such as when all samples are stored in memory only during the time step of their collection. This dynamic scenario adds an extra layer of complexity to the federated learning framework, requiring adaptive strategies to effectively learn from data that is only fully disclosed post-prediction.

The authors of [MHP21] makes no statistical assumptions about the data stream and considers a worst-case scenario where an adversary picks at every time-step the worst instance to provide clients with. This work provides performance guarantees in terms of an appropriately defined collective regret metric. However, this worst-case analysis falls short of demonstrating any collaboration benefit. In fact, as we have seen in Chapter 3, collaborative learning is impossible with no assumption on local data distributions, in the sense that some assumption on local data distributions are needed for collaboration to be provably beneficial (Section 3.4). Motivated by this impossibility result, our analysis in Section 4.4 considers online federated learning in constrained adversarial scenarios, as opposed to [MHP21] that does not constrain the adversary. In particular, we consider a dynamic version of the mixture assumption introduced in Section 3.5: client data distributions are mixtures of a finite number of fixed undisclosed common underlying distributions, each varying (potentially adversarially) in terms of mixing coefficients. Our formulation has the advantage of showing the benefit of collaboration. In particular, our proposed Federated EM Online Mirror Descent (FEM-OMD, Algorithm 14) algorithm leverages all of the data stored across clients to learn the parameters of the underlying distributions using EM-type updates, while enabling each client to adapt to the temporal variation of its data distribution, by locally learning the mixing coefficients.

### 4.1.1 Contributions

In this chapter, we study two orthogonal challenges encountered in federated learning within dynamic environments. The first challenge takes place due to the limited storage capacity at each client and to the variability in the number of new samples arriving across time. As a result, various samples reside in memory for varying durations. In Section 4.3, we formulate and study the problem of learning from separate data streams, in the particular (and “easy”) setting where samples within the data stream are independently drawn from an undisclosed stationary distribution. Our analysis shows a bias-optimization trade-off: by controlling the relative importance of older samples in comparison to newer ones, one can speed training up at the cost of a larger bias of the learned model, or reduce the bias at the cost of a longer training time. The analysis also provides insights to optimally configure our federated algorithm. We demonstrate the relevance of our theoretical results through simulations spanning a wide range of machine learning tasks. In particular, experiments show that “reasonable” ways to extend FedAvg to data streams may lead to poor learned models, while our configuration rule consistently leads to almost-optimal performance.

The second challenge takes place due to the variability of the client’ underlying distributions variability across time. In Section 4.4, we provide a novel formulation for the problem of online federated learning based on the assumption that clients’ data distributions are mixtures of a finite number of unknown underlying distributions with varying mixing weights. In comparison to previous work, e.g. [MHP21], our assumption allows the clients to provably benefit from collaboration, while allowing clients’ data distributions to vary in a potentially (constrained) adversarial manner. Afterwards, we propose *Federated Expectation-Maximization Online Mirror Descent* (FEM-OMD), a federated variant of the OMD algorithm, where the gradient of the cost function is estimated through an EM-like algorithm at each time-step. FEM-OMD leverages all of the data stored across clients to learn the parameters of the underlying distributions using EM updates, while enabling each client to adapt to the temporal variation of its data distribution. We analyze the regret guarantees of FEM-OMD in the case of well-separated spherical Gaussian mixture models. Specifically, we establish a  $\mathcal{O}(\sqrt{T \log(m)} + T/\sqrt{n})$  regret bound, where  $T$  is the time horizon,  $m$  is the number of the underlying distributions, and  $n$  is the number of samples received by each client. Finally, through experimental results on synthetic datasets and FL benchmarks, we demonstrate the effectiveness of our approach in online federated settings and show that our scheme allows the clients to benefit from collaboration.

### 4.1.2 Organization

The rest of this chapter is organized as follows. Section 4.2 provides a review of related work. Section 4.3.1 formulates the problem of federated learning for data streams, when each clients have limited storage capacity, and receives a varying number of new samples at each time-step from some stationary undisclosed underlying data distribution, that potentially depends on the client identity. Section 4.3.2 describes our FL algorithm for data streams and states its convergence results. Section 4.3.3 studies a scenario of practical interest and exploits the theoretical result in Section 4.3.2 to provide configuration rules for our algorithm. Section 4.3.4 empirically evaluates the performance of our algorithm.

In Section 4.4.1, we formulate the problem of online federated learning within constrained adversarial scenarios, and introduces the mixture assumption: *clients’ data distributions are mixtures of a finite number of unknown underlying distributions with varying mixing coefficients.*

Section 4.4.2 describes our algorithm `FEM-OMD`, used to learn under the mixture assumption. Section 4.4.3.1 studies the particular scenario where the underlying distributions are well-separated Gaussians, and states the regret guarantees of `FEM-OMD` in this scenario. Finally, we provide experimental results in Section 4.4.5 before concluding in Section 4.4.6.

## 4.2 Related Work

Since its introduction in the seminal works [Kon+17b; McM+17], federated learning has received increasing attention as a promising large-scale distributed learning framework and has been applied to a wide range of tasks, including language modeling [Yan+18], automatic speech recognition [Gao+22], medical imaging [Cou+19; Sil+19], and recommender systems [Yan+20a]. Our focus on data streams is a key difference with respect to most of the FL literature, which assumes clients have static datasets. In particular, this assumption is shared by the theoretical work studying FL algorithms’ convergence on non-iid data and under partial clients’ participation [Li+19], PAC learning bounds [MSS19], privacy guarantees [Wei+20], or resilience to Byzantine faults [Bla+17].

Learning from a data stream enjoys an extensive literature with applications, for example, to the financial sector [ZS02], network monitoring [BW01], and sensor networks [De +16]. In this field, we can roughly distinguish three main lines of research corresponding to different assumptions about the data process. The first focuses on the case where samples in the data stream are drawn independently from some fixed unknown distribution; this setting can be analyzed through stochastic approximation [MB11]. The second line allows the data distribution to change over time and falls then in the context of continual learning, where a model is trained on a sequence of tasks and each task can correspond to a different data distribution [Thr94; KD12; RE13; Kir+17; Sch+18]. Finally, the third line drops any assumption about the data stream, which may be thought to be generated by an adversary. This setting can be studied in the framework of online learning with regret guarantees [Zin03]. We consider that data at each client is drawn from the same distribution. Learning from multiple data streams with different samples’ generation rates and clients’ memory sizes sets our work apart from the papers mentioned above.

There is almost no work formalizing the problem of federated learning for data streams and providing a theoretical analysis. To the best of our knowledge, the only exceptions are [Che+20b], [Yoo+21], and [OZ21].

[Che+20b] propose `ASO-Fed`, an asynchronous FL algorithm to minimize the empirical loss computed over the aggregation of clients’ data streams. Although some convergence results are stated in the paper, their interest and applicability are questionable, as the analysis requires that all clients have the same optimal model and that updates at any time  $t$  are consistent with new samples arriving in the future. Indeed, the paper mentions that clients can receive new samples during training (see Fig. 2), but also requires that, at any time  $t$  and for any client  $k$ , the expected value of the update  $\nabla \zeta_k(w)$  has a non-null component in the direction of the gradient of the global empirical loss  $F$ , which depends on samples arriving *after* time  $t$  (see Assumption 1). Moreover, the bounded gradient dissimilarity assumption implies that the minimizer of  $F$  ( $F$  is assumed to be strongly-convex) is also a stationary point of each local objective function  $f_k$  (consider  $\beta = 0$  and  $\lambda = 0$ ). On the contrary, the theoretical analysis in our work holds under statistical heterogeneity across clients’ local data distributions and accounts for the bias due to working with samples currently stored at clients. Moreover, we provide statistical learning guarantees for our algorithm.

[Yoo+21] propose `FedWeIT`, which extends regularization-based algorithms for continual

learning to the FL setting. The main goal of `FedWeIT` is to minimize interference between incompatible tasks while allowing positive knowledge transfer across clients during learning, but no generalization guarantee is provided. [OZ21] consider the problem of online federated learning under constraints on the amount of resources consumed over the whole time horizon and proposes an online mirror descent-based algorithm with regret guarantees. Differently from our contribution, both [OZ21] and [Yoo+21] assume each client can only use the most recent data. Our experiments show that reusing as little as 5% of the collected samples may be highly beneficial.

Federated learning from temporally shifting distributions [Zhu+22; Eic+19; Din+20; GLT23] is a related, yet different, problem to learning from a data stream. These papers assume the shift is due to changes in the set of available clients (e.g., because of diurnal patterns), but clients’ local datasets do not change. The only exception is [GLT23], which can capture a setting where clients keep collecting data during training without storage constraints. The model considered in [GLT23] can capture a setting where clients keep collecting data during training without storage constraints. Indeed, clients track the dynamic objective in [GLT23, Eq. (2)] which depends on data samples received until the current time. Theoretical results assume that new data is drawn from a client-independent distribution. This is shown by [GLT23, Eq. (5)], which requires that local gradients computed on new data samples are unbiased estimators of the gradient of the global objective function. Instead, our analysis takes into account both memory constraints and statistical heterogeneity across clients’ local data distributions.

Finally, we mention a number of papers studying different variants of “online federated learning” problems, mostly focusing on dynamic resource allocation. Many of them are discussed in the recent survey [DM22]. Among these papers, [Dam+20] propose `Fleet`, a middleware between the edge device operating system and the machine learning application, which can be used to learn on data streams. The middleware is designed with the device’s energy minimization as the main concern. [Jin+20] propose an online algorithm to dynamically select the participating clients and their number of local gradient iterations at each communication round to minimize the cumulative resource usage over time under a constraint on the quality of the final model. [Zho+20] study a similar problem. They include the possibility of discarding new data points or distributing them to clients with more resources and propose a resource allocation algorithm based on Lyapunov optimization [Nee10]. Both [Jin+20] and [Zho+20] ignore the possibility of reusing samples across multiple communication rounds.

### 4.3 Federated Learning for Data Streams

In this section, we aim to investigate the challenge of learning from distributed data streams. Our focus is on scenarios where samples collected by each client are independently drawn from undisclosed stationary distributions, potentially unique to each client. The technical focal point of this section centers around a new form of heterogeneity termed *temporal data-access heterogeneity*. This heterogeneity stems from the evolving nature of local datasets over time, a consequence of the limited storage capacity at each client and the variability in the influx of new samples. It manifests as variability in sample stay-time, indicating that samples may reside in memory for varying durations and be used different numbers of times during training.

To address the variability in stay-time, we propose a general FL algorithm designed for learning from dynamic data streams. This algorithm relies on a thoughtful weighted empirical risk minimization approach, as outlined in Section 4.3.2. Our theoretical analysis, detailed in Section 4.3.1,

provides insights for configuring such an algorithm. We subsequently assess its performance across a wide range of machine learning tasks, as outlined in Section 4.3.4.

### 4.3.1 Problem Formulation

In this work, we use  $[M] \triangleq \{1, \dots, M\}$  to denote the set of positive integers up to  $M$ . We consider  $M > 0$  clients; each of them corresponds to a potentially different learning task. We associate to each client  $m \in [M]$ : 1) a probability distribution  $\mathcal{P}_m$  over a domain  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ , 2) a counting process  $N_m^{(t)}, t \geq 0$ , and 3) a dynamic memory/cache  $\mathcal{M}_m^{(t)}, t > 0$  of capacity  $C_m > 0$ . At time step  $t > 0$ , client  $m \in [M]$  receives a batch  $\mathcal{B}_m^{(t)} = \{\mathbf{z}_m^{(t,i)} = (\mathbf{x}_m^{(t,i)}, y_m^{(t,i)}), i \in [b_m^{(t)}]\}$  containing  $b_m^{(t)} \triangleq N_m^{(t)} - N_m^{(t-1)}$  samples drawn i.i.d. from  $\mathcal{P}_m$ . Client  $m \in [M]$  can cache a sub-part of the samples in its local memory, without exceeding the capacity  $C_m$ . Without loss of generality we suppose that  $1 \leq b_m^{(t)} \leq C_m$ . We consider a finite time horizon  $T > 0$ , and we let  $N_m \triangleq N_m^{(T)}$  and  $\mathcal{S}_m \triangleq \bigcup_{t=1}^T \mathcal{B}_m^{(t)}$  denote the number and the set of samples gathered by client  $m$  up to the time horizon  $T$ . We write  $\mathcal{S}_m = \{\mathbf{z}_m^{(i)}, i \in [N_m]\}$ , where we arbitrarily ordered the elements of  $\mathcal{S}_m$ . We define  $\mathcal{I}_m^{(t)} \subset [N_m]$  to be the set of the indices of samples present at memory  $\mathcal{M}_m^{(t)}$ , i.e.,  $j \in \mathcal{I}_m^{(t)}$  if and only if  $\mathbf{z}_m^{(j)} \in \mathcal{M}_m^{(t)}$ . Finally,  $\mathcal{S} \triangleq \bigcup_{m=1}^M \mathcal{S}_m$  denotes the training dataset (aggregated across clients and across time) with size  $N \triangleq \sum_{m=1}^M N_m$ . The relative size of client- $m$ 's dataset is  $n_m \triangleq N_m/N$ .

Let  $\mathcal{H} = \{h_\theta : \mathcal{X} \mapsto \mathcal{Y}, \theta \in \Theta \subset \mathbb{R}^d\}$  be a set of parametric hypotheses/models mapping  $\mathcal{X}$  to  $\mathcal{Y}$ , and  $\ell : \Theta \times \mathcal{Z} \mapsto \mathbb{R}^+$  be a loss function.

We use  $\text{Pdim}(\ell \circ \mathcal{H})$  to denote the pseudo-dimension [MRT18] of the hypothesis class  $\mathcal{H}$  w.r.t. the loss  $\ell$ . The pseudo-dimension generalizes the Vapnik–Chervonenkis (VC) dimension [VC15] to loss functions different from the 0–1 loss.

We define  $\mathcal{L}_{\mathcal{P}}(\theta) \triangleq \mathbb{E}_{\mathbf{z} \sim \mathcal{P}}[\ell(\theta; \mathbf{z})]$  to be the true (expected) risk of hypothesis  $h_\theta \in \mathcal{H}$  under a generic probability distribution  $\mathcal{P}$  over  $\mathcal{Z}$  and we define  $\mathcal{L}_{\mathcal{S}}(\theta) = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{x}, y) \in \mathcal{S}} \ell(\theta; \mathbf{z})$  to be the empirical risk of model (hypothesis)  $h_\theta \in \mathcal{H}$  on a generic dataset  $\mathcal{S}$  of samples from  $\mathcal{Z}$ .

In federated learning, clients, usually, collaborate to solve

$$\underset{\theta \in \Theta}{\text{minimize}} \mathcal{L}_{\mathcal{P}^{(\alpha)}}(\theta) = \sum_{m=1}^M \alpha_m \mathcal{L}_{\mathcal{P}_m}(\theta), \quad (4.1)$$

where  $\mathcal{P}^{(\alpha)} \triangleq \sum_{m=1}^M \alpha_m \cdot \mathcal{P}_m$  and  $\alpha \triangleq (\alpha_m)_{1 \leq m \leq M}$  with  $\alpha_m \geq 0$  and  $\|\alpha\|_1 = 1$ . Common choices for  $\alpha$  are  $\alpha_m = n_m$  and  $\alpha_m = \frac{1}{M}$ . The first one corresponds to minimizing the empirical loss over the aggregate training dataset  $\mathcal{S} = \bigcup_{m=1}^M \mathcal{S}_m$ , which gives the same importance to each sample. The second choice instead targets per-client fairness, by giving the same importance to each client.

In standard federated learning, local datasets  $\{\mathcal{S}_m\}_{m \in [M]}$  are available since the beginning of the training and the following empirical risk minimization problem is considered as a proxy for Problem (4.1):

$$\underset{\theta \in \Theta}{\text{minimize}} \sum_{m=1}^M \alpha_m \cdot \mathcal{L}_{\mathcal{S}_m}(\theta). \quad (4.2)$$

Our goal is to design a potentially randomized algorithm  $A$  solving, in a federated fashion, Problem (3.1) using clients' data streams and taking into account clients' memory constraints.

**Algorithm 13:** Meta Algorithm for Federated Learning from Data Streams

---

**Input** : Nbr of local epochs  $E$ ; mini-batch size  $K$ ; local learning rate  $\eta > 0$ ;  
sample weights  $\lambda = \left\{ \lambda_m^{(t,j)} ; m \in [M], t \in [T], j \in \mathcal{I}_m^{(t)} \right\}$

**Output** :  $\bar{\theta}^{(T)} = \sum_{t=1}^T q^{(t)} \theta^{(t)}$

- 1 **for**  $t = 1, \dots, T$  **do**
- 2     Server selects a subset  $\mathbb{S}^{(t)} \subseteq [M]$  of clients;
- 3     **for**  $m \in \mathbb{S}^{(t)}$  (in parallel) **do**
- 4          $\theta_m^{(t,1)} \leftarrow \theta^{(t)}$ ;
- 5         Sample  $\mathcal{B}_m^{(t)} = \{\mathbf{z}_m^{(t,1)}, \dots, \mathbf{z}_m^{(t,b_m^{(t)})}\} \sim \mathcal{P}_m^{b_m^{(t)}}$ ;
- 6          $\mathcal{M}_m^{(t)} \leftarrow \text{Update} \left( \mathcal{M}_m^{(t-1)}, \mathcal{B}_m^{(t)} \right)$ ;
- 7         **for**  $e = 1, \dots, E$  **do**
- 8             Sample  $\min \left\{ K, |\mathcal{I}_m^{(t)}| \right\}$  indices  $\xi_m^{(t,e)}$  uniformly from  $\mathcal{I}_m^{(t)}$ ;
- 9              $\mathbf{g}_m^{(t,e)} \leftarrow \frac{|\mathcal{I}_m^{(t)}|}{|\xi_m^{(t,e)}|} \sum_{j \in \xi_m^{(t,e)}} \frac{\lambda_m^{(t,j)}}{\sum_{j' \in \mathcal{I}_m^{(t)}} \lambda_m^{(t,j')}} \cdot \nabla \ell(\theta_m^{(t,e)}; \mathbf{z}_m^{(t,j)})$ ;
- 10              $\theta_m^{(t,e+1)} \leftarrow \theta_m^{(t,e)} - \eta \cdot \mathbf{g}_m^{(t,e)}$ ;
- 11         **end**
- 12     **end**
- 13      $\Delta^{(t)} \leftarrow \sum_{m=1}^M p_m^{(t)} \cdot \left( \theta_m^{(t,E+1)} - \theta^{(t)} \right)$ ;
- 14      $\theta^{(t+1)} \leftarrow \Pi_{\Theta} \left( \theta^{(t)} + \Delta^{(t)} \right)$ ;
- 15 **end**

---

**4.3.2 Federated Learning Meta-Algorithm for Data Streams**

When learning from a data stream, every client only has access to samples currently present in its local memory. Due to the limited storage capacity at each client and to the variability in the number of new samples arriving across time, samples may spend different amounts of time in memory and then be used a different number of times during training. In order to potentially compensate for such heterogeneity, we allow samples to be weighted differently over time and across clients. In particular, we denote by  $\lambda_m^{(t,j)} \geq 0$  the weight assigned at time  $t$  to sample  $j$  stored in client  $m$ 's memory (then  $j \in \mathcal{I}_m^{(t)}$ ), and by  $\lambda \triangleq \left\{ \lambda_m^{(t,j)} ; m \in [M], t \in [T], j \in \mathcal{I}_m^{(t)} \right\}$  the set of all weights. We define the weighted local objective associated to client- $m$ 's local memory at time step  $t \in [T]$  as

$$\mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta) \triangleq \frac{\sum_{j \in \mathcal{I}_m^{(t)}} \lambda_m^{(t,j)} \ell(\theta, \mathbf{z}_m^{(j)})}{\sum_{j \in \mathcal{I}_m^{(t)}} \lambda_m^{(t,j)}}, \quad (4.3)$$

and similarly the global weighted empirical risk as

$$\mathcal{L}_S^{(\lambda)}(\theta) \triangleq \frac{\sum_{m=1}^M \sum_{t=1}^T \sum_{j \in \mathcal{I}_m^{(t)}} \lambda_m^{(t,j)} \cdot \ell(\theta; \mathbf{z}_m^{(j)})}{\sum_{m=1}^M \sum_{t=1}^T \sum_{j \in \mathcal{I}_m^{(t)}} \lambda_m^{(t,j)}}. \quad (4.4)$$

We additionally define client- $m$ 's *aggregation weight* as

$$p_m^{(t)} \triangleq \frac{\sum_{j \in \mathcal{I}_m^{(t)}} \lambda_m^{(t,j)}}{\sum_{m'=1}^M \sum_{j \in \mathcal{I}_{m'}^{(t)}} \lambda_{m'}^{(t,j)}}, \quad (4.5)$$

and

$$q^{(t)} \triangleq \frac{\sum_{m=1}^M \sum_{j \in \mathcal{I}_m^{(t)}} \lambda_m^{(t,j)}}{\sum_{s=1}^T \sum_{m'=1}^M \sum_{j \in \mathcal{I}_{m'}^{(s)}} \lambda_{m'}^{(s,j)}}. \quad (4.6)$$

In this work we consider a meta-algorithm similar to vanilla FedAvg [McM+17] to minimize the weighted empirical risk (4.4). Algorithm 13 operates in an iterative fashion: at time step  $t \in [T]$  (also called communication round), the central server broadcasts the global model  $\theta^{(t)}$  to a subset of clients (line 4). Then every selected client, say it  $m$ , receives a new batch of data (line 5) that is used to update the client's local memory  $\mathcal{M}_m^{(t)}$  (line 6). The selected clients perform  $E$  local stochastic gradient steps (line 10), where the stochastic gradient  $\mathbf{g}_m^{(t,e)}$  is an unbiased estimator of  $\nabla \mathcal{L}^{(\lambda)}_{\mathcal{M}_m^{(t)}}(\theta_m^{(t,e)})$  computed using at most  $K$  samples (line 9). After  $E$  local steps, clients send back their models to the central server for aggregation (line 13, 14). The update at time step  $t$  can also be written as follows

$$\theta^{(t+1)} = \Pi_{\Theta} \left( \theta^{(t)} - \eta \cdot \sum_{m=1}^M p_m^{(t)} \sum_{e=1}^E \mathbf{g}_m^{(t,e)} \right), \quad (4.7)$$

where  $\Pi_{\Theta}(\cdot)$  denotes the projection over the set  $\Theta$ .

Note that the output of Algorithm 13 depends on the actual sample arrival sequences at clients, on the memory update rule, and on the weights  $\lambda$ . In particular, the memory update rule determines which samples can be considered at a given time step and then which weights can be different from zero. Nevertheless, for the sake of simplicity, we denote the output simply as  $A^{(\lambda)}(\mathcal{S})$ .

In this work, we restrict our analysis to the case where both the memory update rule and the weight selection rule are deterministic and do not depend on the features or the labels of the samples in the memory. More formally, given a particular instance of the counting process  $N_m^{(t)}$ , the weights  $\{\lambda_m^{(t,i)}\}_{t \in [T]}$  of sample  $\mathbf{z}_m^{(i)} \in \mathcal{S}_m$  remain unchanged if  $\mathbf{z}_m^{(i)} = (\mathbf{x}_m^{(i)}, y_m^{(i)})$  is replaced by  $\mathbf{z}_m^{(i)} = (\tilde{\mathbf{x}}_m^{(i)}, \tilde{y}_m^{(i)})$  with  $\tilde{\mathbf{x}}_m^{(i)} \neq \mathbf{x}_m^{(i)}$  or  $\tilde{y}_m^{(i)} \neq y_m^{(i)}$ .

For a given sample arrival sequence and memory update rule, the quality of the algorithm is evaluated through the *true error*

$$\epsilon_{\text{true}} \triangleq \mathbb{E}_{A^{(\lambda)}, \mathcal{S}} \left[ \mathcal{L}_{\mathcal{P}^{(\alpha)}} \left( A^{(\lambda)}(\mathcal{S}) \right) \right] - \min_{\theta \in \Theta} \mathcal{L}_{\mathcal{P}^{(\alpha)}}(\theta), \quad (4.8)$$

where the expectation is taken over the potential randomness of algorithm  $A^{(\lambda)}$ , i.e., clients' (line 2) and batches' (line 8) sampling processes, and the samples collected.

### 4.3.2.1 General Analysis

The true error  $\epsilon_{\text{true}}$  of our meta-algorithm in (4.8) can be bounded as follows (see proof in Appendix G.1.1)

$$\epsilon_{\text{true}} \leq \underbrace{\mathbb{E}_{\mathcal{S}, A^{(\lambda)}} \left[ \mathcal{L}_{\mathcal{S}}^{(\lambda)} \left( A^{(\lambda)} \left( \mathcal{S}^{(T)} \right) \right) - \min_{\theta \in \Theta} \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) \right]}_{\triangleq \epsilon_{\text{opt}}} + 2 \underbrace{\mathbb{E}_{\mathcal{S}} \left[ \sup_{\theta \in \Theta} \left| \mathcal{L}_{\mathcal{P}(\alpha)}(\theta) - \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) \right| \right]}_{\triangleq \epsilon_{\text{gen}}}. \quad (4.9)$$

The generalization error  $\epsilon_{\text{gen}}$  is the expected value of the *representativeness* of the dataset  $\mathcal{S}$ , which is the maximal distance between the true risk  $\mathcal{L}_{\mathcal{P}(\alpha)}$  and the empirical risk  $\mathcal{L}_{\mathcal{S}}^{(\lambda)}$ . Intuitively, the smaller the generalization error, the better we can approach the minimum of  $\mathcal{L}_{\mathcal{P}(\alpha)}$  by minimizing  $\mathcal{L}_{\mathcal{S}}^{(\lambda)}$ .

The optimization error  $\epsilon_{\text{opt}}$  measures how well Algorithm 13 approaches the minimizer of the weighted empirical risk  $\mathcal{L}_{\mathcal{S}}^{(\lambda)}$ .

In the rest of this section, we first provide bounds for the generalization error  $\epsilon_{\text{gen}}$  (Theorem 4.3.1) and for the optimization error  $\epsilon_{\text{opt}}$  (Theorem 4.3.3) and then combine them to bound the overall error  $\epsilon_{\text{true}}$  (Theorem 4.3.4). Our results rely on the following assumptions:

**Assumption 22.** (*Bounded loss*) The loss function is bounded, i.e.,  $\forall \theta \in \Theta, \mathbf{z} \in \mathcal{Z}, \ell(\theta; \mathbf{z}) \in [0, B]$ .

**Assumption 23.** (*Bounded domain*) We suppose that  $\Theta$  is convex, closed and bounded with diameter  $D$ .

**Assumption 24.** (*Convexity*) For all  $\mathbf{z} \in \mathcal{Z}$ , the function  $\theta \mapsto \ell(\theta; \mathbf{z})$  is convex on  $\mathbb{R}^d$ .

**Assumption 25.** (*Smoothness*) For all  $\mathbf{z} \in \mathcal{Z}$ , the function  $\theta \mapsto \ell(\theta; \mathbf{z})$  is  $L$ -smooth on  $\mathbb{R}^d$ .

Assumption 18 is a standard assumption in statistical learning theory (e.g., [MRT18] and [SB14]). Assumptions 23–25 are common assumptions in the analysis of (stochastic) gradient methods (see for example [Bub15] and [BCN18]) and online convex optimization [Haz19].

**Remark 6.** Assumptions 22 and 25 imply that (it follows from Lemma G.2 in Appendix G.1.2)

$$\sigma_0^2 \triangleq \max_m \mathbb{E}_{\mathbf{z} \sim \mathcal{P}_m} \left[ \sup_{\theta \in \Theta} \|\nabla \ell(\theta; \mathbf{z}) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta)\|^2 \right] \leq \left( 2 \cdot \sqrt{2LB} \right)^2, \quad (4.10)$$

and (it follows from Lemma G.3 in Appendix G.1.2)

$$\zeta \triangleq \max_{m, m'} \sup_{\theta \in \Theta} \left\| \nabla \mathcal{L}_{\mathcal{P}_{m'}}(\theta) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) \right\| \leq 2 \cdot \sqrt{2LB}. \quad (4.11)$$

These properties are similar to the stochastic gradients' bounded variance, and the clients' bounded dissimilarity assumptions usually employed in the analysis of federated learning algorithms [Wan+21a].



### 4.3.2.2 Bounding the Generalization Error

Theorem 4.3.1 (proof in Appendix G.1.3) quantifies the generalization error and in particular how the weighted empirical risk  $\mathcal{L}_S^{(\lambda)}$  differs from the target expected risk  $\mathcal{L}_{\mathcal{P}^{(\alpha)}}$  for the minimizer of the first one, i.e., it bounds  $|\mathcal{L}_{\mathcal{P}^{(\alpha)}}(\theta') - \mathcal{L}_S^{(\lambda)}(\theta')|$  for  $\theta' \in \arg \min_{\theta \in \Theta} \mathcal{L}_S^{(\lambda)}(\theta)$ . The bound differs from classic statistical learning results (as those in [SB14]) because  $\mathcal{L}_S^{(\lambda)}$  is a weighted empirical risk and its expected value does not necessarily coincide with  $\mathcal{L}_{\mathcal{P}^{(\alpha)}}$ . We recall that the label discrepancy associated to a hypothesis class  $\mathcal{H}$  quantifies the distance between two distributions  $\mathcal{P}$  and  $\mathcal{P}'$  as follows  $\text{disc}_{\mathcal{H}}(\mathcal{P}, \mathcal{P}') \triangleq \max_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{P}}(h) - \mathcal{L}_{\mathcal{P}'}(h)|$  [Man+20].

**Theorem 4.3.1.** *Suppose that Assumption 22 holds, and that  $1 < \text{Pdim}(\ell \circ \mathcal{H}) < N$ . When using Algorithm 13 with weights  $\lambda$ , it follows that*

$$\epsilon_{\text{gen}} \leq \text{disc}_{\mathcal{H}}(\mathcal{P}^{(\alpha)}, \mathcal{P}^{(\mathbf{p})}) + \tilde{O} \left( \sqrt{\frac{\text{Pdim}(\ell \circ \mathcal{H})}{N_{\text{eff}}}} \right), \quad (4.12)$$

where  $N_{\text{eff}} = \left( \sum_{m=1}^M \sum_{i=1}^{N_m} p_{m,i}^2 \right)^{-1}$ ,

$$p_{m,i} = \frac{\sum_{t=1}^T \sum_{j \in \mathcal{I}_m^{(t)}} \mathbb{1}\{j = i\} \cdot \lambda_m^{(t,j)}}{\sum_{m'=1}^M \sum_{t=1}^T \sum_{j \in \mathcal{I}_{m'}^{(t)}} \lambda_{m'}^{(t,j)}}, \quad i \in [N_m], \quad (4.13)$$

and  $\mathbf{p} = \left( \sum_{i=1}^{N_m} p_{m,i} \right)_{1 \leq m \leq M}$ .

The coefficient  $p_{m,i}$  represents the *relative importance* given, during the whole training period, to sample  $i$  with respect to all the samples collected by all clients and  $p_m = \sum_{i=1}^{N_m} p_{m,i}$  represents the relative importance given to client  $m$  during training. Note that  $p_m = \sum_{t=1}^T q^{(t)} p_m^{(t)}$  and the  $p_m^{(t)}$  coincides with the relative importance  $p_m$ , when  $p_m^{(t)}$  is constant over time.

In general, there is an inconsistency between the importance we should give to clients (quantified by  $\alpha$  in (4.1)) and the one we actually give them during training (quantified by  $\mathbf{p}$ ). The first term on the RHS of (4.12) captures the mismatch between the target distribution  $\mathcal{P}^{(\alpha)}$  and the “*effective distribution*”  $\mathcal{P}^{(\mathbf{p})} = \sum_{m=1}^M p_m \mathcal{P}_m$  through the discrepancy.

The second term in the RHS of (4.12) is similar in shape to the usual bounds observed in statistical learning theory, e.g., [SB14], which are proportional to the square root of the ratio of the VC dimension of the hypotheses class and the total number of samples  $N$ .

In our case,  $N_{\text{eff}}$  plays the role of the *effective number of samples* and Lemma 4.3.2 (proof in Appendix G.2) shows that, as expected,  $N_{\text{eff}}$  is at most  $N$ , and reaches this value when each sample is given the same importance.

**Lemma 4.3.2.** *It holds  $N_{\text{eff}} \leq N$  and the bound is attained when each sample has the same relative importance, i.e.,  $p_{m,i} = p_{m,j}$ , for each  $i, j \in [N_m]$ .*

The generalization error  $\epsilon_{\text{gen}}$  decreases the closer  $\alpha$  and  $\mathbf{p}$  are and the larger  $N_{\text{eff}}$  is. When  $\alpha_m = n_m$  (remember that  $n_m = N_m/N$ ), the choice  $p_{m,i} = 1/N$  minimizes the bound, as it leads both to  $\mathbf{p} = \mathbf{n} = \alpha$  and to  $N_{\text{eff}} = N$ .

In our streaming learning setting,  $p_{m,i} = 1/N$  can be obtained by different combinations of memory update rules and sample weight selection rules. For example, this is the

case when clients' memories only contain the samples received during the current round (i.e.,  $\text{Update}(\mathcal{M}_m^{(t-1)}, \mathcal{B}_m^{(t)}) = \mathcal{B}_m^{(t)}$  in line 6 of Alg. 13) and all samples currently in the memory get weight 1 (i.e.,  $\lambda_m^{(t,j)} = 1$  for each  $j \in \mathcal{I}_m^{(t)}$ ). But it is also the case when the memory update rule lets samples stay in memory for multiple consecutive rounds (e.g.,  $\tau_m^{(j)}$  rounds for sample  $j$  at client  $m$ ) and samples receive a weight inversely proportional to the number of consecutive rounds (i.e.,  $\lambda_m^{(t,j)} = 1/\tau_m^{(j)}$ ). In what follows, we refer to any combination of memory update rules and weight selection rules leading to  $p_{m,i} = 1/N$  as a `Uniform` strategy.

While a `Uniform` strategy minimizes the bound for the generalization error  $\epsilon_{\text{gen}}$  when  $\alpha = \mathbf{n}$ , it is in general suboptimal in terms of the optimization error  $\epsilon_{\text{opt}}$ , as we are going to show in the next section.

### 4.3.2.3 Bounding the Optimization Error

We provide our bound on  $\epsilon_{\text{opt}}$  under full clients participation ( $\mathbb{S}^{(t)} = [M]$ ) with full batch ( $K \geq |\mathcal{I}_m^{(t)}|$ ). Under mini-batch gradients an additional vanishing error term appears. The proof is provided in Appendix G.2.1.

**Theorem 4.3.3.** *Suppose that Assumptions 22–25 hold, the sequence  $(q^{(t)})_t$  is non increasing, and verifies  $q^{(1)} = \mathcal{O}(1/T)$ , and  $\eta \propto 1/\sqrt{T} \cdot \min\{1, 1/\bar{\sigma}(\lambda)\}$ . Under full clients participation ( $\mathbb{S}^{(t)} = [M]$ ) with full batch ( $K \geq |\mathcal{I}_m^{(t)}|$ ), we have*

$$\epsilon_{\text{opt}} \leq \mathcal{O}(\bar{\sigma}(\lambda)) + \mathcal{O}\left(\frac{\bar{\sigma}(\lambda)}{\sqrt{T}}\right) + \mathcal{O}\left(\frac{1}{\sqrt{T}}\right), \quad (4.14)$$

where,

$$\bar{\sigma}^2(\lambda) \triangleq \sum_{t=1}^T q^{(t)} \times \mathbb{E}_{\mathcal{S}} \left[ \sup_{\theta \in \Theta} \left\| \nabla \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) - \sum_{m=1}^M p_m^{(t)} \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta) \right\|^2 \right]. \quad (4.15)$$

Moreover, there exist a data arrival process and a loss function  $\ell$ , such that, under FIFO memory update rule,\* for any choice of weights  $\lambda$ ,  $\epsilon_{\text{opt}} = \Omega(\bar{\sigma}(\lambda))$ .

The coefficient  $\bar{\sigma}^2(\lambda)$  quantifies the variability of the gradient considered in the update at round  $t$  w.r.t. the gradient of the global objective  $\mathcal{L}_{\mathcal{S}}^{(\lambda)}$  and, as shown by Theorem 4.3.3, it prevents the optimization error to vanish when  $T$  diverges. Lemma G.5 provides a general upper bound for  $\bar{\sigma}^2(\lambda)$  in terms of stochastic gradients' variance and clients' dissimilarity.

The optimization error  $\epsilon_{\text{opt}}$  is smaller the closer  $\bar{\sigma}^2(\lambda)$  is to zero. In our streaming learning setting,  $\bar{\sigma}^2(\lambda) = 0$  may be obtained if the memory is never updated ( $\text{Update}(\mathcal{M}_m^{(t-1)}, \mathcal{B}_m^{(t)}) = \mathcal{M}_m^{(t-1)}, \forall t \geq 1$ ) and the aggregation weights are constant over time ( $p_m^{(t)} = p_m, \forall t \in [T]$ ). It is indeed easy to check that under these conditions  $\mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) = \sum_{m=1}^M p_m^{(t)} \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta)$  (and they equal  $\sum_{m=1}^M p_m \mathcal{L}_{\mathcal{M}_m^{(0)}}^{(\lambda)}(\theta)$ ). Any set of time-independent sample weights leads to constant aggregation weights, but, among them, the choice  $\lambda_m^{(t,j)} = 1$  reduces the generalization bound  $\epsilon_{\text{gen}}$ . We refer to these memory update and weight selection rules as the `Historical` strategy.

The `Historical` strategy minimizes the optimization bound by ignoring all the samples collected during training. It is in sharp contrast with the `Uniform` strategy, which assigns the same relative importance to all collected samples.

\*The FIFO (First-In-First-Out) update rule evicts the oldest samples in the memory to store the most recent ones.

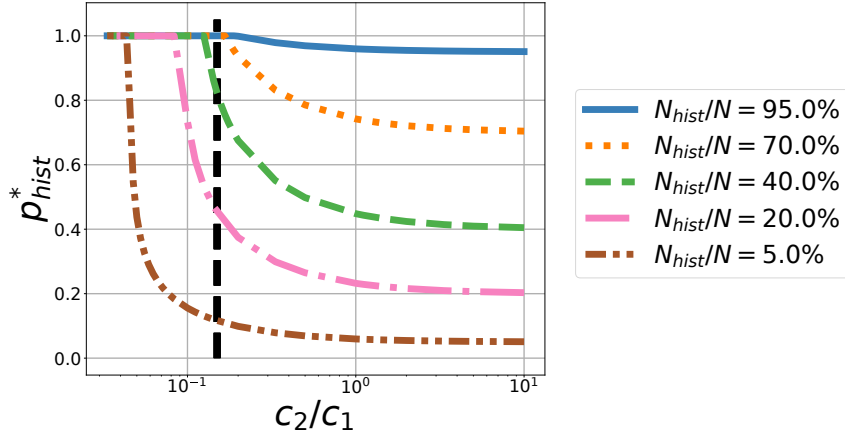


Figure 4.2: Effect of  $c_2/c_1$  on the historical clients relative importance  $p_{hist}^*$  for different values of  $N_{hist}/N$ , when  $M = 50$  and  $M_{hist} = 25$ . The dashed vertical line corresponds to our estimation of  $c_2/c_1$  on CIFAR-10 experiments ( $\hat{c}_2/\hat{c}_1 = 0.15$ ).

#### 4.3.2.4 Main Result

The tension between the two error components  $\epsilon_{gen}$  and  $\epsilon_{opt}$  is evident from our discussion above. One can minimize  $\epsilon_{gen}$  by considering at each time only the most recent samples, and, at the opposite,  $\epsilon_{opt}$  by ignoring those samples. By combining Theorems 4.3.1 and 4.3.3, Theorem 4.3.4 formally quantifies this trade-off and provides a bound on  $\epsilon_{true}$ .

**Theorem 4.3.4.** *Under the same assumptions as in Theorem 4.3.1 and Theorem 4.3.3,*

$$\epsilon_{true} \leq \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) + \mathcal{O}(\bar{\sigma}(\lambda)) + 2\text{disc}_{\mathcal{H}}(\mathcal{P}^{(\alpha)}, \mathcal{P}^{(\mathbf{p})}) + \tilde{\mathcal{O}}\left(\sqrt{\frac{\text{Pdim}(\ell \circ \mathcal{H})}{N_{\text{eff}}}}\right). \quad (4.16)$$

### 4.3.3 Case Study

In fog computing environments, IoT devices, edge servers, and cloud servers can jointly participate to train an ML model [Bon+12]. IoT devices keep generating new data, but may not be able to store them permanently due to sever memory constraints. Instead, edge servers may contribute with larger static datasets [Hos+20b; Wan+21b]. Motivated by this scenario, we consider two groups of clients:  $M_{hist}$  clients with “historical” datasets, which do not change during training, and  $M - M_{hist}$  clients, who collect “fresh” samples with constant rates  $\{b_m > 0, m \in \llbracket M_{hist} + 1, M \rrbracket\}$  and only store the most recent  $b_m$  samples due to memory constraints (i.e.,  $C_m = b_m$ ).<sup>\*</sup> We refer to these two categories as historical clients and fresh clients, respectively. Fresh clients can also capture the setting where clients are available during a single communication round.

At each client all samples are used the same number of times ( $T$  and 1 at historical and fresh clients, respectively). Then, one can prove that each client, say it  $m$ , should assign the same weight to any sample currently available at its local memory, i.e.,  $\lambda_m^{(t,j)} = \lambda_m^{(t)}$ . For simplicity, we consider stationary weights, i.e.,  $\lambda_m^{(t)} = \lambda_m$ , and we want then to determine per-client sample weights

<sup>\*</sup>Note that we are implicitly selecting FIFO as memory update rule.

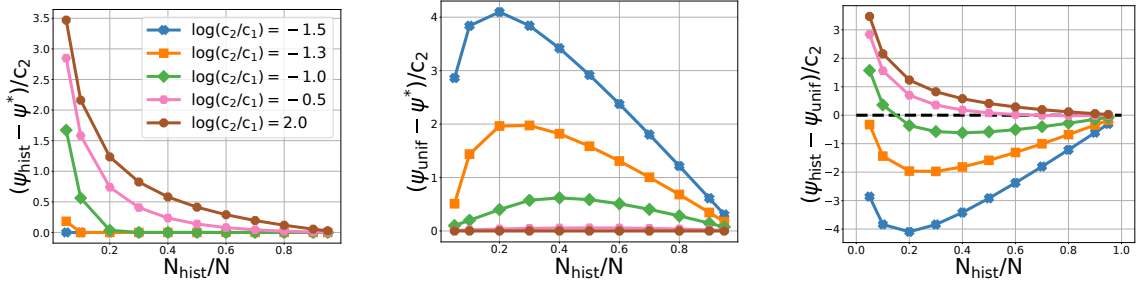


Figure 4.3: The differences  $\psi_{\text{hist}} - \psi^*$  (left),  $\psi_{\text{unif}} - \psi^*$  (center), and  $\psi_{\text{hist}} - \psi_{\text{unif}}$  (right) as a function of  $N_{\text{hist}}/N$  for different values of  $c_2/c_1$ , on CIFAR-10 dataset ( $N = 5 \times 10^5$ ) when  $M = 50$  and  $M_{\text{hist}} = 25$ .

$(\lambda_m)_{m \in [M]}$  leading to the best guarantees in terms of  $\epsilon_{\text{true}}$ .<sup>\*</sup> Equivalently, we want to determine the clients' relative importance values  $\mathbf{p} = (p_m)_{m \in [M]}$ , where  $p_m = \lambda_m N_m / \left( \sum_{m'=1}^M \lambda_{m'} N_{m'} \right)$ . Note that in this setting aggregation weights and relative importance values coincide (i.e.,  $p_m^{(t)} = p_m$ ). Corollary 4.3.5' (Appendix G.4) bounds  $\epsilon_{\text{true}}$  as a function of  $\mathbf{p}$  in this scenario. For the sake of simplicity, we provide here the bound for the case  $\alpha_m = n_m, m \in [M]$  (which we assume to hold in the rest of this section):

**Corollary 4.3.5.** *Consider the scenario with  $M_{\text{hist}}$  historical clients, and  $M - M_{\text{hist}}$  fresh clients. Suppose that the same assumptions of Theorem 4.3.4 hold, that  $\alpha = \mathbf{n}$ , and that Algorithm 13 is used with clients' aggregation weights  $\mathbf{p} = (p_m)_{m \in [M]} \in \Delta^{M-1}$ , then*

$$\epsilon_{\text{true}} \leq \psi(\mathbf{p}; \mathbf{c}) \triangleq c_0 + c_1 \cdot \sqrt{\sum_{m=M_{\text{hist}}+1}^M p_m^2} + c_2 \cdot \sqrt{\sum_{m=1}^M \frac{p_m^2}{n_m}}, \quad (4.17)$$

where  $\mathbf{c} = (c_0, c_1, c_2)$  are non-negative constants not depending on  $\mathbf{p}$ , given as:

$$c_0 = (C_1 + C_3) + \frac{C_2}{T} - 2 \cdot \max_{m, m'} \text{disc}(\mathcal{P}_m, \mathcal{P}_{m'}) \quad (4.18)$$

$$c_1 = \sigma_0 \sqrt{M - M_0} \cdot \left( D + \frac{2}{\sqrt{T}} \right) \quad (4.19)$$

$$c_2 = 10B \sqrt{1 + \log \left( \frac{N}{\text{Pdim}(\ell \circ \mathcal{H})} \right)} \sqrt{\frac{\text{Pdim}(\ell \circ \mathcal{H})}{N}} + 2 \cdot \max_{m, m'} \text{disc}(\mathcal{P}_m, \mathcal{P}_{m'}) \quad (4.20)$$

and  $C_1, C_2$ , and  $C_3$  are the constants defined in the proof of Theorem 4.3.3, and  $\sigma_0$  is defined in Remark 6.

The second term in (4.17) captures the gradient variability (second term in (4.16)), while the third term in (4.17) captures both contributions to the generalization error, i.e., the distribution discrepancy and the effective number of samples (third and fourth terms in (4.17)). In particular, it holds  $\sum_{m=1}^M \frac{p_m^2}{n_m} \propto 1/N_{\text{eff}}$ .

<sup>\*</sup>Restricting the weights to be stationary, i.e.,  $\lambda_m^{(t)} = \lambda_m$ , might be suboptimal.

Table 4.1: Average test accuracy across clients for different datasets in the settings when  $N_{\text{hist}}/N = 50\%$ .

DATASET	$D$	$G$	$B$	$d$
SYNTHETIC	1.9	0.4	0.7	21
CIFAR-10	1.0	5.5	2.3	3, 353, 034
CIFAR-100	1.0	4.7	4.6	3, 537, 444
FEMNIST	5.9	12.9	3.5	867, 390
SHAKESPEARE	2.6	1.4	6.1	226, 180

The minimization of  $\psi$  over the unitary simplex is a convex optimization problem (proof in Appendix G.4.4), which can then be solved efficiently with, for example, projected gradient descent. We use  $\psi^*$ ,  $\mathbf{p}^*$ , and  $p_{\text{hist}}^*$  to denote the minimum of  $\psi$ , its minimizer, and the aggregate relative importance given to historical clients ( $p_{\text{hist}}^* \triangleq \sum_{m=1}^{M_{\text{hist}}} p_m^*$ ), respectively.

The solution  $\mathbf{p}^*$  depends on the value of  $\mathbf{n}$ —in particular on the fraction of historical samples  $N_{\text{hist}}/N$  (where  $N_{\text{hist}} \triangleq \sum_{m=1}^{M_{\text{hist}}} N_m$ )—and on the ratio  $c_2/c_1$ . The ratio  $c_2/c_1$  only depends on the intrinsic properties of the learning problem ( $\text{Pdim}(\ell \circ \mathcal{H})$ ,  $D$ ,  $B$ , and  $\sigma_0$ ), and the total number of samples  $N$  (see Appendix G.4.3).

Figure 4.2 illustrates how the optimal clients’ importance values change as a function of the ratio  $c_2/c_1$  and the fraction of historical samples  $N_{\text{hist}}/N$  (other results are in Figure G.22). Beside the specific numerical values, one can distinguish two corner cases. When  $c_2/c_1 \gg 1$ , the optimal solution corresponds to minimize  $\sum_{m=1}^M p_m^2/n_m$ , i.e., to maximize the effective number of samples. The optimal strategy is then the `Uniform` one and the aggregate relative importance for historical clients is  $p_{\text{hist}}^* = N_{\text{hist}}/N$ . On the contrary, when  $c_2/c_1 \ll 1$ , the optimal solution corresponds to minimize  $\sum_{m>M_{\text{hist}}} p_m^2$ , i.e., the gradient variability. The `Historical` strategy is then optimal and corresponds to  $p_m^* = N_m/N_{\text{hist}} = \frac{N}{N_{\text{hist}}} n_m$  for  $m \in [M_{\text{hist}}]$  and  $p_{\text{hist}}^* = 1$ .

For general values of  $c_2/c_1$ , the optimal strategy to assign clients’ importance values—or equivalently sample weights—differs from both the `Uniform` and the `Historical` ones. We propose then the following heuristic, which we evaluate in the next section. At the beginning of training, clients cooperatively estimate  $c_2/c_1$  using a fraction of their historical samples, as  $\hat{c}_2/\hat{c}_1 \approx \frac{B+\sqrt{d/N}}{GD\sqrt{M-M_{\text{hist}}}}$  (see details in Appendix G.4.6). Then, clients’ importance values are selected minimizing the bound in (4.17), i.e.,  $\hat{\mathbf{p}}^* = \arg \min \psi(\cdot, \hat{\mathbf{c}})$ .

Beside providing configuration rules for our meta-algorithm, our analysis allows us also to evaluate how the performances of different strategies like `Uniform` and `Historical` depend on the different parameters as in Figure 4.3. Our experimental results in the next section confirm these theoretical predictions.

## 4.3.4 Numerical Experiments

### 4.3.4.1 Datasets and Models

In this section, we provide detailed description of the datasets and models used in our experiments. We considered five federated benchmark datasets with different machine learning tasks: image classification (CIFAR10 and CIFAR100 [Kri09]), handwritten character recognition (FEM-

NIST [Cal+19]), and language modeling (Shakespeare [Cal+19; McM+17]), as well as a synthetic dataset described in Appendix 4.3.4.1. For Shakespeare and FEMNIST datasets there is a natural way to partition data through clients (by character and by writer, respectively). We relied on common approaches in the literature to sample heterogeneous local datasets from CIFAR-10 and CIFAR-100. Below, we give a detailed description of the datasets and the models / tasks considered for each of them.

**Synthetic Dataset.** Our synthetic dataset has been generated as follows:

1. Sample  $\theta_0 \in \mathbb{R}^d \sim \mathcal{N}(0, I_d)$ , from the multivariate normal distribution of dimension  $d$ , with zero mean and unitary variance
2. Sample  $\theta_m \in \mathbb{R}^d \sim \mathcal{N}(\theta_0, \varepsilon^2 I_d)$ ,  $m \in [M]$  from from the multivariate normal distribution of dimension  $d$ , centered around  $\theta_0$  and variance equal to  $\varepsilon^2$
3. For  $m \in [M]$  and  $i \in [N_m]$ , sample  $\mathbf{x}_m^{(i)} \sim \mathcal{U}([-1, 1]^d)$  from a uniform distribution over  $[-1, 1]^d$
4. For  $m \in [M]$  and  $i \in [N_m]$ , sample  $y_m^{(i)} \sim \mathcal{B}(\text{sigmoid}(\langle \mathbf{x}_m^{(i)}, \theta_m \rangle))$ , where  $\mathcal{B}$  is the standard Bernoulli distribution

**CIFAR-10 / CIFAR-100** We created federated versions of CIFAR-10 by distributing samples with the same label across the clients according to a symmetric Dirichlet distribution with parameter 0.4, as in [Wan+20a]. For CIFAR100, we exploited the availability of “coarse” and “fine” labels, using a two-stage Pachinko allocation method [LM06] to distribute the samples across the clients, as in [Red+21]. We train a shallow convolutional neural network for CIFAR-10/100 datasets.

**FEMNIST.** FEMNIST (Federated Extended MNIST) is a 62-class image classification dataset built by partitioning the data of Extended MNIST based on the writer of the digits/characters. We train two-layer fully connected neural network for FEMNIST dataset

**Shakespeare.** Shakespeare is a language modeling dataset built from the collective works of William Shakespeare. In this dataset, each client corresponds to a speaking role with at least two lines. The task is next character prediction. We use an RNN that first takes a series of characters as input and embeds each of them into a learned 8-dimensional space. The embedded characters are then passed through 2 RNN layers, each with 256 nodes, followed by a densely connected softmax output layer. We split the lines of each speaking role into into sequences of 80 characters, padding if necessary.

#### 4.3.4.2 Training Details.

In all experiments, the learning rate was tuned via grid search on the grid  $\{10^{-3.5}, 10^{-3}, 10^{-2.5}, 10^{-2}, 10^{-1.5}, 10^{-1}\}$  using the validation set. Once the learning rate had been selected, we retrained the models on the concatenation of the training and validation sets. Each experiment was repeated for three different seeds for the random number generator; we report the mean value and the 95% confidence bound.

Table 4.2: Datasets and models.

DATASET	CLIENTS	TOTAL SAMPLES	MODEL
SYNTHETIC	11	200	LINEAR MODEL
CIFAR-10 / 100	50	50,000	2 CNN + 2 FC
FEMNIST	3,597	817,851	2 FC
SHAKESPEARE	916	3,436,096	STACKED-LSTM

#### 4.3.4.3 Arrival Process

For CIFAR-10/100 datasets, we consider an arrival process with  $M_{\text{hist}} = 25$  clients with “historical” datasets, which do not change during training, and  $M - M_{\text{hist}} = 25$  clients, who collect “fresh” samples with constant rates  $\{b_m > 0, m \in \llbracket M_{\text{hist}} + 1, M \rrbracket\}$  and only store the most recent  $b_m$  samples due to memory constraints (i.e.,  $C_m = b_m$ ). For a given value of  $N_{\text{hist}}/N$ , we split the train part of the original CIFAR-10/100 into two groups, historical and fresh, with  $N_{\text{hist}}$  and  $N - N_{\text{hist}}$  samples, respectively. We then distribute the samples from the historical (resp. fresh) group across  $M_{\text{hist}}$  historical (resp.  $M - M_{\text{hist}}$  fresh) clients. A symmetric Dirichlet distribution is employed in the case of CIFAR-10, and a Pachinko allocation method is employed in the case of CIFAR-100.

Shakespeare and FEMNIST datasets have a natural partition across clients—by character and by writer, respectively. In our experiments, we split the natural clients of FEMNIST and Shakespeare into two groups, historical and fresh, with  $M_{\text{hist}}$  and  $M - M_{\text{hist}}$  clients, respectively. The historical clients participate to every communication round, while each fresh client is only available in a single communication round in the case of FEMNIST and for at most two consecutive communication rounds for Shakespeare dataset.

#### 4.3.4.4 Numerical Values for $\hat{c}_2/\hat{c}_1$

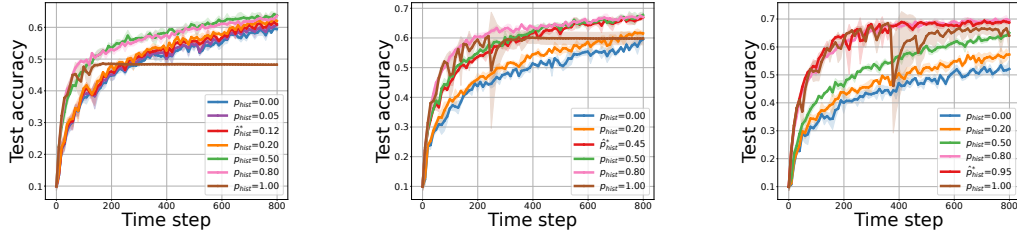
Table 4.1 provide the values of  $D$ ,  $G$ ,  $B$ , and  $d$  used to estimate the ratio  $\hat{c}_2/\hat{c}_1$ .

#### 4.3.4.5 Benchmarks

We compared our strategy to select clients’ importance values, (see Section 4.3.3), with three baselines: the `Uniform` and `Historical` strategies described above as well as the `Fresh` strategy which only considers fresh clients. We observe that under our samples’ arrival process and  $\alpha = \mathbf{n}$ , there could be two natural ways to extend the classic `FedAvg`’s aggregation rule [McM+17]: set each client’s aggregation weight proportional to (1) the number of samples collected by the client over the whole time-horizon, or (2) the number of samples currently in the client’s memory. The first aggregation rule coincides with the `Uniform` strategy, the second one leads in all settings we considered to very small aggregation weights for fresh clients so that it is practically indistinguishable from the `Historical` strategy. Interestingly, both these rules are in general suboptimal, motivating the practical interest of our study and of the strategy we propose.

Table 4.3: Average test accuracy across clients for different datasets in the settings when  $N_{\text{hist}}/N = 20\%$ .

DATASET	$\hat{c}_2/\hat{c}_1$ $\hat{p}_{\text{HIST}}^*$		TEST ACCURACY				
			FRESH	HISTORICAL	UNIFORM	OURS	OPTIMAL
SYNTHETIC	0.092	0.20	$84.7 \pm 1.44$	$77.3 \pm 3.15$	<b><math>85.5 \pm 1.60</math></b>	<b><math>85.5 \pm 1.60</math></b>	$85.5 \pm 1.60$
CIFAR-10	0.150	0.45	$59.6 \pm 0.94$	$59.8 \pm 2.16$	$61.5 \pm 0.63$	<b><math>66.9 \pm 0.81</math></b>	$67.7 \pm 0.91$
CIFAR-100	0.284	0.32	$22.4 \pm 0.57$	$22.6 \pm 0.50$	$25.3 \pm 0.43$	<b><math>28.5 \pm 0.57</math></b>	$31.5 \pm 0.25$
FEMNIST	0.001	1.00	$53.3 \pm 1.85$	<b><math>66.1 \pm 0.20</math></b>	$55.4 \pm 0.80$	<b><math>66.1 \pm 0.20</math></b>	$66.1 \pm 0.80$
SHAKESPEARE	0.064	1.00	$38.4 \pm 0.43$	<b><math>49.0 \pm 0.26</math></b>	$39.3 \pm 0.38$	<b><math>49.0 \pm 0.26</math></b>	$49.0 \pm 0.26$

Figure 4.4: Evolution of the test accuracy when using different values of  $p_{\text{hist}}$  for CIFAR-10 (left) dataset, when  $N_{\text{hist}}/N = 5\%$  (left),  $20\%$  (center), and  $50\%$  (right). The setting  $p_{\text{hist}} = N_{\text{hist}}/N$  corresponds to Uniform strategy.

#### 4.3.4.6 Main Results

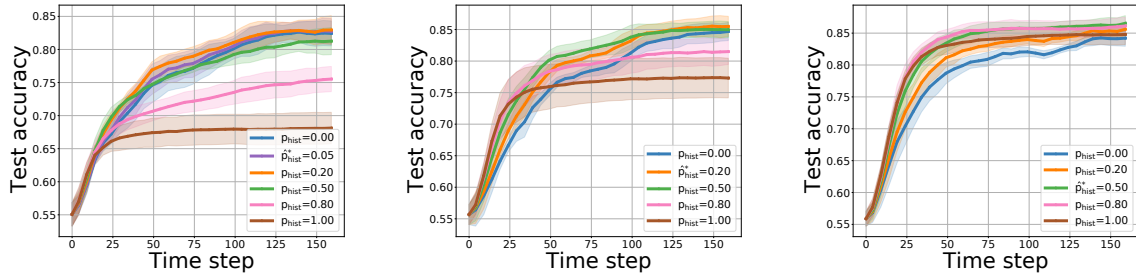
Table 4.3 reports the test accuracy when  $N_{\text{hist}}/N = 20\%$  for the different strategies together with the optimal test accuracy obtained selecting the value of  $p_{\text{hist}} = \sum_{m=1}^{M_{\text{hist}}} p_m$  in the grid  $\{0, 0.2, 0.5, 0.8, 1.0\}$ . Our observations are confirmed for other values of  $N_{\text{hist}}/N$  (see Table 4.4 and Table 4.5). A first remark is that working only with new data (as `FRESH` does) is never optimal, not even when historical data account for just 5% of the total dataset (Table 4.4). Second, neither of the two “reasonable” ways to extend FedAvg consistently achieves good accuracy: `HISTORICAL` performs poorly over Synthetic and `UNIFORM` over FEMNIST and Shakespeare. On the contrary, our method always performs at least as well as the best baseline and it often achieves a test accuracy similar to the (estimated) optimal one. In particular, it correctly sets weights as `UNIFORM` over Synthetic and as `HISTORICAL` over FEMNIST and Shakespeare. We observe that our analysis also helps to explain the counter-intuitive conclusion that, on FEMNIST and Shakespeare, it is beneficial to ignore new collected samples (even for  $N_{\text{hist}}/N = 5\%$ , see Table 4.4). Our strategy correctly sets  $\hat{p}_{\text{hist}}^* = 1$ , because it estimates that, for these two datasets, the ratio of the number of parameters to the aggregate training dataset size ( $d/N$ ) is much smaller than the gradients’ norm ( $G$ )—numerical values are provided in Appendix 4.3.4.4. This information suggests that we can use a small subset of the original dataset to identify a good model in the selected hypotheses class, and in particular we can rely only on historical data avoiding the potential noise introduced by new samples.

Figure 4.4 shows the effect of  $p$  on CIFAR-10 test accuracy for different values of the ratio  $N_{\text{hist}}/N$ —similar figures for other datasets are provided in Figures 4.5—4.8. It confirms that



Table 4.4: Average test accuracy across clients for different datasets in the settings when  $N_{\text{hist}}/N = 5\%$ .

DATASET	$\hat{c}_2/\hat{c}_1$ $p_{\text{HIST}}^*$		TEST ACCURACY				
			FRESH	HISTORICAL	UNIFORM	OURS	OPTIMAL
SYNTHETIC	0.094	0.06	$82.4 \pm 1.89$	$68.1 \pm 2.39$	<b><math>82.7 \pm 1.94</math></b>	<b><math>82.7 \pm 1.90</math></b>	$82.9 \pm 2.17$
CIFAR-10	0.150	0.12	$59.5 \pm 0.77$	$48.2 \pm 0.21$	$60.7 \pm 0.58$	<b><math>61.0 \pm 0.42</math></b>	$63.7 \pm 0.57$
CIFAR-100	0.284	0.08	$23.5 \pm 0.65$	$13.5 \pm 0.41$	$24.4 \pm 0.54$	<b><math>25.2 \pm 0.66</math></b>	$27.8 \pm 0.39$
FEMNIST	0.001	1.00	$55.2 \pm 1.79$	<b><math>65.7 \pm 0.09</math></b>	$58.4 \pm 1.80$	<b><math>65.7 \pm 0.09</math></b>	$65.7 \pm 0.09$
SHAKESPEARE	0.064	1.00	$40.2 \pm 0.34$	<b><math>49.0 \pm 0.06</math></b>	$41.0 \pm 1.33$	<b><math>49.0 \pm 0.06</math></b>	$49.0 \pm 0.06$

Figure 4.5: Evolution of the test accuracy when using different values of  $p_{\text{hist}}$  for the synthetic dataset, when  $N_{\text{hist}}/N = 5\%$  (left), 20% (center), and 50% (right).

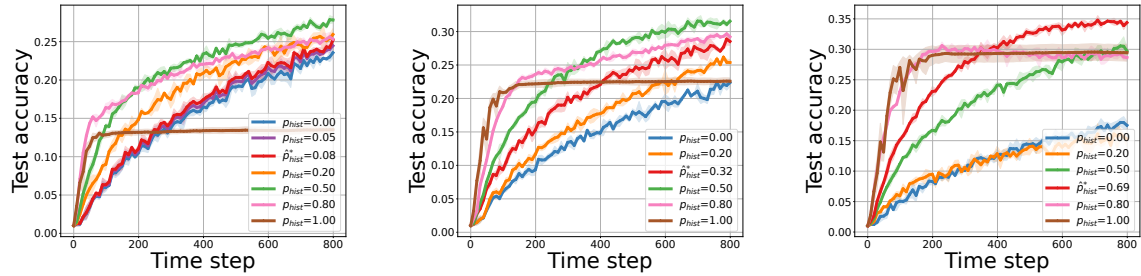
performances in terms of final test accuracy match the predictions of our model on the bound  $\psi$  illustrated in Figure 4.3. First, Figure 4.4 shows that the performance gap between Historical and the optimal assignment  $\mathbf{p}^*$  decreases when  $N_{\text{hist}}/N$  increases (as predicted in Figure 4.3 (left)): the gap is  $15.5 \pm 0.30$ ,  $7.9 \pm 1.17$ , and  $5.3 \pm 2.8$  pp when  $N_{\text{hist}}/N$  is 5%, 20%, and 50%, respectively. Second, Figure 4.4 confirms that the performance gap between Uniform and the optimal assignment first increases and then decreases, when  $N_{\text{hist}}/N$  increases (as in Figure 4.3 (center)): the gap is  $3.0 \pm 0.57$ ,  $6.2 \pm 0.55$ , and  $4.3 \pm 0.35$  pp when  $N_{\text{hist}}/N$  is 5%, 20%, and 50%, respectively. Finally, Figure 4.4 shows that the relative ranking of Uniform and Historical changes, with Uniform being a better option for smaller values of  $N_{\text{hist}}/N$  and Historical becoming slightly better for larger values. Again, this behavior is predicted by our analysis. Indeed, in this experiment, our estimation for the ratio  $c_2/c_1$  is  $\hat{c}_2/\hat{c}_1 \approx 0.15 \in [10^{-1.3}, 10^{-0.5}]$  corresponding to a setting for which  $\psi_{\text{hist}} - \psi_{\text{unif}}$  changes sign in Figure 4.3 (right).

#### 4.3.4.7 Effect of the optimization algorithm

We experimentally evaluated the performance of our heuristic when the federated optimization algorithm is SCAFFOLD and FedProx for CIFAR-10 dataset ( $N_{\text{hist}}/N = 20\%$ ). While SCAFFOLD and FedProx provide some performance improvement, they do not alter the relative performance of the aggregation strategies and our heuristic is still the best one. FedProx with penalization parameter 0.1 (/SCAFFOLD) achieves a test accuracy of 59.6% (/60.1%), 59.8% (/59.8%), 61.6% (/62.6%), and 67.1% (/67.4%) for Fresh, Historical, Uniform, and Ours, respectively.

Table 4.5: Average test accuracy across clients for different datasets in the settings when  $N_{\text{hist}}/N = 50\%$ .

DATASET	$\hat{c}_2/\hat{c}_1$ $p_{\text{HIST}}$		TEST ACCURACY				
			FRESH	HISTORICAL	UNIFORM	OURS	OPTIMAL
SYNTHETIC	0.085	0.50	$84.2 \pm 1.27$	$84.8 \pm 1.58$	<b><math>86.5 \pm 1.20</math></b>	<b><math>86.5 \pm 1.20</math></b>	$86.5 \pm 1.20$
CIFAR-10	0.150	0.95	$52.1 \pm 2.98$	$64.1 \pm 5.60$	$65.1 \pm 0.66$	<b><math>68.7 \pm 0.37</math></b>	$69.4 \pm 0.25$
CIFAR-100	0.284	0.69	$17.5 \pm 0.57$	$29.4 \pm 1.40$	$29.7 \pm 0.55$	<b><math>34.4 \pm 0.31</math></b>	$34.4 \pm 0.31$
FEMNIST	0.001	1.00	$48.3 \pm 2.98$	<b><math>66.2 \pm 0.23</math></b>	$57.8 \pm 1.93$	<b><math>66.2 \pm 0.23</math></b>	$66.2 \pm 0.23$
SHAKESPEARE	0.095	1.00	$30.9 \pm 0.51$	<b><math>44.1 \pm 0.27</math></b>	$41.1 \pm 0.56$	<b><math>44.1 \pm 0.27</math></b>	$44.1 \pm 0.27$

Figure 4.6: Evolution of the test accuracy when using different values of  $p_{\text{hist}}$  for CIFAR-100 dataset, when  $N_{\text{hist}}/N = 5\%$  (left), 20% (center), and 50% (right).

### 4.3.5 Conclusion

In this section, we formalized the problem of federated learning for data streams and highlighted a new source of heterogeneity resulting from local datasets’ variability over time. We proposed a general federated algorithm to learn in this setting and studied its theoretical guarantees. Our analysis reveals a new bias-optimization trade-off controlled by the relative importance of older samples in comparison to newer ones and leads to practical guidelines to configure such importance in our algorithm. Experiments show that our configuration rule outperforms natural ways to extend the usual FedAvg aggregation rule in the presence of data streams. Moreover, experimental results confirm other theoretical conclusions, despite the theoretical assumptions and the mismatch in the corresponding performance metrics (e.g., test accuracy versus a loss bound).

To the best of our knowledge, this work is the first to frame the problem of federated learning for data streams. It highlights new challenges and—we believe—lays the foundations for further research. For example, part of our results are restricted to the important, but still quite specific, scenario where some clients have static datasets and others process new samples at each step. In this setting, samples are used a different number of times across clients but exactly the same number of times at a given client, simplifying the analysis. But what happens if heterogeneity in samples’ availability also appears at the level of a single client? How do different memory update rules affect such heterogeneity, and how can we design such policies to minimize the total error of the final model? Finally, how do our results change if local data distributions change over time?

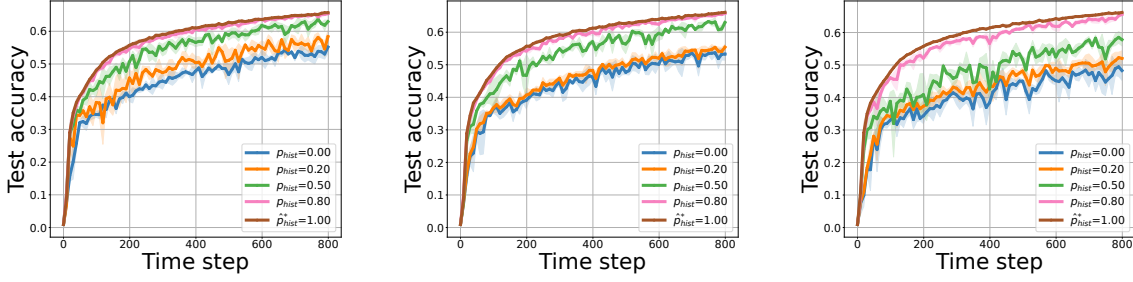


Figure 4.7: Evolution of the test accuracy when using different values of  $p_{\text{hist}}$  for FEMNIST dataset, when  $M_{\text{hist}}/M = 5\%$  (left),  $20\%$  (center), and  $50\%$  (right).

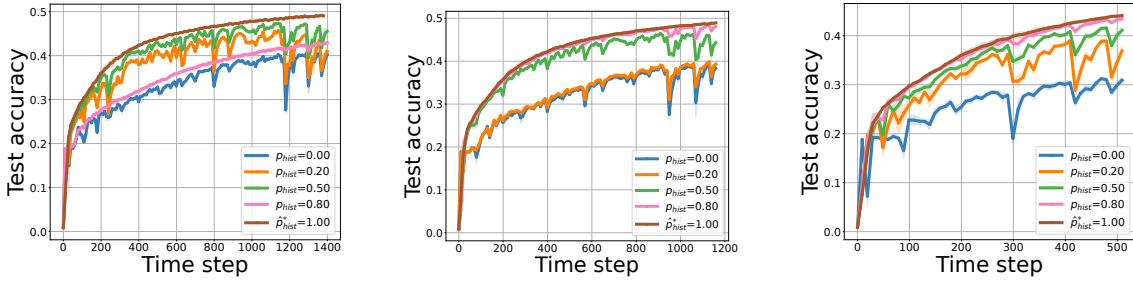


Figure 4.8: Evolution of the test accuracy when using different values of  $p_{\text{hist}}$  for Shakespeare dataset, when  $M_{\text{hist}}/M = 5\%$  (left),  $20\%$  (center), and  $50\%$  (right).

## 4.4 Online Federated Learning with Mixture Models

In this section, we propose to study online federated learning under the assumption that clients' data distributions are mixtures of a finite number of unknown underlying distributions with varying mixing weights. We propose *Federated Expectation-Maximization Online Mirror Descent* (FEM-OMD), a federated variant of the OMD algorithm, where the gradient of the cost function is estimated through an EM-like algorithm at each time-step. We analyze the regret guarantees of this algorithm in the case of *Gaussian Mixture Models* (GMMs); in particular, we show that the regret is asymptotically (in the sample size) sub-linear.

### 4.4.1 Problem Formulation

Let  $T > 0$  be a finite time horizon. We consider a setting with a central server and a finite set of clients  $\mathcal{C} = \{1, \dots, C\}$ . Data at each time  $t \in [T]$  and each client  $c \in \mathcal{C}$  is generated according to a parametric distribution  $\mathcal{P}_{t,c} \triangleq \mathcal{P}_{\theta_{t,c}^*}$  over a domain  $\mathcal{Z}$ , with unknown parameter  $\theta_{t,c}^* \in \Theta$ . The data distributions  $\mathcal{P}_{t,c}$  generally vary both in the dimension of clients and time, thus it is natural to fit a separate model  $\theta_{t,c} \in \Theta$  to each distribution  $\mathcal{P}_{t,c}$ . At each time-step  $t \in [T]$ , clients  $c \in \mathcal{C}$  collaboratively make predictions  $\theta_{t,c} \in \Theta$  based on the information they have acquired up to time-step  $t - 1$ . Once the predictions  $\theta_{t,c}$  have been made, each client  $c \in \mathcal{C}$  suffers the following expected log-loss [SB14, Chapter 24]

$$\mathcal{L}_{\mathcal{P}_{t,c}}(\theta_{t,c}) \triangleq -\mathbb{E}_{\mathcal{Z} \sim \mathcal{P}_{\theta_{t,c}^*}} \left[ \log(\mathcal{P}_{\theta_{t,c}}) \right] = \text{D}_{\text{KL}}(\mathcal{P}_{\theta_{t,c}^*} \parallel \mathcal{P}_{\theta_{t,c}}) + \mathbb{E}_{\mathcal{Z} \sim \mathcal{P}_{\theta_{t,c}^*}} \left[ \log(\mathcal{P}_{\theta_{t,c}^*}) \right]. \quad (4.21)$$

Finally, the data distribution  $\mathcal{P}_{t,c}$  is partially revealed to client  $c \in \mathcal{C}$  through a finite sample  $\mathcal{S}_{t,c}$  of  $n > 0$  examples drawn i.i.d. from  $\mathcal{P}_{t,c}$ . Crucially, the clients predict the models  $\theta_{t,c}$  before getting any information on the actual distributions  $\mathcal{P}_{t,c}$ . In this online setting, the goal of each client  $c \in \mathcal{C}$  is to minimize its regret

$$R_{T,c} \triangleq \sum_{t=1}^T \mathcal{L}_{\mathcal{P}_{t,c}}(\theta_{t,c}) - \min_{\theta \in \Theta} \sum_{s=1}^T \mathcal{L}_{\mathcal{P}_{s,c}}(\theta). \quad (4.22)$$

This regret is the difference between the total loss that client  $c$  suffers during  $T$  rounds and the loss suffered by the optimal decision-maker with full access to distributions  $\mathcal{P}_{t,c}$ ,  $t \in [T]$  in hindsight.

The goal of FL is to enable each client  $c$  to benefit from samples available at other clients in order to get a better estimation of  $\mathcal{L}_{\mathcal{P}_{t,c}}$ , and therefore, obtain a model that generalizes better to unseen examples drawn from  $\mathcal{P}_{t,c}$ . However, as we argue in Section 3.4, some assumption on clients distributions  $\mathcal{P}_{t,c}$ ,  $c \in \mathcal{C}$  is essential for FL to be provably beneficial, and in the most general case, there may be no gain in collaboration. Hence, similar to Section 3.5, we make the following assumption which is relatively weak and only requires the data for each client to be drawn from a mixture of  $m$  distributions, while the mixing coefficients are changing over time and are different for different clients.

**Assumption 26.** *There exists  $m > 0$  distributions  $\check{\mathcal{P}}_j \triangleq \check{\mathcal{P}}_{\omega_j^*}$ ,  $j \in [m]$  (parameterized with unknown  $\omega_j^* \in \Omega$ ) over  $\mathcal{Z}$ , such that, at time-step  $t \in [T]$ , client  $c$ 's distribution  $\mathcal{P}_{t,c}$  is a mixture of the distributions  $\check{\mathcal{P}}_j$ ,  $j \in [m]$  with weights  $\pi_{t,c}^* \in \Delta^{m-1}$ , i.e.,*

$$\forall \mathbf{z} \in \mathcal{Z}, \quad \mathcal{P}_{t,c}(\mathbf{z}) = \sum_{j=1}^m \pi_{t,c,j}^* \check{\mathcal{P}}_j(\mathbf{z}). \quad (4.23)$$

Assumption 26 motivates a multiple-model solution consisting in learning a set of global models  $\omega_{t,j}$  for each underlying distribution  $\check{\mathcal{P}}_j$ ,  $j \in [m]$  and personalized mixture weights  $\pi_{t,c} \in \Delta^{m-1}$ . In this case,  $\theta_{t,c} = \{(\omega_{t,j}, \pi_{t,c,j}) : j \in [m]\}$ , while  $\theta_{t,c}^* = \{(\omega_j^*, \pi_{t,c,j}^*) : j \in [m]\}$ . Under Assumption 26, our learning problem can be interpreted as a game where  $\mathcal{C}$  clients collaborate against a potentially adversarial environment. At the start of the game, the environment selects the parameters  $\omega_j^*$  of the underlying distributions  $\check{\mathcal{P}}_j$ ,  $j \in [m]$ . At time-step  $t \in [T]$ , the clients collaboratively make predictions  $\omega_{t,j} \in \Omega$ ,  $j \in [m]$  of the underlying distribution parameters, and  $\pi_{t,c} \in \Delta^{m-1}$  of their personalized mixture weights. Then, the environment independently chooses mixture weights  $\pi_{t,c}^* \in \Delta^{m-1}$  for each client  $c \in \mathcal{C}$ . Afterwards, each client  $c \in \mathcal{C}$  suffers a loss  $\varphi_{t,c}(\theta_{t,c})$ , defined as

$$\begin{aligned} \varphi_{t,c}(\theta_{t,c}) &\triangleq \varphi\left(\theta_{t,c} = \{(\omega_{t,j}, \pi_{t,c,j}) : j \in [m]\}; \theta_{t,c}^* = \{(\omega_j^*, \pi_{t,c,j}^*) : j \in [m]\}\right) \\ &\triangleq \text{D}_{\text{KL}}\left(\sum_{j=1}^m \pi_{t,c,j}^* \mathcal{P}_{\omega_j^*} \parallel \sum_{j=1}^m \pi_{t,c,j} \mathcal{P}_{\omega_{t,j}}\right). \end{aligned} \quad (4.24)$$

Finally, the environment (partially) reveals the selected mixture weights to the clients, through  $n$  samples drawn i.i.d. from  $\mathcal{P}_{t,c} = \sum_{j=1}^m \pi_{t,c,j}^* \check{\mathcal{P}}_j$ . The goal of client  $c$  is to minimize its personal regret given by

$$R_{T,c} \triangleq \sum_{t=1}^T \varphi_{t,c}(\theta_{t,c}) - \min_{\theta \in \Theta} \sum_{s=1}^T \varphi_{s,c}(\theta). \quad (4.25)$$

The environment cannot change the parameters  $\omega_j^*$  of the underlying distributions across time. Instead, the dynamic of the problem comes from how the mixing weights  $\pi_{t,c}$  change across time.

---

**Algorithm 14:** Federated Expectation-Maximization Online Mirror Descent  
(FEM-OMD)

---

**Input :** learning rate  $\eta > 0$ , number  $K$  of EM steps

- 1 Initialize  $\omega_{1,1}, \dots, \omega_{1,m} \in \Omega$  and  $\pi_{1,1}, \dots, \pi_{1,C} \in \Delta^{m-1}$  ;
- 2 **for**  $t = 1, \dots, T$  **do**
- 3     Server broadcasts  $\omega_{t,j}, j \in [m]$  to each client  $c \in \mathcal{C}$ ;
- 4     **for** *client*  $c \in \mathcal{C}$  *in parallel over*  $C$  *clients* **do**
- 5         Play parameters  $\theta_{t,c} = \{(\omega_{t,j}, \pi_{t,c,j}) : j \in [m]\}$ ;
- 6         Receive sample  $\mathcal{S}_{t,c} = z_{t,c,1}, \dots, z_{t,c,n} \stackrel{\text{i.i.d.}}{\sim} \mathcal{P}_{t,c} = \sum_{j=1}^m \pi_{t,c,j}^* \check{\mathcal{P}}_{\omega_j^*}$  ;
- 7         Initialize  $\tilde{\pi}_{t,c}^{(1)} \in \Delta^{m-1}$  and set  $\tilde{\omega}_{t,j}^{(1)} \leftarrow \omega_{t,j}, j \in [m]$  ;
- 8         **for**  $k = 1, \dots, K$  **do**
- 9              $\{(\tilde{\omega}_{t,c,j}^{(k+1)}, \tilde{\pi}_{t,c,j}^{(k+1)}) : j \in [m]\} \leftarrow$   
              EM\_update  $\left(\mathcal{S}_{t,c}, \{(\tilde{\omega}_{t,j}^{(k)}, \tilde{\pi}_{t,c,j}^{(k)}) : j \in [m]\}\right)$  ;
- 10            Synchronize  $\tilde{\omega}_{t+1,j}^{(k+1)} \leftarrow \sum_{c=1}^C \tilde{\pi}_{t,c,j}^{(k+1)} \tilde{\omega}_{t+1,c,j}^{(k+1)} / \sum_{c=1}^C \tilde{\pi}_{t,c,j}^{(k+1)}$  with the  
              server and other clients ;
- 11         **end**
- 12          $\omega_{t+1,j} \leftarrow \frac{t}{t+1} \omega_{t+1,j} + \frac{1}{t+1} \tilde{\omega}_{t+1,j}^{(K+1)}, j \in [m]$  ;
- 13          $\hat{\nabla}_{t,c,j} \leftarrow$   
              Approx $_n \left( -\mathbb{E}_{Z \sim \sum_{r=1}^m \tilde{\pi}_{t,c,r}^{(K+1)} \check{\mathcal{P}}_{\omega_{t+1,r}^{(K+1)}}} \left[ \frac{\check{\mathcal{P}}_{\omega_{t+1,j}^{(K+1)}}(Z)}{\sum_{l=1}^m \pi_{t,l} \check{\mathcal{P}}_{\omega_{t+1,c,l}}(Z)} \right] \right), j \in [m]$  ;
- 14          $\pi_{t+1,c,j} \leftarrow \pi_{t,c,j} \exp \left( -\eta \hat{\nabla}_{t,c,j} \right) / \sum_{l=1}^m \pi_{t,c,l} \exp \left( -\eta \cdot \hat{\nabla}_{t,c,l} \right), j \in [m]$  ;
- 15     **end**
- 16 **end**

---

#### 4.4.2 FEM-OMD Algorithm

Unlike traditional online convex optimization, where the decision maker directly receives the cost function from the environment, in our case, the cost function is indirectly revealed to each client through a set of samples. In practical terms, this means that each client has to estimate their cost function at each time step using a finite number of samples. Once clients obtain a good estimate of their cost functions, they can apply an online mirror descent step to decide their next action.

This iterative process is outlined in Algorithm 14. After committing to a set of actions (Line 5), the samples are received from the environment (Line 6), and clients collaborate to build a global estimator  $\tilde{\omega}_{t,j}^{(K+1)}$  of the underlying distribution  $\check{\mathcal{P}}_j$  parameter  $\omega_j^*$  for  $j \in [m]$ . Meanwhile, each client  $c \in \mathcal{C}$  builds locally an estimator  $\tilde{\pi}_{t,c}^{(K+1)}$  of its current mixing weight  $\pi_{t,c}^*$  (Lines 7–11). In order to build the estimators  $\tilde{\theta}_{t,c} \triangleq \{(\tilde{\omega}_{t,j}^{(K+1)}, \tilde{\pi}_{t,c,j}^{(K+1)}) : j \in [m]\}$ , FEM-OMD alternates between the client local updates and the synchronization steps at the server. In round  $k \in [K]$  of the inner loop, each client  $c \in \mathcal{C}$  performs an EM local update leading to local estimators  $\tilde{\omega}_{t,c,j}^{(k+1)}, j \in [m]$  and  $\tilde{\pi}_{t,c,j}^{(k+1)}$  of the distribution  $\mathcal{P}_{t,c}$  parameters (Line 9). Subsequently, clients synchronize their estimators  $\tilde{\omega}_{t,c,j}^{(k+1)}, j \in [m], c \in \mathcal{C}$  with the aid of a central server; the central server averages the

local estimators  $\tilde{\omega}_{t,c,j}^{(k+1)}$ ,  $j \in [m]$  and broadcasts the result back to all clients. The purpose of the inner loop (Lines 7–11) is to solve, in a distributed manner, the following maximum likelihood problem

$$\underset{(\omega_j)_{j \in [m]}, (\pi_c)_{c \in \mathcal{C}}}{\text{maximize}} \quad \frac{1}{nC} \sum_{c=1}^C \sum_{i=1}^n \log \left( \sum_{j=1}^m \pi_{c,j} \check{\mathcal{P}}_j(z_{t,c,i}) \right). \quad (4.26)$$

The empirical likelihood in (4.26) is based on the aggregated samples  $\mathcal{S}_t \triangleq \cup_{c \in \mathcal{C}} \mathcal{S}_{t,c}$ , i.e., containing all samples collected by the clients at time step  $t$ . Solving this problem collaboratively enables each client to benefit from other clients' data to build a more accurate estimator of  $\omega_j^*$ .

To approximate the unknown cost function  $\varphi_{t,c}$ , the client  $c \in \mathcal{C}$  uses the estimator  $\tilde{\theta}_{t,c}$  and computes its corresponding approximation  $\tilde{\varphi}_{t,c}$ . In Algorithm 14, the gradient of the true cost function  $\nabla \varphi_{t,c}$  is approximated using the gradient of the estimate  $\nabla \tilde{\varphi}_{t,c}$ . The gradients of the cost function  $\varphi_{t,c}$  and its estimate  $\tilde{\varphi}_{t,c}$  with respect to  $\pi$ , evaluated at  $\theta_{t,c} = \{(\omega_{t,j}, \pi_{t,c,j}) : j \in [m]\}$ , are given by

$$\nabla_{t,c,j} \triangleq \frac{\partial \varphi_{t,c}}{\partial \pi_j}(\theta_{t,c}) = -\mathbb{E}_{Z \sim \sum_{r=1}^m \pi_{t,c,r}^* \check{\mathcal{P}}_{\omega_r^*}} \left[ \frac{\check{\mathcal{P}}_{\omega_j^*}(Z)}{\sum_{l=1}^m \pi_{t,l} \check{\mathcal{P}}_{\omega_l^*}(Z)} \right], \quad (4.27)$$

$$\tilde{\nabla}_{t,c,j} \triangleq \frac{\partial \tilde{\varphi}_{t,c}}{\partial \pi_j}(\theta_{t,c}) = -\mathbb{E}_{Z \sim \sum_{r=1}^m \tilde{\pi}_{t,c,r}^{(K+1)} \check{\mathcal{P}}_{\omega_{t+1,r}}} \left[ \frac{\check{\mathcal{P}}_{\omega_{t+1,j}}(Z)}{\sum_{l=1}^m \pi_{t,l} \check{\mathcal{P}}_{\omega_{t+1,l}}(Z)} \right]. \quad (4.28)$$

Both  $\nabla_{t,c,j}$  and  $\tilde{\nabla}_{t,c,j}$  are given as expectations (integrals) that cannot be computed analytically. Therefore, in FEM-OMD,  $\nabla_{t,c,j}$  is approximated by  $\hat{\nabla}_{t,c,j}$ , which is computed using Monte-Carlo approximation (Line 13). Finally, client  $c \in \mathcal{C}$  computes the mixing weights  $\pi_{t+1,c}$  using a multiplicative update rule (Line 14). In Section 4.4.3.2, we show that this update rule is an Online Mirror Descent (OMD) step with negative entropy regularization.

### 4.4.3 Federated Online Learning with Gaussian Mixture Models

In this section, we analyze an instance of Assumption 8 where the underlying distributions are spherical Gaussian distributions, i.e.,  $\forall j \in [m]$ ,  $\check{\mathcal{P}}_{\omega_j^*} = \mathcal{N}(\mu_j^*, \mathbf{I}_d)$ , and  $\mu_j \in \mathbb{R}^d$ . In this case, the distribution  $\mathcal{P}_{t,c}$  of client  $c$  at time  $t$  is a Gaussian Mixture Model (GMM), with mixing weights  $\pi_{t,c}^* \in \Delta^{m-1}$ . The goal of the clients is to collaboratively learn the parameters of a sequence of GMMs, that share the same components but have time-varying mixing weights.

Learning the parameters of a GMM is a well-established problem [Das99; KSV05; GHK15; KC20] in the offline setting. The authors of [RV17] established that separation of  $\Omega(\sqrt{\log(m)})$  is necessary and sufficient for identifiability of the GMM's parameters with polynomial sample complexity. Assumption 27 formalizes a similar separation condition for the online setting that we consider.

**Assumption 27.** (*Separation*) Suppose that there exists a constant  $C' \geq 128$  such that

$$\forall j' \neq j \in [m], \quad \|\mu_{*,j} - \mu_{*,j'}\| \geq C' \sqrt{\log \left( \frac{m}{\min_{l,t} \pi_{*,l}^{(t)}} \right)}. \quad (4.29)$$

In [KC20], the authors show that, with separation  $\Omega\left(\sqrt{\log(m)}\right)$ , the (sample-splitting) finite-sample EM algorithm converges to the ground truth given an  $\mathcal{O}(1)$ -close initialization. Assumptions 28 and 29 formalizes the requirements for the initialization of the EM algorithm.

**Assumption 28.** (*Mean initialization*) The means initialization  $\boldsymbol{\mu}_{1,1}, \dots, \boldsymbol{\mu}_{1,m}$  satisfies

$$\forall j \in [m], \quad \left\| \boldsymbol{\mu}_{1,j} - \boldsymbol{\mu}_j^* \right\| \leq \min_{j' \neq j} \left\| \boldsymbol{\mu}_j^* - \boldsymbol{\mu}_{j'}^* \right\| / 16. \quad (4.30)$$

**Assumption 29.** (*Mixture weights initialization*) At every time step  $t \geq 0$ , the mixture weights initialization  $\tilde{\pi}_t^{(1)}$  satisfies

$$\forall j \in [m], \quad \left| \tilde{\pi}_{t,c,j}^{(1)} - \pi_{t,c,j}^* \right| \leq \pi_{t,c,j}^* / 2. \quad (4.31)$$

When the separation (Assumption 27), and the initialization (Assumption 28 and 29) assumptions hold, [KC20, Theorem 7] shows that  $n = \mathcal{O}(d/\epsilon^2 \min_l \pi_{*,l}^{(t)})$  samples are sufficient to recover the ground truth parameters up to  $\epsilon$  accuracy with high probability.

**Theorem 4.4.1.** [KC20, Theorem 7] If Assumptions 27–29 hold, and  $n \geq \frac{C'' d}{\epsilon^2 \min_l \pi_{*,l}^{(t)}} \cdot \log^2(m^2 T K / \delta)$  for a sufficiently large universal constant  $C''$ , and further the number of inner iterations is  $K = \mathcal{O}(\log(1/\epsilon))$ , then for all  $j \in [m]$ , with probability at least  $1 - \mathcal{O}(\delta/T) - \mathcal{O}(K/n^{c-2} m^{30})$  we have

$$\left| \tilde{\pi}_{t,j}^{(K+1)} - \pi_{t,c,j}^* \right| \leq \epsilon \pi_{t,c,j}^*, \quad \left\| \tilde{\boldsymbol{\mu}}_{t,c,j}^{(K+1)} - \boldsymbol{\mu}_j^* \right\| \leq \epsilon. \quad (4.32)$$

**Remark 7.** Note that the initialization assumptions, i.e., Assumptions 28 and 29, could be relaxed and replaced by the following separation condition, at the cost of running one step of  $k$ -means [KC20];

$$\forall j \in [m], \left\| \boldsymbol{\mu}_{1,j} - \boldsymbol{\mu}_j^* \right\| \leq \min_{j' \neq j} \left\| \boldsymbol{\mu}_j^* - \boldsymbol{\mu}_{j'}^* \right\| / 4. \quad (4.33)$$

Since the sample complexity of Theorem 4.4.1 depends on the inverse of the minimal mixing weight, we make the following assumption in order to guarantee finite sample complexity.

**Assumption 30.** (*Positive mixture weights*) Suppose that there exists a positive constant  $\beta \in (0, 1/m]$ , such that  $\min_j \pi_{t,j}^* \geq \beta$  for  $t \geq 0$ .

#### 4.4.3.1 FEM-OMD for Gaussian Mixture Models

Now we are ready to present the realization of FEM-OMD for the GMM setting. The steps of this algorithm is presented in Algorithm 15. For a mixture of spherical Gaussian distributions, each step of the EM algorithm is given in closed form as

$$\text{E-step:} \quad \tilde{w}_{t,c,j,i}^{(k)} = \frac{\tilde{\pi}_{t,c,j}^{(k)} f_{\tilde{\boldsymbol{\mu}}_{t,j}^{(k)}}(\mathbf{x}_{t,c,i}^{(k)})}{\sum_{l=1}^m \tilde{\pi}_{t,c,l}^{(k)} f_{\tilde{\boldsymbol{\mu}}_{t,l}^{(k)}}(\mathbf{x}_{t,c,i}^{(k)})}, \quad i \in [n/K], j \in [m] \quad (4.34)$$

$$\text{M-step:} \quad \tilde{\pi}_{t,c,j}^{(k+1)} = \sum_{i=1}^n \tilde{w}_{t,c,j,i}^{(k)} / n, \quad j \in [m] \quad (4.35)$$

$$\tilde{\boldsymbol{\mu}}_{t+1,c,j}^{(k+1)} = \sum_{i=1}^n \tilde{w}_{t,j,i}^{(k)} \mathbf{x}_{t,c,i}^{(k)} / \sum_{i=1}^n \tilde{w}_{t,j,i}^{(k)}, \quad j \in [m] \quad (4.36)$$

**Algorithm 15:** FEM-OMD for Gaussian Mixture Models

---

**Input :** learning rate  $\eta > 0$ , number of inner loop steps  $K$

- 1 Initialize  $\boldsymbol{\mu}_{1,1}, \dots, \boldsymbol{\mu}_{1,m} \in \mathbb{R}^d$  and  $\boldsymbol{\pi}_{1,1}, \dots, \boldsymbol{\pi}_{1,C} \in \Delta^{m-1}$  ;
- 2 **for**  $t = 1, \dots, T$  **do**
- 3     Server broadcasts  $\boldsymbol{\omega}_{t,j}, j \in [m]$  to each client  $c \in \mathcal{C}$ ;
- 4     **for** client  $c \in \mathcal{C}$  in parallel over  $C$  clients **do**
- 5         Play parameters  $\theta_{t,c} = \{(\boldsymbol{\mu}_{t,j}, \pi_{t,c,j}) : j \in [m]\}$ ;
- 6         Receive sample  $\mathbf{x}_{t,c,1}, \dots, \mathbf{x}_{t,c,n} \stackrel{\text{i.i.d.}}{\sim} \sum_{j=1}^m \pi_{t,j}^* \mathcal{N}(\boldsymbol{\mu}_j^*, I_d)$  ;
- 7         Split samples into  $K$  equally-sized batches  
 $\mathcal{B}_{t,c}^{(k)} = \{(\mathbf{x}_{t,c,1}^{(k)}, \dots, \mathbf{x}_{t,c,n/K}^{(k)})\}, k \in [K]$  ;
- 8         Initialize  $\tilde{\boldsymbol{\pi}}_{t,c}^{(1)} \in \Delta^{m-1}$  and set  $\tilde{\boldsymbol{\mu}}_{t,j}^{(1)} \leftarrow \boldsymbol{\mu}_{t,j}, j \in [m]$  ;
- 9         **for**  $k = 1, \dots, K$  **do**
- 10             Update  $\{(\tilde{\boldsymbol{\mu}}_{t,c,j}^{(k+1)}, \tilde{\pi}_{t,c,j}^{(k+1)}) : j \in [m]\}$  using Equations (4.34)–(4.36) ;
- 11             Synchronize  $\tilde{\boldsymbol{\mu}}_{t+1,j}^{(k+1)} \leftarrow \sum_{c=1}^C \tilde{\pi}_{t,c,j}^{(k+1)} \tilde{\boldsymbol{\mu}}_{t+1,c,j}^{(k+1)} / \sum_{c=1}^C \tilde{\pi}_{t,c,j}^{(k+1)}$  with the server and other clients ;
- 12         **end**
- 13          $\boldsymbol{\mu}_{t+1,j} \leftarrow \frac{t}{t+1} \boldsymbol{\mu}_{t+1,j} + \frac{1}{t+1} \tilde{\boldsymbol{\mu}}_{t+1,j}^{(K+1)}$  ;
- 14          $\hat{\nabla}_{t,j} \leftarrow$   
 $\text{Approx}_n \left( -\mathbb{E}_{X \sim \sum_{r=1}^m \tilde{\pi}_{t,c,r}^{(K+1)} \mathcal{N}(\boldsymbol{\mu}_{t+1,r}, I_d)} \left[ \frac{f_{\boldsymbol{\mu}_{t+1,j}}(X)}{\sum_{l=1}^m \pi_{t,l} f_{\boldsymbol{\mu}_{t+1,l}}(X)} \right] \right), j \in [m]$  ;
- 15          $\pi_{t+1,j} \leftarrow \pi_{t,j} \exp(-\eta \cdot \hat{\nabla}_{t,j}) / \sum_{l=1}^m \pi_{t,l} \exp(-\eta \cdot \hat{\nabla}_{t,l}), j \in [m]$  ;
- 16     **end**
- 17 **end**

---

The E-step constructs the expectation of the log-likelihood on the current estimators, and the M-step maximizes this expectation.

**Sample-splitting EM.** Algorithm 15 employs the common variant of the iterative EM algorithm which is often referred to as the *sample-splitting* scheme. This scheme divides the  $n$  examples into  $K$  batches of size  $n/K$ , and uses a new batch of samples in each iteration, which removes the probabilistic dependency between the iterations of the inner loop (Line 7–12).

**Remark 8.** Let  $\check{\theta}_{t,c}^* \triangleq \{(\boldsymbol{\mu}_j^*, \pi_{t,c,j}) : j \in [m]\}$ , and  $\check{\theta}_{t,c} \triangleq \{(\boldsymbol{\mu}_{t+1,j}, \pi_{t,c,j}) : j \in [m]\}$ . We remark that

$$\pi_{t,c,j} \nabla_{t,c,j} = -\mathbb{E}_{X \sim \mathcal{D}_{\theta_t^*}} [w(X; \check{\theta}_{t,c}^*)], \quad \pi_{t,c,j} \tilde{\nabla}_{t,c,j} = -\mathbb{E}_{X \sim \mathcal{D}_{\check{\theta}_t}} [w(X; \check{\theta}_{t,c})], \quad (4.37)$$

where  $w(\mathbf{x}; \theta) \triangleq \frac{f_{\boldsymbol{\mu}_j}(\mathbf{x})}{\sum_{l=1}^m \pi_l f_{\boldsymbol{\mu}_l}(\mathbf{x})}$  is the weight assigned by the EM algorithm to the example  $\mathbf{x} \in \mathbb{R}^d$ .

#### 4.4.3.2 Analysis

Algorithm 15 can be interpreted as an *online mirror descent with incorrect gradients* (Algorithm 16). Line 15 of Algorithm 15 is an online mirror descent step with entropy regularization using the



gradient of  $\tilde{\varphi}_{t,c}$  instead of the gradient of  $\varphi_{t,c}$ . In order to analyze the performance of Algorithm 15, we first analyze the more general online mirror descent with incorrect gradients. Our analysis (Theorem 4.4.3) shows that the regret of Algorithm 16 is upper bounded by the sum of a sub-linear term and a term that depends on the distance between the gradients of  $\tilde{\varphi}_{t,c}$  and  $\varphi_{t,c}$ . Second, we obtain an upper bound on the distance between the gradients of  $\tilde{\varphi}_{t,c}$  and  $\varphi_{t,c}$  (Lemma H.1). The bound is expressed using the distance between the true parameters  $\theta_{t,c}^*$  of the environment's GMM and their estimation  $\tilde{\theta}_{t,c}$  obtained using  $K$  steps of the EM algorithm, at time step  $t$  (Theorem 4.4.4). Finally, we use previous results on the convergence of the EM algorithm [KC20], to prove that, under Assumption 27–30, the distance between  $\theta_{t,c}^*$  and  $\tilde{\theta}_{t,c}$  is upper bounded by a term  $\mathcal{O}(1/\sqrt{n})$  with high probability (Theorem 4.4.1). By combining Theorem 4.4.3 and Theorem 4.4.5, we obtain our main regret bound presented in the following theorem. The proof is available in Appendix H.1.

**Theorem 4.4.2.** *Suppose that assumptions 27–30 hold. Suppose that  $n \geq \frac{C''d}{\beta c^2} \cdot \log^2\left(\frac{m^2TK}{\delta}\right)$  with sufficiently large universal constant  $C''$ , and  $K = \mathcal{O}(\log(1/\epsilon))$ . Algorithm 15 has regret bounded by*

$$\forall c \in \mathcal{C}, \quad R_{T,c} = \mathcal{O}(T\epsilon) + \mathcal{O}\left(\sqrt{T \log(m)}\right), \quad (4.38)$$

with probability at least  $1 - \mathcal{O}(\delta) - \mathcal{O}(TK/n^{c'-2}m^{30})$ .

Next, we characterize the steps mentioned above to obtain the result in Theorem 4.4.2.

**Online Mirror Descent with Inexact Gradients** The update rule (Line 15) of Algorithm 15 is an online mirror descent (OMD) step with entropy regularization [Haz16, Section 5.4.2]. The OMD step (in Line 15) employees  $\hat{\nabla}_t$  instead of the correct gradient  $\nabla\psi_t(\theta_t)$  of the cost function  $\psi_t$ . Theorem 4.4.3 bounds the regret of OMD when the correct gradient is replaced by an estimate (Algorithm 16).

---

**Algorithm 16:** Online Mirror Descent with Incorrect Gradients

---

**Input :** learning rate sequence  $\eta$ , regularization function  $R(\mathbf{x})$

- 1 Initialize  $\mathbf{y}_1$  such that  $\nabla\mathcal{R}(\mathbf{y}_1) = 0$  and  $\mathbf{x}_1 \in \arg \min_{\mathbf{x} \in \mathcal{X}} B_{\mathcal{R}}(\mathbf{x} \parallel \mathbf{y}_1)$ ;
- 2 **for**  $t = 1, \dots, T$  **do**
- 3     Play  $\mathbf{x}_t$ ;
- 4     Observe  $\hat{\nabla}_t$ ;
- 5      $\nabla\mathcal{R}(\mathbf{y}_{t+1}) \leftarrow \nabla\mathcal{R}(\mathbf{x}_t) - \eta\hat{\nabla}_t$ ;
- 6      $\mathbf{x}_{t+1} \leftarrow \arg \min_{\mathbf{x} \in \mathcal{X}} B_{\mathcal{R}}(\mathbf{x} \parallel \mathbf{y}_{t+1})$
- 7 **end**

---

In this section, we borrow the notation of [Haz16, Chapter 5]. We consider regularization functions, denoted  $\mathcal{R} : \mathcal{X} \mapsto \mathbb{R}$ , which are smooth, strongly convex and twice differentiable. We denote the diameter of the set  $\mathcal{X}$  relative to the function  $\mathcal{R}$  as  $D_{\mathcal{R}} \triangleq \sqrt{\max_{\mathbf{x}, \mathbf{y} \in \mathcal{X}} \{\mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{y})\}}$ . We use  $B_{\mathcal{R}}(\cdot \parallel \cdot)$  to denote the Bregman divergence with respect to the function  $\mathcal{R}$ , defined for  $\mathbf{x}, \mathbf{y} \in \mathcal{X}$  as  $B_{\mathcal{R}}(\mathbf{x} \parallel \mathbf{y}) \triangleq \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{y}) - \langle \nabla\mathcal{R}(\mathbf{x}), \mathbf{x} - \mathbf{y} \rangle$ .

For  $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ , we consider the norm  $\|\mathbf{x}\|_{\mathbf{y}} \triangleq \sqrt{\mathbf{x}^\top \nabla^2 \mathcal{R}(\mathbf{y}) \mathbf{x}}$ , and its dual norm  $\|\mathbf{x}\|_{\mathbf{y}}^* \triangleq \sqrt{\mathbf{x}^\top \nabla^{-2} \mathcal{R}(\mathbf{y}) \mathbf{x}}$ . The mean-value theorem asserts the existence of a point  $\mathbf{z} \in [\mathbf{x}, \mathbf{y}]$  such that  $B_{\mathcal{R}}(\mathbf{x} \parallel \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_{\mathbf{z}}$ . Therefore, the Bregman divergence defines a local norm, which has

a dual norm. We denote this dual norm by  $\|\cdot\|_{\mathbf{x},\mathbf{y}}^* \triangleq \|\cdot\|_{\mathbf{z}}^*$ . For two consecutive decision points  $\mathbf{x}_t$  and  $\mathbf{x}_{t+1}$  of Algorithm 16, we use  $\|\cdot\|_t \triangleq \|\cdot\|_{\mathbf{x}_t,\mathbf{x}_{t+1}}$  to denote the local norm at iteration  $t$  of Algorithm 16.

**Theorem 4.4.3.** *Let  $(\psi_t)_{0 \leq t \leq T}$  be a sequence of convex functions, and  $\mathbf{u} \in \mathcal{X}$ . Suppose the gradient norm is bounded as  $\|\hat{\nabla}_t\|_t^* \leq G_{\mathcal{R}}$ , and the stepsize is  $\eta = \frac{D_{\mathcal{R}}}{G_{\mathcal{R}}\sqrt{T}}$ . Then, for Algorithm 16 we have*

$$\sum_{t=1}^T \psi_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{s=1}^T \psi_s(\mathbf{x}) \leq D_{\mathcal{R}} G_{\mathcal{R}} \sqrt{T} + D_{\mathcal{X}} \cdot \sum_{t=1}^T \left\| \nabla \psi_t(\mathbf{x}_t) - \hat{\nabla}_t \right\|, \quad (4.39)$$

where  $D_{\mathcal{X}} \triangleq \sup_{\mathbf{x}, \mathbf{y} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|$  is the diameter of  $\mathcal{X}$ .

The first term in the above upper bound is the standard regret bound of the online mirror descent algorithm [Haz16, Theorem 5.6]. The second term is due to the fact that we have access to inexact gradients, and it is proportional to the cumulative distance between  $\hat{\nabla}_t$  and  $\nabla \psi_t(\mathbf{x}_t)$ .

**Application to FEM-OMD.** The regret bound of Algorithm 15 can be obtained by using the result in Theorem 4.4.3. More precisely, suppose that the decision set is the  $(m-1)$ -unitary simplex, i.e.,  $\mathcal{X} = \Delta^{m-1}$ , the regularization function is the negative entropy function, i.e.,  $\mathcal{R}(\boldsymbol{\pi}) \triangleq \sum_{j=1}^m \pi_j \log(\pi_j)$ , and the cost function is defined as  $\psi(\boldsymbol{\pi}) = \text{D}_{\text{KL}}\left(\sum_{j=1}^m \pi_{t,c,j}^* f_{\mu_j}^* \parallel \sum_{j=1}^m \pi_j f_{\mu_j}^*\right)$ . Note that in this case, the gradient of the negative entropy function is given by  $\frac{\partial \mathcal{R}}{\partial \pi_j}(\boldsymbol{\pi}) = 1 + \log(\pi_j)$ . Therefore, the update rule of Algorithm 16 becomes

$$\log(q_{t+1,c,j}) \triangleq \log(\pi_{t,c,j}) - \eta \hat{\nabla}_{t,c,j}, \quad \boldsymbol{\pi}_{t+1,c} = \arg \min_{\boldsymbol{\pi} \in \Delta^{m-1}} \sum_{j=1}^m q_{t+1,c,j} \log(\pi_j). \quad (4.40)$$

Hence,

$$\pi_{t+1,c,j} = \frac{q_{t+1,c,j}}{\sum_{l=1}^m q_{t+1,c,l}} = \frac{\pi_{t,c,j} \exp(-\eta \cdot \hat{\nabla}_{t,c,j})}{\sum_{l=1}^m \pi_{t,c,l} \exp(-\eta \cdot \hat{\nabla}_{t,c,l})}, \quad (4.41)$$

which corresponds to Line 15 of Algorithm 15. In this case, the diameter of the set  $\Delta^{m-1}$  relative to the negative entropy functions is  $D_{\mathcal{R}} = \sqrt{\log(m)}$  (Lemma H.4). Additionally, one can prove that  $\|\hat{\nabla}_t\|_t^* \leq \|\hat{\nabla}_t\|_{\infty} = \mathcal{O}(1)$  (Lemma H.5). Considering these points, Theorem 4.4.3 implies that the regret of Algorithm 15 is upper-bounded as

$$\forall c \in \mathcal{C}, \quad R_{T,c} \leq \mathcal{O}\left(\sqrt{T \log(m)}\right) + 2 \sum_{t=1}^T \left\| \nabla_{t,c} - \hat{\nabla}_{t,c} \right\|. \quad (4.42)$$

**Gradient Estimation Error** In Section 4.4.3.2, we viewed Algorithm 15 as a particular instance of online mirror descent with incorrect gradients (Algorithm 16). The expression in (4.42) bounds the regret of Algorithm 15 by the sum of a sub-linear term and the cumulative sum of the estimation error of cost functions' gradients. In this section, we bound the gradient estimation error  $\|\nabla_{t,c} - \hat{\nabla}_{t,c}\|$  that appears in (4.42). We remind that  $\hat{\nabla}_{t,c}$  is a Monte-Carlo approximation of  $\nabla_{t,c}$  using  $n$  samples, hence  $\|\hat{\nabla}_{t,c} - \tilde{\nabla}_{t,c}\| = \mathcal{O}(1/\sqrt{n})$ . Therefore, in order to prove that  $\|\nabla_{t,c} - \hat{\nabla}_{t,c}\| =$

$\mathcal{O}(1/\sqrt{n})$ , it is enough to prove that  $\|\nabla_{t,c} - \tilde{\nabla}_{t,c}\| = \mathcal{O}(1/\sqrt{n})$ . Starting from Remark 8, we bound the components-wise gradient estimation error  $|\nabla_{t,c,j} - \tilde{\nabla}_{t,c,j}|$  using the distance between the true parameters  $\theta_{t,c}^*$  of the environment's GMM and their estimation  $\tilde{\theta}_{t,c}$  obtained using  $K$  steps of the EM algorithm, at time step  $t$ , as shown by Theorem 4.4.4.

**Theorem 4.4.4.** *Suppose Assumptions 27–30 hold, and the number of samples satisfies  $n \geq \frac{C''d}{\beta\epsilon^2} \cdot \log^2(m^2TK/\delta)$ , where  $C''$  is a sufficiently large universal constant, and the number of inner loop steps satisfies  $K = \mathcal{O}(\log(1/\epsilon))$ . Then, for all  $t \in [T]$ ,  $c \in \mathcal{C}$  and  $j \in [m]$ , we have*

$$\pi_{t,c,j} \cdot |\nabla_{t,c,j} - \tilde{\nabla}_{t,c,j}| \leq \frac{3}{2} \cdot \sup_{q \in [m]} \|\mu_{t+1,q} - \mu_q^*\| + \|\tilde{\pi}_{t,c,j}^{(K+1)} - \pi_{t,c,j}^*\|_1. \quad (4.43)$$

The next step consists in upper bounding the RHS of (4.43). We use the convergence results (Theorem 4.4.1) of the EM algorithm from [KC20] to prove that the output  $\tilde{\theta}_{t,c}$  of the inner loop of Algorithm 15 is close to the true parameters  $\theta_{t,c}^*$ . When EM is initialized close to the ground truth (Assumptions 28, and 29), and the ground truth components are separated (Assumption 27), EM converges to the ground truth with high probability (Theorem 4.4.1). Combining Theorem 4.4.1 and Theorem 4.4.4, we prove Theorem 4.4.5 showing that the gradient estimation error  $\|\nabla_t - \hat{\nabla}_t\|$  is upper bounded by  $\mathcal{O}(1/\sqrt{n})$  with high probability, where  $n$  is the number of samples that the environment reveals to the decision-maker at each time step.

**Theorem 4.4.5.** *Suppose Assumptions 27–30 hold, and the number of samples satisfies the condition  $n \geq \frac{C''d}{\beta\epsilon^2} \cdot \log^2(m^2TK/\delta)$ ,  $C''$  is a sufficiently large universal constant, and the number of inner loop iterations is selected as  $K = \mathcal{O}(\log(1/\epsilon))$ . Then, for all  $t \in [T]$  and for all  $c \in \mathcal{C}$ , with probability at least  $1 - \mathcal{O}(\delta/T) - \mathcal{O}(K/n^{\epsilon-2}m^{30})$ , we have  $\|\nabla_{t,c} - \tilde{\nabla}_{t,c}\| = \mathcal{O}(\epsilon)$ .*

#### 4.4.4 FEM-OMD for Discriminative Models

Our problem formulation in Section 4.4.1 focuses on generative models, where the purpose is to learn the underlying data generation process. In this section, we focus on the discriminative case, where the purpose is to model the conditional probability distribution of the target variable given the input variables. In this section, we consider the case that  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ , where  $\mathcal{X}$  is the input space and  $\mathcal{Y}$  is the target/output space. We consider a set  $\mathcal{H}_\Omega = \{h_\omega : \mathcal{X} \mapsto \mathcal{Y}, \omega \in \Omega\}$  of parametric hypotheses/models mapping  $\mathcal{X}$  to  $\mathcal{Y}$ . We use  $\mathcal{H}$  to denote the convex hull of  $\mathcal{H}_\Omega$ , and we consider a loss function  $\ell : \mathcal{H} \times \mathcal{Z} \mapsto \mathbb{R}_+$ , quantifying the discrepancy between the predicted output  $h(\mathbf{x})$  of a hypothesis  $h$  and the target value  $\mathbf{y}$ , for  $(\mathbf{x}, \mathbf{y}) \in \mathcal{Z}$ . For  $\omega \in \Omega$ , let  $\check{\mathcal{P}}_\omega$  be the distribution defined by  $\check{\mathcal{P}}_\omega(\mathbf{z}) \propto \exp\{-\ell(h_\omega; \mathbf{z})\}$  for  $\mathbf{z} \in \mathcal{Z}$ .

In the discriminative case, the goal is to learn a hypothesis/model mapping the input space to the target space. In our settings, where the mixture assumption (Assumption 8) holds, we learn a set of global models  $\{h_{\omega_{t,j}}\}$  for each underlying distribution  $j \in [m]$ , and time-varying personalized mixing weights  $\pi_{t,c}$  for each client  $c \in \mathcal{C}$ . Motivated by [Mar+21b, Proposition 2.1], client  $c$ 's model at time-step  $t$  is given by  $h_{t,c} \triangleq \sum_{j=1}^m \pi_{t,c} h_{\omega_{t,j}}$ .

The steps of FEM-OMD for discriminative models are presented in Algorithm 17. In particular,

**Algorithm 17:** FEM-OMD for discriminative models

---

**Input :** learning rate  $\eta > 0$ , number  $K$  of EM steps

- 1 Initialize  $\omega_{1,1}, \dots, \omega_{1,m} \in \Omega$  and  $\pi_{1,1}, \dots, \pi_{1,C} \in \Delta^{m-1}$  ;
- 2 **for**  $t = 1, \dots, T$  **do**
- 3     Server broadcasts  $\omega_{t,j}, j \in [m]$  to each client  $c \in \mathcal{C}$ ;
- 4     **for** client  $c \in \mathcal{C}$  in parallel over  $C$  clients **do**
- 5         Play parameters  $\theta_{t,c} = \{(\omega_{t,j}, \pi_{t,c,j}) : j \in [m]\}$ ;
- 6         Receive sample  $\mathcal{S}_{t,c} = z_{t,c,1}, \dots, z_{t,c,n} \stackrel{\text{i.i.d.}}{\sim} \mathcal{P}_{t,c} = \sum_{j=1}^m \pi_{t,c,j}^* \check{\mathcal{P}}_{\omega_j^*}$  ;
- 7         Initialize  $\tilde{\pi}_{t,c}^{(1)} \in \Delta^{m-1}$  and set  $\tilde{\omega}_{t,j}^{(1)} \leftarrow \omega_{t,j}, j \in [m]$  ;
- 8         **for**  $k = 1, \dots, K$  **do**
- 9             Update  $\tilde{\omega}_{t,c,j}^{(k+1)}$  and  $\tilde{\pi}_{t,c,j}^{(k+1)}$  according to (4.45) and (4.46);
- 10            Synchronize  $\tilde{\omega}_{t+1,j}^{(k+1)} \leftarrow \sum_{c=1}^C \tilde{\pi}_{t,c,j}^{(k+1)} \tilde{\omega}_{t+1,c,j}^{(k+1)} / \sum_{c=1}^C \tilde{\pi}_{t,c,j}^{(k+1)}$  with the server and other clients ;
- 11         **end**
- 12          $\omega_{t+1,j} \leftarrow \frac{t}{t+1} \omega_{t+1,j} + \frac{1}{t+1} \tilde{\omega}_{t+1,j}^{(K+1)}, j \in [m]$  ;
- 13          $\hat{\nabla}_{t,c,j} \leftarrow$
- 14             Approx $_n \left( -\mathbb{E}_{Z \sim \sum_{r=1}^m \tilde{\pi}_{t,c,r}^{(K+1)} \check{\mathcal{P}}_{\omega_{t+1,j}^{(K+1)}}} \left[ \frac{\check{\mathcal{P}}_{\omega_{t+1,j}^{(K+1)}}(Z)}{\sum_{l=1}^m \pi_{t,l} \check{\mathcal{P}}_{\omega_{t+1,c,l}^{(K+1)}}(Z)} \right] \right), j \in [m]$  ;
- 15          $\pi_{t+1,c,j} \leftarrow \pi_{t,c,j} \exp \left( -\eta \cdot \hat{\nabla}_{t,c,j} \right) / \sum_{l=1}^m \pi_{t,c,l} \exp \left( -\eta \cdot \hat{\nabla}_{t,c,l} \right), j \in [m]$
- 16         ;
- 17     **end**
- 18 **end**

---

the local EM update performed by client  $c$  is summarized as follows:

$$\text{E-step: } \tilde{w}_{t,c,j,i}^{(k+1)} = \frac{\tilde{\pi}_{t,c,j}^{(k)} \check{\mathcal{P}}_{\tilde{\omega}_{t,j}^{(k)}}(\mathbf{x}_{t,c,i}, y_{t,c,i})}{\sum_{l=1}^m \tilde{\pi}_{t,c,l}^{(k)} \check{\mathcal{P}}_{\tilde{\omega}_{t,l}^{(k)}}(\mathbf{x}_{t,c,i}, y_{t,c,i})}, \quad i \in [n], j \in [m] \quad (4.44)$$

$$\text{M-step: } \tilde{\pi}_{t,c,j}^{(k+1)} = \sum_{i=1}^n \tilde{w}_{t,c,j,i}^{(k)} / n, \quad j \in [m] \quad (4.45)$$

$$\tilde{\omega}_{t,c,j}^{(k+1)} = \tilde{\omega}_{t,c,j}^{(k)} - \eta \sum_{i=1}^n \tilde{w}_{t,c,j,i}^{(k)} \nabla_{\omega} \ell \left( h_{\omega_{t,c,j}^{(k)}}(\mathbf{x}_{t,c,i}), y_{t,c,i} \right), \quad j \in [m] \quad (4.46)$$

The local EM update consists in performing the steps in (4.44) and (4.45) and updating the local estimates of  $\omega^*$  through one step of stochastic gradient descent using only the local dataset  $\mathcal{S}_{t,c}$ .

## 4.4.5 Experimental Results

### 4.4.5.1 Synthetic data.

We investigate the Gaussian mixture model analyzed in Section 4.4.3.2. We randomly generate centers  $\mu_j^* \in \mathbb{R}^d$  by sampling from  $\mathcal{N}(0, s\mathbf{I}_d)$ , where  $s = 5$  controls the distance between the

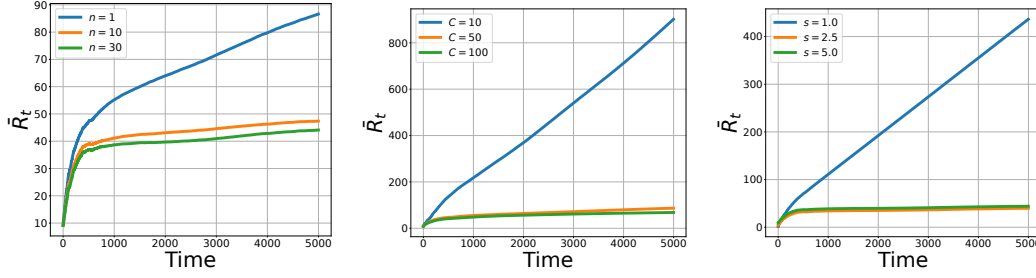


Figure 4.9: Evolution of average regret across clients ( $\bar{R}_t$ ) as a function of number of samples and clients. Left:  $\bar{R}_t$  for different values of  $n$ . Center:  $\bar{R}_t$  for different values of  $C$  with each client receiving only one sample per time-step. Right:  $\bar{R}_t$  for different values of  $s$ .

centers. The mixing weights  $\pi_{t,c}^* \in \Delta^{m-1}$  associated with each client  $c$  at time step  $t \in [T]$  are generated using a symmetric Dirichlet distribution with parameter  $\alpha = 0.1$ . We set  $C = 50$  and  $n = 30$ , unless otherwise specified.

To investigate the effect of the number of samples, we plot the evolution of the average regret across clients,  $\bar{R}_t$ , in Figure 4.9 (**left**) for different values of  $n$ . We observe that the regret is sub-linear for  $n \geq 10$  and linear for  $n = 1$ , in accordance with Theorem 4.4.2. For large values of  $n$ , the second term  $\mathcal{O}(\sqrt{T \log(m)})$  dominates the first term  $\mathcal{O}(T/\sqrt{n})$  in the RHS of (4.38). In Figure 4.9 (**center**), we investigate the benefit of collaboration by plotting the evolution of  $\bar{R}_t$  for different values of the number of clients  $C$ , when  $n = 1$ . The plot shows the regret of FEM-OMD improves with larger values of  $C$ . More precisely, larger values of  $C$  ( $= 100$ ) lead to a sub-linear regret, while smaller values lead to a linear regret. This result demonstrates the benefit of collaboration when the number of local samples  $n$  is small. In Figure 4.9 (**right**), we plot the evolution of the average regret across clients for different values of the parameter  $s$  controlling the distance between the centers of the Gaussian distributions. We observe that the regret is sub-linear for large  $s \geq 2.5$  and linear for  $s = 1.0$ . This result demonstrates the necessity of the separation assumption (Assumption 27); when Assumption 27 does not hold, the regret of Algorithm 15 is linear, even if the number of participating clients is large ( $C = 50$ ), and each client receives a large number of samples ( $n = 30$ ).

#### 4.4.5.2 Federated Learning Datasets

**Datasets and models.** We consider two image classification datasets: MNIST [LC10] and CIFAR-10 [Kri09]. To create distinct subsets, we divide the 10 classes of CIFAR-10 and MNIST into  $m = 4$  subsets, where each subset corresponds to an underlying distribution. For the CIFAR-10 dataset, we train a shallow convolutional neural network with two convolutional layers followed by two fully connected layers. For MNIST, we use a two-layer fully connected neural network. At each time-step  $t$ , let  $\mathbf{n}_t$  (resp.  $\mathbf{n}'_t$ ) be a realization of the random variable distributed according to the multinomial distribution  $\mathcal{M}(n, \pi_{t,c,j})$  (resp.  $\mathcal{M}(n', \pi_{t,c,j})$ ). Each client  $c$  receives  $\mathbf{n}_{t,j}$  training samples, denoted as  $\mathcal{S}_{t,c}$ , and  $\mathbf{n}_{t,j}$  test samples, denoted as  $\mathcal{S}'_{t,c}$ , drawn from the  $j$ -th subset for  $j \in [m]$ . The mixing weights  $\pi_{t,c}^* \in \Delta^{m-1}$  associated with each client  $c$  at time-step  $t \in [T]$ , are generated according to a symmetric Dirichlet distribution with parameter  $\alpha = 0.1$ . In our experiments, we set  $C = 10$ ,  $n' = 1$ ,  $n = 5$  for CIFAR-10, and  $n = 6$  for MNIST, resulting in a time horizon of approximately  $T = 1,000$ . Our code will be made available upon acceptance.

**Baseline.** In Section 4.2, we discussed that most online federated learning approaches have

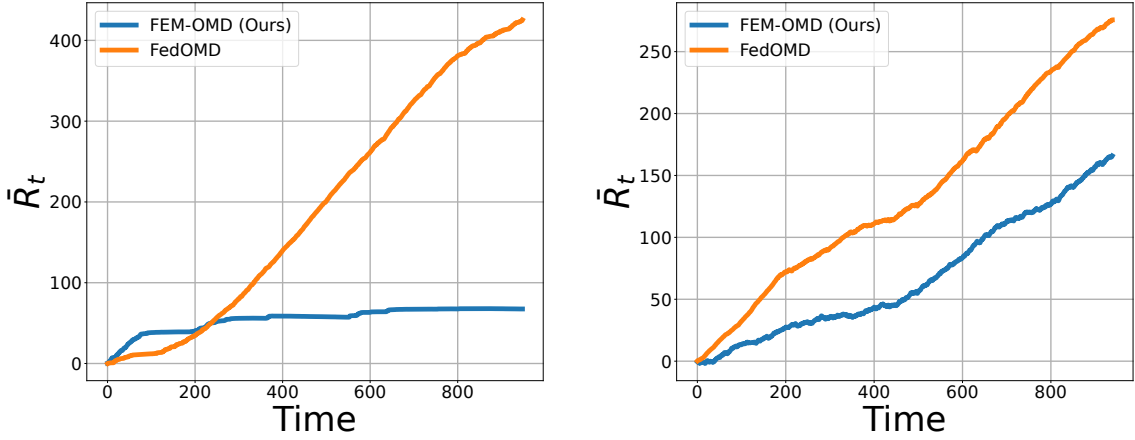


Figure 4.10: Evolution of average regret across clients ( $\bar{R}'_t$ ) for CIFAR-10 (**right**) and MNIST (**left**). The curves are smoothed using a discount factor of 0.7.

two main characteristics. Firstly, they either require prior knowledge about the proportion of clients/samples from each underlying distribution, as seen in [Eic+19; Din+20; Zhu+22]. Secondly, they determine the models to be served to clients based on feedback from the environment, as described in [Jot+23]. These approaches differ from our framework, where clients predict the models  $\theta_{t,c}$  before receiving any information about the actual distributions  $\mathcal{P}_{t,c}$ . It is important to note that FedOMD [MHP21] is the only exception to this. In our evaluation, we compare our algorithm with the  $L_2$ -regularized implementation of FedOMD

**Metric.** In this section, we introduce an alternative formulation of regret denoted as  $R'_{T,c}$ , which is defined as the cumulative difference between the accuracy of the optimal model  $h^*$  trained on the aggregated samples  $\mathcal{S} \triangleq \cup_{t \in [T]} \cup_{c \in \mathcal{C}} \mathcal{S}_{t,c}$  and the accuracy of the model  $h_{t,c}$  trained during each learning round  $t$ . The formulation is given by the equation:

$$R'_{T,c} \triangleq \sum_{t=1}^T \left\{ \text{Acc} \left( h^*, \mathcal{S}'_{t,c} \right) - \text{Acc} \left( h_{t,c}, \mathcal{S}'_{t,c} \right) \right\}. \quad (4.47)$$

Here,  $\text{Acc}(h, \mathcal{S})$  represents the accuracy of the model  $h$  on the dataset  $\mathcal{S}$ . It's important to note that  $h^*$  is the optimal model trained using the aggregated samples  $\mathcal{S}$  collected by all clients during the  $T$  learning rounds

**Machines and libraries.** We used PyTorch [Pas+19] to build and train our models. We ran the experiments on a GeForce GTX 1080 Ti Nvidia card.

**Results.** In Figure 4.10, we present the evolution of the average regret across clients ( $\bar{R}'_t$ ) for the CIFAR-10 and MNIST datasets, depicted on the right and left sides, respectively. When considering the MNIST dataset, our proposed method (FED-OMD) achieves a sub-linear regret before reaching time-step  $t_0 = 200$ , whereas FedOMD takes longer to enter the sub-linear regime. However, for the more challenging CIFAR-10 dataset, neither FedOMD nor FED-OMD achieves sub-linear regret with  $T = 1,000$  time-steps. It remains unclear whether this absence of sub-linear behavior stems from the inherent limitations of FedOMD and FED-OMD, or if the time horizon ( $T = 1,000$ ) is simply insufficient to demonstrate the sub-linear behavior.

#### **4.4.6 Conclusion and Perspectives**

We have proposed a novel formulation for online federated learning under the assumption that clients' data distributions are mixtures of a finite number of unknown underlying distributions with varying mixing weights. Our proposed Federated Expectation-Maximization Online Mirror Descent (FEM-OMD) algorithm leverages all of the data stored across clients to learn the parameters of the underlying distributions using EM updates, while enabling each client to adapt to the temporal variation of its data distribution. Through theoretical analysis and experimental results, we have demonstrated the effectiveness of our approach in online federated settings, particularly in the case of Gaussian mixture models. We believe that our work opens up new directions for research in online federated learning, where clients' data distributions are allowed to vary in constrained adversarial manners, and we hope that our proposed algorithm will pave the way for further improvements in this field.

# CHAPTER 5

---

## Conclusion

In this manuscript, we conducted a comprehensive investigation into various sources of heterogeneity in federated learning, proposing novel algorithms to mitigate their adverse impacts on federated and collaborative learning systems.

**System Heterogeneity.** Chapter 2 is dedicated to addressing system heterogeneity in cross-silo (Section 2.1) and cross-device (Section 2.2) federated learning. Additionally, Section 3.6 (within Chapter 3) introduces the local memorization technique, proven effective in addressing system heterogeneity in federated learning scenarios featuring highly diverse hardware, such as smartphones, IoT devices, edge computing servers, and the cloud.

**Statistical Heterogeneity.** Chapter 3 focuses on tackling statistical heterogeneity in federated learning. It provides an overview of personalization techniques, presents a learning impossibility result, and introduces two novel personalization algorithms, namely `FedEM` (Section 3.5) and `kNN-Per` (Section 3.6), with applications to federated learning.

**Temporal Heterogeneity.** Chapter 4 is devoted to the exploration of federated learning in dynamic environments, where clients collaboratively learn from distributed data streams characterized by the continuous generation of data. Specifically, Chapter 4 aims to address two orthogonal challenges encountered in federated learning within dynamic environments. The first challenge (addressed in Section 4.3) arises from the variability, across time and across clients, in the duration that different samples reside in memory, while the second challenge (addressed in Section 4.4) is attributed to the variability in the underlying distributions of clients across time.

In this chapter, we provide a concise summary of the main contributions of the manuscript in Section 5.1, followed by an overview of potential future research directions in Section 5.2. The manuscript concludes with final remarks presented in Section 5.3.

### 5.1 Summary of the Main Contributions

#### Throughput-Optimal Topology Design for Cross-Silo Federated Learning

Section 2.1 sheds light on the inefficiencies of the standard federated learning approach, particularly in cross-silo settings, where the server-client architecture may lead to suboptimal communication speeds due to potential bottlenecks at the orchestrator. Recognizing this limitation, the central question addressed in this chapter is: “how can we design a communication topology that facilitates the fastest convergence, considering the varied communication capabilities of different silos.” The contributions of Section 2.1 are threefold:

- The section formulates the problem of topology design for cross-silo federated learning, employing the theory of max-plus linear systems to quantify system throughput.



- It proposes practical algorithms that, based on measurable network characteristics, can identify topologies with either the maximum throughput or guaranteed throughput.
- It empirically demonstrates the practical impact of the proposed algorithms, showcasing significant speed-ups in real-world Internet networks. Specifically, our algorithms a  $9\times$  acceleration compared to the server-client architecture and  $1.5\times$  faster than state-of-the-art MATCHA, with even more pronounced speed-ups in scenarios featuring slower access links.

Building upon the idea, introduced in Section 2.1, of prioritizing high throughput in topologies, as opposed to exclusively targeting optimal consensus rates, recent research by Takezawa et al. [Tak+23] introduces a novel class of topologies. These topologies not only exhibit rapid consensus rates but also maintain a minimal maximum degree.

### Federated Learning under Heterogeneous and Correlated Client Availability

Section 2.2 analyzes a FedAvg-like algorithm under heterogeneous and correlated client availability. The analysis highlights how correlation adversely affects the algorithm’s convergence rate and how the aggregation strategy can alleviate this effect at the cost of steering training toward a biased model. Guided by the theoretical analysis, we propose Correlation-Aware FL (CA-Fed), a new FL algorithm that tries to balance the conflicting goals of maximizing convergence speed and minimizing model bias. To this purpose, CA-Fed dynamically adapts the weight given to each client and may ignore clients with low availability and large correlation. The contributions of Section 2.2 are threefold:

- The section provides a novel analysis of FedAvg under heterogeneous and correlated client availability. The analysis assumes clients’ temporal and spatial availability follows an arbitrary finite-state Markov process, providing a realistic modeling of correlated client activity while maintaining analytical tractability. The theoretical quantifies the negative effect of correlation on convergence rate, introducing an additional term dependent on Markov chain spectral properties, and highlights a trade-off between slow convergence to the optimal model and fast convergence to a biased model, providing theoretical insights.
- Guided by insights from the theoretical analysis, the section proposes CA-Fed, a federated learning algorithm designed to dynamically assign weights to clients. The algorithm aims at balancing the trade-off between maximizing convergence speed and minimizing model bias based on the theoretical analysis.
- Empirically demonstrates that CA-Fed achieves comparable maximum test accuracy as state-of-the-art methods (F3AST[RVd23] and AdaFed[Tan+22a]) while achieving higher time-average and lower standard deviation of the test accuracy. Moreover, the experimental results demonstrate the effectiveness of excluding clients with high temporal correlation and low availability in federated learning.

### Personalized Federated Learning under a Mixture of Distributions

Section 3.5 studies personalized federated learning (also known as federated multi-task learning) under the flexible assumption that each local data distribution is a mixture of unknown underlying distributions. This formulation allows every client to harness insights distilled from the diverse

datasets of all other clients, even in scenarios where clients exhibit substantial dissimilarities. Additionally, this assumption encompasses the majority of personalized federated learning approaches previously proposed in the literature. Beyond the flexible mixture assumption, Section 3.5 makes the following contributions:

- It establishes that, under the mixture assumption, a personalized model is elegantly expressed as a linear combination of a finite number of shared component models. The collaborative learning process involves all clients jointly acquiring knowledge of the shared components, while each client fine-tunes its personalized mixture weights, thereby facilitating personalized federated learning.
- It introduces innovative federated EM-like algorithms, namely  $\text{FedEM}$  tailored for the client-server setting and  $\text{D-FedEM}$  designed for fully decentralized settings.
- It provides rigorous theoretical proofs establishing convergence guarantees for the introduced algorithms. This contributes to a principled and efficient methodology for inferring personalized models for clients not encountered during the training phase.
- It conducts extensive experiments on benchmark datasets for federated learning in Section 3.5, illustrating that our proposed approach consistently produces models that are, on average, more accurate, fairer across clients, and better generalize to unseen clients compared to contemporary state-of-the-art personalized and non-personalized federated learning methods.

After introducing the mixture assumption and  $\text{FedEM}$  algorithm in [Mar+21b], personalized federated learning approaches like  $\text{FedSoft}$  [RJ22],  $\text{FedGMM}$  [Wu+23],  $\text{FedMN}$  [Wan+22a], and  $\text{FedRiCo}$  [Sui+22] have emerged. The  $\text{FedEM}$  approach, originally designed for federated learning, has been applied to characterize internal evasion attacks [Kim+23] and address distribution shifts [GTL23; Jot+23; Zhu+22].

### Personalized Federated Learning through Local Memorization

In Section 3.6, we exploit the ability of deep neural networks to extract high quality vectorial representations (embeddings) from non-tabular data (e.g., images and text) to propose  $\text{kNN-Per}$ , a personalization mechanism based on local memorization.  $\text{kNN-Per}$  combines a global model trained collectively (e.g., via  $\text{FedAvg}$ ) with a  $\text{kNN}$  model on a client’s local datastore. The global model also provides the shared representation used by the local  $\text{kNN}$ . Local memorization at each FL client can capture the client’s local distribution shift with respect to the global distribution. The contributions of Section 3.6 are threefold:

- It proposes  $\text{kNN-Per}$  a simple personalization mechanism based on local memorization.
- It provides generalization bounds for the proposed approach in the case of binary classification
- Through extensive experiments on FL benchmarks, it shows that  $\text{kNN-Per}$  achieves significantly higher accuracy and fairness than state-of-the-art methods.

$\text{kNN-Per}$  offers a simple and effective way to address statistical heterogeneity even in a dynamic environment where client’s data distributions change after training. It is indeed sufficient to update the local datastore with new data without the need to retrain the global model. Moreover, each client can independently tune the local  $\text{kNN}$  to its storage and computing capabilities, partially relieving the most powerful clients from the need to align their model to the weakest ones.

## Federated Learning for Data Streams

Section 4.3 drifts from the standard federated approach consisting in learning from static datasets collected before the start of the training and considers the problem of learning from distributed data streams. The contributions of Section 4.3 are threefold:

- Formulates the problem of federated learning for data streams.
- Proposes and theoretically analyze a general federated algorithm for learning from distributed data streams. Our analysis shows a bias-optimization trade-off: by controlling the relative importance of older samples in comparison to newer ones, one can speed training up at the cost of a larger bias of the learned model, or reduce the bias at the cost of a longer training time. The analysis also provides insights to optimally configure our federated algorithm.
- Empirically demonstrates the relevance of our theoretical results through simulations spanning a wide range of machine learning tasks. In particular, experiments show that “reasonable” ways to extend `FedAvg` to data streams may lead to poor learned models, while our configuration rule consistently leads to almost-optimal performance.

## Online Federated Learning with Mixture Models

Section 4.4 studies online federated learning under the assumption that clients’ data distributions are mixtures of a finite number of unknown underlying distributions with varying mixing weights. Within this context, we introduced *Federated Expectation-Maximization Online Mirror Descent* (FEM-OMD), a federated adaptation of the OMD algorithm, for which the gradient of the cost function is estimated using an EM-like algorithm at each time-step.

The contributions of Section 4.4 are threefold:

- It provides a novel formulation for the problem of online federated learning based on the assumption that clients’ data distributions are mixtures of a finite number of unknown underlying distributions with varying mixing weights. In comparison to previous work, our assumption allows the clients to provably benefit from collaboration, while allowing clients’ data distributions to vary in a potentially (constrained) adversarial manner.
- It proposes *Federated Expectation-Maximization Online Mirror Descent* (FEM-OMD), a federated variant of the OMD algorithm, where the gradient of the cost function is estimated through an EM-like algorithm at each time-step. FEM-OMD leverages all of the data stored across clients to learn the parameters of the underlying distributions using Expectation-Maximization updates, while enabling each client to adapt to the temporal variation of its data distribution. We analyze the regret guarantees of FEM-OMD in the case of well-separated spherical Gaussian mixture models. Specifically, we establish a  $\mathcal{O}(\sqrt{T \log(m)} + T/\sqrt{n})$  regret bound, where  $T$  is the time horizon,  $m$  is the number of the underlying distributions, and  $n$  is the number of samples received by each client.
- Through experimental results on synthetic datasets and FL benchmarks, it demonstrates the effectiveness of our approach in online federated settings and show that our scheme allows the clients to benefit from collaboration.

## 5.2 Perspectives and Future Research Directions

In this manuscript, our emphasis has been on addressing the multifaceted challenges stemming from system, statistical, and temporal heterogeneity within collaborative and federated learning systems. Although we have introduced novel algorithms to alleviate the adverse impacts of these factors, it is crucial to acknowledge that several challenges persistently resist complete resolution. Within this section, we provide a comprehensive overview of these challenges, elucidating their complexities, and suggest potential strategies to navigate them.

### Quantification of Statistical Heterogeneity

In this manuscript, we have explored the various generalization bounds associated with collaborative learning algorithms, all of which encompass a term quantifying the “dissimilarity” between the underlying distributions of clients. For example, Proposition 1.2.5 reveals that the generalization error of the global model at a specific client is upper-bounded by an expression featuring a term related to the label discrepancy between the average distribution and the underlying distribution of that client.

Similar dependencies arise in the generalization bound of  $k\text{NN-Per}$  (as per Theorem 3.6.1) and the generalization bound of Algorithm 13 designed for learning from distributed data streams (refer to Theorem 4.3.1 and Corollary 4.3.5'). Notably, the generalization bound of Algorithm 13 introduces a pairwise label discrepancy among clients' underlying distributions. The quantification of statistical heterogeneity plays a pivotal role in determining the hyperparameters of these algorithms, such as the interpolation parameter  $\lambda$  for  $k\text{NN-Per}$  and the samples' weights  $\lambda$  for Algorithm 13.

Beyond the algorithms elucidated in this manuscript, other works (e.g., [Sui+22; EMS22; DKM20]) that examine the statistical learning properties of collaborative learning often hinge on prior knowledge of some form of pseudo-distance between clients' underlying distributions. For instance, [EMS22] proposes a gradient filtering approach for collaborative learning, where clients filter and aggregate stochastic gradients received from other clients based on the knowledge of the Integral Probability Metric associated with the gradient of the loss function.

However, the quantification of statistical heterogeneity in collaborative learning remains poorly understood. In this context, we believe that two fundamental questions warrant attention: 1) how to choose among the different notions of pseudo-distance? 2) how to estimate the dissimilarity between two distributions for a specific notion of pseudo-distance?

To the best of our knowledge, the first question has not been thoroughly investigated. On the other hand, the second question has garnered significant interest within the federated learning community. For instance, [Kim+23] quantifies data distribution similarity among clients based on our introduced mixture assumption (in Section 3.5), while [EMS22] estimates pairwise distribution distances under the structural assumption of a low-dimensional linear representation (introduced in [Col+21]). Nevertheless, it is crucial to note that the current attempts to address the second question may fall short of providing comprehensive solutions, as they only cover particular assumptions on the clients' underlying distributions.

### Data-Heterogeneity-Aware Topology Design

In Chapter 2.1, our focus centered on addressing the topology design challenge in cross-silo federated learning. Utilizing the framework of max-plus linear systems, we aimed to compute

the system throughput—the number of communication rounds per unit of time. Our contribution included the development of practical algorithms that, based on measurable network characteristics, identify a topology maximizing throughput or offering provable throughput guarantees. However, these algorithms did not account for the statistical heterogeneity across different clients; rather, they concentrated solely on optimizing the system throughput.

Recent works, such as [Dan+22; Le +23], have begun incorporating data heterogeneity considerations into the crafting of fully-decentralized learning topologies. Notably, [Le +23] introduces a novel concept, termed neighborhood heterogeneity, and highlights its crucial role in influencing the convergence rate of decentralized SGD. This analysis sheds light on the intricate interplay between communication topology and statistical heterogeneity. However, minimizing neighborhood heterogeneity in a general setting proves to be challenging without additional statistical assumptions and can only be optimized in specific cases, particularly in scenarios involving classification with label skew. We believe that further exploration of this research avenue is essential.

### Privacy-Preserving Personalized Federated Learning

As we have seen in Chapter 1, privacy remains a significant concern in federated learning even when data is kept locally. One of the foremost concerns is the potential leakage of private data through model updates and gradients exchanged between the central server and participating devices [McM+17]. Even though efforts are made to anonymize these updates, there exists a risk of reverse engineering and information inference, potentially revealing sensitive attributes about users [SS15]. Traditional differential privacy mechanisms, while effective at a global level [GKN18; Bel+18], confront novel challenges in the personalized landscape.

In personalized FL, the heterogeneity of user data, varying levels of individual sensitivity, and dynamic participation patterns introduce complexities that may render standard differential privacy mechanisms less effective. For example, in our FedEM approach, each client needs to update and transmit  $M$  components at each round, meaning that it is potentially revealing more information to the server in comparison with the FedAvg baseline algorithm. However, as we previously remarked in Section 3.5.7, some features of our FedEM approach may be beneficial for privacy, e.g., the fact that personalized weights are kept locally and that all users contribute to all shared models. It is worth highlighting that the privacy guarantees of other prominent personalized federated learning paradigms, such as ClusteredFL [SMS20], APFL [DKM20], and FedRep [Col+21], are not yet fully understood. To the best of our knowledge, a unified framework for privacy-preserving federated learning is still elusive. We contend that formulating such a framework merits considerable attention from the personalized federated learning community. Addressing this gap could pave the way for more robust and standardized approaches to privacy preservation in personalized FL scenarios.

### Local Cache Update Rules for Federated Learning

In this manuscript, we considered two scenarios where each client is associated with a local memory/caches that can be used to store data samples, as it is the case in Section 4.3, or their embeddings, as it is the case in Section 3.6. In both cases, we have discussed how the caching policy—e.g., the local memory update rule—may influence the performance of the algorithms we proposed in Section 3.6 and Section 4.3. Both algorithms consider simple memory update rules, usually variations of the *first-in-first-out* (FIFO) update rule. The new work [WBX23], in addition

to FIFO, considers two local cache update rules; Static Ratio Selective Replacement (SRSR), and Dynamic Ratio Selective Replacement (DRSR). However, all the aforementioned update rules are deterministic and do not depend on the features or the labels of the samples currently in the memory. We deem it important to answer the following question: *how do different memory update rules affect the performance of such algorithms, and how can we design sample dependent caching policies?*

### **Incentivizing Client Participation in Federated Learning**

One interesting avenue to explore involves investigating the adoption of federated learning (FL) within a setting where users have the option to opt out of the federation. This raises questions about the stability of the federation and the economic incentives that can be developed to encourage user participation. Game-theoretic studies of federated learning’s stability can provide valuable insights into this aspect, drawing on existing research such as [Tu+22], [DK21], and [Blu+21]. Additionally, investigating economic incentives for users, building upon [Kan+19] and [Cho+22], can offer novel approaches to encourage engagement and active participation. Another intriguing direction for future exploration involves developing novel FL algorithms that empower clients to learn personalized models adapted to their local data distribution. By addressing open issues related to quantifying statistical heterogeneity across clients and determining the value of each client’s dataset, it becomes possible to unlock the potential of personalized models within the FL framework. These challenges gain further significance in the context of the evolving data economy, which encompasses various online data exchange platforms like AWS data exchange. The inherent complexity of these challenges is magnified within the FL setting, where participants only have access to their own data.

## **5.3 Concluding Reflections**

In concluding this thesis, I humbly acknowledge the modest role this work plays within the vast landscape of knowledge. The insights gained and the contributions made are but small steps forward in the ongoing journey of understanding. As we reflect on the limitations and possibilities outlined herein, it is my sincere hope that future researchers will build upon these foundations with humility, recognizing the collaborative nature of academic progress. In the grand tapestry of research, each thread, no matter how modest, contributes to the richness of the whole.



# Bibliography

---

- [Aba+16] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. “Deep Learning with Differential Privacy”. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’16. Vienna, Austria: Association for Computing Machinery, 2016, pp. 308–318. ISBN: 9781450341394. DOI: 10.1145/2976749.2978318. URL: <https://doi.org/10.1145/2976749.2978318>.
- [Aca+21] Durmus Alp Emre Acar, Yue Zhao, Ruizhao Zhu, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. “Debiasing Model Updates for Improving Personalized Federated Training”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 21–31. URL: <https://proceedings.mlr.press/v139/acar21a.html>.
- [Ach+21] Idan Achituve, Aviv Shamsian, Aviv Navon, Gal Chechik, and Ethan Fetaya. “Personalized Federated Learning With Gaussian Processes”. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 8392–8406. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2021/file/46d0671dd4117ea366031f87f3aa0093-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2021/file/46d0671dd4117ea366031f87f3aa0093-Paper.pdf).
- [AR19] Patrick J. Andersen and Charl J. Ras. “Algorithms for Euclidean Degree Bounded Spanning Tree Problems”. In: *Int. J. Comput. Geometry Appl.* 29.2 (2019), pp. 121–160.
- [AR16] Patrick J. Andersen and Charl J. Ras. “Minimum bottleneck spanning trees with degree bounds”. In: *Networks* 68.4 (2016), pp. 302–314. DOI: 10.1002/net.21710.
- [AZ05] Rie Kubota Ando and Tong Zhang. “A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data”. In: *Journal of Machine Learning Research* 6.61 (2005), pp. 1817–1853.
- [App19] Apple. *Designing for privacy (video and slide deck)*. <https://developer.apple.com/videos/play/wwdc2019/708>[Retrieved: Aug 2023]. 2019.
- [App+07] David L. Applegate, Robert E. Bixby, Vasek Chvatal, and William J. Cook. *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)*. USA: Princeton University Press, 2007. ISBN: 0691129932.
- [Ara+16] Toshinori Araki, Jun Furukawa, Yehuda Lindell, Ariel Nof, and Kazuma Ohara. “High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority”. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’16. Vienna, Austria: Association for Computing Machinery, 2016, pp. 805–817. ISBN: 9781450341394. DOI: 10.1145/2976749.2978331. URL: <https://doi.org/10.1145/2976749.2978331>.



- [Ass+19] Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Mike Rabbat. “Stochastic Gradient Push for Distributed Deep Learning”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 344–353. URL: <https://proceedings.mlr.press/v97/assran19a.html>.
- [AWS20] AWS. *The AWS Cloud in North America*. [https://aws.amazon.com/about-aws/global-infrastructure/?nc1=h\\_ls](https://aws.amazon.com/about-aws/global-infrastructure/?nc1=h_ls)[Retrieved: Aug 2020]. 2020.
- [BW01] Shivnath Babu and Jennifer Widom. “Continuous Queries over Data Streams”. In: *SIGMOD Rec.* 30.3 (Sept. 2001), pp. 109–120. ISSN: 0163-5808. DOI: 10.1145/603867.603884. URL: <https://doi.org/10.1145/603867.603884>.
- [Bac92] F. Baccelli. *Synchronization and Linearity: An Algebra for Discrete Event Systems*. Probability and Statistics Series. Wiley, 1992. ISBN: 9780471936091. URL: <https://books.google.co.ma/books?id=l8FnQgAACAAJ>.
- [Bad+15] Aldo Badano, Craig Revie, Andrew Casertano, Wei-Chung Cheng, Phil Green, Tom Kimpe, Elizabeth Krupinski, Christye Sisson, Stein Skrøvseth, Darren Treanor, et al. “Consistency and standardization of color in medical imaging: a consensus report”. In: *Journal of digital imaging* 28.1 (2015), pp. 41–52.
- [BPS19] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. “Differential privacy has disparate impact on model accuracy”. In: *Advances in neural information processing systems* 32 (2019).
- [Bea+21] Martin Beaussart, Felix Grimberg, Mary-Anne Hartley, and Martin Jaggi. *WAF-FLE: Weighted Averaging for Personalized Federated Learning*. 2021. arXiv: 2110.06978 [cs.LG].
- [Bel+18] Aurélien Bellet, Rachid Guerraoui, Mahsa Taziki, and Marc Tommasi. “Personalized and Private Peer-to-Peer Machine Learning”. In: *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*. Ed. by Amos Storkey and Fernando Perez-Cruz. Vol. 84. Proceedings of Machine Learning Research. PMLR, Apr. 2018, pp. 473–481. URL: <https://proceedings.mlr.press/v84/bellet18a.html>.
- [BHS15] Aurélien Bellet, Amaury Habrard, and Marc Sebban. *Metric Learning*. Vol. 9. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers (USA), Synthesis Lectures on Artificial Intelligence and Machine Learning, pp 1-151, Jan. 2015, pp. 1–151. DOI: 10.2200/S00626ED1V01Y201501AIM030. URL: <https://hal.archives-ouvertes.fr/hal-01121733>.
- [BLP08] Shai Ben-David, Tyler Lu, and D. Pál. “Does Unlabeled Data Provably Help? Worst-case Analysis of the Sample Complexity of Semi-Supervised Learning”. In: *COLT*. 2008.
- [Ben12] Yoshua Bengio. “Deep learning of representations for unsupervised and transfer learning”. In: *Proceedings of ICML workshop on unsupervised and transfer learning*. JMLR Workshop and Conference Proceedings. 2012, pp. 17–36.
- [Ber+19] Daniel Bernau, Philip-William Grassal, Jonas Robl, and Florian Kerschbaum. “Assessing differentially private deep learning with membership inference”. In: *arXiv preprint arXiv:1912.11328* (2019).

- [Bez+22] Aleksandr Beznosikov, Samuel Horváth, Peter Richtárik, and Mher Safaryan. *On Biased Compression for Distributed Learning*. 2022. arXiv: 2002.12410 [cs.LG].
- [Bha+22] Romil Bhardwaj, Zhengxu Xia, Ganesh Ananthanarayanan, Junchen Jiang, Yuanchao Shu, Nikolaos Karianakis, Kevin Hsieh, Paramvir Bahl, and Ion Stoica. “Ekya: Continuous Learning of Video Analytics Models on Edge Compute Servers”. In: *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*. Renton, WA: USENIX Association, Apr. 2022, pp. 119–135. ISBN: 978-1-939133-27-4. URL: <https://www.usenix.org/conference/nsdi22/presentation/bhardwaj>.
- [Bic+93] Peter J Bickel, Chris AJ Klaassen, Peter J Bickel, Ya’acov Ritov, J Klaassen, Jon A Wellner, and YA’Acov Ritov. *Efficient and adaptive estimation for semiparametric models*. Vol. 4. Johns Hopkins University Press Baltimore, 1993.
- [Bla+17] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. “Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: <https://proceedings.neurips.cc/paper/2017/file/f4b9ec30ad9f68f89b29639786cb62ef-Paper.pdf>.
- [Blu+21] Avrim Blum, Nika Haghtalab, Richard Lanus Phillips, and Han Shao. “One for One, or All for All: Equilibria and Optimality of Collaboration in Federated Learning”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 1005–1014. URL: <https://proceedings.mlr.press/v139/blum21a.html>.
- [Bon+19] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konečný, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roslander. “Towards Federated Learning at Scale: System Design”. In: *Proceedings of Machine Learning and Systems*. Ed. by A. Talwalkar, V. Smith, and M. Zaharia. Vol. 1. 2019, pp. 374–388. URL: [https://proceedings.mlsys.org/paper\\_files/paper/2019/file/7b770da633baf74895be22a8807f1a8f-Paper.pdf](https://proceedings.mlsys.org/paper_files/paper/2019/file/7b770da633baf74895be22a8807f1a8f-Paper.pdf).
- [Bon+17] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. “Practical Secure Aggregation for Privacy-Preserving Machine Learning”. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. CCS ’17*. Dallas, Texas, USA: Association for Computing Machinery, 2017, pp. 1175–1191. ISBN: 9781450349468. DOI: 10.1145/3133956.3133982. URL: <https://doi.org/10.1145/3133956.3133982>.
- [Bon+12] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. “Fog Computing and Its Role in the Internet of Things”. In: *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing. MCC ’12*. Helsinki, Finland: Association for Computing Machinery, 2012, pp. 13–16. ISBN: 9781450315197.

- DOI: 10.1145/2342509.2342513. URL: <https://doi.org/10.1145/2342509.2342513>.
- [BCN18] Léon Bottou, Frank E Curtis, and Jorge Nocedal. “Optimization Methods for Large-Scale Machine Learning”. In: *Siam Review* 60.2 (2018), pp. 223–311.
- [BDX03] Stephen Boyd, Persi Diaconis, and Lin Xiao. “Fastest Mixing Markov Chain on A Graph”. In: *SIAM REVIEW* 46 (2003), pp. 667–689.
- [BV04] Stephen Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, Mar. 2004. ISBN: 9780511804441. DOI: 10.1017/CBO9780511804441. (Visited on 05/23/2023).
- [Bra+18] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. 2018. URL: <http://github.com/google/jax>.
- [Bra12] Zvika Brakerski. “Fully homomorphic encryption without modulus switching from classical GapSVP”. In: *Annual Cryptology Conference*. Springer. 2012, pp. 868–886.
- [Bra08] Ulrik Brandes. “On variants of shortest-path betweenness centrality and their generic computation”. In: *Social Networks* 30.2 (2008), pp. 136–145. ISSN: 0378-8733. DOI: <https://doi.org/10.1016/j.socnet.2007.11.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0378873307000731>.
- [BRH12] T. Brunsch, J. Raisch, and L. Hardouin. “Modeling and control of high-throughput screening systems”. In: *Control Engineering Practice* 20.1 (2012). Special Section: IFAC Conference on Analysis and Design of Hybrid Systems (ADHS’09) in Zaragoza, Spain, 16th-18th September, 2009, pp. 14–23. ISSN: 0967-0661. DOI: <https://doi.org/10.1016/j.conengprac.2010.12.006>. URL: <https://www.sciencedirect.com/science/article/pii/S0967066110002662>.
- [Bub15] Sébastien Bubeck. *Convex Optimization: Algorithms and Complexity*. 2015. arXiv: 1405.4980 [math.OC].
- [Cal+19] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. *LEAF: A Benchmark for Federated Settings*. 2019. arXiv: 1812.01097 [cs.LG].
- [CC96] Robert L. Carter and Mark E. Crovella. “Measuring Bottleneck Link Speed in Packet-Switched Networks”. In: *Performance Evaluation* 27-28 (1996), pp. 297–318. ISSN: 0166-5316. DOI: [https://doi.org/10.1016/S0166-5316\(96\)90032-2](https://doi.org/10.1016/S0166-5316(96)90032-2). URL: <http://www.sciencedirect.com/science/article/pii/S0166531696900322>.
- [CJB04] Rich Caruana, Thorsten Joachims, and Lars Backstrom. “KDD-Cup 2004: results and analysis”. In: *ACM SIGKDD Explorations Newsletter* 6.2 (2004), pp. 95–108.
- [CLG00] Rich Caruana, Steve Lawrence, and C Giles. “Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping”. In: *Advances in neural information processing systems* 13 (2000).

- [CMK01] V. Chandra, S.R. Mohanty, and R. Kumar. “Automated control synthesis for an assembly line using discrete event system control theory”. In: *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*. Vol. 6. 2001, 4956–4961 vol.6. DOI: 10.1109/ACC.2001.945770.
- [Cha+22] El Mahdi Chayti, Sai Praneeth Karimireddy, Sebastian U. Stich, Nicolas Flammarion, and Martin Jaggi. *Linear Speedup in Personalized Collaborative Learning*. 2022. arXiv: 2111.05968 [cs.LG].
- [Che+20a] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. “Gan-leaks: A taxonomy of membership inference attacks against generative models”. In: *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*. 2020, pp. 343–362.
- [CC22] Hong-You Chen and Wei-Lun Chao. “On Bridging Generic and Personalized Federated Learning for Image Classification”. In: *International Conference on Learning Representations*. 2022. URL: <https://openreview.net/forum?id=I1hQbx10Kxn>.
- [Che+19] Mingqing Chen, Rajiv Mathews, Tom Ouyang, and Françoise Beaufays. *Federated Learning Of Out-Of-Vocabulary Words*. 2019. arXiv: 1903.10635 [cs.CL].
- [CHR22] Wenlin Chen, Samuel Horváth, and Peter Richtárik. “Optimal Client Sampling for Federated Learning”. In: *Transactions on Machine Learning Research* (Aug. 2022). ISSN: 2835-8856. (Visited on 05/23/2023).
- [Che+20b] Yujing Chen, Yue Ning, Martin Slawski, and Huzefa Rangwala. “Asynchronous Online Federated Learning for Edge Devices with Non-IID Data”. In: *2020 IEEE International Conference on Big Data (Big Data)*. IEEE. 2020, pp. 15–24.
- [CCD22] Gary Cheng, Karan Chadha, and John Duchi. *Federated Asymptotics: a model to compare federated learning algorithms*. 2022. arXiv: 2108.07313 [cs.LG].
- [Cho+14] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. “On the Properties of Neural Machine Translation: Encoder-Decoder Approaches”. In: *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*. Ed. by Dekai Wu, Marine Carpuat, Xavier Carreras, and Eva Maria Vecchi. Association for Computational Linguistics, 2014, pp. 103–111. DOI: 10.3115/v1/W14-4012. URL: <https://www.aclweb.org/anthology/W14-4012/>.
- [Cho+22] Yae Jee Cho, Divyansh Jhunjhunwala, Tian Li, Virginia Smith, and Gauri Joshi. “To Federate or Not To Federate: Incentivizing Client Participation in Federated Learning”. In: *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*. 2022. URL: <https://openreview.net/forum?id=pG08eM0CQba>.
- [Cho+21] Christopher A Choquette-Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. “Label-only membership inference attacks”. In: *International conference on machine learning*. PMLR. 2021, pp. 1964–1974.

- [Çiç+16] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. “3D U-Net: learning dense volumetric segmentation from sparse annotation”. In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2016, pp. 424–432.
- [Coh+17] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. “EMNIST: Extending MNIST to handwritten letters”. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2017, pp. 2921–2926.
- [Col+21] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. “Exploiting Shared Representations for Personalized Federated Learning”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 2089–2099. URL: <https://proceedings.mlr.press/v139/collins21a.html>.
- [Com+16] Federal Communications Commission et al. “Protecting the Privacy of Customers of Broadband and Other Telecommunications Service (2016)”. In: (2016).
- [COR19] CORDIS. Machine Learning Ledger Orchestration for Drug Discovery (MELLODY). [https://cordis.europa.eu/project/id/831472?WT.mc\\_id=RSS-Feed&WT.rss\\_f=project&WT.rss\\_a=223634&WT.rss\\_ev=a](https://cordis.europa.eu/project/id/831472?WT.mc_id=RSS-Feed&WT.rss_f=project&WT.rss_a=223634&WT.rss_ev=a) [Retrieved: Aug 2019]. 2019.
- [CBB21] Luca Corinzia, Ami Beuret, and Joachim M. Buhmann. *Variational Federated Multi-Task Learning*. 2021. arXiv: 1906.06268 [cs.LG].
- [CLT14] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. “Scale-invariant fully homomorphic encryption over the integers”. In: *Public-Key Cryptography–PKC 2014: 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings 17*. Springer. 2014, pp. 311–328.
- [CMM10] Corinna Cortes, Yishay Mansour, and Mehryar Mohri. “Learning Bounds for Importance Weighting”. In: *Advances in Neural Information Processing Systems*. Ed. by J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta. Vol. 23. Curran Associates, Inc., 2010. URL: <https://proceedings.neurips.cc/paper/2010/file/59c33016884a62116be975a9bb8257e3-Paper.pdf>.
- [Cor+08] Corinna Cortes, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh. “Sample Selection Bias Correction Theory”. In: *ALT*. 2008.
- [Cou+19] Pierre Courtiol, Charles Maussion, Matahi Moarii, Elodie Pronier, Samuel Pilcer, Meriem Sefta, Pierre Manceron, Sylvain Toldo, Mikhail Zaslavskiy, Nolwenn Le Stang, et al. “Deep Learning-Based Classification of Mesothelioma Improves Prediction of Patient Outcome”. In: *Nature medicine* 25.10 (2019), pp. 1519–1525.
- [Cox72] David R Cox. “Regression models and life-tables”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 34.2 (1972), pp. 187–202.

- [CB22] Edwige Cyffers and Aurélien Bellet. “Privacy Amplification by Decentralization”. In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Ed. by Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera. Vol. 151. Proceedings of Machine Learning Research. PMLR, Mar. 2022, pp. 5334–5353. URL: <https://proceedings.mlr.press/v151/cyffers22a.html>.
- [DM22] Shuang Dai and Fanlin Meng. “Addressing Modern and Practical Challenges in Machine Learning: A Survey of Online Federated and Transfer Learning”. In: *arXiv preprint arXiv:2202.03070* (2022).
- [Dam+20] Georgios Damaskinos, Rachid Guerraoui, Anne-Marie Kermarrec, Vlad Nitu, Rhicheek Patra, and François Taiani. “FLeet: Online Federated Learning via Staleness Awareness and Performance Prediction”. In: *ACM/IFIP Middleware conference*. 2020.
- [Dan+22] Yatin Dandi, Anastasia Koloskova, Martin Jaggi, and Sebastian U Stich. “Data-heterogeneity-aware Mixing for Decentralized Learning”. In: *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*. 2022.
- [DSS13] Malte Darnstädt, H. U. Simon, and Balázs Szörényi. “Unlabeled Data Does Provably Help”. In: *STACS*. 2013.
- [DG98] A. Dasdan and R.K. Gupta. “Faster Maximum and Minimum Mean Cycle Algorithms for System-Performance Analysis”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 17.10 (1998), pp. 889–899. DOI: 10.1109/43.728912.
- [Das99] S. Dasgupta. “Learning Mixtures of Gaussians”. In: *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*. 1999, pp. 634–644. DOI: 10.1109/SFFCS.1999.814639.
- [Dav+14] Xavier David-Henriet2014, Laurent Hardouin, Jörg Raisch, and Bertrand Cottenceau. “Holding Time Maximization Preserving Output Performance for Timed Event Graphs”. In: *IEEE Transactions on Automatic Control* 59.7 (2014), pp. 1968–1973. DOI: 10.1109/TAC.2013.2297202.
- [Day+21] Ittai Dayan, Holger R Roth, Aoxiao Zhong, Ahmed Harouni, Amilcare Gentili, Anas Z Abidin, Andrew Liu, Anthony Beardsworth Costa, Bradford J Wood, Chien-Sung Tsai, et al. “Federated learning for predicting clinical outcomes in patients with COVID-19”. In: *Nature medicine* 27.10 (2021), pp. 1735–1743.
- [De +16] Gianmarco De Francisci Morales, Albert Bifet, Latifur Khan, Joao Gama, and Wei Fan. “IoT Big Data Stream Mining”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD ’16*. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 2119–2120. ISBN: 9781450342322. DOI: 10.1145/2939672.2945385. URL: <https://doi.org/10.1145/2939672.2945385>.
- [Den+09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *CVPR09*. 2009.
- [DKM20] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. *Adaptive Personalized Federated Learning*. 2020. arXiv: 2003.13461 [cs.LG].

- [DS16] Paolo Di Lorenzo and Gesualdo Scutari. “Distributed Nonconvex Optimization over Time-Varying Networks”. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, pp. 4124–4128. DOI: 10.1109/ICASSP.2016.7472453.
- [DDT20] Enmao Diao, Jie Ding, and Vahid Tarokh. “HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients”. In: *International Conference on Learning Representations*. 2020.
- [Die+21] Aymeric Dieuleveut, Gersende Fort, Eric Moulines, and Geneviève Robin. “Federated-EM with heterogeneity mitigation and variance reduction”. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 29553–29566. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2021/file/f740c8d9c193f16d8a07d3a8a751d13f-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2021/file/f740c8d9c193f16d8a07d3a8a751d13f-Paper.pdf).
- [DW22] Shu Ding and Wei Wang. “Collaborative Learning by Detecting Collaboration Partners”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 15629–15641. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/646ca7b994bc46afe33d680dbe7ed67a-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/646ca7b994bc46afe33d680dbe7ed67a-Paper-Conference.pdf).
- [Din+20] Yucheng Ding, Chaoyue Niu, Yikai Yan, Zhenzhe Zheng, Fan Wu, Guihai Chen, Shaojie Tang, and Rongfei Jia. *Distributed Optimization over Block-Cyclic Data*. 2020. arXiv: 2002.07454 [cs.LG].
- [Din+22] Canh T. Dinh, Tung T. Vu, Nguyen H. Tran, Minh N. Dao, and Hongyu Zhang. “A New Look and Convergence Rate of Federated Multitask Learning With Laplacian Regularization”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2022), pp. 1–11. DOI: 10.1109/TNNLS.2022.3224252.
- [Doa20] Thinh T. Doan. *Local Stochastic Approximation: A Unified View of Federated Learning and Distributed Multi-Task Reinforcement Learning Algorithms*. 2020. arXiv: 2006.13460 [cs.LG].
- [Doa+20a] Thinh T. Doan, Lam M. Nguyen, Nhan H. Pham, and Justin Romberg. *Convergence Rates of Accelerated Markov Gradient Descent with Applications in Reinforcement Learning*. 2020. arXiv: 2002.02873 [math.OC].
- [Doa+20b] Thinh T. Doan, Lam M. Nguyen, Nhan H. Pham, and Justin Romberg. *Finite-Time Analysis of Stochastic Gradient Descent under Markov Randomness*. 2020. arXiv: 2003.10973 [math.OC].
- [DK21] Kate Donahue and Jon Kleinberg. “Model-sharing Games: Analyzing Federated Learning Under Voluntary Participation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.6 (May 2021), pp. 5303–5311. DOI: 10.1609/aaai.v35i6.16669. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/16669>.

- [DAW12] John C. Duchi, Alekh Agarwal, and Martin J. Wainwright. “Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling”. In: *IEEE Transactions on Automatic Control* 57.3 (2012), pp. 592–606. DOI: 10.1109/TAC.2011.2161027.
- [DR+14] Cynthia Dwork, Aaron Roth, et al. “The algorithmic foundations of differential privacy.” In: *Found. Trends Theor. Comput. Sci.* 9.3-4 (2014), pp. 211–407.
- [Eic+19] Hubert Eichner, Tomer Koren, Brendan McMahan, Nathan Srebro, and Kunal Talwar. “Semi-Cyclic Stochastic Gradient Descent”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 1764–1773. URL: <https://proceedings.mlr.press/v97/eichner19a.html>.
- [Elv17] Stacy-Ann Elvy. “Paying for privacy and the personal data economy”. In: *Colum. L. Rev.* 117 (2017), p. 1369.
- [ER59] P. Erdős and A. Rényi. “On Random Graphs I”. In: *Publicationes Mathematicae Debrecen* 6 (1959), p. 290.
- [EMS22] Mathieu Even, Laurent Massoulié, and Kevin Scaman. “On Sample Optimality in Personalized Collaborative and Federated Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 212–225. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/01cea7793f3c68af2e4989fc66bf8fb0-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/01cea7793f3c68af2e4989fc66bf8fb0-Paper-Conference.pdf).
- [FMO20] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. “Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 3557–3568. URL: <https://proceedings.neurips.cc/paper/2020/file/24389bfe4fe2eba8bf9aa9203a44cdad-Paper.pdf>.
- [FGQ11] N. Farhi, M. Goursat, and J.-P. Quadrat. “The traffic phases of road networks”. In: *Transportation Research Part C: Emerging Technologies* 19.1 (2011), pp. 85–102. ISSN: 0968-090X. DOI: <https://doi.org/10.1016/j.trc.2010.03.011>. URL: <https://www.sciencedirect.com/science/article/pii/S0968090X10000379>.
- [Fra+21] Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. “Clustered Sampling: Low-Variance and Improved Representativity for Clients Selection in Federated Learning”. In: *Proceedings of the 38th International Conference on Machine Learning*. PMLR, July 2021, pp. 3407–3416. (Visited on 05/23/2023).
- [FJR15] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. “Model inversion attacks that exploit confidence information and basic countermeasures”. In: *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. 2015, pp. 1322–1333.



- [Fu+20] Yu Fu, Alexander W Jung, Ramon Viñas Torne, Santiago Gonzalez, Harald Vöhringer, Artem Shmatko, Lucy R Yates, Mercedes Jimenez-Linan, Luiza Moore, and Moritz Gerstung. “Pan-cancer computational histopathology reveals mutations, tumor composition and prognosis”. In: *Nature Cancer* 1.8 (2020), pp. 800–810.
- [GOD22] Georgi Ganey, Bristena Oprisanu, and Emiliano De Cristofaro. “Robin Hood and Matthew Effects: Differential Privacy Has Disparate Impact on Synthetic Data”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 6944–6959.
- [Gan+18] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. “Property inference attacks on fully connected neural networks using permutation invariant representations”. In: *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. 2018, pp. 619–633.
- [Gao+22] Yan Gao, Titouan Parcollet, Salah Zaiem, Javier Fernandez-Marques, Pedro PB de Gusmao, Daniel J Beutel, and Nicholas D Lane. “End-to-end Speech Recognition from Federated Acoustic Models”. In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2022, pp. 7227–7231.
- [Gar+21] Abhinav Garg, Naman Shukla, Lavanya Marla, and Sriram Somanchi. *Distribution Shift in Airline Customer Behavior during COVID-19*. 2021. arXiv: 2111.14938 [cs.LG].
- [GG23] Guillaume Garrigos and Robert M. Gower. *Handbook of Convergence Theorems for (Stochastic) Gradient Methods*. 2023. arXiv: 2301.11235 [math.OA].
- [Gau95] S. Gaubert. “Resource Optimization and (min,+) Spectral Theory”. In: *IEEE Transactions on Automatic Control* 40.11 (1995), pp. 1931–1934. DOI: 10.1109/9.471219.
- [GHK15] Rong Ge, Qingqing Huang, and Sham M. Kakade. “Learning Mixtures of Gaussians in High Dimensions”. In: *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*. STOC ’15. Portland, Oregon, USA: Association for Computing Machinery, 2015, pp. 761–770. ISBN: 9781450335362. DOI: 10.1145/2746539.2746616. URL: <https://doi.org/10.1145/2746539.2746616>.
- [20a] *GÉANT - the pan-european research and education network*. <https://www.geant.org/Networks>[Retrieved: Aug 2020]. 2020.
- [Gen09] Craig Gentry. “Fully Homomorphic Encryption Using Ideal Lattices”. In: *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*. STOC ’09. Bethesda, MD, USA: Association for Computing Machinery, 2009, pp. 169–178. ISBN: 9781605585062. DOI: 10.1145/1536414.1536440. URL: <https://doi.org/10.1145/1536414.1536440>.
- [GKN18] Robin C. Geyer, Tassilo Klein, and Moin Nabi. *Differentially Private Federated Learning: A Client Level Perspective*. 2018. arXiv: 1712.07557 [cs.CR].

- [Gho+20] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. “An Efficient Framework for Clustered Federated Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 19586–19597. URL: <https://proceedings.neurips.cc/paper/2020/file/e32cc80bf07915058ce90722ee17bb71-Paper.pdf>.
- [GBH09] Alec Go, Richa Bhayani, and Lei Huang. “Twitter Sentiment Classification using Distant Supervision”. In: *Processing (2009)*, pp. 1–6. URL: <http://www.stanford.edu/~alecmgo/papers/TwitterDistantSupervision09.pdf>.
- [Goo+20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative adversarial networks”. In: *Communications of the ACM* 63.11 (2020), pp. 139–144.
- [Goo+15] Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. *An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks*. 2015. arXiv: 1312.6211 [stat.ML].
- [Göp+19] Christina Göpfert, Shai Ben-David, Olivier Bousquet, Sylvain Gelly, Ilya Tolstikhin, and Ruth Uerner. “When can unlabeled data improve the learning rate?” In: *Conference on Learning Theory*. PMLR, 2019, pp. 1500–1518.
- [Gov98] Rob MP Goverde. “The max-plus algebra approach to railway timetable design”. In: *WIT Transactions on The Built Environment* 37 (1998).
- [Gre+15] Edward Grefenstette, Karl Moritz Hermann, Mustafa Suleyman, and Phil Blunsom. “Learning to Transduce with Unbounded Memory”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., 2015. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2015/file/b9d487a30398d42ecff55c228ed5652b-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2015/file/b9d487a30398d42ecff55c228ed5652b-Paper.pdf).
- [Gri+21] Felix Grimberg, Mary-Anne Hartley, Sai P. Karimireddy, and Martin Jaggi. *Optimal Model Averaging: Towards Personalized Collaborative Learning*. 2021. arXiv: 2110.12946 [cs.LG].
- [Gue+04] Bamba Gueye, Artur Ziviani, Mark Crovella, and Serge Fdida. “Constraint-Based Geolocation of Internet Hosts”. In: *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*. IMC ’04. Taormina, Sicily, Italy: Association for Computing Machinery, 2004, pp. 288–293. ISBN: 1581138210. DOI: 10.1145/1028788.1028828. URL: <https://doi.org/10.1145/1028788.1028828>.
- [GLT23] Yongxin Guo, Tao Lin, and Xiaoying Tang. *Towards Federated Learning on Time-Evolving Heterogeneous Data*. 2023. arXiv: 2112.13246 [cs.LG].
- [GTL23] Yongxin Guo, Xiaoying Tang, and Tao Lin. *FedRC: Tackling Diverse Distribution Shifts Challenge in Federated Learning by Robust Clustering*. 2023. arXiv: 2301.12379 [cs.LG].

- [Gup+17] Chirag Gupta, Arun Sai Suggala, Ankit Goyal, Harsha Vardhan Simhadri, Bhargavi Paranjape, Ashish Kumar, Saurabh Goyal, Raghavendra Udupa, Manik Varma, and Prateek Jain. “ProtoNN: Compressed and Accurate kNN for Resource-scarce Devices”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, Aug. 2017, pp. 1331–1340. URL: <https://proceedings.mlr.press/v70/gupta17a.html>.
- [GP06] Gregory Gutin and Abraham P Punnen. *The traveling salesman problem and its variations*. Vol. 12. Springer Science & Business Media, 2006.
- [Haa+21] Kevin de Haan, Yijie Zhang, Jonathan E Zuckerman, Tairan Liu, Anthony E Sisk, Miguel FP Diaz, Kuang-Yu Jen, Alexander Nobori, Sofia Liou, Sarah Zhang, et al. “Deep learning-based transformation of H&E stained tissues into special stains”. In: *Nature communications* 12.1 (2021), pp. 1–13.
- [Had+21] Farzin Haddadpour, Mohammad Mahdi Kamani, Aryan Mokhtari, and Mehrdad Mahdavi. “Federated Learning with Compression: Unified Analysis and Sharp Guarantees”. In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Ed. by Arindam Banerjee and Kenji Fukumizu. Vol. 130. Proceedings of Machine Learning Research. PMLR, Apr. 2021, pp. 2350–2358. URL: <https://proceedings.mlr.press/v130/haddadpour21a.html>.
- [HBB06] Peter F Hahn, Michael A Blake, and Giles WL Boland. “Adrenal lesions: attenuation measurement differences between CT scanners”. In: *Radiology* 240.2 (2006), pp. 458–463.
- [Han+20a] Catherine Han, Irwin Reyes, Álvaro Feal, Joel Reardon, Primal Wijesekera, Narseo Vallina-Rodriguez, Amit Elazar, Kenneth A Bamberger, and Serge Egelman. “The price is (not) right: Comparing privacy in free and paid apps”. In: *Proceedings on Privacy Enhancing Technologies* 2020.3 (2020).
- [Han+20b] Filip Hanzely, Slavomír Hanzely, Samuel Horváth, and Peter Richtárik. “Lower Bounds and Optimal Algorithms for Personalized Federated Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 2304–2315. URL: <https://proceedings.neurips.cc/paper/2020/file/187acf7982f3c169b3075132380986e4-Paper.pdf>.
- [HR21] Filip Hanzely and Peter Richtárik. *Federated Learning of a Mixture of Global and Local Models*. 2021. arXiv: 2002.05516 [cs.LG].
- [Har+19] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. *Federated Learning for Mobile Keyboard Prediction*. 2019. arXiv: 1811.03604 [cs.CL].
- [Haz16] Elad Hazan. “Introduction to Online Convex Optimization”. In: *Foundations and Trends® in Optimization* 2.3-4 (2016), pp. 157–325. ISSN: 2167-3888. DOI: 10.1561/2400000013. URL: <http://dx.doi.org/10.1561/2400000013>.
- [Haz19] Elad Hazan. “Introduction to Online Convex Optimization”. In: *arXiv preprint arXiv:1909.05207* (2019).

- [He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.
- [He+21] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. “Stealing links from graph neural networks”. In: *30th USENIX Security Symposium (USENIX Security 21)*. 2021, pp. 2669–2686.
- [Him97] Michael Himsolt. *GML: A portable graph file format*. Tech. rep. Technical report, Universitat Passau, 1997.
- [HS97] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780.
- [Hor+18] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. “The iNaturalist Species Classification and Detection Dataset”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2018, pp. 8769–8778. DOI: 10.1109/CVPR.2018.00914. URL: <https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00914>.
- [Hor+21] Samuel Horváth, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos Venieris, and Nicholas Donald Lane. “FjORD: Fair and Accurate Federated Learning under heterogeneous targets with Ordered Dropout”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan. 2021. URL: [https://openreview.net/forum?id=4fLr7H5D\\_eT](https://openreview.net/forum?id=4fLr7H5D_eT).
- [Hos+20a] Seyyedali Hosseinalipour, Christopher G Brinton, Vaneet Aggarwal, Huaiyu Dai, and Mung Chiang. “From federated to fog learning: Distributed machine learning over heterogeneous wireless networks”. In: *IEEE Communications Magazine* 58.12 (2020), pp. 41–47.
- [Hos+20b] Seyyedali Hosseinalipour, Christopher G. Brinton, Vaneet Aggarwal, Huaiyu Dai, and Mung Chiang. “From Federated to Fog Learning: Distributed Machine Learning over Heterogeneous Wireless Networks”. In: *IEEE Communications Magazine* 58.12 (Dec. 2020). Conference Name: IEEE Communications Magazine, pp. 41–47. ISSN: 1558-1896. DOI: 10.1109/MCOM.001.2000410.
- [How+21] Frederick M Howard, James Dolezal, Sara Kochanny, Jefree Schulte, Heather Chen, Lara Heij, Dezheng Huo, Rita Nanda, Olufunmilayo I Olopade, Jakob N Kather, et al. “The impact of site-specific digital histology signatures on deep learning model accuracy and bias”. In: *Nature communications* 12.1 (2021), pp. 1–13.
- [Hsi+17] Kevin Hsieh, Aaron Harlap, Nandita Vijaykumar, Dimitris Konomis, Gregory R. Ganger, Phillip B. Gibbons, and Onur Mutlu. “Gaia: Geo-Distributed Machine Learning Approaching LAN Speeds”. In: *Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation*. NSDI’17. Boston, MA, USA: USENIX Association, 2017, pp. 629–647. ISBN: 9781931971379.
- [HQB20] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. “Federated visual classification with real-world data distribution”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 76–92.

- [HQB19] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. “Measuring the effects of non-identical data distribution for federated visual classification”. In: *arXiv preprint arXiv:1909.06335* (2019).
- [ITW] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. *Attention-based Deep Multiple Instance Learning*. <https://github.com/AMLab-Amsterdam/AttentionDeepMIL>. Accessed: 2022-02-02.
- [Ise+21] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. “nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation”. In: *Nature methods* 18.2 (2021), pp. 203–211.
- [JD02] Manish Jain and Constantinos Dovrolis. “End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput”. In: *SIGCOMM Comput. Commun. Rev.* 32.4 (Aug. 2002), pp. 295–308. ISSN: 0146-4833. DOI: 10.1145/964725.633054. URL: <https://doi.org/10.1145/964725.633054>.
- [Jan+19] Andrew Janowczyk, Ren Zuo, Hannah Gilmore, Michael Feldman, and Anant Madabhushi. “HistoQC: an open-source quality control tool for digital pathology slides”. In: *JCO clinical cancer informatics* 3 (2019), pp. 1–7.
- [JWJ22] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. “Towards Understanding Biased Client Selection in Federated Learning”. In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 10351–10375.
- [Jia+19] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. “Memguard: Defending against black-box membership inference attacks via adversarial examples”. In: *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*. 2019, pp. 259–274.
- [Jia+23] Yihan Jiang, Jakub Konečný, Keith Rush, and Sreeram Kannan. *Improving Federated Learning Personalization via Model Agnostic Meta Learning*. 2023. arXiv: 1909.12488 [cs.LG].
- [Jia+17] Zhanhong Jiang, Aditya Balu, Chinmay Hegde, and Soumik Sarkar. “Collaborative Deep Learning in Fixed Topology Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/a74c3bae3e13616104c1b25f9da1f11f-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/a74c3bae3e13616104c1b25f9da1f11f-Paper.pdf).
- [Jin+20] Yibo Jin, Lei Jiao, Zhuzhong Qian, Sheng Zhang, Sanglu Lu, and Xiaoliang Wang. “Resource-Efficient and Convergence-Preserving Online Participant Selection in Federated Learning”. In: *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)* (2020), pp. 606–616.
- [JDJ19] Jeff Johnson, Matthijs Douze, and Herve Jegou. “Billion-scale similarity search with GPUs”. In: *IEEE Transactions on Big Data* (2019), pp. 1–1.

- [Jot+23] Ellango Jothimurugesan, Kevin Hsieh, Jianyu Wang, Gauri Joshi, and Phillip B. Gibbons. “Federated Learning under Distributed Concept Drift”. In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. Ed. by Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent. Vol. 206. Proceedings of Machine Learning Research. PMLR, Apr. 2023, pp. 5834–5853. URL: <https://proceedings.mlr.press/v206/jothimurugesan23a.html>.
- [JM15] Armand Joulin and Tomas Mikolov. “Inferring Algorithmic Patterns with Stack-Augmented Recurrent Nets”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., 2015. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2015/file/26657d5ff9020d2abefe558796b99584-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2015/file/26657d5ff9020d2abefe558796b99584-Paper.pdf).
- [Kai+21] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D’Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrede Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. *Advances and Open Problems in Federated Learning*. 2021. arXiv: 1912.04977 [cs.LG].
- [Kan+19] Jiawen Kang, Zehui Xiong, Dusit Niyato, Han Yu, Ying-Chang Liang, and Dong In Kim. “Incentive Design for Efficient Federated Learning in Mobile Networks: A Contract Theory Approach”. In: *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*. 2019, pp. 1–5. DOI: 10.1109/VTS-APWCS.2019.8851649.
- [KSV05] Ravindran Kannan, Hadi Salmasian, and Santosh Vempala. “The Spectral Method for General Mixture Models”. In: *Learning Theory*. Ed. by Peter Auer and Ron Meir. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 444–457. ISBN: 978-3-540-31892-7.
- [Kap+24b] Caelin Kaplan, Chuan Xu, Othmane Marfoq, Giovanni Neglia, and Anderson Santana de Oliveira. “A Cautionary Tale: On the Role of Reference Data in Empirical Privacy Defenses”. In: *Proceedings on Privacy Enhancing Technologies* (2024).
- [Kar+17] Can Karakus, Yifan Sun, Suhas Diggavi, and Wotao Yin. “Straggler Mitigation in Distributed Optimization Through Data Encoding”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/663772ea088360f95bac3dc7ffb841be-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/663772ea088360f95bac3dc7ffb841be-Paper.pdf).

- [Kar+20a] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. “SCAFFOLD: Stochastic Controlled Averaging for Federated Learning”. In: *International Conference on Machine Learning*. PMLR, 2020, pp. 5132–5143.
- [Kar+20b] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. “SCAFFOLD: Stochastic Controlled Averaging for Federated Learning”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 5132–5143. URL: <https://proceedings.mlr.press/v119/karimireddy20a.html>.
- [Kar78] Richard M. Karp. “A Characterization of the Minimum Cycle Mean in a Digraph”. In: *Discrete Mathematics* 23.3 (1978), pp. 309–311. ISSN: 0012-365X. DOI: [https://doi.org/10.1016/0012-365X\(78\)90011-0](https://doi.org/10.1016/0012-365X(78)90011-0). URL: <https://www.sciencedirect.com/science/article/pii/0012365X78900110>.
- [Kat+18] P. Kathiravelu, M. Chiesa, P. Marcos, M. Canini, and L. Veiga. “Moving Bits with a Fleet of Shared Virtual Routers”. In: *2018 IFIP Networking Conference (IFIP Networking) and Workshops*. 2018, pp. 1–9.
- [Kem+18] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. “Measuring Catastrophic Forgetting in Neural Networks”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018). ISSN: 2374-3468. DOI: 10.1609/aaai.v32i1.11651. (Visited on 05/23/2023).
- [KMR20] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtarik. “Tighter Theory for Local SGD on Identical and Heterogeneous Data”. In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by Silvia Chiappa and Roberto Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, Aug. 2020, pp. 4519–4529. URL: <https://proceedings.mlr.press/v108/bayoumi20a.html>.
- [Kha+21] Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. *Nearest Neighbor Machine Translation*. 2021. arXiv: 2010.00710 [cs.CL].
- [Kha+19] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. “Generalization through Memorization: Nearest Neighbor Language Models”. In: *International Conference on Learning Representations*. 2019.
- [Kha+23] Mehrdad Khani, Ganesh Ananthanarayanan, Kevin Hsieh, Junchen Jiang, Ravi Ne-travali, Yuanchao Shu, Mohammad Alizadeh, and Victor Bahl. “RECL: Responsive Resource-Efficient Continuous Learning for Video Analytics”. In: *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. Boston, MA: USENIX Association, Apr. 2023, pp. 917–932. ISBN: 978-1-939133-33-5. URL: <https://www.usenix.org/conference/nsdi23/presentation/khani>.
- [KBT19] Mikhail Khodak, Maria-Florina F Balcan, and Ameet S Talwalkar. “Adaptive Gradient-Based Meta-Learning Methods”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2019/file/Khodak\\_19.pdf](https://proceedings.neurips.cc/paper_files/paper/2019/file/Khodak_19.pdf).

- //proceedings.neurips.cc/paper\_files/paper/2019/file/f4aa0dd960521e045ae2f20621fb4ee9-Paper.pdf.
- [Kim+23] Taejin Kim, Shubhranshu Singh, Nikhil Madaan, and Carlee Joe-Wong. *Characterizing Internal Evasion Attacks in Federated Learning*. 2023. arXiv: 2209.08412 [cs.LG].
- [KB14] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
- [Kir+17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. “Overcoming Catastrophic Forgetting in Neural Networks”. In: *Proceedings of the national academy of sciences* 114.13 (2017), pp. 3521–3526.
- [Kni+11] S. Knight, H.X. Nguyen, N. Falkner, R. Bowden, and M. Roughan. “The Internet Topology Zoo”. In: *Selected Areas in Communications, IEEE Journal on* 29.9 (Oct. 2011), pp. 1765–1775. ISSN: 0733-8716. DOI: 10.1109/JSAC.2011.111002.
- [Koh+21] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanus Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. “WILDS: A Benchmark of in-the-Wild Distribution Shifts”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 5637–5664. URL: <https://proceedings.mlr.press/v139/koh21a.html>.
- [Kol+20] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. “A Unified Theory of Decentralized SGD with Changing Topology and Local Updates”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 5381–5393. URL: <https://proceedings.mlr.press/v119/koloskova20a.html>.
- [KSJ19] Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. “Decentralized Stochastic Optimization and Gossip Algorithms with Compressed Communication”. In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, June 2019, pp. 3478–3487. URL: <http://proceedings.mlr.press/v97/koloskova19a.html>.
- [Kon+17a] Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. *Federated Learning: Strategies for Improving Communication Efficiency*. 2017. arXiv: 1610.05492 [cs.LG].
- [Kon+17b] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. *Federated Learning: Strategies for Improving Communication Efficiency*. 2017. arXiv: 1610.05492 [cs.LG].



- [Kri+19] Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. “Thieves on sesame street! model extraction of bert-based apis”. In: *arXiv preprint arXiv:1910.12366* (2019).
- [Kri09] Alex Krizhevsky. “Learning multiple layers of features from tiny images”. MSc thesis. 2009.
- [Kul+19] Bogdan Kulynych, Mohammad Yaghini, Giovanni Cherubin, Michael Veale, and Carmela Troncoso. “Disparate vulnerability to membership inference attacks”. In: *arXiv preprint arXiv:1906.00389* (2019).
- [KD12] Abhishek Kumar and Hal Daumé III. “Learning Task Grouping and Overlap in Multi-Task Learning”. In: *Proceedings of the 29th International Conference on International Conference on Machine Learning*. 2012, pp. 1723–1730.
- [KC20] Jeongyeol Kwon and Constantine Caramanis. “The EM Algorithm gives Sample-Optimality for Learning Mixtures of Well-Separated Gaussians”. In: *Proceedings of Thirty Third Conference on Learning Theory*. Ed. by Jacob Abernethy and Shivani Agarwal. Vol. 125. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 2425–2487. URL: <https://proceedings.mlr.press/v125/kwon20a.html>.
- [Lah+20] Amal Lahiani, Irina Klamann, Nassir Navab, Shadi Albarqouni, and Eldad Klaiman. “Seamless virtual whole slide image synthesis and validation using perceptual embedding consistency”. In: *IEEE Journal of Biomedical and Health Informatics* 25.2 (2020), pp. 403–411.
- [Lai+22] Fan Lai, Yinwei Dai, Sanjay Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha Madhyastha, and Mosharaf Chowdhury. “FedScale: Benchmarking model and system performance of federated learning at scale”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 11814–11827.
- [LHY00] Kenneth Lange, David R. Hunter, and Ilsoon Yang. “Optimization Transfer Using Surrogate Objective Functions”. In: *Journal of Computational and Graphical Statistics* 9.1 (2000), pp. 1–20. ISSN: 10618600. URL: <http://www.jstor.org/stable/1390605>.
- [Lap+16] Andrei Lapets, Nikolaj Volgushev, Azer Bestavros, Frederick Jansen, and Mayank Varia. “Secure MPC for Analytics as a Web Application”. In: *2016 IEEE Cybersecurity Development (SecDev)*. 2016, pp. 73–74. DOI: 10.1109/SecDev.2016.027.
- [Lau96] Steffen L. Lauritzen. *Graphical models*. English. Oxford Statistical Science Series 17. Clarendon Press, 1996. ISBN: 0198522193.
- [Le +23] Batiste Le Bars, Aurélien Bellet, Marc Tommasi, Erick Lavoie, and Anne-Marie Kermarrec. “Refined Convergence and Topology Learning for Decentralized SGD with Heterogeneous Data”. In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. Ed. by Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent. Vol. 206. Proceedings of Machine Learning Research. PMLR, 25–27 Apr 2023, pp. 1672–1702. URL: <https://proceedings.mlr.press/v206/le-bars23a.html>.

- [LT01] Jean-Yves Le Boudec and Patrick Thiran. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Berlin, Heidelberg: Springer-Verlag, 2001. ISBN: 354042184X.
- [LC10] Yann LeCun and Corinna Cortes. “MNIST Handwritten Digit Database”. In: (2010). URL: <http://yann.lecun.com/exdb/mnist/>.
- [LP17] David A Levin and Yuval Peres. *Markov Chains and Mixing Times: Second Edition*. Vol. 107. American Mathematical Soc., 2017.
- [LW19] Daliang Li and Junpu Wang. *FedMD: Heterogenous Federated Learning via Model Distillation*. 2019. arXiv: 1910.03581 [cs.LG].
- [LLR21] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. “Membership inference attacks and defenses in classification models”. In: *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*. 2021, pp. 5–16.
- [Li+18] Songze Li, Seyed Mohammadreza Mousavi Kalan, A. Salman Avestimehr, and Mahdi Soltanolkotabi. “Near-Optimal Straggler Mitigation for Distributed Gradient Methods”. In: *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. 2018, pp. 857–866. DOI: 10.1109/IPDPSW.2018.00137.
- [Li+21] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. “Ditto: Fair and Robust Federated Learning Through Personalization”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 6357–6368. URL: <https://proceedings.mlr.press/v139/li21h.html>.
- [Li+20a] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. “Federated Learning: Challenges, Methods, and Future Directions”. In: *IEEE Signal Processing Magazine* 37.3 (2020), pp. 50–60. DOI: 10.1109/MSP.2020.2975749.
- [Li+20b] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. “Federated Optimization in Heterogeneous Networks”. In: *Proceedings of Machine Learning and Systems*. Ed. by I. Dhillon, D. Papailiopoulos, and V. Sze. Vol. 2. 2020, pp. 429–450. URL: [https://proceedings.mlsys.org/paper\\_files/paper/2020/file/1f5fe83998a09396ebe6477d9475ba0c-Paper.pdf](https://proceedings.mlsys.org/paper_files/paper/2020/file/1f5fe83998a09396ebe6477d9475ba0c-Paper.pdf).
- [LM06] Wei Li and Andrew McCallum. “Pachinko Allocation: DAG-Structured Mixture Models of Topic Correlations”. In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML ’06. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 577–584. ISBN: 1595933832. DOI: 10.1145/1143844.1143917. URL: <https://doi.org/10.1145/1143844.1143917>.
- [Li+19] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. “On the Convergence of FedAvg on Non-IID Data”. In: *International Conference on Learning Representations*. 2019.

- [Li+20c] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. “On the Convergence of FedAvg on Non-IID Data”. In: *International Conference on Learning Representations*. 2020. URL: <https://openreview.net/forum?id=HJxNAnVtDS>.
- [Lia+04] Athanassios Liakopoulos, Basil Maglaris, Christos Bouras, and Afrodite Sevasti. “Providing and verifying advanced IP services in hierarchical DiffServ networks-the case of GEANT”. In: *International Journal of Communication Systems* 17.4 (2004), pp. 321–336. DOI: 10.1002/dac.645. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/dac.645>.
- [Lia+17] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. “Can Decentralized Algorithms Outperform Centralized Algorithms? A Case Study for Decentralized Parallel Stochastic Gradient Descent”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/f75526659f31040afeb61cb7133e4e6d-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/f75526659f31040afeb61cb7133e4e6d-Paper.pdf).
- [Lia+18] Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. “Asynchronous Decentralized Parallel Stochastic Gradient Descent”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 3043–3052. URL: <https://proceedings.mlr.press/v80/lian18a.html>.
- [Lin91] J. Lin. “Divergence Measures Based on the Shannon Entropy”. In: *IEEE Transactions on Information Theory* 37.1 (1991), pp. 145–151.
- [Lin+20a] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. “Ensemble Distillation for Robust Model Fusion in Federated Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 2351–2363. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/18df51b97ccd68128e994804f3eccc87-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/18df51b97ccd68128e994804f3eccc87-Paper.pdf).
- [Lin+20b] Tao Lin, Sebastian U. Stich, Kumar Kshitij Patel, and Martin Jaggi. “Don’t Use Large Mini-batches, Use Local SGD”. In: *International Conference on Learning Representations*. 2020. URL: <https://openreview.net/forum?id=BleyO1BFPr>.
- [LB03] Lin Xiao and S. Boyd. “Fast linear iterations for distributed averaging”. In: *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*. Vol. 5. Dec. 2003, 4997–5002 Vol.5. DOI: 10.1109/CDC.2003.1272421.
- [LL17] S. Liu and B. Li. “Stemflow: Software-Defined Inter-Datacenter Overlay as a Service”. In: *IEEE Journal on Selected Areas in Communications* 35.11 (2017), pp. 2563–2573.
- [Liu+15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. “Deep learning face attributes in the wild”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 3730–3738.
- [LB22] Heiko Ludwig and Nathalie Baracaldo. *Federated Learning: A Comprehensive Overview of Methods and Applications*. Springer Cham, 2022, pp. VI, 534. DOI: <https://doi.org/10.1007/978-3-030-96896-0>.

- [Luo+19] Qinyi Luo, Jinkun Lin, Youwei Zhuo, and Xuehai Qian. “Hop: Heterogeneity-Aware Decentralized Training”. In: *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. ASPLOS ’19. Providence, RI, USA: Association for Computing Machinery, 2019, pp. 893–907. ISBN: 9781450362405. DOI: 10.1145/3297858.3304009. URL: <https://doi.org/10.1145/3297858.3304009>.
- [Mah+02] Ratul Mahajan, Neil Spring, David Wetherall, and Tom Anderson. “Inferring Link Weights using End-to-End Measurements”. In: *Workshop on Internet measurement (IMW)*. Aug. 2002.
- [Mai13] Julien Mairal. “Optimization with First-Order Surrogate Functions”. In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, June 2013, pp. 783–791. URL: <https://proceedings.mlr.press/v28/mairal13.html>.
- [20b] *Mammogram Assessment with NVIDIA Clara Federated Learning*. EU research project. 2020. URL: <https://blogs.nvidia.com/blog/2020/04/15/federated-learning-mammogram-assessment/>.
- [Man+20] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. *Three Approaches for Personalization with Applications to Federated Learning*. 2020. arXiv: 2002.10619 [cs.LG].
- [MR10] Sébastien Marcel and Yann Rodriguez. “Torchvision the Machine-Vision Package of Torch”. In: *Proceedings of the 18th ACM International Conference on Multimedia*. MM ’10. Firenze, Italy: Association for Computing Machinery, 2010, pp. 1485–1488. ISBN: 9781605589336. DOI: 10.1145/1873951.1874254. URL: <https://doi.org/10.1145/1873951.1874254>.
- [MMb] Othmane Marfoq and Aryan Mokhtari. *Online Federated Learning with Mixture Models*.
- [Mar+21b] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. “Federated Multi-Task Learning under a Mixture of Distributions”. In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021.
- [Mar+23b] Othmane Marfoq, Giovanni Neglia, Laetitia Kameni, and Richard Vidal. “Federated Learning for Data Streams”. In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. Ed. by Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent. Vol. 206. Proceedings of Machine Learning Research. PMLR, Apr. 2023, pp. 8889–8924. URL: <https://proceedings.mlr.press/v206/marfoq23a.html>.
- [Mar+22b] Othmane Marfoq, Giovanni Neglia, Laetitia Kameni, and Richard Vidal. “Personalized Federated Learning through Local Memorization”. In: *Proceedings of the 39th International Conference on Machine Learning*. Proceedings of Machine Learning Research. PMLR, 2022.
- [Mar+20b] Othmane Marfoq, Chuan Xu, Giovanni Neglia, and Richard Vidal. “Throughput-Optimal Topology Design for Cross-Silo Federated Learning”. In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020.

- [Mar+15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: <https://www.tensorflow.org/>.
- [Mas+07] Laurent Massoulié, Andy Twigg, Christos Gkantsidis, and Pablo Rodriguez. “Randomized Decentralized Broadcasting Algorithms”. In: *Proceedings of the IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*. USA: IEEE Computer Society, 2007, pp. 1073–1081. ISBN: 1424410479. DOI: 10.1109/INFCOM.2007.129. URL: <https://doi.org/10.1109/INFCOM.2007.129>.
- [MC89] Michael McCloskey and Neal J. Cohen. “Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem”. In: ed. by Gordon H. Bower. Vol. 24. *Psychology of Learning and Motivation*. Academic Press, 1989, pp. 109–165. DOI: [https://doi.org/10.1016/S0079-7421\(08\)60536-8](https://doi.org/10.1016/S0079-7421(08)60536-8). URL: <https://www.sciencedirect.com/science/article/pii/S0079742108605368>.
- [McM+17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agueray Arcas. “Communication-efficient learning of deep networks from decentralized data”. In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.
- [McM+18] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. “Learning Differentially Private Recurrent Language Models”. In: *International Conference on Learning Representations*. 2018. URL: <https://openreview.net/forum?id=BJ0hF1Z0b>.
- [Mey01] Carl D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2001.
- [MY21] Adam Meyers and Hui Yang. “Markov Chains for Fault-Tolerance Modeling of Stochastic Networks”. In: *IEEE Transactions on Automation Science and Engineering* (2021).
- [MNA16] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. “V-net: Fully convolutional neural networks for volumetric medical image segmentation”. In: *2016 fourth international conference on 3D vision (3DV)*. IEEE. 2016, pp. 565–571.
- [MHP21] Aritra Mitra, Hamed Hassani, and George J. Pappas. “Online Federated Learning”. In: *2021 60th IEEE Conference on Decision and Control (CDC)*. 2021, pp. 4083–4090. DOI: 10.1109/CDC45484.2021.9683589.
- [MRT18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. 2nd ed. Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press, 2018. 504 pp. ISBN: 978-0-262-03940-6.

- [MSS19] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. “Agnostic Federated Learning”. In: *International Conference on Machine Learning*. 2019, pp. 4615–4625.
- [MPT02] Jérôme Monnot, Vangelis Th. Paschos, and Sophie Toulouse. “Approximation algorithms for the traveling salesman problem”. In: *Mathematical Models of Operations Research* 56 (2002), pp. 387–405. URL: <https://hal.archives-ouvertes.fr/hal-00003997>.
- [MB11] Eric Moulines and Francis Bach. “Non-Asymptotic Analysis of Stochastic Approximation Algorithms for Machine Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger. Vol. 24. Curran Associates, Inc., 2011. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2011/file/40008b9a5380fcacce3976bf7c08af5b-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2011/file/40008b9a5380fcacce3976bf7c08af5b-Paper.pdf).
- [Mus19] Musketeer. Musketeer: About, 2019. <http://musketeer.eu/project/> [Retrieved: Aug 2019]. 2019.
- [NSH19a] Milad Nasr, Reza Shokri, and Amir Houmansadr. “Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning”. In: *2019 IEEE Symposium on Security and Privacy (SP)*. 2019, pp. 739–753. DOI: 10.1109/SP.2019.00065.
- [NSH19b] Milad Nasr, Reza Shokri, and Amir Houmansadr. “Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning”. In: *2019 IEEE symposium on security and privacy (SP)*. IEEE. 2019, pp. 739–753.
- [NSH18] Milad Nasr, Reza Shokri, and Amir Houmansadr. “Machine learning with membership privacy using adversarial regularization”. In: *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. 2018, pp. 634–646.
- [Nas+21] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papemoti, and Nicholas Carlin. “Adversary instantiation: Lower bounds for differentially private machine learning”. In: *2021 IEEE Symposium on Security and Privacy (SP)*. 2021, pp. 866–882.
- [NOS17] Angelia Nedic, Alex Olshevsky, and Wei Shi. “Achieving Geometric Convergence for Distributed Optimization Over Time-Varying Graphs”. In: *SIAM J. Optimization* 27.4 (2017), pp. 2597–2633.
- [NOR18] Angelia Nedić, Alex Olshevsky, and Michael G. Rabbat. “Network Topology and Communication-Computation Tradeoffs in Decentralized Optimization”. In: *Proceedings of the IEEE* 106.5 (2018), pp. 953–976. DOI: 10.1109/JPROC.2018.2817461.
- [NO09] Angelia Nedić and Asuman E. Ozdaglar. “Distributed Subgradient Methods for Multi-Agent Optimization”. In: *IEEE Trans. Automat. Contr.* 54.1 (2009), pp. 48–61.
- [Nee10] Michael J Neely. “Stochastic Network Optimization with Application to Communication and Queueing Systems”. In: *Synthesis Lectures on Communication Networks* 3.1 (2010), pp. 1–211.

- [Neg+19] Giovanni Neglia, Gianmarco Calbi, Don Towsley, and Gayane Vardoyan. “The Role of Network Topology for Distributed Machine Learning”. In: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. 2019, pp. 2350–2358. DOI: 10.1109/INFOCOM.2019.8737602.
- [Neg+20] Giovanni Neglia, Chuan Xu, Don Towsley, and Gianmarco Calbi. “Decentralized gradient methods: does topology matter?” In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by Silvia Chiappa and Roberto Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, Aug. 2020, pp. 2348–2358. URL: <https://proceedings.mlr.press/v108/neglia20a.html>.
- [Nes03] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. 1st ed. Applied Optimization. Springer, 2003. URL: <http://gen.lib.rus.ec/book/index.php?md5=488d3c36f629a6e021fc011675df02ef>.
- [NAS18] Alex Nichol, Joshua Achiam, and John Schulman. *On First-Order Meta-Learning Algorithms*. 2018. arXiv: 1803.02999 [cs.LG].
- [Nik19] Adaloglou Nikolaos. “Deep learning in medical image analysis: a comparative analysis of multi-modal brain-MRI segmentation with 3D deep neural networks”. <https://github.com/black0017/MedicalZooPytorch>. MA thesis. University of Patras, 2019.
- [NY19] Takayuki Nishio and Ryo Yonetani. “Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge”. In: *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. 2019, pp. 1–7. DOI: 10.1109/ICC.2019.8761315.
- [OZ21] Olusola Odeyomi and Gergely Zaruba. “Differentially-Private Federated Learning with Long-Term Constraints Using Online Mirror Descent”. In: *2021 IEEE International Symposium on Information Theory (ISIT)*. 2021, pp. 1308–1313. DOI: 10.1109/ISIT45174.2021.9518177.
- [Ogi+22b] Jean Ogier du Terrail, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold, Tanguy Marchand, Othmane Marfoq, Erum Mushtaq, Boris Muzellec, Constantin Philippenko, Santiago Silva, Maria Teleńczuk, Shadi Albarqouni, Salman Avestimehr, Aurélien Bellet, Aymeric Dieuleveut, Martin Jaggi, Sai Praneeth Karimireddy, Marco Lorenzi, Giovanni Neglia, Marc Tommasi, and Mathieu Andreux. “FLamby: Datasets and Benchmarks for Cross-Silo Federated Learning in Realistic Settings”. Proceedings of The 36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks. <https://openreview.net/forum?id=GgM5DiAb6A2>. 2022.
- [Oli+23] Anderson Santana de Oliveira, Caelin Kaplan, Khawla Mallat, and Tanmay Chakraborty. “An Empirical Analysis of Fairness Notions under Differential Privacy”. In: *arXiv preprint arXiv:2302.02910* (2023).
- [OYJ97] Häggström Olle, Peres Yuval, and E Steif Jeffrey. “Dynamical Percolation”. In: *Annales de l’Institut Henri Poincaré (B) Probability and Statistics*. Vol. 33. 4. Elsevier, 1997, pp. 497–528.

- [Orh18] Emin Orhan. “A Simple Cache Model for Image Recognition”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2018/file/6e0917469214d8fbd8c517dcdc6b8dcf-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2018/file/6e0917469214d8fbd8c517dcdc6b8dcf-Paper.pdf).
- [Pap+16] Nicolas Papernot, Martín Abadi, Ulkar Erlingsson, Ian Goodfellow, and Kunal Talwar. “Semi-supervised knowledge transfer for deep learning from private training data”. In: *arXiv preprint arXiv:1610.05755* (2016).
- [PM18] Nicolas Papernot and Patrick McDaniel. *Deep k-Nearest Neighbors: Towards Confidential, Interpretable and Robust Deep Learning*. 2018. arXiv: 1803.04765 [cs.LG].
- [Pap+18] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. “Scalable private learning with pate”. In: *arXiv preprint arXiv:1802.08908* (2018).
- [PS21] Nicolas Papernot and Thomas Steinke. “Hyperparameter tuning with renyi differential privacy”. In: *arXiv preprint arXiv:2110.03620* (2021).
- [Par16] European Parliament. *General Data Protection Regulation (GDPR)*. European Parliament. Apr. 14, 2016. URL: <https://gdpr-info.eu/> (visited on 11/30/2022).
- [Par20] European Parliament. *The impact of the General Data Protection Regulation (GDPR) on artificial intelligence*. European Parliamentary Research Service. June 1, 2020. URL: [https://www.europarl.europa.eu/RegData/etudes/STUD/2020/641530/EPRS\\_STU\(2020\)641530\\_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/STUD/2020/641530/EPRS_STU(2020)641530_EN.pdf) (visited on 11/30/2022).
- [Pas+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 8024–8035. URL: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [PSM14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “Glove: Global Vectors for Word Representation.” In: *EMNLP*. Vol. 14. 2014, pp. 1532–1543.
- [Per+17] Valerio Persico, Alessio Botta, Pietro Marchetta, Antonio Montieri, and Antonio Pescapé. “On the performance of the wide-area networks interconnecting public-cloud datacenters around the globe”. In: *Computer Networks* 112 (2017), pp. 67–83. ISSN: 1389-1286. DOI: <https://doi.org/10.1016/j.comnet.2016.10.013>. URL: <http://www.sciencedirect.com/science/article/pii/S138912861630353X>.
- [PD22] Constantin Philippenko and Aymeric Dieuleveut. *Bidirectional compression in heterogeneous settings for distributed or federated learning with partial participation: tight convergence guarantees*. 2022. arXiv: 2006.14591 [cs.LG].



- [PD21] Constantin Philippenko and Aymeric Dieuleveut. “Preserved central model for faster bidirectional compression in distributed settings”. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 2387–2399. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2021/file/13d63838ef1fb6f34ca2dc6821c60e49-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2021/file/13d63838ef1fb6f34ca2dc6821c60e49-Paper.pdf).
- [PFT21] Amaury Bouchra Pilet, Davide Frey, and François Taiani. “Simple, Efficient and Convenient Decentralized Multi-task Learning for Neural Networks.” In: *IDA*. 2021, pp. 37–49.
- [Pra+03] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy. “Bandwidth Estimation: Metrics, Measurement Techniques, and Tools”. In: *IEEE Network* 17.6 (2003), pp. 27–35.
- [Pri57] R. C. Prim. “Shortest Connection Networks and Some Generalizations”. In: *The Bell System Technical Journal* 36.6 (1957), pp. 1389–1401. DOI: 10.1002/j.1538-7305.1957.tb01515.x.
- [POP20a] Shi Pu, Alex Olshevsky, and Ioannis Ch. Paschalidis. “Asymptotic Network Independence in Distributed Stochastic Optimization for Machine Learning: Examining Distributed and Centralized Stochastic Gradient Descent”. In: *IEEE Signal Process. Mag.* 37.3 (2020), pp. 114–122.
- [POP20b] Shi Pu, Alex Olshevsky, and Ioannis Ch. Paschalidis. “Asymptotic Network Independence in Distributed Stochastic Optimization for Machine Learning: Examining Distributed and Centralized Stochastic Gradient Descent”. In: *IEEE Signal Processing Magazine* 37.3 (2020), pp. 114–122. DOI: 10.1109/MSP.2020.2975212.
- [Qiu+23] Xinchu Qiu, Titouan Parcollet, Javier Fernandez-Marques, Pedro P. B. Gusmao, Yan Gao, Daniel J. Beutel, Taner Topal, Akhil Mathur, and Nicholas D. Lane. “A First Look into the Carbon Footprint of Federated Learning”. In: *Journal of Machine Learning Research* 24.129 (2023), pp. 1–23. URL: <http://jmlr.org/papers/v24/21-0445.html>.
- [RNV12] S. Sundhar Ram, Angelia Nedic, and Venugopal V. Veeravalli. “A New Class of Distributed Optimization Algorithms: Application to Regression of Distributed Data”. In: *Optimization Methods and Software* 27.1 (2012), pp. 71–88. DOI: 10.1080/10556788.2010.511669. URL: <https://doi.org/10.1080/10556788.2010.511669>.
- [Ram+19] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. *Federated Learning for Emoji Prediction in a Mobile Keyboard*. 2019. arXiv: 1906.04329 [cs.CL].
- [Red+21] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. “Adaptive Federated Optimization”. In: *International Conference on Learning Representations*. 2021. URL: <https://openreview.net/forum?id=LkFG31B13U5>.
- [RV17] Oded Regev and Aravindan Vijayaraghavan. “On Learning Mixtures of Well-Separated Gaussians”. In: *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. 2017, pp. 85–96. DOI: 10.1109/FOCS.2017.17.

- [Res12] Meta AI Research. *Federated Learning Simulator (FLSim)*. <https://github.com/facebookresearch/FLSim/tree/main/examples>. 2012.
- [RVd23] Mónica Ribero, Haris Vikalo, and Gustavo de Veciana. “Federated Learning Under Intermittent Client Availability and Time-Varying Communication Constraints”. In: *IEEE Journal of Selected Topics in Signal Processing* 17.1 (Jan. 2023), pp. 98–111. ISSN: 1941-0484. DOI: 10.1109/JSTSP.2022.3224590.
- [Rod+23b] Angelo Rodio, Francescomaria Faticanti, Othmane Marfoq, Giovanni Neglia, and Emilio Leonardi. “Federated Learning under Heterogeneous and Correlated Client Availability”. In: *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*. 2023, pp. 1–10. DOI: 10.1109/INFOCOM53939.2023.10228876.
- [RJ22] Yichen Ruan and Carlee Joe-Wong. “FedSoft: Soft Clustered Federated Learning with Proximal Local Updating”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.7 (June 2022), pp. 8124–8131. DOI: 10.1609/aaai.v36i7.20785. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/20785>.
- [RE13] Paul Ruvolo and Eric Eaton. “ELLA: An Efficient Lifelong Learning Algorithm”. In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 1. Atlanta, Georgia, USA: PMLR, June 2013, pp. 507–515. URL: <https://proceedings.mlr.press/v28/ruvolo13.html>.
- [Sal+20] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. “{Updates-Leak}: Data Set Inference and Reconstruction Attacks in Online Learning”. In: *29th USENIX security symposium (USENIX Security 20)*. 2020, pp. 1291–1308.
- [San+18] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.
- [SMS20] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. “Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2020).
- [Sat+19] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. “Sparse binary compression: Towards distributed deep learning with minimal communication”. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, pp. 1–8.
- [Sca+18] Kevin Scaman, Francis Bach, Sebastien Bubeck, Laurent Massoulié, and Yin Tat Lee. “Optimal Algorithms for Non-Smooth Distributed Optimization in Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2018/file/8fb21ee7a2207526da55a679f0332de2-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2018/file/8fb21ee7a2207526da55a679f0332de2-Paper.pdf).

- [Sca+17] Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. “Optimal Algorithms for Smooth and Strongly Convex Distributed Optimization in Networks”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, Aug. 2017, pp. 3027–3036. URL: <https://proceedings.mlr.press/v70/scaman17a.html>.
- [Sch+18] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. “Progress & Compress: A Scalable Framework for Continual Learning”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4528–4537.
- [SL21] Suhail Mohmad Shah and Vincent K. N. Lau. “Model Compression for Communication Efficient Federated Learning”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2021), pp. 1–15. DOI: 10.1109/TNNLS.2021.3131614.
- [SB14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- [Sha+21] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. “Personalized Federated Learning using Hypernetworks”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 9489–9502. URL: <https://proceedings.mlr.press/v139/shamsian21a.html>.
- [SH21] Virat Shejwalkar and Amir Houmansadr. “Membership privacy for machine learning models through knowledge transfer”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 11. 2021, pp. 9549–9557.
- [Shi+15] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. “EXTRA: An Exact First-Order Algorithm for Decentralized Consensus Optimization”. In: *SIAM J. Optimization* 25.2 (2015), pp. 944–966.
- [SS15] Reza Shokri and Vitaly Shmatikov. “Privacy-Preserving Deep Learning”. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. CCS ’15. Denver, Colorado, USA: Association for Computing Machinery, 2015, pp. 1310–1321. ISBN: 9781450338325. DOI: 10.1145/2810103.2813687. URL: <https://doi.org/10.1145/2810103.2813687>.
- [Sho+17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. “Membership inference attacks against machine learning models”. In: *2017 IEEE symposium on security and privacy (SP)*. IEEE. 2017, pp. 3–18.
- [Sil+20] Santiago Silva, Andre Altmann, Boris Gutman, and Marco Lorenzi. “Fed-BioMed: A General Open-Source Frontend Framework for Federated Learning in Healthcare”. In: *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*. Ed. by Shadi Albarqouni, Spyridon Bakas, Konstantinos Kamnitsas, M. Jorge Cardoso, Bennett Landman, Wenqi Li, Fausto Milletari, Nicola Rieke, Holger Roth, Daguang Xu, and Ziyue Xu. Cham: Springer International Publishing, 2020, pp. 201–210. ISBN: 978-3-030-60548-3.

- [Sil+19] Santiago Silva, Boris A Gutman, Eduardo Romero, Paul M Thompson, Andre Altmann, and Marco Lorenzi. “Federated Learning in Distributed Medical Databases: Meta-Analysis of Large-Scale subcortical brain data”. In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE. 2019, pp. 270–274.
- [Smi+17] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. “Federated Multi-Task Learning”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 4427–4437. ISBN: 9781510860964.
- [SSZ17] Jake Snell, Kevin Swersky, and Richard Zemel. “Prototypical Networks for Few-shot Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf).
- [SM21] Liwei Song and Prateek Mittal. “Systematic evaluation of privacy risks of machine learning models”. In: *30th USENIX Security Symposium (USENIX Security 21)*. 2021, pp. 2615–2632.
- [Spr+04] Neil Spring, Ratul Mahajan, David Wetherall, and Thomas Anderson. “Measuring ISP Topologies with Rocketfuel”. In: *IEEE/ACM Trans. Netw.* 12.1 (Feb. 2004), pp. 2–16. ISSN: 1063-6692. DOI: 10.1109/TNET.2003.822655. URL: <https://doi.org/10.1109/TNET.2003.822655>.
- [Sti19] Sebastian U. Stich. “Local SGD Converges Fast and Communicates Little”. In: *International Conference on Learning Representations*. 2019. URL: <https://openreview.net/forum?id=Slg2JnRcFX>.
- [Str18] Volker Strobel. *Pold87/academic-keyword-occurrence: First release*. Version v1.0.0. Apr. 2018. DOI: 10.5281/zenodo.1218409. URL: <https://doi.org/10.5281/zenodo.1218409>.
- [Sug+07] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Buenau, and Motoaki Kawanabe. “Direct Importance Estimation with Model Selection and Its Application to Covariate Shift Adaptation”. In: *Advances in Neural Information Processing Systems*. Ed. by J. Platt, D. Koller, Y. Singer, and S. Roweis. Vol. 20. Curran Associates, Inc., 2007. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2007/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2007/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper.pdf).
- [Sui+22] Yi Sui, Junfeng Wen, Yenson Lau, Brendan Leigh Ross, and Jesse C Cresswell. “Find Your Friends: Personalized Federated Learning with the Right Collaborators”. In: *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*. 2022.
- [SSY18] Tao Sun, Yuejiao Sun, and Wotao Yin. “On Markov Chain Gradient Descent”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018. URL: [https://proceedings.neurips.cc/paper\\_files/](https://proceedings.neurips.cc/paper_files/)

- paper/2018/file/1371bccec2447b5aa6d96d2a540fb401-Paper.pdf.
- [Sup+] Abhijit Suprem, Joy Arulraj, Calton Pu, and Joao Ferreira. “ODIN: Automated Drift Detection and Recovery in Video Analytics”. In: *Proceedings of the VLDB Endowment* 13.11 ().
- [TTN20] Canh T. Dinh, Nguyen Tran, and Josh Nguyen. “Personalized Federated Learning with Moreau Envelopes”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 21394–21405. URL: <https://proceedings.neurips.cc/paper/2020/file/f4f1f13c8289ac1b1ee0ff176b56fc60-Paper.pdf>.
- [Tak+23] Yuki Takezawa, Ryoma Sato, Han Bao, Kenta Niwa, and Makoto Yamada. *Beyond Exponential Graph: Communication-Efficient Topologies for Decentralized Learning via Finite-time Convergence*. 2023. arXiv: 2305.11420 [cs.LG].
- [Tan+22a] Lei Tan, Xiaoxi Zhang, Yipeng Zhou, Xinkai Che, Miao Hu, Xu Chen, and Di Wu. “AdaFed: Optimizing Participation-Aware Federated Learning with Adaptive Aggregation Weights”. In: *IEEE Transactions on Network Science and Engineering* (2022).
- [TL19] Mingxing Tan and Quoc Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 6105–6114. URL: <https://proceedings.mlr.press/v97/tan19a.html>.
- [Tan+22b] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. “FedProto: Federated Prototype Learning across Heterogeneous Clients”. In: *AAAI Conference on Artificial Intelligence*. 2022.
- [Tan+18] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. “ $D^2$ : Decentralized Training over Decentralized Data”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 4848–4856. URL: <https://proceedings.mlr.press/v80/tang18a.html>.
- [Tan+22c] Minxue Tang, Xuefei Ning, Yitu Wang, Jingwei Sun, Yu Wang, Hai Li, and Yiran Chen. “FedCor: Correlation-Based Active Client Selection Strategy for Heterogeneous Federated Learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.
- [Tan+21] Xinyu Tang, Saeed Mahloujifar, Liwei Song, Virat Shejwalkar, Milad Nasr, Amir Houmansadr, and Prateek Mittal. “Mitigating membership inference attacks by self-distillation through a novel ensemble architecture”. In: *arXiv preprint arXiv:2110.08324* (2021).
- [Ten19] Tensorflow. *TensorFlow Federated Stack Overflow Dataset*. [https://www.tensorflow.org/federated/api\\_docs/python/tff/simulation/datasets/stackoverflow/load\\_data](https://www.tensorflow.org/federated/api_docs/python/tff/simulation/datasets/stackoverflow/load_data). 2019.

- [TH23] Naoyuki Terashita and Satoshi Hara. *Decentralized Hyper-Gradient Computation over Time-Varying Directed Networks*. 2023. arXiv: 2210.02129 [stat.ML].
- [Ter+21] Jean Ogier du Terrail, Armand Léopold, Clément Joly, Constance Beguier, Mathieu Andreux, Charles Maussion, Benoit Schmauch, Eric W. Tramel, Etienne Bendjebbar, Mikhail Zaslavskiy, Gilles Wainrib, Maud Milder, Julie Gervasoni, Julien Guérin, Thierry Durand, Alain Livartowski, Kelvin Moutet, Clément Gautier, Inal Djafar, Anne-Laure Moisson, Camille Marini, Mathieu Galtier, Guillaume Bataillon, and Pierre-Etienne Heudel. “Collaborative Federated Learning behind Hospitals’ Firewalls for Predicting Histological Response to Neoadjuvant Chemotherapy in Triple-Negative Breast Cancer”. In: *medRxiv* (2021). DOI: 10.1101/2021.10.27.21264834. eprint: <https://www.medrxiv.org/content/early/2021/10/28/2021.10.27.21264834.full.pdf>. URL: <https://www.medrxiv.org/content/early/2021/10/28/2021.10.27.21264834>.
- [Thr94] S. Thrun. “A Lifelong Learning Perspective for Mobile Robot Control”. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’94)*. Vol. 1. 1994, 23–30 vol.1. DOI: 10.1109/IROS.1994.407413.
- [TE11] Antonio Torralba and Alexei A Efros. “Unbiased look at dataset bias”. In: *CVPR 2011*. IEEE. 2011, pp. 1521–1528.
- [TB20] Florian Tramer and Dan Boneh. “Differentially private learning needs better features (or much more data)”. In: *arXiv preprint arXiv:2011.11660* (2020).
- [Tra+16] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. “Stealing machine learning models via prediction {APIs}”. In: *25th USENIX security symposium (USENIX Security 16)*. 2016, pp. 601–618.
- [TLR12] Konstantinos I. Tsianos, Sean Lawlor, and Michael G. Rabbat. “Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning”. In: *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2012, pp. 1543–1550. DOI: 10.1109/Allerton.2012.6483403.
- [Tsy08] Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. 1st. Springer Publishing Company, Incorporated, 2008. ISBN: 0387790519.
- [Tu+22] Xuezheng Tu, Kun Zhu, Nguyen Cong Luong, Dusit Niyato, Yang Zhang, and Juan Li. “Incentive Mechanisms for Federated Learning: From Economic and Game Theoretic Perspective”. In: *IEEE Transactions on Cognitive Communications and Networking* 8.3 (2022), pp. 1566–1593. DOI: 10.1109/TCCN.2022.3177522.
- [Uni+21] Archit Uniyal, Rakshit Naidu, Sasikanth Kotti, Sahib Singh, Patrik Joslin Kenfack, Fatemehsadat Mireshghallah, and Andrew Trask. “DP-SGD vs PATE: Which Has Less Disparate Impact on Model Accuracy?” In: *arXiv preprint arXiv:2106.12576* (2021).

- [Vol+17] Michael Volske, Martin Potthast, Shahbaz Syed, and Benno Stein. “TL;DR: Mining Reddit to Learn Automatic Summarization”. In: *Proceedings of the Workshop on New Frontiers in Summarization*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 59–63. DOI: 10.18653/v1/W17-4508. URL: <https://www.aclweb.org/anthology/W17-4508>.
- [VH08] Laurens Van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.11 (2008).
- [Van+18] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. “The iNaturalist Species Classification and Detection Dataset”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8769–8778.
- [VBT17] Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi. “Decentralized Collaborative Learning of Personalized Models over Networks”. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by Aarti Singh and Jerry Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, Apr. 2017, pp. 509–517. URL: <https://proceedings.mlr.press/v54/vanhaesebrouck17a.html>.
- [VC15] V. N. Vapnik and A. Ya. Chervonenkis. “On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities”. In: *Measures of Complexity: Festschrift for Alexey Chervonenkis*. Cham: Springer International Publishing, 2015, pp. 11–30. ISBN: 978-3-319-21852-6. DOI: 10.1007/978-3-319-21852-6\_3. URL: [https://doi.org/10.1007/978-3-319-21852-6\\_3](https://doi.org/10.1007/978-3-319-21852-6_3).
- [Vet+14] Mitko Veta, Josien PW Pluim, Paul J Van Diest, and Max A Viergever. “Breast cancer histopathology image analysis: A review”. In: *IEEE transactions on biomedical engineering* 61.5 (2014), pp. 1400–1411.
- [Vog+20] Robin Vogel, Mastane Achab, Stéphan Cléménçon, and Charles Tillier. “Weighted Empirical Risk Minimization: Transfer Learning based on Importance Sampling”. In: *ESANN*. 2020.
- [WBX23] Heqiang Wang, Jieming Bian, and Jie Xu. *On the Local Cache Update Rules in Streaming Federated Learning*. 2023. arXiv: 2303.16340 [cs.LG].
- [Wan+20a] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. “Federated Learning with Matched Averaging”. In: *International Conference on Learning Representations*. 2020. URL: <https://openreview.net/forum?id=BkluqlSFDS>.
- [Wan+21a] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. “A Field Guide to Federated Optimization”. In: *arXiv preprint arXiv:2107.06917* (2021).
- [WJ19] Jianyu Wang and Gauri Joshi. “Adaptive Communication Strategies to Achieve the Best Error-Runtime Trade-off in Local-Update SGD”. In: *Proceedings of Machine Learning and Systems*. Ed. by A. Talwalkar, V. Smith, and M. Zaharia. Vol. 1. 2019, pp. 212–229. URL: [https://proceedings.mlsys.org/paper\\_files/](https://proceedings.mlsys.org/paper_files/)

- paper/2019/file/4a0151b47bd93c5de2a0b57831981a0d-Paper.pdf.
- [WJ21] Jianyu Wang and Gauri Joshi. “Cooperative SGD: A Unified Framework for the Design and Analysis of Local-Update SGD Algorithms”. In: *Journal of Machine Learning Research* 22.213 (2021), pp. 1–50. URL: <http://jmlr.org/papers/v22/20-147.html>.
- [Wan+20b] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. “Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 7611–7623. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/564127c03caab942e503ee6f810f54fd-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/564127c03caab942e503ee6f810f54fd-Paper.pdf).
- [Wan+19a] Jianyu Wang, Anit Kumar Sahu, Zhouyi Yang, Gauri Joshi, and Soumya Kar. “MATCHA: Speeding Up Decentralized SGD via Matching Decomposition Sampling”. In: *2019 Sixth Indian Control Conference (ICC)*. 2019, pp. 299–300. DOI: 10.1109/ICC47138.2019.9123209.
- [Wan+20c] Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. *SlowMo: Improving Communication-Efficient Distributed SGD with Slow Momentum*. 2020. arXiv: 1910.00643 [cs.LG].
- [Wan+21b] Su Wang, Yichen Ruan, Yuwei Tu, Satyavrat Wagle, Christopher G. Brinton, and Carlee Joe-Wong. “Network-Aware Optimization of Distributed Learning for Fog Computing”. In: *IEEE/ACM Transactions on Networking* 29.5 (Oct. 2021). Conference Name: IEEE/ACM Transactions on Networking, pp. 2019–2032. ISSN: 1558-2566. DOI: 10.1109/TNET.2021.3075432.
- [Wan+22a] Tianchun Wang, Wei Cheng, Dongsheng Luo, Wenchao Yu, Jingchao Ni, Liang Tong, Haifeng Chen, and Xiang Zhang. “Personalized Federated Learning via Heterogeneous Modular Networks”. In: *2022 IEEE International Conference on Data Mining (ICDM)*. 2022, pp. 1197–1202. DOI: 10.1109/ICDM54844.2022.00154.
- [Wan+22b] Tianchun Wang, Wei Cheng, Dongsheng Luo, Wenchao Yu, Jingchao Ni, Liang Tong, Haifeng Chen, and Xiang Zhang. “Personalized Federated Learning via Heterogeneous Modular Networks”. In: *2022 IEEE International Conference on Data Mining (ICDM)*. 2022, pp. 1197–1202. DOI: 10.1109/ICDM54844.2022.00154.
- [Wan+19b] Yan Wang, Wei-Lun Chao, Kilian Q. Weinberger, and Laurens van der Maaten. *SimpleShot: Revisiting Nearest-Neighbor Classification for Few-Shot Learning*. 2019. arXiv: 1911.04623 [cs.CV].
- [Wan+20d] Yijue Wang, Chenghong Wang, Zigeng Wang, Shanglin Zhou, Hang Liu, Jinbo Bi, Caiwen Ding, and Sanguthevar Rajasekaran. “Against membership inference attack: Pruning is all you need”. In: *arXiv preprint arXiv:2008.13578* (2020).
- [WeB19] WeBank. WeBank and Swiss Resigned Cooperation MOU, 2019. <https://finance.yahoo.com/news/webank-swiss-signed-cooperation-mou-112300218.html> [Retrieved: Aug 2019]. 2019.



- [Wei+20] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. “Federated Learning with Differential Privacy: Algorithms and Performance Analysis”. In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 3454–3469.
- [Woo+20] Blake Woodworth, Kumar Kshitij Patel, Sebastian Stich, Zhen Dai, Brian Bullins, Brendan McMahan, Ohad Shamir, and Nathan Srebro. “Is Local SGD Better than Minibatch SGD?” In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 10334–10343. URL: <https://proceedings.mlr.press/v119/woodworth20a.html>.
- [Wu+23] Yue Wu, Shuaicheng Zhang, Wenchao Yu, Yanchi Liu, Quanquan Gu, Dawei Zhou, Haifeng Chen, and Wei Cheng. *Personalized Federated Learning under Mixture of Distributions*. 2023. arXiv: 2305.01068 [cs.LG].
- [XNS21] Chuan Xu, Giovanni Neglia, and Nicola Sebastianelli. “Dynamic backup workers for parallel machine learning”. In: *Computer Networks* 188 (2021), p. 107846. ISSN: 1389-1286. DOI: <https://doi.org/10.1016/j.comnet.2021.107846>. URL: <https://www.sciencedirect.com/science/article/pii/S1389128621000256>.
- [Yan+20a] Liu Yang, Ben Tan, Vincent W. Zheng, Kai Chen, and Qiang Yang. “Federated Recommendation Systems”. In: *Federated Learning: Privacy and Incentive*. Ed. by Qiang Yang, Lixin Fan, and Han Yu. Cham: Springer International Publishing, 2020, pp. 225–239. ISBN: 978-3-030-63076-8. DOI: 10.1007/978-3-030-63076-8\_16. URL: [https://doi.org/10.1007/978-3-030-63076-8\\_16](https://doi.org/10.1007/978-3-030-63076-8_16).
- [Yan+18] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. *Applied Federated Learning: Improving Google Keyboard Query Suggestions*. 2018. arXiv: 1812.02903 [cs.LG].
- [Yan+20b] Ziqi Yang, Bin Shao, Bohan Xuan, Ee-Chien Chang, and Fan Zhang. “Defending model inversion and membership inference attacks via prediction purification”. In: *arXiv preprint arXiv:2005.03915* (2020).
- [Yao86] Andrew Chi-Chih Yao. “How to generate and exchange secrets”. In: *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*. 1986, pp. 162–167. DOI: 10.1109/SFCS.1986.25.
- [Ye+22] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. “Enhanced membership inference attacks against machine learning models”. In: *CCS '22* (2022).
- [Yeo+20] Samuel Yeom, Irene Giacomelli, Alan Menaged, Matt Fredrikson, and Somesh Jha. “Overfitting, robustness, and malicious algorithms: A study of potential causes of privacy risk in machine learning”. In: *Journal of Computer Security* 28.1 (2020), pp. 35–70.

- [Yoo+21] Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. “Federated Continual Learning with Weighted Inter-client Transfer”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 12073–12086. URL: <https://proceedings.mlr.press/v139/yoon21b.html>.
- [YBS22] Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. *Salvaging Federated Learning by Local Adaptation*. 2022. arXiv: 2002.04758 [cs.LG].
- [YLY16] Kun Yuan, Qing Ling, and Wotao Yin. “On the Convergence of Decentralized Gradient Descent”. In: *SIAM Journal on Optimization* 26.3 (2016), pp. 1835–1854. DOI: 10.1137/130943170. eprint: <https://doi.org/10.1137/130943170>. URL: <https://doi.org/10.1137/130943170>.
- [Yua+19] Kun Yuan, Bicheng Ying, Xiaochuan Zhao, and Ali H. Sayed. “Exact Diffusion for Distributed Optimization and Learning—Part I: Algorithm Development”. In: *IEEE Transactions on Signal Processing* 67.3 (2019), pp. 708–723. DOI: 10.1109/TSP.2018.2875898.
- [Yua+23] Liangqi Yuan, Lichao Sun, Philip S. Yu, and Ziran Wang. *Decentralized Federated Learning: A Survey and Perspective*. 2023. arXiv: 2306.01603 [cs.LG].
- [Yua+21] Ye Yuan, Jun Liu, Dou Jin, Zuogong Yue, Ruijuan Chen, Maolin Wang, Chuan Sun, Lei Xu, Feng Hua, Xin He, Xinlei Yi, Tao Yang, Hai-Tao Zhang, Shaochun Sui, and Han Ding. *DeceFL: A Principled Decentralized Federated Learning Framework*. 2021. arXiv: 2107.07171 [cs.LG].
- [Yur+19] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. “Bayesian Nonparametric Federated Learning of Neural Networks”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 7252–7261. URL: <https://proceedings.mlr.press/v97/yurochkin19a.html>.
- [ZBT20] Valentina Zantedeschi, Aurélien Bellet, and Marc Tommasi. “Fully Decentralized Joint Learning of Personalized Models and Collaboration Graphs”. In: ed. by Silvia Chiappa and Roberto Calandra. Vol. 108. Proceedings of Machine Learning Research. Online: PMLR, Aug. 2020, pp. 864–874. URL: <http://proceedings.mlr.press/v108/zantedeschi20a.html>.
- [ZY17] Jinshan Zeng and Wotao Yin. “Extrapush for Convex Smooth Decentralized Optimization Over Directed Networks”. In: *Journal of Computational Mathematics* 35.4 (2017), pp. 383–396. ISSN: 02549409, 19917139. URL: <http://www.jstor.org/stable/45151444> (visited on 07/30/2023).
- [ZWY22] L. Zhang, D. Wu, and X. Yuan. “FedZKT: Zero-Shot Knowledge Transfer towards Resource-Constrained Federated Learning with Heterogeneous On-Device Models”. In: *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. Los Alamitos, CA, USA: IEEE Computer Society, July 2022, pp. 928–938. DOI: 10.1109/ICDCS54860.2022.00094. URL: <https://doi.ieeeecomputersociety.org/10.1109/ICDCS54860.2022.00094>.

- [Zha+21] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M. Alvarez. “Personalized Federated Learning with First Order Model Optimization”. In: *International Conference on Learning Representations*. 2021. URL: <https://openreview.net/forum?id=ehJqJQk9cw>.
- [ZY10] Yu Zhang and Dit Yan Yeung. “A Convex Formulation for Learning Task Relationships in Multi-task Learning”. In: *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, UAI 2010*. 2010, p. 733.
- [ZCY11] Jiayu Zhou, Jianhui Chen, and Jieping Ye. “Clustered Multi-Task Learning Via Alternating Structure Optimization”. In: *Advances in Neural Information Processing Systems*. Ed. by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger. Vol. 24. Curran Associates, Inc., 2011. URL: <https://proceedings.neurips.cc/paper/2011/file/a516a87cfcaef229b342c437fe2b95f7-Paper.pdf>.
- [Zho+20] Zhi Zhou, Song Yang, Lingjun Pu, and Shuai Yu. “CEFL: Online Admission Control, Data Scheduling, and Accuracy Tuning for Cost-Efficient Federated Learning Across Edge Nodes”. In: *IEEE Internet of Things Journal* 7 (2020), pp. 9341–9356.
- [Zhu+22] Chen Zhu, Zheng Xu, Mingqing Chen, Jakub Konečný, Andrew Hard, and Tom Goldstein. “Diurnal or Nocturnal? Federated Learning of Multi-branch Networks from Periodically Shifting Distributions”. In: *International Conference on Learning Representations*. 2022. URL: [https://openreview.net/forum?id=E4EE\\_ohFGz](https://openreview.net/forum?id=E4EE_ohFGz).
- [ZS02] Yunyue Zhu and Dennis Shasha. “StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time”. In: *VLDB*. 2002.
- [ZHZ21] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. “Data-Free Knowledge Distillation for Heterogeneous Federated Learning”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 12878–12889. URL: <https://proceedings.mlr.press/v139/zhu21b.html>.
- [Zin03] Martin Zinkevich. “Online Convex Programming and Generalized Infinitesimal Gradient Ascent.” In: *ICML*. Ed. by Tom Fawcett and Nina Mishra. AAAI Press, 2003, pp. 928–936. ISBN: 1-57735-189-4. URL: <http://dblp.uni-trier.de/db/conf/icml/icml2003.html#Zinkevich03>.

# List of Figures

---

1.1	Occurrences of the key word “federated learning” over time in academic papers (from Google Scholar). The results were obtained using the code from <a href="https://github.com/Pold87/academic-keyword-occurrence">https://github.com/Pold87/academic-keyword-occurrence</a> [Str18] . . . . .	2
1.2	Federated learning system. <b>Left:</b> the cross-device scenario includes a large number of unreliable mobile devices with limited computing resources and slow Internet connections; it requires a client server architecture where mobiles communicate only with the server. <b>Right:</b> the cross-silo scenario includes at most a few hundred reliable data silos with powerful computing resources and high-speed access links; it may take advantage of peer-to-peer communications. . . . .	3
1.3	Tradeoff between a defended classifier’s prediction accuracy on test data (i.e., its model utility), membership inference attack accuracy on training data (i.e., training data privacy leakage), membership inference attack accuracy on reference data (i.e., reference data privacy leakage) for Purchase100 dataset. “AdvReg” corresponds to the original formulation of adversarial regularization [NSH18] and “AdvReg-RT” corresponds to a revisited version that we propose. . . . .	16
2.1	Examples for underlay, connectivity graph, and overlay, with routers (blue nodes), silos (red nodes), underlay links (solid black lines), and information exchanges (dashed lines). . . . .	26
2.2	Networks where a directed topology outperforms an undirected one. . . . .	29
2.10	Throughput speedup in comparison to the STAR, when training iNaturalist over Exodus network. All links with 1 Gbps capacity. . . . .	41
2.11	Average test accuracy among $N = 100$ clients achieved by the algorithms on the Synthetic, MNIST, and CIFAR-10 datasets. Cumulative importance assigned by the algorithms to the clients after $T = 200$ rounds on the Synthetic dataset. . . . .	53
2.12	<i>Convergence speed vs. Model bias</i> trade-off for different values of $\bar{\kappa}^2$ on the Synthetic dataset, for $\gamma = \delta = 0.5$ . . . . .	56
2.13	Effects of <i>data heterogeneity</i> on the Synthetic dataset after $T = 200$ rounds. . . . .	56
2.14	Estimation of the <i>clients’ activities</i> $(\hat{\pi}_k^{(t)}, \hat{\lambda}_k^{(t)})$ for different priors $t \in \{10^1, 10^{1.5}, 10^2, 10^{2.5}, 10^3, 10^{3.5}, 10^4\}$ and test accuracy after $T = 50$ rounds on the MNIST dataset. . . . .	56
2.15	Clients’ activities and CA-Fed’s inclusion/exclusion decisions in the presence of <i>spatial correlation</i> for different degrees of <i>intra-cluster/inter-cluster</i> data distributions. Average test accuracy after $T = 100$ rounds on the MNIST dataset. . . . .	57
3.1	Effect of client sampling rate (left) and FedEM number of mixture components $M$ (right) on the test accuracy for CIFAR10 [Kri09]. . . . .	81
3.10	Effect of system heterogeneity across clients on CIFAR-100 dataset. The size of the local datastore increases (resp. decreases) with $\Delta C$ for strong (resp. weak) clients. . . . .	95

3.11	Test accuracy vs capacity (local datastore size) for different methods on CIFAR-10. The capacity is normalized with respect to the initial size of the client’s dataset partition. . . .	95
4.1	A depiction of a data stream: The client/device, with a limited storage capacity ( $C = 3$ ), updates its local memory following a FIFO (First-In-First-Out) rule. This involves evicting the oldest samples from memory to make space for the most recent ones. Consequently, various samples, represented by distinct colors, reside in memory for varying durations. . . . .	98
4.2	Effect of $c_2/c_1$ on the historical clients relative importance $p_{\text{hist}}^*$ for different values of $N_{\text{hist}}/N$ , when $M = 50$ and $M_{\text{hist}} = 25$ . The dashed vertical line corresponds to our estimation of $c_2/c_1$ on CIFAR-10 experiments ( $\hat{c}_2/\hat{c}_1 = 0.15$ ). . . . .	108
4.3	The differences $\psi_{\text{hist}} - \psi^*$ (left), $\psi_{\text{uniform}} - \psi^*$ (center), and $\psi_{\text{hist}} - \psi_{\text{uniform}}$ (right) as a function of $N_{\text{hist}}/N$ for different values of $c_2/c_1$ , on CIFAR-10 dataset ( $N = 5 \times 10^5$ ) when $M = 50$ and $M_{\text{hist}} = 25$ . . . . .	109
4.4	Evolution of the test accuracy when using different values of $p_{\text{hist}}$ for CIFAR-10 (left) dataset, when $N_{\text{hist}}/N = 5\%$ (left), $20\%$ (center), and $50\%$ (right). The setting $p_{\text{hist}} = N_{\text{hist}}/N$ corresponds to <code>Uniform</code> strategy. . . . .	113
4.5	Evolution of the test accuracy when using different values of $p_{\text{hist}}$ for the synthetic dataset, when $N_{\text{hist}}/N = 5\%$ (left), $20\%$ (center), and $50\%$ (right). . . . .	114
4.6	Evolution of the test accuracy when using different values of $p_{\text{hist}}$ for CIFAR-100 dataset, when $N_{\text{hist}}/N = 5\%$ (left), $20\%$ (center), and $50\%$ (right). . . . .	115
4.7	Evolution of the test accuracy when using different values of $p_{\text{hist}}$ for FEMNIST dataset, when $M_{\text{hist}}/M = 5\%$ (left), $20\%$ (center), and $50\%$ (right). . . . .	116
4.8	Evolution of the test accuracy when using different values of $p_{\text{hist}}$ for Shakespeare dataset, when $M_{\text{hist}}/M = 5\%$ (left), $20\%$ (center), and $50\%$ (right). . . . .	116
4.9	Evolution of average regret across clients ( $\bar{R}_t$ ) as a function of number of samples and clients. Left: $\bar{R}_t$ for different values of $n$ . Center: $\bar{R}_t$ for different values of $C$ with each client receiving only one sample per time-step. Right: $\bar{R}_t$ for different values of $s$ . . . . .	126
4.10	Evolution of average regret across clients ( $\bar{R}'_t$ ) for CIFAR-10 ( <b>right</b> ) and MNIST ( <b>left</b> ). The curves are smoothed using a discount factor of 0.7. . . . .	127
E.15	Effect of the number of samples on the average test accuracy across clients unseen at training on CIFAR100 dataset. . . . .	287
E.16	Train loss, train accuracy, test loss, and test accuracy for CIFAR10 [Kri09]. . . .	290
E.17	Train loss, train accuracy, test loss, and test accuracy for CIFAR100 [Kri09]. . . .	291
E.18	Train loss, train accuracy, test loss, and test accuracy for EMNIST [Coh+17]. . . .	292
E.19	Train loss, train accuracy, test loss, and test accuracy for FEMNIST [Cal+19; McM+17]. . . . .	293
E.20	Train loss, train accuracy, test loss, and test accuracy for Shakespeare [Cal+19; McM+17]. . . . .	294
E.21	Train loss, train accuracy, test loss, and test accuracy for synthetic dataset. . . . .	295

- G.22 From left to the right: effect of  $c_2/c_1$  on the effective number of samples, the normalized gradient noise, and the historical clients relative importance  $p_{\text{hist}}^*$  for CIFAR-10 dataset ( $N = 5 \times 10^5$ ) and different values of  $N_{\text{hist}}/N$ , when  $M = 50$ , and  $M_{\text{hist}} = 25$ . The dashed vertical line corresponds to our estimation of  $c_2/c_1$  on CIFAR-10 experiments ( $\hat{c}_2/\hat{c}_1 = 0.15$ ). . . . . 321



# List of Tables

---

1.1	Comparison of existing privacy defenses by reference data treatment. In the second column, “relative level unspecified” means the target level of relative privacy requirements between training and reference data is not stated. In the third column, “single privacy level” means the reference data privacy leakage is evaluated at a single point on the utility-privacy curve. We use a dashed line (—) to convey that the defense either does not use reference data (column 2) or does not need to evaluate reference data privacy leakage (column 3). . . . .	17
1.2	Overview of the datasets, tasks, metrics and baseline models in FLamby. For Fed-Camelyon16 the two different sizes refer to the size of the dataset before and after tiling. . . . .	19
2.1	Algorithms to design the overlay $\mathcal{G}_o$ from the connectivity graph $\mathcal{G}_c$ . . . . .	28
2.2	Statistics of iNaturalist dataset distribution for different networks. . . . .	36
2.3	Statistics of LEAF dataset distribution for AWS North America network (22 silos). . . . .	37
2.4	Datasets and Models. Mini-batch gradient computation time with NVIDIA Tesla P100. . . . .	38
2.5	Sub-iNaturalist training over different networks. 1 Gbps core links capacities, 10 Gbps access links capacities. One local computation step ( $s = 1$ ). . . . .	39
2.6	iNaturalist training over different networks. 1 Gbps core links capacities, 1 Gbps access links capacities. One local computation step ( $s = 1$ ). . . . .	39
3.1	Average computation time and used GPU for each dataset. . . . .	79
3.2	Test accuracy: average across clients / bottom decile. . . . .	80
3.3	Average test accuracy across clients unseen at training (train accuracy in parenthesis). . . . .	81
3.4	Datasets and models. . . . .	86
3.5	Test accuracy: average across clients / bottom decile. . . . .	88
3.6	Average test accuracy across clients unseen at training (train accuracy between parentheses). . . . .	88
4.1	Average test accuracy across clients for different datasets in the settings when $N_{\text{hist}}/N = 50\%$ . . . . .	110
4.2	Datasets and models. . . . .	112
4.3	Average test accuracy across clients for different datasets in the settings when $N_{\text{hist}}/N = 20\%$ . . . . .	113
4.4	Average test accuracy across clients for different datasets in the settings when $N_{\text{hist}}/N = 5\%$ . . . . .	114
4.5	Average test accuracy across clients for different datasets in the settings when $N_{\text{hist}}/N = 50\%$ . . . . .	115



1	Sub-iNaturalist training over different networks. 1 Gbps core links capacities, 10 Gbps access links capacities. Five local computation steps. . . . .	194
2	Sub-iNaturalist training over different networks. 1 Gbps core links capacities, 10 Gbps access links capacities. Ten local computation steps. . . . .	194
3	Average computation time and used CPU/GPU for each dataset. . . . .	235
4	Learning rates $\eta$ and $\bar{\eta}$ used for the experiments in Figure 2.11. . . . .	235
5	Test accuracy: average across clients. . . . .	286
6	Test and train accuracy comparison across different tasks. For each method, the best test accuracy is reported. For FedEM we run only $\frac{K}{M}$ rounds, where $K$ is the total number of rounds for other methods— $K = 80$ for Shakespeare and $K = 200$ for all other datasets—and $M = 3$ is the number of components used in FedEM. . . . .	288
7	Test accuracy under 20% client sampling: average across clients with +/- standard deviation over 3 independent runs. All experiments with 1200 communication rounds. . . . .	289

# List of Algorithms

---

1	FedAvg: Federated Averaging [McM+17, Algorithm 1]. . . . .	5
2	FedOpt Algorithm [Red+21, Algorithm 1]. . . . .	6
3	Approximation algorithm for MCT on node-capacitated networks. . . . .	31
4	$\delta$ -PRIM[AR19] . . . . .	32
5	Time Simulator . . . . .	32
6	CA-Fed (Correlation-Aware FL) . . . . .	51
7	FedEM: Federated Expectation-Maximization . . . . .	70
8	D-FedEM: Fully Decentralized Federated Expectation-Maximization . . . . .	72
9	Basic Surrogate Optimization . . . . .	74
10	Federated Surrogate Optimization . . . . .	75
11	Fully-Decentralized Federated Surrogate Optimization . . . . .	77
12	kNN-Per (Typical usage) . . . . .	84
13	Meta Algorithm for Federated Learning from Data Streams . . . . .	103
14	Federated Expectation-Maximization Online Mirror Descent (FEM-OMD) . . . . .	118
15	FEM-OMD for Gaussian Mixture Models . . . . .	121
16	Online Mirror Descent with Incorrect Gradients . . . . .	122
17	FEM-OMD for discriminative models . . . . .	125



# **Appendix**



# Background

---

## A Background on Numeric Optimization

In this section, we revisit essential concepts utilized in numerical optimization, drawing insights from the comprehensive handbook by Garrigos et al. [GG23].

### A.1 Differentiability

**Definition 2.** (*Differentiability and Jacobian*) A function  $f : \mathbb{R}^m \mapsto \mathbb{R}^n$  is said to be differentiable at point  $\mathbf{x}_0 \in \mathbb{R}^m$  if there exists a linear map  $\mathbf{D}_f(\mathbf{x}_0) : \mathbb{R}^m \mapsto \mathbb{R}^n$  such that

$$\lim_{\mathbf{h} \rightarrow 0} \frac{\|f(\mathbf{x}_0 + \mathbf{h}) - f(\mathbf{x}_0) - \mathbf{D}_f(\mathbf{x}_0) \cdot \mathbf{h}\|_{\mathbb{R}^n}}{\|\mathbf{h}\|_{\mathbb{R}^m}} = 0.$$

If the function  $f$  is differentiable at  $\mathbf{x}_0$ , then all of its first partial derivatives exist at  $\mathbf{x}_0$ , and the linear map  $\mathbf{D}_f(\mathbf{x}_0)$  is given by the **Jacobian** matrix  $\mathbf{J}(\mathbf{x}_0) \in \mathbb{R}^{n \times m}$ , which is the matrix defined by the first partial derivatives of  $f$ :

$$\mathbf{J}_{i,j}(\mathbf{x}) = \frac{\partial f_i}{\partial x_j}(\mathbf{x}), \quad i = 1, \dots, n, \quad j = 1, \dots, m,$$

where we write  $f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))$ .

**Definition 3.** (*Gradient*) If  $f : \mathbb{R}^d \mapsto \mathbb{R}$  is differentiable, then  $\mathbf{J}(\mathbf{x}) \in \mathbb{R}^{1 \times d}$  is a row vector, whose transpose is called the **gradient** of  $f$  at  $\mathbf{x}$ :  $\nabla f(\mathbf{x}) = \mathbf{J}(\mathbf{x})^\top \in \mathbb{R}^{d \times 1}$ .

**Definition 4.** (*Hessian*) Let  $f : \mathbb{R}^d \mapsto \mathbb{R}$  be twice differentiable, and  $\mathbf{x} \in \mathbb{R}^d$ . Then we note  $\nabla^2 f(\mathbf{x}) \in \mathbb{R}^{d \times d}$  the **Hessian** of  $f$  at  $\mathbf{x}$ , which is the matrix defined by its second-order partial derivatives:

$$\left[\nabla^2 f(\mathbf{x})\right]_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}(\mathbf{x}), \quad i, j = 1, \dots, d.$$

### A.2 Lipschitzianity and Smoothness

**Definition 5.** (*Lipschitzianity*) Let  $f : \mathbb{R}^m \mapsto \mathbb{R}^n$ , and  $L > 0$ . We say that  $f$  is  **$L$ -Lipschitz** if for all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$ ,

$$\|f(\mathbf{x}) - f(\mathbf{y})\|_{\mathbb{R}^n} \leq L \cdot \|\mathbf{x} - \mathbf{y}\|_{\mathbb{R}^m}.$$

A differentiable function is  $L$ -Lipschitz if and only if its differential is uniformly bounded by  $L$ .

**Lemma A.1.** ([GG23, Lemm 2.6]) Let  $f : \mathbb{R}^m \mapsto \mathbb{R}^n$ , and  $L > 0$ . Then,  $f$  is  $L$ -Lipschitz if and only if for all  $\mathbf{x} \in \mathbb{R}^m$ ,

$$\|\mathbf{D}_f(\mathbf{x})\| \leq L.$$

**Definition 6.** (*Smoothness*) Let  $f : \mathbb{R}^d \mapsto \mathbb{R}$ , and  $L > 0$ . We say that  $f$  is  $L$ -smooth if it is differentiable and if  $\nabla f : \mathbb{R}^d \mapsto \mathbb{R}^d$  is  $L$ -Lipschitz: for all  $\mathbf{x} \in \mathbb{R}^d$ ,

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|.$$

**Lemma A.2.** ([GG23, Lemm 2.25]) If  $f : \mathbb{R}^d \mapsto \mathbb{R}$  is  $L$ -smooth then, for all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ ,

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|^2.$$

**Lemma A.3.** ([GG23, Lemm 2.26]) Let  $f : \mathbb{R}^d \mapsto \mathbb{R}$  be a twice differentiable  $L$ -smooth function. Then for all  $\mathbf{x} \in \mathbb{R}^d$ , for every eigenvalue  $\lambda$  of  $\nabla^2 f(\mathbf{x})$ , we have  $|\lambda| \leq L$ .

### A.3 Convexity

**Definition 7.** (*Convexity*) We say that  $f : \mathbb{R}^d \mapsto \mathbb{R}$  is convex if, for all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ , and for all  $t \in [0, 1]$ ,

$$f(t\mathbf{x} + (1-t)\mathbf{y}) \leq tf(\mathbf{x}) + (1-t)f(\mathbf{y}).$$

The next two lemmas characterize the convexity of a function with the help of first and second-order derivatives.

**Lemma A.4.** ([GG23, Lemma 2.8]) If  $f : \mathbb{R}^d \mapsto \mathbb{R}$  is convex and differentiable then, for all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ ,

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle$$

**Lemma A.5.** ([GG23, Lemma 2.9]) Let  $f : \mathbb{R}^d \mapsto \mathbb{R}$  be convex and twice differentiable. Then for all  $\mathbf{x} \in \mathbb{R}^d$ , for every eigenvalue  $\lambda$  of  $\nabla^2 f(\mathbf{x})$ , we have  $\lambda \geq 0$ .

**Definition 8.** (*Strong Convexity*) Let  $f : \mathbb{R}^d \mapsto \mathbb{R}$ , and  $\mu > 0$ . We say that  $f$  is  $\mu$ -strongly convex if, for every  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ , and every  $t \in [0, 1]$  we have that

$$\mu \frac{t(t-1)}{2} \|\mathbf{x} - \mathbf{y}\|^2 + f(t\mathbf{x} + (1-t)\mathbf{y}) \leq tf(\mathbf{x}) + (1-t)f(\mathbf{y}).$$

The next two lemmas characterize the strong convexity of a function with the help of first and second-order derivatives.

**Lemma A.6.** ([GG23, Lemma 2.14]) If  $f : \mathbb{R}^d \mapsto \mathbb{R}$  is  $\mu$ -strongly convex and differentiable then, for all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ ,

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\mu}{2} \|\mathbf{x} - \mathbf{y}\|^2$$

**Lemma A.7.** ([GG23, Lemma 2.15]) Let  $f : \mathbb{R}^d \mapsto \mathbb{R}$  be  $\mu$ -strongly convex and twice differentiable. Then for all  $\mathbf{x} \in \mathbb{R}^d$ , for every eigenvalue  $\lambda$  of  $\nabla^2 f(\mathbf{x})$ , we have  $\lambda \geq \mu > 0$ .

## B Background on Graph Theory

We now list concepts of graph theory which will be used later on.

- **Predecessor, successor, neighbour:** If in a graph  $(i, j) \in \mathcal{E}$ , then  $i$  is called a predecessor of  $j$ ,  $j$  is called a successor of  $i$  and  $j$ , resp.  $i$  is called a neighbour of  $i$ , resp.  $j$ . The set of predecessors of  $j$  is indicated by  $\pi(j)$  (or  $\mathcal{N}_j^-$ ), the set of all successors of  $i$  is denoted  $\sigma(i)$  (or  $\mathcal{N}_i^+$ ) and the set of neighbours of  $i$  is denoted  $\mathcal{N}_i$ . Note that in the case of undirected graphs,  $\mathcal{N}_i = \pi(i) = \sigma(i)$ .
- **Path, circuit and full walk:** A path is a sequence of nodes  $(i_1, \dots, i_p)$ ,  $p > 1$ , such that  $i_j \in \pi(i_{j+1})$ ,  $j = 1, \dots, p-1$ . An elementary path is a path where no node appears more than once. When the initial node and the final node coincide, one speaks of circuit. A circuit  $C = (i_1, \dots, i_p = i_1)$  is an elementary circuit if the path  $(i_1, \dots, i_{p-1})$  is elementary, an elementary circuit is sometimes referred to as a cycle. If a cycle spans all vertices of the graph it is called a *Hamiltonian cycle*. The length of circuit  $C = (i_1, \dots, i_p)$  is the number of the arcs of which it is composed, i.e.,  $|C| = p$ , and its weight is the sum of the weights of its arcs, i.e.,  $d(C) = \sum_{k=1}^{p-1} d(i_k, i_{k+1})$ . We define also the notion of "Full-Walk" on a graph as the result of a depth-first search (DFS) of this graph. We define the weight of a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  as the weight of the circuit made of all its arcs.
- **Subgraph, spanning subgraph:** Given a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , a graph  $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$  is said to be a subgraph of  $\mathcal{G}$  if  $\mathcal{V}' \subset \mathcal{V}$  and if  $\mathcal{E}'$  consists of the set of arcs of  $\mathcal{G}$  which have their destination and origins in  $\mathcal{V}'$ .  $\mathcal{G}'$  is said to be a spanning subgraph if  $\mathcal{V}' = \mathcal{V}$ .
- **Strongly connected graph:** A digraph is said to be *strongly connected* or *strong* if for any two different nodes  $i$  and  $j$  in  $\mathcal{V}$  there exists a path from  $i$  to  $j$ .
- **Optimal tour:** In a Hamiltonian graph (i.e., a graph having a Hamiltonian cycle) if a Hamiltonian cycle is the Hamiltonian cycle with the lowest weight we say that it is an *optimal tour*. Finding the optimal tour in a complete graph is a well known problem and is referred to as the Traveling Salesman Problem (TSP), see for example [App+07] for the definition of this problem.
- **Tree, acyclic graph and Minimum Spanning Tree (MST):** A tree, or equivalently a connected acyclic undirected graph, is an undirected graph in which any two vertices are connected by exactly one path,=. An acyclic graph  $T$  is said to be a spanning tree of an undirected graph  $\mathcal{G}$  if  $T$  is a spanning subgraph of  $\mathcal{G}$ .  $T$  is said to be an MST of  $\mathcal{G}$  if it has a minimal weight, sum of the weights of all its edges, among all spanning trees of  $\mathcal{G}$ .
- **Cut, cut-set and cut property:** A *cut* is a partition of the vertices of a graph into two disjoint subsets. For a cut  $c$ , the cut-set is the set of edges connecting two nodes from the two disjoint subsets. In a tree, deleting an edge, induces a partition of the vertices sets into two disjoint subsets. For any cut  $c$  of the graph, if the weight of an edge  $e$  in the cut-set of  $c$  is strictly smaller than the weights of all other edges of the cut-set of  $c$ , then this edge belongs to all MSTs of the graph.





# Proofs and Experiments

---

## C Throughput-Optimal Topology Design for Cross-Silo Federated Learning

### C.1 Proofs

We use some graph terminology and notation introduced in Appendix B.

#### C.1.1 Proof of Proposition 2.1.1

When we require the overlay  $\mathcal{G}_o$  to be undirected, if we include link  $(i, j) \in \mathcal{G}_c$  then we will also include link  $(j, i)$ . It is then convenient to consider the undirected graph  $\mathcal{G}_c^{(u)} = (\mathcal{V}, \mathcal{E}_c^{(u)})$ , where  $(i, j) \in \mathcal{E}_c^{(u)}$  iff  $(i, j) \in \mathcal{E}_c$ , from which we want to extract an undirected strong subgraph  $\mathcal{G}_o$  with minimal cycle time. We also associate to each edge  $(i, j) \in \mathcal{G}_c^{(u)}$  the weight  $d_c^{(u)}(i, j) = (d_c(i, j) + d_c(j, i))/2$ . Remember that  $d_c(i, j)$  is defined as follows

$$d_c(i, j) \triangleq s \times T_c(i) + l(i, j) + M/A(i', j').$$

Note that an undirected weighted graph can be also seen as a particular directed graph where for each link  $(i, j)$  in one direction, there exists a link  $(j, i)$  with the opposite direction and the same weight. The concept of cycle time can then immediately be extended to undirected graphs.

**Lemma C.1.** *Consider the undirected weighted graph  $\mathcal{G}_c^{(u)} = (\mathcal{V}, \mathcal{E}_c^{(u)})$ , where  $(i, j) \in \mathcal{E}_c^{(u)}$  iff  $(i, j) \in \mathcal{E}_c$ . There exists a spanning tree of  $\mathcal{G}_c^{(u)}$  in the set of solutions MCT when  $\mathcal{G}_c$  is edge-capacitated and  $\mathcal{G}_o$  is required to be undirected.*

*Proof.* MCT is a discrete optimization problem on a finite set, \* thus the set of solution of MCT is non-empty. Suppose by contradiction that the set of solutions does not contain any spanning tree of  $\mathcal{G}_c$  and consider  $\mathcal{G}_o^*$  an element in the set of solutions which is an undirected spanning subgraph of  $\mathcal{G}_c$ .

As  $\mathcal{G}_o^*$  is not a spanning tree and it is strongly connected, there exists circuits in  $\mathcal{G}_o^*$ . For any circuit  $C = (i_1, i_2, \dots, i_p = i_1)$  in  $\mathcal{G}_o^*$ , we consider the edge  $e_C$ , such that  $d_c^{(u)}(e_C) = \max_{k=1, \dots, p-1} d_c^{(u)}(i_k, i_{k+1})$ . The graph obtained from  $\mathcal{G}_o^*$  by deleting  $e_C$  for every circuit  $C$  of  $\mathcal{G}_o^*$  is a spanning tree of  $\mathcal{G}_c^{(u)}$  and its cycle time is not greater then the cycle time of  $\mathcal{G}_o^*$ . Thus, it is also a solution of MCT, and this contradicts the fact that no spanning tree is in the set of solutions.  $\square$

**Lemma C.2.** *Consider an undirected tree  $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ , weighted with a delay function  $d_c^{(u)} : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}_+$ . Its cycle time is  $\tau(\mathcal{T}) = \max_{\{i, j\} \in \mathcal{E}} d_c^{(u)}(i, j)$ .*

---

\*The set of subgraphs of an undirected graph  $\mathcal{G}_c$  is finite.

*Proof.* The cycle time of  $\mathcal{T}$  is given by Equation (2.5).  $\tau(\mathcal{T}) = \max_C \frac{w(C)}{|C|}$ , where the maximum is taken over all the elementary circuits of  $\mathcal{T}$ . Since  $\mathcal{T}$  is acyclic, the only elementary circuits of  $\mathcal{T}$  are of the form  $(i, j, i)$  for some  $\{i, j\} \in \mathcal{E}$ . By definition  $|(i, j, i)| = 2$  and  $w((i, j, i)) = d^{(u)}_c$ . It follows that  $\tau(\mathcal{T}) = \max_{\{i,j\} \in \mathcal{E}} \frac{d^{(u)}_c(i,j) + d^{(u)}_c(j,i)}{2} = d^{(u)}_c(i, j)$ .  $\square$

**Proposition 2.1.1.** *Consider an undirected weighted graph  $\mathcal{G}_c^{(u)} = (\mathcal{V}, \mathcal{E}_c^{(u)})$ , where  $(i, j) \in \mathcal{E}_c^{(u)}$  iff  $(i, j) \in \mathcal{E}_c$  and  $(j, i) \in \mathcal{E}_c$  and where  $(i, j) \in \mathcal{E}_c^{(u)}$  has weight  $d^{(u)}_c(i, j) = (d_o(i, j) + d_o(j, i))/2$ . A minimum weight spanning tree of  $\mathcal{G}_c^{(u)}$  is a solution of MCT when  $\mathcal{G}_c$  is edge-capacitated and  $\mathcal{G}_o$  is required to be undirected.*

*Proof.* Denote  $\mathcal{G}^*$  the solution of MCT when  $\mathcal{G}_c$  is edge-capacitated and  $\mathcal{G}_o$  is required to be undirected, and denote  $\mathcal{T}^*$  an MST of  $\mathcal{G}_c^{(u)}$  weighted with  $d^{(u)}_c$ , and suppose by contradiction that  $\tau(\mathcal{T}^*) > \tau(\mathcal{G}^*)$ . By Lemma C.2, it follows that there is an edge  $e_{\mathcal{T}^*}$  of  $\mathcal{T}^*$  such that  $d^{(u)}_c(e_{\mathcal{T}^*}) = \tau(\mathcal{T}^*)$ . Moreover, it follows that  $\forall e \in \mathcal{E}(\mathcal{G}^*), d^{(u)}_c(e) \leq \tau(\mathcal{G}^*) < \tau(\mathcal{T}^*) = d^{(u)}_c(e_{\mathcal{T}^*})$ . If we remove  $e_{\mathcal{T}^*}$  from  $\mathcal{T}^*$ , the two components define a cut of  $\mathcal{G}_c$ . The edge of  $\mathcal{G}^*$ , say  $e_{cut}$  belonging to the cut-set is such that  $d^{(u)}_c(e_{cut}) < d^{(u)}_c(e_{\mathcal{T}^*})$ , and this is a contradiction with the cut property satisfied by minimum cost spanning trees.  $\square$

### C.1.2 Proof of Proposition 2.1.2

**Proposition 2.1.2.** *MCT is NP-hard even when  $\mathcal{G}_c$  is a complete Euclidean edge-capacitated graph.*

*Proof.* When  $\mathcal{G}_c$  is an edge-capacitated graph,  $d_c(i, j) = s \times T_c(i) + l(i, j) + \frac{M}{A(i, j)}$ .  $\mathcal{G}_c$  is complete and Euclidean means that  $d_c(i, j) = d_c(j, i)$ , for all  $(i, j) \in \mathcal{V} \times \mathcal{V}$  and that  $d_c$  verifies triangular inequality, i.e.,  $d_c(i, j) \leq d_c(i, k) + d_c(k, j)$ , for every  $i, j, k \in \mathcal{V}$ .

We consider the decision problem associated to the particular case of MCT when  $\mathcal{G}_c$  is an Euclidean edge-capacitated graph, namely Euclidean Edge-Capacitated Minimal Cycle Time - Decision- (MCT-DECISION) and we prove that it is NP-complete.

Euclidean Edge-Capacitated Minimal Cycle Time - Decision (MCT-DECISION)

**Input:** A strong digraph  $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$ , delays function  $d_c$  and a real number  $\tau_0$

**Question:** Is there a strong spanning subdigraph of  $\mathcal{G}_c$  with cycle time at most  $\tau_0$ ?

We first prove that MCT-DECISION is NP.\* Several algorithms (e.g., Karp's Algorithm [DG98]) determines the cycle time of a given graph in a polynomial time. Thus for a proposed solution of MCT-DECISION, we can compute its cycle time in polynomial time, and we can verify if the graph is strongly connected using for example depth first search. It follows that MCT-DECISION is NP.

To prove that MCT-DECISION is NP-complete, we show that Hamiltonian Cycle (HC) can be reduced in a polynomial time into MCT-DECISION, i.e.,  $HC \leq_p$  MCT-DECISION.

Hamiltonian cycle problem is the following decision problem:

Hamiltonian Cycle (HC)

**Input:** A strongly connected directed graph  $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ .

**Question:** Is there a Hamiltonian cycle in  $\mathcal{D}$ ?

---

\*A decision problem is NP if we can verify in a polynomial time that the answer for a given instance is YES.

Given an instance of HC with a directed graph  $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ , we construct an instance of MCT-DECISION with a complete digraph  $\mathcal{G}_c = (\mathcal{V}, \mathcal{V} \times \mathcal{V})$ , a real number  $\tau_0 = \frac{N+2}{N}$  where  $N$  is the size of  $\mathcal{V}$ , and delay function  $d_c$ , where for a given arbitrary choice of vertex  $v_0$ ,  $d_c$  is defined as:

$$d_c(i, j) = \begin{cases} 1 & \text{if } ((i, j) \in \mathcal{E}) \wedge (j \neq v_0) \wedge (i \neq v_0), \\ 2 & \text{if } [((i, j) \in \mathcal{E}) \wedge ((j = v_0) \vee (i = v_0))] \vee [((i, j) \notin \mathcal{E}) \wedge (j \neq v_0)], \\ 3 & \text{otherwise.} \end{cases}$$

The constructed digraph  $\mathcal{G}_c$  is complete and the delays are symmetric and verify the triangular inequality. In fact for three distinct nodes  $i, j$  and  $k$  in  $\mathcal{V}$ , two cases are possible: 1) If they are all different from  $v_0$ , then  $d_c(i, j) \leq 2$  and  $2 \leq d_c(i, k) + d_c(k, j)$ , it follows that  $d_c(i, j) \leq d_c(i, k) + d_c(k, j)$ ; 2) If one of them is  $v_0$ , say  $k = v_0$ , then  $d(i, j) \leq 2 \leq d_c(i, v_0) \leq d_c(i, v_0) + d_c(v_0, j)$ , where the first inequality is due to the fact that  $d(i, j) = 3$  only when  $j = v_0$ , and  $d(i, v_0) \leq 3 \leq d_c(i, j) + d_c(j, v_0)$ , where the second inequality is due to the fact  $2 \leq d_c(j, v_0)$ .

If  $\mathcal{D}$  has a Hamiltonian cycle, then the graph induced by this cycle is a strong spanning subdigraph of  $\mathcal{G}_c$  and its cycle time is  $\tau_{\text{HC}} = \frac{1 \times (N-2) + 2 + 2}{N} = \frac{N+2}{N} \leq \tau_0$ .

If  $\mathcal{G}_c$  has a strong spanning sub-digraph, say  $\mathcal{G}^*$ , having a cycle time  $\tau^* \leq \frac{N+2}{N}$ , let  $C$  be an elementary circuit of  $\mathcal{G}^*$  containing  $v_0$  (such a circuit always exists because the graph is strongly connected). By definition of cycle time,  $\frac{d_c(C)}{|C|} \leq \tau^* = 1 + \frac{2}{N}$ . We are going to prove that  $C$  is a Hamiltonian cycle of  $\mathcal{D}$ .

We prove first by contradiction that  $C$  contains only the arcs from  $\mathcal{E}$ . Suppose by contradiction that there exists an arc  $(i, j) \notin \mathcal{E}$  in  $C$ , two cases are possible:

1. If  $j \neq v_0$ , then  $d_c(i, j) = 2$  and since  $v_0 \in C$ , there exist two nodes  $v_0^- \in \sigma(v_0)$  and  $v_0^+ \in \pi(v_0)$  in  $C$ . It follows that  $d_c(C) \geq d_c(i, j) + d(v_0^+, v_0) + d_c(v_0, v_0^-) + 1 \times (|C| - 3) \geq 2 + 2 + 2 + |C| - 3 = |C| + 3$ . Since  $C$  is an elementary circuit, it follows that  $|C| \leq N$ , thus  $\frac{d_c(C)}{|C|} \geq 1 + \frac{3}{N}$ , and this contradicts  $\frac{d_c(C)}{|C|} \leq 1 + \frac{2}{N}$ .
2. If  $j = v_0$ , let  $v_0^-$  the successor of  $v_0$  in  $C$ , it follows that  $d_c(C) \geq d_c(i, v_0) + d(v_0, v_0^-) + 1 \times (|C| - 2) \geq 3 + 2 + |C| - 2 = 3 + |C|$ , thus  $\frac{d_c(C)}{|C|} \geq 1 + \frac{3}{|C|}$ , and using the same argument as for the first case we get a contradiction.

It follows that any arc of  $C$  is in  $\mathcal{E}$ .

We prove next that  $C$  is a Hamiltonian Cycle, i.e.,  $|C| = N$ . Since  $v_0 \in C$ , there exist two nodes  $v_0^+ \in \sigma(v_0)$  and  $v_0^- \in \pi(v_0)$  in  $C$ , it follows that  $d_c(C) = d_c(v_0^-, v_0) + d_c(v_0, v_0^+) + 1 \times (|C| - 2) = 2 + 2 + |C| - 2 = 2 + |C|$ .

Since  $\frac{d_c(C)}{|C|} \leq \tau^* = 1 + \frac{2}{N}$ , it follows that  $1 + \frac{2}{|C|} \leq 1 + \frac{2}{N}$ , thus  $|C| \geq N$ . As  $C$  is an elementary circuit it follows that  $|C| = N$ , i.e.,  $C$  is a Hamiltonian cycle. Since  $C$  is a circuit containing only arcs from  $\mathcal{D}$ , it follows that  $\mathcal{D}$  has a Hamiltonian cycle.

So we have proved that  $\mathcal{D}$  has a Hamiltonian cycle if and only if  $\mathcal{G}_c$  has strong spanning subdigraph of cycle time at most  $\tau_0 = \frac{N+2}{N}$ . It follows that MCT-DECISION is NP-complete, thus MCT is NP-hard even when  $\mathcal{G}_c$  is a complete Euclidean edge-capacitated graph.  $\square$

### C.1.3 Proof of Proposition 2.1.3

Under the assumption that the connectivity topology is Euclidean (delays are symmetric and verify triangular inequality), we first show that the solution of Traveling Salesman Problem (TSP) [GP06]

is guaranteed to be within a  $2N$ -multiplicative factor of the solution of MCT (Lemma C.3). As a result, the Christofides algorithm [MPT02] which is a 1.5-approximation algorithm for TSP, is a  $3N$ -approximation algorithm for MCT (Prop. 2.1.3).

**Lemma C.3.** *Consider an Euclidean digraph  $\mathcal{G}_c$  with  $N$  nodes and let  $\mathcal{H}^*$  denote its optimal tour. Then  $\frac{d_c(\mathcal{H}^*)}{|\mathcal{H}^*|} \leq 2N \times \tau_*$ , where  $\tau_*$  is the optimal cycle time that can be achieved by a strong spanning subdigraph of  $\mathcal{G}_c$ .*

*Proof.* Let  $\mathcal{G}^*$  be a spanning digraph of  $\mathcal{G}_c$  with optimal cycle time  $\tau^*$ .

Let  $\{\mathcal{C}_i\}_{i=1,\dots,c}$  be a minimal set of elementary circuits of  $\mathcal{G}_*$ , so that  $\cup_{i=1}^c \mathcal{C}_i = \mathcal{G}_*$  and  $\cup_{i \neq j} \mathcal{C}_i \neq \mathcal{G}_*$  for each  $j$ . Consider an auxiliary graph  $\mathcal{G}'$  whose  $c$  nodes represent the  $c$  circuits and whose links correspond to two circuits sharing a node. Let  $\mathcal{T}$  be a spanning tree of  $\mathcal{G}'$ . Starting from the root of  $\mathcal{T}$ , we can define an order of the nodes in each circuit and an order of the children of each circuit as follows. Given the orientation of the circuit corresponding to the root, consider the first node they share with each child. We order the children according to such order (solving arbitrarily possible ties). For each child we reorder its nodes starting from the node they share with the father and following the orientation of the circuit. We consider then the ordered traversal of the circuits  $\Gamma = (\mathcal{C}_{i_1}, \mathcal{C}_{i_2}, \dots, \mathcal{C}_{i_{2c+1}} = \mathcal{C}_{i_1})$  obtained using DFS on  $\mathcal{T}$  and visiting the children according to the order introduced above.

From  $\Gamma$  we can build two closed walks  $\mathcal{W}_1$  and  $\mathcal{W}_2$ , both spanning all nodes of  $\mathcal{G}^*$ . The walk  $\mathcal{W}_1$  is built by considering all circuits in the order they appear in  $\Gamma$ , and then concatenating their nodes as follows. The first time we visit one circuit we take all nodes in the circuit in their order (but the last one in each circuit that coincides with the first one). When we come back to the circuit, we only pick the nodes needed to move to the following circuit in  $\Gamma$ . The walk  $\mathcal{W}_2$  is built by considering the  $c$  circuits in the order they first appear in  $\Gamma$ , and then again concatenating their nodes (but the last one in each circuit that coincides with the first one). Both sequences of nodes define walks as  $\mathcal{G}_c$  is Euclidean and then complete. The length of  $\mathcal{W}_2$  is  $|\mathcal{W}_2| = \sum_{i=1}^c |\mathcal{C}_i| \leq N^2$ , as we can have at most  $N - 1$  elementary circuits and each of them has length at most  $N$ .

We observe that  $d_c(\mathcal{W}_1) \leq 2 \sum_{i=1}^c d_c(\mathcal{C}_i)$  as the walk  $\mathcal{W}_2$  passes through each link in each circuit  $\mathcal{C}_i$  at most twice: it walks through the first  $|\mathcal{C}_i| - 1$  edges of  $\mathcal{C}_i$  the first time it visits  $\mathcal{C}_i$ , and uses once more the edges in  $\mathcal{C}_i$  to visit the other circuits and go back to the root. As  $\mathcal{W}_2$  is a sublist of the nodes in  $\mathcal{W}_1$  and delays satisfy the triangle inequality, it holds  $d_c(\mathcal{W}_2) \leq d_c(\mathcal{W}_1)$ .

Finally, from the walk  $\mathcal{W}_2$  we can extract a Hamiltonian cycle  $\mathcal{H}$  that has an even smaller delay. Let  $\mathcal{H}^*$  be an optimal tour. It follows

$$\tau(\mathcal{H}^*) = \frac{d_c(\mathcal{H}^*)}{|\mathcal{H}^*|} \leq \frac{d_c(\mathcal{H})}{|\mathcal{H}^*|} \leq \frac{d_c(\mathcal{W}_2)}{|\mathcal{H}^*|} \quad (\text{C.1})$$

$$= \frac{|\mathcal{W}_2| d_c(\mathcal{W}_2)}{|\mathcal{H}^*| |\mathcal{W}_2|} \quad (\text{C.2})$$

$$\leq \frac{N^2 d_c(\mathcal{W}_1)}{N \sum_{i=1}^c |\mathcal{C}_i|} \quad (\text{C.3})$$

$$\leq 2N \frac{\sum_{i=1}^c d_c(\mathcal{C}_i)}{\sum_{i=1}^c |\mathcal{C}_i|} \quad (\text{C.4})$$

$$\leq 2N \max_{i=1,\dots,c} \frac{d_c(\mathcal{C}_i)}{|\mathcal{C}_i|} = \tau^*. \quad (\text{C.5})$$

□

**Proposition 2.1.3.** *Christofides' algorithm [MPT02] is a  $3N$ -approximation algorithm for MCT when  $\mathcal{G}_c$  is edge-capacitated and Euclidean.*

*Proof.* Christofides algorithm provides a  $\frac{3}{2}$ -approximation for the traveling salesman problem TSP defined in [App+07].\* Given an instance of MCT let  $\hat{C}$  denote the output of Christofides algorithm and  $C^*$  denote the optimal tour of  $\mathcal{G}_c$ . It follows that  $d_c(\hat{C}) \leq \frac{3}{2}d_c(C^*)$ . Since both  $\hat{C}$  and  $C^*$  are Hamiltonian cycles,  $|\hat{C}| = |C^*|$ . Using Lemma C.3. it follows that  $\frac{d_c(\hat{C})}{|\hat{C}|} \leq 2N \times \frac{3}{2} \times \tau_* = 3N \times \tau_*$ . Thus the graph obtained using only the edges of  $\hat{C}$  is a  $3N$ -approximation of the MCT problem when  $\mathcal{G}_c$  is edge-capacitated and Euclidean.  $\square$

#### C.1.4 Proof of Proposition 2.1.4

We prove that in a node-capacitated network, MCT is NP-hard even when  $\mathcal{G}_o$  is required to be undirected. We start introducing the associated decision problem:

MCT-U-Decision

**Input:** A strongly connected directed graph  $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$ , model size  $M$ ,  $\{C_{UP}(i), C_{DN}(j), l(i, j), A(i', j'), T_c(i), \forall(i, j) \in \mathcal{E}_c\}$ , and a constant  $\tau_0 > 0$

**Question:** Is there a strong spanning undirected subgraph  $\mathcal{G}_o$  of  $\mathcal{G}_c$ , such that  $\tau(\mathcal{G}_o) \leq \tau_0$ ?

MCT-U-Decision is closely related to the *degree-constrained spanning tree* (DCST) defined below:

Degree-constrained spanning tree (DCST)

**Input:** An  $N$ -node connected undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ; positive integer  $k \leq n$

**Question:** Does  $\mathcal{G}$  have a spanning tree in which no node has degree greater than  $k$ ?

DCST is a simpler version of  $\delta$ -MBST, where we look for a spanning tree with degree at most  $k$  and minimum bottleneck.

DCST is NP-complete.  $\dagger$  For example for  $k = 2$  it can be shown by a reduction from HC.

**Proposition 2.1.4.** *In node-capacitated networks MCT is NP-hard even when the overlay is required to be undirected.*

*Proof.* Our proof is based on a reduction of DCST to MCT-U-Decision.

Given an instance of DCST with an  $N$ -node connected undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  and a positive integer  $k \leq N$ , we define an instance of MCT-U-Decision with a connected graph  $\Pi(\mathcal{G}) := \mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$  where, for each node  $v$  in  $\mathcal{V}$ , there are two nodes  $v^{(1)}$  and  $v^{(2)}$  in  $\mathcal{V}_c$  and  $(v^{(1)}, v^{(2)}) \in \mathcal{E}_c$ , and for an arc  $(v_i, v_j) \in \mathcal{E}$ , there is an arc  $(v_i^{(1)}, v_j^{(1)})$  in  $\mathcal{E}_c$ . We set  $\frac{M}{C_{UP}(v^{(1)})} = 1$ ,  $\frac{M}{C_{UP}(v^{(2)})} = k + 1$  for all  $v \in \mathcal{V}$ ,  $T_c(i) = 0$ ,  $C_{DN}(i) = \infty$  for all  $i \in \mathcal{V}_c$ , and  $l(i, j) = 0$  for all  $(i, j) \in \mathcal{E}_c$ . Finally, we consider  $\tau_0 = k + 1$ .

Suppose that  $\mathcal{G}$  has a spanning tree  $\mathcal{T} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$  in which no node has degree greater than  $k$ , and denote  $\mathcal{T}_c = \Pi(\mathcal{T})$  (i.e., we apply the same mapping described above).  $\mathcal{T}_c$  is a spanning tree of  $\mathcal{G}_c$  (it is acyclic and spans all nodes of  $\mathcal{G}_c$ ). All elementary circuits of  $\mathcal{T}_c$  are either of the form  $(v_i^{(1)}, v_i^{(2)}, v_i^{(1)})$  for some  $v_i \in \mathcal{V}$ , or of the form  $(v_i^{(1)}, v_j^{(1)}, v_i^{(1)})$  for some  $(v_i, v_j) \in$

\*See [MPT02] for the proof.

$\dagger$ See reference: M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness, 1979.

$\mathcal{E}_T$ , moreover  $\tau((v_i^{(1)}, v_i^{(2)}, v_i^{(1)})) = \frac{k+1+\text{degree}_{\mathcal{T}}(v_i)+1}{2} \leq k+1$  and  $\tau((v_i^{(1)}, v_j^{(1)}, v_i^{(1)})) = \frac{\text{degree}_{\mathcal{T}}(v_i)+1+\text{degree}_{\mathcal{T}}(v_j)+1}{2} \leq k+1$ . \* It follows that  $\tau(\mathcal{T}_c) \leq k+1 = \tau_0$ .

Inversely, suppose that  $\mathcal{G}_c$  has an MST  $\mathcal{T}_c$  having a cycle time at most  $\tau_0$ , and let  $\mathcal{T} = \Pi^{-1}(\mathcal{T}_c)$ , where  $\Pi^{-1}(\mathcal{T})$  is obtained by deleting all the vertices of the form  $v_i^{(2)}$  for  $v_i \in \mathcal{V}$ .  $\mathcal{T}$  is a spanning tree of  $\mathcal{G}$  (it contains all nodes of  $\mathcal{G}$  and is acyclic). We prove by contradiction that  $\text{degree}(\mathcal{T}) \leq k$ . Suppose that there exists a node  $v \in \mathcal{V}$  such that  $|\mathcal{N}_v^-(\mathcal{T})| > k$ , it follows that circuit  $\{v_i^{(1)}, v_i^{(2)}, v_i^{(1)}\}$  is a circuit of  $\mathcal{T}_c$ , and  $\tau((v_i^{(1)}, v_i^{(2)}, v_i^{(1)})) = \frac{k+1+|\mathcal{N}_v^-(\mathcal{T})|+1}{2} > k+1$ . It follows that  $\tau(\mathcal{T}) > k+1$ , thus  $k+1 < \tau_0 = k+1$  (contradiction).  $\square$

### C.1.5 Proof of Proposition 2.1.5

To prove Prop. 2.1.5, we start by proving the bottleneck of MBST of the particular-built undirected graph  $\mathcal{G}_c^{(u)}$  (lines 1-3 in Algo. 3) is smaller than or equal to the minimal cycle time of connectivity graph  $\mathcal{G}_c$ . The bottleneck of a tree  $\mathcal{T}$  is its maximum edge weight, denoted by  $B(\mathcal{T})$ .

Since we consider the node-capacitated case where  $A(i', j') = +\infty$  and  $C_{DN}(i) = \infty$ , the overlay  $\mathcal{G}_o$  has weights

$$d_o(i, j) = s \times T_c(i) + l(i, j) + \frac{M|\mathcal{N}_i^-|}{C_{UP}(i)}, \quad \forall (i, j) \in \mathcal{V}. \quad (\text{C.6})$$

Remind that the weights defined for the particular-built undirected graph  $\mathcal{G}_c^{(u)} = (\mathcal{V}, \mathcal{E}_c^{(u)})$  are

$$d^{(u)}(i, j) = [s \times (T_c(i) + T_c(j)) + l(i, j) + l(j, i) + \frac{M}{C_{UP}(i)} + \frac{M}{C_{UP}(j)}] / 2, \quad \forall (i, j) \in \mathcal{E}_c^{(u)}. \quad (\text{C.7})$$

**Lemma C.4.** *Consider the case where  $\mathcal{G}_c$  is node-capacitated with  $C_{DN}(j) = A(i', j') = \infty$  for all  $i, j \in \mathcal{V}$  and the overlay is required to be undirected. Let  $\tau^*(\mathcal{G}_c)$  be the cycle time of MCT on  $\mathcal{G}_c$  and  $\mathcal{T}_{BST}(\mathcal{G}_c^{(u)})$  be the MBST of  $\mathcal{G}_c^{(u)}$ . The bottleneck of  $\mathcal{T}_{BST}(\mathcal{G}_c^{(u)})$  is smaller than or equal to  $\tau^*(\mathcal{G}_c)$ , i.e.  $B(\mathcal{T}_{BST}(\mathcal{G}_c^{(u)})) \leq \tau^*(\mathcal{G}_c)$ .*

*Proof.* Denote  $\mathcal{T}^*(\mathcal{G}_c)$  the undirected overlay of  $\mathcal{G}_c$  with minimal cycle time. We consider the edge

$$(w, v) = \arg \max_{(i, j) \in \mathcal{E}(\mathcal{T}^*(\mathcal{G}_c))} d^{(u)}(i, j).$$

By definition,  $B(\mathcal{T}_{BST}(\mathcal{G}_c^{(u)})) = \min_{\mathcal{T} \in ST(\mathcal{G}_c^{(u)})} \max_{(i, j) \in \mathcal{E}(\mathcal{T})} d^{(u)}(i, j)$  where  $ST(\mathcal{G}_c^{(u)})$  is the set of spanning trees of  $\mathcal{G}_c^{(u)}$ . Since  $\mathcal{T}^*(\mathcal{G}_c) \in ST(\mathcal{G}_c^{(u)})$ , we have:

$$\begin{aligned} B(\mathcal{T}_{BST}(\mathcal{G}_c^{(u)})) &\leq d^{(u)}(w, v) \\ &\stackrel{(\text{C.7})}{=} \frac{s \times (T_c(w) + T_c(v)) + l(w, v) + l(v, w) + M/C_{UP}(w) + M/C_{UP}(v)}{2} \\ &\leq \frac{s \times (T_c(w) + T_c(v)) + l(w, v) + l(v, w) + |\mathcal{N}_w^-| M/C_{UP}(w) + |\mathcal{N}_v^-| M/C_{UP}(v)}{2} \\ &\stackrel{(\text{C.6})}{=} \frac{d_o(w, v) + d_o(v, w)}{2} \\ &\leq \tau^*(\mathcal{G}_c), \end{aligned}$$

---

\*Note that a circuit, like  $(v_i^{(1)}, v_i^{(2)}, v_i^{(1)})$ , is also a graph, and as such its cycle time  $\tau((v_i^{(1)}, v_i^{(2)}, v_i^{(1)}))$  is well defined.

where the second inequality follows from  $|\mathcal{N}_w^-|, |\mathcal{N}_v^-| \geq 1$ , and the last inequality comes from the definition of cycle time.  $\square$

**Lemma C.5.** *If  $\mathcal{G}_c$  is Euclidean, then  $\mathcal{G}_c^{(u)}$  is Euclidean.*

*Proof.* Remind that the connectivity graph  $\mathcal{G}_c$  is Euclidean on a node-capacitated network, if its delays  $d_c(i, j) = s \times T_c(i) + l(i, j)$  are symmetric ( $d_c(i, j) = d_c(j, i), \forall i, j \in \mathcal{V}$ ) and satisfy the triangle inequality. Consider three nodes  $i, j, k \in \mathcal{V}$ , we have:

$$\begin{aligned} d^{(u)}(i, j) &= \frac{d_c(i, j) + d_c(j, i) + M/C_{\text{UP}}(i) + M/C_{\text{UP}}(j)}{2} \\ &\leq \frac{d_c(i, k) + d_c(k, j) + d_c(j, k) + d_c(k, i) + M/C_{\text{UP}}(i) + M/C_{\text{UP}}(j)}{2} \\ &\leq \frac{d_c(i, k) + d_c(k, j) + d_c(j, k) + d_c(k, i) + M/C_{\text{UP}}(i) + M/C_{\text{UP}}(j) + 2M/C_{\text{UP}}(k)}{2} \\ &= d^{(u)}(i, k) + d^{(u)}(k, j), \end{aligned}$$

where the first inequality follows from the triangle inequality for  $d_c(i, j)$  and the second inequality from  $C_{\text{UP}}(k) \geq 0$ .  $\square$

**Proposition 2.1.5.** *Algorithm 3 is a 6-approximation algorithm for MCT when  $\mathcal{G}_c$  is node-capacitated and Euclidean with  $C_{\text{DN}}(j) = A(i', j') = \infty$  for all  $i, j \in \mathcal{V}$ , and  $\mathcal{G}_o$  is required to be undirected.*

*Proof.* Algorithm 3 considers, as a candidate solution, an opportune Hamiltonian path  $\mathcal{H}$  (line 8) for which reference [AR16, Thm. 8] proves that  $B(\mathcal{H}) \leq 3 \times B(\mathcal{T}_{\text{BST}}(\mathcal{G}_c^{(u)}))$  as  $\mathcal{G}_c^{(u)}$  is Euclidean (Lemma C.5). Moreover,

$$\begin{aligned} \tau(\mathcal{H}) &= \max_{(i,j) \in \mathcal{E}(\mathcal{H})} \frac{d_o(i, j) + d_o(j, i)}{2} \\ &= \max_{(i,j) \in \mathcal{E}(\mathcal{H})} \frac{s \times T_c(i) + s \times T_c(j) + l(i, j) + l(j, i) + \frac{M|\mathcal{N}_i^-|}{C_{\text{UP}}(i)} + \frac{M|\mathcal{N}_j^-|}{C_{\text{UP}}(j)}}{2} \\ &\leq \max_{(i,j) \in \mathcal{E}(\mathcal{H})} \frac{s \times T_c(i) + s \times T_c(j) + l(i, j) + l(j, i) + 2\frac{M}{C_{\text{UP}}(i)} + 2\frac{M}{C_{\text{UP}}(j)}}{2} \\ &\leq \max_{(i,j) \in \mathcal{E}(\mathcal{H})} s \times T_c(i) + s \times T_c(j) + l(i, j) + l(j, i) + \frac{M}{C_{\text{UP}}(i)} + \frac{M}{C_{\text{UP}}(j)} \\ &= 2 \max_{(i,j) \in \mathcal{E}(\mathcal{H})} d^{(u)}(i, j) \\ &= 2B(\mathcal{H}), \end{aligned}$$

where the first inequality follows from nodes in a path having degree at most 2. Combining these results with Lemma C.4, it follows that  $\tau(\mathcal{H}) \leq 6 \times \tau^*(\mathcal{G}_c)$ .  $\square$

### C.1.6 Proof of Proposition 2.1.6

**Proposition 2.1.6.** *Christofides' algorithm is a  $3N$ -approximation algorithm for MCT when  $\mathcal{G}_c$  is node-capacitated and Euclidean.*



Table 1: Sub-iNaturalist training over different networks. 1 Gbps core links capacities, 10 Gbps access links capacities. Five local computation steps.

Network name	Silos	Links	Cycle time (ms)					Ring's training speed-up	
			STAR	MATCHA <sup>(+)</sup>	MST	$\delta$ -MBST	RING	vs STAR	vs MATCHA <sup>(+)</sup>
Gaia [Hsi+17]	11	55	492.4	329.3(329.3)	239.7	239.8	219.7	1.79	1.50(1.50)
AWS NA [AWS20]	22	231	389.8	226.0(226.0)	191.3	191.3	182.9	1.40	1.24(1.24)
Géant [20a]	40	61	736.0	553.8(207.4)	202.6	202.6	210.6	3.49	2.63(2.96)
Exodus(us) [Mah+02]	79	147	1013.4	695.0(243.8)	246.9	246.9	205.5	3.95	2.25(1.18)
Ebone(eu) [Mah+02]	87	161	1003.2	681.6(224.9)	223.2	223.2	196.9	3.04	2.29(1.21)

Table 2: Sub-iNaturalist training over different networks. 1 Gbps core links capacities, 10 Gbps access links capacities. Ten local computation steps.

Network name	Silos	Links	Cycle time (ms)					Ring's training speed-up	
			STAR	MATCHA <sup>(+)</sup>	MST	$\delta$ -MBST	RING	vs STAR	vs MATCHA <sup>(+)</sup>
Gaia [Hsi+17]	11	55	619.4	456.4(456.4)	366.7	366.7	346.7	1.79	1.32(1.32)
AWS NA [AWS20]	22	231	516.8	353.2(353.2)	318.3	318.3	309.9	0.69	0.47(0.47)
Géant [20a]	40	61	609.0	680.8(334.7)	329.6	329.6	337.6	0.90	1.00(1.98)
Exodus(us) [Mah+02]	79	147	1140.4	822.0(370.9)	373.9	373.9	332.5	1.52	1.10(1.23)
Ebone(eu) [Mah+02]	87	161	1130.2	808.6(352.1)	350.4	350.4	323.9	1.74	1.25(1.09)

*Proof.* Let  $\mathcal{G}'_c$  be a weighted graph with the same topology as  $\mathcal{G}_c$  with weights  $d'(i, j) = s \times T_c(i) + l(i, j) + \frac{M}{\min(C_{UP}(i), C_{DN}(j), A(i', j'))}$ . Denote  $\hat{C}$  the output of Christofides' algorithm when used on  $\mathcal{G}'_c$ , and denote  $C^*$  the optimal tour of  $\mathcal{G}'_c$ . Since Christofides' algorithm provides a  $\frac{3}{2}$ -approximation to TSP, it follows that  $d'(\hat{C}) \leq \frac{3}{2}d'(C^*)$ . As  $\hat{C}$  and  $C^*$  are rings, it holds  $d'(\hat{C}) = d_o(\hat{C})$  and  $d'(C^*) = d_o(C^*)$ . Using Lemma C.3 it follows that

$$\tau(\hat{C}) = \frac{d_o(\hat{C})}{|\hat{C}|} = \frac{d'(\hat{C})}{|\hat{C}|} \leq \frac{3}{2} \frac{d'(C^*)}{|C^*|} = \frac{3}{2} \frac{d_o(C^*)}{|C^*|} = \frac{3}{2} \tau(C^*) \leq 3N\tau^*.$$

Thus the graph obtained using only the edges of  $\hat{C}$  is a  $3N$ -approximation algorithm for MCT when  $\mathcal{G}_c$  is node-capacitated and Euclidean.  $\square$

## C.2 Additional Experiments

### C.2.1 Similar tables of Table 2.5 for different local steps

Tables 1 and 2 show the effect of 6 different overlays when training ResNet-18 over sub-iNaturalist in networks with 1 Gbps core links and 10 Gbps access links and local steps equal to 5 and 10, respectively. For 5 local steps, the training time is evaluated as the time to reach a training accuracy equal to 65%, 55%, 60%, 45%, and 45% for Gaia, AWS North America, Géant, Exodus and Ebone, respectively. For 10 local steps, the training time is evaluated as the time to reach a training accuracy equal to 65%, 50%, 50%, 45% and 40%, respectively.

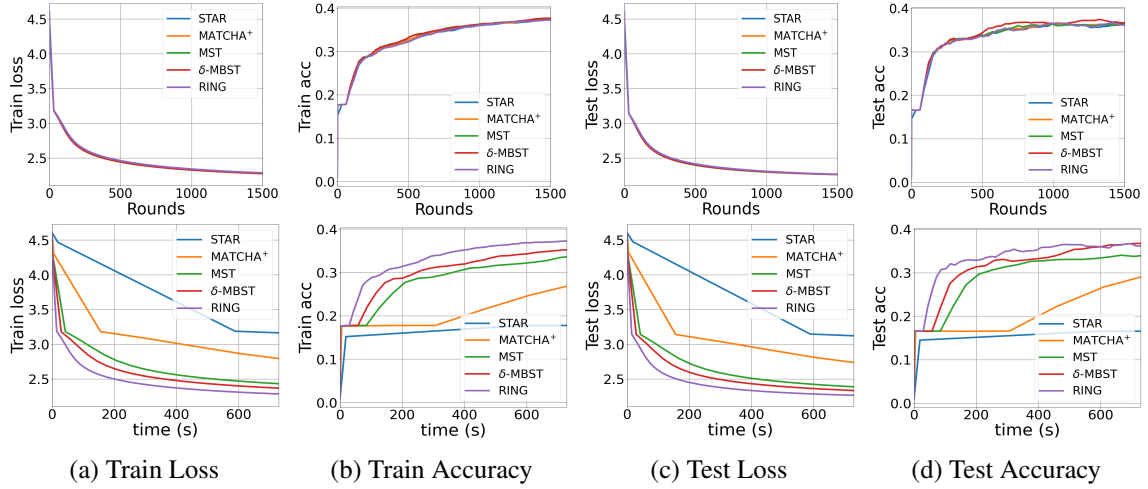


Figure C.1: Effect of overlays on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row) when training Shakespeare on AWS North America underlay. 1 Gbps core links capacities, 100 Mbps access links capacities,  $s = 1$ .

## C.2.2 Full results for training every dataset on AWS North America

In Figure 2.8, we have shown the training loss w.r.t. communication rounds and wall-clock time when training four different datasets on AWS North America. Here we give the complete results (Figures C.1-C.4) which include training loss, training accuracy, test loss, and test accuracy w.r.t. communication rounds and wall-clock time.

## C.2.3 Additional experiments

In our experiments, we considered 5 underlays, for which we compared 6 different overlays (e.g., Table 2.5). Moreover, we tested 4 different datasets (e.g., Fig. 2.8) and 3 different values for the number of local steps  $s = 1, 5, 10$  (e.g., Tables 1 and 2). We were not able to run experiments for all 360 possible combinations. Here, we show some representative additional results. For each experimental result, four metrics are shown including the train loss, train accuracy, test loss, and test accuracy w.r.t. communication rounds and wall-clock time. The common observation is that the overlay Ring converges faster than MATCHA<sup>+</sup> and STAR in terms of wall-clock time. In some cases, the test loss and accuracy of the model learned by the RING start becoming worse after some time, with overfitting being a possible explanation in some cases (see Figs. C.5, C.7, C.10 and C.12).

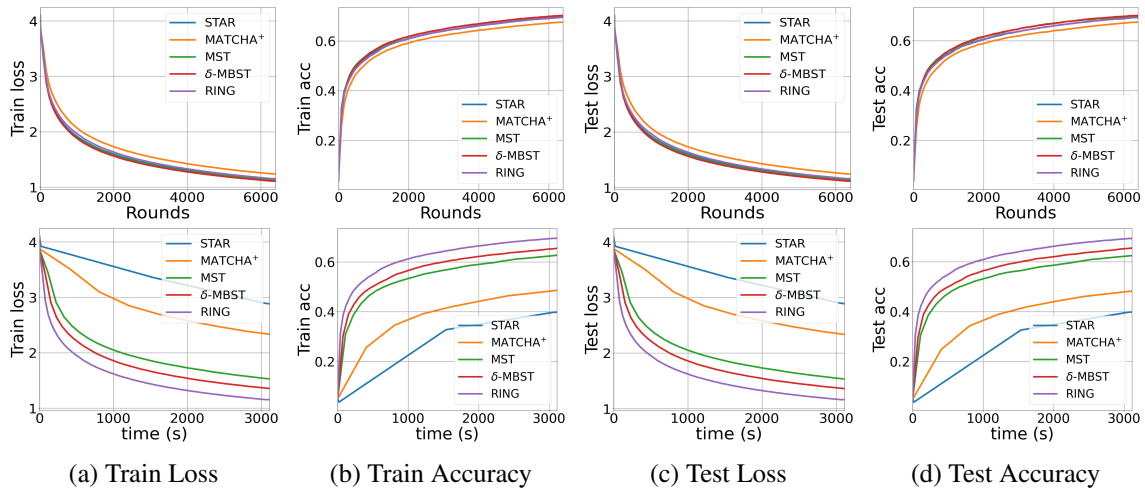


Figure C.2: Effect of overlays on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row) when training FEMNIST on AWS North America underlay. 1 Gbps core links capacities, 100 Mbps access links capacities,  $s = 1$ .

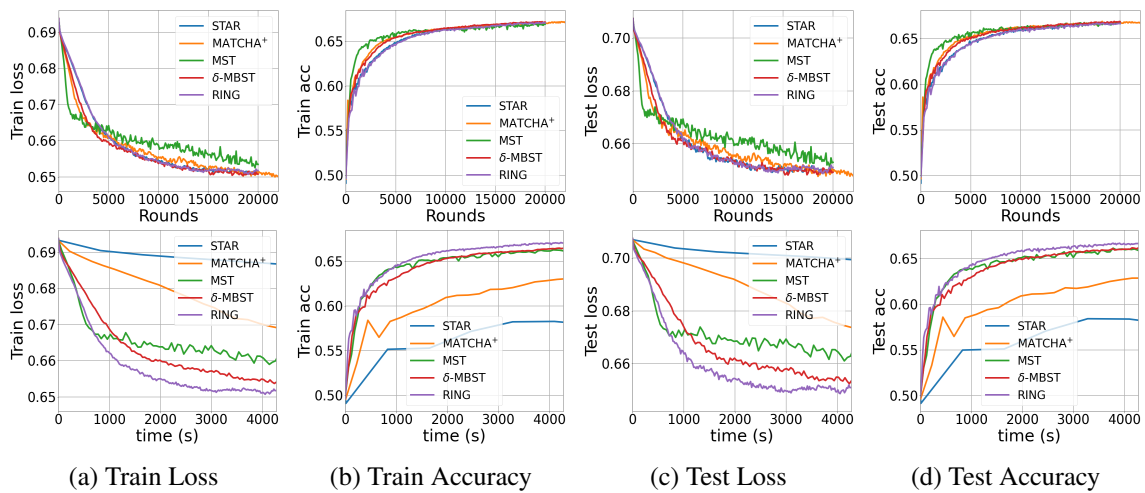


Figure C.3: Effect of overlays on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row) when training Sentiment140 on AWS North America underlay. 1 Gbps core links capacities, 100 Mbps access links capacities,  $s = 1$ .

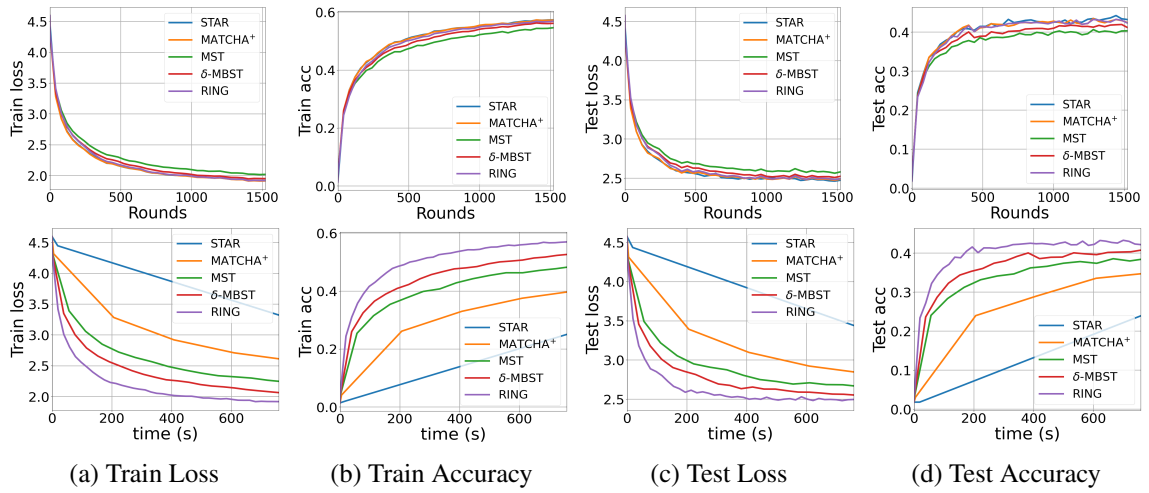


Figure C.4: Effect of overlays on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row) when training iNaturalist on AWS North America underlay. 1 Gbps core links capacities, 100 Mbps access links capacities,  $s = 1$ .

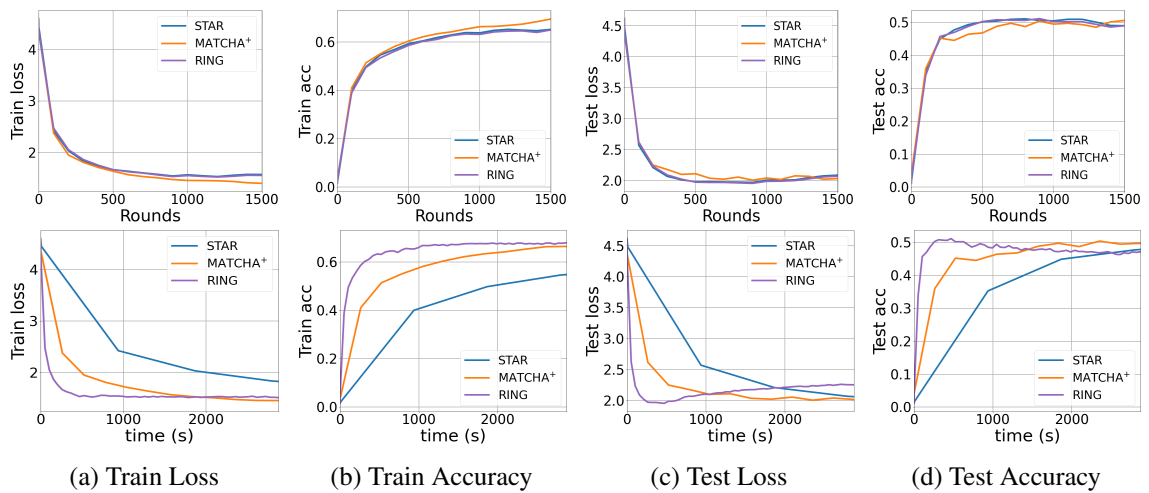


Figure C.5: Effect of overlays on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row) when training ResNet-18 image classification model using iNaturalist on Gaia underlay. 1 Gbps core links capacities, 100 Mbps access links capacities,  $s = 1$ .

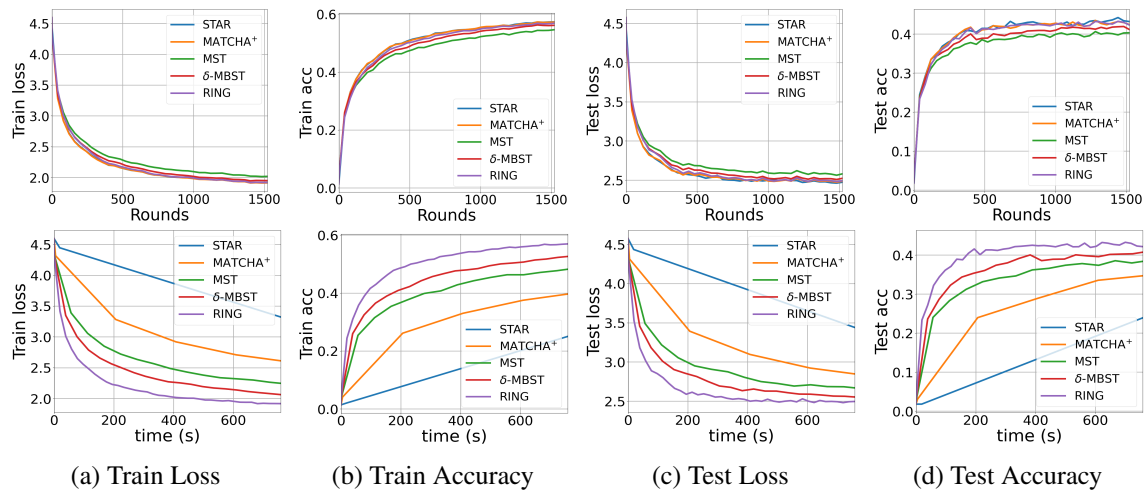


Figure C.6: Effect of overlays on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row) when training ResNet-18 image classification model using iNaturalist on AWS North America underlay. 1 Gbps core links capacities, 100 Mbps access links capacities,  $s = 1$ .

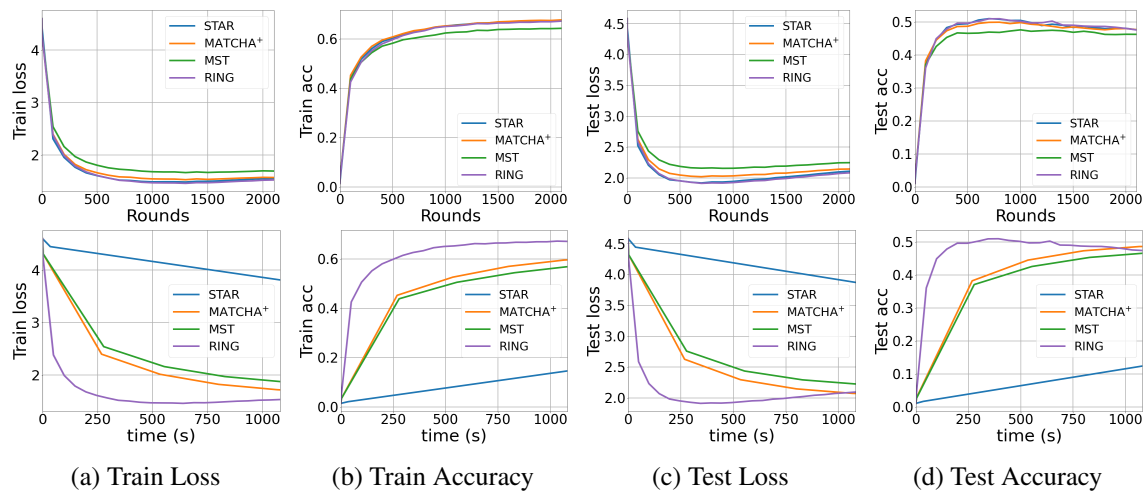


Figure C.7: Effect of overlays on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row) when training ResNet-18 image classification model using iNaturalist on Géant underlay. 1 Gbps core links capacities, 100 Mbps access links capacities,  $s = 1$ .

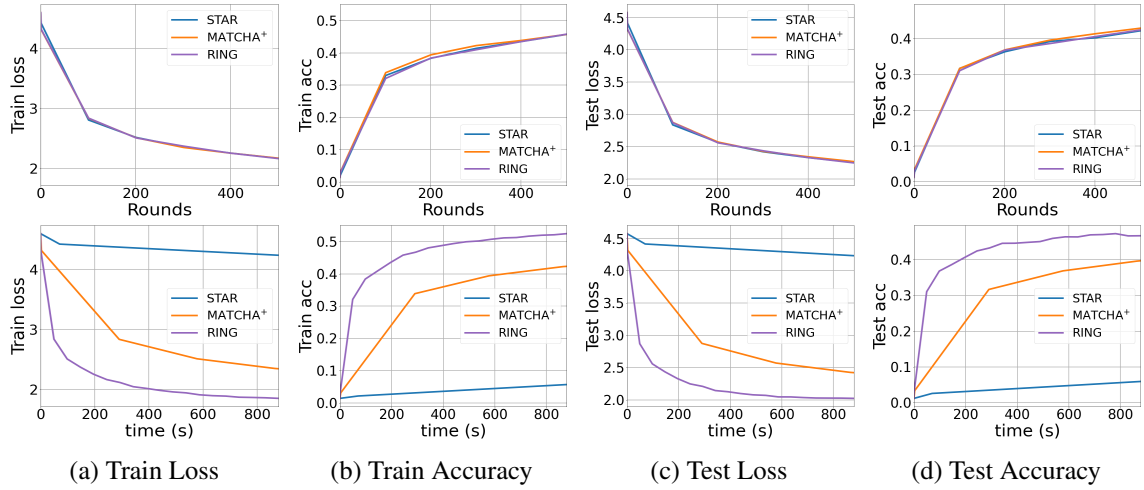


Figure C.8: Effect of overlays on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row) when training ResNet-18 image classification model using iNaturalist on Exodus underlay. 1 Gbps core links capacities, 100 Mbps access links capacities,  $s = 1$ .

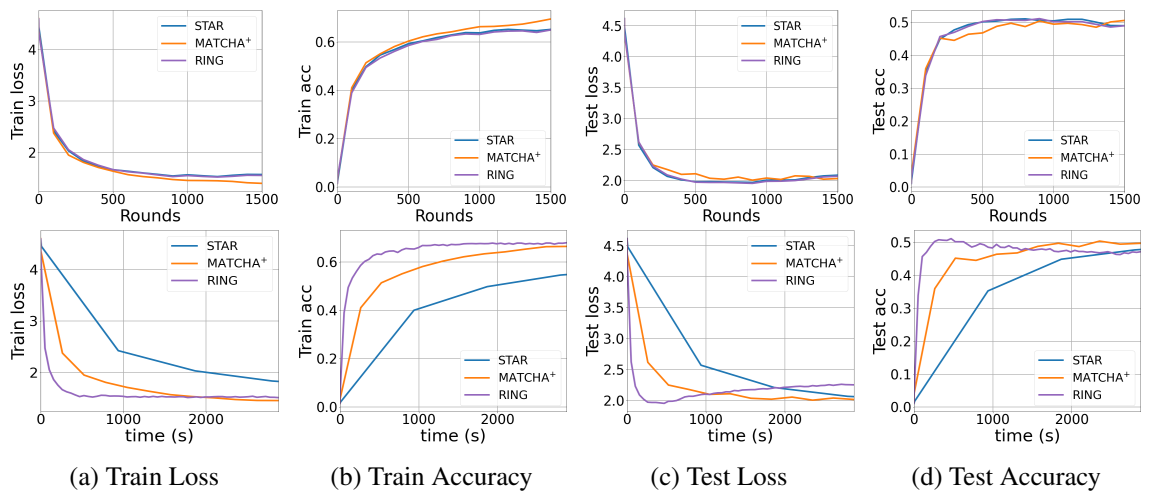


Figure C.9: Effect of overlays on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row) when training ResNet-18 image classification model using iNaturalist on Ebone underlay. 1 Gbps core links capacities, 100 Mbps access links capacities,  $s = 1$ .

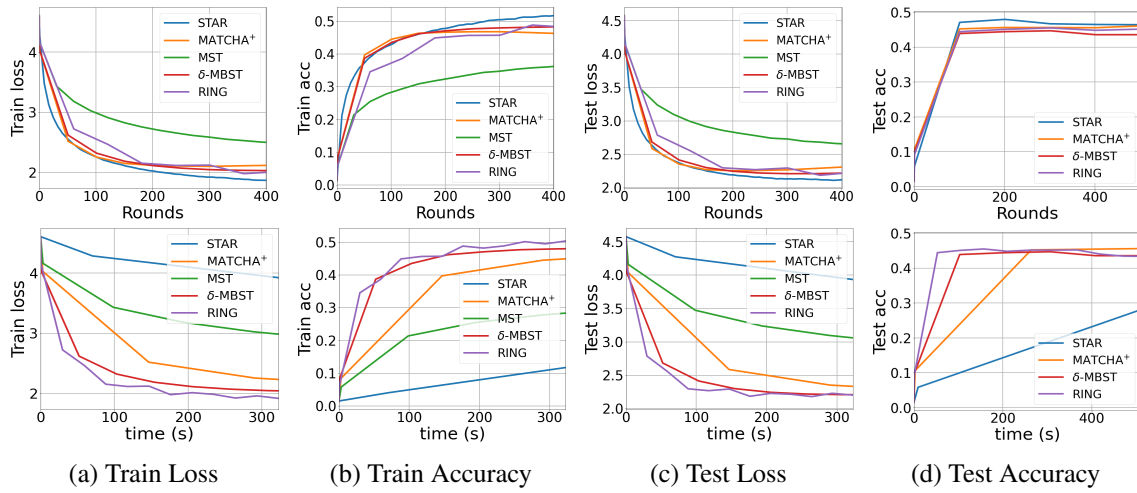


Figure C.10: Effect of overlays on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row) when training ResNet-18 image classification model using iNaturalist on Gaia underlay. 1 Gbps core links capacities, 100 Mbps access links capacities,  $s = 5$ .

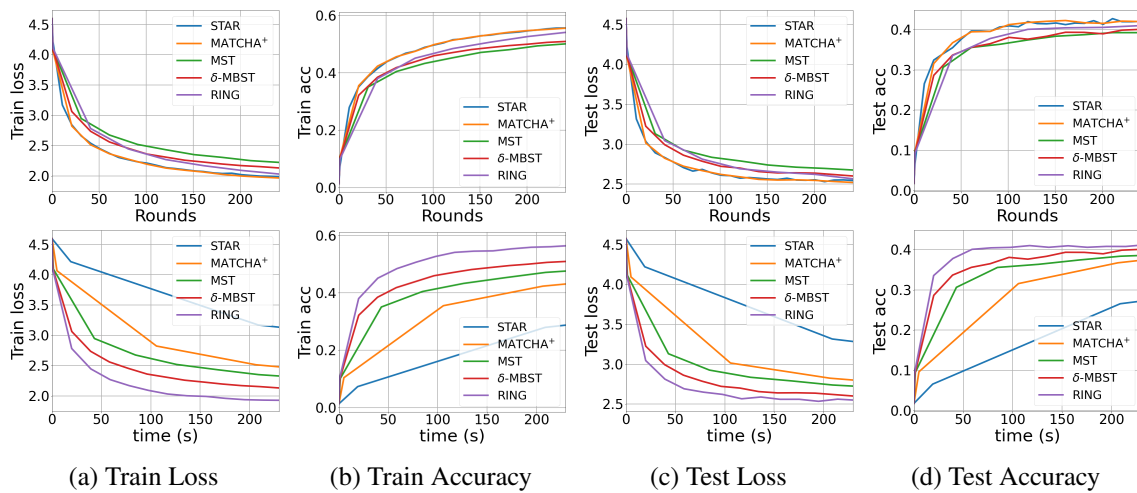


Figure C.11: Effect of overlays on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row) when training ResNet-18 image classification model using iNaturalist on AWS North America underlay. 1 Gbps core links capacities, 100 Mbps access links capacities,  $s = 5$ .

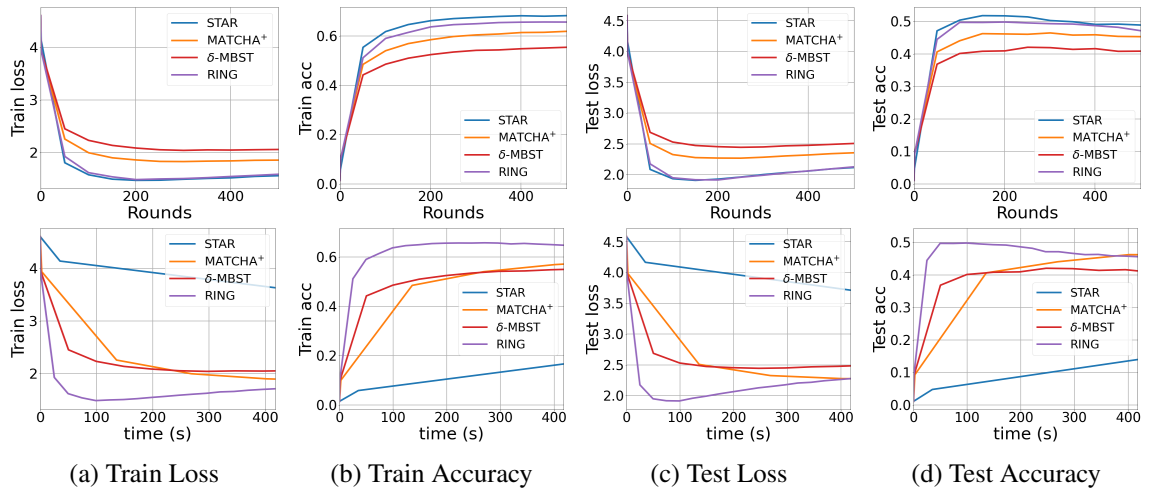


Figure C.12: Effect of overlays on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row) when training ResNet-18 image classification model using iNaturalist on Géant underlay. 1 Gbps core links capacities, 100 Mbps access links capacities,  $s = 5$ .

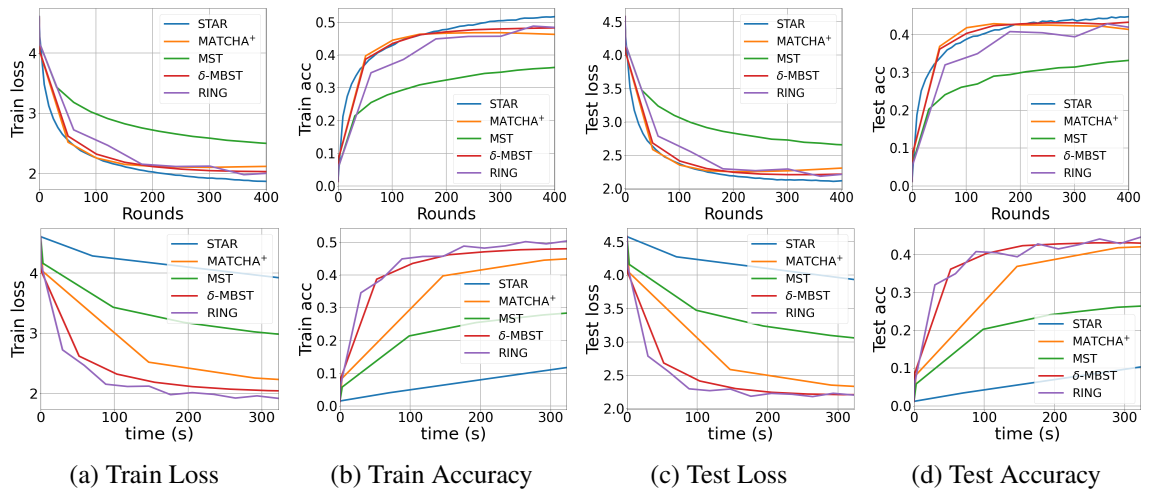


Figure C.13: Effect of overlays on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row) when training ResNet-18 image classification model using iNaturalist on Exodus underlay. 1 Gbps core links capacities, 100 Mbps access links capacities,  $s = 5$ .



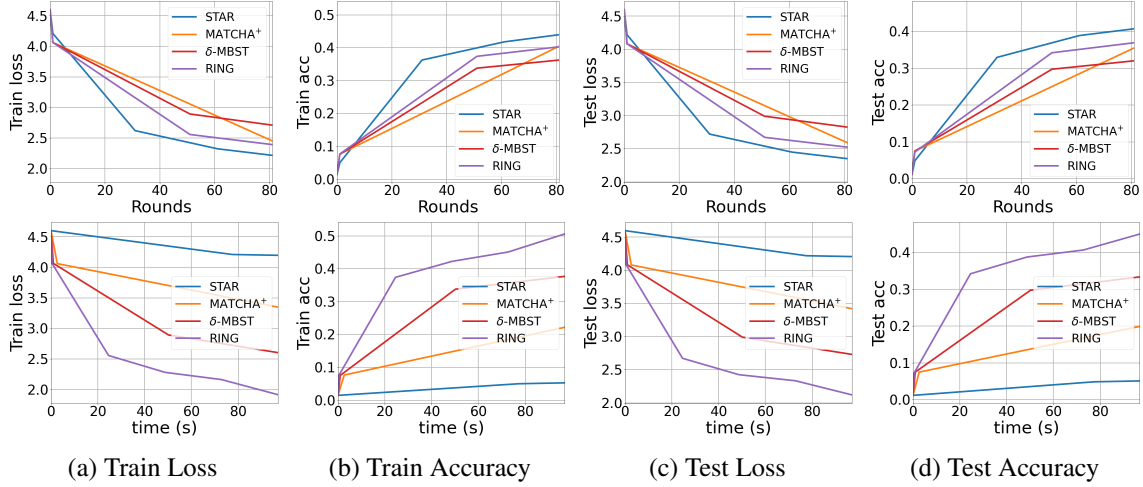


Figure C.14: Effect of overlays on the convergence w.r.t. communication rounds (top row) and wall-clock time (bottom row) when training ResNet-18 image classification model using iNaturalist on Ebone underlay. 1 Gbps core links capacities, 100 Mbps access links capacities,  $s = 5$ .

## D Federated Learning under Heterogeneous and Correlated Client Availability

### D.1 Proof of Theorem 2.2.2

**Theorem D.1** (Decomposing the total error). *Let  $\kappa := L/\mu$ . Under Assumptions 4–6, the optimization error of the target global objective  $\epsilon = F(\mathbf{w}) - F^*$  can be bounded as follows:*

$$\epsilon \leq 2\kappa^2 \underbrace{(F_B(\mathbf{w}) - F_B^*)}_{:=\epsilon_{opt}} + \underbrace{F(\mathbf{w}_B^*) - F^*}_{:=\epsilon_{bias}}. \quad (2.23)$$

Moreover, let  $\chi_{\alpha\|p}^2 := \sum_{k=1}^N (\alpha_k - p_k)^2 / p_k$ . Then:

$$\epsilon_{bias} \leq \kappa^2 \cdot \underbrace{\chi_{\alpha\|p}^2}_{:=\epsilon_{bias}} \cdot \Gamma. \quad (2.24)$$

The proof of Theorem D.1 employs well-established techniques from convex optimization. It is based on the proof presented in [Wan+20b, Theorem 2].

*Proof of Theorem D.1.* By leveraging the  $L$ -smoothness and  $\mu$ -strong convexity properties of  $F$ , we obtain:

$$F(\mathbf{w}) - F^* \leq \frac{1}{2\mu} \|\nabla F(\mathbf{w})\|^2 \quad (D.8)$$

$$\leq \frac{L^2}{2\mu} \|\mathbf{w} - \mathbf{w}^*\|^2 \quad (D.9)$$

$$\leq \frac{L^2}{\mu} (\|\mathbf{w} - \mathbf{w}_B^*\|^2 + \|\mathbf{w}_B^* - \mathbf{w}^*\|^2) \quad (D.10)$$

$$\leq \frac{2L^2}{\mu^2} \left( \underbrace{F_B(\mathbf{w}) - F_B^*}_{:=\epsilon_{\text{opt}}} + \underbrace{F(\mathbf{w}_B^*) - F^*}_{:=\epsilon_{\text{bias}}} \right), \quad (\text{D.11})$$

where the inequality in (D.8) follows from Assumption 6 and is commonly referred to as the *Polyak-Lojasiewicz inequality*; the inequality in (D.9) is derived using the fact that  $\nabla F(\mathbf{w}^*) = 0$  (Assumption 4) and the definition of  $L$ -Lipschitz continuous gradient for  $F$  (Assumption 5); the inequality in (D.10) is based on  $(a + b)^2 \leq 2(a^2 + b^2)$ ; lastly, the inequality in (D.11) follows from the  $\mu$ -strong convexity of both  $F_B$  and  $F$  (Assumptions 6), and uses  $\nabla F_B(\mathbf{w}_B^*) = 0$  and  $\nabla F(\mathbf{w}^*) = 0$  (Assumption 4). The obtained results complete the first part of the proof, establishing the bound in (2.23).

Next, to prove the relation in (2.24), we proceed by bounding the term  $\epsilon_{\text{bias}}$  as follows:

$$\epsilon_{\text{bias}} := (F(\mathbf{w}_B^*) - F^*) \leq \frac{1}{2\mu} \|\nabla F(\mathbf{w}_B^*)\|^2, \quad (\text{D.12})$$

where the inequality in (D.12) directly follows from the Polyak-Lojasiewicz inequality (Assumption 6).

Furthermore, we bound the term  $\|\nabla F(\mathbf{w}_B^*)\|$  as follows:

$$\|\nabla F(\mathbf{w}_B^*)\| = \left\| \sum_{k=1}^N (\alpha_k - p_k) \nabla F_k(\mathbf{w}_B^*) \right\| \quad (\text{D.13})$$

$$\leq \sum_{k=1}^N |\alpha_k - p_k| \|\nabla F_k(\mathbf{w}_B^*)\| \quad (\text{D.14})$$

$$\leq L \sum_{k=1}^N |\alpha_k - p_k| \|\mathbf{w}_B^* - \mathbf{w}_k^*\| \quad (\text{D.15})$$

$$\leq L \sqrt{\frac{2}{\mu}} \sum_{k=1}^N |\alpha_k - p_k| \sqrt{(F_k(\mathbf{w}_B^*) - F_k^*)}, \quad (\text{D.16})$$

where, in (D.13), we use  $\nabla F_B(\mathbf{w}_B^*) = 0$  (Assumption 4) and apply the definitions of  $F$  and  $F_B$  given in (2.14) and (2.17), respectively. The bound in (D.14) follows from the triangle inequality. Next, the inequality in (D.15) uses  $\nabla F_k(\mathbf{w}_k^*) = 0$  (Assumption 4) and the  $L$ -smoothness of  $F_k$  (Assumption 5). Finally, the inequality in (D.16) leverages the  $\mu$ -strong convexity of  $F_k$  (Assumption 6) and  $\nabla F_k(\mathbf{w}_k^*) = 0$  (Assumption 4), and follows multiplying and dividing by  $\sqrt{p_k}$ .

By squaring both sides of Equation (D.16), we obtain:

$$\|\nabla F(\mathbf{w}_B^*)\|^2 \leq \frac{2L^2}{\mu} \left( \sum_{k=1}^N \frac{|\alpha_k - p_k|}{\sqrt{p_k}} \sqrt{p_k (F_k(\mathbf{w}_B^*) - F_k^*)} \right)^2 \quad (\text{D.17})$$

$$\leq \frac{2L^2}{\mu} \left( \sum_{k=1}^N \frac{(\alpha_k - p_k)^2}{p_k} \right) \left( \sum_{k=1}^N p_k (F_k(\mathbf{w}_B^*) - F_k^*) \right) \quad (\text{D.18})$$

$$\leq \frac{2L^2}{\mu} \cdot \chi_{\alpha\|p}^2 \cdot \Gamma, \quad (\text{D.19})$$

where the inequality in (D.18) follows from the Cauchy-Schwarz inequality. Furthermore, the inequality in (D.19) holds because:

$$\sum_{k=1}^N p_k (F_k(\mathbf{w}_B^*) - F_k^*) = F_B^* - \sum_{k=1}^N p_k F_k^* \quad (\text{D.20})$$

$$\leq F_B(\mathbf{w}^*) - \sum_{k=1}^N p_k F_k^* \quad (\text{D.21})$$

$$= \sum_{k=1}^N p_k (F_k(\mathbf{w}^*) - F_k^*) \quad (\text{D.22})$$

$$\leq \max_{k \in \mathcal{K}} \{F_k(\mathbf{w}^*) - F_k^*\} := \Gamma. \quad (\text{D.23})$$

We remark that the inequality in (D.21) only holds if  $\mathbf{w}_B^*$  is the global minimizer of  $F_B$ , as guaranteed by Assumption 4. By replacing (D.19) into (D.12), we have:

$$\epsilon_{\text{bias}} \leq \frac{1}{2\mu} \|\nabla F(\mathbf{w}_B^*)\|^2 \leq \frac{L^2}{\mu^2} \cdot \chi_{\alpha\|p}^2 \cdot \Gamma, \quad (\text{D.24})$$

which concludes the proof of Equation (2.24), and therefore, of Theorem D.1.  $\square$

## D.2 Proof of Theorem 2.2.3

### D.2.1 Algorithm Overview and Supplementary Notation

Let  $\mathbf{w}_{t,j}^k$  represent the model parameter maintained by the  $k$ -th client during the  $t$ -th global communication round and the  $j$ -th local step. The  $t$ -th global communication round can be described as follows: 1) The server broadcasts the model parameter  $\mathbf{w}_{t,0}$  to the active clients, which adopt it as their local model, i.e.,  $\mathbf{w}_{t,0}^k = \mathbf{w}_{t,0}$  for  $k \in \mathcal{A}_t$ ; 2) Each active client  $k \in \mathcal{A}_t$  generates a sequence of local models  $\{\mathbf{w}_{t,j}^k\}_{j=1}^E$  using the local-SGD update rule defined in (2.15); 3) The active clients send their model updates  $\Delta_t^k := \mathbf{w}_{t,E}^k - \mathbf{w}_{t,0}$  back to the server; 4) The server aggregates the model updates using the aggregation rule specified in (2.16), resulting in the new global model parameter  $\mathbf{w}_{t+1,0}$ .

$$\mathbf{w}_{t,j+1}^k = \mathbf{w}_{t,j}^k - \eta_t \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) \text{ for } j = 0, \dots, E-1; \quad (\text{2.15})$$

$$\mathbf{w}_{t+1,0} = \Pi_W \mathbf{w}_{t,0} + \sum_{k \in \mathcal{A}_t} q_k (\mathbf{w}_{t,E}^k - \mathbf{w}_{t,0}) \text{ for } j = E. \quad (\text{2.16})$$

The projection operator in (2.16) ensures that the current iterate  $\mathbf{w}_{t+1,0}$  in the optimization algorithm defined by (2.15) and (2.16) remains within the feasible region  $W$ .

**Sources of randomness.** In the system, we model two sources of randomness. The first arises from the availability of random clients, which follows a Markov process as stated in Assumption 3. The second source of randomness originates from the random sampling of batches for computing stochastic gradients. Remember that  $\mathcal{A}_t$  denotes the random set of clients available at the  $t$ -th communication round and that  $\mathcal{B}_{t,j}^k$  denotes the random batch independently sampled from client- $k$ 's local dataset at round  $t$ , local iteration  $j$ . For the analysis, we introduce the following additional notation:

- $\mathcal{A}_{i:j} := \{\mathcal{A}_i, \dots, \mathcal{A}_j\}$ : the family of random sets of clients available from the  $i$ -th to the  $j$ -th communication rounds,  $i < j$ ;

- $\mathcal{B}_t^k := \{\mathcal{B}_{t,j}^k\}_{j=0}^{E-1}$ : the set of random batches sampled by the  $k$ -th client at the  $t$ -th communication round;
- $\mathcal{B}_t := \{\mathcal{B}_t^k\}_{k \in \mathcal{A}_t}$ : the set of random batches sampled by the available clients ( $\mathcal{A}_t$ ) in the  $t$ -th communication round;
- $\mathcal{B}_{t,i:j}^k := \{\mathcal{B}_{t,i}^k, \dots, \mathcal{B}_{t,j}^k\}$ : the set of random batches sampled by the  $k$ -th client at the  $t$ -th communication round between the  $i$ -th and the  $j$ -th local iterations,  $i < j$ ;
- $\mathcal{B}_{i:j} := \{\mathcal{B}_i, \dots, \mathcal{B}_j\}$ : the set of random batches sampled by the available clients ( $\mathcal{A}_{i:j}$ ) between the  $i$ -th and  $j$ -th communication rounds,  $i < j$ .

With this notation established, the randomness in the  $t$ -th communication round, which starts with the initial model  $\mathbf{w}_{t,0}$  and yields the updated model  $\mathbf{w}_{t+1,0}$ , is fully determined by the sets  $\mathcal{A}_t$  and  $\mathcal{B}_t$ . This implies that the evolution of the algorithm, governed by the update rules in (2.15) and (2.16), from round 0 to round  $t$  can be completely described by the tuple:

$$\mathcal{H}_t := (\mathcal{A}_0, \dots, \mathcal{A}_{t-1}; \mathcal{B}_0, \dots, \mathcal{B}_{t-1}), \quad (\text{D.25})$$

which represents the historical information up to the  $t$ -th communication round.

We introduce the following additional quantities for our analysis:

$$\mathbf{g}_t(\mathcal{A}_t, \mathcal{B}_t) := \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k), \quad (\text{D.26})$$

and

$$\bar{\mathbf{g}}_t(\mathcal{A}_t, \mathcal{B}_t) := \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \nabla F_k(\mathbf{w}_{t,j}^k), \quad (\text{D.27})$$

where  $\mathbf{g}_t(\mathcal{A}_t, \mathcal{B}_t)$  denotes the global pseudo-gradient computed at communication round  $t$ , aggregated from the active clients in  $\mathcal{A}_t$ , and  $\bar{\mathbf{g}}_t(\mathcal{A}_t, \mathcal{B}_t)$  denotes its expected value with respect to the choices of the random batches  $\mathcal{B}_{t,j}^k$ , for all  $j = 0, \dots, E-1$  and  $k \in \mathcal{A}_t$ . With this notation established, the global update rule for the  $t$ -th communication round can be expressed as:

$$\mathbf{w}_{t+1,0} = \Pi W \mathbf{w}_{t,0} - \eta_t \mathbf{g}_t(\mathcal{A}_t, \mathcal{B}_t). \quad (\text{D.28})$$

## D.2.2 Supporting Lemmas

In this section, we introduce several lemmas that are instrumental in proving Theorem D.20. Firstly, we prove Lemma 2.2.1, introduced in Section 2.2.3.1. Its proof relies on the convexity and compactness of the hypothesis class  $W$  (Assumption 4), on the  $L$ -smoothness of the functions  $\{F_k\}_{k \in \mathcal{K}}$  (Assumption 5), and on the bounded variance of the stochastic gradients (Assumption 7).

**Lemma D.2.** *Under Assumptions 4, 5, and 7, there exist constants  $D$ ,  $G$ , and  $H > 0$ , such that, for  $\mathbf{w} \in W$  and  $k \in \mathcal{K}$ , we have:*

$$\|\nabla F_k(\mathbf{w})\| \leq D, \quad (\text{2.19})$$

$$\mathbb{E} \|\nabla F_k(\mathbf{w}, \xi)\|^2 \leq G^2, \quad (\text{2.20})$$

$$|F_k(\mathbf{w}) - F_k(\mathbf{w}_B^*)| \leq H. \quad (\text{2.21})$$

*Proof of Lemma D.2.* The boundedness of the hypothesis class  $W$  (Assumption 4) provides a bound on the sequence  $(\mathbf{w}_{t,0})_{t \geq 0}$  generated by the scheme defined in Equations (2.15) and (2.16). Moreover, since  $\mathbf{w}_k^*$  minimizes  $\nabla F_k(\mathbf{w})$ , we have  $\nabla F_k(\mathbf{w}_k^*) = 0$ . Furthermore, the  $L$ -smoothness of  $\{F_k\}_{k \in \mathcal{K}}$  (Assumption 5) leads to the following inequality:

$$\|\nabla F_k(\mathbf{w})\| = \|\nabla F_k(\mathbf{w}) - \nabla F_k(\mathbf{w}_k^*)\| \leq L \|\mathbf{w} - \mathbf{w}_k^*\| := D < +\infty. \quad (\text{D.29})$$

The bound in (2.19) is directly derived from (D.29), while the bound in (2.21) follows from the continuity of  $\{F_k\}_{k \in \mathcal{K}}$  over the compact set  $W$  (Assumption 4). Finally, the inequality in (2.20) requires a bound on the variance of the stochastic gradients (Assumption 7). In particular, it holds that:

$$\mathbb{E}\|\nabla F_k(\mathbf{w}, \xi)\|^2 \leq D^2 + \max_{k \in \mathcal{K}} \{\sigma_k^2\} := G^2. \quad (\text{D.30})$$

□

The following lemma proves that the global pseudo-gradient  $\mathbf{g}_t(\mathcal{A}_t, \mathcal{B}_t)$  is an unbiased estimator of  $\bar{\mathbf{g}}_t(\mathcal{A}_t, \mathcal{B}_t)$ . A similar result has been used in previous works, specifically in [Wan+20b, Appendix C1]. Here, we provide a comprehensive proof for this result.

**Lemma D.3.** *Let  $\mathbf{g}_t(\mathcal{A}_t, \mathcal{B}_t)$  and  $\bar{\mathbf{g}}_t(\mathcal{A}_t, \mathcal{B}_t)$  be defined as in (D.26) and (D.27), respectively. The following equality holds:*

$$\mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} [\mathbf{g}_t(\mathcal{A}_t, \mathcal{B}_t)] = \mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} [\bar{\mathbf{g}}_t(\mathcal{A}_t, \mathcal{B}_t)]. \quad (\text{D.31})$$

*Proof of Lemma D.3.*

$$\mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} [\mathbf{g}_t(\mathcal{A}_t, \mathcal{B}_t)] = \quad (\text{D.32})$$

$$= \mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) \right] \quad (\text{D.33})$$

$$= \sum_{k \in \mathcal{A}_t} q_k \mathbb{E}_{\mathcal{B}_t^k} \left[ \sum_{j=0}^{E-1} \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) \right] \quad (\text{D.34})$$

$$= \sum_{k \in \mathcal{A}_t} q_k \left[ \mathbb{E}_{\mathcal{B}_{t,0}^k} [\nabla F_k(\mathbf{w}_{t,0}, \mathcal{B}_{t,0}^k)] + \mathbb{E}_{\mathcal{B}_{t,0}^k, \mathcal{B}_{t,1}^k} [\nabla F_k(\mathbf{w}_{t,1}, \mathcal{B}_{t,1}^k)] + \dots \right. \\ \left. + \mathbb{E}_{\mathcal{B}_{t,0}^k, E-1} [\nabla F_k(\mathbf{w}_{t,E-1}, \mathcal{B}_{t,E-1}^k)] \right] \quad (\text{D.35})$$

$$= \sum_{k \in \mathcal{A}_t} q_k \left[ \nabla F_k(\mathbf{w}_{t,0}) + \mathbb{E}_{\mathcal{B}_{t,0}^k} \left[ \mathbb{E}_{\mathcal{B}_{t,1}^k | \mathcal{B}_{t,0}^k} [\nabla F_k(\mathbf{w}_{t,1}, \mathcal{B}_{t,1}^k)] \right] + \dots \right. \\ \left. + \mathbb{E}_{\mathcal{B}_{t,0}^k, E-2} \left[ \mathbb{E}_{\mathcal{B}_{t,E-1}^k | \mathcal{B}_{t,0}^k, E-2} [\nabla F_k(\mathbf{w}_{t,E-1}, \mathcal{B}_{t,E-1}^k)] \right] \right] \quad (\text{D.36})$$

$$= \sum_{k \in \mathcal{A}_t} q_k \left[ \nabla F_k(\mathbf{w}_{t,0}) + \mathbb{E}_{\mathcal{B}_{t,0}^k} [\nabla F_k(\mathbf{w}_{t,1}^k)] + \dots + \mathbb{E}_{\mathcal{B}_{t,0}^k, E-2} [\nabla F_k(\mathbf{w}_{t,E-1}^k)] \right] \quad (\text{D.37})$$

$$= \sum_{k \in \mathcal{A}_t} q_k \mathbb{E}_{\mathcal{B}_{t,0}^k, E-2} \left[ \sum_{j=0}^{E-1} \nabla F_k(\mathbf{w}_{t,j}^k) \right] \quad (\text{D.38})$$

$$= \mathbb{E}_{\mathcal{B}_t|\mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \nabla F_k(\mathbf{w}_{t,j}^k) \right] = \mathbb{E}_{\mathcal{B}_t|\mathcal{A}_t, \mathcal{H}_t} [\bar{\mathbf{g}}_t(\mathcal{A}_t, \mathcal{B}_t)], \quad (\text{D.39})$$

where, in (D.34), we considered that both the evolution of the local models  $\{\mathbf{w}_{t,j}^k\}_{j=0}^{E-1}$  and the choices of the random batches  $\{\mathcal{B}_{t,j}^k\}_{j=0}^{E-1}$  are independent among different clients  $k \in \mathcal{A}_t$  within the same communication round  $t \in \mathcal{T}$ .  $\square$

For the sake of simplicity, we will henceforth denote  $\mathbf{g}_t(\mathcal{A}_t, \mathcal{B}_t)$  and  $\bar{\mathbf{g}}_t(\mathcal{A}_t, \mathcal{B}_t)$  as  $\mathbf{g}_t$  and  $\bar{\mathbf{g}}_t$ , respectively. The following lemma decomposes the optimization error into multiple components, which we will bound separately in subsequent lemmas.

**Lemma D.4** (Decomposition of the error in a global communication round). *Let Assumption 4 hold. We have:*

$$\begin{aligned} \mathbb{E}_{\mathcal{B}_t|\mathcal{A}_t, \mathcal{H}_t} \|\mathbf{w}_{t+1,0} - \mathbf{w}_B^*\|^2 &\leq \|\mathbf{w}_{t,0} - \mathbf{w}_B^*\|^2 \underbrace{- 2\eta_t \mathbb{E}_{\mathcal{B}_t|\mathcal{A}_t, \mathcal{H}_t} \langle \mathbf{w}_{t,0} - \mathbf{w}_B^*, \bar{\mathbf{g}}_t \rangle}_{\text{bounded in Lemma D.5}} + \underbrace{\eta_t^2 \mathbb{E}_{\mathcal{B}_t|\mathcal{A}_t, \mathcal{H}_t} \|\bar{\mathbf{g}}_t\|^2}_{\text{bounded in Lemma D.6}} \\ &+ \underbrace{2\eta_t \mathbb{E}_{\mathcal{B}_t|\mathcal{A}_t, \mathcal{H}_t} \langle \mathbf{w}_{t,0} - \mathbf{w}_B^* - \eta_t \bar{\mathbf{g}}_t, \bar{\mathbf{g}}_t - \mathbf{g}_t \rangle}_{\text{bounded in Lemma D.7}} + \underbrace{\eta_t^2 \mathbb{E}_{\mathcal{B}_t|\mathcal{A}_t, \mathcal{H}_t} \|\mathbf{g}_t - \bar{\mathbf{g}}_t\|^2}_{\text{bounded in Lemma D.8}}. \end{aligned} \quad (\text{D.40})$$

*Proof of Lemma D.4.*

$$\|\mathbf{w}_{t+1,0} - \mathbf{w}_B^*\|^2 = \|\Pi W \mathbf{w}_{t,0} - \eta_t \mathbf{g}_t - \Pi W \mathbf{w}_B^*\|^2 \quad (\text{D.41})$$

$$\leq \|\mathbf{w}_{t,0} - \eta_t \mathbf{g}_t - \mathbf{w}_B^* + \eta_t \bar{\mathbf{g}}_t - \eta_t \bar{\mathbf{g}}_t\|^2 \quad (\text{D.42})$$

$$= \|\mathbf{w}_{t,0} - \mathbf{w}_B^* - \eta_t \bar{\mathbf{g}}_t\|^2 + 2\eta_t \langle \mathbf{w}_{t,0} - \mathbf{w}_B^* - \eta_t \bar{\mathbf{g}}_t, \bar{\mathbf{g}}_t - \mathbf{g}_t \rangle + \eta_t^2 \|\mathbf{g}_t - \bar{\mathbf{g}}_t\|^2 \quad (\text{D.43})$$

$$\begin{aligned} &= \|\mathbf{w}_{t,0} - \mathbf{w}_B^*\|^2 - 2\eta_t \langle \mathbf{w}_{t,0} - \mathbf{w}_B^*, \bar{\mathbf{g}}_t \rangle + \eta_t^2 \|\bar{\mathbf{g}}_t\|^2 \\ &\quad + 2\eta_t \langle \mathbf{w}_{t,0} - \mathbf{w}_B^* - \eta_t \bar{\mathbf{g}}_t, \bar{\mathbf{g}}_t - \mathbf{g}_t \rangle + \eta_t^2 \|\mathbf{g}_t - \bar{\mathbf{g}}_t\|^2, \end{aligned} \quad (\text{D.44})$$

where, in (D.41), we used Assumption 4; whereas, the inequality in (D.42) is due to the contracting property of projection. We observe that (D.42) does not hold in general if  $\mathbf{w}_B^* \notin W$ .  $\square$

In what follows, we present a series of lemmas to establish bounds for the error in (D.40).

**Lemma D.5.** *Let Assumption 5 hold and the local functions  $\{F_k\}_{k=1}^N$  be convex. We have:*

$$\begin{aligned} -2\eta_t \langle \mathbf{w}_{t,0} - \mathbf{w}_B^*, \bar{\mathbf{g}}_t \rangle &\leq -2\eta_t (1 - \eta_t L) \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left( F_k(\mathbf{w}_{t,j}^k) - F_k(\mathbf{w}_B^*) \right) \\ &\quad + \underbrace{\sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \|\mathbf{w}_{t,j}^k - \mathbf{w}_{t,0}\|^2}_{\text{bounded in Lemma D.10}} + \underbrace{2\eta_t^2 L E \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_B^*) - F_k^*)}_{\text{bounded in Lemma D.11}}. \end{aligned} \quad (\text{D.45})$$

*Proof of Lemma D.5.* We decompose the term  $-2\eta_t \langle \mathbf{w}_{t,0} - \mathbf{w}_B^*, \bar{\mathbf{g}}_t \rangle$ , by adding and subtracting  $\mathbf{w}_{t,j}^k$ :

$$-2\eta_t \langle \mathbf{w}_{t,0} - \mathbf{w}_B^*, \bar{\mathbf{g}}_t \rangle = \underbrace{-2\eta_t \langle \mathbf{w}_{t,0} - \mathbf{w}_{t,j}^k, \bar{\mathbf{g}}_t \rangle}_{\text{developed in Eq. (D.47)}} - \underbrace{2\eta_t \langle \mathbf{w}_{t,j}^k - \mathbf{w}_B^*, \bar{\mathbf{g}}_t \rangle}_{\text{developed in Eq. (D.51)}}. \quad (\text{D.46})$$

We bound the two terms separately. We bound the first term in (D.46) as:

$$-2\eta_t \langle \mathbf{w}_{t,0} - \mathbf{w}_{t,j}^k, \bar{\mathbf{g}}_t \rangle = -2\eta_t \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \langle \nabla F_k(\mathbf{w}_{t,j}^k), \mathbf{w}_{t,0} - \mathbf{w}_{t,j}^k \rangle \quad (\text{D.47})$$

$$\leq \eta_t^2 \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left\| \nabla F_k(\mathbf{w}_{t,j}^k) \right\|^2 + \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left\| \mathbf{w}_{t,j}^k - \mathbf{w}_{t,0} \right\|^2 \quad (\text{D.48})$$

$$\leq 2\eta_t^2 L \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left( F_k(\mathbf{w}_{t,j}^k) - F_k^* \right) + \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left\| \mathbf{w}_{t,j}^k - \mathbf{w}_{t,0} \right\|^2 \quad (\text{D.49})$$

$$\begin{aligned} &= 2\eta_t^2 L \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left( F_k(\mathbf{w}_{t,j}^k) - F_k(\mathbf{w}_B^*) \right) + \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left\| \mathbf{w}_{t,j}^k - \mathbf{w}_{t,0} \right\|^2 \\ &\quad + 2\eta_t^2 L E \sum_{k \in \mathcal{A}_t} q_k \left( F_k(\mathbf{w}_B^*) - F_k^* \right), \end{aligned} \quad (\text{D.50})$$

where, in (D.48), we used  $|\langle \mathbf{a}, \mathbf{b} \rangle| \leq \frac{1}{2} \|\mathbf{a}\|^2 + \frac{1}{2} \|\mathbf{b}\|^2$ ; in (D.49), we applied the  $L$ -smoothness of  $\{F_k(\mathbf{w})\}_{k \in \mathcal{K}}$  (Assumption 5); in (D.50), we added and subtracted  $F_k(\mathbf{w}_B^*)$ .

We bound the second term in (D.46) as:

$$-2\eta_t \langle \mathbf{w}_{t,j}^k - \mathbf{w}_B^*, \bar{\mathbf{g}}_t \rangle = -2\eta_t \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \langle \mathbf{w}_{t,j}^k - \mathbf{w}_B^*, \nabla F_k(\mathbf{w}_{t,j}^k) \rangle \quad (\text{D.51})$$

$$\leq -2\eta_t \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left( F_k(\mathbf{w}_{t,j}^k) - F_k(\mathbf{w}_B^*) \right), \quad (\text{D.52})$$

where, in (D.52), we use the convexity of  $\{F_k(\mathbf{w})\}_{k \in \mathcal{K}}$ .

By summing the bounds provided in (D.50) and (D.52), we conclude the proof.  $\square$

**Lemma D.6** (Bound on the squared norm of a global gradient step). *Let Assumption 5 hold. We have:*

$$\eta_t^2 \|\bar{\mathbf{g}}_t\|^2 \leq 2\eta_t^2 L E Q \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left( F_k(\mathbf{w}_{t,j}^k) - F_k(\mathbf{w}_B^*) \right) + 2\eta_t^2 L E^2 Q \underbrace{\sum_{k \in \mathcal{A}_t} q_k \left( F_k(\mathbf{w}_B^*) - F_k^* \right)}_{\text{bounded in Lemma D.11}}. \quad (\text{D.53})$$

*Proof of Lemma D.6.*

$$\eta_t^2 \|\bar{\mathbf{g}}_t\|^2 = \eta_t^2 \left\| \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \nabla F_k(\mathbf{w}_{t,j}^k) \right\|^2 \quad (\text{D.54})$$

$$\leq \eta_t^2 \sum_{k' \in \mathcal{A}_t} q_{k'} \sum_{k \in \mathcal{A}_t} q_k \left\| \sum_{j=0}^{E-1} \nabla F_k(\mathbf{w}_{t,j}^k) \right\|^2 \quad (\text{D.55})$$

$$\leq \eta_t^2 Q E \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left\| \nabla F_k(\mathbf{w}_{t,j}^k) \right\|^2 \quad (\text{D.56})$$

$$\leq 2\eta_t^2 QLE \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left( F_k(\mathbf{w}_{t,j}^k) - F_k^* \right) \quad (\text{D.57})$$

$$= 2\eta_t^2 LEQ \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left( F_k(\mathbf{w}_{t,j}^k) - F_k(\mathbf{w}_B^*) \right) + 2\eta_t^2 LE^2 Q \sum_{k \in \mathcal{A}_t} q_k \left( F_k(\mathbf{w}_B^*) - F_k^* \right), \quad (\text{D.58})$$

where, in (D.55) and in (D.56), we applied the Jensen's inequality; in (D.56), we also observed that  $\sum_{k \in \mathcal{A}_t} q_k \leq \sum_{k \in \mathcal{K}} q_k := Q$ ; in (D.57), we used the  $L$ -smoothness of  $\{F_k(\mathbf{w})\}_{k \in \mathcal{K}}$  (Assumption 5); in (D.58), we added and subtracted  $F_k(\mathbf{w}_B^*)$  to the sum.  $\square$

**Lemma D.7.** *Let Assumption 7 hold. We have:*

$$\begin{aligned} 2\eta_t \mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} [\langle \mathbf{w}_{t,0} - \mathbf{w}_B^* - \eta_t \bar{\mathbf{g}}_t, \bar{\mathbf{g}}_t - \mathbf{g}_t \rangle] &\leq \\ &2\eta_t^2 LEQ \sum_{k \in \mathcal{A}_t} q_k \sum_{j=1}^{E-1} \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left[ F_k(\mathbf{w}_{t,j}^k) - F_k(\mathbf{w}_B^*) \right] \\ &+ \frac{1}{2} \eta_t^2 E(E-1) \sum_{k \in \mathcal{A}_t} q_k^2 \sigma_k^2 \\ &+ 2\eta_t^2 LE^2 Q \underbrace{\sum_{k \in \mathcal{A}_t} q_k \left( F_k(\mathbf{w}_B^*) - F_k^* \right)}_{\text{bounded in Lemma D.11}}. \end{aligned} \quad (\text{D.59})$$

*Proof of Lemma D.7.* We decompose the term  $\langle \mathbf{w}_{t,0} - \mathbf{w}_B^* - \eta_t \bar{\mathbf{g}}_t, \bar{\mathbf{g}}_t - \mathbf{g}_t \rangle$  in two parts:

$$2\eta_t \langle \mathbf{w}_{t,0} - \mathbf{w}_B^* - \eta_t \bar{\mathbf{g}}_t, \bar{\mathbf{g}}_t - \mathbf{g}_t \rangle = 2\eta_t \langle \mathbf{w}_{t,0} - \mathbf{w}_B^*, \bar{\mathbf{g}}_t - \mathbf{g}_t \rangle - 2\eta_t^2 \langle \bar{\mathbf{g}}_t, \bar{\mathbf{g}}_t - \mathbf{g}_t \rangle. \quad (\text{D.60})$$

From Lemma D.3, we conclude that  $\mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} \langle \mathbf{w}_{t,0} - \mathbf{w}_B^*, \bar{\mathbf{g}}_t - \mathbf{g}_t \rangle = 0$ .

We now focus on:

$$-2\eta_t^2 \mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} [\langle \bar{\mathbf{g}}_t, \bar{\mathbf{g}}_t - \mathbf{g}_t \rangle] = \quad (\text{D.61})$$

$$= -2\eta_t^2 \mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{k \in \mathcal{A}_t} \sum_{k' \in \mathcal{A}_t} q_k q_{k'} \sum_{j=0}^{E-1} \sum_{j'=0}^{E-1} \langle \nabla F_k(\mathbf{w}_{t,j}^k), \nabla F_{k'}(\mathbf{w}_{t,j'}^{k'}) - \nabla F_{k'}(\mathbf{w}_{t,j'}^{k'}, \mathcal{B}_{t,j'}^{k'}) \rangle \right] \quad (\text{D.62})$$

$$\begin{aligned} &= -2\eta_t^2 \mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=0}^{E-1} \sum_{j'=0}^{E-1} \langle \nabla F_k(\mathbf{w}_{t,j}^k), \nabla F_k(\mathbf{w}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}^k, \mathcal{B}_{t,j'}^k) \rangle \right] \\ &\quad - 2\eta_t^2 \mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{\substack{k \in \mathcal{A}_t \\ k' \neq k}} \sum_{k' \in \mathcal{A}_t} q_k q_{k'} \sum_{j=0}^{E-1} \sum_{j'=0}^{E-1} \langle \nabla F_k(\mathbf{w}_{t,j}^k), \nabla F_{k'}(\mathbf{w}_{t,j'}^{k'}) - \nabla F_{k'}(\mathbf{w}_{t,j'}^{k'}, \mathcal{B}_{t,j'}^{k'}) \rangle \right] \end{aligned} \quad (\text{D.63})$$

$$= -2\eta_t^2 \sum_{k \in \mathcal{A}_t} q_k^2 \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{j=0}^{E-1} \sum_{j'=0}^{E-1} \langle \nabla F_k(\mathbf{w}_{t,j}^k), \nabla F_k(\mathbf{w}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}^k, \mathcal{B}_{t,j'}^k) \rangle \right]$$



$$\begin{aligned}
& -2\eta_t^2 \sum_{k \in \mathcal{A}_t} \sum_{\substack{k' \in \mathcal{A}_t \\ k' \neq k}} q_k q_{k'} \sum_{j=0}^{E-1} \sum_{j'=0}^{E-1} \left\langle \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \nabla F_k(\mathbf{w}_{t,j}^k) \right], \right. \\
& \left. \mathbb{E}_{\mathcal{B}_{t,0:j'-1}^{k'} | \mathcal{A}_t, \mathcal{H}_t} \left[ \underbrace{\mathbb{E}_{\mathcal{B}_{t,j'}^{k'} | \mathcal{B}_{t,0:j'-1}^{k'}, \mathcal{A}_t, \mathcal{H}_t} \left[ \nabla F_{k'}(\mathbf{w}_{t,j'}^{k'}) - \nabla F_{k'}(\mathbf{w}_{t,j'}, \mathcal{B}_{t,j'}^{k'}) \right]}_{=0} \right] \right\rangle, \quad (\text{D.64})
\end{aligned}$$

where, in (D.62), we replaced the definitions of  $g_t$  and  $\bar{g}_t$  given in (D.26) and in (D.27), respectively; in (D.63), we consider the cases  $k = k'$  and  $k \neq k'$  separately; (D.64) follows from the consideration that local models of different clients evolve independently and then all the terms with  $k' \neq k$  equal zero because  $\nabla F_k(\mathbf{w}, \mathcal{B})$  is an unbiased estimator of  $\nabla F_k(\mathbf{w})$ . It follows that:

$$-2\eta_t^2 \mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} [\langle \bar{\mathbf{g}}_t, \bar{\mathbf{g}}_t - \mathbf{g}_t \rangle] = \quad (\text{D.65})$$

$$= -2\eta_t^2 \sum_{k \in \mathcal{A}_t} q_k^2 \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{j=0}^{E-1} \sum_{j'=0}^{E-1} \langle \nabla F_k(\mathbf{w}_{t,j}^k), \nabla F_k(\mathbf{w}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}, \mathcal{B}_{t,j'}^k) \rangle \right] \quad (\text{D.66})$$

$$\begin{aligned}
& = -2\eta_t^2 \sum_{k \in \mathcal{A}_t} q_k^2 \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{\substack{j=0 \\ j' < j}}^{E-1} \sum_{j'=0}^{E-1} \langle \nabla F_k(\mathbf{w}_{t,j}^k), \nabla F_k(\mathbf{w}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}, \mathcal{B}_{t,j'}^k) \rangle \right] \\
& - 2\eta_t^2 \sum_{k \in \mathcal{A}_t} q_k^2 \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{\substack{j=0 \\ j' \geq j}}^{E-1} \sum_{j'=0}^{E-1} \langle \nabla F_k(\mathbf{w}_{t,j}^k), \nabla F_k(\mathbf{w}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}, \mathcal{B}_{t,j'}^k) \rangle \right] \quad (\text{D.67})
\end{aligned}$$

$$\begin{aligned}
& = -2\eta_t^2 \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=0}^{E-1} \sum_{\substack{j'=0 \\ j' < j}}^{E-1} \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \langle \nabla F_k(\mathbf{w}_{t,j}^k), \nabla F_k(\mathbf{w}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}, \mathcal{B}_{t,j'}^k) \rangle \right] \\
& - 2\eta_t^2 \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=0}^{E-1} \sum_{\substack{j'=0 \\ j' \geq j}}^{E-1} \mathbb{E}_{\mathcal{B}_{t,0:j'-1}^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \mathbb{E}_{\mathcal{B}_{t,j'}^k | \mathcal{B}_{t,0:j'-1}^k, \mathcal{A}_t, \mathcal{H}_t} \left[ \langle \nabla F_k(\mathbf{w}_{t,j}^k), \right. \right. \\
& \left. \left. \nabla F_k(\mathbf{w}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}, \mathcal{B}_{t,j'}^k) \rangle \right] \right] \quad (\text{D.68})
\end{aligned}$$

$$\begin{aligned}
& = -2\eta_t^2 \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=0}^{E-1} \sum_{\substack{j'=0 \\ j' < j}}^{E-1} \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \langle \nabla F_k(\mathbf{w}_{t,j}^k), \nabla F_k(\mathbf{w}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}, \mathcal{B}_{t,j'}^k) \rangle \right] \\
& - 2\eta_t^2 \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=0}^{E-1} \sum_{\substack{j'=0 \\ j' \geq j}}^{E-1} \mathbb{E}_{\mathcal{B}_{t,0:j'-1}^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \langle \nabla F_k(\mathbf{w}_{t,j}^k), \right.
\end{aligned}$$

$$\underbrace{\mathbb{E}_{\mathcal{B}_{t,j}^k | \mathcal{B}_{t,0:j'-1}^k, \mathcal{A}_t, \mathcal{H}_t} \left[ \nabla F_k(\mathbf{w}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}^k, \mathcal{B}_{t,j'}^k) \right]}_{=0} \rangle, \quad (\text{D.69})$$

where, in (D.67), we consider the cases  $j' < j$  and  $j' \geq j$  separately; then, in (D.68) and in (D.69), we use the law of total expectation.

Finally, we bound the remaining term in the right-hand side of (D.69) as follows:

$$-2\eta_t^2 \mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} [\langle \bar{\mathbf{g}}_t, \bar{\mathbf{g}}_t - \mathbf{g}_t \rangle] \quad (\text{D.70})$$

$$= -2\eta_t^2 \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=1}^{E-1} \sum_{j' < j} \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \langle \nabla F_k(\mathbf{w}_{t,j}^k), \nabla F_k(\mathbf{w}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}^k, \mathcal{B}_{t,j'}^k) \rangle \quad (\text{D.71})$$

$$= \eta_t^2 \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=1}^{E-1} \sum_{j' < j} \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \left\| \nabla F_k(\mathbf{w}_{t,j}^k) \right\|^2 + \left\| \nabla F_k(\mathbf{w}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}^k, \mathcal{B}_{t,j'}^k) \right\|^2 \right] \quad (\text{D.72})$$

$$\begin{aligned} &= \eta_t^2 \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=1}^{E-1} \sum_{j' < j} \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \left\| \nabla F_k(\mathbf{w}_{t,j}^k) \right\|^2 \right] \\ &\quad + \eta_t^2 \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=1}^{E-1} \sum_{j' < j} \mathbb{E}_{\mathcal{B}_{t,0:j'-1}^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \underbrace{\left\| \nabla F_k(\mathbf{w}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}^k, \mathcal{B}_{t,j'}^k) \right\|^2}_{\text{bounded with Assumption 7}} \right] \end{aligned} \quad (\text{D.73})$$

$$\leq \eta_t^2 \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=1}^{E-1} \sum_{j' < j} \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left\| \nabla F_k(\mathbf{w}_{t,j}^k) \right\|^2 + \frac{1}{2} \eta_t^2 E(E-1) \sum_{k \in \mathcal{A}_t} q_k^2 \sigma_k^2 \quad (\text{D.74})$$

$$\leq \eta_t^2 L(E-1) \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=1}^{E-1} \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \left( F_k(\mathbf{w}_{t,j}^k) - F_k^* \right) \right] + \frac{1}{2} \eta_t^2 E(E-1) \sum_{k \in \mathcal{A}_t} q_k^2 \sigma_k^2 \quad (\text{D.75})$$

$$\begin{aligned} &= \eta_t^2 L(E-1) \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=1}^{E-1} \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \left( F_k(\mathbf{w}_{t,j}^k) - F_k(\mathbf{w}_B^*) \right) \right] \\ &\quad + \eta_t^2 LE(E-1) \sum_{k \in \mathcal{A}_t} q_k^2 \left( F_k(\mathbf{w}_B^*) - F_k^* \right) + \frac{1}{2} \eta_t^2 E(E-1) \sum_{k \in \mathcal{A}_t} q_k^2 \sigma_k^2 \end{aligned} \quad (\text{D.76})$$

$$\begin{aligned} &\leq \eta_t^2 L(E-1) Q \sum_{k \in \mathcal{A}_t} q_k \sum_{j=1}^{E-1} \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \left( F_k(\mathbf{w}_{t,j}^k) - F_k(\mathbf{w}_B^*) \right) \right] \\ &\quad + \underbrace{\eta_t^2 LE(E-1) Q \sum_{k \in \mathcal{A}_t} q_k \left( F_k(\mathbf{w}_B^*) - F_k^* \right)}_{\text{bounded in Lemma D.11}} + \frac{1}{2} \eta_t^2 E(E-1) \sum_{k \in \mathcal{A}_t} q_k^2 \sigma_k^2, \end{aligned} \quad (\text{D.77})$$

where, in (D.72), we used  $|\langle \mathbf{a}, \mathbf{b} \rangle| \leq \frac{1}{2} \|\mathbf{a}\|^2 + \frac{1}{2} \|\mathbf{b}\|^2$ ; in (D.74), we applied Assumption 7; in (D.75), we used the  $L$ -smoothness of  $\{F_k(\mathbf{w})\}_{k \in \mathcal{K}}$ ; in (D.76), we added and subtracted  $F_k(\mathbf{w}_B^*)$  from the sum; finally, in (D.77), we used  $\sum_{k \in \mathcal{A}_t} q_k^2 f(k) \leq (\sum_{k \in \mathcal{A}_t} q_k)(\sum_{k \in \mathcal{A}_t} q_k f(k))$  and  $\sum_{k \in \mathcal{A}_t} q_k \leq \sum_{k=1}^N q_k := Q$ . Noting that  $E-1 < 2E$  concludes the proof of Lemma D.7.  $\square$

**Lemma D.8** (Bound on the variance of the stochastic gradients). *Let Assumption 7 hold. Similarly to [Li+19, Lemma 2], we have:*

$$\eta_t^2 \mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} \|\mathbf{g}_t - \bar{\mathbf{g}}_t\|^2 \leq \eta_t^2 E \sum_{k \in \mathcal{A}_t} q_k^2 \sigma_k^2. \quad (\text{D.78})$$

*Proof of Lemma D.8.*

$$\begin{aligned} & \mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} \|\mathbf{g}_t - \bar{\mathbf{g}}_t\|^2 = \\ &= \mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} \left\| \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left( \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) - \nabla F_k(\mathbf{w}_{t,j}^k) \right) \right\|^2 \quad (\text{D.79}) \\ &= \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=0}^{E-1} \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left\| \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) - \nabla F_k(\mathbf{w}_{t,j}^k) \right\|^2 \\ &+ \sum_{k \in \mathcal{A}_t} q_k^2 \mathbb{E}_{\mathcal{B}_t^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{j=0}^{E-1} \sum_{\substack{j'=0 \\ j' \neq j}}^{E-1} \left\langle \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) - \nabla F_k(\mathbf{w}_{t,j}^k), \right. \right. \\ &\quad \left. \left. \nabla F_k(\mathbf{w}_{t,j'}^k, \mathcal{B}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}^k) \right\rangle \right] \\ &+ \sum_{k \in \mathcal{A}_t} \sum_{\substack{k' \in \mathcal{A}_t \\ k' \neq k}} q_k q_{k'} \sum_{j=0}^{E-1} \left\langle \mathbb{E}_{\mathcal{B}_{t,0:j-1}^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \underbrace{\mathbb{E}_{\mathcal{B}_{t,j}^k | \mathcal{B}_{t,0:j-1}^k, \mathcal{A}_t, \mathcal{H}_t} \left[ \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) - \nabla F_k(\mathbf{w}_{t,j}^k) \right]}_{=0} \right], \right. \\ &\quad \left. \mathbb{E}_{\mathcal{B}_{t,0:j-1}^{k'} | \mathcal{A}_t, \mathcal{H}_t} \left[ \underbrace{\mathbb{E}_{\mathcal{B}_{t,j}^{k'} | \mathcal{B}_{t,0:j-1}^{k'}, \mathcal{A}_t, \mathcal{H}_t} \left[ \nabla F_{k'}(\mathbf{w}_{t,j}^{k'}, \mathcal{B}_{t,j}^{k'}) - \nabla F_{k'}(\mathbf{w}_{t,j}^{k'}) \right]}_{=0} \right] \right\rangle \\ &+ \sum_{k \in \mathcal{A}_t} \sum_{\substack{k' \in \mathcal{A}_t \\ k' \neq k}} q_k q_{k'} \sum_{j=0}^{E-1} \sum_{\substack{j'=0 \\ j' \neq j}}^{E-1} \left\langle \mathbb{E}_{\mathcal{B}_{t,0:j-1}^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \underbrace{\mathbb{E}_{\mathcal{B}_{t,j}^k | \mathcal{B}_{t,0:j-1}^k, \mathcal{A}_t, \mathcal{H}_t} \left[ \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) - \nabla F_k(\mathbf{w}_{t,j}^k) \right]}_{=0} \right], \right. \\ &\quad \left. \mathbb{E}_{\mathcal{B}_{t,0:j-1}^{k'} | \mathcal{A}_t, \mathcal{H}_t} \left[ \underbrace{\mathbb{E}_{\mathcal{B}_{t,j}^{k'} | \mathcal{B}_{t,0:j-1}^{k'}, \mathcal{A}_t, \mathcal{H}_t} \left[ \nabla F_{k'}(\mathbf{w}_{t,j'}^{k'}, \mathcal{B}_{t,j'}^{k'}) - \nabla F_{k'}(\mathbf{w}_{t,j'}^{k'}) \right]}_{=0} \right] \right\rangle \quad (\text{D.80}) \\ &= \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=0}^{E-1} \underbrace{\mathbb{E}_{\mathcal{B}_{t,j}^k | \mathcal{A}_t, \mathcal{H}_t} \left\| \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) - \nabla F_k(\mathbf{w}_{t,j}^k) \right\|^2}_{\text{bounded with Assumption 7}} \\ &+ \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=0}^{E-1} \sum_{\substack{j'=0 \\ j' < j}}^{E-1} \mathbb{E}_{\mathcal{B}_{t,0:j-1}^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \mathbb{E}_{\mathcal{B}_{t,j}^k | \mathcal{B}_{t,0:j-1}^k, \mathcal{A}_t, \mathcal{H}_t} \left[ \left\langle \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) - \nabla F_k(\mathbf{w}_{t,j}^k), \right. \right. \right. \\ &\quad \left. \left. \nabla F_k(\mathbf{w}_{t,j'}^k, \mathcal{B}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}^k) \right\rangle \right] \right] \end{aligned}$$

$$\begin{aligned}
& + \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=0}^{E-1} \sum_{\substack{j'=0 \\ j' > j}}^{E-1} \mathbb{E}_{\mathcal{B}_{t,0:j'-1}^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \mathbb{E}_{\mathcal{B}_{t,j'}^k | \mathcal{B}_{t,0:j'-1}^k, \mathcal{A}_t, \mathcal{H}_t} \left[ \left\langle \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) - \nabla F_k(\mathbf{w}_{t,j}^k), \right. \right. \right. \\
& \qquad \qquad \qquad \left. \left. \left. \nabla F_k(\mathbf{w}_{t,j'}^k, \mathcal{B}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}^k) \right\rangle \right] \right] \tag{D.81}
\end{aligned}$$

$$\begin{aligned}
& = \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=0}^{E-1} \underbrace{\mathbb{E}_{\mathcal{B}_{t,j}^k | \mathcal{A}_t, \mathcal{H}_t} \left\| \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) - \nabla F_k(\mathbf{w}_{t,j}^k) \right\|^2}_{\text{bounded with Assumption 7}} \\
& + \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=0}^{E-1} \sum_{\substack{j'=0 \\ j' < j}}^{E-1} \mathbb{E}_{\mathcal{B}_{t,0:j-1}^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \underbrace{\mathbb{E}_{\mathcal{B}_{t,j}^k | \mathcal{B}_{t,0:j-1}^k, \mathcal{A}_t, \mathcal{H}_t} \left[ \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) - \nabla F_k(\mathbf{w}_{t,j}^k) \right]}_{=0}, \right. \\
& \qquad \qquad \qquad \left. \left. \nabla F_k(\mathbf{w}_{t,j'}^k, \mathcal{B}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}^k) \right\rangle \right] \\
& + \sum_{k \in \mathcal{A}_t} q_k^2 \sum_{j=0}^{E-1} \sum_{\substack{j'=0 \\ j' > j}}^{E-1} \mathbb{E}_{\mathcal{B}_{t,0:j'-1}^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \left\langle \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) - \nabla F_k(\mathbf{w}_{t,j}^k), \right. \right. \\
& \qquad \qquad \qquad \left. \left. \underbrace{\mathbb{E}_{\mathcal{B}_{t,j'}^k | \mathcal{B}_{t,0:j'-1}^k, \mathcal{A}_t, \mathcal{H}_t} \left[ \nabla F_k(\mathbf{w}_{t,j'}^k, \mathcal{B}_{t,j'}^k) - \nabla F_k(\mathbf{w}_{t,j'}^k) \right]}_{=0} \right\rangle \right] \tag{D.82}
\end{aligned}$$

$$\leq E \sum_{k \in \mathcal{A}_t} q_k^2 \sigma_k^2, \tag{D.83}$$

where, in (D.80), (D.81), and (D.82), we used the law of total expectation; in (D.83), we applied Assumption 7.

Multiplying both sides of (D.83) by  $\eta_t^2$  completes the proof of Lemma D.8.  $\square$

**Lemma D.9.** *Let Assumption 5 hold and let the local functions  $\{F_k\}_{k=1}^N$  be convex. Define  $\gamma_t := 2\eta_t(1 - \eta_t L(1 + 2EQ))$ .*

*For a diminishing step-size  $0 < \eta_t \leq \frac{1}{2L(1+2EQ)}$ , satisfying  $\gamma_t > 0$ , we have:*

$$\begin{aligned}
& -\gamma_t \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left( F_k(\mathbf{w}_{t,j}^k) - F_k(\mathbf{w}_B^*) \right) \\
& \leq -\frac{1}{2} \eta_t E \sum_{k \in \mathcal{A}_t} q_k \left( F_k(\mathbf{w}_{t,0}) - F_k(\mathbf{w}_B^*) \right) + \underbrace{\sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left\| \mathbf{w}_{t,j}^k - \mathbf{w}_{t,0} \right\|^2}_{\text{bounded in Lemma D.10}} \\
& \quad + \underbrace{2\eta_t^2 L E \sum_{k \in \mathcal{A}_t} q_k \left( F_k(\mathbf{w}_B^*) - F_k^* \right)}_{\text{bounded in Lemma D.11}}, \tag{D.84}
\end{aligned}$$

*Proof of Lemma D.9.* In the following, we require  $\gamma_t > 0$ .

$$-\gamma_t \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left( F_k(\mathbf{w}_{t,j}^k) - F_k(\mathbf{w}_B^*) \right) \tag{D.85}$$

$$= -\gamma_t \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left( F_k(\mathbf{w}_{t,j}^k) - F_k(\mathbf{w}_{t,0}) \right) - \gamma_t \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left( F_k(\mathbf{w}_{t,0}) - F_k(\mathbf{w}_B^*) \right) \quad (\text{D.86})$$

$$\leq -\gamma_t \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \langle \nabla F_k(\mathbf{w}_{t,0}), \mathbf{w}_{t,j}^k - \mathbf{w}_{t,0} \rangle - \gamma_t E \sum_{k \in \mathcal{A}_t} q_k \left( F_k(\mathbf{w}_{t,0}) - F_k(\mathbf{w}_B^*) \right) \quad (\text{D.87})$$

$$\leq \gamma_t \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \frac{1}{2} \left[ \eta_t \|\nabla F_k(\mathbf{w}_{t,0})\|^2 + \frac{1}{\eta_t} \|\mathbf{w}_{t,j}^k - \mathbf{w}_{t,0}\|^2 \right] - \gamma_t E \sum_{k \in \mathcal{A}_t} q_k \left( F_k(\mathbf{w}_{t,0}) - F_k(\mathbf{w}_B^*) \right) \quad (\text{D.88})$$

$$\leq \gamma_t \eta_t L E \sum_{k \in \mathcal{A}_t} q_k \left( F_k(\mathbf{w}_{t,0}) - F_k^* \right) + \frac{\gamma_t}{2\eta_t} \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \|\mathbf{w}_{t,j}^k - \mathbf{w}_{t,0}\|^2 - \gamma_t E \sum_{k \in \mathcal{A}_t} q_k \left( F_k(\mathbf{w}_{t,0}) - F_k(\mathbf{w}_B^*) \right) \quad (\text{D.89})$$

$$\leq -\gamma_t E (1 - \eta_t L) \sum_{k \in \mathcal{A}_t} q_k \left( F_k(\mathbf{w}_{t,0}) - F_k(\mathbf{w}_B^*) \right) + \frac{\gamma_t}{2\eta_t} \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \|\mathbf{w}_{t,j}^k - \mathbf{w}_{t,0}\|^2 + \gamma_t \eta_t L E \sum_{k \in \mathcal{A}_t} q_k \left( F_k(\mathbf{w}_B^*) - F_k^* \right) \quad (\text{D.90})$$

where, in (D.86), we added and subtracted  $F_k(\mathbf{w}_{t,0})$  to the sum; in (D.87), we used the convexity of  $\{F_k(\mathbf{w})\}_{k \in \mathcal{K}}$ ; note that (D.87) also requires  $\gamma_t > 0$ ; in (D.88), we used the inequality  $|\langle \mathbf{a}, \mathbf{b} \rangle| \leq \frac{1}{2} \|\mathbf{a}\|^2 + \frac{1}{2} \|\mathbf{b}\|^2$ ; in (D.89), we applied the  $L$ -smoothness of  $\{F_k(\mathbf{w})\}_{k \in \mathcal{K}}$  (Assumption 5); finally, in (D.90), we added and subtracted  $F_k(\mathbf{w}_B^*)$  to the sum.

In particular, for  $\gamma_t := 2\eta_t(1 - \eta_t L(1 + 2EQ)) > 0$ , since  $0 < \eta_t \leq \frac{1}{2L(1+2EQ)}$ , we further obtain:

$$\begin{aligned} & -\gamma_t \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left( F_k(\mathbf{w}_{t,j}^k) - F_k(\mathbf{w}_B^*) \right) \\ & \leq -\frac{1}{2} \eta_t E \sum_{k \in \mathcal{A}_t} q_k \left( F_k(\mathbf{w}_{t,0}) - F_k(\mathbf{w}_B^*) \right) + \underbrace{\sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \|\mathbf{w}_{t,j}^k - \mathbf{w}_{t,0}\|^2}_{\text{bounded in Lemma D.10}} \end{aligned} \quad (\text{D.91})$$

$$+ \underbrace{2\eta_t^2 L E \sum_{k \in \mathcal{A}_t} q_k \left( F_k(\mathbf{w}_B^*) - F_k^* \right)}_{\text{bounded in Lemma D.11}}, \quad (\text{D.92})$$

where, in (D.92), we used  $0 < \eta_t \leq \frac{1}{2L(1+2EQ)}$ , which gives  $-\gamma_t E(1 - \eta_t L) = -2\eta_t E(1 - \eta_t L(1 + 2EQ))$  ( $1 - \eta_t L \leq -\frac{1}{2}\eta_t E$ ). Moreover, since  $\gamma_t \leq 2\eta_t$ , we also used  $\gamma_t \eta_t \leq 2\eta_t^2$ , and  $\frac{\gamma_t}{2\eta_t} \leq 1$ .  $\square$

**Lemma D.10** (Bound on the divergence of local models). *Let Assumption 4, 5, and 7 hold, the local functions  $\{F_k\}_{k=1}^N$  be convex and  $G$  be defined as in Lemma D.2, Equation (2.20). Similarly to [Li+19, Lemma 3], we obtain the following inequality:*

$$\mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \|\mathbf{w}_{t,j}^k - \mathbf{w}_{t,0}\|^2 \right] \leq \frac{1}{2} \eta_t^2 E^3 G^2 \left( \sum_{k \in \mathcal{A}_t} q_k \right). \quad (\text{D.93})$$

*Proof of Lemma D.10.*

$$\mathbb{E}_{\mathcal{B}_t|\mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \left\| \mathbf{w}_{t,j}^k - \mathbf{w}_{t,0} \right\|^2 \right] = \mathbb{E}_{\mathcal{B}_t|\mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{k \in \mathcal{A}_t} q_k \sum_{j=1}^{E-1} \eta_t^2 \left\| \sum_{j'=0}^{j-1} \nabla F_k(\mathbf{w}_{t,j'}^k, \mathcal{B}_{t,j'}^k) \right\|^2 \right] \quad (\text{D.94})$$

$$\leq \eta_t^2 \sum_{k \in \mathcal{A}_t} q_k \sum_{j=1}^{E-1} j \sum_{j'=0}^{j-1} \mathbb{E}_{\mathcal{B}_t^k|\mathcal{A}_t, \mathcal{H}_t} \left[ \left\| \nabla F_k(\mathbf{w}_{t,j'}^k, \mathcal{B}_{t,j'}^k) \right\|^2 \right] \quad (\text{D.95})$$

$$\leq \eta_t^2 G^2 \left( \sum_{j=1}^{E-1} j^2 \right) \left( \sum_{k \in \mathcal{A}_t} q_k \right) \quad (\text{D.96})$$

$$= \frac{1}{6} \eta_t^2 E(E-1)(2E-1) G^2 \left( \sum_{k \in \mathcal{A}_t} q_k \right), \quad (\text{D.97})$$

where, in (D.95), we used the triangle and the Jensen's inequalities; in (D.96), we applied the bound in Lemma 2.2.1, Equation (2.20); finally, in (D.97), we developed the sum of sequence of squares  $\sum_{j=1}^{E-1} j^2 = \frac{1}{6} E(E-1)(2E-1) \leq \frac{1}{2} E^3$  since  $E \geq 1$ .  $\square$

**Lemma D.11** (Bound on the dissimilarity of local functions). *Let Assumption 3 hold and  $(\mathcal{A}_t)_{t \geq 0}$  defined therein. We have:*

$$\mathbb{E} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_B^*) - F_k^*) \right] \leq \left( \sum_{k=1}^N \pi_k q_k \right) \Gamma, \quad (\text{D.98})$$

where  $\Gamma$  is defined in (2.22).

*Proof of Lemma D.11.*

$$\mathbb{E} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_B^*) - F_k^*) \right] = \sum_{k=1}^N \pi_k q_k (F_k(\mathbf{w}_B^*) - F_k^*) \quad (\text{D.99})$$

$$= \left( \sum_{k'=1}^N \pi_{k'} q_{k'} \right) \sum_{k=1}^N p_k (F_k(\mathbf{w}_B^*) - F_k^*) \quad (\text{D.100})$$

$$\leq \left( \sum_{k'=1}^N \pi_{k'} q_{k'} \right) \sum_{k=1}^N p_k (F_k(\mathbf{w}^*) - F_k^*) \quad (\text{D.101})$$

$$\leq \left( \sum_{k'=1}^N \pi_{k'} q_{k'} \right) \underbrace{\max_{k \in \mathcal{K}} \{ (F_k(\mathbf{w}^*) - F_k^*) \}}_{:=\Gamma} = \left( \sum_{k=1}^N \pi_k q_k \right) \Gamma, \quad (\text{D.102})$$

where, in (D.99), we solved the total expectation, observing that  $\mathbb{E} [\sum_{k \in \mathcal{A}_t} q_k f(k)] = \sum_{k=1}^N \pi_k q_k f(k)$  (Assumption 3); in (D.100), we applied  $p_k := \frac{\pi_k q_k}{\sum_{k'=1}^N \pi_{k'} q_{k'}}$ ; in (D.101), we used  $F_B(\mathbf{w}) := \sum_{k=1}^N p_k F_k(\mathbf{w})$  and we observed  $F_B(\mathbf{w}_B^*) \leq F_B(\mathbf{w}^*)$ ; finally, in (D.102), we used  $\sum_{k=1}^N p_k = 1$  and  $\Gamma := \max_{k \in \mathcal{K}} \{ (F_k(\mathbf{w}^*) - F_k^*) \}$ .  $\square$

**Lemma D.12** (Convergence results under heterogeneous client availability). *Let Assumptions 3–5 and 7 hold and the functions  $\{F_k\}_{k=1}^N$  be convex. For a diminishing step-size  $0 < \eta_t \leq \frac{1}{2L(1+2EQ)}$  satisfying  $\sum_{t=1}^{+\infty} \eta_t^2 < +\infty$ , for any  $t_0 \leq T$ , we have:*

$$\begin{aligned} \sum_{t=t_0}^T \eta_t \mathbb{E} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_{t,0}) - F_k(\mathbf{w}_B^*)) \right] &\leq \frac{2}{E} \text{diam}(W)^2 + (E+1) \left( \sum_{k=1}^N \pi_k q_k^2 \sigma_k^2 \right) \left( \sum_{t=1}^{+\infty} \eta_t^2 \right) \\ &\quad + 2E^2 G^2 \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=1}^{+\infty} \eta_t^2 \right) \\ &\quad + 4L(1+EQ) \Gamma \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=1}^{+\infty} \eta_t^2 \right) \\ &:= C_0 < +\infty. \end{aligned} \tag{D.103}$$

*Proof of Lemma D.12.* We take expectation over  $\mathcal{B}_t \mid \mathcal{A}_t, \mathcal{H}_t$  on Lemma D.4:

$$\begin{aligned} \mathbb{E}_{\mathcal{B}_t \mid \mathcal{A}_t, \mathcal{H}_t} \|\mathbf{w}_{t+1,0} - \mathbf{w}_B^*\|^2 &\leq \underbrace{\|\mathbf{w}_{t,0} - \mathbf{w}_B^*\|^2 - 2\eta_t \mathbb{E}_{\mathcal{B}_t \mid \mathcal{A}_t, \mathcal{H}_t} \langle \mathbf{w}_{t,0} - \mathbf{w}_B^*, \bar{\mathbf{g}}_t \rangle}_{\text{bounded in Lemma D.5}} + \underbrace{\eta_t^2 \mathbb{E}_{\mathcal{B}_t \mid \mathcal{A}_t, \mathcal{H}_t} \|\bar{\mathbf{g}}_t\|^2}_{\text{bounded in Lemma D.6}} \\ &\quad + \underbrace{2\eta_t \mathbb{E}_{\mathcal{B}_t \mid \mathcal{A}_t, \mathcal{H}_t} \langle \mathbf{w}_{t,0} - \mathbf{w}_B^* - \eta_t \bar{\mathbf{g}}_t, \bar{\mathbf{g}}_t - \mathbf{g}_t \rangle}_{\text{bounded in Lemma D.7}} + \underbrace{\eta_t^2 \mathbb{E}_{\mathcal{B}_t \mid \mathcal{A}_t, \mathcal{H}_t} \|\mathbf{g}_t - \bar{\mathbf{g}}_t\|^2}_{\text{bounded in Lemma D.8}}. \end{aligned} \tag{D.104}$$

Replacing Lemmas D.5–D.8 in (D.104), we obtain:

$$\begin{aligned} \mathbb{E}_{\mathcal{B}_t \mid \mathcal{A}_t, \mathcal{H}_t} \|\mathbf{w}_{t+1,0} - \mathbf{w}_B^*\|^2 &\leq \|\mathbf{w}_{t,0} - \mathbf{w}_B^*\|^2 + 2\eta_t^2 LE(1+2EQ) \mathbb{E}_{\mathcal{B}_t \mid \mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_B^*) - F_k^*) \right] \\ &\quad - \underbrace{2\eta_t(1 - \eta_t L(1+2EQ)) \mathbb{E}_{\mathcal{B}_t \mid \mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{k \in \mathcal{A}_t} q_k \sum_{j=1}^{E-1} (F_k(\mathbf{w}_{t,j}^k) - F_k(\mathbf{w}_B^*)) \right]}_{\gamma_t} \\ &\quad + \frac{1}{2} \eta_t^2 E(E+1) \sum_{k \in \mathcal{A}_t} q_k^2 \sigma_k^2 + \underbrace{\mathbb{E}_{\mathcal{B}_t \mid \mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \|\mathbf{w}_{t,j}^k - \mathbf{w}_{t,0}\|^2 \right]}_{\text{bounded in Lemma D.10}} \end{aligned} \tag{D.105}$$

We apply Lemmas D.9 and D.10 to (D.105) with  $\gamma_t := 2\eta_t(1 - \eta_t L(1+2EQ))$ . We observe that  $\gamma_t > 0$  because:

$$0 \leq \eta_t \leq \frac{1}{2L(1+2EQ)}. \tag{D.106}$$

We obtain:

$$\mathbb{E}_{\mathcal{B}_t \mid \mathcal{A}_t, \mathcal{H}_t} \|\mathbf{w}_{t+1,0} - \mathbf{w}_B^*\|^2 \leq \|\mathbf{w}_{t,0} - \mathbf{w}_B^*\|^2 - \frac{1}{2} \eta_t E \mathbb{E}_{\mathcal{B}_t \mid \mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_{t,0}) - F_k(\mathbf{w}_B^*)) \right]$$

$$\begin{aligned}
& + \frac{1}{2} \eta_t^2 E(E+1) \sum_{k \in \mathcal{A}_t} q_k^2 \sigma_k^2 + \eta_t^2 E^3 G^2 \sum_{k \in \mathcal{A}_t} q_k \\
& + 4\eta_t^2 LE(1+EQ) \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_B^*) - F_k^*) \right]. \tag{D.107}
\end{aligned}$$

Computing the total expectation on (D.107), we have:

$$\begin{aligned}
& \mathbb{E}_{\mathcal{A}_t, \mathcal{B}_t, \mathcal{H}_t} \|\mathbf{w}_{t+1,0} - \mathbf{w}_B^*\|^2 \\
& \leq \mathbb{E}_{\mathcal{H}_t} \|\mathbf{w}_{t,0} - \mathbf{w}_B^*\|^2 - \frac{1}{2} \eta_t E \mathbb{E}_{\mathcal{A}_t, \mathcal{B}_t, \mathcal{H}_t} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_{t,0}) - F_k(\mathbf{w}_B^*)) \right] \\
& \quad + \frac{1}{2} \eta_t^2 E(E+1) \mathbb{E}_{\mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{k \in \mathcal{A}_t} q_k^2 \sigma_k^2 \right] + \eta_t^2 E^3 G^2 \mathbb{E}_{\mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{k \in \mathcal{A}_t} q_k \right] \\
& \quad + 4\eta_t^2 LE(1+EQ) \underbrace{\mathbb{E}_{\mathcal{A}_t, \mathcal{H}_t} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_B^*) - F_k^*) \right]}_{\text{bounded in Lemma D.11}} \tag{D.108}
\end{aligned}$$

Applying Lemma D.11 to (D.108) and considering  $\mathbb{E} [\sum_{k \in \mathcal{A}_t} a_k] = \sum_{k=1}^N \pi_k a_k$  (Assumption 3), the following inequality holds:

$$\begin{aligned}
\mathbb{E} \|\mathbf{w}_{t+1,0} - \mathbf{w}_B^*\|^2 & \leq \mathbb{E} \|\mathbf{w}_{t,0} - \mathbf{w}_B^*\|^2 - \frac{1}{2} \eta_t E \mathbb{E} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_{t,0}) - F_k(\mathbf{w}_B^*)) \right] \\
& \quad + \frac{1}{2} \eta_t^2 E(E+1) \left( \sum_{k=1}^N \pi_k q_k^2 \sigma_k^2 \right) + \eta_t^2 E^3 G^2 \left( \sum_{k=1}^N \pi_k q_k \right) + 4\eta_t^2 LE(1+EQ) \Gamma \left( \sum_{k=1}^N \pi_k q_k \right). \tag{D.109}
\end{aligned}$$

Rearranging and summing over  $t = t_0, \dots, T$ , we obtain the following inequality:

$$\begin{aligned}
\sum_{t=t_0}^T \eta_t \mathbb{E} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_{t,0}) - F_k(\mathbf{w}_B^*)) \right] & \leq \frac{2}{E} \sum_{t=t_0}^T \mathbb{E} \left[ \left( \|\mathbf{w}_{t,0} - \mathbf{w}_B^*\|^2 - \|\mathbf{w}_{t+1,0} - \mathbf{w}_B^*\|^2 \right) \right] \\
& \quad + (E+1) \left( \sum_{k=1}^N \pi_k q_k^2 \sigma_k^2 \right) \left( \sum_{t=t_0}^T \eta_t^2 \right) \\
& \quad + 2E^2 G^2 \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=t_0}^T \eta_t^2 \right) \\
& \quad + 4L(1+EQ) \Gamma \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=t_0}^T \eta_t^2 \right). \tag{D.110}
\end{aligned}$$

The first term in the right-hand side of (D.110) is a telescoping sum and we remove the negative term  $-\mathbb{E} \|\mathbf{w}_{T+1,0} - \mathbf{w}_B^*\|^2$ :

$$\sum_{t=t_0}^T \eta_t \mathbb{E} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_{t,0}) - F_k(\mathbf{w}_B^*)) \right]$$



$$\begin{aligned}
&\leq \frac{2}{E} \mathbb{E} \|\mathbf{w}_{t_0,0} - \mathbf{w}_B^*\|^2 + (E+1) \left( \sum_{k=1}^N \pi_k q_k^2 \sigma_k^2 \right) \left( \sum_{t=t_0}^T \eta_t^2 \right) \\
&\quad + 2E^2 G^2 \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=t_0}^T \eta_t^2 \right) \\
&\quad + 4L(1+EQ) \Gamma \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=t_0}^T \eta_t^2 \right). \tag{D.111}
\end{aligned}$$

Finally, by noting that  $\|\mathbf{w}_{t_0,0} - \mathbf{w}_B^*\| \leq \text{diam}(W)$  and  $\sum_{t=t_0}^T \eta_t^2 \leq \sum_{t=1}^{+\infty} \eta_t^2 < +\infty$ , we complete the proof of Lemma D.12.  $\square$

**Lemma D.13.** *Let Assumptions 4 and 5 hold, and the local functions  $\{F_k\}_{k=1}^N$  be convex. We have:*

$$|F_k(\mathbf{v}) - F_k(\mathbf{w})| \leq D \cdot \|\mathbf{v} - \mathbf{w}\|, \quad \forall \mathbf{v}, \mathbf{w} \in W \tag{D.112}$$

*Proof of Lemma D.13.* In Lemma D.2, under Assumptions 4 and 5, we have already proved that:

$$\|\nabla F_k(\mathbf{w})\| \leq D. \tag{2.19}$$

Moreover, from the convexity of  $\{F_k\}_{k \in \mathcal{K}}$ , it follows that:

$$\langle \nabla F_k(\mathbf{v}), \mathbf{v} - \mathbf{w} \rangle \leq F_k(\mathbf{v}) - F_k(\mathbf{w}) \leq \langle \nabla F_k(\mathbf{w}), \mathbf{v} - \mathbf{w} \rangle. \tag{D.113}$$

The Cauchy–Schwarz inequality completes the proof of Lemma D.13:

$$|F_k(\mathbf{v}) - F_k(\mathbf{w})| \leq \max\{\|\nabla F_k(\mathbf{v})\|, \|\nabla F_k(\mathbf{w})\|\} \cdot \|\mathbf{v} - \mathbf{w}\| \leq D \cdot \|\mathbf{v} - \mathbf{w}\|. \tag{D.114}$$

$\square$

**Lemma D.14.** *Let Assumptions 4, 5, and 7 hold. We have:*

$$\mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} \|\mathbf{w}_{t+1,0} - \mathbf{w}_{t,0}\| \leq \eta_t EG \left( \sum_{k \in \mathcal{A}_t} q_k \right). \tag{D.115}$$

*Proof of Lemma D.14.* The proof is based on [SSY18, Proposition 1.4].

$$\begin{aligned}
&\mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} \|\mathbf{w}_{t+1,0} - \mathbf{w}_{t,0}\| = \\
&\quad \mathbb{E}_{\mathcal{B}_t | \mathcal{A}_t, \mathcal{H}_t} \left\| -\eta_t \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k) \right\| \tag{D.116}
\end{aligned}$$

$$\leq \eta_t \sum_{k \in \mathcal{A}_t} q_k \sum_{j=0}^{E-1} \mathbb{E}_{\mathcal{B}_{t,0:j-1}^k | \mathcal{A}_t, \mathcal{H}_t} \left[ \mathbb{E}_{\mathcal{B}_{t,j}^k | \mathcal{B}_{t,0:j-1}^k, \mathcal{A}_t, \mathcal{H}_t} \left[ \|\nabla F_k(\mathbf{w}_{t,j}^k, \mathcal{B}_{t,j}^k)\| \right] \right] \tag{D.117}$$

$$\leq \eta_t EG \left( \sum_{k \in \mathcal{A}_t} q_k \right), \tag{D.118}$$

where, in (D.117), we used the triangle inequality and the law of total expectation; in (D.118), we applied Lemma 2.2.1, Equation (2.20).  $\square$

Similarly to [SSY18, Theorem 1], we provide the following definition.

**Definition 9.** For communication round  $t \geq 1$ , denote the positive integer  $\mathcal{J}_t$  as follows:

$$\mathcal{J}_t := \min \left\{ \max \left\{ \left\lceil \frac{\ln(2C_P H t)}{\ln(1/\lambda(\mathbf{P}))} \right\rceil, T_P \right\}, t \right\}. \quad (\text{D.119})$$

The parameter  $\mathcal{J}_t$  is crucial in our analysis: it represents the communication rounds needed to bound the stationary distribution convergence of the Markov process  $(\mathcal{A}_t)_{t>0}$ . It will play a key role in Lemmas D.15–D.19 and in the proof of Theorem D.20. We remark that, by definition:  $T_P \leq \mathcal{J}_t \leq t$ .

Our definition of  $\mathcal{J}_t$  corrects a typo in [SSY18, (6.27)], which considered  $\ln(t/(2C_P H))$  rather than  $\ln(2C_P H t)$ . In fact, we observe that [SSY18, (6.28)] and consequently [SSY18, (6.35)] do not hold when  $\mathcal{J}_t$  is defined as in [SSY18, (6.27)].

**Lemma D.15** (Convergence results under heterogeneous and correlated client availability after  $\mathcal{J}_t$  communication rounds). *Let Assumptions 3–5, and 7 hold, the local functions  $\{F_k\}_{k=1}^N$  be convex, and the parameter  $\mathcal{J}_t \leq t$  be as in Definition 9. For a diminishing step-size  $\{\eta_t\}_{t \geq 1}$  satisfying  $\sum_{t=1}^{+\infty} \ln(t) \cdot \eta_t^2$ , for any  $t_0 \leq T$ , we have:*

$$\sum_{t=t_0}^T \eta_t \mathbb{E} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_{t,0})) \right] \leq \frac{C_1}{\ln(1/\lambda(\mathbf{P}))} < +\infty, \quad (\text{D.120})$$

where:

$$C_1 := EDGQ \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=1}^{+\infty} \ln(2C_P H t) \eta_{t-\mathcal{J}_t}^2 \right). \quad (\text{D.121})$$

*Proof of Lemma D.15.* This proof is based on [SSY18, Equation (6.31)].

$$\begin{aligned} & \sum_{t=t_0}^T \eta_t \mathbb{E} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_{t,0})) \right] \\ & \leq Q \sum_{t=t_0}^T \eta_t \mathbb{E} \left[ \max_{k \in \mathcal{K}} \{F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_{t,0})\} \right] \end{aligned} \quad (\text{D.122})$$

$$\leq DQ \sum_{t=t_0}^T \eta_t \mathbb{E} \|\mathbf{w}_{t-\mathcal{J}_t,0} - \mathbf{w}_{t,0}\| \quad (\text{D.123})$$

$$\leq DQ \sum_{t=t_0}^T \eta_t \sum_{d=t-\mathcal{J}_t}^{t-1} \mathbb{E}_{\mathcal{A}_d, \mathcal{H}_d} \left[ \mathbb{E}_{\mathcal{B}_d | \mathcal{A}_d, \mathcal{H}_d} \|\mathbf{w}_{d,0} - \mathbf{w}_{d+1,0}\| \right] \quad (\text{D.124})$$

$$\leq EDGQ \sum_{t=t_0}^T \sum_{d=t-\mathcal{J}_t}^{t-1} \eta_t \eta_d \mathbb{E} \left[ \sum_{k \in \mathcal{A}_d} q_k \right] \quad (\text{D.125})$$

$$\leq EDGQ \left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=t_0}^T \sum_{d=t-\mathcal{J}_t}^{t-1} \eta_t \eta_d \quad (\text{D.126})$$

$$\leq \frac{EDGQ}{2} \left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=t_0}^T \sum_{d=t-\mathcal{J}_t}^{t-1} (\eta_t^2 + \eta_d^2) \quad (\text{D.127})$$

$$\leq EDGQ \left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=t_0}^T \mathcal{J}_t \eta_{t-\mathcal{J}_t}^2, \quad (\text{D.128})$$

where, in (D.122), we used  $\sum_{k \in \mathcal{A}_t} q_k a_k \leq \sum_{k=1}^N q_k a_k \leq (\sum_{k=1}^N q_k) \cdot \max_{k \in \mathcal{K}} \{a_k\} = Q \cdot \max_{k \in \mathcal{K}} \{a_k\}$ ; in (D.123), we applied Lemma D.13; in (D.124), we used the triangle inequality and the law of total expectation; in (D.125), we applied Lemma D.14 and again the law of total expectation; in (D.126), we observed that  $\mathbb{E} \left[ \sum_{k \in \mathcal{A}_d} q_k \right] = \sum_{k=1}^N \pi_k q_k$  (Assumption 3); in (D.127), we used  $2ab \leq a^2 + b^2$ ; finally, in (D.128), we applied  $\eta_t < \eta_d \leq \eta_{t-\mathcal{J}_t}$  due to the diminishing learning rate.

We apply then the definition of  $\mathcal{J}_t$  in (D.119) and we observe that  $\sum_{t=t_0}^T \ln(t) \eta_{t-\mathcal{J}_t}^2 \leq \sum_{t=1}^{+\infty} \ln(t) \eta_{t-\mathcal{J}_t}^2$ :

$$\sum_{t=t_0}^T \eta_t \mathbb{E} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_{t,0})) \right] \leq EDGQ \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=t_0}^T \frac{\ln(2C_P H t)}{\ln(1/\lambda(\mathbf{P}))} \eta_{t-\mathcal{J}_t}^2 \right) \quad (\text{D.129})$$

$$\leq EDGQ \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=1}^{+\infty} \frac{\ln(2C_P H t)}{\ln(1/\lambda(\mathbf{P}))} \eta_{t-\mathcal{J}_t}^2 \right) = \frac{C_1}{\ln(1/\lambda(\mathbf{P}))}. \quad (\text{D.130})$$

Finally, we conclude that  $C_1$  is finite. To this purpose, we observe that  $\mathcal{J}_t \leq a \ln(t) + b$ , for opportune positive values  $a$  and  $b$ . Let  $t'$  be a positive integer such that  $t \geq a \ln(t) + b$  for any  $t \geq t'$ . Then:

$$\sum_{t=t'}^T \ln(t) \cdot \eta_{t-\mathcal{J}_t}^2 = \sum_{t=t'-\mathcal{J}_t}^{T-\mathcal{J}_t} \ln(t + \mathcal{J}_t) \cdot \eta_t^2 \quad (\text{D.131})$$

$$\leq \sum_{t=1}^{+\infty} \ln(t + a \ln t + b) \cdot \eta_t^2 \quad (\text{D.132})$$

$$\leq \sum_{t=1}^{+\infty} \ln((1 + a + b)t) \cdot \eta_t^2 < +\infty. \quad (\text{D.133})$$

□

**Lemma D.16.** *Let Assumptions 4, 5 and 7 hold, the local functions  $\{F_k\}_{k=1}^N$  be convex, and  $\mathcal{J}_t \leq t$  be as in Definition 9. Let the step-size be decreasing and satisfy:  $\sum_{t=1}^{+\infty} \ln(t) \cdot \eta_t^2 < +\infty$ . For any  $t_0 \leq T$ , we have:*

$$\left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=t_0}^T \eta_t \mathbb{E} [F_B(\mathbf{w}_{t,0}) - F_B(\mathbf{w}_{t-\mathcal{J}_t,0})] \leq \frac{C_1}{\ln(1/\lambda(\mathbf{P}))} < +\infty, \quad (\text{D.134})$$

where:

$$C_1 := EDGQ \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=1}^{+\infty} \ln(2C_P H t) \eta_{t-\mathcal{J}_t}^2 \right). \quad (\text{D.135})$$

*Proof of Lemma D.16.* This proof is based on [SSY18, Equation (6.38)].

$$\begin{aligned} \left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=t_0}^T \eta_t \mathbb{E} [F_B(\mathbf{w}_{t,0}) - F_B(\mathbf{w}_{t-\mathcal{J}_t,0})] &= \\ \sum_{t=t_0}^T \eta_t \sum_{k=1}^N \pi_k q_k \mathbb{E} [F_k(\mathbf{w}_{t,0}) - F_k(\mathbf{w}_{t-\mathcal{J}_t,0})] &\quad (\text{D.136}) \end{aligned}$$

$$\leq D \left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=t_0}^T \eta_t \mathbb{E} \|\mathbf{w}_{t-\mathcal{J}_t,0} - \mathbf{w}_{t,0}\| \quad (\text{D.137})$$

$$\leq D \left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=t_0}^T \eta_t \sum_{d=t-\mathcal{J}_t}^{t-1} \mathbb{E}_{\mathcal{A}_d, \mathcal{H}_d} \left[ \mathbb{E}_{\mathcal{B}_d | \mathcal{A}_d, \mathcal{H}_d} \|\mathbf{w}_{d,0} - \mathbf{w}_{d+1,0}\| \right] \quad (\text{D.138})$$

$$\leq DEGQ \left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=t_0}^T \sum_{d=t-\mathcal{J}_t}^{t-1} \eta_t \eta_d \quad (\text{D.139})$$

$$\leq \frac{DEGQ}{2} \left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=t_0}^T \sum_{d=t-\mathcal{J}_t}^{t-1} (\eta_t^2 + \eta_d^2) \quad (\text{D.140})$$

$$\leq DEGQ \left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=t_0}^T \mathcal{J}_t \cdot \eta_{t-\mathcal{J}_t}^2 \quad (\text{D.141})$$

$$\leq EDGQ \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=1}^{+\infty} \frac{\ln(2C_P H t)}{\ln(1/\lambda(\mathbf{P}))} \eta_{t-\mathcal{J}_t}^2 \right) = \frac{C_1}{\ln(1/\lambda(\mathbf{P}))}, \quad (\text{D.142})$$

where, in (D.136), we applied  $F_B(\mathbf{w}) = \sum_{k=1}^N p_k F_k(\mathbf{w})$ , where  $p_k = \frac{\pi_k q_k}{\sum_{h=1}^N \pi_h q_h}$ ; in (D.137), we applied Lemma D.13; in (D.138), we applied the triangle inequality and the law of total expectation; in (D.139), we applied Lemma D.14; in (D.140), we used  $2ab \leq a^2 + b^2$ ; in (D.141), we observed that  $\eta_t^2 + \eta_d^2 \leq 2\eta_{t-\mathcal{J}_t}^2$  due to the diminishing learning rate; finally, in (D.142), we applied the definition of  $\mathcal{J}_t$  given in (D.119) and we observed that  $\sum_{t=t_0}^T \ln(t) \eta_{t-\mathcal{J}_t}^2 \leq \sum_{t=1}^{+\infty} \ln(t) \eta_{t-\mathcal{J}_t}^2 < +\infty$  and then  $C_1 < +\infty$ .  $\square$

**Lemma D.17** (Bound on the distance dynamics between the current and the stationary distributions of the Markov process). *Let Assumption 3 hold, and  $\mathbf{P}$ ,  $\boldsymbol{\rho}$  defined therein. The following inequality holds:*

$$\max_{i,j \in [M]} |[\mathbf{P}^t]_{i,j} - \rho_j| \leq C_P \cdot \lambda(\mathbf{P})^t, \quad \text{for } t \geq T_P, \quad (5)$$

where  $C_P$  and  $T_P$  are positive constants defined as:

$$C_P := \left( \sum_{i=2}^d n_i^2 \right)^{\frac{1}{2}} \cdot \|\mathbf{U}\|_F \|\mathbf{U}^{-1}\|_F, \quad (\text{D.143})$$

$$T_P := \max \left\{ \max_{1 \leq i \leq d} \left\{ \left\lceil \frac{2n_i(n_i - 1)(\ln(\frac{2n_i}{\ln \lambda(\mathbf{P})/|\lambda_2(\mathbf{P})|}) - 1)}{(n_i + 1) \ln(\lambda(\mathbf{P})/|\lambda_2(\mathbf{P})|)} \right\rceil \right\}, 0 \right\}. \quad (\text{D.144})$$

Here,  $d$ ,  $n_i$ , and  $\mathbf{U}$  are quantities related to the Jordan canonical form of  $\mathbf{P}$ . Specifically,  $\mathbf{P} = \mathbf{U} \mathbf{J} \mathbf{U}^{-1}$ , where  $\mathbf{J}$  denotes the Jordan  $M \times M$  matrix with  $d$  blocks  $\mathbf{J}_i$ ,  $i = 2, \dots, d$ . Each block

$\mathbf{J}_i$ ,  $i = 2, 3, \dots, d$ , has a dimension  $n_i \geq 1$ , and  $\sum_{i=1}^d n_i = M$ . Moreover,  $|\mathbf{U}|_F$  denotes the Frobenius norm of the matrix  $\mathbf{U}$ .

Furthermore, let Assumptions 4 and 5 hold,  $H$  be defined as in Lemma D.2, Equation (2.21), and  $T_P \leq \mathcal{J}_t \leq t$  be defined in (D.119). We obtain the additional inequality:

$$\left| [\mathbf{P}^{\mathcal{J}_t}]_{i,j} - \rho_j \right| \leq C_P \cdot \lambda(\mathbf{P})^t \leq C_P \lambda(\mathbf{P})^{\mathcal{J}_t} = \frac{1}{2Ht}, \quad \forall i, j \in [M] \text{ and } \forall t \geq T_P. \quad (\text{D.145})$$

*Proof of Lemma D.17.* The inequality in (2.18) is proven in [SSY18, Lemma 1] and holds for any  $t \geq T_P$ . Here,  $T_P$  is a constant dependent on the transition matrix  $\mathbf{P}$  of the Markov chain  $(\mathcal{A}_t)_{t \geq 0}$  defined in Assumption 3. To prove (D.145), we further observe that  $0 < \lambda(\mathbf{P}) \leq 1$  and  $T_P \leq \mathcal{J}_t \leq t$ . The last inequality in (D.145) follows from the definition of  $\mathcal{J}_t$  in (D.119).  $\square$

We remark that the bounds in [SSY18, Lemma 1], and consequently our (D.145), require  $t \geq T_P$ . Therefore, the derivations in [SSY18, (6.28)] and [SSY18, (6.35)–(6.37)] are not accurate, since they hold for  $t \geq T_P$ . We address this problem with Lemmas D.18 and D.19.

**Lemma D.18.** *Let Assumptions 3–5 hold, and  $T_P$  be defined as in (D.144). The following inequality holds:*

$$\left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=1}^{T_P-1} \eta_t \mathbb{E} [F_B(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_B^*] \leq C_2 < +\infty, \quad (\text{D.146})$$

where:

$$C_2 := H \left( \sum_{t=1}^{T_P-1} \eta_t \right) \left( \sum_{k=1}^N \pi_k q_k \right) < +\infty. \quad (\text{D.147})$$

*Proof of Lemma D.18.*

$$\left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=1}^{T_P-1} \eta_t \mathbb{E} [F_B(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_B^*] = \sum_{t=1}^{T_P-1} \eta_t \sum_{k=1}^N \pi_k q_k \mathbb{E} [F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_B^*)] \quad (\text{D.148})$$

$$\leq H \left( \sum_{t=1}^{T_P-1} \eta_t \right) \left( \sum_{k=1}^N \pi_k q_k \right) := C_2 < +\infty, \quad (\text{D.149})$$

where, in (D.148), we used the definition of  $F_B$  from (2.17), and in (D.149), we applied Lemma 2.2.1, Equation (2.21), which holds for any  $\mathbf{w} \in W$ . Lastly, it is worth noting that  $C_2$  is a sum of finite elements, and is therefore finite.  $\square$

**Lemma D.19.** *Let Assumptions 3–5 and 7 hold, and  $\{F_k\}_{k=1}^N$  be convex. Recall the definitions of  $\mathcal{J}_t$  and  $T_P$  in (D.119) and in (D.144), respectively. Let the step-size  $(\eta_t)_{t \geq 1}$  decrease and satisfy  $\eta_1 \leq \frac{1}{2L(1+2EQ)}$ ,  $\sum_{t=1}^{+\infty} \eta_t^2 < +\infty$ , and  $\sum_{t=1}^{+\infty} \ln(t) \cdot \eta_t^2 < +\infty$ . For  $t \geq T_P$ , we have:*

$$\left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=T_P}^T \eta_t \mathbb{E} [F_B(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_B^*] \leq \frac{C_1}{\ln(1/\lambda(\mathbf{P}))} + C_3 < +\infty, \quad (\text{D.150})$$

where:

$$C_1 := EDGQ \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=1}^{+\infty} \ln(2C_P H t) \cdot \eta_{t-\mathcal{J}_t}^2 \right) < +\infty. \quad (\text{D.151})$$

$$C_3 := C_0 + \frac{MQ}{4} \sum_{t=1}^{+\infty} \left( \eta_t^2 + \frac{1}{t^2} \right) < +\infty; \quad (\text{D.152})$$

*Proof of Lemma D.19.* Assume  $t \geq T_P$ . With a similar proof technique to [SSY18, (6.35)], we derive the following lower bound:

$$\begin{aligned} & \mathbb{E}_{\mathcal{A}_t | \mathcal{A}_{t-\mathcal{J}_t}, \mathcal{H}_{t-\mathcal{J}_t}} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_B^*)) \right] = \\ &= \sum_{a \in \mathcal{M}} \mathbb{P}(\mathcal{A}_t = a | \mathcal{A}_{t-\mathcal{J}_t}, \mathcal{H}_{t-\mathcal{J}_t}) \sum_{k \in a} q_k (F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_B^*)) \end{aligned} \quad (\text{D.153})$$

$$= \sum_{a \in \mathcal{M}} [\mathbf{P}^{\mathcal{J}_t}]_{\mathcal{A}_{t-\mathcal{J}_t}, a} \sum_{k \in a} q_k (F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_B^*)) \quad (\text{D.154})$$

$$\geq \sum_{a \in \mathcal{M}} \left( \rho_a - \frac{1}{2Ht} \right) \sum_{k \in a} q_k (F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_B^*)) \quad (\text{D.155})$$

$$= \sum_{k=1}^N \mathbb{E} [\mathbf{1}_{k \in \mathcal{A}_t}] q_k (F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_B^*)) - \frac{1}{2Ht} \sum_{a \in \mathcal{M}} \sum_{k \in a} q_k (F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_B^*)) \quad (\text{D.156})$$

$$\geq \sum_{k=1}^N \pi_k q_k (F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_B^*)) - \frac{MQ}{2Ht} \max_{k \in \mathcal{K}} \{F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_B^*)\} \quad (\text{D.157})$$

$$\geq \left( \sum_{k=1}^N \pi_k q_k \right) \cdot (F_B(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_B^*) - \frac{MQ}{2t}, \quad (\text{D.158})$$

where, in (D.153), we applied the definition of expected value to the random variable  $\mathcal{A}_t$ , with  $a$  representing a realization of  $\mathcal{A}_t$ , that is a state in the state space  $\mathcal{M}$ , and  $\mathbb{P}(\mathcal{A}_t = a | \mathcal{A}_{t-\mathcal{J}_t}, \mathcal{H}_{t-\mathcal{J}_t})$  denoting the conditional probability of the event  $\mathcal{A}_t = a$  given  $(\mathcal{A}_{t-\mathcal{J}_t}, \mathcal{H}_{t-\mathcal{J}_t})$ ; in (D.154), we applied the Markov property (Assumption 3), observing that  $\mathbb{P}(\mathcal{A}_t = a | \mathcal{A}_{t-\mathcal{J}_t}) = [\mathbf{P}^{\mathcal{J}_t}]_{\mathcal{A}_{t-\mathcal{J}_t}, a}$ , where  $[\mathbf{P}^k]_{i,j}$  denotes the  $(i, j)$ -th element of the  $k$ -th power of the transition matrix  $\mathbf{P}$ ; in (D.155), we applied Lemma D.17, Equation (D.145); for the first term in (D.156), we used  $\sum_{a \in \mathcal{M}} \rho_a \sum_{k \in a} f(k) = \sum_{a \in \mathcal{M}} \rho_a \sum_{k=1}^N \mathbf{1}_{\{k \in a\}} f(k) = \sum_{k=1}^N f(k) \sum_{a \in \mathcal{M}} \rho_a \mathbf{1}_{k \in a} = \sum_{k=1}^N f(k) \mathbb{E} [\mathbf{1}_{k \in \mathcal{A}_t}]$ , where  $\mathbf{1}_{k \in \mathcal{A}_t}$  is the indicator function that equals 1 if and only if  $k \in \mathcal{A}_t$ ; in (D.157), we used  $\mathbb{E} [\mathbf{1}_{k \in \mathcal{A}_t}] = \mathbb{P}(k \in \mathcal{A}_t) := \pi_k$  for the first term, and  $\sum_{k \in a} q_k f(k) \leq \sum_{k=1}^N q_k f(k) \leq (\sum_{k=1}^N q_k) (\max_{k \in \mathcal{K}} f(k)) = Q \max_{k \in \mathcal{K}} f(k)$  and  $\sum_{a \in \mathcal{M}} 1 = M$  for the second term; finally, in (D.158), we used the definition of  $F_B$  in (2.17) for the first term, and we used Lemma 2.2.1, Equation (2.21) for the second term.

Our derivations in (D.157) and (D.158) correct a typo in [SSY18, (6.35)], which considered  $Q/(2t)$  instead of  $(MQ)/(2t)$ . In (D.158), the dimension ( $M$ ) of the state space ( $\mathcal{M}$ ) of the Markov chain  $(\mathcal{A}_t)_{t \geq 0}$  appears in the numerator of the second term.

Note that the steps in (D.155)–(D.158) require  $t \geq T_P$ . Multiplying by  $\eta_t$  and summing for  $t = T_P, \dots, T$ , rearranging, and computing the total expectation, we obtain the following

inequality:

$$\left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=T_P}^T \eta_t \mathbb{E} [F_B(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_B^*] \leq \sum_{t=T_P}^T \eta_t \mathbb{E} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_B^*)) \right] + \frac{MQ}{2} \sum_{t=T_P}^T \frac{\eta_t}{t} \quad (\text{D.159})$$

$$\leq \underbrace{\sum_{t=T_P}^T \eta_t \mathbb{E} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_B^*)) \right]}_{\text{bounded with Lemma D.12 + Lemma D.15}} + \frac{MQ}{4} \sum_{t=1}^T \left( \eta_t^2 + \frac{1}{t^2} \right), \quad (\text{D.160})$$

where, in (D.160), we used  $2ab \leq a^2 + b^2$  and we observed that  $\sum_{t=T_P}^T \left( \eta_t^2 + \frac{1}{t^2} \right) \leq \sum_{t=1}^T \left( \eta_t^2 + \frac{1}{t^2} \right)$  since  $t > 0$  and  $\eta_t > 0$ .

Moreover, if the step-size  $(\eta_t)_{t \geq 1}$  decreases and satisfies  $\eta_1 \leq \frac{1}{2L(1+2EQ)}$ ,  $\sum_{t=1}^{+\infty} \eta_t^2 < +\infty$ , and  $\sum_{t=1}^{+\infty} \ln(t) \cdot \eta_t^2 < +\infty$ , we can further bound the first term in (D.160) by combining Lemma D.12 and Lemma D.15 for  $t_0 = T_P$ , and we obtain:

$$\sum_{t=T_P}^T \eta_t \mathbb{E} \left[ \sum_{k \in \mathcal{A}_t} q_k (F_k(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_k(\mathbf{w}_B^*)) \right] \leq C_0 + \frac{C_1}{\ln(1/\lambda(\mathbf{P}))} < +\infty, \quad (\text{D.161})$$

where:

$$\begin{aligned} C_0 &:= \frac{2}{E} \text{diam}(W)^2 + (E+1) \left( \sum_{k=1}^N \pi_k q_k^2 \sigma_k^2 \right) \left( \sum_{t=1}^{+\infty} \eta_t^2 \right) \\ &\quad + 2E^2 G^2 \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=1}^{+\infty} \eta_t^2 \right) \\ &\quad + 4L(1+EQ) \Gamma \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=1}^{+\infty} \eta_t^2 \right). \end{aligned} \quad (\text{D.162})$$

Finally, plugging (D.161) into (D.160), observing that  $\sum_{t=1}^T \left( \eta_t^2 + \frac{1}{t^2} \right) \leq \sum_{t=1}^{+\infty} \left( \eta_t^2 + \frac{1}{t^2} \right) < +\infty$  because  $\sum_{t=1}^{+\infty} \eta_t^2 < +\infty$  and  $\sum_{t=1}^{+\infty} \frac{1}{t^2} = \frac{\pi}{6} < +\infty$ , and denoting  $C_3 := C_0 + \frac{MQ}{4} \sum_{t=1}^{+\infty} \left( \eta_t^2 + \frac{1}{t^2} \right) < +\infty$ , we conclude the proof of Lemma D.19.  $\square$

### D.2.3 Proof of Theorem 2.2.3

**Theorem D.20** (Convergence of the optimization error  $\epsilon_{\text{opt}}$ ). *Let Assumptions 3–5 and 7 hold and the functions  $\{F_k\}_{k=1}^N$  be convex. Recall the constants  $M, L, D, G, H, \Gamma, \sigma_k, C_P, T_P, \mathcal{J}_t$ , and  $\lambda(\mathbf{P})$  defined above. Let  $Q = \sum_{k \in \mathcal{K}} q_k$ .*

*Let the step-size  $\eta_t > 0$  decrease and satisfy:*

$$\eta_1 \leq \frac{1}{2L(1+2EQ)}, \quad \sum_{t=1}^{+\infty} \eta_t = +\infty, \quad \sum_{t=1}^{+\infty} \ln(t) \cdot \eta_t^2 < +\infty. \quad (2.25)$$

Let  $T$  denote the total communication rounds.

For  $T \geq T_P$ , the expected optimization error  $\mathbb{E}[F_B(\bar{\mathbf{w}}_{T,0}) - F_B^*]$  can be bounded as follows:

$$\mathbb{E}[F_B(\bar{\mathbf{w}}_{T,0}) - F_B^*] \leq \frac{\frac{1}{2}\mathbf{q}^\top \Sigma \mathbf{q} + v}{\sum_{t=1}^T \eta_t} + \frac{\psi + \frac{\phi}{\ln(1/\lambda(\mathbf{P}))}}{\sum_{t=1}^T \eta_t}, \quad (2.26)$$

where  $\bar{\mathbf{w}}_{T,0} = \frac{\sum_{t=1}^T \eta_t \mathbf{w}_{t,0}}{\sum_{t=1}^T \eta_t}$ , and:

$$\Sigma := \text{diag} \left( 2(E+1) \pi_k \sigma_k^2 \sum_{t=1}^{+\infty} \eta_t^2 \right); \quad (D.163)$$

$$v := \frac{2}{E} \text{diam}(W)^2 + \frac{MQ}{4} \sum_{t=1}^{+\infty} \left( \eta_t^2 + \frac{1}{t^2} \right); \quad (D.164)$$

$$\psi := 4L(1 + EQ) \Gamma \left( \sum_{t=1}^{+\infty} \eta_t^2 \right) + 2E^2 G^2 \left( \sum_{t=1}^{+\infty} \eta_t^2 \right) + H \left( \sum_{t=1}^{T_P-1} \eta_t \right); \quad (D.165)$$

$$\phi := 2EDGQ \left( \sum_{t=1}^{+\infty} \ln(2C_P H t) \cdot \eta_{t-\mathcal{J}_t}^2 \right). \quad (D.166)$$

*Proof of Theorem D.20.* The proof involves three main steps.

**Step 1.** From Lemma D.16, observe that:

$$\left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=1}^T \eta_t \mathbb{E}[F_B(\mathbf{w}_{t,0}) - F_B(\mathbf{w}_{t-\mathcal{J}_t,0})] \leq \frac{C_1}{\ln(1/\lambda(\mathbf{P}))} < +\infty, \quad (D.167)$$

where:

$$C_1 := EDGQ \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=1}^{+\infty} \ln(2C_P H t) \cdot \eta_{t-\mathcal{J}_t}^2 \right) < +\infty. \quad (D.168)$$

**Step 2** By combining Lemma D.18 and Lemma D.19, we obtain:

$$\left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=1}^T \eta_t \mathbb{E}[F_B(\mathbf{w}_{t-\mathcal{J}_t,0}) - F_B^*] \leq \frac{C_1}{\ln(1/\lambda(\mathbf{P}))} + C_2 + C_3 < +\infty, \quad (D.169)$$

where  $C_1$  is defined in (D.168), and:

$$C_2 := H \left( \sum_{t=1}^{T_P-1} \eta_t \right) \left( \sum_{k=1}^N \pi_k q_k \right) < +\infty; \quad (D.170)$$

$$\begin{aligned} C_3 &:= \frac{2}{E} \text{diam}(W)^2 + (E+1) \left( \sum_{k=1}^N \pi_k q_k^2 \sigma_k^2 \right) \left( \sum_{t=1}^{+\infty} \eta_t^2 \right) \\ &\quad + 2E^2 G^2 \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=1}^{+\infty} \eta_t^2 \right) \\ &\quad + 4L(1 + EQ) \Gamma \left( \sum_{k=1}^N \pi_k q_k \right) \left( \sum_{t=1}^{+\infty} \eta_t^2 \right) + \frac{MQ}{4} \sum_{t=1}^{+\infty} \left( \eta_t^2 + \frac{1}{t^2} \right) < +\infty. \end{aligned} \quad (D.171)$$



**Step 3.** By summing the results from Steps 1 and 2, given in (D.167) and (D.169), respectively, we have:

$$\left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=1}^T \eta_t \mathbb{E}[F_B(\mathbf{w}_{t,0}) - F_B^*] \leq \frac{2C_1}{\ln(1/\lambda(\mathbf{P}))} + C_2 + C_3 < +\infty. \quad (\text{D.172})$$

With the convexity of  $F_B(\cdot)$ , applying the Jensen's inequality, we complete Step 3:

$$\left( \sum_{t=1}^T \eta_t \right) \left( \sum_{k=1}^N \pi_k q_k \right) \mathbb{E}[F_B(\bar{\mathbf{w}}_{T,0}) - F_B^*] \leq \left( \sum_{k=1}^N \pi_k q_k \right) \sum_{t=1}^T \eta_t \mathbb{E}[F_B(\mathbf{w}_{t,0}) - F_B^*] \quad (\text{D.173})$$

$$\leq \frac{2C_1}{\ln(1/\lambda(\mathbf{P}))} + C_2 + C_3 < +\infty, \quad (\text{D.174})$$

where  $\bar{\mathbf{w}}_{T,0} := \frac{\sum_{t=1}^T \eta_t \mathbf{w}_{t,0}}{\sum_{t=1}^T \eta_t}$ , and the constants  $C_1$ ,  $C_2$ , and  $C_3$  are defined in (D.168), (D.170), and (D.171), respectively.

By dividing (D.173) and (D.174) by  $\left( \sum_{t=1}^T \eta_t \right) \cdot \left( \sum_{k=1}^N \pi_k q_k \right)$ , we obtain the expression for Theorem D.20 given in (2.26).  $\square$

### D.3 Proof of Theorem 2.2.4

**Theorem D.21** (An alternative bound on the bias error  $\epsilon_{\text{bias}}$ ). *Under the same assumptions of Theorem 2.2.2, define  $\Gamma' := \max_k \{F_k(\mathbf{w}_B^*) - F_k^*\}$ . The following result holds:*

$$\epsilon_{\text{bias}} \leq 4\kappa^2 \cdot \underbrace{d_{TV}^2(\boldsymbol{\alpha}, \mathbf{p})}_{:= \tilde{\epsilon}'_{\text{bias}}} \cdot \Gamma', \quad (2.28)$$

where  $d_{TV}(\boldsymbol{\alpha}, \mathbf{p}) := \frac{1}{2} \sum_{k=1}^N |\alpha_k - p_k|$  denotes the total variation distance between the probability distributions  $\boldsymbol{\alpha}$  and  $\mathbf{p}$ .

*Proof of Theorem D.21.* The proof follows the same steps as in Theorem D.1, proceeding from (D.16) as follows:

$$\|\nabla F(\mathbf{w}_B^*)\| \leq L \sqrt{\frac{2}{\mu} \sum_{k=1}^N |\alpha_k - p_k| \sqrt{(F_k(\mathbf{w}_B^*) - F_k^*)}} \quad (\text{D.16})$$

$$\leq 2L \sqrt{\frac{2}{\mu}} d_{TV}(\boldsymbol{\alpha}, \mathbf{p}) \sqrt{\Gamma'}, \quad (\text{D.175})$$

where, in (D.175), we applied the definitions of  $d_{TV}(\boldsymbol{\alpha}, \mathbf{p}) := \frac{1}{2} \sum_{k=1}^N |\alpha_k - p_k|$  and  $\Gamma' := \max_k \{F_k(\mathbf{w}_B^*) - F_k^*\}$ .

Squaring (D.175), we obtain the following expression:

$$\|\nabla F(\mathbf{w}_B^*)\|^2 \leq \frac{8L^2}{\mu} d_{TV}^2(\boldsymbol{\alpha}, \mathbf{p}) \Gamma'. \quad (\text{D.176})$$

Then, replacing (D.176) in (D.12), we obtain:

$$\epsilon_{\text{bias}} := (F(\mathbf{w}_B^*) - F^*) \leq \frac{1}{2\mu} \|\nabla F(\mathbf{w}_B^*)\|^2 \leq 4 \frac{L^2}{\mu^2} \underbrace{d_{TV}^2(\boldsymbol{\alpha}, \mathbf{p}) \Gamma'}_{:= \tilde{\epsilon}'_{\text{bias}}}, \quad (\text{D.177})$$

which concludes the proof of Theorem D.21.  $\square$

#### D.4 Convexity of $\bar{\epsilon}_{\text{opt}} + \bar{\epsilon}_{\text{bias}}$

For the proof of the convexity of  $\bar{\epsilon}_{\text{opt}}(\mathbf{q})$ , please refer to Appendix D.5.1. To prove that  $\bar{\epsilon}_{\text{bias}}(\mathbf{q})$  is also convex, we need to study the convexity of  $\chi_{\alpha\|p}^2 := \sum_{k=1}^N (\alpha_k - p_k)^2 / p_k$  in  $\mathbf{q} \in \{q_k > 0 \forall k, \|\mathbf{q}\|_1 = Q > 0\}$ . To this purpose, we define the following functions:

$$h_k : \mathbb{R}_{\geq 0}^N \setminus \{\mathbf{0}\} \rightarrow \mathbb{R}_{\geq 0}, \quad h_k(\mathbf{q}) := \frac{\pi_k q_k}{\sum_{k'=1}^N \pi_{k'} q_{k'}}; \quad (\text{D.178})$$

$$g_k : \mathbb{R}_{> 0} \rightarrow \mathbb{R}_{\geq 0}, \quad g_k(p_k) := \frac{(p_k - \alpha_k)^2}{p_k}. \quad (\text{D.179})$$

Finally, we write the chi-square divergence  $\chi_{\alpha\|p}^2$  between the target and biased probability distributions  $\alpha$  and  $p$  as:

$$\chi_{\alpha\|p}^2(\mathbf{q}) = \sum_{k=1}^N (g_k \circ h_k)(\mathbf{q}) = \sum_{k=1}^N g_k(h_k(\mathbf{q})). \quad (\text{D.180})$$

We observe that:

- $h_k(\mathbf{q})$  is a particular case of linear-fractional functions [BV04, Example 3.32, p. 97];
- $g_k(\cdot)$  is a convex in  $p_k$  over  $\mathbb{R}_{> 0}$  because sum of convex functions;
- each  $g_k \circ h_k$  is quasi-convex in  $\mathbf{q} \in \mathbb{R}_{> 0}^N$  because composition of a convex function ( $g_k$ ) and a linear-fractional function ( $h_k$ ) [BV04, p. 102].

However, note that the sum of quasi-convex functions is not necessarily quasi-convex.

**Proposition D.22.** *The function  $\chi_{\alpha\|p}^2(\mathbf{q})$  is not convex over  $\mathbb{R}_{> 0}^N$ .*

*Proof of Proposition D.22.* To analyze the convexity of  $\chi_{\alpha\|p}^2(\mathbf{q}) = \sum_{k=1}^N (g_k \circ h_k)(\mathbf{q})$  over  $\mathbb{R}_{> 0}^N$ , a possible approach is to check whether each function  $(g_k \circ h_k)(\mathbf{q})$  is convex over  $\mathbb{R}_{> 0}^N$ . In what follows, we show that  $(g_k \circ h_k)$  is not convex over  $\mathbb{R}_{> 0}^N$ .

Consider the case when  $\pi_k = 1 \forall k \in \mathcal{K}$ . We can rewrite  $(g_k \circ h_k)(\mathbf{q})$  as follows:

$$(g_k \circ h_k)(\mathbf{q}) = \frac{\left(\frac{q_k}{\|\mathbf{q}\|_1} - \alpha_k\right)^2}{\frac{q_k}{\|\mathbf{q}\|_1}}. \quad (\text{D.181})$$

We show that this function fails to satisfy the definition of convexity, i.e.,  $\exists \mathbf{q}, \mathbf{q}' \in \mathbb{R}_{> 0}^N, \zeta \in [0, 1]$  such that:

$$(g_k \circ h_k)(\zeta \mathbf{q} + (1 - \zeta) \mathbf{q}') > \zeta (g_k \circ h_k)(\mathbf{q}) + (1 - \zeta) (g_k \circ h_k)(\mathbf{q}'). \quad (\text{D.182})$$

The left-hand side (LHS) of (D.182) is:

$$(g_k \circ h_k)(\zeta \mathbf{q} + (1 - \zeta) \mathbf{q}') = \frac{\left(\frac{\zeta q_k + (1 - \zeta) q'_k}{\zeta \|\mathbf{q}\|_1 + (1 - \zeta) \|\mathbf{q}'\|_1} - \alpha_k\right)^2}{\frac{\zeta q_k + (1 - \zeta) q'_k}{\zeta \|\mathbf{q}\|_1 + (1 - \zeta) \|\mathbf{q}'\|_1}}. \quad (\text{D.183})$$

If we take  $\mathbf{q} : \|\mathbf{q}\|_1 = 1$ ,  $q_k = \alpha_k$ ,  $\zeta = \frac{1}{2}$ ,  $\mathbf{q}' = \frac{Q}{N}\mathbf{1}$ , and we let  $Q \rightarrow +\infty$ , then the LHS in (D.183) converges to:

$$\lim_{Q \rightarrow +\infty} \frac{\left( \frac{\frac{1}{2}\alpha_k + \frac{1}{2}\frac{Q}{N} - \alpha_k}{\frac{1}{2}1 + \frac{1}{2}Q} \right)^2}{\frac{\frac{1}{2}\alpha_k + \frac{1}{2}\frac{Q}{N}}{\frac{1}{2}1 + \frac{1}{2}Q}} = \frac{\left( \frac{1}{N} - \alpha_k \right)^2}{\frac{1}{N}}. \quad (\text{D.184})$$

On the other hand, for the same choices of  $q_k$ ,  $\mathbf{q}$ ,  $\mathbf{q}'$ , and  $\zeta$ , and if we let  $Q \rightarrow +\infty$ , the right-hand side (RHS) of (D.182) is:

$$\zeta (g_k \circ h_k)(\mathbf{q}) + (1 - \zeta) (g_k \circ h_k)(\mathbf{q}') = 0 + \frac{1}{2} \frac{\left( \frac{1}{N} - \alpha_k \right)^2}{\frac{1}{N}}. \quad (\text{D.185})$$

Finally, comparing (D.184) and (D.185), we conclude that, for  $Q$  large enough, the LHS in (D.182) is larger than the RHS.  $\square$

**Proposition D.23.** *The function  $\chi_{\alpha\|p}^2(\mathbf{q})$  is convex over  $\mathbb{R}_{>0}^N \cap \{\mathbf{q} : \|\mathbf{q}\|_1 = Q > 0\}$ .*

*Proof of Proposition D.23.* To verify the convexity of  $\chi_{\alpha\|p}^2(\mathbf{q}) = \sum_{k=1}^N (g_k \circ h_k)(\mathbf{q})$  over  $\mathbb{R}_{>0}^N \cap \{\mathbf{q} : \|\mathbf{q}\|_1 = Q > 0\}$ , one possible approach is to demonstrate the convexity of each function  $(g_k \circ h_k)(\mathbf{q})$  over the set  $\mathbb{R}_{>0}^N \cap \{\mathbf{q} : \|\mathbf{q}\|_1 = Q > 0\}$ .

We prove this result for a more general case. We show that, if

$$\tilde{g} \text{ is a convex function over its domain } \mathcal{D}_g \quad (\text{D.186})$$

and

$$\tilde{h}(\mathbf{q}) = \frac{\mathbf{A}\mathbf{q} + b}{\mathbf{c}^\top \mathbf{q} + d}, \quad (\text{D.187})$$

then

$$\tilde{g} \circ \tilde{h} \text{ is convex over } \mathcal{D} = \mathbb{R}_{>0}^N \cap \left\{ \mathbf{q} : \mathbf{c}^\top \mathbf{q} + d = Q > 0, \frac{\mathbf{A}\mathbf{q} + b}{\mathbf{c}^\top \mathbf{q} + d} \in \mathcal{D}_g \right\}. \quad (\text{D.188})$$

It is then sufficient to apply this result to each pair  $(g_k, h_k)$  to conclude that  $(g_k \circ h_k)$  is convex and then  $\chi_{\alpha\|p}^2(\mathbf{q})$  is convex.

By direct inspection, for all  $\mathbf{q}, \mathbf{q}' \in \mathcal{D}$ ,  $\forall \zeta \in [0, 1]$ , the following equality holds:

$$\left( \tilde{g} \circ \tilde{h} \right) (\zeta \mathbf{q} + (1 - \zeta) \mathbf{q}') = \tilde{g} \left( \tilde{h} (\zeta \mathbf{q} + (1 - \zeta) \mathbf{q}') \right) = \tilde{g} \left( \zeta' \frac{\mathbf{A}\mathbf{q} + b}{\mathbf{c}^\top \mathbf{q} + d} + (1 - \zeta') \frac{\mathbf{A}\mathbf{q}' + b}{\mathbf{c}^\top \mathbf{q}' + d} \right), \quad (\text{D.189})$$

where:

$$\zeta' = \frac{\zeta (\mathbf{c}^\top \mathbf{q} + d)}{\zeta (\mathbf{c}^\top \mathbf{q} + d) + (1 - \zeta) (\mathbf{c}^\top \mathbf{q}' + d)} \in [0, 1]. \quad (\text{D.190})$$

Applying the convexity of  $\tilde{g}$ , we bound Equation (D.189) as follows:

$$\tilde{g}\left(\zeta' \frac{\mathbf{A}\mathbf{q} + \mathbf{b}}{\mathbf{c}^\top \mathbf{q} + d} + (1 - \zeta') \frac{\mathbf{A}\mathbf{q}' + \mathbf{b}}{\mathbf{c}^\top \mathbf{q}' + d}\right) \stackrel{\text{convexity of } \tilde{g}}{\leq} \zeta' \tilde{g}\left(\frac{\mathbf{A}\mathbf{q} + \mathbf{b}}{\mathbf{c}^\top \mathbf{q} + d}\right) + (1 - \zeta') \tilde{g}\left(\frac{\mathbf{A}\mathbf{q}' + \mathbf{b}}{\mathbf{c}^\top \mathbf{q}' + d}\right) \quad (\text{D.191})$$

$$= \zeta' (\tilde{g} \circ \tilde{h})(\mathbf{q}) + (1 - \zeta') (\tilde{g} \circ \tilde{h})(\mathbf{q}'). \quad (\text{D.192})$$

Finally, to conclude the proof, we show that  $\zeta' = \zeta$ . This is true because, for any  $\mathbf{q}$  and  $\mathbf{q}' \in \mathcal{D}$ ,  $\mathbf{c}^\top \mathbf{q} + d = \mathbf{c}^\top \mathbf{q}' + d = Q > 0$ . In fact, by using this condition in Equation (D.190), we have that:

$$\zeta' = \frac{\zeta Q}{\zeta Q + (1 - \zeta)Q} = \zeta, \quad (\text{D.193})$$

which establishes the convexity of  $\tilde{g} \circ \tilde{h}$  by definition.  $\square$

## D.5 Minimizing $\bar{\epsilon}_{\text{opt}}$

Equation (2.26) can be rewritten as:

$$\left(\sum_{t=1}^T \eta_t\right) \mathbb{E}[F_B(\bar{\mathbf{w}}_{T,0}) - F_B^*] \leq \frac{\frac{1}{2} \mathbf{q}^\top \Sigma \mathbf{q} + v}{\boldsymbol{\pi}^\top \mathbf{q}} + \psi + \frac{\phi}{\ln(1/\lambda(\mathbf{P}))} \quad (\text{D.194})$$

$$= \frac{\frac{1}{2} \mathbf{q}^\top \mathbf{A} \mathbf{q} + B}{\boldsymbol{\pi}^\top \mathbf{q}} + C := J(\mathbf{q}), \quad (\text{D.195})$$

where:

$$\mathbf{A} := \Sigma = \text{diag}\left(2(E+1)\pi_k \sigma_k^2 \sum_{t=1}^{+\infty} \eta_t^2\right); \quad (\text{D.196})$$

$$B := v = \frac{2}{E} \text{diam}(W)^2 + \frac{MQ}{4} \sum_{t=1}^{+\infty} \left(\eta_t^2 + \frac{1}{t^2}\right); \quad (\text{D.197})$$

$$C := \psi + \frac{\phi}{\ln(1/\lambda(\mathbf{P}))} \quad (\text{D.198})$$

$$= \left(4L(1 + EQ)\Gamma + 2E^2G^2\right) \left(\sum_{t=1}^{+\infty} \eta_t^2\right) + 2EDGQ \left(\sum_{t=1}^{+\infty} \mathcal{J}_t \cdot \eta_{t-\mathcal{J}_t}^2\right) + H \left(\sum_{t=1}^{T_P-1} \eta_t\right). \quad (\text{D.199})$$

The minimization of (D.195), defines the following optimization problem:

$$\begin{aligned} & \underset{\mathbf{q}}{\text{minimize}} && J(\mathbf{q}) := \frac{\frac{1}{2} \mathbf{q}^\top \mathbf{A} \mathbf{q} + B}{\boldsymbol{\pi}^\top \mathbf{q}} + C; \\ & \text{subject to} && \mathbf{q} \geq 0, \\ & && \boldsymbol{\pi}^\top \mathbf{q} > 0, \\ & && \|\mathbf{q}\|_1 = Q. \end{aligned} \quad (\text{D.200})$$

**Remark 9.** In Problem (D.200), when setting some  $q_k$  to zero, we do not consider the possibility of redefining the Markov chain  $(\mathcal{A}_t)_{t \geq 0}$  in Assumption 3 by considering the reduced state space of clients with  $q_k > 0$ . In this case, the redefined Markov chain would have a different transition matrix  $\mathbf{P}' \neq \mathbf{P}$  with  $\lambda(\mathbf{P}') \neq \lambda(\mathbf{P})$ , resulting in  $C$  no longer being constant.

### D.5.1 The optimization problem in (D.200) is convex

Let us rewrite the problem by adding a variable  $s := 1/\pi^\top \mathbf{q}$  and then replacing  $\mathbf{y} := s\mathbf{q}$ . We have:

$$J(\mathbf{y}, s) = s \left( \frac{1}{2} \frac{\mathbf{y}^\top}{s} \mathbf{A} \frac{\mathbf{y}}{s} + B \right) + C = s \cdot K \left( \frac{\mathbf{y}}{s} \right) + C, \quad (\text{D.201})$$

where  $K : \mathbb{R}^N \rightarrow \mathbb{R}$ ,  $K(\mathbf{q}) := \frac{1}{2} \mathbf{q}^\top \mathbf{A} \mathbf{q} + B$  is a (strictly) convex function, and:

$$\begin{aligned} & \underset{\mathbf{y}, s}{\text{minimize}} && J(\mathbf{y}, s) = \frac{1}{2s} \mathbf{y}^\top \mathbf{A} \mathbf{y} + Bs + C \\ & \text{subject to} && \mathbf{y} \geq 0, \\ & && s > 0, \\ & && \pi^\top \mathbf{y} = 1, \\ & && \|\mathbf{y}\|_1 = Qs. \end{aligned} \quad (\text{D.202})$$

Note that the objective function  $J(\mathbf{y}, s) : \mathbb{R}^{N+1} \rightarrow \mathbb{R}$ ,  $J(\mathbf{y}, s) = s \cdot K(\mathbf{y}/s) + C$  in (D.201) is the perspective of the convex function  $K(\mathbf{q}) + C$ , and is therefore convex [BV04, pp. 89–90]. Moreover, the constraints in (D.202) define a convex set, and then the optimization problem (D.202) is convex. We solve it with the method of Lagrange multipliers.

### D.5.2 Support for Guideline A (Section 4.4.3.2)

The Lagrangian function  $\mathcal{L}$  is as follows:

$$\mathcal{L}(\mathbf{y}, s, \iota, \theta, \boldsymbol{\omega}) = \frac{1}{2s} \mathbf{y}^\top \mathbf{A} \mathbf{y} + Bs + C + \iota(1 - \pi^\top \mathbf{y}) + \theta(\|\mathbf{y}\|_1 - Qs) - \boldsymbol{\omega}^\top \mathbf{y}. \quad (\text{D.203})$$

Since the constraint  $s > 0$  defines an open set, the set defined by the constraints in (D.202) is not closed. However, the solution of the optimization problem (D.202) is never on the boundary  $s = 0$  because  $\mathcal{L} \rightarrow +\infty$  as  $s \rightarrow 0^+$ , therefore we can consider  $s \geq 0$ . Moreover, strong duality holds for the Slater's constraint qualification for convex problems.

The KKT conditions read:

$$\frac{\partial \mathcal{L}}{\partial s}(\mathbf{y}^*, s^*, \iota^*, \theta^*, \boldsymbol{\omega}^*) = 0, \quad (\text{D.204})$$

$$\nabla_{\mathbf{y}} \mathcal{L}(\mathbf{y}^*, s^*, \iota^*, \theta^*, \boldsymbol{\omega}^*) = 0, \quad (\text{D.205})$$

$$\pi^\top \mathbf{y}^* - 1 = 0, \quad (\text{D.206})$$

$$\|\mathbf{y}^*\|_1 - Qs^* = 0, \quad (\text{D.207})$$

$$\boldsymbol{\omega}^{*\top} \mathbf{y}^* = 0, \quad (\text{D.208})$$

$$\mathbf{y}^*, \boldsymbol{\omega}^* \geq 0. \quad (\text{D.209})$$

In particular, the KKT condition for  $\mathbf{y}^*$  read:

$$\nabla_{\mathbf{y}} \mathcal{L}(\mathbf{y}^*, s^*, \iota^*, \theta^*, \boldsymbol{\omega}^*) = \frac{1}{s^*} \mathbf{A} \mathbf{y}^* - \iota^* \boldsymbol{\pi} + \theta^* \mathbf{1} - \boldsymbol{\omega}^* = 0, \quad (\text{D.210})$$

which is satisfied when:

$$\frac{\partial \mathcal{L}}{\partial y_k^*} = \frac{1}{s^*} A_{kk} y_k^* - \iota^* \pi_k + \theta^* - \omega_k^* = 0, \quad \forall k \in \mathcal{K}, \quad (\text{D.211})$$

where  $A_{ij}$  denotes the element on the  $i$ -th row and the  $j$ -th column of matrix  $\mathbf{A}$ .

Furthermore, the Complementary Slackness conditions in (D.208) and (D.209) present two cases:

1. If  $y_k^* > 0$  (and  $q_k^* > 0$ ), then  $\omega_k^* = 0$  and:

$$y_k^* = \frac{s^*}{A_{kk}}(\iota^* \pi_k - \theta^*), \quad q_k^* = \frac{1}{A_{kk}}(\iota^* \pi_k - \theta^*); \quad (\text{D.212})$$

2.  $y_k^* = q_k^* = 0$  otherwise.

By replacing the third equality constraint in Problem (D.202) with the inequality constraint  $\boldsymbol{\pi}^\top \mathbf{y} \geq 1$ , we establish an equivalent optimization problem. The equivalence holds because, for any feasible solution  $\mathbf{y}'$  with  $\boldsymbol{\pi}^\top \mathbf{y}' > 1$ , we can consider the solution  $\mathbf{y}'' = \frac{\mathbf{y}'}{\boldsymbol{\pi}^\top \mathbf{y}'} < \mathbf{y}'$ , leading to a lower objective function value. Additionally, the new problem states that the Lagrange multiplier ( $\iota^*$ ) associated with the inequality constraint must be non-negative. By considering  $A_{kk} \geq 0$  and  $\iota^* \geq 0$  in Equation (D.212), we conclude that  $q_k^*$  increases with  $\pi_k$ , providing analytical support for Guideline A.

### D.5.3 Closed-form solution of the optimization problem (D.200)

The solution of the optimization problem in (D.200) is not of practical utility because its constants (e.g.,  $L, \omega, \Gamma, C_P$ ) are in general problem-dependent and difficult to estimate during training. In particular,  $\Gamma$  poses particular difficulties as it is defined in terms of the minimizer of the target objective  $F$ , but the FL algorithm generally minimizes the biased function  $F_B$ . Nevertheless, we include the closed-form solution of the optimization problem (D.200) for completeness.

We use the active-set method: let  $\mathcal{X}$  be the set of coordinates corresponding to the active inequalities, i.e.,  $\mathcal{X} = \{k \mid y_k^* = 0\}$ .

From the KKT condition in (D.206), we derive a relation between  $\iota^*$  and  $\theta^*$ :

$$\boldsymbol{\pi}^\top \mathbf{y}^* = \sum_{k \notin \mathcal{X}} \pi_k y_k^* = \sum_{k \notin \mathcal{X}} \pi_k \frac{s^*}{A_{kk}} (\iota^* \pi_k - \theta^*) = \iota^* s^* \sum_{k \notin \mathcal{X}} \frac{\pi_k^2}{A_{kk}} - \theta^* s^* \sum_{k \notin \mathcal{X}} \frac{\pi_k}{A_{kk}} = 1. \quad (\text{D.213})$$

We use the KKT condition in (D.207) to derive another relation between  $\iota^*$  and  $\theta^*$ :

$$\|\mathbf{y}^*\|_1 = \sum_{k \notin \mathcal{X}} y_k^* = \sum_{k \notin \mathcal{X}} \frac{s^*}{A_{kk}} (\iota^* \pi_k - \theta^*) = Qs \quad \Leftrightarrow \quad \iota^* = \frac{Q + \theta^* \sum_{k \notin \mathcal{X}} \frac{1}{A_{kk}}}{\sum_{k \notin \mathcal{X}} \frac{\pi_k}{A_{kk}}}, \quad (\text{D.214})$$

and, replacing (D.214) in (D.213), we derive the closed-form solution for  $\theta^*$ :

$$\theta^* = \frac{\sum_{k \notin \mathcal{X}} \frac{\pi_k}{A_{kk}} - Qs^* \sum_{k \notin \mathcal{X}} \frac{\pi_k^2}{A_{kk}}}{s^* \left[ \left( \sum_{k \notin \mathcal{X}} \frac{1}{A_{kk}} \right) \cdot \left( \sum_{k \notin \mathcal{X}} \frac{\pi_k^2}{A_{kk}} \right) - \left( \sum_{k \notin \mathcal{X}} \frac{\pi_k}{A_{kk}} \right)^2 \right]}. \quad (\text{D.215})$$

## D.6 Background on Markov Chains

### D.6.1 Markov Chain for the Analysis (Section 4.4.3.2)

We recall some existing results [LP17; SSY18] for the Markov chain  $(\mathcal{A}_t)_{t \geq 0}$  used in our analysis (Assumption 3).

**Assumption 31.** *The Markov chain  $(\mathcal{A}_t)_{t \geq 0}$  on the  $M$ -finite state space  $\mathcal{M}$  is time-homogeneous, irreducible, and aperiodic. It has transition matrix  $\mathbf{P}$ , stationary distribution  $\boldsymbol{\rho}$ , and has state distribution  $\boldsymbol{\rho}$  at time  $t = 0$ .*

Let  $\boldsymbol{\rho}^{(t)} = [\rho_1^{(t)}, \rho_2^{(t)}, \dots, \rho_M^{(t)}]$ ,  $\sum_{i=1}^M \rho_i^{(t)} = 1$  be the state probability distribution on the Markov chain  $(\mathcal{A}_t)_{t \geq 0}$  at time step  $t$ . Assumption 31 guarantees the existence of a stationary distribution  $\boldsymbol{\rho} = \lim_{t \rightarrow +\infty} \boldsymbol{\rho}^{(t)} = [\rho_1, \rho_2, \dots, \rho_M]$  with  $\min_i \{\rho_i\} > 0$  and  $\boldsymbol{\rho}^\top \mathbf{P} = \boldsymbol{\rho}^\top$ . Then  $\boldsymbol{\rho}$  is a left eigenvector relative to the eigenvalue 1, which is the largest eigenvalue of the matrix  $\mathbf{P}$ .

For the transition matrix  $\mathbf{P}$ , we label its eigenvalues in decreasing order:

$$1 = \lambda_1(\mathbf{P}) > \lambda_2(\mathbf{P}) \geq \dots \geq \lambda_M(\mathbf{P}). \quad (\text{D.216})$$

We define:

$$\bar{\lambda}_2(\mathbf{P}) := \max \{|\lambda_2(\mathbf{P})|, |\lambda_M(\mathbf{P})|\} \quad \text{and} \quad \lambda(\mathbf{P}) := \frac{\bar{\lambda}_2(\mathbf{P}) + 1}{2}. \quad (\text{D.217})$$

The second largest absolute eigenvalue  $\bar{\lambda}_2(\mathbf{P})$  of the transition matrix  $\mathbf{P}$  characterizes the mixing time of a Markov chain. The absolute spectral gap  $\gamma := 1 - \bar{\lambda}_2(\mathbf{P})$  and its reciprocal, the relaxation time  $t_{\text{rel}} := \frac{1}{\gamma}$ , play a role in this relationship. To quantify the convergence of the Markov chain towards stationarity, we use the parameter  $d(t) := \max_{a \in \mathcal{M}} \|[P^t]_{a,\cdot} - \boldsymbol{\rho}\|_{TV}$ , which measures the maximum distance between the distribution  $[P^t]_{a,\cdot}$  and the stationary distribution  $\boldsymbol{\rho}$  for all initial states  $a \in \mathcal{M}$ . The mixing time  $t_{\text{mix}}(\varepsilon)$  is defined as the minimum time at which the distance  $d(t)$  becomes less than or equal to a given threshold  $\varepsilon$ :  $t_{\text{mix}}(\varepsilon) := \min \{t : d(t) \leq \varepsilon\}$ . Upper and lower bounds exist for the mixing time based on the relaxation time and the stationary distribution:  $(t_{\text{rel}} - 1) \log\left(\frac{1}{2\varepsilon}\right) \leq t_{\text{mix}}(\varepsilon) \leq \log\left(\frac{1}{\varepsilon \rho_{\min}}\right) t_{\text{rel}}$ , where  $\rho_{\min} := \min_{a \in \mathcal{M}} \rho_a$  [LP17, pp. 154–156].

## D.6.2 Markov Chain for Guideline B (Section 2.2.4)

In Section 2.2.3.4 (Guideline B), we examine a specific scenario where the availability of each client  $k$  follows an independent Markov chain  $(\mathcal{A}_t^k)_{t \geq 0}$  with transition probability matrix  $\mathbf{P}_k$ . This setup allows us to model the aggregate process as a product of independent Markov chains, known as a Product Chain [LP17, Section 12.4].

**Definition 10** (Product Chain). *Let  $\mathbf{P}_1$  and  $\mathbf{P}_2$  be transition matrices on state spaces  $\mathcal{M}_1$  and  $\mathcal{M}_2$  respectively, with corresponding stationary distributions  $\boldsymbol{\pi}_1$  and  $\boldsymbol{\pi}_2$ . We consider a Markov Chain on the state space  $\mathcal{M}_1 \times \mathcal{M}_2$  that moves independently in the first and second coordinates according to  $\mathbf{P}_1$  and  $\mathbf{P}_2$  respectively. The transition matrix of this Markov Chain is the Kronecker product  $\tilde{\mathbf{P}} = \mathbf{P}_1 \otimes \mathbf{P}_2$ , defined as:*

$$\tilde{\mathbf{P}}((x, y), (z, w)) = \mathbf{P}_1(x, z) \mathbf{P}_2(y, w). \quad (\text{D.218})$$

**Proposition D.24.** *The stationary distribution of the Markov chain defined by  $\tilde{\mathbf{P}} = \mathbf{P}_1 \otimes \mathbf{P}_2$  is the Kronecker product  $\tilde{\boldsymbol{\rho}} = \boldsymbol{\pi}_1 \otimes \boldsymbol{\pi}_2$ .*

*Proof.* We can observe the following:

$$\tilde{\boldsymbol{\rho}}^\top \tilde{\mathbf{P}} = (\boldsymbol{\pi}_1 \otimes \boldsymbol{\pi}_2)^\top \cdot (\mathbf{P}_1 \otimes \mathbf{P}_2) = (\boldsymbol{\pi}_1^\top \mathbf{P}_1) \otimes (\boldsymbol{\pi}_2^\top \mathbf{P}_2) = \boldsymbol{\pi}_1^\top \otimes \boldsymbol{\pi}_2^\top = \tilde{\boldsymbol{\rho}}^\top, \quad (\text{D.219})$$

where, in (D.219), we used the mixed-product property of the Kronecker product in the second step, and in the third step, we noted that  $\pi_1$  and  $\pi_2$  are the stationary distributions for  $\mathbf{P}_1$  and  $\mathbf{P}_2$ , respectively. For a comprehensive list of properties that the Kronecker product satisfies, please refer to [Mey01, p. 597].  $\square$

**Proposition D.25** ([LP17, Exercise 12.6]). *Let  $\mathbf{u}$  and  $\mathbf{v}$  be eigenvectors of  $\mathbf{P}_1$  and  $\mathbf{P}_2$ , respectively, with eigenvalues  $\lambda$  and  $\mu$ . Then  $\mathbf{u} \otimes \mathbf{v}$  is an eigenvector of  $\mathbf{P}_1 \otimes \mathbf{P}_2$  with eigenvalue  $\lambda\mu$ .*

*Proof.* We can verify the following:

$$(\mathbf{u} \otimes \mathbf{v})^\top (\mathbf{P}_1 \otimes \mathbf{P}_2) = (\mathbf{u}^\top \mathbf{P}_1) \otimes (\mathbf{v}^\top \mathbf{P}_2) = (\lambda \mathbf{u}^\top) \otimes (\mu \mathbf{v}^\top) = \lambda\mu (\mathbf{u} \otimes \mathbf{v})^\top. \quad (\text{D.220})$$

In (D.220), we used the mixed-product property and the associativity of the scalar multiplication with the Kronecker product.  $\square$

In general, let  $\mathbf{P}_1$  be a  $m \times m$  matrix with eigenvalues  $\lambda_1, \dots, \lambda_m$ , and  $\mathbf{P}_2$  be a  $n \times n$  matrix with eigenvalues  $\mu_1, \dots, \mu_n$ . The complete eigen-decomposition of  $\mathbf{P}_1 \otimes \mathbf{P}_2$  depends on the Kronecker product structure and involves combinations of the eigenvalues and eigenvectors of  $\mathbf{P}_1$  and  $\mathbf{P}_2$ .

**Proposition D.26** (Spectrum of the Kronecker product, [Mey01, Exercise 7.8.11]). *Let the eigenvalues of  $\mathbf{P}_1 \in \mathbb{R}^{m \times m}$  be denoted by  $\lambda_i$  and let the eigenvalues of  $\mathbf{P}_2 \in \mathbb{R}^{n \times n}$  be denoted by  $\mu_j$ . The eigenvalues of  $\mathbf{P}_1 \otimes \mathbf{P}_2$  are the  $mn$  numbers  $\{\lambda_i \mu_j\}_{i=1, j=1}^{m, n}$ .*

*Proof.* Let  $\mathbf{J}_1 = \mathbf{A}_1^{-1} \mathbf{P}_1 \mathbf{A}_1$  and  $\mathbf{J}_2 = \mathbf{A}_2^{-1} \mathbf{P}_2 \mathbf{A}_2$  be the respective Jordan forms for  $\mathbf{P}_1$  and  $\mathbf{P}_2$ . We use the mixed-product property and the inverse property of the Kronecker product to show that  $\mathbf{P}_1 \otimes \mathbf{P}_2$  is similar to  $\mathbf{J}_1 \otimes \mathbf{J}_2$ :

$$\mathbf{J}_1 \otimes \mathbf{J}_2 = (\mathbf{A}_1^{-1} \mathbf{P}_1 \mathbf{A}_1) \otimes (\mathbf{A}_2^{-1} \mathbf{P}_2 \mathbf{A}_2) \quad (\text{D.221})$$

$$= (\mathbf{A}_1^{-1} \otimes \mathbf{A}_2^{-1}) (\mathbf{P}_1 \otimes \mathbf{P}_2) (\mathbf{A}_1 \otimes \mathbf{A}_2) \quad (\text{D.222})$$

$$= (\mathbf{A}_1 \otimes \mathbf{A}_2)^{-1} (\mathbf{P}_1 \otimes \mathbf{P}_2) (\mathbf{A}_1 \otimes \mathbf{A}_2). \quad (\text{D.223})$$

Consequently, the eigenvalues of  $\mathbf{P}_1 \otimes \mathbf{P}_2$  coincide with those of  $\mathbf{J}_1 \otimes \mathbf{J}_2$ . Since  $\mathbf{J}_1$  and  $\mathbf{J}_2$  are upper triangular with  $\{\lambda_i\}_{i=1}^m$  and  $\{\mu_j\}_{j=1}^n$  on the diagonals, respectively,  $\mathbf{J}_1 \otimes \mathbf{J}_2$  is also upper triangular with diagonal entries given by  $\{\lambda_i \mu_j\}_{i=1, j=1}^{m, n}$ .  $\square$

**Proposition D.27.** *Let  $\bar{\lambda}_2(\mathbf{P}_k)$  denote the second largest eigenvalue in absolute value of the transition matrix  $\mathbf{P}_k$  associated with the  $k$ -th client, and define  $\lambda(\mathbf{P}_k) := \frac{\bar{\lambda}_2(\mathbf{P}_k) + 1}{2}$ . For the product chain defined by  $\mathbf{P} = \bigotimes_{k \in \mathcal{K}} \mathbf{P}_k$ , the second largest eigenvalue in absolute value  $\bar{\lambda}_2(\mathbf{P})$  and  $\lambda(\mathbf{P}) := \frac{\bar{\lambda}_2(\mathbf{P}) + 1}{2}$  satisfy:*

$$\bar{\lambda}_2(\mathbf{P}) = \max_{k \in \mathcal{K}} \bar{\lambda}_2(\mathbf{P}_k) \quad \text{and} \quad \lambda(\mathbf{P}) = \max_{k \in \mathcal{K}} \lambda(\mathbf{P}_k). \quad (\text{D.224})$$

The proof of Proposition D.27 follows a similar structure to the one in [LP17, Corollary 12.13].

*Proof.* From Proposition D.26, we know that the eigenvalues of  $\mathbf{P} = \bigotimes_{k \in \mathcal{K}} \mathbf{P}_k$  are given by:

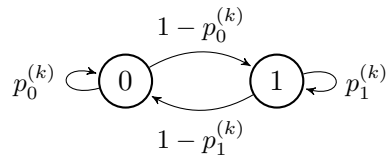
$$\left\{ \prod_{k \in \mathcal{K}} \lambda_i(\mathbf{P}_k) : \lambda_i(\mathbf{P}_k) \text{ an eigenvalue of } \mathbf{P}_k \right\}. \quad (\text{D.225})$$



Recall that  $\bar{\lambda}_2(\mathbf{P}_k)$  is the second largest eigenvalue of  $\mathbf{P}_k$  in absolute value. If  $k^*$  denotes the index such that  $\bar{\lambda}_2(\mathbf{P}_{k^*}) = \max_{k \in \mathcal{K}} \bar{\lambda}_2(\mathbf{P}_k)$ , the second largest eigenvalue in module of  $\mathbf{P}$  is the product of  $\bar{\lambda}_2(\mathbf{P}_{k^*})$  for the  $k^*$ -th client and  $\lambda_1(\mathbf{P}_j) = 1$  for the remaining clients  $j \neq k^*$ . The second result in (D.224) follows from the definitions of  $\lambda(\mathbf{P})$  and  $\lambda(\mathbf{P}_k)$ .  $\square$

### D.6.3 Markov Chain for the Experiments (Section 2.2.6)

In the experiments (Section 2.2.6.1), we consider a scenario where the activity of each client  $k \in \mathcal{K}$  follows a two-state homogeneous Markov process. The state space  $\mathcal{M}$  consists of two states: “inactive” (with value 0) and “active” (with value 1):



We provide detailed expressions of the transition matrix  $\mathbf{P}_k$ , stationary distribution  $\pi^{(k)}$ , and the second eigenvalue  $\lambda_2(\mathbf{P}_k)$  used in the experiments for each client  $k \in \mathcal{K}$ :

$$\mathbf{P}_k = \begin{bmatrix} p_0^{(k)} & 1 - p_0^{(k)} \\ 1 - p_1^{(k)} & p_1^{(k)} \end{bmatrix} = \begin{bmatrix} 1 - (1 - \lambda_2(\mathbf{P}_k))\pi_k & (1 - \lambda_2(\mathbf{P}_k))\pi_k \\ (1 - \lambda_2(\mathbf{P}_k))(1 - \pi_k) & \lambda_2(\mathbf{P}_k) + (1 - \lambda_2(\mathbf{P}_k))\pi_k \end{bmatrix}. \quad (\text{D.226})$$

$$\pi^{(k)} = [1 - \pi_k, \pi_k] = \left[ \frac{1 - p_1^{(k)}}{2 - p_0^{(k)} - p_1^{(k)}}, \frac{1 - p_0^{(k)}}{2 - p_0^{(k)} - p_1^{(k)}} \right]. \quad (\text{D.227})$$

$$\lambda_2(\mathbf{P}_k) = p_0^{(k)} + p_1^{(k)} - 1. \quad (\text{D.228})$$

## D.7 Details on Experimental Setup

**Datasets and Models** In this section, we provide a detailed description of the datasets and models used in our experiments. We considered a total of  $N = 100$  clients. We tested CA-Fed on the benchmark synthetic LEAF dataset [Cal+19] for regularized logistic regression tasks, which satisfy Assumptions 5-6. Additionally, we incorporated two “real-world” datasets: MNIST [LC10] for handwritten digit recognition and CIFAR-10 [Kri09] for image recognition. Detailed descriptions of the datasets and the models used for each of them are provided below.

**Synthetic LEAF dataset** Synthetic data provides us with precise control over heterogeneity. The Synthetic LEAF dataset achieves this by using parameters  $\gamma$  and  $\delta$ , where  $\gamma$  determines the degree of variation among local models and  $\delta$  determines the variability in the local data across different devices. The generation process follows the setup described in [Li+20a; Li+19]:

1. For each client  $k \in \mathcal{K}$ , sample the model parameters  $\mathbf{W}_k \in \mathbb{R}^{10 \times 60}$  and  $\mathbf{b}_k \in \mathbb{R}^{10}$  from a normal distribution with mean  $\mu_k$  and standard deviation 1, where  $\mu_k$  is sampled from  $\mathcal{N}(0, \gamma)$ .

Table 3: Average computation time and used CPU/GPU for each dataset.

Dataset	CPU/GPU	Simulation time
Binary Synthetic	Intel(R) Xeon(R) CPU	10min
Synthetic LEAF	Intel(R) Xeon(R) CPU	6min
MNIST [LC10]	GeForce GTX 1080 Ti	42min
CIFAR10 [Kri09]	GeForce GTX 1080 Ti	2h37min

Table 4: Learning rates  $\eta$  and  $\bar{\eta}$  used for the experiments in Figure 2.11.

Dataset	Unbiased	More available	CA-Fed ( $\bar{\kappa} = 1$ )	AdaFed [Tan+22a]	F3AST [RVd23]
Synthetic LEAF	2.0/2.0	1.0/7.0	2.0/3.0	1.0/1.0	2.0/2.0
MNIST	0.03/1.0	0.1/4.0	0.1/1.0	0.03/1.0	0.1/0.3
CIFAR10	0.03/1.0	0.03/3.0	0.03/1.0	0.03/1.0	0.03/0.3

- For each client  $k \in \mathcal{K}$ , generate the client’s input data  $\mathbf{X}_k \in \mathbb{R}^{n_k \times 60}$  as follows: sample each element  $(x_k)_j$  from a normal distribution with mean  $v_k$  and standard deviation  $\frac{1}{j^{1.2}}$ , where  $v_k$  is sampled from  $\mathcal{N}(B_k, 1)$  and  $B_k$  is sampled from  $\mathcal{N}(0, \delta)$ .
- Generate synthetic samples  $(\mathbf{X}_k, \mathbf{Y}_k)$ , where  $\mathbf{Y}_k \in \mathbb{R}^{n_k}$ , according to the model  $y = \arg \max(\text{softmax}(\mathbf{W}_k \mathbf{x} + \mathbf{b}_k))$ , where  $\mathbf{x} \in \mathbb{R}^{60}$ .

The distribution of samples  $n_k = |D_k|$  among the clients follows a power law, resulting in an imbalanced data distribution. We refer to the synthetic dataset with parameters  $\gamma$  and  $\delta$  as  $\text{synthetic}(\gamma, \delta)$ . We set  $(\gamma, \delta)$  values to  $(0, 0)$ ,  $(0.25, 0.25)$ ,  $(0.5, 0.5)$ ,  $(0.75, 0.75)$ , and  $(1, 1)$  to investigate various levels of heterogeneity in the data.

**MNIST** To classify handwritten digits in the MNIST dataset, we employ multinomial logistic regression. The model takes a flattened 784-dimensional ( $28 \times 28$ ) image as input and predicts a class label from 0 to 9 as output. To introduce heterogeneity in the data distribution, we distribute the dataset among  $N = 100$  clients using a Dirichlet allocation method [Wan+20a] with parameter  $\varsigma$ . This allocation scheme allows for varying proportions of the dataset to be assigned to each client, contributing to the heterogeneous nature of our experimental setting.

**CIFAR-10** The CIFAR-10 dataset consists of 60,000 input images, sourced from a collection of 80 million tiny images, with 10 distinct labels. To partition the CIFAR-10 dataset among  $N = 100$  clients, we employ a Dirichlet allocation [Wan+20a] with parameter  $\varsigma$ . For this particular dataset, we train a shallow neural network comprising two convolutional layers followed by one fully connected layer. This network architecture is designed to capture relevant features from the CIFAR-10 images and facilitate accurate classification.

### D.7.1 Implementation Details

**Machines** The experiments were conducted on a CPU/GPU cluster, utilizing various available GPUs such as Nvidia Tesla V100, GeForce GTX 1080 Ti, and Quadro RTX 8000. The majority of experiments involving Synthetic datasets were executed on an Intel(R) Xeon(R) CPU E5-1660 v3 @ 3.00GHz. On the other hand, experiments involving MNIST and CIFAR-10 datasets were performed using GeForce GTX 1080 Ti cards. For each dataset, we conducted approximately 50 experiments, excluding the time dedicated to development and debugging. Due to the usage of a train batch size of 32 samples, the experiments with MNIST and CIFAR-10 datasets exhibited slower execution times. Table 3 provides the average duration required to execute one simulation for each dataset. The authors are grateful to the OPAL infrastructure from Université Côte d’Azur for providing resources and support.

**Libraries** We extensively employed the PyTorch deep learning framework throughout our experiments. PyTorch provided us with a comprehensive set of tools and functionalities for model construction, training, and evaluation. It allowed us to efficiently implement and optimize various neural network architectures, including the multinomial logistic regression model for the MNIST dataset and the shallow neural network for the CIFAR-10 dataset. To simplify the data preparation process, we utilized Torchvision, a PyTorch package designed for computer vision tasks. Torchvision facilitated seamless dataset management, including the download and pre-processing of MNIST and CIFAR-10, enabling us to transform the raw image data into a suitable format for training and evaluation.

**Hyper-parameters** For each method and task, we performed a grid search to determine the optimal learning rates  $\eta$  and  $\bar{\eta}$ . For the MNIST and CIFAR-10 datasets, we explored the grids  $\eta = \{2.0, 1.0, 0.3, 0.1, 0.03, 0.01\}$  and  $\bar{\eta} = \{5.0, 4.0, 3.0, 2.0, 1.0, 0.3, 0.1\}$ . For the Synthetic LEAF dataset, we shifted the grid to  $\bar{\eta} = \{8.0, 7.0, 6.0, 5.0, 4.0, 3.0, 2.0, 1.0\}$ . Table 4 reports the learning rates  $\eta$  and  $\bar{\eta}$  corresponding to the results in Figure 2.11 for each dataset and method. For CA-Fed, we use the hyper-parameters  $\beta = \tau = 0$ . In the case of AdaFed, we set full device participation, where the parameter server samples all active clients ( $|\mathcal{S}_t| = |\mathcal{A}_t|$ ). To ensure a fair comparison, we set the number of clients sampled by F3AST to the average number of clients included by CA-Fed, which is 45 on average. Furthermore, we set the smoothness parameter  $\beta$  of F3AST to be  $\mathcal{O}(1/T)$ , as suggested by the authors in [RVd23, Appendix D].

## E Personalized Federated Learning under a Mixture of Distributions

### E.1 Proof of Proposition 3.5.1

For  $h \in \mathcal{H}$  and  $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ , let  $p_h(y|\mathbf{x})$  denote the conditional probability distribution of  $y$  given  $\mathbf{x}$  under model  $h$ , i.e.,

$$p_h(y|\mathbf{x}) \triangleq e^{c_h(\mathbf{x})} \times \exp \left\{ -l(h(\mathbf{x}), y) \right\}, \quad (\text{E.229})$$

where

$$c_h(\mathbf{x}) \triangleq -\log \left[ \int_{y \in \mathcal{Y}} \exp \left\{ -l(h(\mathbf{x}), y) \right\} \mathrm{d}y \right]. \quad (\text{E.230})$$

We also remind that the entropy of a probability distribution  $q$  over  $\mathcal{Y}$  is given by

$$H(q) \triangleq - \int_{y \in \mathcal{Y}} q(y) \cdot \log q(y) \mathrm{d}y, \quad (\text{E.231})$$

and that the Kullback-Leibler divergence between two probability distributions  $q_1$  and  $q_2$  over  $\mathcal{Y}$  is given by

$$\mathcal{KL}(q_1||q_2) \triangleq \int_{y \in \mathcal{Y}} q_1(y) \cdot \log \frac{q_1(y)}{q_2(y)} \mathrm{d}y. \quad (\text{E.232})$$

**Proposition 3.5.1.** *Let  $l(\cdot, \cdot)$  be the mean squared error loss, the logistic loss or the cross-entropy loss, and  $\check{\Theta}$  and  $\check{\Pi}$  be a solution of the following optimization problem:*

$$\underset{\Theta, \Pi}{\text{minimize}} \mathbb{E}_{t \sim D_{\mathcal{T}}} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} [-\log p_t(\mathbf{x}, y | \Theta, \pi_t)], \quad (3.4)$$

where  $D_{\mathcal{T}}$  is any distribution with support  $\mathcal{T}$ . Under Assumptions 8, 9, and 10, the predictors

$$h_t^* = \sum_{m=1}^M \check{\pi}_{tm} h_{\check{\theta}_m}, \quad \forall t \in \mathcal{T} \quad (3.5)$$

minimize  $\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} [l(h_t(\mathbf{x}), y)]$  and thus solve Problem (3.1).

*Proof.* We prove the result for each of the three possible cases of the loss function. We verify that  $c_h$  does not depend on  $h$  in each of the three cases, then we use Lemma E.3 to conclude.

**Mean Squared Error Loss** This is the case of a regression problem where  $\mathcal{Y} = \mathbb{R}^d$  for some  $d > 0$ . For  $\mathbf{x}, y \in \mathcal{X} \times \mathcal{Y}$  and  $h \in \mathcal{H}$ , we have

$$p_h(y|\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d}} \cdot \exp \left\{ -\frac{\|h(\mathbf{x}) - y\|^2}{2} \right\}, \quad (\text{E.233})$$

and

$$c_h(\mathbf{x}) = -\log \left( \sqrt{(2\pi)^d} \right) \quad (\text{E.234})$$

**Logistic Loss** This is the case of a binary classification problem where  $\mathcal{Y} = \{0, 1\}$ . For  $\mathbf{x}, y \in \mathcal{X} \times \mathcal{Y}$  and  $h \in \mathcal{H}$ , we have

$$p_h(y|\mathbf{x}) = (h(\mathbf{x}))^y \cdot (1 - h(\mathbf{x}))^{1-y}, \quad (\text{E.235})$$

and

$$c_h(\mathbf{x}) = 0 \quad (\text{E.236})$$

**Cross-entropy loss** This is the case of a classification problem where  $\mathcal{Y} = [L]$  for some  $L > 1$ . For  $\mathbf{x}, y \in \mathcal{X} \times \mathcal{Y}$  and  $h \in \mathcal{H}$ , we have

$$p_h(y|\mathbf{x}) = \prod_{l=1}^L (h(\mathbf{x}))^{1_{\{y=l\}}}, \quad (\text{E.237})$$

and

$$c_h(\mathbf{x}) = 0 \quad (\text{E.238})$$

**Conclusion** For  $t \in \mathcal{T}$ , consider a predictor  $h_t^*$  minimizing  $\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} [l(h_t(\mathbf{x}), y)]$ . Using Lemma E.3, for  $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ , we have

$$p_{h_t^*}(y|\mathbf{x}) = \sum_{m=1}^M \check{\pi}_{tm} \cdot p_m(y|\mathbf{x}, \check{\theta}_m). \quad (\text{E.239})$$

We multiply both sides of this equality by  $y$  and we integrate over  $y \in \mathcal{Y}$ . Note that in all three cases we have

$$\forall \mathbf{x} \in \mathcal{X}, \quad \int_{y \in \mathcal{Y}} y \cdot p_h(\cdot|\mathbf{x}) \, dy = h(\mathbf{x}). \quad (\text{E.240})$$

It follows that

$$h_t^* = \sum_{m=1}^M \check{\pi}_{tm} h_{\check{\theta}_m}, \quad \forall t \in \mathcal{T}. \quad (\text{E.241})$$

□

## Supporting Lemmas

**Lemma E.1.** Suppose that Assumptions 8 and 10 hold, and consider  $\check{\Theta}$  and  $\check{\Pi}$  to be a solution of Problem (3.4). Then

$$p_t(\mathbf{x}, y|\check{\Theta}, \check{\pi}_t) = p_t(\mathbf{x}, y|\Theta^*, \pi_t^*), \quad \forall t \in \mathcal{T}. \quad (\text{E.242})$$

*Proof.* For  $t \in \mathcal{T}$ ,

$$\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} \left[ -\log p_t(\mathbf{x}, y|\check{\Theta}, \check{\pi}_t) \right] \quad (\text{E.243})$$

$$= - \int_{(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}} p_t(\mathbf{x}, y|\Theta^*, \pi_t^*) \cdot \log p_t(\mathbf{x}, y|\check{\Theta}, \check{\pi}_t) \, d\mathbf{x} \, dy \quad (\text{E.244})$$

$$= - \int_{(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}} p_t(\mathbf{x}, y|\Theta^*, \pi_t^*) \cdot \log \frac{p_t(\mathbf{x}, y|\check{\Theta}, \check{\pi}_t)}{p_t(\mathbf{x}, y|\Theta^*, \pi_t^*)} \, d\mathbf{x} \, dy$$

$$- \int_{(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}} p_t(\mathbf{x}, y | \Theta^*, \pi_t^*) \cdot \log p_t(\mathbf{x}, y | \Theta^*, \pi_t^*) \, d\mathbf{x} \, dy \quad (\text{E.245})$$

$$= \mathcal{KL} \left( p_t(\cdot | \Theta^*, \pi_t^*) \| p_t(\cdot | \check{\Theta}, \check{\pi}_t) \right) + H [p_t(\cdot | \Theta^*, \pi_t^*)], \quad (\text{E.246})$$

Since the  $\mathcal{KL}$  divergence is non-negative, we have

$$\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} \left[ -\log p_t(\mathbf{x}, y | \check{\Theta}, \check{\pi}_t) \right] \geq H [p_t(\cdot | \Theta^*, \pi_t^*)] = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} \left[ -\log p_t(\mathbf{x}, y | \Theta^*, \pi_t^*) \right]. \quad (\text{E.247})$$

Taking the expectation over  $t \sim \mathcal{D}_{\mathcal{T}}$ , we write

$$\mathbb{E}_{t \sim \mathcal{D}_{\mathcal{T}}} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} \left[ -\log p_t(\mathbf{x}, y | \check{\Theta}, \check{\pi}_t) \right] \geq \mathbb{E}_{t \sim \mathcal{D}_{\mathcal{T}}} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} \left[ -\log p_t(\mathbf{x}, y | \Theta^*, \pi_t^*) \right]. \quad (\text{E.248})$$

Since  $\check{\Theta}$  and  $\check{\pi}$  is a solution of Problem (3.4), we also have

$$\mathbb{E}_{t \sim \mathcal{D}_{\mathcal{T}}} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} \left[ -\log p_t(\mathbf{x}, y | \check{\Theta}, \check{\pi}_t) \right] \leq \mathbb{E}_{t \sim \mathcal{D}_{\mathcal{T}}} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} \left[ -\log p_t(\mathbf{x}, y | \Theta^*, \pi_t^*) \right]. \quad (\text{E.249})$$

Combining (E.248), (E.249), and (E.246), we have

$$\mathbb{E}_{t \sim \mathcal{D}_{\mathcal{T}}} \mathcal{KL} \left( p_t(\cdot | \Theta^*, \pi_t^*) \| p_t(\cdot | \check{\Theta}, \check{\pi}_t) \right) = 0. \quad (\text{E.250})$$

Since  $\mathcal{KL}$  divergence is non-negative, and the support of  $\mathcal{D}_{\mathcal{T}}$  is the countable set  $\mathcal{T}$ , it follows that

$$\forall t \in \mathcal{T}, \quad \mathcal{KL} \left( p_t(\cdot | \Theta^*, \pi_t^*) \| p_t(\cdot | \check{\Theta}, \check{\pi}_t) \right) = 0. \quad (\text{E.251})$$

Thus,

$$p_t(\mathbf{x}, y | \check{\Theta}, \check{\pi}_t) = p_t(\mathbf{x}, y | \Theta^*, \pi_t^*), \quad \forall t \in \mathcal{T}. \quad (\text{E.252})$$

□

**Lemma E.2.** Consider  $M$  probability distributions on  $\mathcal{Y}$ , that we denote  $q_m$ ,  $m \in [M]$ , and  $\alpha = (\alpha_1, \dots, \alpha_m) \in \Delta^M$ . For any probability distribution  $q$  over  $\mathcal{Y}$ , we have

$$\sum_{m=1}^M \alpha_m \cdot \mathcal{KL} \left( q_m \| \sum_{m'=1}^M \alpha_{m'} \cdot q_{m'} \right) \leq \sum_{m=1}^M \alpha_m \cdot \mathcal{KL} (q_m \| q), \quad (\text{E.253})$$

with equality if and only if,

$$q = \sum_{m=1}^M \alpha_m \cdot q_m. \quad (\text{E.254})$$

*Proof.*

$$\begin{aligned} & \sum_{m=1}^M \alpha_m \cdot \mathcal{KL} (q_m \| q) - \sum_{m=1}^M \alpha_m \cdot \mathcal{KL} \left( q_m \| \sum_{m'=1}^M \alpha_{m'} \cdot q_{m'} \right) \\ &= \sum_{m=1}^M \alpha_m \cdot \left[ \mathcal{KL} (q_m \| q) - \mathcal{KL} \left( q_m \| \sum_{m'=1}^M \alpha_{m'} \cdot q_{m'} \right) \right] \end{aligned} \quad (\text{E.255})$$

$$= - \sum_{m=1}^M \alpha_m \int_{y \in \mathcal{Y}} q_m(y) \cdot \log \left( \frac{q(y)}{\sum_{m'=1}^M \alpha_{m'} \cdot q_{m'}(y)} \right) \quad (\text{E.256})$$

$$= - \int_{y \in \mathcal{Y}} \left\{ \sum_{m=1}^M \alpha_m \cdot q_m(y) \right\} \cdot \log \left( \frac{q(y)}{\sum_{m'=1}^M \alpha_{m'} \cdot q_{m'}(y)} \right) \mathrm{d}y \quad (\text{E.257})$$

$$= \mathcal{KL} \left( \sum_{m=1}^M \alpha_m \cdot q_m \| q \right) \geq 0. \quad (\text{E.258})$$

The equality holds, if and only if,

$$q = \sum_{m=1}^M \alpha_m \cdot q_m. \quad (\text{E.259})$$

□

**Lemma E.3.** Consider  $\check{\Theta}$  and  $\check{\Pi}$  to be a solution of Problem (3.4). Under Assumptions 8, 9, and 10, if  $c_h$  does not depend on  $h \in \mathcal{H}$ , then the predictors  $h_t^*$ ,  $t \in \mathcal{T}$ , minimizing  $\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} [l(h_t(\mathbf{x}), y)]$ , verify for  $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$

$$p_{h_t^*}(y|\mathbf{x}) = \sum_{m=1}^M \check{\pi}_{tm} \cdot p_m(y|\mathbf{x}, \check{\theta}_m). \quad (\text{E.260})$$

*Proof.* For  $t \in \mathcal{T}$  and  $h_t \in \mathcal{H}$ , under Assumptions 8, 9, and 10, we have

$$\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} [l(h_t(\mathbf{x}), y)] = \int_{\mathbf{x}, y \in \mathcal{X} \times \mathcal{Y}} l(h_t(\mathbf{x}), y) \cdot p_t(\mathbf{x}, y | \Theta^*, \pi_t^*) \mathrm{d}\mathbf{x} \mathrm{d}y. \quad (\text{E.261})$$

Using Lemma E.1, it follows that

$$\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} [l(h_t(\mathbf{x}), y)] = \int_{\mathbf{x}, y \in \mathcal{X} \times \mathcal{Y}} l(h_t(\mathbf{x}), y) \cdot p_t(\mathbf{x}, y | \check{\Theta}, \check{\pi}_t) \mathrm{d}\mathbf{x} \mathrm{d}y. \quad (\text{E.262})$$

Thus, using Assumptions 8 and 9 we have,

$$\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} [l(h_t(\mathbf{x}), y)] \quad (\text{E.263})$$

$$= \int_{\mathbf{x}, y \in \mathcal{X} \times \mathcal{Y}} l(h_t(\mathbf{x}), y) \cdot p_t(\mathbf{x}, y | \check{\Theta}, \check{\pi}_t) \mathrm{d}\mathbf{x} \mathrm{d}y \quad (\text{E.264})$$

$$= \int_{\mathbf{x}, y \in \mathcal{X} \times \mathcal{Y}} l(h_t(\mathbf{x}), y) \cdot \left( \sum_{m=1}^M \check{\pi}_{tm} \cdot p_m(y|\mathbf{x}, \check{\theta}_m) \right) p(\mathbf{x}) \mathrm{d}\mathbf{x} \mathrm{d}y \quad (\text{E.265})$$

$$= \int_{\mathbf{x} \in \mathcal{X}} \left[ \sum_{m=1}^M \check{\pi}_{tm} \int_{y \in \mathcal{Y}} l(h_t(\mathbf{x}), y) \cdot p_m(y|\mathbf{x}, \check{\theta}_m) \mathrm{d}y \right] p(\mathbf{x}) \mathrm{d}\mathbf{x} \quad (\text{E.266})$$

$$= \int_{\mathbf{x} \in \mathcal{X}} \left[ \sum_{m=1}^M \check{\pi}_{tm} \left\{ c_{h_t}(\mathbf{x}) - \int_{y \in \mathcal{Y}} p_m(y|\mathbf{x}, \check{\theta}_m) \log p_{h_t}(y|\mathbf{x}) \mathrm{d}y \right\} \right] p(\mathbf{x}) \mathrm{d}\mathbf{x} \quad (\text{E.267})$$

$$= \int_{\mathbf{x} \in \mathcal{X}} \left[ c_{h_t}(\mathbf{x}) - \sum_{m=1}^M \check{\pi}_{tm} \int_{y \in \mathcal{Y}} p_m(y|\mathbf{x}, \check{\theta}_m) \log p_{h_t}(y|\mathbf{x}) \mathrm{d}y \right] p(\mathbf{x}) \mathrm{d}\mathbf{x} \quad (\text{E.268})$$

$$= \int_{\mathbf{x} \in \mathcal{X}} \left[ c_{h_t}(\mathbf{x}) + \sum_{m=1}^M \check{\pi}_{tm} \cdot H(p_m(\cdot|\mathbf{x}, \check{\theta}_m)) \right] p(\mathbf{x}) \mathrm{d}\mathbf{x} \\ + \int_{\mathbf{x} \in \mathcal{X}} \left[ \sum_{m=1}^M \check{\pi}_{tm} \cdot \mathcal{KL}(p_m(\cdot|\mathbf{x}, \check{\theta}_m) \| p_{h_t}(\cdot|\mathbf{x})) \right] p(\mathbf{x}) \mathrm{d}\mathbf{x}. \quad (\text{E.269})$$

Let  $h_t^\circ$  be a predictor satisfying the following equality:

$$p_{h_t^\circ}(y|\mathbf{x}) = \sum_{m=1}^M \check{\pi}_{tm} \cdot p_m(y|\mathbf{x}, \check{\theta}_m).$$

Using Lemma E.2, we have

$$\sum_{m=1}^M \check{\pi}_{tm} \cdot \mathcal{KL}\left(p_m(\cdot|\mathbf{x}, \check{\theta}_m) \| p_{h_t}(\cdot|\mathbf{x})\right) \geq \sum_{m=1}^M \check{\pi}_{tm} \cdot \mathcal{KL}\left(p_m(\cdot|\mathbf{x}, \check{\theta}_m) \| p_{h_t^\circ}(\cdot|\mathbf{x})\right) \quad (\text{E.270})$$

with equality if and only if

$$p_{h_t}(\cdot|\mathbf{x}) = p_{h_t^\circ}(\cdot|\mathbf{x}). \quad (\text{E.271})$$

Since  $c_h$  does not depend on  $h$ , replacing (E.270) in (E.269), it follows that

$$\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} [l(h_t(\mathbf{x}), y)] \geq \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} [l(h_t^\circ(\mathbf{x}), y)]. \quad (\text{E.272})$$

This inequality holds for any predictor  $h_t$  and in particular for  $h_t^* \in \arg \min \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} [l(h_t(\mathbf{x}), y)]$ , for which it also holds the opposite inequality, then:

$$\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} [l(h_t^*(\mathbf{x}), y)] = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t} [l(h_t^\circ(\mathbf{x}), y)], \quad (\text{E.273})$$

and the equality implies that

$$p_{h_t^*}(\cdot|\mathbf{x}) = p_{h_t^\circ}(\cdot|\mathbf{x}) = \sum_{m=1}^M \check{\pi}_{tm} \cdot p_m(\cdot|\mathbf{x}, \check{\theta}_m). \quad (\text{E.274})$$

□

## E.2 Proofs for Centralized Expectation Maximization

**Proposition 3.5.2.** *Under Assumptions 8 and 9, at the  $k$ -th iteration the EM algorithm updates parameter estimates through the following steps:*

$$\mathbf{E}\text{-step:} \quad q_t^{k+1}(z_t^{(i)} = m) \propto \pi_{tm}^k \cdot \exp\left(-l(h_{\theta_m^k}(\mathbf{x}_t^{(i)}), y_t^{(i)})\right), \quad t \in [T], m \in [M], i \in [n_t] \quad (3.12)$$

$$\mathbf{M}\text{-step:} \quad \pi_{tm}^{k+1} = \frac{\sum_{i=1}^{n_t} q_t^{k+1}(z_t^{(i)} = m)}{n_t}, \quad t \in [T], m \in [M] \quad (3.13)$$

$$\theta_m^{k+1} \in \arg \min_{\theta \in \mathbb{R}^d} \sum_{t=1}^T \sum_{i=1}^{n_t} q_t^{k+1}(z_t^{(i)} = m) l(h_\theta(\mathbf{x}_t^{(i)}), y_t^{(i)}), \quad m \in [M] \quad (3.14)$$



*Proof.* The objective is to learn parameters  $\{\check{\Theta}, \check{\Pi}\}$  from the data  $\mathcal{S}_{1:T}$  by maximizing the likelihood  $p(\mathcal{S}_{1:T}|\Theta, \Pi)$ . We introduce functions  $q_t(z)$ ,  $t \in [T]$  such that  $q_t \geq 0$  and  $\sum_{z=1}^M q_t(z) = 1$  in the expression of the likelihood. For  $\Theta \in \mathbb{R}^{M \times d}$  and  $\Pi \in \Delta^{T \times M}$ , we have

$$\log p(\mathcal{S}_{1:T}|\Theta, \Pi) = \sum_{t=1}^T \sum_{i=1}^{n_t} \log p_t(s_t^{(i)}|\Theta, \pi_t) \quad (\text{E.275})$$

$$= \sum_{t=1}^T \sum_{i=1}^{n_t} \log \left[ \sum_{m=1}^M \left( \frac{p_t(s_t^{(i)}, z_t^{(i)} = m|\Theta, \pi_t)}{q_t(z_t^{(i)} = m)} \right) q_t(z_t^{(i)} = m) \right] \quad (\text{E.276})$$

$$\geq \sum_{t=1}^T \sum_{i=1}^{n_t} \sum_{m=1}^M q_t(z_t^{(i)} = m) \log \frac{p_t(s_t^{(i)}, z_t^{(i)} = m|\Theta, \pi_t)}{q_t(z_t^{(i)} = m)} \quad (\text{E.277})$$

$$\begin{aligned} &= \sum_{t=1}^T \sum_{i=1}^{n_t} \sum_{m=1}^M q_t(z_t^{(i)} = m) \log p_t(s_t^{(i)}, z_t^{(i)} = m|\Theta, \pi_t) \\ &\quad - \sum_{t=1}^T \sum_{i=1}^{n_t} \sum_{m=1}^M q_t(z_t^{(i)} = m) \log q_t(z_t^{(i)} = m) \end{aligned} \quad (\text{E.278})$$

$$\triangleq \mathfrak{L}(\Theta, \Pi, Q_{1:T}), \quad (\text{E.279})$$

where we used Jensen's inequality because  $\log$  is concave.  $\mathfrak{L}(\Theta, \Pi, Q_{1:T})$  is an *evidence lower bound*. The centralized EM-algorithm corresponds to iteratively maximizing this bound with respect to  $Q_{1:T}$  (E-step) and with respect to  $\{\Theta, \Pi\}$  (M-step).

**E-step.** The difference between the log-likelihood and the evidence lower bound  $\mathfrak{L}(\Theta, \Pi, Q_{1:T})$  can be expressed in terms of a sum of  $\mathcal{KL}$  divergences:

$$\begin{aligned} &\log p(\mathcal{S}_{1:T}|\Theta, \Pi) - \mathfrak{L}(\Theta, \Pi, Q_{1:T}) = \\ &= \sum_{t=1}^T \sum_{i=1}^{n_t} \left\{ \log p_t(s_t^{(i)}|\Theta, \pi_t) - \sum_{m=1}^M q_t(z_t^{(i)} = m) \log \frac{p_t(s_t^{(i)}, z_t^{(i)} = m|\Theta, \pi_t)}{q_t(z_t^{(i)} = m)} \right\} \end{aligned} \quad (\text{E.280})$$

$$= \sum_{t=1}^T \sum_{i=1}^{n_t} \sum_{m=1}^M q_t(z_t^{(i)} = m) \left( \log p_t(s_t^{(i)}|\Theta, \pi_t) - \log \frac{p_t(s_t^{(i)}, z_t^{(i)} = m|\Theta, \pi_t)}{q_t(z_t^{(i)} = m)} \right) \quad (\text{E.281})$$

$$= \sum_{t=1}^T \sum_{i=1}^{n_t} \sum_{m=1}^M q_t(z_t^{(i)} = m) \log \frac{p_t(s_t^{(i)}|\Theta, \pi_t) \cdot q_t(z_t^{(i)} = m)}{p_t(s_t^{(i)}, z_t^{(i)} = m|\Theta, \pi_t)} \quad (\text{E.282})$$

$$= \sum_{t=1}^T \sum_{i=1}^{n_t} \sum_{m=1}^M q_t(z_t^{(i)} = m) \log \frac{q_t(z_t^{(i)} = m)}{p_t(z_t^{(i)} = m|s_t^{(i)}, \Theta, \pi_t)} \quad (\text{E.283})$$

$$= \sum_{t=1}^T \sum_{i=1}^{n_t} \mathcal{KL}(q_t(z_t^{(i)}) || p_t(z_t^{(i)}|s_t^{(i)}, \Theta, \pi_t)) \geq 0. \quad (\text{E.284})$$

For fixed parameters  $\{\Theta, \Pi\}$ , the maximum of  $\mathcal{L}(\Theta, \Pi, Q_{1:T})$  is reached when

$$\sum_{t=1}^T \sum_{i=1}^{n_t} \mathcal{KL} \left( q_t \left( z_t^{(i)} \right) \parallel p_t \left( z_t^{(i)} \mid s_t^{(i)}, \Theta, \pi_t \right) \right) = 0.$$

Thus for  $t \in [T]$  and  $i \in [n_t]$ , we have:

$$q_t(z_t^{(i)} = m) = p_t(z_t^{(i)} = m \mid s_t^{(i)}, \Theta, \pi_t) \quad (\text{E.285})$$

$$= \frac{p_t(s_t^{(i)} \mid z_t^{(i)} = m, \Theta, \pi_t) \times p_t(z_t^{(i)} = m \mid \Theta, \pi_t)}{p_t(s_t^{(i)} \mid \Theta, \pi_t)} \quad (\text{E.286})$$

$$= \frac{p_m(s_t^{(i)} \mid \theta_m) \times \pi_{tm}}{\sum_{m'=1}^M p_{m'}(s_t^{(i)}) \times \pi_{tm'}} \quad (\text{E.287})$$

$$= \frac{p_m(y_t^{(i)} \mid \mathbf{x}_t^{(i)}, \theta_m) \times p_m(\mathbf{x}_t^{(i)}) \times \pi_{tm}}{\sum_{m'=1}^M p_{m'}(y_t^{(i)} \mid \mathbf{x}_t^{(i)}, \theta_{m'}) \times p_{m'}(\mathbf{x}_t^{(i)}) \times \pi_{tm'}} \quad (\text{E.288})$$

$$= \frac{p_m(y_t^{(i)} \mid \mathbf{x}_t^{(i)}, \theta_m) \times p(\mathbf{x}_t^{(i)}) \times \pi_{tm}}{\sum_{m'=1}^M p_{m'}(y_t^{(i)} \mid \mathbf{x}_t^{(i)}, \theta_{m'}) \times p(\mathbf{x}_t^{(i)}) \times \pi_{tm'}}, \quad (\text{E.289})$$

where (E.289) relies on Assumption 9. It follows that

$$q_t(z_t^{(i)} = m) = p_t(z_t^{(i)} = m \mid s_t^{(i)}, \Theta, \pi_t) = \frac{p_m(y_t^{(i)} \mid \mathbf{x}_t^{(i)}, \theta_m) \times \pi_{tm}}{\sum_{m'=1}^M p_{m'}(y_t^{(i)} \mid \mathbf{x}_t^{(i)}, \theta_{m'}) \times \pi_{tm'}}. \quad (\text{E.290})$$

**M-step.** We maximize now  $\mathcal{L}(\Theta, \Pi, Q_{1:T})$  with respect to  $\{\Theta, \Pi\}$ . By dropping the terms not depending on  $\{\Theta, \Pi\}$  in the expression of  $\mathcal{L}(\Theta, \Pi, Q_{1:T})$  we write:

$$\begin{aligned} & \mathcal{L}(\Theta, \Pi, Q_{1:T}) \\ &= \sum_{t=1}^T \sum_{i=1}^{n_t} \sum_{m=1}^M q_t(z_t^{(i)} = m) \log p_t(s_t^{(i)}, z_t^{(i)} = m \mid \Theta, \pi_t) + c \end{aligned} \quad (\text{E.291})$$

$$= \sum_{t=1}^T \sum_{i=1}^{n_t} \sum_{m=1}^M q_t(z_t^{(i)} = m) \left[ \log p_t(s_t^{(i)} \mid z_t^{(i)} = m, \Theta, \pi_t) + \log p_t(z_t^{(i)} = m \mid \Theta, \pi_t) \right] + c \quad (\text{E.292})$$

$$= \sum_{t=1}^T \sum_{i=1}^{n_t} \sum_{m=1}^M q_t(z_t^{(i)} = m) \left[ \log p_{\theta_m}(s_t^{(i)}) + \log \pi_{tm} \right] + c \quad (\text{E.293})$$

$$= \sum_{t=1}^T \sum_{i=1}^{n_t} \sum_{m=1}^M q_t(z_t^{(i)} = m) \left[ \log p_{\theta_m}(y_t^{(i)} \mid \mathbf{x}_t^{(i)}) + \log p_m(\mathbf{x}_t^{(i)}) + \log \pi_{tm} \right] + c \quad (\text{E.294})$$

$$= \sum_{t=1}^T \sum_{i=1}^{n_t} \sum_{m=1}^M q_t(z_t^{(i)} = m) \left[ \log p_{\theta_m}(y_t^{(i)} \mid \mathbf{x}_t^{(i)}) + \log \pi_{tm} \right] + c', \quad (\text{E.295})$$

$$(\text{E.296})$$

where  $c$  and  $c'$  are constant not depending on  $\{\Theta, \Pi\}$ .

Thus, for  $t \in [T]$  and  $m \in [M]$ , by solving a simple optimization problem we update  $\pi_{tm}$  as follows:

$$\pi_{tm} = \frac{\sum_{i=1}^{n_t} q_t(z_t^{(i)} = m)}{n_t}. \quad (\text{E.297})$$

On the other hand, for  $m \in [M]$ , we update  $\theta_m$  by solving:

$$\theta_m \in \arg \min_{\theta \in \mathbb{R}^d} \sum_{t=1}^T \sum_{i=1}^{n_t} q_t(z_t^{(i)} = m) \times l\left(h_\theta(\mathbf{x}_t^{(i)}), y_t^{(i)}\right). \quad (\text{E.298})$$

□

### E.3 Proofs for Client-Server Setting

#### E.3.1 Additional Notations

**Remark 10.** For convenience and without loss of generality, we suppose in this section that  $\omega \in \Delta^T$ , i.e.,  $\forall t \in [T]$ ,  $\omega_t \geq 0$  and  $\sum_{t=1}^T \omega_t = 1$ .

At iteration  $k > 0$ , we use  $\mathbf{u}_t^{k-1,j}$  to denote the  $j$ -th iterate of the local solver at client  $t \in [T]$ , thus

$$\mathbf{u}_t^{k-1,0} = \mathbf{u}^{k-1}, \quad (\text{E.299})$$

and

$$\mathbf{u}^k = \sum_{t=1}^T \omega_t \cdot \mathbf{u}_t^{k-1,J}. \quad (\text{E.300})$$

At iteration  $k > 0$ , the local solver's updates at client  $t \in [T]$  can be written as (for  $0 \leq j \leq J-1$ ):

$$\mathbf{u}_t^{k-1,j+1} = \mathbf{u}_t^{k-1,j} - \eta_{k-1,j} \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1}; \xi_t^{k-1,j} \right), \quad (\text{E.301})$$

where  $\xi_t^{k-1,j}$  is the batch drawn at the  $j$ -th local update of  $\mathbf{u}_t^{k-1}$ .

We introduce  $\eta_{k-1} = \sum_{j=0}^{J-1} \eta_{k-1,j}$ , and we define the normalized update of the local solver at client  $t \in [T]$  as,

$$\hat{\delta}_t^{k-1} \triangleq -\frac{\mathbf{u}_t^{k-1,J} - \mathbf{u}_t^{k-1,0}}{\eta_{k-1}} = \frac{\sum_{j=0}^{J-1} \eta_{k-1,j} \cdot \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1}; \xi_t^{k-1,j} \right)}{\sum_{j=0}^{J-1} \eta_{k-1,j}}, \quad (\text{E.302})$$

and also define

$$\delta_t^{k-1} \triangleq \frac{\sum_{j=0}^{J-1} \eta_{k-1,j} \cdot \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1} \right)}{\eta_{k-1}}. \quad (\text{E.303})$$

With this notation,

$$\mathbf{u}^k - \mathbf{u}^{k-1} = -\eta_{k-1} \cdot \sum_{t=1}^T \omega_t \cdot \hat{\delta}_t^{k-1}. \quad (\text{E.304})$$

Finally, we define  $g^k$ ,  $k > 0$  as

$$g^k(\mathbf{u}, \mathbf{v}_{1:T}) \triangleq \sum_{t=1}^T \omega_t \cdot g_t^k(\mathbf{u}, \mathbf{v}_t). \quad (\text{E.305})$$

Note that  $g^k$  is a convex combination of functions  $g_t^k$ ,  $t \in [T]$ .

### E.3.2 Proof of Theorem 3.5.3'

**Lemma E.4.** *Suppose that Assumptions 12'–14' hold. Then, for  $k > 0$ , and  $(\eta_{k,j})_{0 \leq j \leq J-1}$  such that  $\eta_k \triangleq \sum_{j=0}^{J-1} \eta_{k,j} \leq \min \left\{ \frac{1}{2\sqrt{2L}}, \frac{1}{4L\beta} \right\}$ , the updates of federated surrogate optimization (Alg 10) verify*

$$\begin{aligned} \mathbb{E} \left[ \frac{f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1})}{\eta_{k-1}} \right] \leq & \\ & - \frac{1}{4} \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 - \frac{1}{\eta_{k-1}} \sum_{t=1}^T \omega_t \cdot d_{\mathcal{V}}(\mathbf{v}_t^{k-1}, \mathbf{v}_t^k) \\ & + 2\eta_{k-1}L \left( \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}^2}{\eta_{k-1}} L + 1 \right) \sigma^2 + 4\eta_{k-1}^2 L^2 G^2. \end{aligned} \quad (\text{E.306})$$

*Proof.* This proof uses standard techniques from distributed stochastic optimization. It is inspired by [Wan+20b, Theorem 1].

For  $k > 0$ ,  $g^k$  is  $L$ -smooth wrt  $\mathbf{u}$ , because it is a convex combination of  $L$ -smooth functions  $g_t^k$ ,  $t \in [T]$ . Thus, we write

$$g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^{k-1}) - g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \leq \left\langle \mathbf{u}^k - \mathbf{u}^{k-1}, \nabla_{\mathbf{u}} g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\rangle + \frac{L}{2} \left\| \mathbf{u}^k - \mathbf{u}^{k-1} \right\|^2, \quad (\text{E.307})$$

where  $\langle \mathbf{u}, \mathbf{u}' \rangle$  denotes the scalar product of vectors  $\mathbf{u}$  and  $\mathbf{u}'$ . Using Eq. (E.304), and taking the expectation over random batches  $(\xi_t^{k-1,j})_{\substack{0 \leq j \leq J-1, \\ 1 \leq t \leq T}}$ , we have

$$\begin{aligned} \mathbb{E} \left[ g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^{k-1}) - g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right] \leq & \\ & - \underbrace{\eta_{k-1} \mathbb{E} \left\langle \sum_{t=1}^T \omega_t \cdot \hat{\delta}_t^{k-1}, \nabla_{\mathbf{u}} g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\rangle}_{\triangleq T_1} + \frac{L\eta_{k-1}^2}{2} \cdot \underbrace{\mathbb{E} \left\| \sum_{t=1}^T \omega_t \cdot \hat{\delta}_t^{k-1} \right\|^2}_{\triangleq T_2}. \end{aligned} \quad (\text{E.308})$$

We bound each of those terms separately. For  $T_1$  we have

$$T_1 = \mathbb{E} \left\langle \sum_{t=1}^T \omega_t \cdot \hat{\delta}_t^{k-1}, \nabla_{\mathbf{u}} g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\rangle \quad (\text{E.309})$$

$$\begin{aligned} &= \mathbb{E} \left\langle \sum_{t=1}^T \omega_t \cdot (\hat{\delta}_t^{k-1} - \delta_t^{k-1}), \nabla_{\mathbf{u}} g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\rangle \\ &+ \mathbb{E} \left\langle \sum_{t=1}^T \omega_t \cdot \delta_t^{k-1}, \nabla_{\mathbf{u}} g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\rangle. \end{aligned} \quad (\text{E.310})$$

Because stochastic gradients are unbiased (Assumption 13'), we have

$$\mathbb{E} \left[ \hat{\delta}_t^{k-1} - \delta_t^{k-1} \right] = 0, \quad (\text{E.311})$$

thus,

$$T_1 = \mathbb{E} \left\langle \sum_{t=1}^T \omega_t \cdot \delta_t^{k-1}, \nabla_{\mathbf{u}} g^k \left( \mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1} \right) \right\rangle \quad (\text{E.312})$$

$$\begin{aligned} &= \frac{1}{2} \left( \left\| \nabla_{\mathbf{u}} g^k \left( \mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1} \right) \right\|^2 + \mathbb{E} \left\| \sum_{t=1}^T \omega_t \cdot \delta_t^{k-1} \right\|^2 \right) \\ &\quad - \frac{1}{2} \mathbb{E} \left\| \nabla_{\mathbf{u}} g^k \left( \mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1} \right) - \sum_{t=1}^T \omega_t \cdot \delta_t^{k-1} \right\|^2. \end{aligned} \quad (\text{E.313})$$

For  $T_2$  we have for  $k > 0$ ,

$$T_2 = \mathbb{E} \left\| \sum_{t=1}^T \omega_t \cdot \hat{\delta}_t^{k-1} \right\|^2 \quad (\text{E.314})$$

$$= \mathbb{E} \left\| \sum_{t=1}^T \omega_t \cdot \left( \hat{\delta}_t^{k-1} - \delta_t^{k-1} \right) + \sum_{t=1}^T \omega_t \cdot \delta_t^{k-1} \right\|^2 \quad (\text{E.315})$$

$$\leq 2 \mathbb{E} \left\| \sum_{t=1}^T \omega_t \cdot \left( \hat{\delta}_t^{k-1} - \delta_t^{k-1} \right) \right\|^2 + 2 \mathbb{E} \left\| \sum_{t=1}^T \omega_t \cdot \delta_t^{k-1} \right\|^2 \quad (\text{E.316})$$

$$\begin{aligned} &= 2 \sum_{t=1}^T \omega_t^2 \cdot \mathbb{E} \left\| \hat{\delta}_t^{k-1} - \delta_t^{k-1} \right\|^2 + 2 \sum_{1 \leq s \neq t \leq T} \omega_t \omega_s \mathbb{E} \left\langle \hat{\delta}_t^{k-1} - \delta_t^{k-1}, \hat{\delta}_s^{k-1} - \delta_s^{k-1} \right\rangle \\ &\quad + 2 \mathbb{E} \left\| \sum_{t=1}^T \omega_t \delta_t^{k-1} \right\|^2. \end{aligned} \quad (\text{E.317})$$

Since clients sample batches independently, and stochastic gradients are unbiased (Assumption 13'), we have

$$\mathbb{E} \left\langle \hat{\delta}_t^{k-1} - \delta_t^{k-1}, \hat{\delta}_s^{k-1} - \delta_s^{k-1} \right\rangle = 0, \quad (\text{E.318})$$

thus,

$$T_2 \leq 2 \sum_{t=1}^T \omega_t^2 \cdot \mathbb{E} \left\| \hat{\delta}_t^{k-1} - \delta_t^{k-1} \right\|^2 + 2 \mathbb{E} \left\| \sum_{t=1}^T \omega_t \delta_t^{k-1} \right\|^2 \quad (\text{E.319})$$

$$\begin{aligned} &= 2 \sum_{t=1}^T \omega_t^2 \mathbb{E} \left\| \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \left[ \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1}; \xi_t^{k-1,j} \right) \right] \right\|^2 \\ &\quad + 2 \mathbb{E} \left\| \sum_{t=1}^T \omega_t \delta_t^{k-1} \right\|^2. \end{aligned} \quad (\text{E.320})$$

Using Jensen inequality, we have

$$\left\| \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \left[ \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1}; \xi_t^{k-1,j} \right) \right] \right\|^2 \leq$$

$$\sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1}; \xi_t^{k-1,j} \right) \right\|^2, \quad (\text{E.321})$$

and since the variance of stochastic gradients is bounded by  $\sigma^2$  (Assumption 13'), it follows that

$$\begin{aligned} \mathbb{E} \left\| \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \left[ \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1}; \xi_t^{k-1,j} \right) \right] \right\|^2 \\ \leq \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \sigma^2 = \sigma^2. \end{aligned} \quad (\text{E.322})$$

Replacing back in the expression of  $T_2$ , we have

$$T_2 \leq 2 \sum_{t=1}^T \omega_t^2 \sigma^2 + 2 \mathbb{E} \left\| \sum_{t=1}^T \omega_t \cdot \delta_t^{k-1} \right\|^2. \quad (\text{E.323})$$

Finally, since  $0 \leq \omega_t \leq 1$ ,  $t \in [T]$  and  $\sum_{t=1}^T \omega_t = 1$ , we have

$$T_2 \leq 2\sigma^2 + 2 \mathbb{E} \left\| \sum_{t=1}^T \omega_t \cdot \delta_t^{k-1} \right\|^2. \quad (\text{E.324})$$

Having bounded  $T_1$  and  $T_2$ , we can replace Eq. (E.313) and Eq. (E.324) in Eq. (E.308), and we get

$$\begin{aligned} \mathbb{E} \left[ g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^{k-1}) - g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right] &\leq -\frac{\eta_{k-1}}{2} \left\| \nabla_{\mathbf{u}} g^k \left( \mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1} \right) \right\|^2 + \eta_{k-1}^2 L \sigma^2 \\ &\quad - \frac{\eta_{k-1}}{2} (1 - 2L\eta_{k-1}) \cdot \mathbb{E} \left\| \sum_{t=1}^T \omega_t \cdot \delta_t^{k-1} \right\|^2 \\ &\quad + \frac{\eta_{k-1}}{2} \mathbb{E} \left\| \nabla_{\mathbf{u}} g^k \left( \mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1} \right) - \sum_{t=1}^T \omega_t \cdot \delta_t^{k-1} \right\|^2. \end{aligned} \quad (\text{E.325})$$

As  $\eta_{k-1} \leq \frac{1}{2\sqrt{2}L} \leq \frac{1}{2L}$ , we have

$$\begin{aligned} \mathbb{E} \left[ g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^{k-1}) - g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right] &\leq -\frac{\eta_{k-1}}{2} \left\| \nabla_{\mathbf{u}} g^k \left( \mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1} \right) \right\|^2 + \eta_{k-1}^2 L \sigma^2 \\ &\quad + \frac{\eta_{k-1}}{2} \mathbb{E} \left\| \nabla_{\mathbf{u}} g^k \left( \mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1} \right) - \sum_{t=1}^T \omega_t \delta_t^{k-1} \right\|^2. \end{aligned} \quad (\text{E.326})$$

Replacing  $\nabla_{\mathbf{u}} g^k \left( \mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1} \right) = \sum_{t=1}^T \omega_t \cdot \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right)$ , and using Jensen inequality to bound the last term in the RHS of Eq. (E.326), we have

$$\begin{aligned} \mathbb{E} \left[ g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^{k-1}) - g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right] &\leq -\frac{\eta_{k-1}}{2} \left\| \nabla_{\mathbf{u}} g^k \left( \mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1} \right) \right\|^2 + \eta_{k-1}^2 L \sigma^2 \\ &\quad + \frac{\eta_{k-1}}{2} \sum_{t=1}^T \omega_t \cdot \underbrace{\mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) - \delta_t^{k-1} \right\|^2}_{\triangleq T_3}. \end{aligned} \quad (\text{E.327})$$

We now bound the term  $T_3$ :

$$T_3 = \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) - \delta_t^{k-1} \right\|^2 \quad (\text{E.328})$$

$$= \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) - \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1} \right) \right\|^2 \quad (\text{E.329})$$

$$= \mathbb{E} \left\| \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \left[ \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1} \right) \right] \right\|^2 \quad (\text{E.330})$$

$$\leq \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1} \right) \right\|^2 \quad (\text{E.331})$$

$$\leq \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} L^2 \mathbb{E} \left\| \mathbf{u}^{k-1} - \mathbf{u}_t^{k-1,j} \right\|^2, \quad (\text{E.332})$$

where the first inequality follows from Jensen inequality and the second one follow from the  $L$ -smoothness of  $g_t^k$  (Assumption 12'). We bound now the term  $\mathbb{E} \left\| \mathbf{u}^{k-1} - \mathbf{u}_t^{k-1,j} \right\|^2$  for  $j \in \{0, \dots, J-1\}$  and  $t \in [T]$ ,

$$\mathbb{E} \left\| \mathbf{u}^{k-1} - \mathbf{u}_t^{k-1,j} \right\|^2 = \mathbb{E} \left\| \mathbf{u}_t^{k-1,j} - \mathbf{u}_t^{k-1,0} \right\|^2 \quad (\text{E.333})$$

$$= \mathbb{E} \left\| \sum_{l=0}^{j-1} \left( \mathbf{u}_t^{k-1,l+1} - \mathbf{u}_t^{k-1,l} \right) \right\|^2 \quad (\text{E.334})$$

$$= \mathbb{E} \left\| \sum_{l=0}^{j-1} \eta_{k-1,l} \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1}; \xi_t^{k-1,l} \right) \right\|^2 \quad (\text{E.335})$$

$$\leq 2 \mathbb{E} \left\| \sum_{l=0}^{j-1} \eta_{k-1,l} \left[ \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,l}, \mathbf{v}_t^{k-1}; \xi_t^{k-1,l} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,l}, \mathbf{v}_t^{k-1} \right) \right] \right\|^2$$

$$+ 2 \mathbb{E} \left\| \sum_{l=0}^{j-1} \eta_{k-1,l} \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,l}, \mathbf{v}_t^{k-1} \right) \right\|^2 \quad (\text{E.336})$$

$$= 2 \sum_{l=0}^{j-1} \eta_{k-1,l}^2 \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,l}, \mathbf{v}_t^{k-1}; \xi_t^{k-1,l} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,l}, \mathbf{v}_t^{k-1} \right) \right\|^2$$

$$+ 2 \mathbb{E} \left\| \sum_{l=0}^{j-1} \eta_{k-1,l} \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,l}, \mathbf{v}_t^{k-1} \right) \right\|^2 \quad (\text{E.337})$$

$$\leq 2\sigma^2 \sum_{l=0}^{j-1} \eta_{k-1,l}^2 + 2 \mathbb{E} \left\| \sum_{l=0}^{j-1} \eta_{k-1,l} \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,l}, \mathbf{v}_t^{k-1} \right) \right\|^2, \quad (\text{E.338})$$

where, in the last two steps, we used the fact that stochastic gradients are unbiased and have bounded variance (Assumption 13'). We bound now the last term in the RHS of Eq. (E.338),

$$\mathbb{E} \left\| \sum_{l=0}^{j-1} \eta_{k-1,l} \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,l}, \mathbf{v}_t^{k-1} \right) \right\|^2 =$$

$$\mathbb{E} \left\| \left( \sum_{l'=0}^{j-1} \eta_{k-1,l'} \right) \cdot \sum_{l=0}^{j-1} \frac{\eta_{k-1,l}}{\sum_{l'=0}^{j-1} \eta_{k-1,l'}} \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,l}, \mathbf{v}_t^{k-1} \right) \right\|^2 \quad (\text{E.339})$$

$$\leq \left( \sum_{l'=0}^{j-1} \eta_{k-1,l'} \right)^2 \cdot \sum_{l=0}^{j-1} \frac{\eta_{k-1,l}}{\sum_{l'=0}^{j-1} \eta_{k-1,l'}} \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,l}, \mathbf{v}_t^{k-1} \right) \right\|^2 \quad (\text{E.340})$$

$$= \left( \sum_{l=0}^{j-1} \eta_{k-1,l} \right) \cdot \sum_{l=0}^{j-1} \eta_{k-1,l} \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,l}, \mathbf{v}_t^{k-1} \right) \right\|^2 \quad (\text{E.341})$$

$$= \left( \sum_{l=0}^{j-1} \eta_{k-1,l} \right) \cdot \sum_{l=0}^{j-1} \eta_{k-1,l} \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,0}, \mathbf{v}_t^{k-1} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,0}, \mathbf{v}_t^{k-1} \right) + \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,l}, \mathbf{v}_t^{k-1} \right) \right\|^2 \quad (\text{E.342})$$

$$\leq 2 \left( \sum_{l=0}^{j-1} \eta_{k-1,l} \right) \cdot \sum_{l=0}^{j-1} \eta_{k-1,l} \cdot \left[ \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,0}, \mathbf{v}_t^{k-1} \right) \right\|^2 + \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,l}, \mathbf{v}_t^{k-1} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,0}, \mathbf{v}_t^{k-1} \right) \right\|^2 \right] \quad (\text{E.343})$$

$$= 2 \left( \sum_{l=0}^{j-1} \eta_{k-1,l} \right) \cdot \sum_{l=0}^{j-1} \eta_{k-1,l} \cdot \left[ \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2 + \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,l}, \mathbf{v}_t^{k-1} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2 \right] \quad (\text{E.344})$$

$$\leq 2 \left( \sum_{l=0}^{j-1} \eta_{k-1,l} \right) \sum_{l=0}^{j-1} \eta_{k-1,l} \left[ \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2 + L^2 \mathbb{E} \left\| \mathbf{u}_t^{k-1,l} - \mathbf{u}^{k-1} \right\|^2 \right] \quad (\text{E.345})$$

$$= 2L^2 \left( \sum_{l=0}^{j-1} \eta_{k-1,l} \right) \sum_{l=0}^{j-1} \eta_{k-1,l} \cdot \mathbb{E} \left\| \mathbf{u}_t^{k-1,l} - \mathbf{u}^{k-1} \right\|^2 + 2 \left( \sum_{l=0}^{j-1} \eta_{k-1,l} \right)^2 \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2, \quad (\text{E.346})$$

where the first inequality is obtained using Jensen inequality, and the last one is a result of the  $L$ -smoothness of  $g_t$  (Assumption 12'). Replacing Eq. (E.346) in Eq. (E.338), we have

$$\begin{aligned} \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \cdot \mathbb{E} \left\| \mathbf{u}^{k-1} - \mathbf{u}_t^{k-1,j} \right\|^2 &\leq 2\sigma^2 \left( \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \cdot \sum_{l=0}^{j-1} \eta_{k-1,l}^2 \right) \\ &+ 4L^2 \sum_{j=0}^{J-1} \left( \frac{\eta_{k-1,j}}{\eta_{k-1}} \sum_{l=0}^{j-1} \eta_{k-1,l} \right) \cdot \left( \sum_{l=0}^{j-1} \eta_{k-1,l} \cdot \mathbb{E} \left\| \mathbf{u}_t^{k-1,l} - \mathbf{u}^{k-1} \right\|^2 \right) \\ &+ 4 \left( \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \left( \sum_{l=0}^{j-1} \eta_{k-1,l} \right)^2 \right) \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2. \end{aligned} \quad (\text{E.347})$$



Since  $\sum_{l=0}^{j-1} \eta_{k-1,l} \cdot \mathbb{E} \left\| \mathbf{u}_t^{k-1,l} - \mathbf{u}_t^{k-1} \right\|^2 \leq \sum_{j=0}^{J-1} \eta_{k-1,j} \cdot \mathbb{E} \left\| \mathbf{u}_t^{k-1,j} - \mathbf{u}_t^{k-1} \right\|^2$ , we have

$$\begin{aligned} \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \cdot \mathbb{E} \left\| \mathbf{u}^{k-1} - \mathbf{u}_t^{k-1,j} \right\|^2 &\leq 2\sigma^2 \left( \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \cdot \sum_{l=0}^{j-1} \eta_{k-1,l}^2 \right) \\ &+ 4L^2 \left( \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \sum_{l=0}^{j-1} \eta_{k-1,l} \right) \cdot \left( \sum_{j=0}^{J-1} \eta_{k-1,j} \cdot \mathbb{E} \left\| \mathbf{u}_t^{k-1,j} - \mathbf{u}^{k-1} \right\|^2 \right) \\ &+ 4 \left( \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \left( \sum_{l=0}^{j-1} \eta_{k-1,l} \right)^2 \right) \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2. \end{aligned} \quad (\text{E.348})$$

We use Lemma E.14 to simplify the last expression, obtaining

$$\begin{aligned} \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \cdot \mathbb{E} \left\| \mathbf{u}^{k-1} - \mathbf{u}_t^{k-1,j} \right\|^2 &\leq 2\sigma^2 \cdot \left\{ \sum_{j=0}^{J-1} \eta_{k-1,j}^2 \right\} \\ &+ 4\eta_{k-1}^2 \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2 + 4\eta_{k-1} L^2 \cdot \sum_{j=0}^{J-1} \eta_{k-1,j} \mathbb{E} \left\| \mathbf{u}_t^{k-1,j} - \mathbf{u}^{k-1} \right\|^2. \end{aligned} \quad (\text{E.349})$$

Rearranging the terms, we have

$$\begin{aligned} \left( 1 - 4\eta_{k-1}^2 L^2 \right) \cdot \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \cdot \mathbb{E} \left\| \mathbf{u}^{k-1} - \mathbf{u}_t^{k-1,j} \right\|^2 &\leq 2\sigma^2 \cdot \left\{ \sum_{j=0}^{J-1} \eta_{k-1,j}^2 \right\} \\ &+ 4\eta_{k-1}^2 \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2. \end{aligned} \quad (\text{E.350})$$

Finally, replacing Eq. (E.350) into Eq. (E.332), we have

$$\left( 1 - 4\eta_{k-1}^2 L^2 \right) \cdot T_3 \leq 2\sigma^2 L^2 \cdot \left( \sum_{j=0}^{J-1} \eta_{k-1,j}^2 \right) + 4\eta_{k-1}^2 L^2 \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2. \quad (\text{E.351})$$

For  $\eta_{k-1}$  small enough, in particular if  $\eta_{k-1} \leq \frac{1}{2\sqrt{2}L}$ , then  $\frac{1}{2} \leq 1 - 4\eta_{k-1}^2 L^2$ , thus

$$\frac{T_3}{2} \leq 2\sigma^2 L^2 \cdot \left( \sum_{j=0}^{J-1} \eta_{k-1,j}^2 \right) + 4\eta_{k-1}^2 L^2 \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2. \quad (\text{E.352})$$

Replacing the bound of  $T_3$  from Eq. (E.352) into Eq. (E.327), we have obtained

$$\begin{aligned} \mathbb{E} \left[ g^k \left( \mathbf{u}^k, \mathbf{v}_{1:T}^{k-1} \right) - g^k \left( \mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1} \right) \right] &\leq -\frac{\eta_{k-1}}{2} \mathbb{E} \left\| \nabla_{\mathbf{u}} g^k \left( \mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1} \right) \right\|^2 \\ &+ 4\eta_{k-1}^3 L^2 \sum_{t=1}^T \omega_t \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2 \\ &+ 2\eta_{k-1} L \left( \sum_{j=0}^{J-1} \eta_{k-1,j}^2 L + \eta_{k-1} \right) \cdot \sigma^2. \end{aligned} \quad (\text{E.353})$$

Using Assumption 14', we have

$$\begin{aligned} \mathbb{E}\left[g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^{k-1}) - g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1})\right] &\leq -\frac{\eta_{k-1}}{2}\mathbb{E}\left\|\nabla_{\mathbf{u}}g^k\left(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}\right)\right\|^2 \\ &\quad + 4\eta_{k-1}^3L^2\beta^2 \cdot \mathbb{E}\left\|\sum_{t=1}^T\omega_t \cdot \nabla_{\mathbf{u}}g_t^k\left(\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}\right)\right\|^2 \\ &\quad + 2\eta_{k-1}L\left(\sum_{j=0}^{J-1}\eta_{k-1,j}^2L + \eta_{k-1}\right) \cdot \sigma^2 + 4\eta_{k-1}^3L^2G^2. \end{aligned} \quad (\text{E.354})$$

Dividing by  $\eta_{k-1}$ , we get

$$\begin{aligned} \mathbb{E}\left[\frac{g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^{k-1}) - g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1})}{\eta_{k-1}}\right] &\leq \frac{8\eta_{k-1}^2L^2\beta^2 - 1}{2}\mathbb{E}\left\|\nabla_{\mathbf{u}}g^k\left(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}\right)\right\|^2 \\ &\quad + 2\eta_{k-1}L\left(\sum_{j=0}^{J-1}\frac{\eta_{k-1,j}^2}{\eta_{k-1}}L + 1\right) \cdot \sigma^2 + 4\eta_{k-1}^2L^2G^2. \end{aligned} \quad (\text{E.355})$$

For  $\eta_{k-1}$  small enough, if  $\eta_{k-1} \leq \frac{1}{4L\beta}$ , then  $8\eta_{k-1}^2L^2\beta^2 - 1 \leq \frac{1}{2}$ . Thus,

$$\begin{aligned} \mathbb{E}\left[\frac{g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^{k-1}) - g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1})}{\eta_{k-1}}\right] &\leq -\frac{1}{4}\mathbb{E}\left\|\nabla_{\mathbf{u}}g^k\left(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}\right)\right\|^2 \\ &\quad + 2\eta_{k-1}L\left(\sum_{j=0}^{J-1}\frac{\eta_{k-1,j}^2}{\eta_{k-1}}L + 1\right) \cdot \sigma^2 + 4\eta_{k-1}^2L^2G^2. \end{aligned} \quad (\text{E.356})$$

Since for  $t \in [T]$ ,  $g_t^k$  is a partial first-order surrogate of  $f_t$  near  $\{\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}\}$ , we have (see Def. 1)

$$g_t^k\left(\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}\right) = f_t\left(\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}\right), \quad (\text{E.357})$$

$$\nabla_{\mathbf{u}}g_t^k\left(\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}\right) = \nabla_{\mathbf{u}}f_t\left(\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}\right), \quad (\text{E.358})$$

$$g_t^k\left(\mathbf{u}^k, \mathbf{v}_t^{k-1}\right) = g_t^k\left(\mathbf{u}^k, \mathbf{v}_t^k\right) + d_{\mathcal{V}}\left(\mathbf{v}_t^{k-1}, \mathbf{v}_t^k\right). \quad (\text{E.359})$$

Multiplying by  $\omega_t$  and summing over  $t \in [T]$ , we have

$$g^k\left(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}\right) = f\left(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}\right), \quad (\text{E.360})$$

$$\nabla_{\mathbf{u}}g^k\left(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}\right) = \nabla_{\mathbf{u}}f\left(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}\right), \quad (\text{E.361})$$

$$g^k\left(\mathbf{u}^k, \mathbf{v}_{1:T}^{k-1}\right) = g^k\left(\mathbf{u}^k, \mathbf{v}_{1:T}^k\right) + \sum_{t=1}^T\omega_t \cdot d_{\mathcal{V}}\left(\mathbf{v}_t^{k-1}, \mathbf{v}_t^k\right). \quad (\text{E.362})$$

Replacing Eq. (E.360), Eq. (E.361) and Eq. (E.362) in Eq. (E.356), we have

$$\begin{aligned} \mathbb{E}\left[\frac{g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1})}{\eta_{k-1}}\right] &\leq \\ &\quad -\frac{1}{4}\mathbb{E}\left\|\nabla_{\mathbf{u}}f\left(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}\right)\right\|^2 - \frac{1}{\eta_{k-1}}\sum_{t=1}^T\omega_t \cdot d_{\mathcal{V}}\left(\mathbf{v}_t^{k-1}, \mathbf{v}_t^k\right) \end{aligned}$$

$$+ 2\eta_{k-1}L \left( \left\{ \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}^2}{\eta_{k-1}} \right\} L + 1 \right) \cdot \sigma^2 + 4\eta_{k-1}^2 L^2 G^2. \quad (\text{E.363})$$

Using again Definition 1, we have

$$g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \geq f(\mathbf{u}^k, \mathbf{v}_{1:T}^k), \quad (\text{E.364})$$

thus,

$$\begin{aligned} \mathbb{E} \left[ \frac{f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1})}{\eta_{k-1}} \right] &\leq \\ &- \frac{1}{4} \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 - \frac{1}{\eta_{k-1}} \sum_{t=1}^T \omega_t \cdot d_{\mathcal{V}}(\mathbf{v}_t^{k-1}, \mathbf{v}_t^k) \\ &+ 2\eta_{k-1}L \left( \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}^2}{\eta_{k-1}} L + 1 \right) \cdot \sigma^2 + 4\eta_{k-1}^2 L^2 G^2. \end{aligned} \quad (\text{E.365})$$

□

**Lemma E.5.** For  $k \geq 0$  and  $t \in [T]$ , the iterates of Alg. 10 verify

$$0 \leq d_{\mathcal{V}}(\mathbf{v}_t^{k+1}, \mathbf{v}_t^k) \leq f_t(\mathbf{u}^k, \mathbf{v}_t^k) - f_t(\mathbf{u}^k, \mathbf{v}_t^{k+1}) \quad (\text{E.366})$$

*Proof.* Since  $\mathbf{v}_t^{k+1} \in \arg \min_{v \in V} g_t^k(\mathbf{u}^{k-1}, v)$ , and  $g_t^k$  is a partial first-order surrogate of  $f_t$  near  $\{\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}\}$ , we have

$$g_t^k(\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}) - g_t^k(\mathbf{u}^{k-1}, \mathbf{v}_t^k) = d_{\mathcal{V}}(\mathbf{v}_t^{k-1}, \mathbf{v}_t^k), \quad (\text{E.367})$$

thus,

$$f_t(\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}) - f_t(\mathbf{u}^{k-1}, \mathbf{v}_t^k) \geq d_{\mathcal{V}}(\mathbf{v}_t^{k-1}, \mathbf{v}_t^k), \quad (\text{E.368})$$

where we used the fact that

$$g_t^k(\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}) = f_t(\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}), \quad (\text{E.369})$$

and,

$$g_t^k(\mathbf{u}^{k-1}, \mathbf{v}_t^k) \geq f_t(\mathbf{u}^{k-1}, \mathbf{v}_t^k). \quad (\text{E.370})$$

□

**Theorem 3.5.3'.** Under Assumptions 11'–14', when clients use SGD as local solver with learning rate  $\eta = \frac{a_0}{\sqrt{K}}$ , after a large enough number of communication rounds  $K$ , the iterates of federated surrogate optimization (Alg. 10) satisfy:

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\|_F^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{K}}\right), \quad \frac{1}{K} \sum_{k=1}^K \mathbb{E} \left[ \Delta_{\mathbf{v}} f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right] \leq \mathcal{O}\left(\frac{1}{K^{3/4}}\right), \quad (\text{3.19})$$

where the expectation is over the random batches samples, and  $\Delta_{\mathbf{v}} f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \triangleq f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - f(\mathbf{u}^k, \mathbf{v}_{1:T}^{k+1}) \geq 0$ .

*Proof.* For  $K$  large enough,  $\eta = \frac{a_0}{\sqrt{K}} \leq \frac{1}{J} \min \left\{ \frac{1}{2\sqrt{2}L}, \frac{1}{4L\beta} \right\}$ , thus the assumptions of Lemma E.4 are satisfied. Lemma E.4 and non-negativity of  $d_{\mathcal{V}}$  lead to

$$\begin{aligned} \mathbb{E} \left[ \frac{f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1})}{J\eta} \right] &\leq -\frac{1}{4} \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 \\ &\quad + 2\eta L(\eta L + 1) \cdot \sigma^2 + 4J^2\eta^2 L^2 G^2. \end{aligned} \quad (\text{E.371})$$

Rearranging the terms and summing for  $k \in [K]$ , we have

$$\begin{aligned} \frac{1}{K} \sum_{k=1}^K \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 \\ \leq 4\mathbb{E} \left[ \frac{f(\mathbf{u}^0, \mathbf{v}_{1:T}^0) - f(\mathbf{u}^K, \mathbf{v}_{1:T}^K)}{J\eta K} \right] + 8 \frac{\eta L(\eta L + 1) \cdot \sigma^2 + 2J^2\eta^2 L^2 G^2}{K} \end{aligned} \quad (\text{E.372})$$

$$\leq 4\mathbb{E} \left[ \frac{f(\mathbf{u}^0, \mathbf{v}_{1:T}^0) - f^*}{J\eta K} \right] + 8 \frac{\eta L(\eta L + 1) \cdot \sigma^2 + 2J^2\eta^2 L^2 G^2}{K}, \quad (\text{E.373})$$

where we use Assumption 11' to obtain (E.373). Thus,

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 = \mathcal{O} \left( \frac{1}{\sqrt{K}} \right). \quad (\text{E.374})$$

To prove the second part of Eq. (3.19), we first decompose  $\Delta_{\mathbf{v}} \triangleq f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - f(\mathbf{u}^k, \mathbf{v}_{1:T}^{k+1}) \geq 0$  as follow,

$$\Delta_{\mathbf{v}} = \underbrace{f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - f(\mathbf{u}^{k+1}, \mathbf{v}_{1:T}^{k+1})}_{\triangleq T_1^k} + \underbrace{f(\mathbf{u}^{k+1}, \mathbf{v}_{1:T}^{k+1}) - f(\mathbf{u}^k, \mathbf{v}_{1:T}^{k+1})}_{\triangleq T_2^k}. \quad (\text{E.375})$$

Using again Lemma E.4 and Eq. (E.374), it follows that

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} [T_1^k] \leq \mathcal{O} \left( \frac{1}{K} \right). \quad (\text{E.376})$$

For  $T_2^k$ , we use the fact that  $f$  is  $2L$ -smooth (Lemma E.15) w.r.t.  $u$  and Cauchy-Schwartz inequality. Thus, for  $k > 0$ , we write

$$T_2^k = f(\mathbf{u}^{k+1}, \mathbf{v}_{1:T}^{k+1}) - f(\mathbf{u}^k, \mathbf{v}_{1:T}^{k+1}) \quad (\text{E.377})$$

$$\leq \left\| \nabla_{\mathbf{u}} f(\mathbf{u}^{k+1}, \mathbf{v}_{1:T}^{k+1}) \right\| \cdot \left\| \mathbf{u}^{k+1} - \mathbf{u}^k \right\| + 2L^2 \left\| \mathbf{u}^{k+1} - \mathbf{u}^k \right\|^2. \quad (\text{E.378})$$

Summing over  $k$  and taking expectation:

$$\begin{aligned} \frac{1}{K} \sum_{k=1}^K \mathbb{E} [T_2^k] &\leq \frac{1}{K} \sum_{k=1}^K \mathbb{E} \left[ \left\| \nabla_{\mathbf{u}} f(\mathbf{u}^{k+1}, \mathbf{v}_{1:T}^{k+1}) \right\| \cdot \left\| \mathbf{u}^{k+1} - \mathbf{u}^k \right\| \right] \\ &\quad + \frac{1}{K} \sum_{k=1}^K 2L^2 \mathbb{E} \left[ \left\| \mathbf{u}^{k+1} - \mathbf{u}^k \right\|^2 \right] \end{aligned} \quad (\text{E.379})$$

$$\begin{aligned}
&\leq \frac{1}{K} \sqrt{\sum_{k=1}^K \mathbb{E} \left[ \left\| \nabla_{\mathbf{u}} f \left( \mathbf{u}^{k+1}, \mathbf{v}_{1:T}^{k+1} \right) \right\|^2 \right]} \sqrt{\sum_{k=1}^K \mathbb{E} \left[ \left\| \mathbf{u}^{k+1} - \mathbf{u}^k \right\|^2 \right]} \\
&\quad + \frac{1}{K} \sum_{k=1}^K 2L^2 \mathbb{E} \left[ \left\| \mathbf{u}^{k+1} - \mathbf{u}^k \right\|^2 \right], \tag{E.380}
\end{aligned}$$

where the second inequality follows from Cauchy-Schwarz inequality. From Eq. (E.350), with  $\eta_{k-1} = J\eta$ , we have for  $t \in [T]$

$$\mathbb{E} \left\| \mathbf{u}^k - \mathbf{u}_t^{k-1, J} \right\|^2 \leq 4\sigma^2 J\eta^2 + 8J^3\eta^2 \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2. \tag{E.381}$$

Multiplying the previous by  $\omega_t$  and summing for  $t \in [T]$ , we have

$$\sum_{t=1}^T \omega_t \cdot \mathbb{E} \left\| \mathbf{u}^{k-1} - \mathbf{u}_t^{k-1, J} \right\|^2 \leq 4J^2\sigma^2\eta^2 + 8J^3\eta^2 \cdot \sum_{t=1}^T \omega_t \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2. \tag{E.382}$$

Using Assumption 14', it follows that

$$\sum_{t=1}^T \omega_t \mathbb{E} \left\| \mathbf{u}^{k-1} - \mathbf{u}_t^{k-1, J} \right\|^2 \leq 4J^2\eta^2 \left( 2JG^2 + \sigma^2 \right) + 8J^3\eta^2\beta^2 \mathbb{E} \left\| \sum_{t=1}^T \omega_t \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2. \tag{E.383}$$

Finally using Jensen inequality and the fact that  $g_t^k$  is a partial first-order of  $f_t$  near  $\{u^{k-1}, v_t^{k-1}\}$ , we have

$$\mathbb{E} \left\| \mathbf{u}^{k-1} - \mathbf{u}^k \right\|^2 \leq 4J^2\eta^2 \left( 2JG^2 + \sigma^2 \right) + 8J^3\eta^2\beta^2 \mathbb{E} \left\| \nabla_{\mathbf{u}} f \left( \mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1} \right) \right\|^2. \tag{E.384}$$

From Eq. (E.374) and  $\eta \leq \mathcal{O}(1/\sqrt{K})$ , we obtain

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \left\| \mathbf{u}^{k-1} - \mathbf{u}^k \right\|^2 \leq \mathcal{O}(1), \tag{E.385}$$

Replacing the last inequality in Eq. (E.380) and using again Eq. (E.374), we obtain

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \left[ T_2^k \right] \leq \mathcal{O} \left( \frac{1}{K^{3/4}} \right). \tag{E.386}$$

Combining Eq. (E.376) and Eq. (E.386), it follows that

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \left[ \Delta_{\mathbf{v}} f(u^k, \mathbf{v}_{1:T}^k) \right] \leq \mathcal{O} \left( \frac{1}{K^{3/4}} \right). \tag{E.387}$$

□

### E.3.3 Proof of Theorem 3.5.3

In this section,  $f$  denotes the negative log-likelihood function defined in Eq. (3.6). Moreover, we introduce the negative log-likelihood at client  $t$  as follows

$$f_t(\Theta, \Pi) \triangleq -\frac{\log p(\mathcal{S}_t|\Theta, \Pi)}{n} \triangleq -\frac{1}{n_t} \sum_{i=1}^{n_t} \log p(s_t^{(i)}|\Theta, \pi_t). \quad (\text{E.388})$$

**Theorem 3.5.3.** *Under Assumptions 8–14, when clients use SGD as local solver with learning rate  $\eta = \frac{\alpha_0}{\sqrt{K}}$ , after a large enough number of communication rounds  $K$ , FedEM's iterates satisfy:*

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \left\| \nabla_{\Theta} f(\Theta^k, \Pi^k) \right\|_F^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{K}}\right), \quad \frac{1}{K} \sum_{k=1}^K \Delta_{\Pi} f(\Theta^k, \Pi^k) \leq \mathcal{O}\left(\frac{1}{K^{3/4}}\right), \quad (3.15)$$

where the expectation is over the random batches samples, and  $\Delta_{\Pi} f(\Theta^k, \Pi^k) \triangleq f(\Theta^k, \Pi^k) - f(\Theta^k, \Pi^{k+1}) \geq 0$ .

*Proof.* We prove this result as a particular case of Theorem 3.5.3'. To this purpose, in this section, we consider that  $\mathcal{V} \triangleq \Delta^M$ ,  $\mathbf{u} = \Theta \in \mathbb{R}^{dM}$ ,  $\mathbf{v}_t = \pi_t$ , and  $\omega_t = n_t/n$  for  $t \in [T]$ . For  $k > 0$ , we define  $g_t^k$  as follows:

$$g_t^k(\Theta, \pi_t) = \frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{m=1}^M q_t^k(z_t^{(i)} = m) \cdot \left( l(h_{\theta_m}(\mathbf{x}_t^{(i)}), y_t^{(i)}) - \log p_m(\mathbf{x}_t^{(i)}) - \log \pi_t \right. \\ \left. + \log q_t^k(z_t^{(i)} = m) - c \right), \quad (\text{E.389})$$

where  $c$  is the same constant appearing in Assumption 10, Eq. (3.3). With this definition, it is easy to check that the federated surrogate optimization algorithm (Alg. 10) reduces to FedEM (Alg. 7). Theorem 3.5.3 then follows immediately from Theorem 3.5.3', once we verify that  $(g_t^k)_{1 \leq t \leq T}$  satisfy the assumptions of Theorem 3.5.3'.

Assumption 11', Assumption 13', and Assumption 14' follow directly from Assumption 11, Assumption 13, and Assumption 14, respectively. Lemma E.6 shows that for  $k > 0$ ,  $g^k$  is smooth w.r.t.  $\Theta$  and then Assumption 12' is satisfied. Finally, Lemmas E.7–E.9 show that for  $t \in [T]$   $g_t^k$  is a partial first-order surrogate of  $f_t$  w.r.t.  $\Theta$  near  $\{\Theta^{k-1}, \pi_t\}$  with  $d_{\mathcal{V}}(\cdot, \cdot) = \mathcal{KL}(\cdot|\cdot)$ .  $\square$

**Lemma E.6.** *Under Assumption 12, for  $t \in [T]$  and  $k > 0$ ,  $g_t^k$  is  $L$ -smooth w.r.t.  $\Theta$ .*

*Proof.*  $g_t^k$  is a convex combination of  $L$ -smooth function  $\theta \mapsto l(\theta; s_i^{(i)})$ ,  $i \in [n_t]$ . Thus it is also  $L$ -smooth.  $\square$

**Lemma E.7.** *Suppose that Assumptions 8–10, hold. Then, for  $t \in [T]$ ,  $\Theta \in \mathbb{R}^{M \times d}$  and  $\pi_t \in \Delta^M$*

$$r_t^k(\Theta, \pi_t) \triangleq g_t^k(\Theta, \pi_t) - f_t(\Theta, \pi_t) = \frac{1}{n_t} \sum_{i=1}^{n_t} \mathcal{KL}\left(q_t^k(z_i^{(t)}) \parallel p_t(z_i^{(t)}|s_i^{(t)}, \Theta, \pi_t)\right),$$

where  $\mathcal{KL}$  is Kullback–Leibler divergence.

*Proof.* Let  $k > 0$  and  $t \in [T]$ , and consider  $\Theta \in \mathbb{R}^{M \times d}$  and  $\pi_t \in \Delta^M$ , then

$$g_t^k(\Theta, \pi_t) = \frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{m=1}^M q_t^k(z_t^{(i)} = m) \cdot \left( l(h_{\theta_m}(\mathbf{x}_t^{(i)}), y_t^{(i)}) - \log p_m(\mathbf{x}_t^{(i)}) - \log \pi_t \right. \\ \left. + \log q_t^k(z_t^{(i)} = m) - c \right), \quad (\text{E.390})$$

$$= \frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{m=1}^M q_t^k(z_t^{(i)} = m) \cdot \left( -\log p_m(y_t^{(i)} | \mathbf{x}_t^{(i)}, \theta_m) - \log p_m(\mathbf{x}_t^{(i)}) - \log \pi_t \right. \\ \left. + \log q_t^k(z_t^{(i)} = m) \right) \quad (\text{E.391})$$

$$= \frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{m=1}^M q_t^k(z_t^{(i)} = m) \cdot \left( -\log p_m(y_t^{(i)} | \mathbf{x}_t^{(i)}, \theta_m) \cdot p_m(\mathbf{x}_t^{(i)}) \cdot p_t(z_t^{(i)} = m) \right. \\ \left. + \log q_t^k(z_t^{(i)} = m) \right) \quad (\text{E.392})$$

$$= \frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{m=1}^M q_t^k(z_t^{(i)} = m) \cdot \left( \log q_t^k(z_t^{(i)} = m) - \log p_t(s_t^{(i)}, z_t^{(i)} = m | \Theta, \pi_t) \right) \quad (\text{E.393})$$

$$= \frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{m=1}^M q_t^k(z_t^{(i)} = m) \log \frac{q_t^k(z_t^{(i)} = m)}{p_t(s_t^{(i)}, z_t^{(i)} = m | \Theta, \pi_t)}. \quad (\text{E.394})$$

Thus,

$$r_t^k(\Theta, \pi_t) \triangleq g_t^k(\Theta, \pi_t) - f_t(\Theta, \pi_t) \quad (\text{E.395})$$

$$= -\frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{m=1}^M \left( q_t^k(z_t^{(i)} = m) \cdot \log \frac{p_t(s_t^{(i)}, z_t^{(i)} = m | \Theta, \pi_t)}{q_t^k(z_t^{(i)} = m)} \right) \\ + \frac{1}{n_t} \sum_{i=1}^{n_t} \log p_t(s_t^{(i)} | \Theta, \pi_t) \quad (\text{E.396})$$

$$= \frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{m=1}^M q_t^k(z_t^{(i)} = m) \left( \log p_t(s_t^{(i)} | \Theta, \pi_t) \right. \\ \left. - \log \frac{p_t(s_t^{(i)}, z_t^{(i)} = m | \Theta, \pi_t)}{q_t^k(z_t^{(i)} = m)} \right) \quad (\text{E.397})$$

$$= \frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{m=1}^M q_t^k(z_t^{(i)} = m) \log \frac{p_t(s_t^{(i)} | \Theta, \pi_t) \cdot q_t^k(z_t^{(i)} = m)}{p_t(s_t^{(i)}, z_t^{(i)} = m | \Theta, \pi_t)} \quad (\text{E.398})$$

$$= \frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{m=1}^M q_t^k(z_t^{(i)} = m) \cdot \log \frac{q_t^k(z_t^{(i)} = m)}{p_t(z_t^{(i)} = m | s_t^{(i)}, \Theta, \pi_t)}. \quad (\text{E.399})$$

Thus,

$$r_t^k(\Theta, \pi_t) = \frac{1}{n_t} \sum_{i=1}^{n_t} \mathcal{KL} \left( q_t^k(\cdot) \| p_t(\cdot | s_i^{(t)}), \Theta, \pi_t \right) \geq 0. \quad (\text{E.400})$$

□

The following lemma shows that  $g_t^k$  and  $g^k$  (as defined in Eq. E.305) satisfy the first two properties in Definition 1.

**Lemma E.8.** *Suppose that Assumptions 8–10 and Assumption 12 hold. For all  $k \geq 0$  and  $t \in [T]$ ,  $g_t^k$  is a majorant of  $f_t$  and  $r_t^k \triangleq g_t^k - f_t$  is  $L$ -smooth in  $\Theta$ . Moreover  $r_t^k(\Theta^{k-1}, \pi_t^{k-1}) = 0$  and  $\nabla_{\Theta} r_t^k(\Theta^{k-1}, \pi_t^{k-1}) = 0$ .*

*The same holds for  $g^k$ , i.e.,  $g^k$  is a majorant of  $f$ ,  $r^k \triangleq g^k - f$  is  $L$ -smooth in  $\Theta$ ,  $r^k(\Theta^{k-1}, \Pi^{k-1}) = 0$  and  $\nabla_{\Theta} r^k(\Theta^{k-1}, \Pi^{k-1}) = 0$*

*Proof.* For  $t \in [T]$ , consider  $\Theta \in \mathbb{R}^{M \times d}$  and  $\pi_t \in \Delta^M$ , we have (Lemma E.7)

$$r_t^k(\Theta, \pi_t) \triangleq g_t^k(\Theta, \pi_t) - f_t(\Theta, \pi_t) = \frac{1}{n_t} \sum_{i=1}^{n_t} \mathcal{KL} \left( q_t^k(z_i^{(t)}) \| p_t(z_i^{(t)} | s_i^{(t)}), \Theta, \pi_t \right). \quad (\text{E.401})$$

Since  $\mathcal{KL}$  divergence is non-negative, it follows that  $g_t^k$  is a majorant of  $f_t$ , i.e.,

$$\forall \Theta \in \mathbb{R}^{M \times d}, \pi_t \in \Delta^M : g_t^k(\Theta, \pi_t) \geq f_t(\Theta, \pi_t). \quad (\text{E.402})$$

Moreover since,  $q_t^k(z_t^{(i)}) = p_t(z_t^{(i)} | s_t^{(i)}, \Theta^{k-1}, \pi_t^{k-1})$  for  $k > 0$ , it follows that

$$r_t^k(\Theta^{k-1}, \pi_t^{k-1}) = 0. \quad (\text{E.403})$$

For  $i \in [n_t]$  and  $m \in [M]$ , from Eq. E.290, we have

$$p_t(z_t^{(i)} = m | s_t^{(i)}, \Theta, \pi_t) = \frac{p_m(y_t^{(i)} | \mathbf{x}_t^{(i)}, \theta_m) \times \pi_{tm}}{\sum_{m'=1}^M p_{m'}(y_t^{(i)} | \mathbf{x}_t^{(i)}, \theta_{m'}) \times \pi_{tm'}} \quad (\text{E.404})$$

$$= \frac{\exp[-l(h_{\theta_m}(\mathbf{x}_t^{(i)}), y_t^{(i)})] \times \pi_{tm}}{\sum_{m'=1}^M \exp[-l(h_{\theta_{m'}}(\mathbf{x}_t^{(i)}), y_t^{(i)})] \times \pi_{tm'}} \quad (\text{E.405})$$

$$= \frac{\exp[-l(h_{\theta_m}(\mathbf{x}_t^{(i)}), y_t^{(i)}) + \log \pi_{tm}]}{\sum_{m'=1}^M \exp[-l(h_{\theta_{m'}}(\mathbf{x}_t^{(i)}), y_t^{(i)}) + \log \pi_{tm'}]}. \quad (\text{E.406})$$

For ease of notation, we introduce

$$l_i(\theta) \triangleq l(h_{\theta}(\mathbf{x}_t^{(i)}), y_t^{(i)}), \quad \theta \in \mathbb{R}^d, m \in [M], i \in [n_t], \quad (\text{E.407})$$

$$\gamma_m(\Theta) \triangleq p_t(z_t^{(i)} = m | s_t^{(i)}, \Theta, \pi_t), \quad m \in [M], \quad (\text{E.408})$$

and,

$$\varphi_i(\Theta) \triangleq \mathcal{KL} \left( q_t^k(z_i^{(t)}) \| p_t(z_i^{(t)} | s_t^{(i)}), \Theta, \pi_t \right). \quad (\text{E.409})$$



For  $i \in [n_t]$ , function  $l_i$  is differentiable because smooth (Assum 12), thus  $\gamma_m$ ,  $m \in [M]$  is differentiable as the composition of the softmax function and the function  $\{\Theta \mapsto -l_i(\Theta) + \log \pi_{tm}\}$ . Its gradient is given by

$$\begin{cases} \nabla_{\theta_m} \gamma_m(\Theta) = -\gamma_m(\Theta) \cdot (1 - \gamma_m(\Theta)) \cdot \nabla l_i(\theta_m), \\ \nabla_{\theta_{m'}} \gamma_m(\Theta) = \gamma_m(\Theta) \cdot \gamma_{m'}(\Theta) \cdot \nabla l_i(\theta_m), \quad m' \neq m. \end{cases} \quad (\text{E.410})$$

Thus for  $m \in [M]$ , we have

$$\nabla_{\theta_m} \varphi_i(\Theta) = \sum_{m'=1}^M q_t^k(z_i^{(t)} = m') \cdot \frac{\nabla_{\theta_m} \gamma_{m'}(\Theta)}{\gamma_{m'}(\Theta)} \quad (\text{E.411})$$

$$\begin{aligned} &= \sum_{\substack{m'=1 \\ m' \neq m}}^M \left[ q_t^k(z_i^{(t)} = m') \cdot \frac{\gamma_m(\Theta) \cdot \gamma_{m'}(\Theta)}{\gamma_{m'}(\Theta)} \cdot \nabla l_i(\theta_m) \right] \\ &\quad - q_t^k(z_i^{(t)} = m) \cdot \frac{\gamma_m(\Theta) \cdot (1 - \gamma_m(\Theta))}{\gamma_m(\Theta)} \cdot \nabla l_i(\theta_m). \end{aligned} \quad (\text{E.412})$$

Using the fact that  $\sum_{m'=1}^M q_t^k(z_i^{(t)} = m') = 1$ , it follows that

$$\nabla_{\theta_m} \varphi_i(\Theta) = \left( \gamma_m(\Theta) - q_t^k(z_i^{(t)} = m) \right) \cdot \nabla l_i(\theta_m). \quad (\text{E.413})$$

Since  $l_i$ ,  $i \in [n_t]$  is twice continuously differentiable (Assumption 12), and  $\gamma_m$ ,  $m \in [M]$  is differentiable, then  $\phi_i$ ,  $i \in [n_t]$  is twice continuously differentiable. We use  $\mathbf{H}(\varphi_i(\Theta)) \in \mathbb{R}^{dM \times dM}$  (resp.  $\mathbf{H}(l_i(\theta)) \in \mathbb{R}^{d \times d}$ ) to denote the Hessian of  $\varphi$  (resp.  $l_i$ ) at  $\Theta$  (resp.  $\theta$ ). The Hessian of  $\varphi_i$  is a block matrix given by

$$\begin{cases} \left( \mathbf{H}(\varphi_i(\Theta)) \right)_{m,m} = -\gamma_m(\Theta) \cdot (1 - \gamma_m(\Theta)) \cdot \left( \nabla l_i(\theta_m) \right) \cdot \left( \nabla l_i(\theta_m) \right)^\top \\ \quad \quad \quad + \left( \gamma_m(\Theta) - q_t^k(z_i^{(t)} = m) \right) \cdot \mathbf{H}(l_i(\theta_m)) \\ \left( \mathbf{H}(\varphi_i(\Theta)) \right)_{m,m'} = \gamma_m(\Theta) \cdot \gamma_{m'}(\Theta) \cdot \left( \nabla l_i(\theta_{m'}) \right) \cdot \left( \nabla l_i(\theta_m) \right)^\top, \quad m' \neq m. \end{cases} \quad (\text{E.414})$$

We introduce the block matrix  $\tilde{\mathbf{H}} \in \mathbb{R}^{dM \times dM}$ , defined by

$$\begin{cases} \tilde{\mathbf{H}}_{m,m} = -\gamma_m(\Theta) \cdot (1 - \gamma_m(\Theta)) \cdot \left( \nabla l_i(\theta_m) \right) \cdot \left( \nabla l_i(\theta_m) \right)^\top \\ \tilde{\mathbf{H}}_{m,m'} = \gamma_m(\Theta) \cdot \gamma_{m'}(\Theta) \cdot \left( \nabla_{\theta} l_i(\theta_m) \right) \cdot \left( \nabla l_i(\theta_{m'}) \right)^\top, \quad m' \neq m, \end{cases} \quad (\text{E.415})$$

Eq. (E.414) can be written as

$$\begin{cases} \left( \mathbf{H}(\varphi_i(\Theta)) \right)_{m,m} - \tilde{\mathbf{H}}_{m,m} = \left( \gamma_m(\Theta) - q_t^k(z_i^{(t)} = m) \right) \cdot \mathbf{H}(l_i(\theta_m)) \\ \left( \mathbf{H}(\varphi_i(\Theta)) \right)_{m,m'} - \tilde{\mathbf{H}}_{m,m'} = 0, \quad m' \neq m. \end{cases} \quad (\text{E.416})$$

We recall that a twice differentiable function is  $L$  smooth if and only if the eigenvalues of its Hessian are smaller than  $L$ , see e.g., [Nes03, Lemma 1.2.2] or [Bub15, Section 3.2]. Since  $l_i$  and also  $-l_i$  are  $L$ -smooth (Assumption 12), we have for  $\theta \in \mathbb{R}^d$ ,

$$-L \cdot I_d \preceq \mathbf{H}(l_i(\theta)) \preceq L \cdot I_d. \quad (\text{E.417})$$

Using Lemma E.18, we can conclude that matrix  $\tilde{\mathbf{H}}$  is semi-definite negative. Since

$$-1 \leq \gamma_m(\Theta) - q_t^k(z_i^{(t)} = m) \leq 1, \quad (\text{E.418})$$

it follows that

$$\mathbf{H}(\varphi_i(\Theta)) \preceq L \cdot I_{dM}. \quad (\text{E.419})$$

The last equation proves that  $\varphi_i$  is  $L$ -smooth. Thus  $r_t^k$  is  $L$ -smooth with respect to  $\Theta$  as the average of  $L$ -smooth function.

Moreover, since  $r_t^k(\Theta^{k-1}, \pi_t^{k-1}) = 0$  and  $\forall \Theta, \Pi; r_t^k(\Theta, \pi_t) \geq 0$ , it follows that  $\Theta^{k-1}$  is a minimizer of  $\{\Theta \mapsto r_t^k(\Theta, \pi_t^{k-1})\}$ . Thus,  $\nabla_{\Theta} r_t^k(\Theta^{k-1}, \pi_t^{k-1}) = 0$ .

For  $\Theta \in \mathbb{R}^{M \times d}$  and  $\Pi \in \Delta^{T \times M}$ , we have

$$r^k(\Theta, \Pi) \triangleq g^k(\Theta, \Pi) - f(\Theta, \Pi) \quad (\text{E.420})$$

$$\triangleq \sum_{t=1}^T \frac{n_t}{n} \cdot [g_t^k(\Theta, \pi_t) - f_t(\Theta, \pi_t)] \quad (\text{E.421})$$

$$= \sum_{t=1}^T \frac{n_t}{n} r_t^k(\Theta, \pi_t). \quad (\text{E.422})$$

We see that  $r^k$  is a weighted average of  $(r_t^k)_{1 \leq t \leq T}$ . Thus,  $r_t^k$  is  $L$ -smooth in  $\Theta$ ,  $r^k(\Theta, \Pi) \geq 0$ , moreover  $r_t^k(\Theta^{k-1}, \Pi^{k-1}) = 0$  and  $\nabla_{\Theta} r_t^k(\Theta^{k-1}, \Pi^{k-1}) = 0$ .  $\square$

The following lemma shows that  $g_t^k$  and  $g^k$  satisfy the third property in Definition 1.

**Lemma E.9.** *Suppose that Assumption 8 holds and consider  $\Theta \in \mathbb{R}^{M \times d}$  and  $\Pi \in \Delta^{T \times M}$ , for  $k > 0$ , the iterates of Alg. 10 verify*

$$g^k(\Theta, \Pi) = g^k(\Theta, \Pi^k) + \sum_{t=1}^T \frac{n_t}{n} \mathcal{KL}(\pi_t^k, \pi_t).$$

*Proof.* For  $t \in [T]$  and  $k > 0$ , consider  $\Theta \in \mathbb{R}^{M \times d}$  and  $\pi_t \in \Delta^M$  such that  $\forall m \in [M]; \pi_{tm} \neq 0$ , we have

$$g_t^k(\Theta, \pi_t) - g_t^k(\Theta, \pi_t^k) = \sum_{m=1}^M \underbrace{\left\{ \frac{1}{n_t} \sum_{i=1}^{n_t} q_t^k(z_i^{(t)} = m) \right\}}_{=\pi_{tm}^k \text{ (Prop. 3.5.2)}} \times (\log \pi_{tm}^k - \log \pi_{tm}) \quad (\text{E.423})$$

$$= \sum_{m=1}^M \pi_{tm}^k \log \frac{\pi_{tm}^k}{\pi_{tm}} \quad (\text{E.424})$$

$$= \mathcal{KL}(\pi_t^k, \pi_t). \quad (\text{E.425})$$

We multiply by  $\frac{n_t}{n}$  and some for  $t \in [T]$ . It follows that

$$g^k(\Theta, \Pi^k) + \sum_{t=1}^T \frac{n_t}{n} \mathcal{KL}(\pi_t^k, \pi_t) = g^k(\Theta, \Pi). \quad (\text{E.426})$$

$\square$

## E.4 Proofs for Fully Decentralized Setting

### E.4.1 Additional Notations

**Remark 11.** For convenience and without loss of generality, we suppose in this section that  $\omega_t = 1$ ,  $t \in [T]$ .

We introduce the following matrix notation:

$$\mathbf{U}^k \triangleq [\mathbf{u}_1^k, \dots, \mathbf{u}_T^k] \in \mathbb{R}^{d_u \times T} \quad (\text{E.427})$$

$$\bar{\mathbf{U}}^k \triangleq [\bar{\mathbf{u}}^k, \dots, \bar{\mathbf{u}}^k] \in \mathbb{R}^{d_u \times T} \quad (\text{E.428})$$

$$\partial g^k(\mathbf{U}^k, \mathbf{v}_{1:T}^k; \xi^k) \triangleq [\nabla_{\mathbf{u}} g_1^k(\mathbf{u}_1^k, \mathbf{v}_1^k, \xi_1^k), \dots, \nabla_{\mathbf{u}} g_T^k(\mathbf{u}_T^k, \mathbf{v}_T^k, \xi_T^k)] \in \mathbb{R}^{d_u \times T} \quad (\text{E.429})$$

where  $\bar{\mathbf{u}}^k = \frac{1}{T} \sum_{t=1}^T \mathbf{u}_t^k$  and  $\mathbf{v}_{1:T}^k = (\mathbf{v}_t^k)_{1 \leq t \leq T} \in \mathcal{V}^T$ .

We denote by  $\mathbf{u}_t^{k-1,j}$  the  $j$ -th iterate of the local solver at global iteration  $k$  at client  $t \in [T]$ , and by  $\mathbf{U}^{k-1,j}$  the matrix whose column  $t$  is  $\mathbf{u}_t^{k-1,j}$ , thus,

$$\mathbf{u}_t^{k-1,0} = \mathbf{u}_t^{k-1}; \quad \mathbf{U}^{k-1,0} = \mathbf{U}^{k-1}, \quad (\text{E.430})$$

and,

$$\mathbf{u}_t^k = \sum_{s=1}^T w_{st}^{k-1} \mathbf{u}_s^{k-1,J}; \quad \mathbf{U}^k = \mathbf{U}^{k-1,J} W^{k-1}. \quad (\text{E.431})$$

Using this notation, the updates of Alg. 11 can be summarized as

$$\mathbf{U}^k = \left[ \mathbf{U}^{k-1} - \sum_{j=0}^{J-1} \eta_{k-1,j} \partial g^k(\mathbf{U}^{k-1,j}, \mathbf{v}_{1:T}^k; \xi^{k-1,j}) \right] W^{k-1}. \quad (\text{E.432})$$

Similarly to the client-server setting, we define the normalized update of local solver at client  $t \in [T]$ :

$$\hat{\delta}_t^{k-1} \triangleq -\frac{\mathbf{u}_t^{k-1,J} - \mathbf{u}_t^{k-1,0}}{\eta_{k-1}} = \frac{\sum_{j=0}^{J-1} \eta_{k-1,j} \nabla_{\mathbf{u}} g_t^k(\mathbf{u}_t^{k-1,j}, \mathbf{v}_t^k; \xi_t^{k-1,j})}{\sum_{j=0}^{J-1} \eta_{k-1,j}}, \quad (\text{E.433})$$

and

$$\delta_t^{k-1} \triangleq \frac{\sum_{j=0}^{J-1} \eta_{k-1,j} \nabla_{\mathbf{u}} g_t^k(\mathbf{u}_t^{k-1,j}, \mathbf{v}_t^k)}{\eta_{k-1}}. \quad (\text{E.434})$$

Because clients updates are independent, and stochastic gradient are unbiased, it is clear that

$$\mathbb{E}[\delta_t^{k-1} - \hat{\delta}_t^{k-1}] = 0, \quad (\text{E.435})$$

and that

$$\forall t, s \in [T] \text{ s.t. } s \neq t, \quad \mathbb{E}\langle \delta_t^{k-1} - \hat{\delta}_t^{k-1}, \delta_s^{k-1} - \hat{\delta}_s^{k-1} \rangle = 0. \quad (\text{E.436})$$

We introduce the matrix notation,

$$\hat{\Upsilon}^{k-1} \triangleq [\hat{\delta}_1^{k-1}, \dots, \hat{\delta}_T^{k-1}] \in \mathbb{R}^{d_u \times T}; \quad \Upsilon^{k-1} \triangleq [\delta_1^{k-1}, \dots, \delta_T^{k-1}] \in \mathbb{R}^{d_u \times T}. \quad (\text{E.437})$$

Using this notation, Eq. (E.432) becomes

$$\mathbf{U}^k = [\mathbf{U}^{k-1} - \eta_{k-1} \hat{\Upsilon}^{k-1}] W^{k-1}. \quad (\text{E.438})$$

### E.4.2 Proof of Theorem 3.5.4'

In fully decentralized optimization, proving the convergence usually consists in deriving a recurrence on a term measuring the optimality of the average iterate (in our case this term is  $\mathbb{E} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^k) \right\|^2$ ) and a term measuring the distance to consensus, i.e.,  $\mathbb{E} \sum_{t=1}^T \left\| \mathbf{u}_t^k - \bar{\mathbf{u}}^k \right\|^2$ . In what follows we obtain those two recurrences, and then prove the convergence.

**Lemma E.10** (Average iterate term recursion). *Suppose that Assumptions 12'–14' and Assumption 15 hold. Then, for  $k > 0$ , and  $(\eta_{k,j})_{1 \leq j \leq J-1}$  such that  $\eta_k \triangleq \sum_{j=0}^{J-1} \eta_{k,j} \leq \min \left\{ \frac{1}{2\sqrt{2}L}, \frac{1}{8L\beta} \right\}$ , the updates of fully decentralized federated surrogate optimization (Alg. 11) verify*

$$\begin{aligned} \mathbb{E} \left[ f(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^k) - f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right] &\leq -\frac{1}{T} \sum_{t=1}^T \mathbb{E} d_{\mathcal{V}}(\mathbf{v}_t^k, \mathbf{v}_t^{k-1}) \\ &\quad - \frac{\eta_{k-1}}{8} \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 + \frac{(12+T)\eta_{k-1}L^2}{4T} \cdot \sum_{t=1}^T \mathbb{E} \left\| \mathbf{u}_t^{k-1} - \bar{\mathbf{u}}^{k-1} \right\|^2 \\ &\quad + \frac{\eta_{k-1}^2 L}{T} \left( 4 \sum_{j=0}^{J-1} \frac{L \cdot \eta_{k-1,j}^2}{\eta_{k-1}} + 1 \right) \sigma^2 + \frac{16\eta_{k-1}^3 L^2}{T} G^2. \end{aligned} \quad (\text{E.439})$$

*Proof.* We multiply both sides of Eq. (E.438) by  $\frac{\mathbf{1}\mathbf{1}^\top}{T}$ , thus for  $k > 0$  we have,

$$\mathbf{U}^k \cdot \frac{\mathbf{1}\mathbf{1}^\top}{T} = \left[ \mathbf{U}^{k-1} - \eta_{k-1} \hat{\mathbf{Y}}^{k-1} \right] W^{k-1} \frac{\mathbf{1}\mathbf{1}^\top}{T}, \quad (\text{E.440})$$

since  $W^{k-1}$  is doubly stochastic (Assumption 15), i.e.,  $W^{k-1} \frac{\mathbf{1}\mathbf{1}^\top}{T} = \frac{\mathbf{1}\mathbf{1}^\top}{T}$ , it follows that,

$$\bar{\mathbf{U}}^k = \bar{\mathbf{U}}^{k-1} - \eta_{k-1} \hat{\mathbf{Y}}^{k-1} \cdot \frac{\mathbf{1}\mathbf{1}^\top}{T}, \quad (\text{E.441})$$

thus,

$$\bar{\mathbf{u}}^k = \bar{\mathbf{u}}^{k-1} - \frac{\eta_{k-1}}{T} \cdot \sum_{t=1}^T \hat{\delta}_t^{k-1}. \quad (\text{E.442})$$

Using the fact that  $g^k$  is  $L$ -smooth with respect to  $\mathbf{u}$  (Assumption 12'), we write

$$\mathbb{E} \left[ g^k(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^k) \right] = \mathbb{E} \left[ g^k \left( \bar{\mathbf{u}}^{k-1} - \frac{\eta_{k-1}}{T} \sum_{t=1}^T \hat{\delta}_t^{k-1}, \mathbf{v}_{1:T}^{k-1} \right) \right] \quad (\text{E.443})$$

$$\begin{aligned} &\leq g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) - \mathbb{E} \left\langle \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}), \frac{\eta_{k-1}}{T} \sum_{t=1}^T \hat{\delta}_t^{k-1} \right\rangle \\ &\quad + \frac{L}{2} \mathbb{E} \left\| \frac{\eta_{k-1}}{T} \sum_{t=1}^T \hat{\delta}_t^{k-1} \right\|^2 \end{aligned} \quad (\text{E.444})$$

$$= g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) - \underbrace{\eta_{k-1} \mathbb{E} \left\langle \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}), \frac{1}{T} \sum_{t=1}^T \hat{\delta}_t^{k-1} \right\rangle}_{\triangleq T_1}$$

$$+ \frac{\eta_{k-1}^2 \cdot L}{2T^2} \underbrace{\mathbb{E} \left\| \sum_{t=1}^T \hat{\delta}_t^{k-1} \right\|^2}_{\triangleq T_2}, \quad (\text{E.445})$$

where the expectation is taken over local random batches. As in the client-server case, we bound the terms  $T_1$  and  $T_2$ . First, we bound  $T_1$ , for  $k > 0$ , we have

$$T_1 = \mathbb{E} \left\langle \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}), \frac{1}{T} \sum_{t=1}^T \hat{\delta}_t^{k-1} \right\rangle \quad (\text{E.446})$$

$$\begin{aligned} &= \underbrace{\mathbb{E} \left\langle \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}), \frac{1}{T} \sum_{t=1}^T (\hat{\delta}_t^{k-1} - \delta_t^{k-1}) \right\rangle}_{=0, \text{ because } \mathbb{E}[\hat{\delta}_t^{k-1} - \delta_t^{k-1}] = 0} \\ &\quad + \mathbb{E} \left\langle \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}), \frac{1}{T} \sum_{t=1}^T \delta_t^{k-1} \right\rangle \end{aligned} \quad (\text{E.447})$$

$$= \mathbb{E} \left\langle \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}), \frac{1}{T} \sum_{t=1}^T \delta_t^{k-1} \right\rangle \quad (\text{E.448})$$

$$\begin{aligned} &= \frac{1}{2} \mathbb{E} \left\| \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 + \frac{1}{2} \mathbb{E} \left\| \frac{1}{T} \sum_{t=1}^T \delta_t^{k-1} \right\|^2 \\ &\quad - \frac{1}{2} \mathbb{E} \left\| \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) - \frac{1}{T} \sum_{t=1}^T \delta_t^{k-1} \right\|^2. \end{aligned} \quad (\text{E.449})$$

We bound now  $T_2$ . For  $k > 0$ , we have,

$$T_2 = \mathbb{E} \left\| \sum_{t=1}^T \hat{\delta}_t^{k-1} \right\|^2 \quad (\text{E.450})$$

$$= \mathbb{E} \left\| \sum_{t=1}^T (\hat{\delta}_t^{k-1} - \delta_t^{k-1}) + \sum_{t=1}^T \delta_t^{k-1} \right\|^2 \quad (\text{E.451})$$

$$\leq 2 \mathbb{E} \left\| \sum_{t=1}^T (\hat{\delta}_t^{k-1} - \delta_t^{k-1}) \right\|^2 + 2 \cdot \mathbb{E} \left\| \sum_{t=1}^T \delta_t^{k-1} \right\|^2 \quad (\text{E.452})$$

$$\begin{aligned} &= 2 \cdot \sum_{t=1}^T \mathbb{E} \left\| \hat{\delta}_t^{k-1} - \delta_t^{k-1} \right\|^2 + 2 \sum_{1 \leq t \neq s \leq T} \underbrace{\mathbb{E} \left\langle \hat{\delta}_t^{k-1} - \delta_t^{k-1}, \hat{\delta}_s^{k-1} - \delta_s^{k-1} \right\rangle}_{=0; \text{ because of Eq. (E.436)}} \\ &\quad + 2 \mathbb{E} \left\| \sum_{t=1}^T \delta_t^{k-1} \right\|^2 \end{aligned} \quad (\text{E.453})$$

$$= 2 \cdot \sum_{t=1}^T \mathbb{E} \left\| \hat{\delta}_t^{k-1} - \delta_t^{k-1} \right\|^2 + 2 \cdot \mathbb{E} \left\| \sum_{t=1}^T \delta_t^{k-1} \right\|^2 \quad (\text{E.454})$$

$$= 2 \cdot \mathbb{E} \left\| \sum_{t=1}^T \delta_t^{k-1} \right\|^2 + 2 \cdot \sum_{t=1}^T \left( \frac{1}{\eta_{k-1}^2} \mathbb{E} \left\| \sum_{j=0}^{J-1} \eta_{k-1,j} \cdot \left[ \nabla_{\mathbf{u}} g_t^k(\mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1}) \right] \right\|^2 \right)$$

$$- \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1}, \xi_t^{k-1,j} \right) \Big] \Big\| \Big\|^2. \quad (\text{E.455})$$

Since batches are sampled independently, and stochastic gradients are unbiased with finite variance (Assumption 13'), the last term in the RHS of the previous equation can be bounded using  $\sigma^2$ , leading to

$$T_2 \leq 2 \cdot \sum_{t=1}^T \left[ \frac{\sum_{j=0}^{J-1} \eta_{k-1,j}^2}{\eta_{k-1}^2} \sigma^2 \right] + 2 \cdot \mathbb{E} \left\| \sum_{t=1}^T \delta_t^{k-1} \right\|^2 \quad (\text{E.456})$$

$$= 2T \cdot \sigma^2 \cdot \left( \sum_{t=1}^T \frac{\sum_{j=0}^{J-1} \eta_{k-1,j}^2}{\eta_{k-1}^2} \right) + 2 \mathbb{E} \left\| \sum_{t=1}^T \delta_t^{k-1} \right\|^2 \quad (\text{E.457})$$

$$\leq 2T \cdot \sigma^2 + 2 \cdot \mathbb{E} \left\| \sum_{t=1}^T \delta_t^{k-1} \right\|^2. \quad (\text{E.458})$$

Replacing Eq. (E.449) and Eq. (E.458) in Eq. (E.445), we have

$$\begin{aligned} \mathbb{E} \left[ g^k(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^{k-1}) - g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right] &\leq \\ &- \frac{\eta_{k-1}}{2} \mathbb{E} \left\| \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 - \frac{\eta_{k-1}}{2} (1 - 2L\eta_{k-1}) \mathbb{E} \left\| \frac{1}{T} \sum_{t=1}^T \delta_t^{k-1} \right\|^2 \\ &+ \frac{L}{T} \eta_{k-1}^2 \sigma^2 + \frac{\eta_{k-1}}{2} \mathbb{E} \left\| \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) - \frac{1}{T} \sum_{t=1}^T \delta_t^{k-1} \right\|^2. \end{aligned} \quad (\text{E.459})$$

For  $\eta_{k-1}$  small enough, in particular for  $\eta_{k-1} \leq \frac{1}{2L}$ , we have

$$\begin{aligned} \mathbb{E} \left[ g^k(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^{k-1}) - g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right] &\leq \\ &- \frac{\eta_{k-1}}{2} \mathbb{E} \left\| \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 + \frac{L}{T} \eta_{k-1}^2 \sigma^2 \\ &+ \frac{\eta_{k-1}}{2} \mathbb{E} \left\| \frac{1}{T} \sum_{t=1}^T \left( \nabla_{\mathbf{u}} g_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) - \delta_t^{k-1} \right) \right\|^2. \end{aligned} \quad (\text{E.460})$$

We use Jensen inequality to bound the last term in the RHS of the previous equation, leading to

$$\begin{aligned} \mathbb{E} \left[ g^k(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^{k-1}) - g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right] &\leq \\ &- \frac{\eta_{k-1}}{2} \mathbb{E} \left\| \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 + \frac{L}{T} \eta_{k-1}^2 \sigma^2 \\ &+ \frac{\eta_{k-1}}{2T} \cdot \underbrace{\sum_{t=1}^T \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) - \delta_t^{k-1} \right\|^2}_{T_3}. \end{aligned} \quad (\text{E.461})$$

We bound now the term  $T_3$ :

$$T_3 = \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1} \right) - \delta_t^{k-1} \right\|^2 \quad (\text{E.462})$$

$$= \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1} \right) - \frac{\sum_{j=0}^{J-1} \eta_{k-1,j} \cdot \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1} \right)}{\eta_{k-1}} \right\|^2 \quad (\text{E.463})$$

$$= \mathbb{E} \left\| \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \cdot \left[ \nabla_{\mathbf{u}} g_t^k \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1} \right) \right] \right\|^2. \quad (\text{E.464})$$

Using Jensen inequality, it follows that

$$T_3 \leq \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1} \right) \right\|^2 \quad (\text{E.465})$$

$$= \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1} \right) \right. \\ \left. + \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1} \right) \right\|^2 \quad (\text{E.466})$$

$$\leq 2 \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2 \\ + 2 \cdot \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1} \right) - \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,j}, \mathbf{v}_t^{k-1} \right) \right\|^2 \quad (\text{E.467})$$

$$\leq 2L^2 \cdot \mathbb{E} \left\| \bar{\mathbf{u}}^{k-1} - \mathbf{u}_t^{k-1} \right\|^2 + 2L^2 \cdot \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \cdot \mathbb{E} \left\| \mathbf{u}_t^{k-1,j} - \mathbf{u}_t^{k-1,0} \right\|^2, \quad (\text{E.468})$$

where we used the  $L$ -smoothness of  $g_t^k$  (Assumption 12') to obtain the last inequality. As in the centralized case (Lemma E.4), we bound terms  $\left\| \mathbf{u}_t^{k-1,j} - \mathbf{u}_t^{k-1,0} \right\|^2$ ,  $j \in \{0, \dots, J-1\}$ . Using exactly the same steps as in the proof of Lemma E.4, Eq. (E.350) holds with  $\mathbf{u}_t^{k-1,0}$  instead of  $\mathbf{u}_t^{k-1}$ , i.e.,

$$\left( 1 - 4\eta_{k-1}^2 L^2 \right) \cdot \sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \cdot \mathbb{E} \left\| \mathbf{u}_t^{k-1,0} - \mathbf{u}_t^{k-1,j} \right\|^2 \leq 2\sigma^2 \cdot \left\{ \sum_{j=0}^{J-1} \eta_{k-1,j}^2 \right\} \\ + 4\eta_{k-1}^2 \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,0}, \mathbf{v}_t^{k-1} \right) \right\|^2. \quad (\text{E.469})$$

For  $\eta_{k-1}$  small enough, in particular for  $\eta_{k-1} \leq \frac{1}{2\sqrt{2}L}$ , we have

$$\sum_{j=0}^{J-1} \frac{\eta_{k-1,j}}{\eta_{k-1}} \cdot \mathbb{E} \left\| \mathbf{u}_t^{k-1,0} - \mathbf{u}_t^{k-1,j} \right\|^2 \\ \leq 8\eta_{k-1}^2 \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^k \left( \mathbf{u}_t^{k-1,0}, \mathbf{v}_t^{k-1} \right) \right\|^2 + 4\sigma^2 \cdot \left\{ \sum_{j=0}^{J-1} \eta_{k-1,j}^2 \right\} \quad (\text{E.470})$$

$$\begin{aligned} &\leq 8\eta_{k-1}^2 \cdot \mathbb{E} \left\| \nabla_u g_t^k \left( \mathbf{u}_t^{k-1,0}, \mathbf{v}_t^{k-1} \right) - \nabla_u g_t^k \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1} \right) + \nabla_u g_t^k \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2 \\ &\quad + 4\sigma^2 \cdot \left\{ \sum_{j=0}^{J-1} \eta_{k-1,j}^2 \right\} \end{aligned} \quad (\text{E.471})$$

$$\begin{aligned} &\leq 16\eta_{k-1}^2 \cdot \mathbb{E} \left\| \nabla_u g_t^k \left( \mathbf{u}_t^{k-1,0}, \mathbf{v}_t^{k-1} \right) - \nabla_u g_t^k \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2 \\ &\quad + 16\eta_{k-1}^2 \cdot \left\| \nabla_u g_t^k \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2 + 4\sigma^2 \cdot \left\{ \sum_{j=0}^{J-1} \eta_{k-1,j}^2 \right\} \end{aligned} \quad (\text{E.472})$$

$$\begin{aligned} &\leq 16\eta_{k-1}^2 L^2 \cdot \mathbb{E} \left\| \mathbf{u}_t^{k-1} - \bar{\mathbf{u}}^{k-1} \right\|^2 + 16\eta_{k-1}^2 \cdot \left\| \nabla_u g_t^k \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2 \\ &\quad + 4\sigma^2 \cdot \left\{ \sum_{j=0}^{J-1} \eta_{k-1,j}^2 \right\}, \end{aligned} \quad (\text{E.473})$$

where the last inequality follows from the  $L$ -smoothness of  $g_t^k$ . Replacing Eq. (E.473) in Eq. (E.468), we have

$$\begin{aligned} T_3 &\leq 32\eta_{k-1}^2 L^4 \cdot \mathbb{E} \left\| \mathbf{u}_t^{k-1} - \bar{\mathbf{u}}^{k-1} \right\|^2 + 8L^2 \sigma^2 \cdot \left\{ \sum_{j=0}^{J-1} \eta_{k-1,j}^2 \right\} \\ &\quad + 32\eta_{k-1}^2 L^2 \cdot \mathbb{E} \left\| \nabla_u g_t^k \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2 + 2L^2 \cdot \mathbb{E} \left\| \bar{\mathbf{u}}^{k-1} - \mathbf{u}_t^{k-1} \right\|^2. \end{aligned} \quad (\text{E.474})$$

For  $\eta_k$  small enough, in particular if  $\eta_k \leq \frac{1}{2\sqrt{2}L}$  we have,

$$T_3 \leq 6L^2 \mathbb{E} \left\| \mathbf{u}_t^{k-1} - \bar{\mathbf{u}}^{k-1} \right\|^2 + 8L^2 \sigma^2 \sum_{j=0}^{J-1} \eta_{k-1,j}^2 + 32\eta_{k-1}^2 L^2 \left\| \nabla_u g_t^k \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2. \quad (\text{E.475})$$

Replacing Eq. (E.475) in Eq. (E.461), we have

$$\begin{aligned} &\mathbb{E} \left[ g^k(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^{k-1}) - g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right] \leq \\ &\quad \frac{3\eta_{k-1} L^2}{T} \cdot \sum_{t=1}^T \mathbb{E} \left\| \mathbf{u}_t^{k-1} - \bar{\mathbf{u}}^{k-1} \right\|^2 + \frac{\eta_{k-1}^2 L}{T} \left( 4 \sum_{j=0}^{J-1} \frac{TL \cdot \eta_{k-1,j}^2}{\eta_{k-1}} + 1 \right) \sigma^2 \\ &\quad - \frac{\eta_{k-1}}{2} \mathbb{E} \left\| \nabla_u g^k \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1} \right) \right\|^2 + \frac{16\eta_{k-1}^3 L^2}{T} \sum_{t=1}^T \left\| \nabla_u g_t^k \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1} \right) \right\|^2. \end{aligned} \quad (\text{E.476})$$

We use now Assumption 14' to bound the last term in the RHS of the previous equation, leading to

$$\begin{aligned} &\mathbb{E} \left[ g^k(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^{k-1}) - g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right] \leq \\ &\quad \frac{3\eta_{k-1} L^2}{T} \cdot \sum_{t=1}^T \mathbb{E} \left\| \mathbf{u}_t^{k-1} - \bar{\mathbf{u}}^{k-1} \right\|^2 + \frac{\eta_{k-1}^2 L}{T} \left( 4 \sum_{j=0}^{J-1} \frac{TL \cdot \eta_{k-1,j}^2}{\eta_{k-1}} + 1 \right) \sigma^2 \end{aligned}$$



$$-\frac{\eta_{k-1} \cdot (1 - 32\eta_{k-1}^2 L^2 \beta^2)}{2} \mathbb{E} \left\| \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 + \frac{16\eta_{k-1}^3 L^2}{T} G^2. \quad (\text{E.477})$$

For  $\eta_{k-1}$  small enough, in particular, if  $\eta_{k-1} \leq \frac{1}{8L\beta}$ , we have

$$\begin{aligned} \mathbb{E} \left[ g^k(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^{k-1}) - g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right] \leq \\ -\frac{\eta_{k-1}}{4} \mathbb{E} \left\| \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 + \frac{3\eta_{k-1} L^2}{T} \cdot \sum_{t=1}^T \mathbb{E} \left\| \mathbf{u}_t^{k-1} - \bar{\mathbf{u}}^{k-1} \right\|^2 \\ + \frac{\eta_{k-1}^2 L}{T} \left( 4 \sum_{j=0}^{J-1} \frac{TL \cdot \eta_{k-1,j}^2}{\eta_{k-1}} + 1 \right) \sigma^2 + \frac{16\eta_{k-1}^3 L^2}{T} G^2. \end{aligned} \quad (\text{E.478})$$

We use Lemma E.17 to get

$$\begin{aligned} \mathbb{E} \left[ g^k(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^{k-1}) - f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right] \leq \\ -\frac{\eta_{k-1}}{8} \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 + \frac{(12+T)\eta_{k-1} L^2}{4T} \cdot \sum_{t=1}^T \mathbb{E} \left\| \mathbf{u}_t^{k-1} - \bar{\mathbf{u}}^{k-1} \right\|^2 \\ + \frac{\eta_{k-1}^2 L}{T} \left( 4 \sum_{j=0}^{J-1} \frac{L \cdot \eta_{k-1,j}^2}{\eta_{k-1}} + 1 \right) \sigma^2 + \frac{16\eta_{k-1}^3 L^2}{T} G^2. \end{aligned} \quad (\text{E.479})$$

Finally, since  $g_t^k$  is a partial first-order surrogate of  $f_t$  near  $\{\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}\}$ , we have

$$\begin{aligned} \mathbb{E} \left[ f(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^k) - f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right] \leq -\frac{1}{T} \sum_{t=1}^T \mathbb{E} d_{\mathcal{V}}(\mathbf{v}_t^k, \mathbf{v}_t^{k-1}) \\ -\frac{\eta_{k-1}}{8} \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 + \frac{(12+T)\eta_{k-1} L^2}{4T} \cdot \sum_{t=1}^T \mathbb{E} \left\| \mathbf{u}_t^{k-1} - \bar{\mathbf{u}}^{k-1} \right\|^2 \\ + \frac{\eta_{k-1}^2 L}{T} \left( 4 \sum_{j=0}^{J-1} \frac{L \cdot \eta_{k-1,j}^2}{\eta_{k-1}} + 1 \right) \sigma^2 + \frac{16\eta_{k-1}^3 L^2}{T} G^2. \end{aligned} \quad (\text{E.480})$$

□

**Lemma E.11** (Recursion for consensus distance, part 1). *Suppose that Assumptions 12'–14' and Assumption 15 hold. For  $k \geq \tau$ , consider  $m = \lfloor * \rfloor \frac{k}{\tau} - 1$  and  $(\eta_{k,j})_{1 \leq j \leq J-1}$  such that  $\eta_k \triangleq \sum_{j=0}^{J-1} \eta_{k,j} \leq \min \left\{ \frac{1}{4L}, \frac{1}{4L\beta} \right\}$  then, the updates of fully decentralized federated surrogate optimization (Alg 11) verify*

$$\begin{aligned} \mathbb{E} \sum_{t=1}^T \left\| \mathbf{u}_t^k - \bar{\mathbf{u}}^k \right\|_F^2 \leq \\ \left(1 - \frac{p}{2}\right) \mathbb{E} \left\| \mathbf{U}^{m\tau} - \bar{\mathbf{U}}^{m\tau} \right\|_F^2 + 44\tau \left(1 + \frac{2}{p}\right) L^2 \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| \mathbf{U}^l - \bar{\mathbf{U}}^l \right\|_F^2 \end{aligned}$$

$$\begin{aligned}
& + T \cdot \sigma^2 \cdot \sum_{l=m\tau}^{k-1} \left\{ \eta_l^2 + 16\tau L^2 \left(1 + \frac{2}{p}\right) \cdot \left\{ \sum_{j=0}^{J-1} \eta_{l,j}^2 \right\} \right\} + 16\tau \left(1 + \frac{2}{p}\right) G^2 \sum_{l=m\tau}^{k-1} \eta_l^2 \\
& + 16\tau \left(1 + \frac{2}{p}\right) \beta^2 \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| \nabla_{\mathbf{u}} f \left( \bar{\mathbf{u}}^{l,j}, \mathbf{v}_{1:T}^l \right) \right\|^2.
\end{aligned}$$

*Proof.* For  $k \geq \tau$ , and  $m = \lfloor * \rfloor_{\frac{k}{\tau}} - 1$ , we have

$$\mathbb{E} \sum_{t=1}^T \left\| \mathbf{u}_t^k - \bar{\mathbf{u}}^k \right\|_F^2 = \mathbb{E} \left\| \mathbf{U}^k - \bar{\mathbf{U}}^k \right\|_F^2 \quad (\text{E.481})$$

$$= \mathbb{E} \left\| \mathbf{U}^k - \bar{\mathbf{U}}^{m\tau} - \left( \bar{\mathbf{U}}^k - \bar{\mathbf{U}}^{m\tau} \right) \right\|_F^2 \quad (\text{E.482})$$

$$\leq \mathbb{E} \left\| \mathbf{U}^k - \bar{\mathbf{U}}^{m\tau} \right\|_F^2, \quad (\text{E.483})$$

where we used the fact that  $\|A - \bar{A}\|_F^2 = \|A \cdot (I - \frac{\mathbf{1}\mathbf{1}^\top}{T})\|_F^2 \leq \|I - \frac{\mathbf{1}\mathbf{1}^\top}{T}\|_2 \cdot \|A\|_F^2 = \|A\|_F^2$  to obtain the last inequality. Using Eq. (E.438) recursively, we have

$$\mathbf{U}^k = \mathbf{U}^{m\tau} \left\{ \prod_{l'=m\tau}^{k-1} W^{l'} \right\} - \sum_{l=m\tau}^{k-1} \eta_l \hat{\Upsilon}^l \left\{ \prod_{l'=l}^{k-1} W^{l'} \right\}. \quad (\text{E.484})$$

Thus,

$$\mathbb{E} \sum_{t=1}^T \left\| \mathbf{u}_t^k - \bar{\mathbf{u}}^k \right\|_F^2 \leq \mathbb{E} \left\| \mathbf{U}^{m\tau} \left\{ \prod_{l'=m\tau}^{k-1} W^{l'} \right\} - \bar{\mathbf{U}}^{m\tau} - \sum_{l=m\tau}^{k-1} \eta_l \hat{\Upsilon}^l \left\{ \prod_{l'=l}^{k-1} W^{l'} \right\} \right\|_F^2 \quad (\text{E.485})$$

$$\begin{aligned}
& = \mathbb{E} \left\| \mathbf{U}^{m\tau} \left\{ \prod_{l'=m\tau}^{k-1} W^{l'} \right\} - \bar{\mathbf{U}}^{m\tau} - \sum_{l=m\tau}^{k-1} \eta_l \Upsilon^l \left\{ \prod_{l'=l}^{k-1} W^{l'} \right\} \right. \\
& \quad \left. + \sum_{l=m\tau}^{k-1} \eta_l (\Upsilon^l - \hat{\Upsilon}^l) \left\{ \prod_{l'=l}^{k-1} W^{l'} \right\} \right\|_F^2 \quad (\text{E.486})
\end{aligned}$$

$$\begin{aligned}
& = \mathbb{E} \left\| \mathbf{U}^{m\tau} \left\{ \prod_{l'=m\tau}^{k-1} W^{l'} \right\} - \bar{\mathbf{U}}^{m\tau} - \sum_{l=m\tau}^{k-1} \eta_l \Upsilon^l \left\{ \prod_{l'=l}^{k-1} W^{l'} \right\} \right\|_F^2 \\
& \quad + \mathbb{E} \left\| \sum_{l=m\tau}^{k-1} \eta_l (\Upsilon^l - \hat{\Upsilon}^l) \left\{ \prod_{l'=l}^{k-1} W^{l'} \right\} \right\|_F^2 \\
& \quad + 2\mathbb{E} \left\langle \mathbf{U}^{m\tau} \left\{ \prod_{l'=m\tau}^{k-1} W^{l'} \right\} - \bar{\mathbf{U}}^{m\tau} - \sum_{l=m\tau}^{k-1} \eta_l \Upsilon^l \left\{ \prod_{l'=l}^{k-1} W^{l'} \right\}, \right. \\
& \quad \quad \left. \sum_{l=m\tau}^{k-1} \eta_l (\Upsilon^l - \hat{\Upsilon}^l) \left\{ \prod_{l'=l}^{k-1} W^{l'} \right\} \right\rangle_F. \quad (\text{E.487})
\end{aligned}$$

Since stochastic gradients are unbiased, the last term in the RHS of the previous equation is equal to zero. Using the following standard inequality for Euclidean norm with  $\alpha > 0$ ,

$$\|\mathbf{a} + \mathbf{b}\|^2 \leq (1 + \alpha) \|\mathbf{a}\|^2 + (1 + \alpha^{-1}) \|\mathbf{b}\|^2, \quad (\text{E.488})$$

we have

$$\mathbb{E} \sum_{t=1}^T \left\| \mathbf{u}_t^k - \bar{\mathbf{u}}^k \right\|_F^2 \leq \quad (\text{E.489})$$

$$\begin{aligned} & (1 + \alpha) \mathbb{E} \left\| \mathbf{U}^{m\tau} \left\{ \prod_{l'=m\tau}^{k-1} W^{l'} \right\} - \bar{\mathbf{U}}^{m\tau} \right\|_F^2 + (1 + \alpha^{-1}) \mathbb{E} \left\| \sum_{l=m\tau}^{k-1} \eta_l \Upsilon^l \left\{ \prod_{l'=l}^{k-1} W^{l'} \right\} \right\|_F^2 \\ & + \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| (\Upsilon^l - \hat{\Upsilon}^l) \left\{ \prod_{l'=l}^{k-1} W^{l'} \right\} \right\|_F^2. \end{aligned} \quad (\text{E.490})$$

Since  $k \geq (m+1)\tau$  and matrices  $(W^l)_{l \geq 0}$  are doubly stochastic, we have

$$\begin{aligned} & \mathbb{E} \sum_{t=1}^T \left\| \mathbf{u}_t^k - \bar{\mathbf{u}}^k \right\|_F^2 \leq \\ & (1 + \alpha) \mathbb{E} \left\| \mathbf{U}^{m\tau} \left\{ \prod_{l'=m\tau}^{(m+1)\tau-1} W^{l'} \right\} - \bar{\mathbf{U}}^{m\tau} \right\|_F^2 + (1 + \alpha^{-1}) \mathbb{E} \left\| \sum_{l=m\tau}^{k-1} \eta_l \Upsilon^l \right\|_F^2 \\ & + \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| \Upsilon^l - \hat{\Upsilon}^l \right\|_F^2 \end{aligned} \quad (\text{E.491})$$

$$\begin{aligned} & \leq (1 + \alpha) \mathbb{E} \left\| \mathbf{U}^{m\tau} \left\{ \prod_{l'=m\tau}^{(m+1)\tau-1} W^{l'} \right\} - \bar{\mathbf{U}}^{m\tau} \right\|_F^2 + (1 + \alpha^{-1}) \cdot (k - m\tau) \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| \Upsilon^l \right\|_F^2 \\ & + \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| \Upsilon^l - \hat{\Upsilon}^l \right\|_F^2, \end{aligned} \quad (\text{E.492})$$

where we use the fact that  $\|AB\|_F \leq \|A\|_2 \|B\|_F$  and that  $\|A\| = 1$  when  $A$  is a doubly stochastic matrix to obtain the first inequality, and Cauchy-Schwarz inequality to obtain the second one. Using Assumption 15 to bound the first term of the RHS of the previous equation and the fact that that  $k \leq (m+2)\tau$ , it follows that

$$\begin{aligned} & \mathbb{E} \sum_{t=1}^T \left\| \mathbf{u}_t^k - \bar{\mathbf{u}}^k \right\|_F^2 \leq \\ & (1 + \alpha)(1 - p) \mathbb{E} \left\| \mathbf{U}^{m\tau} - \bar{\mathbf{U}}^{m\tau} \right\|_F^2 + 2\tau (1 + \alpha^{-1}) \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| \Upsilon^l \right\|_F^2 \\ & + \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| \Upsilon^l - \hat{\Upsilon}^l \right\|_F^2. \end{aligned} \quad (\text{E.493})$$

We use the fact that stochastic gradients have bounded variance (Assumption 13') to bound  $\mathbb{E} \left\| \Upsilon^l - \hat{\Upsilon}^l \right\|_F^2$  as follows,

$$\mathbb{E} \left\| \Upsilon^l - \hat{\Upsilon}^l \right\|_F^2 = \sum_{t=1}^T \mathbb{E} \left\| \delta_t^l - \hat{\delta}_t^l \right\|_F^2 \quad (\text{E.494})$$

$$= \sum_{t=1}^T \mathbb{E} \left\| \sum_{j=0}^{J-1} \frac{\eta_{l,j}}{\eta_l} \cdot \left( \nabla_{\mathbf{u}} g_t^{l+1}(\mathbf{u}_t^{l,j}, \mathbf{v}_t^{k-1}) - \nabla_{\mathbf{u}} g_t^{l+1}(\mathbf{u}_t^{l,j}, \mathbf{v}_t^l; \xi_t^{l,j}) \right) \right\|^2 \quad (\text{E.495})$$

$$\leq \sum_{t=1}^T \sum_{j=0}^{J-1} \frac{\eta_{l,j}}{\eta_l} \cdot \mathbb{E} \left\| \left( \nabla_{\mathbf{u}} g_t^{l+1}(\mathbf{u}_t^{l,j}, \mathbf{v}_t^{k-1}) - \nabla_{\mathbf{u}} g_t^{l+1}(\mathbf{u}_t^{l,j}, \mathbf{v}_t^l; \xi_t^{l,j}) \right) \right\|^2 \quad (\text{E.496})$$

$$\leq \sum_{t=1}^T \sum_{j=0}^{J-1} \frac{\eta_{l,j}}{\eta_l} \sigma^2 \quad (\text{E.497})$$

$$= T \cdot \sigma^2, \quad (\text{E.498})$$

where we used Jensen inequality to obtain the first inequality and Assumption 13' to obtain the second inequality. Replacing back in Eq. (E.493), we have

$$\begin{aligned} \mathbb{E} \sum_{t=1}^T \left\| \mathbf{u}_t^k - \bar{\mathbf{u}}^k \right\|_F^2 &\leq \\ (1 + \alpha)(1 - p) \mathbb{E} \left\| \mathbf{U}^{m\tau} - \bar{\mathbf{U}}^{m\tau} \right\|_F^2 &+ 2\tau (1 + \alpha^{-1}) \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| \Upsilon^l \right\|_F^2 + T \cdot \sigma^2 \cdot \left\{ \sum_{l=m\tau}^{k-1} \eta_l^2 \right\}. \end{aligned} \quad (\text{E.499})$$

The last step of the proof consists in bounding  $\mathbb{E} \left\| \Upsilon^l \right\|_F^2$  for  $l \in \{m\tau, \dots, k-1\}$ ,

$$\mathbb{E} \left\| \Upsilon^l \right\|_F^2 = \sum_{t=1}^T \mathbb{E} \left\| \delta_t^l \right\|^2 \quad (\text{E.500})$$

$$= \sum_{t=1}^T \mathbb{E} \left\| \sum_{j=0}^{J-1} \frac{\eta_{l,j}}{\eta_l} \cdot \nabla_{\mathbf{u}} g_t^{l+1}(\mathbf{u}_t^{l,j}, \mathbf{v}_t^l) \right\|^2 \quad (\text{E.501})$$

$$\leq \sum_{t=1}^T \sum_{j=0}^{J-1} \frac{\eta_{l,j}}{\eta_l} \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^{l+1}(\mathbf{u}_t^{l,j}, \mathbf{v}_t^l) \right\|^2 \quad (\text{E.502})$$

$$\leq \sum_{t=1}^T \sum_{j=0}^{J-1} \frac{\eta_{l,j}}{\eta_l} \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^{l+1}(\mathbf{u}_t^{l,j}, \mathbf{v}_t^l) - \nabla_{\mathbf{u}} f_t(\mathbf{u}_t^l, \mathbf{v}_t^l) + \nabla_{\mathbf{u}} f_t(\mathbf{u}_t^l, \mathbf{v}_t^l) \right\|^2 \quad (\text{E.503})$$

$$\begin{aligned} &\leq 2 \sum_{t=1}^T \sum_{j=0}^{J-1} \frac{\eta_{l,j}}{\eta_l} \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^{l+1}(\mathbf{u}_t^{l,j}, \mathbf{v}_t^l) - \nabla_{\mathbf{u}} f_t(\mathbf{u}_t^l, \mathbf{v}_t^l) \right\|^2 \\ &\quad + 2 \sum_{t=1}^T \mathbb{E} \left\| \nabla_{\mathbf{u}} f_t(\mathbf{u}_t^l, \mathbf{v}_t^l) \right\|^2. \end{aligned} \quad (\text{E.504})$$

Since  $g_t^{l+1}$  is a first order surrogate of  $f$  near  $\{\mathbf{u}_t^l, \mathbf{v}_t^l\}$ , we have

$$\mathbb{E} \left\| \Upsilon^l \right\|_F^2 \leq 2 \sum_{t=1}^T \sum_{j=0}^{J-1} \frac{\eta_{l,j}}{\eta_l} \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^{l+1}(\mathbf{u}_t^{l,j}, \mathbf{v}_t^l) - \nabla_{\mathbf{u}} g_t^{l+1}(\mathbf{u}_t^{l,0}, \mathbf{v}_t^l) \right\|^2$$

$$+ 2 \sum_{t=1}^T \mathbb{E} \left\| \nabla_{\mathbf{u}} f_t(\mathbf{u}_t^l, \mathbf{v}_t^l) - \nabla_{\mathbf{u}} f_t(\bar{\mathbf{u}}^l, \mathbf{v}_t^l) + \nabla_{\mathbf{u}} f_t(\bar{\mathbf{u}}^l, \mathbf{v}_t^l) \right\|^2 \quad (\text{E.505})$$

$$\leq 2 \sum_{t=1}^T \sum_{j=0}^{J-1} \frac{\eta_{l,j}}{\eta_l} \cdot \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^{l+1}(\mathbf{u}_t^{l,j}, \mathbf{v}_t^l) - \nabla_{\mathbf{u}} g_t^{l+1}(\mathbf{u}_t^{l,0}, \mathbf{v}_t^l) \right\|^2 \\ + 4 \sum_{t=1}^T \mathbb{E} \left\| \nabla_{\mathbf{u}} f_t(\mathbf{u}_t^l, \mathbf{v}_t^l) - \nabla_{\mathbf{u}} f_t(\bar{\mathbf{u}}^l, \mathbf{v}_t^l) \right\|^2 + 4 \sum_{t=1}^T \mathbb{E} \left\| \nabla_{\mathbf{u}} f_t(\bar{\mathbf{u}}^l, \mathbf{v}_t^l) \right\|^2. \quad (\text{E.506})$$

Since  $f$  is  $2L$ -smooth w.r.t  $\mathbf{u}$  (Lemma E.15) and  $g$  is  $L$ -smooth w.r.t  $\mathbf{u}$  (Assumption 12'), we have

$$\mathbb{E} \left\| \Upsilon^l \right\|_F^2 \leq 2 \sum_{t=1}^T \sum_{j=0}^{J-1} \frac{\eta_{l,j}}{\eta_l} \cdot L^2 \mathbb{E} \left\| \mathbf{u}_t^{l,j} - \mathbf{u}_t^{l,0} \right\|^2 + 16L^2 \cdot \sum_{t=1}^T \mathbb{E} \left\| \mathbf{u}_t^l - \bar{\mathbf{u}}^l \right\|^2 \\ + 4 \sum_{t=1}^T \mathbb{E} \left\| \nabla_{\mathbf{u}} f_t(\bar{\mathbf{u}}^l, \mathbf{v}_t^l) \right\|^2. \quad (\text{E.507})$$

We use Eq. (E.473) to bound the first term in the RHS of the previous equation, leading to

$$\mathbb{E} \left\| \Upsilon^l \right\|_F^2 \leq 32\eta_l^2 L^2 \sum_{t=1}^T \mathbb{E} \left\| \nabla_{\mathbf{u}} g_t^{l+1}(\bar{\mathbf{u}}^{l,j}, \mathbf{v}_t^l) \right\|^2 + 16L^2 (1 + 2\eta_l^2 L^2) \cdot \sum_{t=1}^T \mathbb{E} \left\| \mathbf{u}_t^l - \bar{\mathbf{u}}^l \right\|^2 \\ + 4 \sum_{t=1}^T \mathbb{E} \left\| \nabla_{\mathbf{u}} f_t(\bar{\mathbf{u}}^l, \mathbf{v}_t^l) \right\|^2 + 8TL^2 \sigma^2 \cdot \left\{ \sum_{j=0}^{J-1} \eta_{l,j}^2 \right\}. \quad (\text{E.508})$$

Using Lemma E.17, we have

$$\mathbb{E} \left\| \Upsilon^l \right\|_F^2 \leq 4 (1 + 16\eta_l^2 L^2) \cdot \sum_{t=1}^T \mathbb{E} \left\| \nabla_{\mathbf{u}} f_t(\bar{\mathbf{u}}^{l,j}, \mathbf{v}_t^l) \right\|^2 \\ + 16L^2 (1 + 6\eta_l^2 L^2) \cdot \sum_{t=1}^T \mathbb{E} \left\| \mathbf{u}_t^l - \bar{\mathbf{u}}^l \right\|^2 + 8L^2 \sigma^2 T \cdot \left\{ \sum_{j=0}^{J-1} \eta_{l,j}^2 \right\}. \quad (\text{E.509})$$

For  $\eta_l$  small enough, in particular, for  $\eta_l \leq \frac{1}{4L}$ , we have

$$\mathbb{E} \left\| \Upsilon^l \right\|_F^2 \leq 8 \sum_{t=1}^T \mathbb{E} \left\| \nabla_{\mathbf{u}} f_t(\bar{\mathbf{u}}^{l,j}, \mathbf{v}_t^l) \right\|^2 + 22L^2 \mathbb{E} \left\| \mathbf{U}^l - \bar{\mathbf{U}}^l \right\|_F^2 + 8L^2 \sigma^2 T \left\{ \sum_{j=0}^{J-1} \eta_{l,j}^2 \right\}. \quad (\text{E.510})$$

Replacing Eq. (E.510) in Eq. (E.499), we have

$$\mathbb{E} \sum_{t=1}^T \left\| \mathbf{u}_t^k - \bar{\mathbf{u}}^k \right\|_F^2 \leq \\ (1 + \alpha)(1 - p) \mathbb{E} \left\| \mathbf{U}^{m\tau} - \bar{\mathbf{U}}^{m\tau} \right\|_F^2 + 44\tau (1 + \alpha^{-1}) L^2 \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| \mathbf{U}^l - \bar{\mathbf{U}}^l \right\|_F^2 \\ + 16\tau (1 + \alpha^{-1}) \sum_{l=m\tau}^{k-1} \eta_l^2 \sum_{t=1}^T \mathbb{E} \left\| \nabla_{\mathbf{u}} f_t(\bar{\mathbf{u}}^{l,j}, \mathbf{v}_t^l) \right\|^2$$

$$+ T \cdot \sigma^2 \cdot \sum_{l=m\tau}^{k-1} \left\{ \eta_l^2 + 16\tau L^2 (1 + \alpha^{-1}) \cdot \left\{ \sum_{j=0}^{J-1} \eta_{l,j}^2 \right\} \right\}. \quad (\text{E.511})$$

Using Lemma E.16 and considering  $\alpha = \frac{p}{2}$ , we have

$$\begin{aligned} \mathbb{E} \sum_{t=1}^T \left\| \mathbf{u}_t^k - \bar{\mathbf{u}}^k \right\|_F^2 &\leq \\ &(1 - \frac{p}{2}) \mathbb{E} \left\| \mathbf{U}^{m\tau} - \bar{\mathbf{U}}^{m\tau} \right\|_F^2 + 44\tau \left(1 + \frac{2}{p}\right) L^2 \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| \mathbf{U}^l - \bar{\mathbf{U}}^l \right\|_F^2 \\ &+ T \cdot \sigma^2 \cdot \sum_{l=m\tau}^{k-1} \left\{ \eta_l^2 + 16\tau L^2 \left(1 + \frac{2}{p}\right) \cdot \left\{ \sum_{j=0}^{J-1} \eta_{l,j}^2 \right\} \right\} + 16\tau \left(1 + \frac{2}{p}\right) G^2 \sum_{l=m\tau}^{k-1} \eta_l^2 \\ &+ 16\tau \left(1 + \frac{2}{p}\right) \beta^2 \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| \nabla_{\mathbf{u}} f \left( \bar{\mathbf{u}}^{l,j}, \mathbf{v}_{1:T}^l \right) \right\|^2. \end{aligned} \quad (\text{E.512})$$

□

**Lemma E.12** (Recursion for consensus distance, part 2). *Suppose that Assumptions 12'–14' and Assumption 15 hold. Consider  $m = \lfloor * \rfloor_{\tau}^k$ , then, for  $(\eta_{k,j})_{1 \leq j \leq J-1}$  such that  $\eta_k \triangleq \sum_{j=0}^{J-1} \eta_{k,j} \leq \min \left\{ \frac{1}{4L}, \frac{1}{4L\beta} \right\}$ , the updates of fully decentralized federated surrogate optimization (Alg 11) verify*

$$\begin{aligned} \mathbb{E} \sum_{t=1}^T \left\| \mathbf{u}_t^k - \bar{\mathbf{u}}^k \right\|_F^2 &\leq \\ &(1 + \frac{p}{2}) \mathbb{E} \left\| \mathbf{U}^{m\tau} - \bar{\mathbf{U}}^{m\tau} \right\|_F^2 + 44\tau \left(1 + \frac{2}{p}\right) L^2 \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| \mathbf{U}^l - \bar{\mathbf{U}}^l \right\|_F^2 \\ &+ T \cdot \sigma^2 \cdot \sum_{l=m\tau}^{k-1} \left\{ \eta_l^2 + 16\tau L^2 \left(1 + \frac{2}{p}\right) \cdot \left\{ \sum_{j=0}^{J-1} \eta_{l,j}^2 \right\} \right\} + 16\tau \left(1 + \frac{2}{p}\right) G^2 \sum_{l=m\tau}^{k-1} \eta_l^2 \\ &+ 16\tau \left(1 + \frac{2}{p}\right) \beta^2 \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| \nabla_{\mathbf{u}} f \left( \bar{\mathbf{u}}^{l,j}, \mathbf{v}_{1:T}^l \right) \right\|^2. \end{aligned} \quad (\text{E.513})$$

*Proof.* We use exactly the same proof as in Lemma E.11, with the only difference that Eq. (E.491)–Eq. (E.493) is replaced by

$$\begin{aligned} \mathbb{E} \sum_{t=1}^T \left\| \mathbf{u}_t^k - \bar{\mathbf{u}}^k \right\|_F^2 &\leq \\ &(1 + \alpha) \mathbb{E} \left\| \mathbf{U}^{m\tau} - \bar{\mathbf{U}}^{m\tau} \right\|_F^2 + 2\tau (1 + \alpha^{-1}) \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| \Upsilon^l \right\|_F^2 \\ &+ \sum_{l=m\tau}^{k-1} \eta_l^2 \mathbb{E} \left\| \Upsilon^l - \hat{\Upsilon}^l \right\|_F^2, \end{aligned} \quad (\text{E.514})$$

resulting from the fact that  $\left\{ \prod_{l'=m\tau}^{(m+1)\tau-1} W^{l'} \right\}$  is a doubly stochastic matrix. □

**Lemma E.13.** Under Assum. 12'-14' and Assum 15. For  $\eta_{k,j} = \frac{\eta}{j}$  with

$$\eta \leq \min \left\{ \frac{1}{4L}, \frac{p}{92\tau L}, \frac{1}{4\beta L}, \frac{1}{32\sqrt{2}} \cdot \frac{p}{\tau\beta} \right\},$$

the iterates of Alg. 11 verifies

$$\frac{(12+T)L^2}{4T} \sum_{k=0}^K \mathbb{E} \left\| \mathbf{U}^k - \bar{\mathbf{U}}^k \right\|_F^2 \leq \frac{1}{16} \sum_{k=0}^K \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^k) \right\|^2 + 16A \cdot \frac{12+T}{T} \cdot \frac{\tau L^2}{p} (K+1)\eta^2, \quad (\text{E.515})$$

for some constant  $A > 0$  and  $K > 0$ .

*Proof.* Note that for  $k > 0$ ,  $\eta_k = \sum_{j=0}^{k-1} \eta_{k,j} = \eta$ , and that  $\sum_{l=m\tau}^{k-1} \eta_l^2 = \sum_{l=m\tau}^{k-1} \eta^2 \leq 2\tau \cdot \eta^2$

Using Lemma E.11 and Lemma E.12, and the fact that  $p \leq 1$ , we have for  $m = \lfloor * \rfloor_{\tau}^{\frac{k}{\tau}} - 1$

$$\begin{aligned} \mathbb{E} \left\| \mathbf{U}^k - \bar{\mathbf{U}}^k \right\|_F^2 &\leq \left(1 - \frac{p}{2}\right) \mathbb{E} \left\| \mathbf{U}^{m\tau} - \bar{\mathbf{U}}^{m\tau} \right\|_F^2 + \frac{132\tau}{p} L^2 \eta^2 \sum_{l=m\tau}^{k-1} \mathbb{E} \left\| \mathbf{U}^l - \bar{\mathbf{U}}^l \right\|_F^2 \\ &\quad + \underbrace{\eta^2 2\tau \left\{ T\sigma^2 \left(1 + \frac{16\tau L^2}{J} \left(1 + \frac{2}{p}\right)\right) + 16\tau \left(1 + \frac{2}{p}\right) G^2 \right\}}_{\triangleq A} \\ &\quad + \frac{16\tau}{p} \beta^2 \eta^2 \sum_{l=m\tau}^{k-1} \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^l, \mathbf{v}_{1:T}^l) \right\|^2. \end{aligned} \quad (\text{E.516})$$

and for  $m = \lfloor * \rfloor_{\tau}^{\frac{k}{\tau}}$ ,

$$\begin{aligned} \mathbb{E} \left\| \mathbf{U}^k - \bar{\mathbf{U}}^k \right\|_F^2 &\leq \left(1 + \frac{p}{2}\right) \mathbb{E} \left\| \mathbf{U}^{m\tau} - \bar{\mathbf{U}}^{m\tau} \right\|_F^2 + \frac{132\tau}{p} L^2 \eta^2 \sum_{l=m\tau}^{k-1} \mathbb{E} \left\| \mathbf{U}^l - \bar{\mathbf{U}}^l \right\|_F^2 \\ &\quad + \underbrace{\eta^2 2\tau \left\{ T\sigma^2 \left(1 + \frac{16\tau L^2}{J} \left(1 + \frac{2}{p}\right)\right) + 16\tau \left(1 + \frac{2}{p}\right) G^2 \right\}}_{\triangleq A} \\ &\quad + \underbrace{\frac{16\tau}{p} \beta^2 \eta^2 \sum_{l=m\tau}^{k-1} \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^l, \mathbf{v}_{1:T}^l) \right\|^2}_{\triangleq D}. \end{aligned} \quad (\text{E.517})$$

Using the fact that  $\eta \leq \frac{p}{92\tau L}$ , it follows that for  $m = \lfloor * \rfloor_{\tau}^{\frac{k}{\tau}} - 1$

$$\begin{aligned} \mathbb{E} \left\| \mathbf{U}^k - \bar{\mathbf{U}}^k \right\|_F^2 &\leq \left(1 - \frac{p}{2}\right) \mathbb{E} \left\| \mathbf{U}^{m\tau} - \bar{\mathbf{U}}^{m\tau} \right\|_F^2 + \frac{p}{64\tau} \sum_{l=m\tau}^{k-1} \mathbb{E} \left\| \mathbf{U}^l - \bar{\mathbf{U}}^l \right\|_F^2 \\ &\quad + \eta^2 A + D\eta^2 \sum_{l=m\tau}^{k-1} \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^l, \mathbf{v}_{1:T}^l) \right\|^2, \end{aligned} \quad (\text{E.518})$$

and for  $m = \lfloor * \rfloor_{\tau}^{\frac{k}{\tau}}$ ,

$$\mathbb{E} \left\| \mathbf{U}^k - \bar{\mathbf{U}}^k \right\|_F^2 \leq \left(1 + \frac{p}{2}\right) \mathbb{E} \left\| \mathbf{U}^{m\tau} - \bar{\mathbf{U}}^{m\tau} \right\|_F^2 + \frac{p}{64\tau} \sum_{l=m\tau}^{k-1} \mathbb{E} \left\| \mathbf{U}^l - \bar{\mathbf{U}}^l \right\|_F^2$$

$$+\eta^2 A + D\eta^2 \sum_{l=m\tau}^{k-1} \mathbb{E} \left\| \nabla_{\mathbf{u}} f \left( \bar{\mathbf{u}}^l, \mathbf{v}_{1:T}^l \right) \right\|^2. \quad (\text{E.519})$$

The rest of the proof follows using [Kol+20, Lemma 14] with  $B = \frac{(12+T)L^2}{4T}$ ,  $b = \frac{1}{8}$ , constant (thus  $\frac{8\tau}{p}$ -slow\*) steps-size  $\eta \leq \frac{1}{32\sqrt{2}} \frac{p}{\tau\beta} = \frac{1}{16} \sqrt{\frac{p/8}{D\tau}}$  and constant weights  $\omega_k = 1$ .  $\square$

**Theorem 3.5.4'.** *Under Assumptions 11'–14' and Assumption 15, when clients use SGD as local solver with learning rate  $\eta = \frac{a_0}{\sqrt{K}}$ , after a large enough number of communication rounds  $K$ , the iterates of fully decentralized federated surrogate optimization (Alg. 11) satisfy:*

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \left\| \nabla_{\mathbf{u}} f \left( \bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^k \right) \right\|^2 \leq \mathcal{O} \left( \frac{1}{\sqrt{K}} \right), \quad (\text{E.520})$$

and,

$$\frac{1}{K} \sum_{k=1}^K \sum_{t=1}^T \omega_t \cdot \mathbb{E} d_{\mathcal{V}} \left( \mathbf{v}_t^k, \mathbf{v}_t^{k+1} \right) \leq \mathcal{O} \left( \frac{1}{K} \right), \quad (\text{E.521})$$

where  $\bar{\mathbf{u}}^k = \frac{1}{T} \sum_{t=1}^T \mathbf{u}_t^k$ . Moreover, local estimates  $\left( \mathbf{u}_t^k \right)_{1 \leq t \leq T}$  converge to consensus, i.e., to  $\bar{\mathbf{u}}^k$ :

$$\frac{1}{K} \sum_{k=1}^K \sum_{t=1}^T \mathbb{E} \left\| \mathbf{u}_t^k - \bar{\mathbf{u}}^k \right\|^2 \leq \mathcal{O} \left( \frac{1}{\sqrt{K}} \right). \quad (\text{E.522})$$

*Proof.* We prove first the convergence to a stationary point in  $\mathbf{u}$ , i.e. Eq. (E.520), using [Kol+20, Lemma 17], then we prove Eq. (E.521) and Eq. (E.522).

Note that for  $K$  large enough,  $\eta \leq \min \left\{ \frac{1}{4L}, \frac{p}{92\tau L}, \frac{1}{4\beta L}, \frac{1}{32\sqrt{2}} \cdot \frac{p}{\tau\beta} \right\}$ .

**Proof of Eq. E.520.** Rearranging the terms in the result of Lemma E.10 and dividing it by  $\eta$  we have

$$\begin{aligned} \frac{1}{\eta} \cdot \mathbb{E} \left[ f(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^k) - f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right] &\leq -\frac{1}{8} \mathbb{E} \left\| \nabla_{\mathbf{u}} f \left( \bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1} \right) \right\|^2 \\ &+ \frac{(12+T)L^2}{4T} \cdot \mathbb{E} \left\| \mathbf{U}^{k-1} - \bar{\mathbf{U}}^{k-1} \right\|^2 + \frac{\eta L}{T} \left( \frac{4L}{J} + 1 \right) \sigma^2 + \frac{16\eta^2 L^2}{T} G^2. \end{aligned} \quad (\text{E.523})$$

Summing over  $k \in [K+1]$ , we have

$$\begin{aligned} \frac{1}{\eta} \cdot \mathbb{E} \left[ f(\bar{\mathbf{u}}^{K+1}, \mathbf{v}_{1:T}^{K+1}) - f(\bar{\mathbf{u}}^0, \mathbf{v}_{1:T}^0) \right] &\leq -\frac{1}{8} \sum_{k=0}^K \mathbb{E} \left\| \nabla_{\mathbf{u}} f \left( \bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^k \right) \right\|^2 \\ &+ \frac{(12+T)L^2}{4T} \cdot \sum_{k=0}^K \mathbb{E} \left\| \mathbf{U}^k - \bar{\mathbf{U}}^k \right\|^2 + \frac{(K+1)\eta L}{T} \left( \frac{4L}{J} + 1 \right) \sigma^2 \\ &+ \frac{16(K+1) \cdot \eta^2 L^2}{T} G^2. \end{aligned} \quad (\text{E.524})$$

\*The notion of  $\tau$ -slow decreasing sequence is defined in [Kol+20, Defintion 2].



Using Lemma E.13, we have

$$\begin{aligned} \frac{1}{\eta} \cdot \mathbb{E} \left[ f(\bar{\mathbf{u}}^{K+1}, \mathbf{v}_{1:T}^{K+1}) - f(\bar{\mathbf{u}}^0, \mathbf{v}_{1:T}^0) \right] &\leq -\frac{1}{16} \sum_{k=0}^K \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^k) \right\|^2 \\ &+ 16A \cdot \frac{12+T}{T} \cdot \frac{\tau L^2}{p} (K+1)\eta^2 + \frac{(K+1)\eta L}{T} \left( \frac{4L}{J} + 1 \right) \sigma^2 \\ &+ \frac{16(K+1)\eta^2 L^2}{T} G^2. \end{aligned} \quad (\text{E.525})$$

Using Assumption 11', it follows that

$$\begin{aligned} \frac{1}{16} \sum_{k=0}^K \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^k) \right\|^2 &\leq \frac{f(\bar{\mathbf{u}}^0, \mathbf{v}_{1:T}^0) - f^*}{\eta} \\ &+ 16A \cdot \frac{12+T}{T} \cdot \frac{\tau L^2}{p} (K+1)\eta^2 + \frac{(K+1)\eta L}{T} \left( \frac{4L}{J} + 1 \right) \sigma^2 + \frac{16(K+1)\eta^2 L^2}{T} G^2. \end{aligned} \quad (\text{E.526})$$

We divide by  $K+1$  and we have

$$\begin{aligned} \frac{1}{16(K+1)} \sum_{k=0}^K \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^k) \right\|^2 &\leq \frac{f(\bar{\mathbf{u}}^0, \mathbf{v}_{1:T}^0) - f^*}{\eta(K+1)} \\ &+ 16A \cdot \frac{12+T}{T} \cdot \frac{\tau L^2}{p} \eta^2 + \frac{\eta L}{T} \left( \frac{4L}{J} + 1 \right) \sigma^2 + \frac{16\eta^2 L^2}{T} G^2. \end{aligned} \quad (\text{E.527})$$

The final result follows from [Kol+20, Lemma 17].

**Proof of Eq. E.522.** We multiply Eq. (E.515) (Lemma E.13) by  $\frac{1}{K+1}$ , and we have

$$\frac{1}{K+1} \sum_{k=0}^K \mathbb{E} \left\| \mathbf{U}^k - \bar{\mathbf{U}}^k \right\|_F^2 \leq \frac{1}{16(K+1)} \sum_{k=0}^K \mathbb{E} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^k) \right\|_F^2 + \frac{64A\tau}{p(K+1)} K\eta^2, \quad (\text{E.528})$$

since  $\eta \leq \mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$ , using Eq. (E.520), it follows that

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \left\| \mathbf{U}^k - \bar{\mathbf{U}}^k \right\|_F^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{K}}\right). \quad (\text{E.529})$$

Thus,

$$\frac{1}{K} \sum_{k=1}^K \sum_{t=1}^T \mathbb{E} \left\| \mathbf{u}_t^k - \bar{\mathbf{u}}^k \right\|_F^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{K}}\right). \quad (\text{E.530})$$

**Proof of Eq. E.521.** Using the result of Lemma E.10 we have

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ d_{\mathcal{V}}(\mathbf{v}_t^k, \mathbf{v}_t^{k-1}) \right] \leq \mathbb{E} \left[ f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) - f(\bar{\mathbf{u}}^k, \mathbf{v}_{1:T}^k) \right]$$

$$\begin{aligned}
& + \frac{(12+T)\eta_{k-1}L^2}{4T} \cdot \sum_{t=1}^T \mathbb{E} \left\| \mathbf{u}_t^{k-1} - \bar{\mathbf{u}}^{k-1} \right\|^2 \\
& + \frac{\eta_{k-1}^2 L}{T} \left( 4 \sum_{j=0}^{J-1} \frac{L \cdot \eta_{k-1,j}^2}{\eta_{k-1}} + 1 \right) \sigma^2 + \frac{16\eta_{k-1}^3 L^2}{T} G^2. \quad (\text{E.531})
\end{aligned}$$

The final result follows from the fact that  $\eta = \mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$  and Eq. (E.522).  $\square$

### E.4.3 Proof of Theorem 3.5.4

We state the formal version of Theorem 3.5.4, for which only an informal version was given in the main text.

**Theorem 3.5.4.** *Under Assumptions 8–15, when clients use SGD as local solver with learning rate  $\eta = \frac{\alpha_0}{\sqrt{K}}$ , D-FedEM's iterates satisfy the following inequalities after a large enough number of communication rounds  $K$ :*

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \left\| \nabla_{\Theta} f(\bar{\Theta}^k, \Pi^k) \right\|_F^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{K}}\right), \quad \frac{1}{K} \sum_{k=1}^K \sum_{t=1}^T \frac{n_t}{n} \mathcal{KL}(\pi_t^k, \pi_t^{k-1}) \leq \mathcal{O}\left(\frac{1}{K}\right), \quad (\text{E.532})$$

where  $\bar{\Theta}^k = [\Theta_1^k, \dots, \Theta_T^k] \cdot \frac{\mathbf{1}\mathbf{1}^\top}{T}$ . Moreover, individual estimates  $(\Theta_t^k)_{1 \leq t \leq T}$  converge to consensus, i.e., to  $\bar{\Theta}^k$ :

$$\min_{k \in [K]} \mathbb{E} \sum_{t=1}^T \left\| \Theta_t^k - \bar{\Theta}^k \right\|_F^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{K}}\right).$$

*Proof.* We prove this result as a particular case of Theorem 3.5.4'. To this purpose, we consider that  $\mathcal{V} \triangleq \Delta^M$ ,  $\mathbf{u} = \Theta \in \mathbb{R}^{dM}$ ,  $\mathbf{v}_t = \pi_t$ , and  $\omega_t = n_t/n$  for  $t \in [T]$ . For  $k > 0$ , we define  $g_t^k$  as follow,

$$\begin{aligned}
g_t^k(\Theta, \pi_t) = \frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{m=1}^M q_t^k(z_t^{(i)} = m) \cdot \left( l(h_{\theta_m}(\mathbf{x}_t^{(i)}), y_t^{(i)}) - \log p_m(\mathbf{x}_t^{(i)}) - \log \pi_t \right. \\
\left. + \log q_t^k(z_t^{(i)} = m) - c \right), \quad (\text{E.533})
\end{aligned}$$

where  $c$  is the same constant appearing in Assumption 10, Eq. (3.3). With this definition, it is easy to check that the federated surrogate optimization algorithm (Alg. 11) reduces to D-FedEM (Alg. 8). Theorem 3.5.4 then follows immediately from Theorem 3.5.4', once we verify that  $(g_t^k)_{1 \leq t \leq T}$  satisfy the assumptions of Theorem 3.5.4'.

Assumption 11', Assumption 13', and Assumption 14' follow directly from Assumption 11, Assumption 13, and Assumption 14, respectively. Lemma E.6 shows that for  $k > 0$ ,  $g^k$  is smooth w.r.t.  $\Theta$  and then Assumption 12' is satisfied. Finally, Lemmas E.7–E.9 show that for  $t \in [T]$   $g_t^k$  is a partial first-order surrogate of  $f_t$  near  $\{\Theta_t^{k-1}, \pi_t\}$  with  $d_{\mathcal{V}}(\cdot, \cdot) = \mathcal{KL}(\cdot \| \cdot)$ .  $\square$

### E.5 Proof of Theorem 3.5.5'

Combining the previous lemmas we prove the convergence of Alg. 10 with a black box solver.

**Theorem 3.5.5'.** *Suppose that Assumptions 11'–14', Assumptions 16' and 17' hold with  $G^2 = 0$  and  $\alpha \leq \frac{1}{\beta^2 \kappa^4}$ , then the updates of federated surrogate optimization (Alg. 10) converge to a stationary point of  $f$ , i.e.,*

$$\lim_{k \rightarrow +\infty} \left\| \nabla_{\mathbf{u}} f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\|^2 = 0, \quad (\text{E.534})$$

and,

$$\lim_{k \rightarrow +\infty} \sum_{t=1}^T \omega_t \cdot d_{\mathcal{V}}(\mathbf{v}_t^k, \mathbf{v}_t^{k-1}) = 0. \quad (\text{E.535})$$

*Proof.*

$$f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) = g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k). \quad (\text{E.536})$$

Computing the gradient norm, we have,

$$\left\| \nabla_{\mathbf{u}} f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\| = \left\| \nabla_{\mathbf{u}} g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - \nabla_{\mathbf{u}} r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\| \quad (\text{E.537})$$

$$\leq \left\| \nabla_{\mathbf{u}} g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\| + \left\| \nabla_{\mathbf{u}} r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\|. \quad (\text{E.538})$$

Since  $g^k$  is  $L$ -smooth in  $\mathbf{u}$ , we write

$$\left\| \nabla_{\mathbf{u}} g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\| = \left\| \nabla_{\mathbf{u}} g^k(\mathbf{u}^k, \mathbf{v}^k) - \nabla_{\mathbf{u}} g^k(\mathbf{u}_*, \mathbf{v}_{1:T}^k) \right\| \quad (\text{E.539})$$

$$\leq L \left\| \mathbf{u}^k - \mathbf{u}_* \right\|. \quad (\text{E.540})$$

Thus by replacing Eq. (E.540) in Eq. (E.538), we have

$$\left\| \nabla_{\mathbf{u}} f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\| \leq L^2 \left\| \mathbf{u}^k - \mathbf{u}_* \right\|^2 + \left\| \nabla_{\mathbf{u}} r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\|. \quad (\text{E.541})$$

Using Lemma E.20, there exists  $0 < \tilde{\alpha} < 1$ , such that

$$\left[ g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - g^k(\mathbf{u}_*, \mathbf{v}_{1:T}^k) \right] \leq \tilde{\alpha} \times \left[ g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) - g^k(\mathbf{u}_*, \mathbf{v}_{1:T}^k) \right]. \quad (\text{E.542})$$

Thus, the conditions of Lemma E.21 hold, and we can use Eq. (E.610) and (E.612), i.e.

$$\left\| \nabla_{\mathbf{u}} r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\|^2 \xrightarrow[k \rightarrow +\infty]{} 0 \quad (\text{E.543})$$

$$\left\| \mathbf{u}^k - \mathbf{u}_* \right\|^2 \xrightarrow[k \rightarrow +\infty]{} 0. \quad (\text{E.544})$$

Finally, combining this with Eq. (E.541), we get the final result

$$\lim_{k \rightarrow +\infty} \left\| \nabla_{\mathbf{u}} f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\| = 0. \quad (\text{E.545})$$

Since  $g_t^k$  is a partial first-order surrogate of  $f_t$  near  $\{\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}\}$  for  $k > 0$  and  $t \in [T]$ , it follows that

$$\sum_{t=1}^T \omega \cdot d_{\mathcal{V}}(\mathbf{v}_t^k, \mathbf{v}_t^{k-1}) = g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) - g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^k) \quad (\text{E.546})$$

$$\leq g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) - g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \quad (\text{E.547})$$

Thus,

$$\sum_{t=1}^T \omega_t \cdot d_{\mathcal{V}}(\mathbf{v}_t^k, \mathbf{v}_t^{k-1}) \leq f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) - f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \quad (\text{E.548})$$

Since  $d_{\mathcal{V}}(\mathbf{v}_t^k, \mathbf{v}_t^{k-1})$  is non-negative for  $k > 0$  and  $t \in [T]$ , it follows that

$$\lim_{k \rightarrow +\infty} \sum_{t=1}^T \omega_t \cdot d_{\mathcal{V}}(\mathbf{v}_t^k, \mathbf{v}_t^{k-1}) = 0 \quad (\text{E.549})$$

□

## E.6 Proof of Theorem 3.5.5

**Theorem 3.5.5.** *Suppose that Assumptions 8–14 and Assumptions 16, 17 hold with  $G^2 = 0$  and  $\alpha \leq \frac{1}{\beta^2 \kappa^5}$ , then the updates of FedEM (Alg. 7) converge to a stationary point of  $f$ , i.e.,*

$$\lim_{k \rightarrow +\infty} \left\| \nabla_{\Theta} f(\Theta^k, \Pi^k) \right\|_F^2 = 0, \quad (\text{E.550})$$

and,

$$\lim_{k \rightarrow +\infty} \sum_{t=1}^T \frac{n_t}{n} \mathcal{KL}(\pi_t^k, \pi_t^{k-1}) = 0. \quad (\text{E.551})$$

*Proof.* We prove this result as a particular case of Theorem 3.5.5'. To this purpose, we consider that  $\mathcal{V} \triangleq \Delta^M$ ,  $u = \Theta \in \mathbb{R}^{dM}$ ,  $v_t = \pi_t$ , and  $\omega_t = n_t/n$  for  $t \in [T]$ . For  $k > 0$ , we define  $g_t^k$  as follow,

$$g_t^k(\Theta, \pi_t) = \frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{m=1}^M q_t^k(z_t^{(i)} = m) \cdot \left( l(h_{\theta_m}(\mathbf{x}_t^{(i)}), y_t^{(i)}) - \log p_m(\mathbf{x}_t^{(i)}) - \log \pi_t + \log q_t^k(z_t^{(i)} = m) - c \right), \quad (\text{E.552})$$

where  $c$  is the same constant appearing in Assumption 10, Eq. (3.3). With this definition, it is easy to check that the federated surrogate optimization algorithm (Alg. 10) reduces to FedEM (Alg. 7). Theorem 3.5.5 then follows immediately from Theorem 3.5.5', once we verify that  $(g_t^k)_{1 \leq t \leq T}$  satisfy the assumptions of Theorem 3.5.5'.

Assumption 11', Assumption 13', Assumption 14', Assumption 16' and Assumption 17' follow directly from Assumption 11, Assumption 13, Assumption 14, Assumption 16 and Assumption 17, respectively. Lemma E.6 shows that for  $k > 0$ ,  $g^k$  is smooth w.r.t.  $\Theta$  and then Assumption 12' is satisfied. Finally, Lemmas E.7–E.9 show that for  $t \in [T]$   $g_t^k$  is a partial first-order surrogate of  $f_t$  w.r.t.  $\Theta$  near  $\{\Theta^{k-1}, \pi_t\}$  with  $d_{\mathcal{V}}(\cdot, \cdot) = \mathcal{KL}(\cdot, \cdot)$ . □

## E.7 Supporting Lemmas

**Lemma E.14.** *Consider  $J \geq 2$  and positive real numbers  $\eta_j$ ,  $j = 0, \dots, J-1$ , then:*

$$\frac{1}{\sum_{j=0}^{J-1} \eta_j} \cdot \sum_{j=0}^{J-1} \left\{ \eta_j \cdot \sum_{l=0}^{j-1} \eta_l \right\} \leq \sum_{j=0}^{J-2} \eta_j,$$

$$\frac{1}{\sum_{j=0}^{J-1} \eta_j} \cdot \sum_{j=0}^{J-1} \left\{ \eta_j \cdot \sum_{l=0}^{j-1} \eta_l^2 \right\} \leq \sum_{j=0}^{J-2} \eta_j^2,$$

$$\frac{1}{\sum_{j=0}^{J-1} \eta_j} \cdot \sum_{j=0}^{J-1} \left\{ \eta_j \cdot \left( \sum_{l=0}^{j-1} \eta_l \right)^2 \right\} \leq \sum_{j=0}^{J-1} \eta_j \cdot \sum_{j=0}^{J-2} \eta_j.$$

*Proof.* For the first inequality,

$$\frac{1}{\sum_{j=0}^{J-1} \eta_j} \cdot \sum_{j=0}^{J-1} \left\{ \eta_j \cdot \sum_{l=0}^{j-1} \eta_l \right\} \leq \frac{1}{\sum_{j=0}^{J-1} \eta_j} \cdot \sum_{j=0}^{J-1} \left\{ \eta_j \cdot \sum_{l=0}^{J-2} \eta_l \right\} = \sum_{l=0}^{J-2} \eta_l. \quad (\text{E.553})$$

For the second inequality

$$\frac{1}{\sum_{j=0}^{J-1} \eta_j} \cdot \sum_{j=0}^{J-1} \left\{ \eta_j \cdot \sum_{l=0}^{j-1} \eta_l^2 \right\} \leq \frac{1}{\sum_{j=0}^{J-1} \eta_j} \cdot \sum_{j=0}^{J-1} \left\{ \eta_j \cdot \sum_{l=0}^{J-2} \eta_l^2 \right\} = \sum_{l=0}^{J-2} \eta_l^2. \quad (\text{E.554})$$

For the third inequality,

$$\frac{1}{\sum_{j=0}^{J-1} \eta_j} \cdot \sum_{j=0}^{J-1} \left\{ \eta_j \cdot \left( \sum_{l=0}^{j-1} \eta_l \right)^2 \right\} \leq \frac{1}{\sum_{j=0}^{J-1} \eta_j} \cdot \sum_{j=0}^{J-1} \left\{ \eta_j \cdot \left( \sum_{l=0}^{J-2} \eta_l \right)^2 \right\} \quad (\text{E.555})$$

$$\leq \left( \sum_{j=0}^{J-2} \eta_j \right)^2 \quad (\text{E.556})$$

$$\leq \sum_{j=0}^{J-1} \eta_j \cdot \sum_{j=0}^{J-2} \eta_j. \quad (\text{E.557})$$

□

**Lemma E.15.** *Suppose that  $g$  is a partial first-order surrogate of  $f$ , and that  $g$  is  $L$ -smooth, where  $L$  is the constant appearing in Definition 1, then  $f$  is  $2L$ -smooth.*

*Proof.* The difference between  $f$  and  $g$  is  $L$ -smooth, and  $g$  is  $L$ -smooth, thus  $f$  is  $2L$ -smooth as the sum of two  $L$ -smooth functions. □

**Lemma E.16.** *Consider  $f = \sum_{t=1}^T \omega_t \cdot f_t$ , for weights  $\omega \in \Delta^T$ . Suppose that for all  $(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^{d_u} \times \mathcal{V}$ , and  $t \in [T]$ ,  $f_t$  admits a partial first-order surrogate  $g_t^{\{\mathbf{u}, \mathbf{v}\}}$  near  $\{\mathbf{u}, \mathbf{v}\}$ , and that  $g^{\{\mathbf{u}, \mathbf{v}\}} = \sum_{t=1}^T \omega_t \cdot g_t^{\{\mathbf{u}, \mathbf{v}\}}$  verifies Assumption 14' for  $t \in [T]$ . Then  $f$  also verifies Assumption 14'.*

*Proof.* Consider arbitrary  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{d_u} \times \mathcal{V}$ , and for  $t \in [T]$ , consider  $g^{\{\mathbf{u}, \mathbf{v}\}}$  to be a partial first-order surrogate of  $f_t$  near  $\{\mathbf{u}, \mathbf{v}\}$ . We write Assumption 14' for  $g^{\{\mathbf{u}, \mathbf{v}\}}$ ,

$$\sum_{t=1}^T \omega_t \cdot \left\| \nabla_{\mathbf{u}} g_t^{\{\mathbf{u}, \mathbf{v}\}}(\mathbf{u}, \mathbf{v}) \right\|^2 \leq G^2 + \beta^2 \left\| \sum_{t=1}^T \omega_t \cdot \nabla_{\mathbf{u}} g_t^{\{\mathbf{u}, \mathbf{v}\}}(\mathbf{u}, \mathbf{v}) \right\|^2. \quad (\text{E.558})$$

Since  $g_t^{\{\mathbf{u}, \mathbf{v}\}}$  is a partial first-order surrogate of  $f_t$  near  $\{u, v\}$ , it follows that

$$\sum_{t=1}^T \omega_t \cdot \left\| \nabla_{\mathbf{u}} f_t(\mathbf{u}, \mathbf{v}) \right\|^2 \leq G^2 + \beta^2 \left\| \sum_{t=1}^T \omega_t \cdot \nabla_{\mathbf{u}} f_t(\mathbf{u}, \mathbf{v}) \right\|^2. \quad (\text{E.559})$$

□

**Remark 12.** Note that the assumption of Lemma E.16 is implicitly verified in Algorithm 10 and Algorithm 11, where we assume that every client  $t \in \mathcal{T}$  can compute a partial first-order surrogate of its local objective  $f_t$  near any iterate  $(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^{d_u} \times \mathcal{V}$ .

**Lemma E.17.** For  $k > 0$ , the iterates of Alg. 11, verify the following inequalities:

$$g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \leq f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) + \frac{L}{2} \sum_{t=1}^T \omega_t \left\| \bar{\mathbf{u}}^{k-1} - \mathbf{u}_t^{k-1} \right\|^2,$$

$$\left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 \leq 2 \left\| \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 + 2L^2 \sum_{t=1}^T \omega_t \left\| \bar{\mathbf{u}}^{k-1} - \mathbf{u}_t^{k-1} \right\|^2,$$

and,

$$\left\| \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 \leq 2 \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 + 2L^2 \sum_{t=1}^T \omega_t \left\| \bar{\mathbf{u}}^{k-1} - \mathbf{u}_t^{k-1} \right\|^2,$$

*Proof.* For  $k > 0$  and  $t \in [T]$ , we have

$$\begin{aligned} g_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) &= \\ &g_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) + f_t(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) - f_t(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) \end{aligned} \quad (\text{E.560})$$

$$= f_t(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) + r_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) \quad (\text{E.561})$$

$$= f_t(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) + r_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) - r_t^k(\mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1}) + r_t^k(\mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1}). \quad (\text{E.562})$$

Since  $g_t^k(\mathbf{u}_t^k, \mathbf{v}_t^{k-1}) = f_t(\mathbf{u}_t^k, \mathbf{v}_t^{k-1})$  (Definition 1), it follows that

$$g_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) = f_t(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) + r_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) - r_t^k(\mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1}). \quad (\text{E.563})$$

Because  $r_t^k$  is  $L$ -smooth in  $\mathbf{u}$  (Definition 1), we have

$$\begin{aligned} r_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) - r_t^k(\mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1}) &\leq \left\langle \nabla_{\mathbf{u}} r_t^k(\mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1}), \bar{\mathbf{u}}^{k-1} - \mathbf{u}_t^{k-1} \right\rangle \\ &\quad + \frac{L}{2} \left\| \bar{\mathbf{u}}^{k-1} - \mathbf{u}_t^{k-1} \right\|^2. \end{aligned} \quad (\text{E.564})$$

Since  $g_t^k$  is a partial first order surrogate of We have  $\nabla_{\mathbf{u}} r_t^k(\mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1}) = 0$ , thus

$$g_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) \leq f_t(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) + \frac{L}{2} \left\| \bar{\mathbf{u}}^{k-1} - \mathbf{u}_t^{k-1} \right\|^2. \quad (\text{E.565})$$

Multiplying by  $\omega_t$  and summing for  $t \in [T]$ , we have

$$g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \leq f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) + \frac{L}{2} \sum_{t=1}^T \omega_t \|\bar{\mathbf{u}}^{k-1} - \mathbf{u}_t^{k-1}\|^2, \quad (\text{E.566})$$

and the first inequality is proved.

Writing the gradient of Eq. (E.563), we have

$$\nabla_{\mathbf{u}} g_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) = \nabla_{\mathbf{u}} f_t(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) + \nabla_{\mathbf{u}} r_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) - \nabla_{\mathbf{u}} r_t^k(\mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1}). \quad (\text{E.567})$$

Multiplying by  $\omega_t$  and summing for  $t \in [T]$ , we have

$$\begin{aligned} \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) &= \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) + \\ &+ \sum_{t=1}^T \omega_t \left[ \nabla_{\mathbf{u}} r_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) - \nabla_{\mathbf{u}} r_t^k(\mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1}) \right]. \end{aligned} \quad (\text{E.568})$$

Thus,

$$\begin{aligned} \left\| \nabla_{\mathbf{u}} g^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 &= \\ \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) + \sum_{t=1}^T \omega_t \left[ \nabla_{\mathbf{u}} r_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) - \nabla_{\mathbf{u}} r_t^k(\mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1}) \right] \right\|^2 & \quad (\text{E.569}) \end{aligned}$$

$$\geq \frac{1}{2} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 - \left\| \sum_{t=1}^T \omega_t \left[ \nabla_{\mathbf{u}} r_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) - \nabla_{\mathbf{u}} r_t^k(\mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1}) \right] \right\|^2 \quad (\text{E.570})$$

$$\geq \frac{1}{2} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 - \sum_{t=1}^T \omega_t \left\| \nabla_{\mathbf{u}} r_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) - \nabla_{\mathbf{u}} r_t^k(\mathbf{u}_t^{k-1}, \mathbf{v}_t^{k-1}) \right\|^2 \quad (\text{E.571})$$

$$\geq \frac{1}{2} \left\| \nabla_{\mathbf{u}} f(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right\|^2 - L^2 \sum_{t=1}^T \omega_t \|\bar{\mathbf{u}}^{k-1} - \mathbf{u}_t^{k-1}\|^2, \quad (\text{E.572})$$

where (E.570) follows from  $\|a\|^2 = \|a + b - b\|^2 \leq 2\|a + b\|^2 + 2\|b\|^2$ . Thus,

$$\left\| \nabla_{\mathbf{u}} f_t(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) \right\|^2 \leq 2 \left\| \nabla_{\mathbf{u}} g_t^k(\bar{\mathbf{u}}^{k-1}, \mathbf{v}_t^{k-1}) \right\|^2 + 2L^2 \sum_{t=1}^T \omega_t \|\bar{\mathbf{u}}^{k-1} - \mathbf{u}_t^{k-1}\|^2. \quad (\text{E.573})$$

The proof of the last inequality is similar, it leverages  $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$  to upper bound (E.569).  $\square$

**Lemma E.18.** Consider  $\mathbf{u}_1, \dots, \mathbf{u}_M \in \mathbb{R}^d$  and  $\alpha = (\alpha_1, \dots, \alpha_M) \in \Delta^M$ . Define the block matrix  $\mathbf{H}$  with

$$\begin{cases} \mathbf{H}_{m,m} = -\alpha_m \cdot (1 - \alpha_m) \cdot \mathbf{u}_m \cdot \mathbf{u}_m^\top \\ \mathbf{H}_{m,m'} = \alpha_m \cdot \alpha_{m'} \cdot \mathbf{u}_m \cdot \mathbf{u}_{m'}^\top; & m' \neq m, \end{cases} \quad (\text{E.574})$$

then  $\mathbf{H}$  is a semi-definite negative matrix.

*Proof.* Consider  $\mathbf{x} = [\mathbf{x}_1, \dots, \mathbf{x}_M] \in \mathbb{R}^{dM}$ , we want to prove that

$$\mathbf{x}^\top \cdot \mathbf{H} \cdot \mathbf{x} \leq 0. \quad (\text{E.575})$$

We have:

$$\mathbf{X}^\top \cdot \mathbf{H} \cdot \mathbf{X} = \sum_{m=1}^M \sum_{m'=1}^M \mathbf{x}_m^\top \cdot \mathbf{H}_{m,m'} \cdot \mathbf{x}_{m'} \quad (\text{E.576})$$

$$= \sum_{m=1}^M \left[ \mathbf{x}_m^\top \cdot \mathbf{H}_{m,m} \cdot \mathbf{x}_m + \sum_{\substack{m'=1 \\ m' \neq m}}^M \mathbf{x}_m^\top \cdot \mathbf{H}_{m,m'} \cdot \mathbf{x}_{m'} \right] \quad (\text{E.577})$$

$$= \sum_{m=1}^M (-\alpha_m \cdot (1 - \alpha_m) \cdot \mathbf{x}_m^\top \cdot \mathbf{u}_m \cdot \mathbf{u}_m^\top \cdot \mathbf{x}_m) \quad (\text{E.578})$$

$$+ \sum_{m=1}^M \left[ \sum_{\substack{m'=1 \\ m' \neq m}}^M (\alpha_m \cdot \alpha_{m'} \cdot \mathbf{x}_m^\top \cdot \mathbf{u}_m \cdot \mathbf{u}_{m'}^\top \cdot \mathbf{x}_{m'}) \right] \quad (\text{E.579})$$

$$= \sum_{m=1}^M \left[ -\alpha_m \cdot (1 - \alpha_m) \cdot \langle \mathbf{x}_m, \mathbf{u}_m \rangle^2 + \alpha_m \cdot \langle \mathbf{x}_m, \mathbf{u}_m \rangle \sum_{\substack{m'=1 \\ m' \neq m}}^M \alpha_{m'} \cdot \langle \mathbf{x}_{m'}, \mathbf{u}_{m'} \rangle \right]. \quad (\text{E.580})$$

Since  $\alpha \in \Delta^M$ ,

$$\forall m \in [M], \sum_{\substack{m'=1 \\ m' \neq m}}^M \alpha_{m'} = (1 - \alpha_m), \quad (\text{E.581})$$

thus,

$$\mathbf{x}^\top \cdot \mathbf{H} \cdot \mathbf{x} = \sum_{m=1}^M \alpha_m \cdot \langle \mathbf{x}_m, \mathbf{u}_m \rangle \cdot \sum_{\substack{m'=1 \\ m' \neq m}}^M \alpha_{m'} \left( \langle \mathbf{x}_{m'}, \mathbf{u}_{m'} \rangle - \langle \mathbf{x}_m, \mathbf{u}_m \rangle \right) \quad (\text{E.582})$$

$$= \sum_{m=1}^M \alpha_m \cdot \langle \mathbf{x}_m, \mathbf{u}_m \rangle \cdot \sum_{m'=1}^M \alpha_{m'} \left( \langle \mathbf{x}_{m'}, \mathbf{u}_{m'} \rangle - \langle \mathbf{x}_m, \mathbf{u}_m \rangle \right) \quad (\text{E.583})$$

$$= \left( \sum_{m=1}^M \alpha_m \cdot \langle \mathbf{x}_m, \mathbf{u}_m \rangle \right)^2 - \sum_{m=1}^M \alpha_m \cdot \langle \mathbf{x}_m, \mathbf{u}_m \rangle^2. \quad (\text{E.584})$$

Using Jensen inequality, we have  $\mathbf{x}^\top \cdot \mathbf{H} \cdot \mathbf{x} \leq 0$ .  $\square$

**Lemma E.19.** Under Assumptions 12', 16' and 17', the iterates of Alg. 7 verify for  $k > 0$  and  $t \in [T]$ ,

$$\forall \mathbf{v} \in \mathcal{V}, \left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}_t^k, \mathbf{v}) \right\| \leq \sqrt{\alpha \kappa} \cdot \left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}^{k-1}, \mathbf{v}) \right\|, \quad (\text{E.585})$$

where  $\kappa = L/\mu$ .



*Proof.* Consider  $\mathbf{v} \in \mathcal{V}$ . Since  $g_t^k$  is  $L$ -smooth in  $\mathbf{u}$  (Assumption 12'), we have using Assumption 16',

$$\left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}_t^k, \mathbf{v}) \right\|_F^2 \leq 2L \left( g_t^k(\mathbf{u}_t^k, \mathbf{v}) - g_t^k(\mathbf{u}_{t,*}^k, \mathbf{v}) \right) \leq 2L\alpha \left( g_t^k(\mathbf{u}^{k-1}, \mathbf{v}) - g_t^k(\mathbf{u}_{t,*}^k, \mathbf{v}) \right). \quad (\text{E.586})$$

Since  $\Phi_t^k$  is  $\mu$ -strongly convex (Assumption 17'), we can use Polyak-Lojasiewicz (PL) inequality,

$$g_t^k(\mathbf{u}^{k-1}, \mathbf{v}) - \frac{1}{2\mu} \left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}^{k-1}, \mathbf{v}) \right\|^2 \leq g_t^k(\mathbf{u}_{t,*}^k, \mathbf{v}), \quad (\text{E.587})$$

thus,

$$2\mu \left( g_t^k(\mathbf{u}^{k-1}, \mathbf{v}) - g_t^k(\mathbf{u}_{t,*}^k, \mathbf{v}) \right) \leq \left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}^{k-1}, \mathbf{v}) \right\|^2. \quad (\text{E.588})$$

Combining Eq. (E.586) and Eq. (E.588), we have

$$\left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}^{k-1}, \mathbf{v}) \right\|^2 \leq \frac{L}{\mu} \alpha \left\| \nabla_{\mathbf{u}} g_t^{k-1}(\mathbf{u}^{k-1}, \mathbf{v}) \right\|^2, \quad (\text{E.589})$$

thus,

$$\left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}_t^k, \mathbf{v}) \right\| \leq \sqrt{\alpha\kappa} \left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}^{k-1}, \mathbf{v}) \right\|. \quad (\text{E.590})$$

□

**Lemma E.20.** *Suppose that Assumptions 12', 14', 16' and 17' hold with  $G^2 = 0$ . Then,*

$$g^k(\mathbf{u}^k, \mathbf{v}^k) - g^k(\mathbf{u}_*^k, \mathbf{v}^k) \leq \tilde{\alpha} \times \left\{ g^k(\mathbf{u}^{k-1}, \mathbf{v}^{k-1}) - g^k(\mathbf{u}_*^k, \mathbf{v}^k) \right\}, \quad (\text{E.591})$$

where  $\tilde{\alpha} = \beta^2 \kappa^4 \alpha$ , and  $\mathbf{u}_*^k \triangleq \arg \min_{\mathbf{u}} g^k(\mathbf{u}, \mathbf{v}_{1:T}^k)$  where  $g^k$  is defined in (E.305)

*Proof.* Consider  $k > 0$  and  $t \in [T]$ . Since  $g_t$  is  $\mu$ -convex in  $\mathbf{u}$  (Assumption 17'), we write

$$\left\| \mathbf{u}_t^k - \mathbf{u}_*^k \right\|_F \leq \frac{1}{\mu} \left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}_t^k, \mathbf{v}_t^k) - \nabla_{\mathbf{u}} g_t^k(\mathbf{u}_*^k, \mathbf{v}_t^k) \right\| \quad (\text{E.592})$$

$$\leq \frac{1}{\mu} \left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}_t^k, \mathbf{v}_t^k) \right\| + \frac{1}{\mu} \left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}_*^k, \mathbf{v}_t^k) \right\| \quad (\text{E.593})$$

$$\leq \frac{\sqrt{\alpha\kappa}}{\mu} \left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}^{k-1}, \mathbf{v}_t^k) \right\| + \frac{1}{\mu} \left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}_*^k, \mathbf{v}_t^k) \right\|, \quad (\text{E.594})$$

where the last inequality is a result of Lemma E.19. Using Jensen inequality, we have

$$\left\| \mathbf{u}^k - \mathbf{u}_*^k \right\|_F = \left\| \sum_{t=1}^T \omega_t \cdot (\mathbf{u}_t^k - \mathbf{u}_*^k) \right\| \quad (\text{E.595})$$

$$\leq \sum_{t=1}^T \omega_t \cdot \left\| \mathbf{u}_t^k - \mathbf{u}_*^k \right\| \quad (\text{E.596})$$

$$\leq \sum_{t=1}^T \omega_t \cdot \left\{ \frac{\sqrt{\alpha\kappa}}{\mu} \left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}^{k-1}, \mathbf{v}_t^k) \right\| + \frac{1}{\mu} \left\| \nabla_{\mathbf{u}} g_t^k(\mathbf{u}_*^k, \mathbf{v}_t^k) \right\| \right\}. \quad (\text{E.597})$$

Using Assumption 14' and Jensen inequality with the " $\sqrt{\cdot}$ " function, it follows that

$$\|\mathbf{u}^k - \mathbf{u}_*\| \leq \sqrt{\alpha\kappa} \frac{\beta}{\mu} \|\nabla_{\mathbf{u}} g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k)\| + \frac{\beta}{\mu} \|\nabla_{\mathbf{u}} g^k(\mathbf{u}_*, \mathbf{v}_{1:T}^k)\| \quad (\text{E.598})$$

$$= \sqrt{\alpha\kappa} \frac{\beta}{\mu} \|\nabla_{\mathbf{u}} g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^k)\|. \quad (\text{E.599})$$

Since  $g^k$  is  $L$ -smooth in  $\mathbf{u}$  as a convex combination of  $L$ -smooth function, we have

$$\|\nabla_{\mathbf{u}} g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k)\| = \|\nabla_{\mathbf{u}} g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^k) - \nabla_{\mathbf{u}} g^k(\mathbf{u}_*, \mathbf{v}_{1:T}^k)\| \quad (\text{E.600})$$

$$\leq L \|\mathbf{u}^k - \mathbf{u}_*\| \quad (\text{E.601})$$

$$\leq \beta\sqrt{\alpha\kappa^3} \|\nabla_{\mathbf{u}} g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^k)\|. \quad (\text{E.602})$$

Using Polyak-Lojasiewicz (PL), we have

$$g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - g^k(\mathbf{u}_*, \mathbf{v}_{1:T}^k) \leq \frac{1}{2\mu} \|\nabla_{\mathbf{u}} g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k)\|^2 \leq \frac{\beta^2 \alpha \kappa^3}{2\mu} \|\nabla_{\mathbf{u}} g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^k)\|^2. \quad (\text{E.603})$$

Using the  $L$ -smoothness of  $g^k$  in  $\mathbf{u}$ , we have

$$\|\nabla_{\mathbf{u}} g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^k)\|^2 \leq 2L [g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^k) - g^k(\mathbf{u}_*, \mathbf{v}_{1:T}^k)]. \quad (\text{E.604})$$

Thus,

$$g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - g^k(\mathbf{u}_*, \mathbf{v}_{1:T}^k) \leq \underbrace{\beta^2 \kappa^4 \alpha}_{\triangleq \tilde{\alpha}} [g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^k) - g^k(\mathbf{u}_*, \mathbf{v}_{1:T}^k)]. \quad (\text{E.605})$$

Since  $\mathbf{v}_t^k = \arg \min_{\mathbf{v} \in \mathcal{V}} g_t^k(\mathbf{u}^{k-1}, \mathbf{v})$ , it follows that

$$g_t^k(\mathbf{u}^{k-1}, \mathbf{v}_t^k) \leq g_t^k(\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}). \quad (\text{E.606})$$

Thus,

$$g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - g^k(\mathbf{u}_*, \mathbf{v}_{1:T}^k) \leq \tilde{\alpha} \times \left\{ g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) - g^k(\mathbf{u}_*, \mathbf{v}_{1:T}^k) \right\}. \quad (\text{E.607})$$

□

For  $t \in [T]$  and  $k > 0$ , we introduce  $r_t^k \triangleq g_t^k - f_t$  and  $r^k \triangleq g^k - f = \sum_{t=1}^T \omega_t (g_t^k - f_t)$ . Since  $g_t^k$  is a partial first-order surrogate of  $f_t$ , it follows that  $r_t^k(\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}) = 0$  and that  $r_t^k$  is non-negative and  $L$ -smooth in  $\mathbf{u}$ .

**Lemma E.21.** *Suppose that Assumptions 11' and 12' hold and that*

$$g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \leq g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}), \quad \forall k > 0, \quad (\text{E.608})$$

then

$$\lim_{k \rightarrow \infty} r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) = 0 \quad (\text{E.609})$$

$$\lim_{k \rightarrow \infty} \left\| \nabla_{\mathbf{u}} r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\|^2 = 0 \quad (\text{E.610})$$

If we moreover suppose that Assumption 17' holds and that there exists  $0 < \tilde{\alpha} < 1$  such that for all  $k > 0$ ,

$$g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - g^k(\mathbf{u}_*, \mathbf{v}_{1:T}^k) \leq \tilde{\alpha} \times \left( g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) - g^k(\mathbf{u}_*, \mathbf{v}_{1:T}^k) \right), \quad (\text{E.611})$$

then,

$$\lim_{k \rightarrow \infty} \left\| \mathbf{u}^k - \mathbf{u}_* \right\|^2 = 0 \quad (\text{E.612})$$

where  $\mathbf{u}_*$  is the minimizer of  $\mathbf{u} \mapsto g^k(\mathbf{u}, \mathbf{v}_{1:T}^k)$ .

*Proof.* Since  $g_t$  is a partial first-order surrogate of  $f$  near  $\{\mathbf{u}^{k-1}, \mathbf{v}_t^{k-1}\}$  for  $t \in [T]$  and  $k > 0$ , it follows that  $g^k$  is a majorant of  $f$  and that  $g^k(\mathbf{u}^{k-1}, \mathbf{v}^{k-1}) = f(\mathbf{u}^{k-1}, \mathbf{v}^{k-1})$ . Thus, the following holds,

$$f(\mathbf{u}^k, \mathbf{v}^k) \leq g^k(\mathbf{u}^k, \mathbf{v}^k) \leq g^k(\mathbf{u}^{k-1}, \mathbf{v}^{k-1}) = f(\mathbf{u}^{k-1}, \mathbf{v}^{k-1}), \quad (\text{E.613})$$

It follows that the sequence  $\left( f(\mathbf{u}^k, \mathbf{v}^k) \right)_{k \geq 0}$  is a non-increasing sequence. Since  $f$  is bounded below (Assum. 11'), it follows that  $\left( f(\mathbf{u}^k, \mathbf{v}^k) \right)_{k \geq 0}$  is convergent. Denote by  $f^\infty$  its limit. The sequence  $\left( g^k(\mathbf{u}^k, \mathbf{v}^k) \right)_{k \geq 0}$  also converges to  $f^\infty$ .

**Proof of Eq. E.609** Using the fact that  $g^k(\mathbf{u}^k, \mathbf{v}^k) \leq g^k(\mathbf{u}^{k-1}, \mathbf{v}^k)$ , we write for  $k > 0$ ,

$$f(\mathbf{u}^k, \mathbf{v}_{1:T}^k) + r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) = g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \leq g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) = f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}), \quad (\text{E.614})$$

Thus,

$$r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \leq f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) - f(\mathbf{u}^k, \mathbf{v}^k), \quad (\text{E.615})$$

By summing over  $k$  then passing to the limit when  $k \rightarrow +\infty$ , we have

$$\sum_{k=1}^{\infty} r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \leq f(\mathbf{u}^0, \mathbf{v}_{1:T}^0) - f^\infty, \quad (\text{E.616})$$

Finally since  $r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k)$  is non negative for  $k > 0$ , the sequence  $\left( r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right)_{k \geq 0}$  necessarily converges to zero, i.e.,

$$\lim_{k \rightarrow \infty} r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) = 0. \quad (\text{E.617})$$

**Proof of Eq. E.610** Because the  $L$ -smoothness of  $\mathbf{u} \mapsto r^k(\mathbf{u}, \mathbf{v}_{1:T}^k)$ , we have

$$r^k \left( \mathbf{u}^k - \frac{1}{L} \nabla_{\mathbf{u}} r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k), \mathbf{v}_{1:T}^k \right) \leq r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - \frac{1}{2L} \left\| \nabla_{\mathbf{u}} r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\|^2 \quad (\text{E.618})$$

Thus,

$$\left\| \nabla_{\mathbf{u}} r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\|_F^2 \leq 2L \left( r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - r^k \left( \mathbf{u}^k - \frac{1}{L} \nabla_{\mathbf{u}} r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k), \mathbf{v}_{1:T}^k \right) \right) \quad (\text{E.619})$$

$$\leq 2Lr^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k), \quad (\text{E.620})$$

because  $r^k$  is a non-negative function (Definition 1). Finally, using Eq. (E.609), it follows that

$$\lim_{k \rightarrow \infty} \left\| \nabla_{\mathbf{u}} r^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \right\|^2 = 0. \quad (\text{E.621})$$

**Proof of Eq. E.612** We suppose now that there exists  $0 < \tilde{\alpha} < 1$  such that

$$\forall k > 0, \quad g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - g^k(\mathbf{u}_*^k, \mathbf{v}_{1:T}^k) \leq \tilde{\alpha} \left( g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) - g^k(\mathbf{u}_*^k, \mathbf{v}_{1:T}^k) \right), \quad (\text{E.622})$$

It follows that,

$$g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - \tilde{\alpha} g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \leq (1 - \tilde{\alpha}) g^k(\mathbf{u}_*^k, \mathbf{v}_{1:T}^k), \quad (\text{E.623})$$

then,

$$g^k(\mathbf{u}_*^k, \mathbf{v}_{1:T}^k) \geq \frac{1}{1 - \tilde{\alpha}} \times \left[ g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - \tilde{\alpha} \times g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right], \quad (\text{E.624})$$

and by using the definition of  $g^k$  we have,

$$g^k(\mathbf{u}_*^k, \mathbf{v}_{1:T}^k) \geq \frac{1}{1 - \tilde{\alpha}} \times \left[ g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - \tilde{\alpha} \times f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right], \quad (\text{E.625})$$

Since  $g^k(\mathbf{u}_*^k, \mathbf{v}_{1:T}^k) \leq g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \leq g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1})$ , we have

$$g^k(\mathbf{u}_*^k, \mathbf{v}_{1:T}^k) \leq g^k(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) = f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}). \quad (\text{E.626})$$

From Eq. (E.625) and Eq. (E.626), it follows that,

$$\frac{1}{1 - \tilde{\alpha}} \times \left[ g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - \tilde{\alpha} \times f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \right] \leq g^k(\mathbf{u}_*^k, \mathbf{v}_{1:T}^k) \leq f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}), \quad (\text{E.627})$$

Finally, since  $f(\mathbf{u}^{k-1}, \mathbf{v}_{1:T}^{k-1}) \xrightarrow[k \rightarrow +\infty]{} f^\infty$  and  $g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) \xrightarrow[k \rightarrow +\infty]{} f^\infty$ , it follows from Eq. (E.627) that,

$$\lim_{k \rightarrow \infty} g^k(\mathbf{u}_*^k, \mathbf{v}_{1:T}^k) = f^\infty. \quad (\text{E.628})$$

Since  $g^k$  is  $\mu$ -strongly convex in  $\mathbf{u}$  (Assumption 17), we write

$$\frac{\mu}{2} \left\| \mathbf{u}^k - \mathbf{u}_*^k \right\|^2 \leq g^k(\mathbf{u}^k, \mathbf{v}_{1:T}^k) - g^k(\mathbf{u}_*^k, \mathbf{v}_{1:T}^k), \quad (\text{E.629})$$

It follows that,

$$\lim_{k \rightarrow +\infty} \left\| \mathbf{u}^k - \mathbf{u}_*^k \right\|^2 = 0. \quad (\text{E.630})$$

□

## E.8 Additional Experiments

### E.9 Fully Decentralized Federated Expectation-Maximization

D-FedEM considers the scenario where clients communicate directly in a peer-to-peer fashion instead of relying on the central server mediation. In order to simulate D-FedEM, we consider a binomial Erdős-Rényi graph [ER59] with parameter  $p = 0.5$ , and we set the mixing weight using *Fast Mixing Markov Chain* [BDX03] rule. We report the result of this experiment in Table 5, showing the average weighted accuracy with weight proportional to local dataset sizes. We observe that D-FedEM often performs better than other FL approaches and slightly worst than FedEM, except on CIFAR-10 where it has low performances.

Table 5: Test accuracy: average across clients.

Dataset	Local	FedAvg	FedAvg+	Clustered	FL	pFedMe	FedEM (Ours)	D-FedEM (Ours)
FEMNIST	71.0	78.6	75.3		73.5	74.9	<b>79.9</b>	77.2
EMNIST	71.9	82.6	83.1		82.7	83.3	<b>83.5</b>	<b>83.5</b>
CIFAR10	70.2	78.2	82.3		78.6	81.7	<b>84.3</b>	77.0
CIFAR100	31.5	40.9	39.0		41.5	41.8	<b>44.1</b>	43.9
Shakespeare	32.0	<b>46.7</b>	40.0		46.6	41.2	<b>46.7</b>	45.4
Synthetic	65.7	68.2	68.9		69.1	69.2	<b>74.7</b>	73.8

## E.10 Comparison with MOCHA

In the case of synthetic dataset, for which train a linear model, we compare FedEM with MOCHA [Smi+17]. We implemented MOCHA in Python following the official implementation\* in MATLAB. We tuned the parameter  $\lambda$  of MOCHA on a holdout validation set via grid search in  $\{10^1, 10^0, 10^{-1}, 10^{-2}, 10^{-3}\}$ , and we found that the optimal value of  $\lambda$  is  $10^0$ . For this value, we ran MOCHA on the synthetic dataset with three different seeds, and we found that the average accuracy is  $73.4 \pm 0.05$  in comparison to  $74.7 \pm 0.01$  achieved by FedEM. Note that MOCHA is the second best method after FedEM on this dataset. Unfortunately, MOCHA only works for linear models.

## E.11 Generalization to Unseen Clients

Table 3.3 shows that FedEM allows new clients to learn a personalized model at least as good as FedAvg’s global one and always better than FedAvg+’s one. Unexpectedly, new clients achieve sometimes a significantly higher test accuracy than old clients (e.g., 47.5% against 44.1% on CIFAR100).

In order to better understand this difference, we looked at the distribution of FedEM personalized weights for the old clients and new ones. The average distribution entropy equals 0.27 and 0.92 for old and new clients, respectively. This difference shows that old clients tend to have more skewed distributions, suggesting that some components may be overfitting the local training dataset leading the old clients to give them a high weight.

We also considered a setting where unseen clients progressively collect their own dataset. We investigate the effect of the number of samples on the average test accuracy across unseen clients, starting from no local data (and therefore using uniform weights to mix the  $M$  components) and progressively adding more labeled examples until the full local labeled training set is assumed to be available. Figure E.15 shows that FedEM achieves a significant level of personalization as soon as clients collect a labeled dataset whose size is about 20% of what the original clients used for training.

As we mentioned in the main text, it is not clear how the other personalized FL algorithms (e.g., pFedMe and Clustered FL) should be extended to handle unseen clients. For example, the global model learned by pFedMe during training can then be used to perform some “fine-tuning” at the new clients, but how exactly? The original pFedMe paper [TTN20] does not even mention this issue. For example, the client could use the global model as initial vector for some local SGD steps (similarly to what done in FedAvg+ or the MAML approaches) or it could perform a local pFedMe update (lines 6-9 in [TTN20, Alg. 1]). The problem is even more complex for

\*<https://github.com/gingsmith/fmtl>

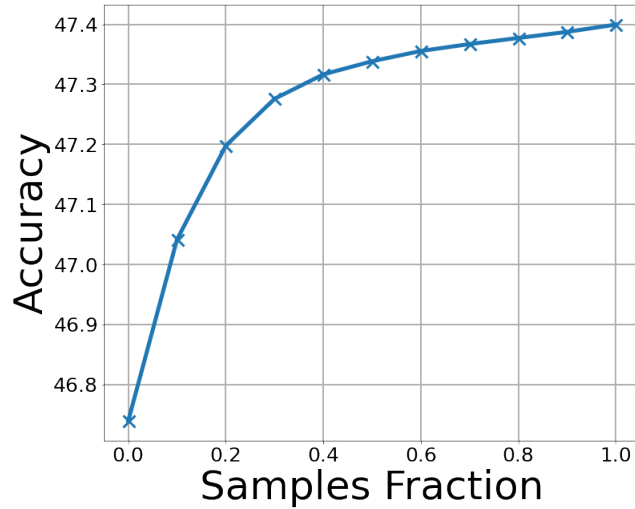


Figure E.15: Effect of the number of samples on the average test accuracy across clients unseen at training on CIFAR100 dataset.

Clustered FL (and again not discussed in [SMS20]). The new client should be assigned to one of the clusters identified. One can think to compute the cosine distances of the new client from those who participated in training, but this would require the server to maintain not only the model learned, but also the last-iteration gradients of all clients that participated in the training. Moreover, it is not clear which metric should be considered to assign the new client to a given cluster (perhaps the average cosine similarity from all clients in the cluster?). This is an arbitrary choice as [SMS20] does not provide a criterion to assign clients to a cluster, but only to decide if a given cluster should be split in two new ones. It appears that many options are possible and they deserve separate investigation. Despite these considerations, we performed an additional experiment extending  $p\text{FedMe}$  to unseen clients as described in the second option above on CIFAR-100 dataset with a sampling rate of 20%.  $p\text{FedMe}$  achieves a test accuracy of  $40.5\% \pm 1.66\%$ , in comparison to  $38.9\% \pm 0.97\%$  for FedAvg and  $42.7\% \pm 0.33\%$  for FedEM. FedEM thus performs better on unseen clients, and  $p\text{FedMe}$ 's accuracy shows a much larger variability.

## E.12 FedEM and Clustering

We performed additional experiments with synthetic datasets to check if FedEM recovers clusters in practice. We modified the synthetic dataset generation so that the mixture weight vector  $\pi_t$  of each client  $t$  has a single entry equal to 1 that is selected uniformly at random. We consider two scenarios both with  $T = 300$  client, the first with  $M = 2$  component and the second with  $M = 3$  components. In both cases FedEM recovered almost the correct  $\Pi^*$  and  $\Theta^*$ : we have  $\text{cosine\_distance}(\Theta^*, \check{\Theta}) \leq 10^{-2}$  and  $\text{cosine\_distance}(\Pi^*, \check{\Pi}) \leq 10^{-8}$ . A simple clustering algorithm that assigns each client to the component with the largest mixture weight achieves 100% accuracy, i.e., it partitions the clients in sets coinciding with the original clusters.

Table 6: Test and train accuracy comparison across different tasks. For each method, the best test accuracy is reported. For FedEM we run only  $\frac{K}{M}$  rounds, where  $K$  is the total number of rounds for other methods— $K = 80$  for Shakespeare and  $K = 200$  for all other datasets—and  $M = 3$  is the number of components used in FedEM.

Dataset	Local	FedAvg	FedProx	FedAvg+	Clustered FL	pFedMe	FedEM (Ours)
FEMNIST	71.0 (99.2)	<b>78.6</b> (79.5)	78.6 (79.6)	75.3 (86.0)	73.5 (74.3)	74.9 (91.9)	74.0 (80.9)
EMNIST	71.9 (99.9)	82.6 (86.5)	82.7 (86.6)	83.1 (93.5)	82.7 (86.6)	<b>83.3</b> (91.1)	82.7 (89.4)
CIFAR10	70.2 (99.9)	78.2 (96.8)	78.0 (96.7)	82.3 (98.9)	78.6 (96.8)	81.7 (99.8)	<b>82.5</b> (92.2)
CIFAR100	31.5 (99.9)	41.0 (78.5)	40.9 (78.6)	39.0 (76.7)	41.5 (78.9)	41.8 (99.6)	<b>42.0</b> (72.9)
Shakespeare	32.0 (95.3)	<b>46.7</b> (48.7)	45.7 (47.3)	40.0 (93.1)	46.6 (48.7)	41.2 (42.1)	43.8 (44.6)
Synthetic	65.7 (91.0)	68.2 (68.7)	68.2 (68.7)	68.9 (71.0)	69.1 (85.1)	69.2 (72.8)	<b>73.2</b> (74.7)

### E.13 Effect of $M$ in Time-Constrained Setting

Recall that in FedEM, each client needs to update and transmit  $M$  components at each round, requiring roughly  $M$  times more computation and  $M$  times larger messages than the competitors in our study. In this experiment, we considered a challenging time-constrained setting, where FedEM is limited to run one third ( $= 1/M$ ) of the rounds of the other methods. The results in Table 6 show that even if FedEM does not reach its maximum accuracy, it still outperforms the other methods on 3 datasets.

We additionally compared FedEM with a model having the same number of parameters in order to check if FedEM’s advantage comes from the additional model parameters rather than by its specific formulation. To this purpose, we trained Resnet-18 and Resnet-34 on CIFAR10. The first one has about 3 times more parameters than MobileNet-v2 and then roughly as many parameters as FedEM with  $M = 3$ . The second one has about 6 times more parameters than FedEM with  $M = 3$ . We observed that both architectures perform even worse than MobileNet-v2, so the comparison with these larger models does not suggest that FedEM’s advantage comes from the larger number of parameters.

We note that there are many possible choices of (more complex) model architectures, and finding one that works well for the task at hand is quite challenging due to the large search space, the bias-variance trade-off, and the specificities of the FL setting.

Table 7: Test accuracy under 20% client sampling: average across clients with +/- standard deviation over 3 independent runs. All experiments with 1200 communication rounds.

Dataset	FedAvg	FedAvg+	pFedMe	APFL	FedEM (Ours)
CIFAR-10	73.1 $\pm$ 0.14	77.7 $\pm$ 0.16	77.8 $\pm$ 0.07	78.2 $\pm$ 0.27	<b>82.1 <math>\pm</math> 0.13</b>
CIFAR-100	40.6 $\pm$ 0.17	39.7 $\pm$ 0.75	39.9 $\pm$ 0.08	40.3 $\pm$ 0.71	<b>43.2 <math>\pm</math> 0.23</b>
Synthetic	68.2 $\pm$ 0.02	69.0 $\pm$ 0.03	69.1 $\pm$ 0.03	69.1 $\pm$ 0.04	<b>74.7 <math>\pm</math> 0.01</b>

### E.14 Additional Results under Client Sampling

In our experiments, except for Figure 3.1, we considered that all clients participate at each round. We run extra experiments with client sampling, by allowing only 20% of the clients to participate at each round. We also incorporate APFL [DKM20] into the comparison. Table 7 summarizes our findings, giving the average and standard deviation of the test accuracy across 3 independent runs.

### E.15 Convergence Plots

Figures E.16 to E.21 show the evolution of average train loss, train accuracy, test loss, and test accuracy over time for each experiment shown in Table 3.2.



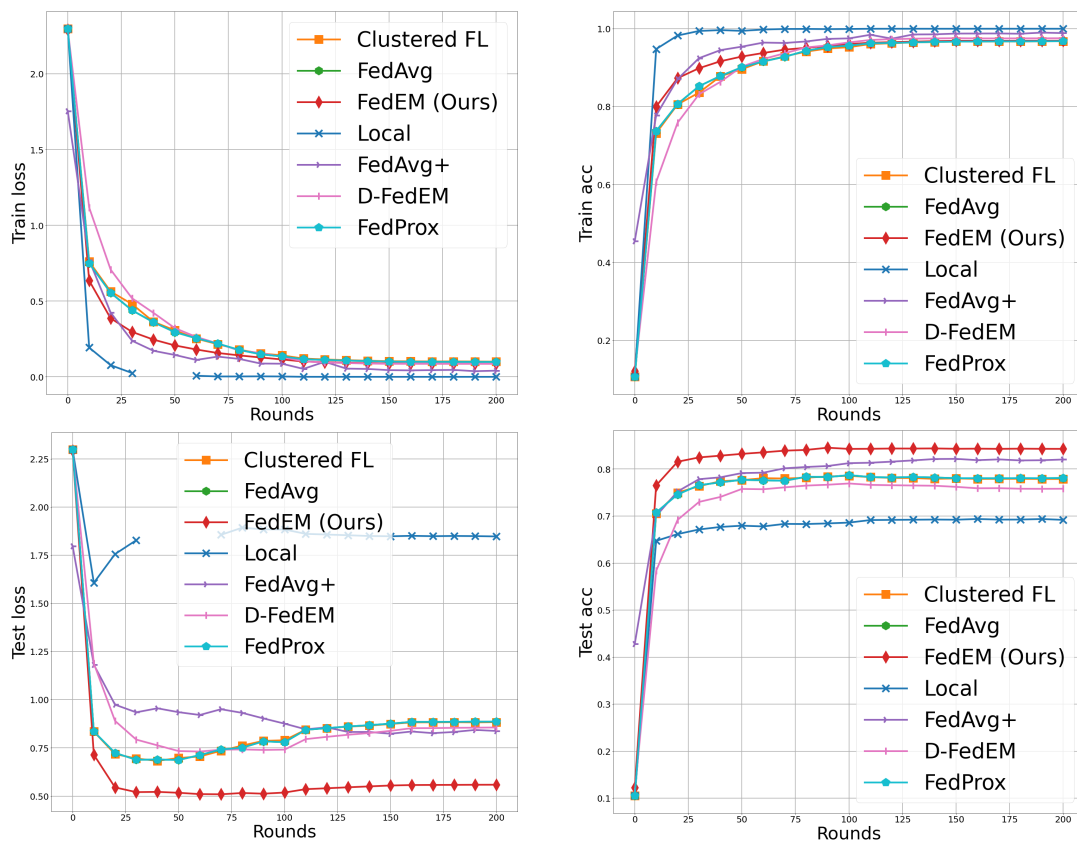


Figure E.16: Train loss, train accuracy, test loss, and test accuracy for CIFAR10 [Kri09]. .

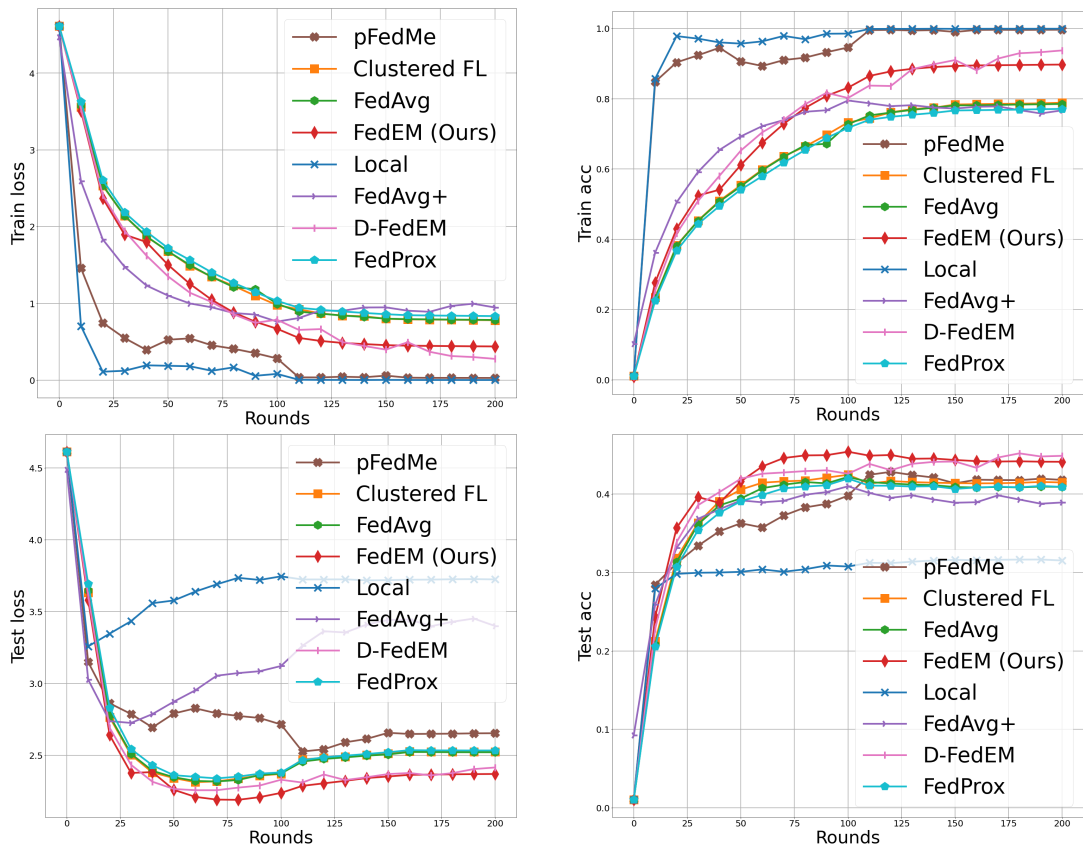


Figure E.17: Train loss, train accuracy, test loss, and test accuracy for CIFAR100 [Kri09].

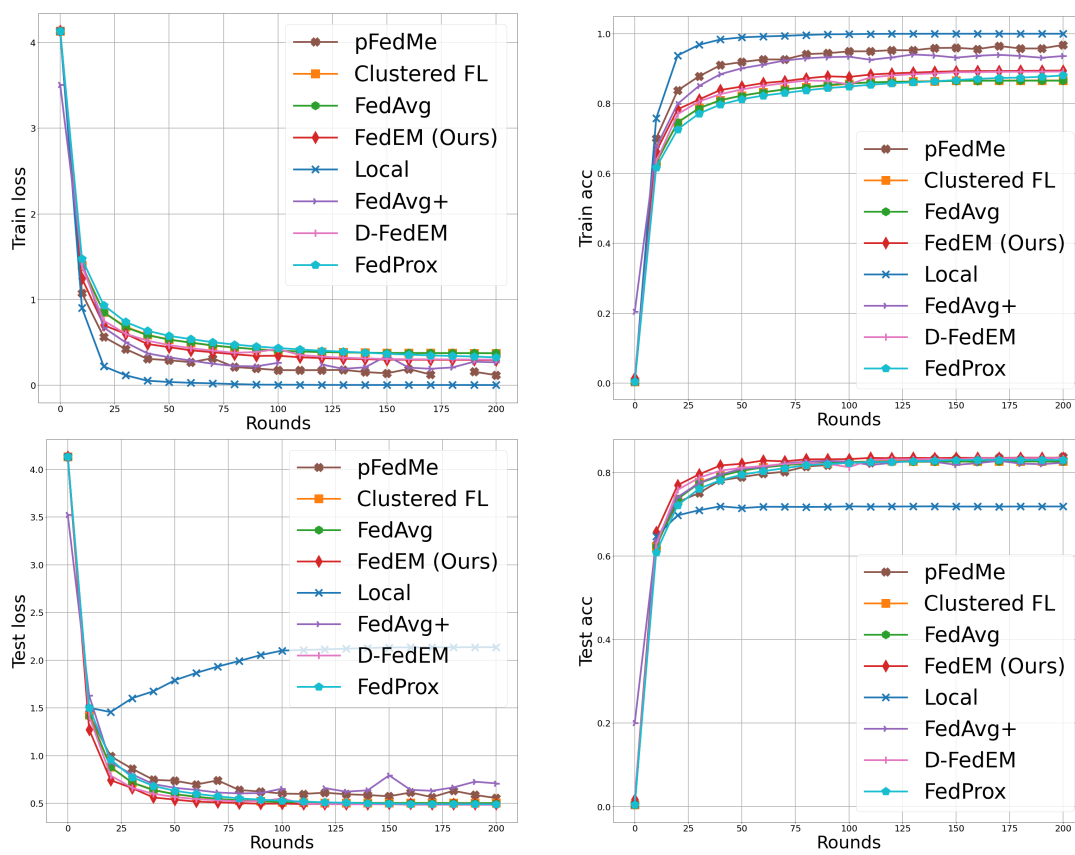


Figure E.18: Train loss, train accuracy, test loss, and test accuracy for EMNIST [Coh+17].

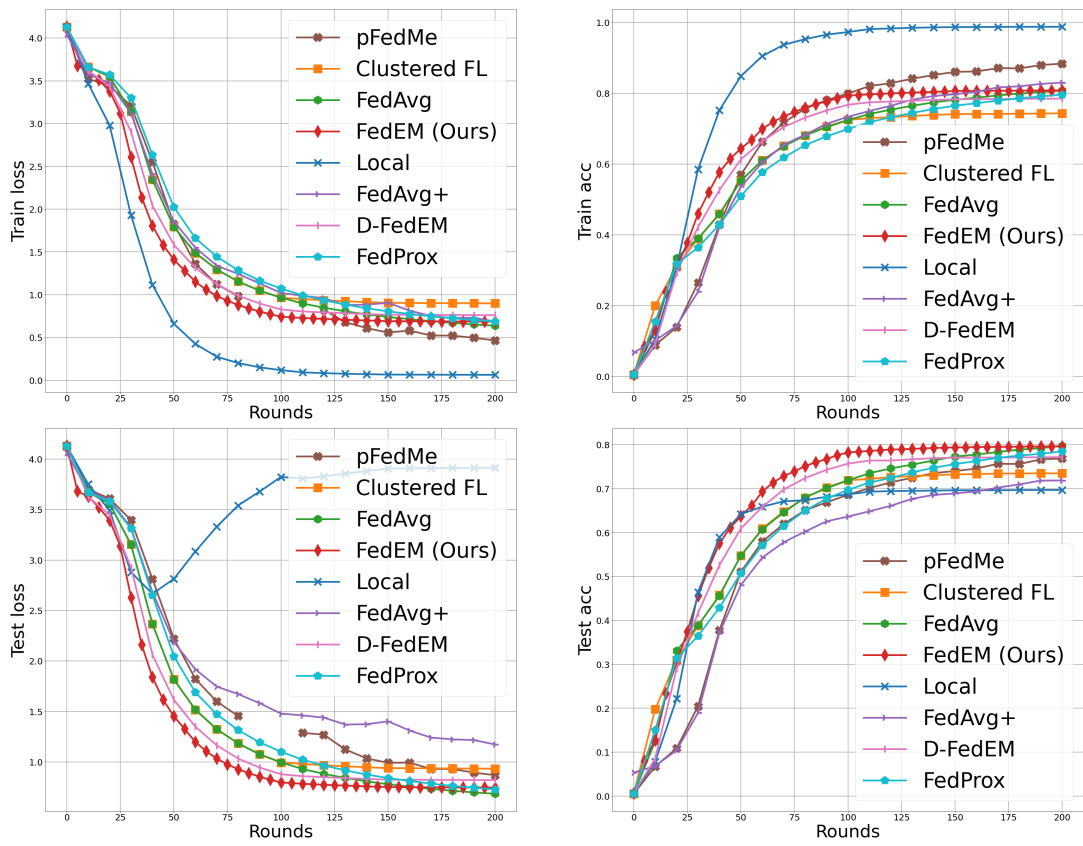


Figure E.19: Train loss, train accuracy, test loss, and test accuracy for FEMNIST [Cal+19; McM+17].

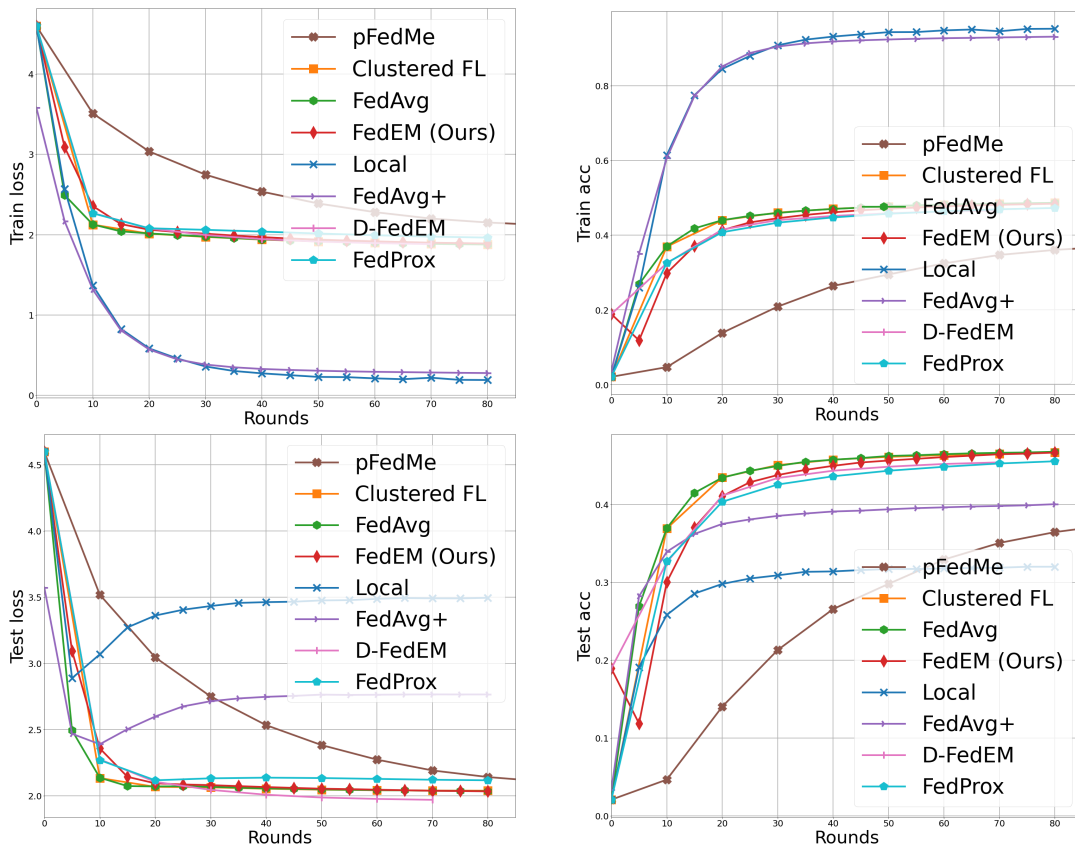


Figure E.20: Train loss, train accuracy, test loss, and test accuracy for Shakespeare [Cal+19; McM+17].

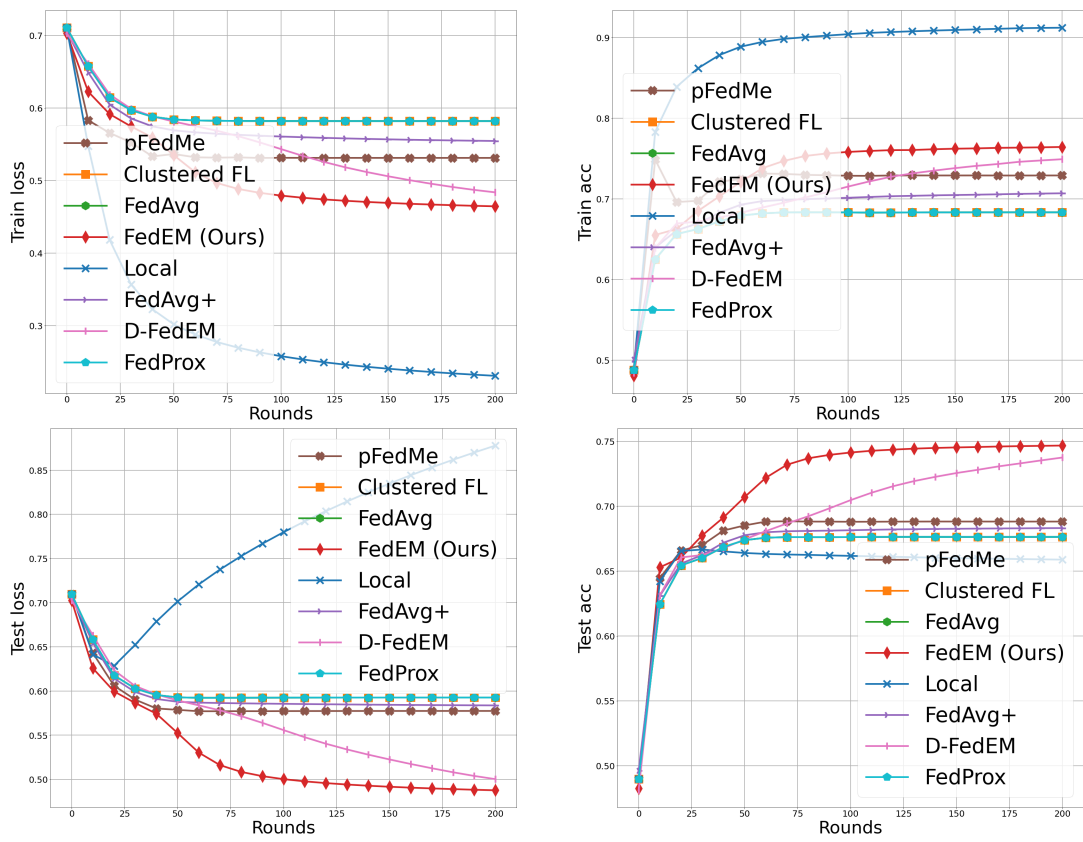


Figure E.21: Train loss, train accuracy, test loss, and test accuracy for synthetic dataset.

## F Personalized Federated Learning through Local Memorization

In the general description of  $k\text{NN-Per}$ , and in our experiments, we considered that each client  $t \in [T]$  uses its whole dataset  $\mathcal{S}_t$  both to train the base shared model  $h_{\mathcal{S}}$ —and the corresponding representation function  $\phi_{h_{\mathcal{S}}}$ —and to populate the local datastore.

In the analysis, for simplicity, we deviate by this operation and consider that each local dataset  $\mathcal{S}_t$  is split in two disjoint parts ( $\mathcal{S}_t = \mathcal{S}'_t \cup \mathcal{S}''_t$ ), with  $\mathcal{S}'_t$  used to train the base model and  $\mathcal{S}''_t$  used to populate the local datastore. Moreover, we assume that the two parts have the same size, i.e.,  $n'_t = n''_t = n_t/2$  for all  $t \in [T]$ , where  $n'_t$  and  $n''_t$  denote the size of  $\mathcal{S}'_t$  and  $\mathcal{S}''_t$ , respectively. In general, the result holds if the two parts have a fixed relative size across clients (i.e.,  $n'_{t_1}/n_{t_1} = n'_{t_2}/n_{t_2}$  for all  $t_1$  and  $t_2$  in  $[T]$ ).

Let  $\mathcal{S}'$  denote the whole data used to train the base model, i.e.,  $\mathcal{S}' = \bigcup_{m \in [M]} \mathcal{S}'_m$ . We observe that the base model  $h_{\mathcal{S}}$  is only function of  $\mathcal{S}'$ , and then we can write  $h_{\mathcal{S}'}$ . Instead, the local model  $h_{\mathcal{S}''_m}^{(1)}$  is both a function of  $\mathcal{S}'$  (used to learn the shared representation  $\phi'_{\mathcal{S}}$ ) and of  $\mathcal{S}''_m$  (used to populate the datastore). In order to stress such dependence, we then write  $h_{\mathcal{S}''_m, \mathcal{S}'}^{(1)}$ .

### F.1 Proof of Theorem 3.6.1

**Theorem 3.6.1.** *Suppose that Assumptions 18–21 hold, and consider  $t \in [T]$  and  $\lambda_t \in (0, 1)$ , then there exist constants  $c_1, c_2, c_3, c_4$ , and  $c_5 \in \mathbb{R}$ , such that*

$$\begin{aligned} \mathbb{E}_{\mathcal{S} \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t}} [\mathcal{L}_{\mathcal{D}_t}(h_{t, \lambda_t})] &\leq (1 + \lambda_t) \cdot \mathcal{L}_{\mathcal{D}_t}(h_t^*) + c_1 (1 - \lambda_t) \cdot \left( \text{disc}_{\mathcal{H}}(\bar{\mathcal{D}}, \mathcal{D}_t) + 1 \right) \\ &+ c_2 \lambda_t \cdot \frac{\sqrt{p}}{p+1\sqrt{n_t}} \cdot \text{disc}_{\mathcal{H}}(\bar{\mathcal{D}}, \mathcal{D}_t) + c_3 (1 - \lambda_t) \cdot \sqrt{\frac{d_{\mathcal{H}}}{n}} \cdot \sqrt{c_4 + \log\left(\frac{n}{d_{\mathcal{H}}}\right)} \\ &+ c_5 \lambda_t \cdot \sqrt{\frac{d_{\mathcal{H}}}{n}} \cdot \sqrt{c_4 + \log\left(\frac{n}{d_{\mathcal{H}}}\right)} \cdot \frac{\sqrt{p}}{p+1\sqrt{n_t}}, \end{aligned} \quad (\text{F.631})$$

where  $d_{\mathcal{H}}$  is the VC dimension of the hypothesis class  $\mathcal{H}$ ,  $n = \sum_{t=1}^T n_t$ ,  $\bar{\mathcal{D}} = \sum_{t=1}^T \frac{n_t}{n} \cdot \mathcal{D}_t$ ,  $p$  is the dimension of representations, and  $\text{disc}_{\mathcal{H}}$  is the label discrepancy associated to the hypothesis class  $\mathcal{H}$ .

*Proof.* The idea of the proof is to bound both the expected error of the *shared* base model (Lemma F.1) and the error of the local  $k\text{NN}$  retrieval mechanism (Lemma F.2) before using the convexity of the loss function to bound the error of  $h_{t, \lambda_t}$ .

Consider  $\mathcal{S} \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t}$  or, equivalently,  $\mathcal{S} = \mathcal{S}' \cup \mathcal{S}''$ , where  $\mathcal{S}' \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t/2}$ , and  $\mathcal{S}'' = \bigcup_{t \in [T]} \mathcal{S}''_t$  and  $\mathcal{S}''_t \sim \mathcal{D}_t^{n_t/2}$ .

For  $t \in [T]$ , and  $\lambda_t \in (0, 1)$ , we have

$$h_{t, \lambda_t} = \lambda_t \cdot h_{\mathcal{S}''_t, \mathcal{S}'}^{(1)} + (1 - \lambda_t) \cdot h_{\mathcal{S}'}. \quad (\text{F.632})$$

From Assumption 20 and the linearity of the expectation, it follows

$$\mathcal{L}_{\mathcal{D}_t}(h_{t, \lambda_t}) \leq \lambda_t \cdot \mathcal{L}_{\mathcal{D}_t}(h_{\mathcal{S}''_t, \mathcal{S}'}^{(1)}) + (1 - \lambda_t) \cdot \mathcal{L}_{\mathcal{D}_t}(h_{\mathcal{S}'}). \quad (\text{F.633})$$

Using Lemma F.2 and Lemma F.1, and applying expectation over samples  $\mathcal{S} \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t}$ , we have

$$\begin{aligned} \mathbb{E}_{\mathcal{S} \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t}} [\mathcal{L}_{\mathcal{D}_t}(h_t, \lambda_t)] &\leq \lambda_t \cdot \mathbb{E}_{\mathcal{S}' \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t/2}} \left[ \mathbb{E}_{\mathcal{S}'' \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t/2}} \left[ \mathcal{L}_{\mathcal{D}_t}(h_{\mathcal{S}''}^{(1)}, \mathcal{S}') \right] \right] \\ &\quad + (1 - \lambda_t) \cdot \mathbb{E}_{\mathcal{S}' \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t/2}} \left[ \mathbb{E}_{\mathcal{S}'' \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t/2}} \left[ \mathcal{L}_{\mathcal{D}_t}(h_{\mathcal{S}'}') \right] \right] \end{aligned} \quad (\text{F.634})$$

$$\begin{aligned} &\leq 2\lambda_t \mathcal{L}_{\mathcal{D}_t}(h_t^*) + 6\lambda_t \gamma_1 \frac{\sqrt{p}}{p+1/\sqrt{n_t}} \\ &\quad + 6\lambda_t \gamma_2 \frac{\sqrt{p}}{p+1/\sqrt{n_t}} \cdot \left( \mathbb{E}_{\mathcal{S}' \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t/2}} [\mathcal{L}_{\mathcal{D}_t}(h_{\mathcal{S}'}')] - \mathcal{L}_{\mathcal{D}_t}(h_t^*) \right) \\ &\quad + (1 - \lambda_t) \cdot \mathbb{E}_{\mathcal{S}' \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t/2}} [\mathcal{L}_{\mathcal{D}_t}(h_{\mathcal{S}'}')] \end{aligned} \quad (\text{F.635})$$

$$\begin{aligned} &\leq 2\lambda_t \mathcal{L}_{\mathcal{D}_t}(h_t^*) + 6\lambda_t \gamma_1 \frac{\sqrt{p}}{p+1/\sqrt{n_t}} \\ &\quad + 6\lambda_t \gamma_2 \frac{\sqrt{p}}{p+1/\sqrt{n_t}} \cdot \left( \delta_1 \cdot \sqrt{\frac{d_{\mathcal{H}}}{n}} \cdot \sqrt{\delta_2 + \log\left(\frac{n}{d_{\mathcal{H}}}\right)} + 2 \cdot \text{disc}_{\mathcal{H}}(\bar{\mathcal{D}}, \mathcal{D}_t) \right) \\ &\quad + (1 - \lambda_t) \cdot \left( \mathcal{L}_{\mathcal{D}_t}(h_t^*) + \delta_1 \cdot \sqrt{\frac{d_{\mathcal{H}}}{n}} \cdot \sqrt{\delta_2 + \log\left(\frac{n}{d_{\mathcal{H}}}\right)} + 2 \cdot \text{disc}_{\mathcal{H}}(\bar{\mathcal{D}}, \mathcal{D}_t) \right) \end{aligned} \quad (\text{F.636})$$

$$\begin{aligned} &= (1 + \lambda_t) \mathcal{L}_{\mathcal{D}_t}(h_t^*) + 6\lambda_t \gamma_1 \frac{\sqrt{p}}{p+1/\sqrt{n_t}} \\ &\quad + 6\lambda_t \gamma_2 \frac{\sqrt{p}}{p+1/\sqrt{n_t}} \delta_1 \cdot \sqrt{\frac{d_{\mathcal{H}}}{n}} \cdot \sqrt{\delta_2 + \log\left(\frac{n}{d_{\mathcal{H}}}\right)} + 12\lambda_t \gamma_2 \frac{\sqrt{p}}{p+1/\sqrt{n_t}} \cdot \text{disc}_{\mathcal{H}}(\bar{\mathcal{D}}, \mathcal{D}_t) \\ &\quad + \delta_1 (1 - \lambda_t) \cdot \sqrt{\frac{d_{\mathcal{H}}}{n}} \cdot \sqrt{\delta_2 + \log\left(\frac{n}{d_{\mathcal{H}}}\right)} + 2 \cdot (1 - \lambda_t) \text{disc}_{\mathcal{H}}(\bar{\mathcal{D}}, \mathcal{D}_t). \end{aligned} \quad (\text{F.637})$$

Rearranging the terms and taking  $c_1 \triangleq 2$ ,  $c_2 \triangleq \max\{12\gamma_2, 6\gamma_1\}$ ,  $c_3 \triangleq \delta_1$ ,  $c_4 \triangleq \delta_2$  and  $c_5 \triangleq 6\gamma_2\delta_1$ , the final result follows.  $\square$

## F.2 Intermediate Lemmas

**Lemma F.1.** Consider  $t \in [T]$ , then there exists constants  $\delta_1, \delta_2 \in \mathbb{R}$  such that

$$\mathbb{E}_{\mathcal{S}' \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t/2}} [\mathcal{L}_{\mathcal{D}_t}(h_{\mathcal{S}'}')] \leq \mathcal{L}_{\mathcal{D}_t}(h_t^*) + \delta_1 \cdot \sqrt{\frac{d_{\mathcal{H}}}{n}} \cdot \sqrt{\delta_2 + \log\left(\frac{n}{d_{\mathcal{H}}}\right)} + 2 \cdot \text{disc}_{\mathcal{H}}(\bar{\mathcal{D}}, \mathcal{D}_t), \quad (\text{F.638})$$

where  $d$  is the VC dimension of the hypothesis class  $\mathcal{H}$ ,  $\bar{\mathcal{D}} = \sum_{t=1}^T \frac{n_t}{n} \cdot \mathcal{D}_t$  and  $\text{disc}_{\mathcal{H}}$  is the label discrepancy associated to the hypothesis class  $\mathcal{H}$ .

*Proof.* We remind that the label discrepancy associated to the hypothesis class  $\mathcal{H}$  for two distributions  $\mathcal{D}_1$  and  $\mathcal{D}_2$  over features and labels is defined as [Man+20]:

$$\text{disc}_{\mathcal{H}}(\mathcal{D}_1, \mathcal{D}_2) = \max_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{D}_1}(h) - \mathcal{L}_{\mathcal{D}_2}(h)|. \quad (\text{F.639})$$



Consider  $t \in [T]$  and  $h^* \in \arg \min_{h \in \mathcal{H}} \mathcal{L}_{\bar{\mathcal{D}}}(h)$ . For  $S' \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t/2}$ , we have

$$\begin{aligned} & \mathcal{L}_{\mathcal{D}_t}(h_{S'}) - \mathcal{L}_{\mathcal{D}_t}(h_t^*) \\ &= \mathcal{L}_{\mathcal{D}_t}(h_{S'}) - \mathcal{L}_{\bar{\mathcal{D}}}(h_{S'}) + \mathcal{L}_{\bar{\mathcal{D}}}(h_{S'}) - \mathcal{L}_{\bar{\mathcal{D}}}(h_t^*) + \mathcal{L}_{\bar{\mathcal{D}}}(h_t^*) - \mathcal{L}_{\bar{\mathcal{D}}}(h^*) + \mathcal{L}_{\bar{\mathcal{D}}}(h^*) - \mathcal{L}_{\mathcal{D}_t}(h_t^*) \end{aligned} \quad (\text{F.640})$$

$$\begin{aligned} &= \underbrace{\mathcal{L}_{\mathcal{D}_t}(h_{S'}) - \mathcal{L}_{\bar{\mathcal{D}}}(h_{S'})}_{\leq \text{disc}_{\mathcal{H}}(\mathcal{D}_t, \bar{\mathcal{D}})} + \underbrace{\mathcal{L}_{\bar{\mathcal{D}}}(h_t^*) - \mathcal{L}_{\mathcal{D}_t}(h_t^*)}_{\leq \text{disc}_{\mathcal{H}}(\mathcal{D}_t, \bar{\mathcal{D}})} + \underbrace{\mathcal{L}_{\bar{\mathcal{D}}}(h^*) - \mathcal{L}_{\bar{\mathcal{D}}}(h_t^*)}_{\leq 0} + \mathcal{L}_{\bar{\mathcal{D}}}(h_{S'}) - \mathcal{L}_{\bar{\mathcal{D}}}(h^*) \end{aligned} \quad (\text{F.641})$$

$$\leq 2 \cdot \text{disc}_{\mathcal{H}}(\mathcal{D}_t, \bar{\mathcal{D}}) + \mathcal{L}_{\bar{\mathcal{D}}}(h_{S'}) - \mathcal{L}_{\bar{\mathcal{D}}}(h^*) \quad (\text{F.642})$$

$$\begin{aligned} &= 2 \cdot \text{disc}_{\mathcal{H}}(\mathcal{D}_t, \bar{\mathcal{D}}) + \mathcal{L}_{\bar{\mathcal{D}}}(h_{S'}) - \mathcal{L}_{S'}(h_{S'}) + \underbrace{\mathcal{L}_{S'}(h_{S'}) - \mathcal{L}_{S'}(h^*)}_{\leq 0} + \mathcal{L}_{S'}(h^*) - \mathcal{L}_{\bar{\mathcal{D}}}(h^*) \end{aligned} \quad (\text{F.643})$$

$$\leq 2 \cdot \text{disc}_{\mathcal{H}}(\mathcal{D}_t, \bar{\mathcal{D}}) + 2 \cdot \sup_{h \in \mathcal{H}} |\mathcal{L}_{\bar{\mathcal{D}}}(h) - \mathcal{L}_{S'}(h)|. \quad (\text{F.644})$$

We now bound  $\mathbb{E}_{S' \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t/2}} \sup_{h \in \mathcal{H}} |\mathcal{L}_{\bar{\mathcal{D}}}(h) - \mathcal{L}_{S'}(h)|$ . We first observe that for every  $h \in \mathcal{H}$ , we can write  $\mathcal{L}_{\bar{\mathcal{D}}}(h) = \mathbb{E}_{S' \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t/2}} \mathcal{L}_{S'}(h)$ . Therefore, despite the fact that the samples in  $S'$  are not i.i.d., we can follow the same steps as in the proof of [SB14, Theorem 6.11], and conclude

$$\mathbb{E}_{S' \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t/2}} \sup_{h \in \mathcal{H}} |\mathcal{L}_{\bar{\mathcal{D}}}(h) - \mathcal{L}_{S'}(h)| \leq \frac{4 + \sqrt{\log(\tau_{\mathcal{H}}(n))}}{\sqrt{n}}, \quad (\text{F.645})$$

where  $\tau_{\mathcal{H}}$  is the growth function of class  $\mathcal{H}$ .

Let  $d$  denote the VC dimension of  $\mathcal{H}$ . From Sauer's lemma [SB14, Lemma 6.10], we have that for  $n > d + 1$ ,  $\tau_{\mathcal{H}}(n) \leq (en/d)^{d_{\mathcal{H}}}$ . Therefore, there exist constants  $\delta_1, \delta_2 \in \mathbb{R}$  (e.g.,  $\delta_1 = 4$ ,  $\delta_2 = \max\{4/\sqrt{d_{\mathcal{H}}}, 1\}$ ), such that

$$\mathbb{E}_{S' \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t/2}} \sup_{h \in \mathcal{H}} |\mathcal{L}_{\bar{\mathcal{D}}}(h) - \mathcal{L}_{S'}(h)| \leq \frac{\delta_1}{2} \cdot \sqrt{\frac{d_{\mathcal{H}}}{n}} \cdot \sqrt{\delta_2 + \log\left(\frac{n}{d_{\mathcal{H}}}\right)}. \quad (\text{F.646})$$

Taking the expectation in (F.644) and using this inequality, we have

$$\mathbb{E}_{S' \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t/2}} [\mathcal{L}_{\mathcal{D}_t}(h_{S'})] \leq \mathcal{L}_{\mathcal{D}_t}(h_t^*) + \delta_1 \cdot \sqrt{\frac{d_{\mathcal{H}}}{n}} \cdot \sqrt{\delta_2 + \log\left(\frac{n}{d_{\mathcal{H}}}\right)} + 2 \cdot \text{disc}_{\mathcal{H}}(\bar{\mathcal{D}}, \mathcal{D}_t). \quad (\text{F.647})$$

□

The following Lemma proves an upper bound on the expected error of the 1-NN learning rule.

**Lemma F.2** (Adapted from [SB14, Thm 19.3]). *Under Assumptions 18, 19, and 21 for all  $t \in [T]$ , it holds*

$$\mathbb{E}_{S''_t \sim \mathcal{D}_t^{n_t/2}} [\mathcal{L}_{\mathcal{D}_t}(h_{S''_t}^{(1)})] \leq 2\mathcal{L}_{\mathcal{D}_t}(h_t^*) + 6 \left\{ \gamma_1 + \gamma_2 \cdot [\mathcal{L}_{\mathcal{D}_t}(h_{S'}) - \mathcal{L}_{\mathcal{D}_t}(h_t^*)] \right\} \cdot \frac{\sqrt{p}}{p+1/\sqrt{n_t}}. \quad (\text{F.648})$$

*Proof.* Recall that for  $t \in [T]$ , the Bayes optimal rule, i.e., the hypothesis that minimizes  $\mathcal{L}_{\mathcal{D}_t}(h)$  over all functions, is

$$h_t^*(\mathbf{x}) = \mathbb{1}_{\{\eta_t(\mathbf{x}) > 1/2\}}. \quad (\text{F.649})$$

We note that the 1-NN rule can be expressed as follows:

$$\left[ h_{S_t'', S'}^{(1)}(\mathbf{x}) \right]_y = \mathbb{1}_{\{y = \pi_{S_t''}^{(1)}(\mathbf{x})\}}, \quad (\text{F.650})$$

where we are putting in evidence that the permutation  $\pi_t$  depends on the dataset  $S_t''$ . Then, under Assumption 19, the loss function  $l(\cdot)$  reduces to the 0-1 loss.

Consider samples  $\mathcal{S} \sim \otimes_{t=1}^T \mathcal{D}_t^{n_t}$ . Using Assumptions 18, 19 and 21, and following the same steps as in [SB14, Lemma 19.1], we have

$$\begin{aligned} \mathbb{E}_{S_t'' \sim \mathcal{D}_t^{n_t/2}} \left[ \mathcal{L}_{\mathcal{D}_t} \left( h_{S_t'', S'}^{(1)} \right) \right] - 2\mathcal{L}_{\mathcal{D}_t} \left( h_t^* \right) &\leq \\ &\left\{ \gamma_1 + \gamma_2 \cdot \left[ \mathcal{L}_{\mathcal{D}_t} \left( h_{S'} \right) - \mathcal{L}_{\mathcal{D}_t} \left( h_t^* \right) \right] \right\} \times \underbrace{\mathbb{E}_{S_t'', \mathcal{X} \sim \mathcal{D}_{t, \mathcal{X}}^{n_t/2}, \mathbf{x} \sim \mathcal{D}_{t, \mathcal{X}}} \left[ d \left( \phi_{h_{S'}}(\mathbf{x}), \phi_{h_{S'}} \left( \pi_{S_t''}^{(1)}(\mathbf{x}) \right) \right) \right]}_{\triangleq \mathcal{T}_{S'}} \end{aligned} \quad (\text{F.651})$$

where  $S_{t, \mathcal{X}}''$  denotes the set of input features in the dataset  $S_t''$  and  $\mathcal{D}_{t, \mathcal{X}}$  the marginal distribution of  $\mathcal{D}_t$  over  $\mathcal{X}$ . Note that  $S_t''$  is independent from  $S'$ .

As in the proof of [SB14, Theorem 19.3], let  $K$  be an integer to be precised later on. We consider  $r = K^p$  and  $C_1, \dots, C_r$  to be the cover of the set  $[0, 1]^p$  using boxes with side  $1/T$ . We bound the term  $\mathcal{T}_{S'}$  independently from  $S'$  as follows

$$\mathbb{E}_{S_t'' \sim \mathcal{D}_{t, \mathcal{X}}^{n_t/2}, \mathbf{x} \sim \mathcal{D}_{t, \mathcal{X}}} \left[ d \left( \phi_{h_{S'}}(\mathbf{x}), \phi_{h_{S'}} \left( \pi_{S_t''}^{(1)}(\mathbf{x}) \right) \right) \right] \leq \sqrt{p} \left( \frac{2K^p}{n_t e} + \frac{1}{K} \right). \quad (\text{F.652})$$

If we set  $\epsilon = 2 \left( \frac{2}{n_t} \right)^{\frac{1}{p+1}}$  and  $K = \lceil 1/\epsilon \rceil$ , it follows  $1/\epsilon \leq K < 2/\epsilon$  and then

$$\mathbb{E}_{S_t'' \sim \mathcal{D}_{t, \mathcal{X}}^{n_t/2}, \mathbf{x} \sim \mathcal{D}_{t, \mathcal{X}}} \left[ d \left( \phi_{h_{S'}}(\mathbf{x}), \phi_{h_{S'}} \left( \pi_{S_t''}^{(1)}(\mathbf{x}) \right) \right) \right] \leq \sqrt{p} \left( \frac{2(2/\epsilon)^p}{n_t e} + \epsilon \right) \quad (\text{F.653})$$

$$= \sqrt{p} \left( \frac{1}{e} + 2 \right) \left( \frac{2}{n_t} \right)^{\frac{1}{p+1}} \quad (\text{F.654})$$

$$\leq 6 \frac{\sqrt{p}}{e^{p+1} n_t}. \quad (\text{F.655})$$

Thus,

$$\mathbb{E}_{S_t'' \sim \mathcal{D}_t^{n_t}} \left[ \mathcal{L}_{\mathcal{D}_t} \left( h_{S_t'', S'}^{(1)} \right) \right] \leq 2\mathcal{L}_{\mathcal{D}_t} \left( h_t^* \right) + 6 \frac{\sqrt{p}}{e^{p+1} n_t} \left\{ \gamma_1 + \gamma_2 \cdot \left[ \mathcal{L}_{\mathcal{D}_t} \left( h_{S'} \right) - \mathcal{L}_{\mathcal{D}_t} \left( h_t^* \right) \right] \right\}. \quad (\text{F.656})$$

□

## G Federated Learning for Data Streams

### G.1 Proofs

We remind that all our results rely on the following assumptions:

**Assumption 18.** (*Bounded loss*) The loss function is bounded, i.e.,  $\forall \theta \in \Theta, \mathbf{z} \in \mathcal{Z}, \ell(\theta; \mathbf{z}) \in [0, B]$

**Assumption 23.** (*Bounded domain*) We suppose that  $\Theta$  is convex, closed and bounded; we use  $D$  to denote its diameter, i.e.,  $\forall \theta, \theta' \in \Theta, \|\theta - \theta'\| \leq D$ .

**Assumption 24.** (*Convexity*) For all  $\mathbf{z} \in \mathcal{Z}$ , the function  $\theta \mapsto \ell(\theta; \mathbf{z})$  is convex on  $\mathbb{R}^d$ .

**Assumption 12.** (*Smoothness*) For all  $\mathbf{z} \in \mathcal{Z}$ , the function  $\theta \mapsto \ell(\theta; \mathbf{z})$  is  $L$ -smooth on  $\mathbb{R}^d$ .

In what follows, we use  $\Delta^{D-1}$  to denote the unitary simplex of dimension  $D - 1$ , i.e.,  $\Delta^{D-1} = \{\mathbf{f} \in \mathbb{R}_+^D, \sum_{i=1}^D f_i = 1\}$

#### G.1.1 Proof of (4.9)

$$\begin{aligned} \epsilon_{\text{true}} &= \mathbb{E}_{\mathcal{S}, A^{(\lambda)}} \left[ \mathcal{L}_{\mathcal{P}(\alpha)} \left( A^{(\lambda)}(\mathcal{S}) \right) - \mathcal{L}_{\mathcal{S}}^{(\lambda)} \left( A^{(\lambda)}(\mathcal{S}) \right) \right] + \mathbb{E}_{\mathcal{S}, A^{(\lambda)}} \left[ \mathcal{L}_{\mathcal{S}}^{(\lambda)} \left( A^{(\lambda)}(\mathcal{S}) \right) - \min_{\theta \in \Theta} \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) \right] \\ &\quad + \mathbb{E}_{\mathcal{S}} \left[ \min_{\theta \in \Theta} \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) \right] - \min_{\theta \in \Theta} \mathcal{L}_{\mathcal{P}(\alpha)}(\theta) \tag{G.657} \\ &\leq 2 \underbrace{\mathbb{E}_{\mathcal{S}} \left[ \sup_{\theta \in \Theta} \left| \mathcal{L}_{\mathcal{P}(\alpha)}(\theta) - \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) \right| \right]}_{\triangleq \epsilon_{\text{gen}}} + \underbrace{\mathbb{E}_{\mathcal{S}, A^{(\lambda)}} \left[ \mathcal{L}_{\mathcal{S}}^{(\lambda)} \left( A^{(\lambda)}(\mathcal{S}) \right) - \min_{\theta \in \Theta} \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) \right]}_{\triangleq \epsilon_{\text{opt}}}, \tag{G.658} \end{aligned}$$

where we exploited the fact that  $\min_{x \in X} f(x) - \min_{x \in X} g(x) \leq \sup_{x \in X} |f(x) - g(x)|$ .

#### G.1.2 Properties

**Lemma G.1.** Let  $f$  be an  $L$ -smooth function taking values in  $[0, B]$ , then  $\|\nabla f\| \leq \sqrt{2LB}$ .

*Proof.* Let  $\theta \in \Theta$ , then using the definition of the  $L$ -smoothness of  $f$  with  $\theta' = \theta - \frac{1}{L} \nabla f(\theta)$ , we have

$$f(\theta') = f\left(\theta - \frac{1}{L} \nabla f(\theta)\right) \leq f(\theta) - \frac{1}{L} \langle \nabla f(\theta), \nabla f(\theta) \rangle + \frac{L}{2} \left\| \frac{1}{L} \nabla f(\theta) \right\|^2 \tag{G.659}$$

$$= f(\theta) - \frac{1}{2L} \|\nabla f(\theta)\|^2. \tag{G.660}$$

It follows that,

$$\|\nabla f(\theta)\|^2 \leq 2L(f(\theta) - f(\theta')) \leq 2LB. \tag{G.661}$$

□

**Lemma G.2.** *Suppose that Assumptions 18, and 12 hold. For all*

$$\sup_{\theta \in \Theta} \|\nabla \ell(\theta; \mathbf{z}) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta)\|^2 \leq \left(2\sqrt{2LB}\right)^2 \quad (\text{G.662})$$

*Proof.* Let  $\mathbf{z} \in \mathcal{Z}$ , and  $m \in [M]$ . Both  $\ell(\cdot, \mathbf{z})$ , and  $\mathcal{L}_{\mathcal{P}_m}$  are  $L$ -smooth and bounded within  $[0, B]$ .

For  $\theta \in \Theta$ , we have

$$\|\nabla \ell(\theta; \mathbf{z}) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta)\|^2 \leq 2 \|\nabla \ell(\theta; \mathbf{z})\|^2 + 2 \|\nabla \mathcal{L}_{\mathcal{P}_m}(\theta)\|^2 \quad (\text{G.663})$$

$$\leq 2 \cdot 2LB + 2 \cdot 2LB \quad (\text{G.664})$$

$$= 8LB = \left(2\sqrt{2LB}\right)^2, \quad (\text{G.665})$$

where we used Lemma G.1 to obtain the last inequality.  $\square$

**Lemma G.3.** *Suppose that Assumptions 18, and 12 hold. For all  $\mathbf{z} \in \mathcal{Z}$ , we have*

$$\max_{m, m'} \sup_{\theta \in \Theta} \left\| \nabla \mathcal{L}_{\mathcal{P}_{m'}}(\theta) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) \right\| \leq 2\sqrt{2LB}. \quad (\text{G.666})$$

*Proof.* The proof follows using the triangular inequality and Lemma G.1.  $\square$

### G.1.3 Proof of Theorem 4.3.1

In this section we express the loss  $\ell$  as a function of the hypothesis function  $h \in \mathcal{H}$ , rather than as a function of the parameter vector  $\theta \in \Theta$ .

### G.1.4 A Particular Case: Binary Classification with 0–1 loss

We first prove the result in the particular case when  $\mathcal{Y} = \{0, 1\}$ , and the loss function is the 0–1 loss..

**Theorem G.4.** *Suppose that  $\mathcal{Y} = \{0, 1\}$ , and the loss function is the 0–1 loss, when using Algorithm 13 with weights  $\lambda$ , it follows that*

$$\epsilon_{gen} \leq \text{disc}_{\mathcal{H}}(\mathcal{P}^{(\alpha)}, \mathcal{P}^{(\mathbf{p})}) + \tilde{O}\left(\sqrt{\frac{\text{VCdim}(\mathcal{H})}{N_{\text{eff}}}}\right),$$

where

$$p_{m,i} = \frac{\sum_{t=1}^T \sum_{j \in \mathcal{I}_m^{(t)}} \mathbb{1}\{j = i\} \cdot \lambda_m^{(t,j)}}{\sum_{m'=1}^M \sum_{t=1}^T \sum_{j \in \mathcal{I}_{m'}^{(t)}} \lambda_{m'}^{(t,j)}}, \quad i \in [N_m],$$

$$\mathbf{p} = \left( \sum_{i=1}^{N_m} p_{m,i} \right)_{1 \leq m \leq M},$$

$$N_{\text{eff}} = \left( \sum_{m=1}^M \sum_{i=1}^{N_m} p_{m,i}^2 \right)^{-1}.$$

*Proof.* For client,  $m \in [M]$ , we remind that  $p_m \triangleq \sum_{i=1}^{N_m} p_{m,i}$  is the relative importance of client  $m$  in comparison to the other clients. We define

$$\mathcal{L}_{\mathcal{S}, \mathbf{p}} = \sum_{m=1}^M \sum_{i=1}^{N_m} p_{m,i} \cdot \ell(\cdot; \mathbf{z}_m^{(i)}). \quad (\text{G.667})$$

Note that  $\mathcal{L}_{\mathcal{S}, \mathbf{p}} = \mathcal{L}_{\mathcal{S}}^{(\lambda)}$ , and  $\mathbb{E}_{\mathcal{S}} [\mathcal{L}_{\mathcal{S}, \mathbf{p}}(\theta)] = \sum_m p_m \mathcal{L}_{\mathcal{P}_m}(\theta) = \mathcal{L}_{\mathcal{P}(\mathbf{p})}(\theta)$  for any  $\theta \in \Theta$ , where  $\mathcal{P}(\mathbf{p}) = \sum_m p_m \mathcal{P}_m$ . We have

$$\epsilon_{\text{gen}} = \mathbb{E}_{\mathcal{S}} \left[ \sup_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{P}(\alpha)}(h) - \mathcal{L}_{\mathcal{S}, \mathbf{p}}(h)| \right] \quad (\text{G.668})$$

$$= \mathbb{E}_{\mathcal{S}} \left[ \sup_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{P}(\alpha)}(h) - \mathcal{L}_{\mathcal{P}(\mathbf{p})}(h) + \mathcal{L}_{\mathcal{P}(\mathbf{p})}(h) - \mathcal{L}_{\mathcal{S}, \mathbf{p}}(h)| \right] \quad (\text{G.669})$$

$$\leq \mathbb{E}_{\mathcal{S}} \left[ \sup_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{P}(\alpha)}(h) - \mathcal{L}_{\mathcal{P}(\mathbf{p})}(h)| \right] + \mathbb{E}_{\mathcal{S}} \left[ \sup_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{P}(\mathbf{p})}(h) - \mathcal{L}_{\mathcal{S}, \mathbf{p}}(h)| \right] \quad (\text{G.670})$$

$$\leq \text{disc}_{\mathcal{H}}(\mathcal{P}^{(\alpha)}, \mathcal{P}(\mathbf{p})) + \mathbb{E}_{\mathcal{S}} \left[ \sup_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{P}(\mathbf{p})}(h) - \mathcal{L}_{\mathcal{S}, \mathbf{p}}(h)| \right]. \quad (\text{G.671})$$

We bound now the second term in the right-hand side of Eq. (G.671). Note that, for  $h \in \mathcal{H}$ , we can write  $\mathcal{L}_{\mathcal{P}(\mathbf{p})}(h) = \mathbb{E}_{\mathcal{S}'}$   $[\mathcal{L}_{\mathcal{S}', \mathbf{p}}(h)]$ , where  $\mathcal{S}' = \bigcup_{m=1}^M \mathcal{S}'_m$  and  $\mathcal{S}'_m \sim \mathcal{P}_m^{N_m}$  is a dataset of  $N_m$  samples drawn i.i.d. from  $\mathcal{P}_m$  such that  $\mathcal{S}_m = \{z_m^{(i)}, i \in [N_m]\}$  and  $\mathcal{S}'_m = \{z'_m{}^{(i)}, i \in [N_m]\}$ . Using triangular inequality, it follows that

$$\mathbb{E}_{\mathcal{S}} \left[ \sup_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{P}(\mathbf{p})}(h) - \mathcal{L}_{\mathcal{S}, \mathbf{p}}(h)| \right] \leq \mathbb{E}_{\mathcal{S}, \mathcal{S}'} \left[ \sup_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{S}', \mathbf{p}}(h) - \mathcal{L}_{\mathcal{S}, \mathbf{p}}(h)| \right] \quad (\text{G.672})$$

$$= \mathbb{E}_{\mathcal{S}, \mathcal{S}'} \left[ \sup_{h \in \mathcal{H}} \left| \sum_{m=1}^M \sum_{i=1}^{N_m} p_{m,i} \left( \ell(h; z_m^{(i)}) - \ell(h; z'_m{}^{(i)}) \right) \right| \right] \quad (\text{G.673})$$

$$= \mathbb{E}_{\mathcal{S}, \mathcal{S}'} \mathbb{E}_{\sigma} \left[ \sup_{h \in \mathcal{H}} \left| \sum_{m=1}^M \sum_{i=1}^{N_m} \sigma_m^{(i)} \cdot p_{m,i} \left( \ell(h; z_m^{(i)}) - \ell(h; z'_m{}^{(i)}) \right) \right| \right], \quad (\text{G.674})$$

where  $\sigma_m^{(i)}$ ,  $m \in [M]$ ,  $i \in [N_m]$  is a random variable drawn from uniform distribution over  $\{\pm 1\}$ . Fix  $\mathcal{S}$  and  $\mathcal{S}'$  and let  $C$  be the instances appearing in  $\mathcal{S}$  and  $\mathcal{S}'$ , and  $\mathcal{H}_C$  be the restriction of  $\mathcal{H}$  to  $C$ , as defined in [SB14, Defintion 6.2]. It follows that

$$\mathbb{E}_{\mathcal{S}} \left[ \sup_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{P}(\mathbf{p})}(h) - \mathcal{L}_{\mathcal{S}, \mathbf{p}}(h)| \right] \leq \mathbb{E}_{\mathcal{S}, \mathcal{S}'} \mathbb{E}_{\sigma} \left[ \sup_{h \in \mathcal{H}_C} \left| \sum_{m=1}^M \sum_{i=1}^{N_m} \sigma_m^{(i)} \cdot p_{m,i} \left( \ell(h; z_m^{(i)}) - \ell(h; z'_m{}^{(i)}) \right) \right| \right]. \quad (\text{G.675})$$

Fix some  $h \in \mathcal{H}_C$  and denote  $\gamma_m^{(i)} = \sigma_m^{(i)} \cdot p_{m,i} \left( \ell(h; z_m^{(i)}) - \ell(h; z'_m{}^{(i)}) \right)$  for  $m \in [M]$  and  $i \in [N_m]$ . We have that  $\mathbb{E} [\gamma_m^{(i)}] = 0$  and from Assumption 18, we have that  $\gamma_m^{(i)} \in [-p_{m,i}, p_{m,i}]$ . Since

the random variables  $\{\gamma_m^{(i)}, m \in [M], i \in [N_m]\}$  are independent, using Hoeffding inequality it follows that, for all  $\rho \geq 0$ , we have

$$\mathbb{P} \left[ \left| \sum_{m=1}^M \sum_{i=1}^{N_m} \sigma_m^{(i)} \cdot p_{m,i} \left( \ell(h; z_m^{(i)}) - \ell(h; z_m'^{(i)}) \right) \right| \geq \rho \right] \leq 2 \exp \left( -2N_{\text{eff}} \rho^2 \right), \quad (\text{G.676})$$

where  $N_{\text{eff}} = \left( \sum_{m=1}^M \sum_{i=1}^{N_m} (p_{m,i})^2 \right)^{-1}$ . Applying the union bound over  $h \in \mathcal{H}_C$  and using [SB14, Lemma A.4],\* it follows that

$$\mathbb{E}_\sigma \left[ \sup_{h \in \mathcal{H}_C} \left| \sum_{m=1}^M \sum_{i=1}^{N_m} \sigma_m^{(i)} \cdot p_{m,i} \left( \ell(h; z_m^{(i)}) - \ell(h; z_m'^{(i)}) \right) \right| \right] \leq \frac{4 + 3\sqrt{\log(|\mathcal{H}_C|)}}{\sqrt{2N_{\text{eff}}}}. \quad (\text{G.677})$$

Let  $\tau_{\mathcal{H}}$  be the growth function of  $\mathcal{H}$  as defined in [SB14, Definition 6.9]. It holds  $|H_{\Theta, C}| \leq \tau_{\mathcal{H}}(|C|) \leq \tau_{\mathcal{H}}(N)$ . This leads to:

$$\mathbb{E}_\sigma \left[ \sup_{h \in \mathcal{H}_C} \left| \sum_{m=1}^M \sum_{i=1}^{N_m} \sigma_m^{(i)} \cdot p_{m,i} \left( \ell(h; z_m^{(i)}) - \ell(h; z_m'^{(i)}) \right) \right| \right] \leq \frac{4 + 3\sqrt{\log(\tau_{\mathcal{H}}(N))}}{\sqrt{2N_{\text{eff}}}}. \quad (\text{G.678})$$

Replacing this bound in (G.675), we obtain:

$$\mathbb{E}_S \left[ \sup_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{P}(\mathfrak{p})}(h) - \mathcal{L}_{S, \mathfrak{p}}(h)| \right] \leq \frac{4 + 3\sqrt{\log(\tau_{\mathcal{H}}(N))}}{\sqrt{2N_{\text{eff}}}}, \quad (\text{G.679})$$

Using Sauer's Lemma [SB14, Lemma 6.10] and following the same steps as in the proof of [Mar+22b, Lemma A.1] we have

$$\mathbb{E}_S \left[ \sup_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{P}(\mathfrak{p})}(h) - \mathcal{L}_{S, \mathfrak{p}}(h)| \right] \leq 5 \sqrt{\frac{\text{VCdim}(\mathcal{H})}{N_{\text{eff}}}} \cdot \sqrt{1 + \log \left( \frac{N}{\text{VCdim}(\mathcal{H})} \right)}. \quad (\text{G.680})$$

Thus,

$$\mathbb{E}_S \left[ \sup_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{P}(\mathfrak{p})}(h) - \mathcal{L}_{S, \mathfrak{p}}(h)| \right] \leq \tilde{O} \left( \sqrt{\frac{\text{VCdim}(\mathcal{H})}{N_{\text{eff}}}} \right), \quad (\text{G.681})$$

thus,

$$\epsilon_{\text{gen}} \leq \tilde{O} \left( \sqrt{\frac{\text{VCdim}(\mathcal{H})}{N_{\text{eff}}}} \right) + \text{disc}_{\mathcal{H}}(\mathcal{P}^{(\alpha)}, \mathcal{P}^{(\mathfrak{p})}). \quad (\text{G.682})$$

□

---

\*If we follow the statement of [SB14, Lemma A.4], the RHS of (G.676) would be  $\frac{4+2\sqrt{\log(|\mathcal{H}_C|)}}{\sqrt{2N_{\text{eff}}}}$ . However, by carefully checking the proof of this lemma, we observe that there is a missing term. Including the missing term leads to a constant 3 rather than 2.

### G.1.5 The General Case

We remind the definition of the pseudo-dimension and shattering from [MRT18].

**Definition G.1.** [MRT18, Definition 11.4] Let  $\mathcal{F}$  be a family of functions mapping from  $\mathcal{X}$  to  $\mathbb{R}$ . A set  $\{\mathbf{x}_1, \dots, \mathbf{x}_m\}$  is said to be shattered by  $\mathcal{F}$  if there exists  $t_1, \dots, t_m \in \mathbb{R}$  such that,

$$\left| \left\{ \begin{bmatrix} \text{sgn}(f(\mathbf{x}_1) - t_1) \\ \vdots \\ \text{sgn}(f(\mathbf{x}_m) - t_m) \end{bmatrix} : f \in \mathcal{F} \right\} \right| = 2^m \quad (\text{G.683})$$

**Definition G.2.** [MRT18, Definition 11.5] Let  $\mathcal{F}$  be a family of functions mapping from  $\mathcal{X}$  to  $\mathbb{R}$ . Then, the pseudo-dimension of  $\mathcal{F}$ , denoted by  $\text{Pdim}(\mathcal{F})$ , is the size of the largest set shattered by  $\mathcal{F}$ .

The notion of pseudo-dimension of a family of real-valued functions coincides with that of the VC-dimension of the corresponding thresholded functions mapping  $\mathcal{X}$  to  $\{0, 1\}$ :

$$\text{Pdim}(\mathcal{F}) = \text{VCdim} \left( \left\{ (\mathbf{x}, s) \mapsto \mathbb{1}_{f(\mathbf{x}) - s > 0} : f \in \mathcal{F} \right\} \right). \quad (\text{G.684})$$

In particular, we call the pseudo-dimension of the family  $\ell \circ \mathcal{H} \triangleq \{z \mapsto \ell(h, z) : h \in \mathcal{H}\}$  the pseudo-dimension of the hypothesis class  $\mathcal{H}$  w.r.t. the loss  $\ell$ .

**Theorem 4.3.1.** *Suppose that Assumption 18 holds, when using Algorithm 13 with weights  $\lambda$ , it follows that*

$$\epsilon_{\text{gen}} \leq \text{disc}_{\mathcal{H}}(\mathcal{P}^{(\alpha)}, \mathcal{P}^{(\mathbf{p})}) + \tilde{O} \left( \sqrt{\frac{\text{Pdim}(\ell \circ \mathcal{H})}{N_{\text{eff}}}} \right),$$

where ,

$$p_{m,i} = \frac{\sum_{t=1}^T \sum_{j \in \mathcal{I}_m^{(t)}} \mathbb{1}\{j = i\} \cdot \lambda_m^{(t,j)}}{\sum_{m'=1}^M \sum_{t=1}^T \sum_{j \in \mathcal{I}_{m'}^{(t)}} \lambda_{m'}^{(t,j)}}, \quad i \in [N_m],$$

$$\mathbf{p} = \left( \sum_{i=1}^{N_m} p_{m,i} \right)_{1 \leq m \leq M},$$

$$N_{\text{eff}} = \left( \sum_{m=1}^M \sum_{i=1}^{N_m} p_{m,i}^2 \right)^{-1}.$$

*Proof.* Using exactly the same steps as in the proof of Theorem G.4, we obtain:

$$\epsilon_{\text{gen}} \leq \text{disc}_{\mathcal{H}}(\mathcal{P}^{(\alpha)}, \mathcal{P}^{(\mathbf{p})}) + \mathbb{E}_{\mathcal{S}} \left[ \sup_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{P}^{(\mathbf{p})}}(h) - \mathcal{L}_{\mathcal{S}, \mathbf{p}}(h)| \right]. \quad (\text{G.671})$$

The rest of the proof employs a technique similar to the one used in the proof of [MRT18, Theorem 11.8] in order to bound the second term in RHS of (G.671). The technique consists of reducing the problem of learning in  $\mathcal{H}$  to that of binary classification.

For  $h \in \mathcal{H}$  and  $t \in \mathbb{R}$ , we denote by  $c_{h,t}$  the classifier defined by  $c_{h,t} : (\mathbf{x}, y) \mapsto \mathbb{1}_{\ell(h, (\mathbf{x}, y)) > t}$ . For such classifier,  $\mathbf{z} \in \mathcal{Z}$  is an input vector and  $\bar{y} \in \{0, 1\}$  is a label. We denote by  $\tilde{\mathcal{H}} \triangleq \{c_{h,t} :$

$h \in \mathcal{H}, t \in [0, B]$  the hypothesis class of these binary classifiers. Let  $\bar{\mathcal{P}}^{(\mathbf{p})}$  denote the distribution over  $\bar{\mathcal{Z}} = \mathcal{Z} \times \{0, 1\}$ , such that  $\bar{\mathcal{P}}^{(\mathbf{p})}(\mathcal{Z} \times \{1\}) = 0$  and  $\bar{\mathcal{P}}^{(\mathbf{p})}(\cdot \times \{0\}) = \mathcal{P}^{(\mathbf{p})}(\cdot)$ , i.e., the label  $\bar{y} = 1$  is observed with probability 0, and the distribution of input vectors when  $\bar{y} = 0$  coincides with  $\bar{\mathcal{P}}^{(\mathbf{p})}$ . Finally, let  $\hat{\mathcal{P}}^{(\mathbf{p})}$  denote the empirical distribution where point  $\mathbf{z}_m^{(i)}$  is drawn with probability  $p_{m,i}$ .

We consider the 0–1 loss function  $\bar{\ell}(c_{h,t}, (\mathbf{z}, \bar{y})) \triangleq \mathbb{1}_{c_{h,t}(\mathbf{x}, y) \neq \bar{y}}$ . The expected risk of  $c_{h,t}$  is then

$$\bar{\mathcal{L}}_{\bar{\mathcal{P}}^{(\mathbf{p})}}(c_{h,t}) = \mathbb{E}_{\bar{\mathbf{z}} \sim \bar{\mathcal{P}}^{(\mathbf{p})}}[\bar{\ell}(c_{h,t}, (\bar{\mathbf{z}}))] = \mathbb{E}_{\mathbf{z} \sim \mathcal{P}^{(\mathbf{p})}}[c_{h,t}(\mathbf{z})] = \mathbb{P}_{\mathbf{z} \sim \mathcal{P}^{(\mathbf{p})}}[\ell(h, \mathbf{z}) > t]. \quad (\text{G.685})$$

Similarly, the (weighted) empirical risk of  $c_{h,t}$  is

$$\bar{\mathcal{L}}_{\mathcal{S}, \mathbf{p}}(c_{h,t}) = \sum_{m=1}^M \sum_{i=1}^{N_m} p_{m,i} \bar{\ell}(c_{h,t}, (\mathbf{z}_m^{(i)}, 0)) = \sum_{m=1}^M \sum_{i=1}^{N_m} p_{m,i} c_{h,t}(\mathbf{z}_m^{(i)}) = \mathbb{E}_{\mathbf{z} \sim \hat{\mathcal{P}}^{(\mathbf{p})}}[c_{h,t}(\mathbf{z})] = \mathbb{P}_{\mathbf{z} \sim \hat{\mathcal{P}}^{(\mathbf{p})}}[\ell(h, \mathbf{z}) > t]. \quad (\text{G.686})$$

For any distribution  $\mathcal{P}$  and any non-negative measurable function  $f$ , it holds [MRT18, Eq. 11.5]:

$$\mathbb{E}_{\mathbf{z} \sim \mathcal{P}}[f(\mathbf{z})] = \int_0^\infty \mathbb{P}_{\mathbf{z} \sim \mathcal{P}}[f(\mathbf{z}) > t] dt. \quad (\text{G.687})$$

In view of identity (G.687) and the fact that the loss function  $\ell$  is bounded by  $B$ , we can write:

$$\mathbb{E}_{\mathcal{S}} \left[ \sup_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{P}^{(\mathbf{p})}}(h) - \mathcal{L}_{\mathcal{S}, \mathbf{p}}(h)| \right] = \mathbb{E}_{\mathcal{S}} \left[ \sup_{h \in \mathcal{H}} \left| \int_0^B \mathbb{P}_{\mathbf{z} \sim \mathcal{P}^{(\mathbf{p})}}[\ell(h, \mathbf{z}) > t] dt - \int_0^B \mathbb{P}_{\mathbf{z} \sim \hat{\mathcal{P}}^{(\mathbf{p})}}[\ell(h, \mathbf{z}) > t] dt \right| \right] \quad (\text{G.688})$$

$$\leq B \cdot \mathbb{E}_{\mathcal{S}} \left[ \sup_{h \in \mathcal{H}, t \in [0, B]} \left| \bar{\mathcal{L}}_{\bar{\mathcal{P}}^{(\mathbf{p})}}(c_{h,t}) - \bar{\mathcal{L}}_{\mathcal{S}, \mathbf{p}}(c_{h,t}) \right| \right] \quad (\text{G.689})$$

$$= B \cdot \mathbb{E}_{\mathcal{S}} \left[ \sup_{h \in \mathcal{H}, t \in \mathbb{R}} \left| \bar{\mathcal{L}}_{\bar{\mathcal{P}}^{(\mathbf{p})}}(c_{h,t}) - \bar{\mathcal{L}}_{\mathcal{S}, \mathbf{p}}(c_{h,t}) \right| \right] \quad (\text{G.690})$$

$$= B \cdot \mathbb{E}_{\mathcal{S}} \left[ \sup_{c_{h,t} \in \bar{\mathcal{H}}_{\Theta}} \left| \bar{\mathcal{L}}_{\bar{\mathcal{P}}^{(\mathbf{p})}}(c_{h,t}) - \bar{\mathcal{L}}_{\mathcal{S}, \mathbf{p}}(c_{h,t}) \right| \right]. \quad (\text{G.691})$$

The right-hand side can be bounded using Theorem G.4 in terms of the VC-dimension of the family of hypothesis  $\bar{\mathcal{H}}$ , which by definition of the pseudo-dimension and of the classifiers  $c_{h,t}$  is precisely  $\text{Pdim}(\ell \circ \mathcal{H})$ . We obtain

$$\mathbb{E}_{\mathcal{S}} \left[ \sup_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{P}^{(\mathbf{p})}}(h) - \mathcal{L}_{\mathcal{S}, \mathbf{p}}(h)| \right] \leq 5B \cdot \sqrt{\frac{\text{Pdim}(\ell \circ \mathcal{H})}{N_{\text{eff}}}} \cdot \sqrt{1 + \log \left( \frac{N}{\text{Pdim}(\ell \circ \mathcal{H})} \right)}. \quad (\text{G.692})$$

□

## G.2 Proof of Lemma 4.3.2

**Lemma 4.3.2.** *With the same notation as in Theorem 4.3.1,  $N_{\text{eff}} \leq N$  and this bound is attained when  $\mathbf{p}$  is uniform.*



*Proof.* We remind that

$$N_{\text{eff}} = \left( \sum_{m=1}^M \sum_{i=1}^{N_m} (p_{m,i})^2 \right)^{-1}. \quad (\text{G.693})$$

Let  $\mathbf{u} \in \Delta^N$  be the vector obtained by concatenating all the values  $p_{m,i}$  for  $m \in [M]$  and  $i \in [N_m]$ . It follows that

$$N_{\text{eff}} = \left( \sum_{n=1}^N u_n^2 \right)^{-1} = \|\mathbf{u}\|_2^{-2}. \quad (\text{G.694})$$

Let  $\mathbf{u}^* \triangleq \mathbf{1}/N$ , it is clear that  $\mathbf{u}^* \in \Delta^N$ , and  $\|\mathbf{u}^*\|_2^2 = 1/N$ . Let  $\mathbf{u} \in \Delta^N$ , using Cauchy-Shwartz inequality, we have

$$1 = \sum_{n=1}^N u_n = \sum_{n=1}^N (u_n \times 1) \leq \sqrt{\sum_{n=1}^N u_n^2} \cdot \sqrt{\sum_{n=1}^N 1} = \|\mathbf{u}\|_2 \cdot \sqrt{N}. \quad (\text{G.695})$$

Thus,  $\|\mathbf{u}\|_2^{-2} \leq N$ , which concludes the proof.  $\square$

### G.2.1 Proof of Theorem 4.3.3

**Theorem 4.3.3.** *Suppose that Assumptions 18–12 hold, the sequence  $(q^{(t)})_t$  is non increasing, and verifies  $q^{(1)} = \mathcal{O}(1/T)$ , and  $\eta \propto 1/\sqrt{T} \cdot \min\{1, 1/\bar{\sigma}(\lambda)\}$ . Under full clients participation ( $\mathcal{S}^{(t)} = [M]$ ) with full batch ( $K \geq |\mathcal{I}_m^{(t)}|$ ), we have*

$$\epsilon_{\text{opt}} \leq \mathcal{O}(\bar{\sigma}(\lambda)) + \mathcal{O}\left(\frac{\bar{\sigma}(\lambda)}{\sqrt{T}}\right) + \mathcal{O}\left(\frac{1}{\sqrt{T}}\right),$$

where,

$$\bar{\sigma}^2(\lambda) \triangleq \sum_{t=1}^T q^{(t)} \times \mathbb{E}_{\mathcal{S}} \left[ \sup_{\theta \in \Theta} \left\| \nabla \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) - \sum_{m=1}^M p_m^{(t)} \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta) \right\|^2 \right].$$

Moreover, there exist a data arrival process and a loss function  $\ell$ , such that, under FIFO memory update rule, for any choice of weights  $\lambda$ ,  $\epsilon_{\text{opt}} = \Omega(\bar{\sigma}(\lambda))$ .

*Proof.* We remind that

$$p_m^{(t)} = \frac{\sum_{j \in \mathcal{I}_m^{(t)}} \lambda_m^{(t,j)}}{\sum_{m'=1}^M \sum_{j \in \mathcal{I}_{m'}^{(t)}} \lambda_{m'}^{(t,j)}}, \quad (\text{G.696})$$

and

$$q^{(t)} = \frac{\sum_{m=1}^M \sum_{j \in \mathcal{I}_m^{(t)}} \lambda_m^{(t,j)}}{\sum_{s=1}^T \sum_{m=1}^M \sum_{j \in \mathcal{I}_m^{(s)}} \lambda_m^{(s,j)}}. \quad (\text{G.697})$$

For ease of notation we introduce the following functions defined on  $\Theta$ ;

$$f_m^{(t)} \triangleq \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}, \quad (\text{G.698})$$

$$F^{(t)} \triangleq \sum_{m=1}^M p_m^{(t)} \cdot \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)} = \sum_{m=1}^M p_m^{(t)} \cdot f_m^{(t)}, \quad (\text{G.699})$$

$$F \triangleq \mathcal{L}_{\mathcal{S}}^{(\lambda)} = \sum_{t=1}^T q^{(t)} \cdot F^{(t)}. \quad (\text{G.700})$$

Note that this notation hides the dependence of the functions  $f_m^{(t)}$ ,  $F^{(t)}$  and  $F$  on the samples  $\mathcal{S}$  and the parameters  $\lambda$ . In this proof we simply use  $\mathbb{E}$  to refer to the expectation of the samples  $\mathcal{S}$ , e.g.,  $\mathbb{E}[\nabla F(\theta)] = \mathbb{E}_{\mathcal{S}}[\nabla \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta)]$ .

We remind that

$$\Delta^{(t)} = \sum_{m=1}^M p_m^{(t)} \cdot \left( \theta_m^{(t,E+1)} - \theta^{(t)} \right) = -\eta \cdot \sum_{e=1}^E \sum_{m=1}^M p_m^{(t)} \cdot \nabla f_m^{(t)} \left( \theta_m^{(t,e)} \right). \quad (\text{G.701})$$

We define  $\tilde{\eta} \triangleq \eta E > 0$  and  $\tilde{\nabla}^{(t)} \triangleq -\frac{\Delta^{(t)}}{\tilde{\eta}} \in \mathbb{R}^d$ . The coefficient  $\tilde{\eta}$  and the vector  $\tilde{\nabla}^{(t)}$  can be seen as the efficient learning rate and the *pseudo-gradient* used at global iteration  $t \in [T]$ , respectively [Wan+21a, Section 2]. With this set of notation, the update rule of Algorithm 13 can be summarized as

$$\tilde{\nabla}^{(t)} = \frac{1}{E} \sum_{e=1}^E \sum_{m=1}^M p_m^{(t)} \cdot \nabla f_m^{(t)} \left( \theta_m^{(t,e)} \right) \quad (\text{G.702})$$

$$\theta^{(t+1)} = \Pi_{\Theta} \left( \theta^{(t)} - \tilde{\eta} \cdot \tilde{\nabla}^{(t)} \right) \quad (\text{G.703})$$

Under Assumptions 24–12, the functions  $f_m^{(t)}$ ,  $F^{(t)}$ , and  $F$  are bounded, convex and  $L$ -smooth as convex combinations of bounded, convex and  $L$ -smooth functions.

Let  $\theta^*$  be a minimizer of  $F$  over  $\Theta$ , and  $F^* \triangleq F(\theta^*)$  (note that  $\theta^*$  and  $F^*$  depend on  $\mathcal{S}$ ). By convexity of  $F$ , we have

$$-\langle \nabla F(\theta), \theta - \theta^* \rangle \leq -(F(\theta) - F^*). \quad (\text{G.704})$$

Lemma G.1 and Jensen inequality imply that

$$\max \left\{ \left\| \nabla f_m^{(t,e)}(\theta) \right\|, \left\| \nabla F^{(t)}(\theta) \right\|, \left\| \nabla F(\theta) \right\|, \left\| \tilde{\nabla}^{(t)} \right\| \right\} \leq G, \quad (\text{G.705})$$

where  $G \triangleq \sqrt{2LB}$ .

For convenience, we quantify the *variance* between the current and global functions' gradients with

$$\sigma_t = \sup_{\theta \in \Theta} \left\| \nabla F(\theta) - \nabla F^{(t)}(\theta) \right\|. \quad (\text{G.706})$$

We define  $\sigma^2(\lambda) \triangleq \sum_{t=1}^T q^{(t)} \sigma_t^2$ . Therefore,  $\bar{\sigma}^2(\lambda) = \mathbb{E}[\sigma^2(\lambda)]$ .

The idea of the proof is to bound the distance between the pseudo-gradient  $\tilde{\nabla}^{(t)}$  and the correct gradient,  $\nabla F(\theta^{(t)})$ , that should have been used at iteration  $t > 0$ . One can write

$$\mathbb{E} \left[ \left\| \theta^{(t+1)} - \theta^* \right\|^2 \right] = \mathbb{E} \left[ \left\| \Pi_{\Theta} \left( \theta^{(t)} - \tilde{\eta} \tilde{\nabla} \right) - \theta^* \right\|^2 \right] \quad (\text{G.707})$$

$$\leq \mathbb{E} \left[ \left\| \theta^{(t)} - \tilde{\eta} \tilde{\nabla} - \theta^* \right\|^2 \right] \quad (\text{G.708})$$

$$= \mathbb{E} \left[ \left\| \theta^{(t)} - \tilde{\eta} \nabla F(\theta^{(t)}) - \theta^* + \tilde{\eta} \left( \nabla F(\theta^{(t)}) - \tilde{\nabla}^{(t)} \right) \right\|^2 \right] \quad (\text{G.709})$$

$$\begin{aligned} &= \mathbb{E} \left[ \underbrace{\left\| \theta^{(t)} - \tilde{\eta} \nabla F(\theta^{(t)}) - \theta^* \right\|^2}_{\triangleq T_1} + \tilde{\eta}^2 \mathbb{E} \left[ \underbrace{\left\| \nabla F(\theta^{(t)}) - \tilde{\nabla}^{(t)} \right\|^2}_{\triangleq T_2} \right] \right. \\ &\quad \left. + 2\tilde{\eta} \mathbb{E} \left[ \underbrace{\left\langle \nabla F(\theta^{(t)}) - \tilde{\nabla}^{(t)}, \theta^{(t)} - \tilde{\eta} \nabla F(\theta^{(t)}) - \theta^* \right\rangle}_{\triangleq T_3} \right] \right]. \quad (\text{G.710}) \end{aligned}$$

**Bound  $T_1$ .** We have,

$$T_1 = \left\| \theta^{(t)} - \tilde{\eta} \nabla F(\theta^{(t)}) - \theta^* \right\|^2 \quad (\text{G.711})$$

$$= \left\| \theta^{(t)} - \theta^* \right\|^2 + \tilde{\eta}^2 \left\| \nabla F(\theta^{(t)}) \right\|^2 - 2\tilde{\eta} \cdot \left\langle \nabla F(\theta^{(t)}), \theta^{(t)} - \theta^* \right\rangle \quad (\text{G.712})$$

$$\leq \left\| \theta^{(t)} - \theta^* \right\|^2 + \tilde{\eta}^2 G^2 - 2\tilde{\eta} \left( F(\theta^{(t)}) - F^* \right), \quad (\text{G.713})$$

where we used (G.704) and (G.705) to obtain the last inequality.

**Bound  $T_2$ .** Let  $\alpha > 0$ , we have,

$$T_2 = \left\| \nabla F(\theta^{(t)}) - \tilde{\nabla}^{(t)} \right\|^2 \quad (\text{G.714})$$

$$= \left\| \nabla F(\theta^{(t)}) - \sum_{m=1}^M p_m^{(t)} \nabla f_m^{(t)}(\theta^{(t)}) + \sum_{m=1}^M p_m^{(t)} \nabla f_m^{(t)}(\theta^{(t)}) - \tilde{\nabla}^{(t)} \right\|^2 \quad (\text{G.715})$$

$$\leq (1 + \alpha) \left\| \nabla F(\theta^{(t)}) - \nabla F^{(t)}(\theta^{(t)}) \right\|^2 + (1 + \alpha^{-1}) \left\| \sum_{m=1}^M p_m^{(t)} \nabla f_m^{(t)}(\theta^{(t)}) - \tilde{\nabla}^{(t)} \right\|^2, \quad (\text{G.716})$$

where we used the fact that for any two vectors  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$  and a coefficient  $\alpha > 0$ , it holds that  $\|\mathbf{a} + \mathbf{b}\|^2 \leq (1 + \alpha) \|\mathbf{a}\|^2 + (1 + \alpha^{-1}) \|\mathbf{b}\|^2$ , with the particular choice  $\mathbf{a} = \nabla F(\theta^{(t)}) - \nabla F^{(t)}(\theta^{(t)})$ , and  $\mathbf{b} = \sum_{m=1}^M p_m^{(t)} \nabla f_m^{(t)}(\theta^{(t)}) - \tilde{\nabla}^{(t)}$ .

We remind that,

$$\tilde{\nabla} = -\frac{\Delta^{(t)}}{\eta E} = \sum_{e=1}^E \sum_{m=1}^M \frac{p_m^{(t)}}{E} \mathbf{g}_m^{(t,e)} = \sum_{e=1}^E \sum_{m=1}^M \frac{p_m^{(t)}}{E} \nabla f_m^{(t)}(\theta_m^{(t,e)}). \quad (\text{G.717})$$

Thus,

$$\left\| \sum_{m=1}^M p_m^{(t)} \nabla f_m^{(t)}(\theta^{(t)}) - \tilde{\nabla}^{(t)} \right\|^2 = \left\| \sum_{e=1}^E \sum_{m=1}^M \frac{p_m^{(t)}}{E} (\nabla f_m^{(t)}(\theta^{(t)}) - \nabla f_m^{(t)}(\theta^{(t,e)})) \right\|^2 \quad (\text{G.718})$$

$$\leq \sum_{e=1}^E \sum_{m=1}^M \frac{p_m^{(t)}}{E} \left\| \nabla f_m^{(t)}(\theta^{(t)}) - \nabla f_m^{(t)}(\theta_m^{(t,e)}) \right\|^2 \quad (\text{G.719})$$

$$= \sum_{e=1}^E \sum_{m=1}^M \frac{p_m^{(t)}}{E} \left\| \nabla f_m^{(t)}(\theta_m^{(t,1)}) - \nabla f_m^{(t)}(\theta_m^{(t,e)}) \right\|^2 \quad (\text{G.720})$$

$$\leq L^2 \sum_{e=1}^E \sum_{m=1}^M \frac{p_m^{(t)}}{E} \left\| \theta_m^{(t,1)} - \theta_m^{(t,e)} \right\|^2 \quad (\text{G.721})$$

$$= L^2 \sum_{e=1}^E \sum_{m=1}^M \frac{p_m^{(t)}}{E} \left\| \sum_{e'=1}^{e-1} \theta_m^{(t,e')} - \theta_m^{(t,e'+1)} \right\|^2 \quad (\text{G.722})$$

$$= \frac{\tilde{\eta}^2 L^2}{E^3} \sum_{m=1}^M p_m^{(t)} \sum_{e=1}^E \left\| \sum_{e'=1}^{e-1} \nabla f_m^{(t)}(\theta_m^{(t,e')}) \right\|^2 \quad (\text{G.723})$$

$$\leq \frac{\tilde{\eta}^2 L^2}{E^3} \sum_{m=1}^M p_m^{(t)} \sum_{e=1}^E (e-1) \sum_{e'=1}^{e-1} \left\| \nabla f_m^{(t)}(\theta_m^{(t,e')}) \right\|^2 \quad (\text{G.724})$$

$$\leq \frac{\tilde{\eta}^2 L^2 G^2}{E^3} \sum_{e=1}^E (e-1)^2 \quad (\text{G.725})$$

$$\leq 2\tilde{\eta}^2 L^2 G^2 (1 - E^{-1}), \quad (\text{G.726})$$

where we used Jensen inequality to obtain (G.719) and (G.724), the  $L$ -smoothness of  $f_m^{(t)}$  to obtain (G.721), and (G.705) to obtain (G.725). Replacing (G.726) in (G.716) and using  $\sigma_t$  defined in (G.706), we have

$$T_2 \leq (1 + \alpha) \sigma_t^2 + 2(1 + \alpha^{-1}) \tilde{\eta}^2 L^2 G^2 (1 - E^{-1}). \quad (\text{G.727})$$

With the particular choice  $\alpha = \frac{\tilde{\eta} L G}{\sigma_t} \cdot \sqrt{2(1 - E^{-1})}$ , it follows that

$$T_2 \leq \left( \sigma_t + \tilde{\eta} L G \sqrt{2(1 - E^{-1})} \right)^2 \leq 2\sigma_t^2 + 4\tilde{\eta}^2 L^2 G^2 (1 - E^{-1}) \quad (\text{G.728})$$

Our bound ((G.728)) shows that, as expected, the term  $T_2$ , measuring the deviation between the true gradient  $\nabla F(\theta^{(t)})$  and the pseudo-gradient  $\tilde{\nabla}^{(t)}$ , is equal to zero when  $E = 1$  and  $\sigma_t = 0$ . This scenario corresponds exactly to the centralized version of gradient descent.

**Bound  $T_3$ .** We have

$$T_3 = \left\langle \nabla F(\theta^{(t)}) - \tilde{\nabla}^{(t)}, \theta^{(t)} - \tilde{\eta} \nabla F(\theta^{(t)}) - \theta^* \right\rangle \quad (\text{G.729})$$

$$\begin{aligned} &= \left\langle \nabla F(\theta^{(t)}) - \nabla F^{(t)}(\theta^{(t)}), \theta^{(t)} - \theta^* \right\rangle + \left\langle \nabla F^{(t)}(\theta^{(t)}) - \tilde{\nabla}^{(t)}, \theta^{(t)} - \theta^* \right\rangle \\ &\quad - \tilde{\eta} \left\langle \nabla F(\theta^{(t)}) - \tilde{\nabla}^{(t)}, \nabla F(\theta^{(t)}) \right\rangle. \end{aligned} \quad (\text{G.730})$$

We remind that  $\Theta$  is bounded and that  $D$  is its diameter. Using Cauchy-Schwarz inequality, we have

$$\langle \nabla F^{(t)}(\theta^{(t)}) - \tilde{\nabla}^{(t)}, \theta^{(t)} - \theta^* \rangle \leq \left\| \nabla F^{(t)}(\theta^{(t)}) - \tilde{\nabla}^{(t)} \right\| \cdot \left\| \theta^{(t)} - \theta^* \right\| \quad (\text{G.731})$$

$$= \left\| \sum_{m=1}^M p_m^{(t)} \nabla f_m^{(t)}(\theta^{(t)}) - \tilde{\nabla}^{(t)} \right\| \cdot \left\| \theta^{(t)} - \theta^* \right\| \quad (\text{G.732})$$

$$\leq \tilde{\eta} L D G \sqrt{2(1-E^{-1})}, \quad (\text{G.733})$$

where we used (G.726) to obtain the last inequality. Using Cauchy-Schwartz inequality again and the fact that gradients are bounded ((G.705)), we have

$$-\tilde{\eta} \langle \nabla F(\theta^{(t)}) - \tilde{\nabla}^{(t)}, \nabla F(\theta^{(t)}) \rangle \leq \tilde{\eta} \left\| \nabla F(\theta^{(t)}) - \tilde{\nabla}^{(t)} \right\| \cdot \left\| \nabla F(\theta^{(t)}) \right\| \leq 2\tilde{\eta} \cdot G^2. \quad (\text{G.734})$$

Finally using Cauchy-Schwartz inequality and the boundedness of  $\Theta$ , we have

$$\langle \nabla F(\theta^{(t)}) - \nabla F^{(t)}(\theta^{(t)}), \theta^{(t)} - \theta^* \rangle \leq \sigma^{(t)} \cdot D. \quad (\text{G.735})$$

Replacing (G.733), (G.734), and (G.735) in (G.730), we have

$$T_3 \leq \sigma^{(t)} \cdot D + \tilde{\eta} G \left( 2G + LD \sqrt{2(1-E^{-1})} \right) \quad (\text{G.736})$$

**Bound  $\epsilon_{\text{opt}}$ .** Replacing (G.713), (G.728), and (G.736) in (G.710), we have

$$\begin{aligned} \mathbb{E} \left[ \left\| \theta^{(t+1)} - \theta^* \right\|^2 \right] &= \mathbb{E} \left[ \left\| \theta^{(t)} - \theta^* \right\|^2 \right] - 2\tilde{\eta} \cdot \mathbb{E} \left[ F(\theta^{(t)}) - F^* \right] + 2\tilde{\eta} \cdot \bar{\sigma}^{(t)} D \\ &\quad + \tilde{\eta}^2 \cdot \left( 2\bar{\sigma}_t^2 + G \left( 5G + 2LD \sqrt{2(1-E^{-1})} \right) \right) + 4\tilde{\eta}^4 \cdot L^2 G^2 (1-E^{-1}), \end{aligned} \quad (\text{G.737})$$

where  $\bar{\sigma}_t^2 = \mathbb{E}[\sigma_t^2] = \mathbb{E} \left[ \sup_{\theta \in \Theta} \left\| \nabla F(\theta) - \nabla F^{(t)}(\theta) \right\|^2 \right]$ .

The sequence  $(q^{(t)})_t$  is non increasing, i.e., for  $t \in [T]$   $q^{(t+1)} \leq q^{(t)}$ . It follows from (G.737) that, for  $t > 0$ , we have

$$q^{(t+1)} \mathbb{E} \left[ \left\| \theta^{(t+1)} - \theta^* \right\|^2 \right] \leq q^{(t)} \mathbb{E} \left[ \left\| \theta^{(t+1)} - \theta^* \right\|^2 \right] \quad (\text{G.738})$$

$$\begin{aligned} &\leq q^{(t)} \mathbb{E} \left[ \left\| \theta^{(t)} - \theta^* \right\|^2 \right] - 2\tilde{\eta} q^{(t)} \mathbb{E} \left[ F(\theta^{(t)}) - F^* \right] + 2\tilde{\eta} \cdot q^{(t)} \bar{\sigma}^{(t)} D \\ &\quad + 2\tilde{\eta}^2 \cdot q^{(t)} \bar{\sigma}_t^2 + 2\tilde{\eta}^2 q^{(t)} \cdot C_1 + 2\tilde{\eta}^4 q^{(t)} \cdot C_2, \end{aligned} \quad (\text{G.739})$$

where  $C_1 = G \left( \frac{5}{2}G + LD \sqrt{2(1-E^{-1})} \right)$ , and  $C_2 = 2L^2 G^2 (1-E^{-1})$ . Rearranging the terms and summing over  $t \in \{1, \dots, T\}$ , we have

$$\sum_{t=1}^T q^{(t)} \mathbb{E} \left[ F(\theta^{(t)}) - F^* \right] \leq \left( \sum_{t=1}^T q^{(t)} \bar{\sigma}_t \right) \cdot D + T q^{(1)} \cdot \frac{D^2}{2\tilde{\eta}T} + \tilde{\eta} \cdot \left( \sum_{t=1}^T q^{(t)} \bar{\sigma}_t^2 \right) + \tilde{\eta} \cdot (C_1 + \tilde{\eta}^2 C_2) \quad (\text{G.740})$$

We remind that  $\bar{\sigma}^2(\lambda) = \sum_{t=1}^T q^{(t)} \bar{\sigma}_t^2$ . Using the concavity of the function  $\sqrt{\cdot}$ , it follows that  $\bar{\sigma}(\lambda) \geq \sum_{t=1}^T q^{(t)} \bar{\sigma}_t$ . It follows that

$$\mathbb{E} \left[ F(\bar{\theta}^{(t)}) - F^* \right] \leq \bar{\sigma}(\lambda) \cdot D + Tq^{(1)} \cdot \frac{D^2}{2\tilde{\eta}T} + \tilde{\eta} \cdot \bar{\sigma}^2(\lambda) + \tilde{\eta}C_1 + \tilde{\eta}^3C_2. \quad (\text{G.741})$$

The final results is obtained by using  $\mathcal{O}(Tq^{(1)}) = 1$ . We have

$$\mathbb{E} \left[ F(\bar{\theta}^{(t)}) - F^* \right] \leq \bar{\sigma}(\lambda) \cdot D + \frac{\bar{\sigma}(\lambda)}{\sqrt{T}} + \frac{C_1 + C_3}{\sqrt{T}} + \frac{C_2}{\sqrt{T^3}}, \quad (\text{G.742})$$

where  $C_3$  is a constant proportional to  $D^2$ .

**Lower Bound.** In the rest of this proof, we use  $\theta$  to denote the model parameters, and  $\theta_1$ , and  $\theta_2$  its components.

We artificially construct a simple problem and a particular arrival process, such that the output of Algorithm 13, with  $M = 1$ ,  $C_1 = 1$ , FIFO update rule, and  $\eta = \Omega(1/\sqrt{T})$ , verifies  $\lim_{T \rightarrow \infty} F(\bar{\theta}^{(T)}) - F^* \geq c \cdot \bar{\sigma}^2(\lambda)$ , where  $c > 0$  is a constant. We consider a setting with  $\Theta = [-1, 1]^2$ ,  $\mathcal{Z} = \{1, 2\}$ , and a loss function defined for  $\theta \in \Theta$  with

$$\ell(\theta; 1) \triangleq (\theta_1 + 1)^2 + \frac{1}{2}(\theta_1 + \theta_2 + 1)^2, \quad (\text{G.743})$$

and

$$\ell(\theta; 2) \triangleq \frac{1}{2}(\theta_1 - 1)^2 + \frac{1}{2}(\theta_1 + \theta_2 - 1)^2. \quad (\text{G.744})$$

We observe that the minimizer of  $\ell(\cdot; 1)$  (resp.  $\ell(\cdot; 2)$ ) is  $\theta_1^* = (-1, 0)$  (resp.  $\theta_2^* = (1, 0)$ ).

For time horizon  $T$ , we consider the arrival process, where one sample, say  $\mathbf{z}_1$ , is drawn uniformly at random from  $\mathcal{Z}$  at time step  $t_1 = 1$ , and a second sample,  $\mathbf{z}_2$ , is drawn uniformly at random from  $\mathcal{Z}$  a time step  $t_2 = T/2$ . We define  $q \triangleq \sum_{t=1}^{T/2} q^{(t)}$ . Since  $(q^{(t)})_{t \geq 1}$  is non increasing, then  $q \geq 1/2$ . We remark that, in this setting, the trajectory of Algorithm 13 is only determined by the values of  $\mathbf{z}_1$  and  $\mathbf{z}_2$ , i.e., the values taken by the sequence  $(\theta^{(t)})_{t \geq 1}$  are only determined by the values of  $\mathbf{z}_1$  and  $\mathbf{z}_2$ .

We have

$$\epsilon_{\text{opt}} = \mathbb{E}_{\mathcal{S}} \left[ \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\bar{\theta}^{(T)}) - \min_{\theta \in \Theta} \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) \right] \quad (\text{G.745})$$

$$= \frac{1}{2} \mathbb{E}_{\mathcal{S}} \left[ \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\bar{\theta}^{(T)}) - \min_{\theta \in \Theta} \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) \mid \mathcal{S} = \{1, 2\} \right] + \frac{1}{4} \mathbb{E}_{\mathcal{S}} \left[ \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\bar{\theta}^{(T)}) - \min_{\theta \in \Theta} \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) \mid \mathcal{S} = \{1\} \right] \quad (\text{G.746})$$

$$+ \frac{1}{4} \mathbb{E}_{\mathcal{S}} \left[ \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\bar{\theta}^{(T)}) - \min_{\theta \in \Theta} \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) \mid \mathcal{S} = \{2\} \right] \quad (\text{G.747})$$

$$\geq \frac{1}{2} \mathbb{E}_{\mathcal{S}} \left[ \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\bar{\theta}^{(T)}) - \min_{\theta \in \Theta} \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) \mid \mathcal{S} = \{1, 2\} \right], \quad (\text{G.748})$$

and

$$\bar{\sigma}^2(\lambda) = q(1-q) \mathbb{E}_{\mathcal{S}} \left[ \max_{\theta \in \Theta} \|\nabla \ell(\theta; \mathbf{z}_1) - \nabla \ell(\theta; \mathbf{z}_2)\|^2 \right] \quad (\text{G.749})$$

$$\leq \frac{q(1-q)}{2} \cdot \max_{\theta \in \Theta} \|\nabla \ell(\theta; 1) - \nabla \ell(\theta; 2)\|^2 \quad (\text{G.750})$$

$$\leq 20 \cdot q(1-q). \quad (\text{G.751})$$

We consider the case when  $\mathbf{z}_1 = 1$ , and  $\mathbf{z}_2 = 2$ . Thus

$$\mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) = q \cdot \ell(\theta; 1) + (1-q) \cdot \ell(\theta; 2). \quad (\text{G.752})$$

Let  $\theta^*$  be a minimizer of  $\mathcal{L}_{\mathcal{S}}^{(\lambda)}$ , then

$$\theta_1^* = \frac{1-3q}{1+q} \quad \text{and} \quad \theta_2^* = 1 - 2q - \frac{1-3q}{1+q}. \quad (\text{G.753})$$

Moreover, one can prove that

$$\min_{\theta \in [-1,1]} \mathcal{L}_{\mathcal{S}}^{(\lambda)}((\theta, 0)) - \min_{\theta \in \Theta} \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) \geq 6 \cdot q(1-q) \quad (\text{G.754})$$

For  $\epsilon > 0$ , it exists  $E \geq 1$ , and  $T_0 \geq 1$ , such that for any  $T \geq T_0$ , we have  $|\bar{\theta}_2^{(T)}| \leq \epsilon$ . Therefore,

$$\mathcal{L}_{\mathcal{S}}^{(\lambda)}(\bar{\theta}^{(T)}) - \min_{\theta \in \Theta} \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) \sim_{\epsilon \rightarrow 0} \mathcal{L}_{\mathcal{S}}^{(\lambda)}((\theta_1^{(T)}, 0)) - \min_{\theta \in \Theta} \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) \quad (\text{G.755})$$

$$\geq \min_{\theta \in [-1,1]} \mathcal{L}_{\mathcal{S}}^{(\lambda)}((\theta, 0)) - \min_{\theta \in \Theta} \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) \quad (\text{G.756})$$

$$\geq 6 \cdot q(1-q) \quad (\text{G.757})$$

$$= \frac{3}{10} \bar{\sigma}^2(\lambda) \quad (\text{G.758})$$

The same holds when  $\mathbf{z}_1 = 2$ , and  $\mathbf{z}_2 = 1$ . It follows that

$$\epsilon_{\text{opt}} \geq \frac{3}{20} \bar{\sigma}^2(\lambda). \quad (\text{G.759})$$

□

### G.3 Bound $\bar{\sigma}^2(\lambda)$

We remind, from Remark 6, that

$$\sigma_0^2 \triangleq \max_m \mathbb{E}_{\mathbf{z} \sim \mathcal{P}_m} \left[ \sup_{\theta \in \Theta} \|\nabla \ell(\theta; \mathbf{z}) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta)\|^2 \right], \quad (\text{G.760})$$

and

$$\zeta \triangleq \max_{m, m'} \sup_{\theta \in \Theta} \left\| \nabla \mathcal{L}_{\mathcal{P}_{m'}}(\theta) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) \right\|. \quad (\text{G.761})$$

**Lemma G.5.** For any memory update rule and any choice of memory parameters  $\lambda$  we have

$$\bar{\sigma}^2(\lambda) = \mathcal{O}\left(\sigma_0^2 + \zeta^2 \cdot \sum_{t=1}^T q^{(t)} \sum_{m=1}^M (p_m - p_m^{(t)})^2\right). \quad (\text{G.762})$$

*Proof.* We remind that

$$\bar{\sigma}^2(\lambda) = \sum_{t=1}^T q^{(t)} \mathbb{E}_{\mathcal{S}} \left[ \sup_{\theta \in \Theta} \left\| \nabla \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) - \sum_{m=1}^M p_m^{(t)} \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta) \right\|^2 \right], \quad (\text{G.763})$$

and, for  $m \in [M]$ , we define

$$\mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\cdot) \triangleq \frac{\sum_{t=1}^T \sum_{j \in \mathcal{I}_m^{(t)}} \lambda_m^{(t,j)} \ell(\cdot, \mathbf{z}_m^{(j)})}{\sum_{s=1}^T \sum_{i \in \mathcal{I}_m^{(s)}} \lambda_m^{(s,i)}}, \quad (\text{G.764})$$

and we remind (see Theorem 4.3.1) that

$$p_m = \frac{\sum_{t=1}^T \sum_{j \in \mathcal{I}_m^{(t)}} \lambda_m^{(t,j)}}{\sum_{m'=1}^M \sum_{s=1}^T \sum_{i \in \mathcal{I}_m^{(s)}} \lambda_m^{(s,i)}}. \quad (\text{G.765})$$

$\mathcal{L}_{\mathcal{S}_m}^{(\lambda)}$  and  $p_m$  represent client  $m$ 's weighted empirical risk of client  $m$  and its relative importance, respectively. We remark that

$$\mathcal{L}_{\mathcal{S}}^{(\lambda)} = \sum_{m=1}^M p_m \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}, \quad (\text{G.766})$$

and

$$p_m = \sum_{t=1}^T q^{(t)} p_m^{(t)}. \quad (\text{G.767})$$

For  $t \in [T]$  and  $\theta \in \Theta$ , we have

$$\begin{aligned} & \left\| \nabla \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) - \sum_{m=1}^M p_m^{(t)} \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta) \right\|^2 \\ &= \left\| \nabla \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) - \sum_{m=1}^M p_m^{(t)} \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) + \sum_{m=1}^M p_m^{(t)} \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) - \sum_{m=1}^M p_m^{(t)} \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta) \right\|^2 \end{aligned} \quad (\text{G.768})$$

$$\leq 2 \left\| \nabla \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) - \sum_{m=1}^M p_m^{(t)} \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) \right\|^2 + 2 \left\| \sum_{m=1}^M p_m^{(t)} \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) - \sum_{m=1}^M p_m^{(t)} \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta) \right\|^2 \quad (\text{G.769})$$

$$= 2 \underbrace{\left\| \sum_{m=1}^M p_m^{(t)} \left( \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) - \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta) \right) \right\|^2}_{\triangleq T_1} + 2 \underbrace{\left\| \sum_{m=1}^M (p_m - p_m^{(t)}) \cdot \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) \right\|^2}_{\triangleq T_2}. \quad (\text{G.770})$$



**Bound  $T_1$ .** We have

$$T_1 = \left\| \sum_{m=1}^M p_m^{(t)} \left( \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) - \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta) \right) \right\|^2 \quad (\text{G.771})$$

$$\leq \sum_{m=1}^M p_m^{(t)} \left\| \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) - \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta) \right\|^2 \quad (\text{G.772})$$

$$= \sum_{m=1}^M p_m^{(t)} \left\| \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) + \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) - \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta) \right\|^2 \quad (\text{G.773})$$

$$\leq 2 \sum_{m=1}^M p_m^{(t)} \left\| \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) \right\|^2 + 2 \sum_{m=1}^M p_m^{(t)} \left\| \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) - \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta) \right\|^2. \quad (\text{G.774})$$

**Bound  $T_2$ .** For  $m' \in [m]$ , we have

$$T_2 = \left\| \sum_{m=1}^M (p_m - p_m^{(t)}) \cdot \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) \right\|^2 \quad (\text{G.775})$$

$$= \left\| \sum_{m=1}^M (p_m - p_m^{(t)}) \cdot \left( \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) - \nabla \mathcal{L}_{\mathcal{S}_{m'}}^{(\lambda)}(\theta) \right) \right\|^2 \quad (\text{G.776})$$

$$\leq \sum_{m=1}^M (p_m - p_m^{(t)})^2 \cdot \sum_{m=1}^M \left\| \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) - \nabla \mathcal{L}_{\mathcal{S}_{m'}}^{(\lambda)}(\theta) \right\|^2 \quad (\text{G.777})$$

$$= \sum_{m=1}^M (p_m - p_m^{(t)})^2 \cdot \sum_{m=1}^M \left\| \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) + \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) - \nabla \mathcal{L}_{\mathcal{P}_{m'}}(\theta) + \nabla \mathcal{L}_{\mathcal{P}_{m'}}(\theta) - \nabla \mathcal{L}_{\mathcal{S}_{m'}}^{(\lambda)}(\theta) \right\|^2 \quad (\text{G.778})$$

$$\leq 3 \sum_{m=1}^M (p_m - p_m^{(t)})^2 \cdot \left( \sum_{m=1}^M \left\| \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) \right\|^2 + \left\| \nabla \mathcal{L}_{\mathcal{S}_{m'}}^{(\lambda)}(\theta) - \nabla \mathcal{L}_{\mathcal{P}_{m'}}(\theta) \right\|^2 \right) + 3 \sum_{m=1}^M (p_m - p_m^{(t)})^2 \cdot \sum_{m=1}^M \left\| \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) - \nabla \mathcal{L}_{\mathcal{P}_{m'}}(\theta) \right\|^2. \quad (\text{G.779})$$

$$\leq 3 \sum_{m=1}^M (p_m - p_m^{(t)})^2 \cdot \left( \sum_{m=1}^M \left\| \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) \right\|^2 + \left\| \nabla \mathcal{L}_{\mathcal{S}_{m'}}^{(\lambda)}(\theta) - \nabla \mathcal{L}_{\mathcal{P}_{m'}}(\theta) \right\|^2 \right) + 3M\zeta^2 \sum_{m=1}^M (p_m - p_m^{(t)})^2. \quad (\text{G.780})$$

We observe that

$$\nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) = \sum_{i=1}^{N_m} \tilde{p}_{m,i} \nabla \ell(\theta; \mathbf{z}_m^{(i)}), \quad (\text{G.781})$$

where, for  $i \in N_m$ ,

$$\tilde{p}_{m,i} = \frac{\sum_{t=1}^T \sum_{j \in \mathcal{I}_m} \mathbf{1}\{j = i\} \cdot \lambda_m^{(t,j)}}{\sum_{t=1}^T \sum_{j \in \mathcal{I}_m^{(t)}} \lambda_m^{(t,j)}}. \quad (\text{G.782})$$

Thus,

$$\mathbb{E}_{\mathcal{S}} \left[ \left\| \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) \right\|^2 \right] = \mathbb{E}_{\mathcal{S}_m} \left[ \left\| \nabla \mathcal{L}_{\mathcal{S}_m}^{(\lambda)}(\theta) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) \right\|^2 \right] \quad (\text{G.783})$$

$$= \mathbb{E}_{\mathcal{S}_m} \left[ \left\| \sum_{i=1}^{N_m} \tilde{p}_{m,i} \nabla \ell(\theta; \mathbf{z}_m^{(i)}) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) \right\|^2 \right] \quad (\text{G.784})$$

$$= \mathbb{E}_{\mathcal{S}_m} \left[ \left\| \sum_{i=1}^{N_m} \tilde{p}_{m,i} \left( \nabla \ell(\theta; \mathbf{z}_m^{(i)}) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) \right) \right\|^2 \right] \quad (\text{G.785})$$

$$\leq \sum_{i=1}^{N_m} \tilde{p}_{m,i} \mathbb{E}_{\mathcal{S}_m} \left[ \left\| \nabla \ell(\theta; \mathbf{z}_m^{(i)}) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) \right\|^2 \right] \quad (\text{G.786})$$

$$= \sum_{i=1}^{N_m} \tilde{p}_{m,i} \mathbb{E}_{\mathbf{z}_m^{(i)}} \left[ \left\| \nabla \ell(\theta; \mathbf{z}_m^{(i)}) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) \right\|^2 \right] \quad (\text{G.787})$$

$$\leq \sum_{i=1}^{N_m} \tilde{p}_{m,i} \sigma_0^2 \quad (\text{G.788})$$

$$= \sigma_0^2. \quad (\text{G.789})$$

In the same way we prove that

$$\mathbb{E}_{\mathcal{S}} \left\| \nabla \mathcal{L}_{\mathcal{P}_m}(\theta) - \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta) \right\|^2 \leq \sigma_0^2. \quad (\text{G.790})$$

We conclude by combining (G.770), (G.774), (G.780), (G.789), and (G.790).  $\square$

### G.3.1 Proof of Theorem 4.3.4

**Theorem 4.3.4.** *Under the same assumptions as in Theorem 4.3.1 and Theorem 4.3.3,*

$$\epsilon_{true} \leq \mathcal{O} \left( \frac{1}{\sqrt{T}} \right) + \mathcal{O}(\bar{\sigma}(\lambda)) + 2 \text{disc}_{\mathcal{H}}(\mathcal{P}^{(\alpha)}, \mathcal{P}^{(\mathfrak{P})}) + \tilde{\mathcal{O}} \left( \sqrt{\frac{\text{Pdim}(\ell \circ \mathcal{H})}{N_{\text{eff}}}} \right).$$

*Proof.* This result is an immediate implication of Theorem 4.3.1 and Theorem 4.3.3 using (4.9).  $\square$

## G.4 Case Study

### G.4.1 Intermittent Client Availability

In Section 4.3.3, we considered the scenario with two groups of clients:  $M_{\text{hist}}$  clients with “historical” datasets, which do not change during training, and  $M - M_{\text{hist}}$  clients, who collect “fresh” samples with constant rates  $\{b_m > 0, m \in \llbracket M_{\text{hist}} + 1, M \rrbracket\}$  and only store the most recent  $b_m$  samples due to memory constraints (i.e.,  $C_m = b_m$ ). Fresh clients can also capture the setting where clients are available during a single communication round: we would then have  $M_{\text{hist}}$  “permanent” clients, which are always available and do not change during training, and  $M - M_{\text{hist}}$  “intermittent” clients, each of them available during one or a few consecutive communication rounds.

In the settings of Section 4.3.3, every client assigns the same weight to all the samples present in its memory independently from the time; let  $\lambda_m$  be the weight assigned by client  $m \in [M]$  to the samples currently present in its memory, i.e.,  $\lambda_m^{(t,j)} = \lambda_m$  for every  $t \in [T]$  and  $j \in \mathcal{I}_m^{(t)}$ .

We remind that the total number of samples collected by client  $m \in [M]$  is  $N_m$ . For a fresh client, say it  $m > M_{\text{hist}}$ ,  $N_m = b_m T$ .

#### G.4.2 General Case

**Corollary 4.3.5'.** *Consider the scenario with  $M_{\text{hist}}$  historical clients, and  $M - M_{\text{hist}}$  fresh clients. Suppose that the same assumption of Theorem 4.3.4 hold, and that Algorithm 13 is used with clients' aggregation weights  $\mathbf{p} = (p_m)_{m \in [M]} \in \Delta^{M-1}$ , then*

$$\epsilon_{\text{true}} \leq \frac{(C_1 + C_3)}{\sqrt{T}} + \frac{C_2}{\sqrt{T^3}} + \left(D + \frac{2}{\sqrt{T}}\right) \sigma_0 \sqrt{M - M_{\text{hist}}} \sqrt{\sum_{m=M_{\text{hist}}+1}^M p_m^2} \quad (\text{G.791})$$

$$+ 2 \cdot \max_{m,m'} \text{disc}(\mathcal{P}_m, \mathcal{P}_{m'}) \cdot \|\alpha - \mathbf{p}\|_1$$

$$+ 10B \cdot \sqrt{1 + \log\left(\frac{N}{\text{Pdim}(\ell \circ \mathcal{H})}\right)} \cdot \sqrt{\frac{\text{Pdim}(\ell \circ \mathcal{H})}{N}} \cdot \sqrt{\sum_{m=1}^M \frac{p_m^2}{n_m}}, \quad (\text{G.792})$$

where  $C_1, C_2$  and  $C_3$  are constants defined in the proof of Theorem 4.3.3, and  $\sigma_0$  is defined in Remark 6.

*Proof.* We remind that

$$p_{m,i} = \frac{\sum_{t=1}^T \sum_{j \in \mathcal{I}_m^{(t)}} \mathbb{1}\{j = i\} \cdot \lambda_m^{(t,j)}}{\sum_{m'=1}^M \sum_{t=1}^T \sum_{j \in \mathcal{I}_{m'}^{(t)}} \lambda_{m'}^{(t,j)}}, \quad i \in N_m^{(T)}, \quad (\text{G.793})$$

and

$$p_m^{(t)} = \frac{\sum_{j \in \mathcal{I}_m^{(t)}} \lambda_m^{(t,j)}}{\sum_{m'=1}^M \sum_{j \in \mathcal{I}_{m'}^{(t)}} \lambda_{m'}^{(t,j)}}, \quad t \in [T]. \quad (\text{G.794})$$

Replacing  $\lambda_m^{(t,j)} = \lambda_m$ , we have

$$p_{m,i} = \frac{\lambda_m \cdot \sum_{t=1}^T \sum_{j \in \mathcal{I}_m^{(t)}} \mathbb{1}\{j = i\}}{\sum_{m'=1}^M \lambda_{m'} \sum_{t=1}^T |\mathcal{I}_{m'}^{(t)}|}, \quad (\text{G.795})$$

and,

$$p_m^{(t)} = \frac{\lambda_m |\mathcal{I}_m^{(t)}|}{\sum_{m'=1}^M \lambda_{m'} |\mathcal{I}_{m'}^{(t)}|}. \quad (\text{G.796})$$

In the settings of Corollary 4.3.5', we have

$$\mathcal{I}_m^{(t)} = \begin{cases} \{1, \dots, N_m\} & , \quad m \in \{1, \dots, M_{\text{hist}}\} \\ \{(t-1) \cdot b_m + 1, \dots, t \cdot b_m - 1\} & , \quad m \in \{M_{\text{hist}} + 1, \dots, M\}. \end{cases} \quad (\text{G.797})$$

Thus,

$$p_m^{(t)} = \frac{N_m \lambda_m \cdot \mathbf{1}\{m \in \llbracket 1, M_{\text{hist}} \rrbracket\} + b_m \lambda_m \cdot \mathbf{1}\{m \in \llbracket M_{\text{hist}} + 1, M \rrbracket\}}{\sum_{m'=1}^{M_{\text{hist}}} N_{m'} \lambda_{m'} + \sum_{m'=M_{\text{hist}}+1}^M b_{m'} \lambda_{m'}}, \quad (\text{G.798})$$

and

$$p_{m,i} = \frac{\lambda_m T \cdot \mathbf{1}\{m \in \llbracket 1, M_{\text{hist}} \rrbracket\} + \lambda_m \cdot \mathbf{1}\{m \in \llbracket M_{\text{hist}} + 1, M \rrbracket\}}{\sum_{m'=1}^M N_{m'} \lambda_{m'}}. \quad (\text{G.799})$$

Therefore,  $p_{m,i} = \frac{p_m}{N_m}$ , for every sample  $i \in [N_m]$ .

**Bound**  $\text{disc}_{\mathcal{H}}(\mathcal{P}^{(\alpha)}, \mathcal{P}^{(\mathbf{p})})$  Let  $m' \in [M]$ , we have

$$\text{disc}_{\mathcal{H}}(\mathcal{P}^{(\alpha)}, \mathcal{P}^{(\mathbf{p})}) = \sup_{\theta \in \Theta} \left| \sum_{m=1}^M (\alpha_m - p_m) \cdot \mathcal{L}_{\mathcal{P}_m}(\theta) \right| \quad (\text{G.800})$$

$$= \sup_{\theta \in \Theta} \left| \sum_{m=1}^M (\alpha_m - p_m) \cdot (\mathcal{L}_{\mathcal{P}_m}(\theta) - \mathcal{L}_{\mathcal{P}_{m'}}(\theta)) \right|, \quad (\text{G.801})$$

where the last equality follows from the fact that  $\sum_{m=1}^M \alpha_m = \sum_{m=1}^M p_m = 1$ . For all  $m \in [M]$ , we have

$$(\alpha_m - p_m) \cdot (\mathcal{L}_{\mathcal{P}_m}(\theta) - \mathcal{L}_{\mathcal{P}_{m'}}(\theta)) \leq |\alpha_m - p_m| \cdot \left| \mathcal{L}_{\mathcal{P}_m}(\theta) - \mathcal{L}_{\mathcal{P}_{m'}}(\theta) \right| \quad (\text{G.802})$$

$$\leq |\alpha_m - p_m| \cdot \sup_{\theta \in \Theta} \left| \mathcal{L}_{\mathcal{P}_m}(\theta) - \mathcal{L}_{\mathcal{P}_{m'}}(\theta) \right| \quad (\text{G.803})$$

$$= |\alpha_m - p_m| \cdot \text{disc}_{\mathcal{H}}(\mathcal{P}_m, \mathcal{P}_{m'}) \quad (\text{G.804})$$

$$\leq |\alpha_m - p_m| \max_{m,m'} \text{disc}_{\mathcal{H}}(\mathcal{P}_m, \mathcal{P}_{m'}). \quad (\text{G.805})$$

Combining (G.801), and (G.805), we have

$$\text{disc}_{\mathcal{H}}(\mathcal{P}^{(\alpha)}, \mathcal{P}^{(\mathbf{p})}) \leq \sum_{m=1}^M |\alpha_m - p_m| \cdot \max_{m,m'} \text{disc}_{\mathcal{H}}(\mathcal{P}_m, \mathcal{P}_{m'}) \quad (\text{G.806})$$

$$= \|\alpha - \mathbf{p}\|_1 \cdot \max_{m,m'} \text{disc}_{\mathcal{H}}(\mathcal{P}_m, \mathcal{P}_{m'}). \quad (\text{G.807})$$

**Compute**  $N_{\text{eff}}^{-1}$  We have  $N_{\text{eff}}^{-1} = \sum_{m=1}^M \sum_{i=1}^{N_m} \left( \frac{p_m}{N_m} \right)^2 = \sum_{m=1}^M \frac{p_m^2}{N_m} = \frac{1}{N} \sum_{m=1}^M \frac{p_m^2}{n_m}$ .

**Bound**  $\bar{\sigma}(\lambda)$  We have

$$\bar{\sigma}^2(\lambda) = \sum_{t=1}^T q^{(t)} \mathbb{E}_{\mathcal{S}} \left[ \sup_{\theta \in \Theta} \left\| \nabla \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) - \sum_{m=1}^M p_m^{(t)} \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta) \right\|^2 \right]. \quad (\text{G.808})$$

In the settings of Corollary 4.3.5',  $q^{(t)} = 1/T$ , and  $p_m^{(t)} = p_m$ , thus

$$\bar{\sigma}^2(\lambda) = \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{\mathcal{S}} \left[ \sup_{\theta \in \Theta} \left\| \nabla \mathcal{L}_{\mathcal{S}}^{(\lambda)}(\theta) - \sum_{m=1}^M p_m \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}^{(\lambda)}(\theta) \right\|^2 \right], \quad (\text{G.809})$$

where  $\mathcal{L}_{\mathcal{M}_m^{(t)}} = \sum_{j \in \mathcal{I}_m^{(t)}} \ell(\cdot, \mathbf{z}_m^{(j)}) / |\mathcal{I}_m^{(t)}|$ . Moreover, it is easy to check that, in this setting,

$$\mathcal{L}_S^{(\lambda)} = \frac{1}{T} \sum_{t=1}^T \sum_{m=1}^M p_m \cdot \mathcal{L}_{\mathcal{M}_m^{(t)}}. \quad (\text{G.810})$$

Moreover,  $\mathcal{M}_m^{(t)} = \mathcal{M}_m^{(1)}$  for  $m \in [M_{\text{hist}}]$ , thus for  $\theta \in \Theta$ ,

$$\nabla \mathcal{L}_S^{(\lambda)}(\theta) - \sum_{m=1}^M p_m \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}(\theta) = \sum_{m=M_{\text{hist}}+1}^M p_m \cdot \frac{1}{T} \sum_{s=1}^T \left( \nabla \mathcal{L}_{\mathcal{M}_m^{(s)}}(\theta) - \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}(\theta) \right). \quad (\text{G.811})$$

It follows that,

$$\left\| \nabla \mathcal{L}_S^{(\lambda)}(\theta) - \sum_{m=1}^M p_m \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}(\theta) \right\|^2 = \left\| \sum_{m=M_{\text{hist}}+1}^M p_m \cdot \frac{1}{T} \sum_{s=1}^T \left( \nabla \mathcal{L}_{\mathcal{M}_m^{(s)}}(\theta) - \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}(\theta) \right) \right\|^2 \quad (\text{G.812})$$

$$\leq (M - M_{\text{hist}}) \sum_{m=M_{\text{hist}}+1}^M p_m^2 \left\| \frac{1}{T} \sum_{s=1}^T \left( \nabla \mathcal{L}_{\mathcal{M}_m^{(s)}}(\theta) - \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}(\theta) \right) \right\|^2 \quad (\text{G.813})$$

$$\leq (M - M_{\text{hist}}) \sum_{m=M_{\text{hist}}+1}^M \frac{p_m^2}{T} \sum_{t=1}^T \left\| \nabla \mathcal{L}_{\mathcal{M}_m^{(s)}}(\theta) - \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}(\theta) \right\|^2. \quad (\text{G.814})$$

For the fresh clients, i.e., for  $m > M_0$ , we have  $\mathcal{L}_{\mathcal{M}_m^{(t)}}(\theta) = \sum_{i=1}^{b_m} \ell(\theta, z_m^{(t,i)}) / b_m$ , thus

$$\mathbb{E}_S \left\| \nabla \mathcal{L}_{\mathcal{M}_m^{(s)}}(\theta) - \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}(\theta) \right\|^2 \leq \mathbb{E}_S \left\| \frac{1}{b_m} \sum_{i=1}^{b_m} \nabla \ell(\theta; z_m^{(t,i)}) - \nabla \ell(\theta; z_m^{(s,i)}) \right\|^2 \quad (\text{G.815})$$

$$\leq \frac{1}{b_m} \sum_{i=1}^{b_m} \mathbb{E}_S \left\| \nabla \ell(\theta; z_m^{(t,i)}) - \nabla \ell(\theta; z_m^{(s,i)}) \right\|^2 \quad (\text{G.816})$$

$$\leq \sigma_0^2. \quad (\text{G.817})$$

Thus,

$$\mathbb{E}_S \left\| \nabla \mathcal{L}_S^{(\lambda)}(\theta) - \sum_{m=1}^M p_m \nabla \mathcal{L}_{\mathcal{M}_m^{(t)}}(\theta) \right\|^2 \leq \sigma_0^2 (M - M_{\text{hist}}) \cdot \sum_{m=1}^M p_m^2 \quad (\text{G.818})$$

**Conclusion** We conclude the proof by precising that:  $\tilde{c}_0 = (C_1 + C_3)/\sqrt{T} + C_2/\sqrt{T^3}$ , where  $C_1$ ,  $C_2$ , and  $C_3$  are the constant introduced in the proof of Theorem 4.3.3.  $\square$

The third term of (G.791) originates from the variability of the gradients across time as captured by  $\bar{\sigma}^2(\lambda)$  in (4.16). In particular, it only depends on the weights of the fresh clients (as there is no gradient variability for the historical clients). The fourth term in (G.791) corresponds to the discrepancy between the target distribution,  $\mathcal{P}^{(\alpha)}$ , and the effective distribution  $\mathcal{P}^{(\mathbf{p})}$  in (4.16). As expected, it vanishes when all clients have the same distribution, and, for a given heterogeneity

of the local distributions, it is smaller the closer the target relative importance of clients and the effective one are (i.e., the closer  $\alpha$  and  $\mathbf{p}$  are). Finally, the fifth term in (G.791), corresponds to the term  $\tilde{O}\left(\sqrt{\text{Pdim}(\ell \circ \mathcal{H})/N_{\text{eff}}}\right)$  in (4.16), as  $N_{\text{eff}} = N / \left(\sum_{m=1}^M p_m^2/n_m\right)$  in this setting.

### G.4.3 Proof of Corollary 4.3.5

**Corollary 4.3.5.** *Consider the scenario with  $M_{\text{hist}}$  historical clients, and  $M - M_{\text{hist}}$  fresh clients. Suppose that the same assumptions of Theorem 4.3.4 hold, that  $\alpha = \mathbf{n}$ , and that Algorithm 13 is used with clients' aggregation weights  $\mathbf{p} = (p_m)_{m \in [M]} \in \Delta^{M-1}$ , then*

$$\epsilon_{\text{true}} \leq \psi(\mathbf{p}; \mathbf{c}) \triangleq c_0 + c_1 \cdot \sqrt{\sum_{m=M_{\text{hist}}+1}^M p_m^2} + c_2 \cdot \sqrt{\sum_{m=1}^M \frac{p_m^2}{n_m}},$$

where  $\mathbf{c} = (c_0, c_1, c_2)$  are non-negative constants not depending on  $\mathbf{p}$ , given as:

$$\begin{aligned} c_0 &= (C_1 + C_3) + \frac{C_2}{T} \\ c_1 &= \sigma_0 \sqrt{M - M_{\text{hist}}} \cdot \left(D + \frac{2}{\sqrt{T}}\right) \\ c_2 &= 10B \cdot \sqrt{1 + \log\left(\frac{N}{\text{Pdim}(\ell \circ \mathcal{H})}\right)} \cdot \sqrt{\frac{\text{Pdim}(\ell \circ \mathcal{H})}{N}} + 2 \cdot \max_{m, m'} \text{disc}(\mathcal{P}_m, \mathcal{P}_{m'}) \end{aligned}$$

and  $C_1, C_2$ , and  $C_3$  are the constants defined in the proof of Theorem 4.3.3, and  $\sigma_0$  is defined in Remark 6.

*Proof.* We remind that Corollary 4.3.5' implies that

$$\begin{aligned} \epsilon_{\text{true}} &\leq \frac{(C_1 + C_3)}{\sqrt{T}} + \frac{C_2}{\sqrt{T^3}} + 2 \cdot \max_{m, m'} \text{disc}(\mathcal{P}_m, \mathcal{P}_{m'}) \cdot \|\mathbf{n} - \mathbf{p}\|_1 \\ &\quad + 10B \sqrt{1 + \log\left(\frac{N}{\text{Pdim}(\ell \circ \mathcal{H})}\right)} \cdot \sqrt{\frac{\text{Pdim}(\ell \circ \mathcal{H})}{N}} \cdot \sqrt{\sum_{m=1}^M \frac{p_m^2}{n_m}} \\ &\quad + \left(D + \frac{2}{\sqrt{T}}\right) \sigma_0 \sqrt{M - M_{\text{hist}}} \sqrt{\sum_{m=M_{\text{hist}}+1}^M p_m^2}. \end{aligned} \tag{G.819}$$

The result follows using the fact that  $\|\mathbf{p} - \mathbf{n}\|_1 \leq \sqrt{\sum_{m=1}^M p_m^2/n_m} - 1$ , which we prove below.

$$\|\mathbf{p} - \mathbf{n}\|_1 = \sum_{m=1}^M |p_m - n_m| \tag{G.820}$$

$$= \sum_{m=1}^M \frac{|p_m - n_m|}{\sqrt{n_m}} \cdot \sqrt{n_m} \tag{G.821}$$

$$\leq \sqrt{\sum_{m=1}^M \frac{(p_m - n_m)^2}{n_m} \cdot \sum_{m=1}^M n_m} \tag{G.822}$$

$$= \sqrt{\sum_{m=1}^M \frac{(p_m - n_m)^2}{n_m}} \quad (\text{G.823})$$

$$= \sqrt{\sum_{m=1}^M \frac{p_m^2}{n_m} - 2 \sum_{m=1}^M \frac{p_m n_m}{n_m} + \sum_{m=1}^M \frac{n_m^2}{n_m}} \quad (\text{G.824})$$

$$= \sqrt{\sum_{m=1}^M \frac{p_m^2}{n_m} - 1}, \quad (\text{G.825})$$

where we used Cauchy-Schwarz inequality to bound  $\sum_{m=1}^M \frac{|p_m - n_m|}{\sqrt{n_m}} \cdot \sqrt{n_m}$ .  $\square$

#### G.4.4 Proof of the Convexity of $\psi$

We remind that for  $\mathbf{p} \in \Delta^{M-1}$ , and  $\mathbf{c} \in \mathbb{R}_+^3$ , we have

$$\psi(\mathbf{p}; \mathbf{c}) = \frac{c_0}{\sqrt{T}} + c_1 \cdot \sqrt{\sum_{m=M_{\text{hist}}+1}^M p_m^2} + c_2 \cdot \sqrt{\sum_{m=1}^M \frac{p_m^2}{n_m}}. \quad (\text{G.826})$$

In order to prove the convexity of  $\mathbf{p} \mapsto \sqrt{\sum_{m=1}^M \frac{p_m^2}{n_m}}$ , and  $\mathbf{p} \mapsto \sqrt{\sum_{m=M_{\text{hist}}}^M p_m^2}$ , it is sufficient to prove that the function  $\varphi_\beta : \mathbf{p} \mapsto \sqrt{\sum_{m=1}^M \beta_m p_m^2}$  is convex for any vector  $\beta \in \mathbb{R}_+^M$ . Let  $\beta \in \mathbb{R}_+^M$ ,  $\mathbf{p}, \tilde{\mathbf{p}} \in \Delta^M$ , and  $\gamma \in [0, 1]$ , we have

$$\varphi_\beta^2(\gamma \cdot \mathbf{p} + (1 - \gamma) \cdot \tilde{\mathbf{p}}) = \sum_{m=1}^M \beta_m \cdot (\gamma \cdot p_m + (1 - \gamma) \cdot \tilde{p}_m)^2 \quad (\text{G.827})$$

$$= \gamma^2 \cdot \sum_{m=1}^M \beta_m p_m^2 + (1 - \gamma)^2 \cdot \sum_{m=1}^M \beta_m \tilde{p}_m^2 + 2\gamma(1 - \gamma) \cdot \sum_{m=1}^M \beta_m p_m \tilde{p}_m \quad (\text{G.828})$$

$$\leq \gamma^2 \cdot \sum_{m=1}^M \beta_m p_m^2 + (1 - \gamma)^2 \cdot \sum_{m=1}^M \beta_m \tilde{p}_m^2 + 2\gamma(1 - \gamma) \cdot \sqrt{\sum_{m=1}^M \beta_m p_m^2} \cdot \sqrt{\sum_{m=1}^M \beta_m \tilde{p}_m^2} \quad (\text{G.829})$$

$$= \left( \gamma \cdot \sqrt{\sum_{m=1}^M \beta_m p_m^2} + (1 - \gamma) \cdot \sqrt{\sum_{m=1}^M \beta_m \tilde{p}_m^2} \right)^2 \quad (\text{G.830})$$

$$= (\gamma \cdot \varphi_\beta(\mathbf{p}) + (1 - \gamma) \cdot \varphi_\beta(\tilde{\mathbf{p}}))^2, \quad (\text{G.831})$$

where we use Cauchy-Schwartz inequality to bound  $\sum_{m=1}^M \beta_m p_m \tilde{p}_m$ , as follows

$$\sum_{m=1}^M \beta_m p_m \tilde{p}_m = \sum_{m=1}^M (p_m \sqrt{\beta_m}) \cdot (\tilde{p}_m \sqrt{\beta_m \tilde{p}_m}) \leq \sqrt{\sum_{m=1}^M \beta_m p_m^2} \cdot \sqrt{\sum_{m=1}^M \beta_m \tilde{p}_m^2}. \quad (\text{G.832})$$

Since  $\varphi_\beta$  is a non-negative function, we have

$$\varphi_\beta(\gamma \cdot \mathbf{p} + (1 - \gamma) \cdot \tilde{\mathbf{p}}) \leq \gamma \cdot \varphi_\beta(\mathbf{p}) + (1 - \gamma) \cdot \varphi_\beta(\tilde{\mathbf{p}}), \quad (\text{G.833})$$

proving that  $\varphi_\beta$  is convex.

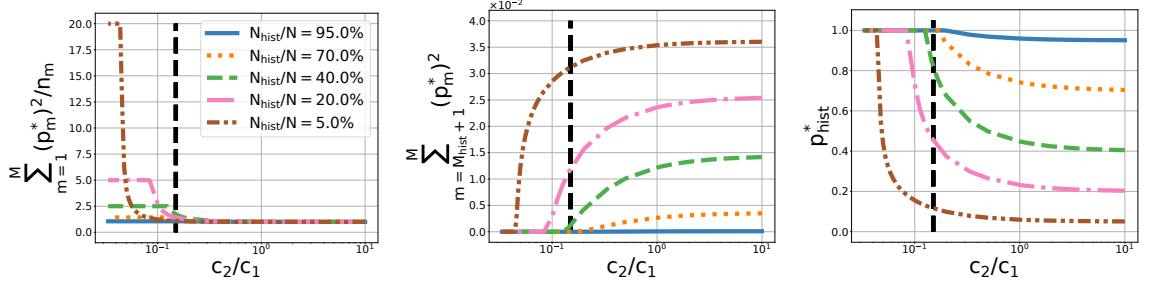


Figure G.22: From left to the right: effect of  $c_2/c_1$  on the effective number of samples, the normalized gradient noise, and the historical clients relative importance  $p_{\text{hist}}^*$  for CIFAR-10 dataset ( $N = 5 \times 10^5$ ) and different values of  $N_{\text{hist}}/N$ , when  $M = 50$ , and  $M_{\text{hist}} = 25$ . The dashed vertical line corresponds to our estimation of  $c_2/c_1$  on CIFAR-10 experiments ( $\hat{c}_2/\hat{c}_1 = 0.15$ ).

#### G.4.5 Numerical Study of Bound Minimization

Figure G.22 illustrates how the solution and important system quantities change as a function of the ratio  $c_2/c_1$ , and fraction of historical samples  $N_{\text{hist}}/N$ , in the particular setting when  $M = 50$  and  $M_{\text{hist}} = 25$ . Beside the specific numerical values, one can distinguish two corner cases. When  $c_2/c_1 \gg 1$ , the optimal solution corresponds to minimize  $\sum_{m=1}^M p_m^2/n_m$ , i.e., to maximize the effective number of samples, and then  $\sum_m (p_m^*)^2/n_m$ . The optimal aggregation vector  $\mathbf{p}^*$  is then the `Uniform` one: each sample is assigned the same importance during the whole training and each client a relative importance proportional to its number of samples ( $p_m^* = n_m$ ). In particular, the aggregate relative importance for historical clients is  $p_{\text{hist}}^* = N_{\text{hist}}/N$ . On the contrary, when  $c_2/c_1 \ll 1$ , the optimal solution corresponds to minimize  $\sum_{m > M_{\text{hist}}} p_m$ , i.e., the gradient variability. The `Historical` strategy is then optimal: fresh clients are ignored and historical clients receive a relative importance proportional to the size of their local dataset (i.e.,  $p_m^* = N_m/N_{\text{hist}} = \frac{N}{N_{\text{hist}}} n_m$  for  $m \in [M_{\text{hist}}]$  and  $p_{\text{hist}}^* = 1$ ). Figure G.22 confirms these qualitative considerations, but also shows that the transition between these two regimes depends on  $N_{\text{hist}}/N$ , with the transition occurring at smaller values of  $c_2/c_1$  for smaller values of the  $N_{\text{hist}}/N$ .

#### G.4.6 Details on the Estimation of the $c_2/c_1$

Using the expression of  $c_1$  and  $c_2$  from Corollary 4.3.5, we have

$$\frac{c_2}{c_1} = 2 \cdot \frac{\max_{m,m'} \text{disc}(\mathcal{P}_m, \mathcal{P}_{m'}) + 5B \cdot \sqrt{1 + \log\left(\frac{N}{\text{Pdim}(\ell \circ \mathcal{H})}\right)} \cdot \sqrt{\frac{\text{Pdim}(\ell \circ \mathcal{H})}{N}}}{\sigma_0 \sqrt{M - M_{\text{hist}}} \cdot \left(D + \frac{2}{\sqrt{T}}\right)}. \quad (\text{G.834})$$

We use the approximations

$$\sqrt{1 + \log\left(\frac{N}{\text{Pdim}(\ell \circ \mathcal{H})}\right)} \approx 2, \quad (\text{G.835})$$

$$D + \frac{2}{\sqrt{T}} \approx D, \quad (\text{G.836})$$

$$\text{Pdim}(\ell \circ \mathcal{H}) \approx d/(10B)^2, \quad (\text{G.837})$$



where  $d$  is the number of parameters of the model  $\theta \in \Theta \subset \mathbb{R}^d$  (see Section 4.3.1). We remind the definition of  $\sigma_0$  from Remark 6:

$$\sigma_0 = \sqrt{\max_m \mathbb{E}_{\mathbf{z} \sim \mathcal{P}_m} \left[ \sup_{\theta \in \Theta} \|\nabla \ell(\theta; \mathbf{z}) - \nabla \mathcal{L}_{\mathcal{P}_m}(\theta)\|^2 \right]} \leq 2\sqrt{2} \cdot LB = 2G, \quad (\text{G.838})$$

where  $G$  was defined in (G.705). We use the approximation  $\sigma_0 \approx 2G$ . Finally, we remark that  $\max_{m,m'} \text{disc}(\mathcal{P}_m, \mathcal{P}_{m'}) \leq B$ , therefore, we approximate  $c_2/c_1$  as

$$\frac{\hat{c}_2}{\hat{c}_1} \approx \frac{B + \sqrt{d/N}}{GD\sqrt{M - M_{\text{hist}}}}. \quad (\text{G.839})$$

In our experiments, clients cooperatively estimate  $\hat{c}_2/\hat{c}_1$  using a fraction of their historical samples, with the particularity that  $D$  is estimated as  $\hat{D} = \max_{m=1}^M \|\hat{\theta}_m^* - \theta^{(1)}\|$ , where  $\hat{\theta}_m^*$  is the model obtained after few iterations of stochastic gradient descent using a fraction of the historical data of client  $m \in [M]$ .

## H Online Federated Learning with Mixture Models

### H.1 Proof of Theorem 4.4.2

**Theorem 4.4.2.** *Suppose that assumptions 27–30 hold. Suppose that  $n \geq \frac{C''d}{\beta\epsilon^2} \cdot \log^2\left(\frac{m^2TK}{\delta}\right)$  with sufficiently large universal constant  $C''$ , and  $K = \mathcal{O}(\log(1/\epsilon))$ . Algorithm 15 has regret bounded by*

$$\forall c \in \mathcal{C}, \quad R_{T,c} = \mathcal{O}(T\epsilon) + \mathcal{O}\left(\sqrt{T \log(m)}\right), \quad (\text{H.840})$$

with probability at least  $1 - \mathcal{O}(\delta) - \mathcal{O}(TK/n^{c'-2}m^{30})$ .

*Proof.* The result follows by combining (4.42) and Theorem 4.4.5.  $\square$

### H.2 Proof of Theorem 4.4.3

**Theorem 4.4.3.** *Let  $(\psi_t)_{0 \leq t \leq T}$  be a sequence of convex functions, and  $\mathbf{u} \in \mathcal{X}$ . Suppose the gradient norm is bounded as  $\|\hat{\nabla}_t\|_t^* \leq G_{\mathcal{R}}$ , and the stepsize is  $\eta = \frac{D_{\mathcal{R}}}{G_{\mathcal{R}}\sqrt{T}}$ . Then, for Algorithm 16 we have*

$$\sum_{t=1}^T \psi_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{s=1}^T \psi_s(\mathbf{x}) \leq D_{\mathcal{R}}G_{\mathcal{R}}\sqrt{T} + D_{\mathcal{X}} \cdot \sum_{t=1}^T \left\| \nabla \psi_t(\mathbf{x}_t) - \hat{\nabla}_t \right\|, \quad (\text{H.841})$$

where  $D_{\mathcal{X}} \triangleq \sup_{\mathbf{x}, \mathbf{y} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|$  is the diameter of  $\mathcal{X}$ .

*Proof.* Let  $\mathbf{x}^* \in \arg \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^T \psi(\mathbf{x})$ . For  $t \in [T]$ , the function  $\psi_t$  is convex. Therefore,

$$\begin{aligned} \psi(\mathbf{x}_t) - \psi(\mathbf{x}^*) &\leq \langle \nabla \psi_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^* \rangle \end{aligned} \quad (\text{H.842})$$

$$= \langle \hat{\nabla}_t, \mathbf{x}_t - \mathbf{x}^* \rangle + \langle \hat{\nabla}_t - \nabla \psi_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^* \rangle \quad (\text{H.843})$$

$$= \frac{\langle \nabla \mathcal{R}(\mathbf{x}_t) - \nabla \mathcal{R}(\mathbf{y}_{t+1}), \mathbf{x}_t - \mathbf{x}^* \rangle}{\eta} + \langle \hat{\nabla}_t - \nabla \psi_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^* \rangle \quad (\text{H.844})$$

$$= \frac{\mathcal{R}(\mathbf{x}^*|\mathbf{x}_t) - \mathcal{R}(\mathbf{x}^*|\mathbf{y}_{t+1}) + \mathcal{R}(\mathbf{x}_t|\mathbf{y}_{t+1})}{\eta} + \langle \hat{\nabla}_t - \nabla \psi_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^* \rangle \quad (\text{H.845})$$

$$\leq \frac{\mathcal{R}(\mathbf{x}^*|\mathbf{x}_t) - \mathcal{R}(\mathbf{x}^*|\mathbf{x}_{t+1}) + \mathcal{R}(\mathbf{x}_t|\mathbf{y}_{t+1})}{\eta} + \left\| \hat{\nabla}_t - \nabla \psi_t(\mathbf{x}_t) \right\| \cdot \|\mathbf{x}_t - \mathbf{x}^*\| \quad (\text{H.846})$$

$$\leq \frac{\mathcal{R}(\mathbf{x}^*|\mathbf{x}_t) - \mathcal{R}(\mathbf{x}^*|\mathbf{x}_{t+1}) + \mathcal{R}(\mathbf{x}_t|\mathbf{y}_{t+1})}{\eta} + D_{\mathcal{X}} \cdot \left\| \hat{\nabla}_t - \nabla \psi_t(\mathbf{x}_t) \right\| \quad (\text{H.847})$$

Thus, summing over time we have

$$\begin{aligned} \sum_{t=1}^T \psi_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{s=1}^T \psi_s(\mathbf{x}) &\leq \frac{\mathcal{R}(\mathbf{x}^*|\mathbf{x}_1) - \mathcal{R}(\mathbf{x}^*|\mathbf{x}_T)}{\eta} + \sum_{t=1}^T \frac{\mathcal{R}(\mathbf{x}_t|\mathbf{y}_{t+1})}{\eta} + D_{\mathcal{X}} \cdot \sum_{t=1}^T \left\| \hat{\nabla}_t - \nabla \psi_t(\mathbf{x}_t) \right\| \end{aligned} \quad (\text{H.848})$$

$$\leq \sum_{t=1}^T \frac{\mathcal{R}(\mathbf{x}_t \| \mathbf{y}_{t+1})}{\eta} + \frac{D_{\mathcal{R}}^2}{\eta} + D_{\mathcal{X}} \cdot \sum_{t=1}^T \left\| \hat{\nabla}_t - \nabla \psi_t(\mathbf{x}_t) \right\|. \quad (\text{H.849})$$

The rest of the proof follows exactly as in the proof of [Haz16, Theorem 5.6].  $\square$

### H.3 Proof of Theorem 4.4.5

**Theorem 4.4.4.** *Suppose Assumptions 27–30 hold, and the number of samples satisfies  $n \geq \frac{C''d}{\beta\epsilon^2} \cdot \log^2(m^2TK/\delta)$ , where  $C''$  is a sufficiently large universal constant, and the number of inner loop steps satisfies  $K = \mathcal{O}(\log(1/\epsilon))$ . Then, for all  $t \in [T]$ ,  $c \in \mathcal{C}$  and  $j \in [m]$ , we have*

$$\pi_{t,c,j} \cdot \left| \nabla_{t,c,j} - \tilde{\nabla}_{t,c,j} \right| \leq \frac{3}{2} \cdot \sup_{q \in [m]} \left\| \boldsymbol{\mu}_{t+1,q} - \boldsymbol{\mu}_q^* \right\| + \left\| \tilde{\pi}_{t,c,j}^{(K+1)} - \pi_{t,c,j}^* \right\|_1. \quad (\text{H.850})$$

*Proof.* From Lemma H.1, we have

$$\begin{aligned} \pi_{t,c,j} \cdot \left| \nabla_{t,c,j} - \tilde{\nabla}_{t,c,j} \right| &\leq \left\| \sum_{j=1}^m \tilde{\pi}_{t,c,j}^{(K+1)} \mathcal{N}(\boldsymbol{\mu}_{t+1,j}, \mathbf{I}_d) - \sum_{j=1}^m \pi_{t,c,j}^* \mathcal{N}(\boldsymbol{\mu}_j^*, \mathbf{I}_d) \right\|_{\text{TV}} \\ &\quad + \gamma \cdot \sup_{q \in [m]} \left\| \boldsymbol{\mu}_{t+1,q} - \boldsymbol{\mu}_q^* \right\|. \end{aligned} \quad (\text{H.851})$$

Therefore, we need to bound  $\left\| \sum_{j=1}^m \tilde{\pi}_{t,c,j}^{(K+1)} \mathcal{N}(\boldsymbol{\mu}_{t+1,j}, \mathbf{I}_d) - \sum_{j=1}^m \pi_{t,c,j}^* \mathcal{N}(\boldsymbol{\mu}_j^*, \mathbf{I}_d) \right\|_{\text{TV}}$  in order to prove Theorem 4.4.4. Using Lemma H.9, we have

$$\begin{aligned} &\left\| \sum_{j=1}^m \tilde{\pi}_{t,c,j}^{(K+1)} \mathcal{N}(\boldsymbol{\mu}_{t+1,j}, \mathbf{I}_d) - \sum_{j=1}^m \pi_{t,c,j}^* \mathcal{N}(\boldsymbol{\mu}_j^*, \mathbf{I}_d) \right\|_{\text{TV}} \\ &\leq \sum_{i=1}^m \pi_{t,c,j}^* \frac{\left\| \boldsymbol{\mu}_{t+1,j} - \boldsymbol{\mu}_j^* \right\|}{2} + \left\| \tilde{\pi}_{t,c,j}^{(K+1)} - \pi_{t,c,j}^* \right\|_1 \end{aligned} \quad (\text{H.852})$$

$$\leq \sum_{i=1}^m \frac{\pi_{t,c,j}^*}{2} \sup_{q \in [m]} \left\| \boldsymbol{\mu}_{t+1,q} - \boldsymbol{\mu}_q^* \right\| + \left\| \tilde{\pi}_{t,c,j}^{(K+1)} - \pi_{t,c,j}^* \right\|_1 \quad (\text{H.853})$$

$$\leq \frac{1}{2} \sup_{q \in [m]} \left\| \boldsymbol{\mu}_{t+1,q} - \boldsymbol{\mu}_q^* \right\| + \left\| \tilde{\pi}_{t,c,j}^{(K+1)} - \pi_{t,c,j}^* \right\|_1. \quad (\text{H.854})$$

$\square$

**Theorem 4.4.5.** *Suppose Assumptions 27–30 hold, and the number of samples satisfies the condition  $n \geq \frac{C''d}{\beta\epsilon^2} \cdot \log^2(m^2TK/\delta)$ ,  $C''$  is a sufficiently large universal constant, and the number of inner loop iterations is selected as  $K = \mathcal{O}(\log(1/\epsilon))$ . Then, for all  $t \in [T]$  and for all  $c \in \mathcal{C}$ , with probability at least  $1 - \mathcal{O}(\delta/T) - \mathcal{O}(K/n^{c'-2}m^{30})$ , we have  $\left\| \nabla_{t,c} - \tilde{\nabla}_{t,c} \right\| = \mathcal{O}(\epsilon)$ .*

*Proof.* From Theorem 4.4.1, with probability at least  $1 - \mathcal{O}(\delta/T) - \mathcal{O}(K/n^{c'-2}m^{30})$ , we have for  $j \in [m]$

$$\left| \tilde{\pi}_{t,j}^{(K+1)} - \pi_{t,c,j}^* \right| \leq \epsilon \pi_{t,c,j}^*, \quad \left\| \tilde{\boldsymbol{\mu}}_{t,c,j}^{(K+1)} - \boldsymbol{\mu}_j^* \right\| \leq \epsilon. \quad (\text{H.855})$$

From Theorem 4.4.4, we have

$$\pi_{t,c,j} \cdot \left| \nabla_{t,c,j} - \tilde{\nabla}_{t,c,j} \right| \leq \frac{3}{2} \cdot \sup_{q \in [m]} \left\| \boldsymbol{\mu}_{t+1,q} - \boldsymbol{\mu}_q^* \right\| + \left\| \tilde{\pi}_{t,c,j}^{(K+1)} - \pi_{t,c,j}^* \right\|_1. \quad (\text{H.856})$$

Therefore

$$\left| \nabla_{t,c,j} - \tilde{\nabla}_{t,c,j} \right| = \mathcal{O}(\epsilon). \quad (\text{H.857})$$

□

**Lemma H.1.** For  $t \in [T]$  and  $j \in [m]$ , we have

$$\begin{aligned} \pi_{t,c,j} \cdot \left| \nabla_{t,c,j} - \tilde{\nabla}_{t,c,j} \right| &\leq \left\| \sum_{j=1}^m \tilde{\pi}_{t,c,j}^{(K+1)} \mathcal{N}(\boldsymbol{\mu}_{t+1,j}, \mathbf{I}_d) - \sum_{j=1}^m \pi_{t,c,j}^* \mathcal{N}(\boldsymbol{\mu}_j^*, \mathbf{I}_d) \right\|_{\text{TV}} \\ &\quad + \gamma \cdot \sup_{q \in [m]} \left\| \boldsymbol{\mu}_{t+1,q} - \boldsymbol{\mu}_q^* \right\|, \end{aligned} \quad (\text{H.858})$$

for some absolute constant  $\gamma < 1$ .

*Proof.* We define  $\tilde{\theta}'_{t,c} \triangleq \left\{ \left( \tilde{\boldsymbol{\mu}}_{t+1,j}, \tilde{\pi}_{t,c,j}^{(K+1)} \right) : j \in [m] \right\}$ . For  $t \in [T]$  and  $j \in [m]$ , we have

$$\pi_{t,j} \cdot \left| \nabla_{t,c,j} - \tilde{\nabla}_{t,c,j} \right| \quad (\text{H.859})$$

$$= \left| \mathbb{E}_{X \sim \mathcal{P}_{\theta_{t,c}^*}} \left[ \frac{\pi_{t,c,j} f_{\mu_j^*}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_l^*}(X)} \right] - \mathbb{E}_{X \sim \mathcal{P}_{\tilde{\theta}'_{t,c}}} \left[ \frac{\pi_{t,c,j} f_{\mu_{t+1,j}}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}}(X)} \right] \right| \quad (\text{H.860})$$

$$\begin{aligned} &= \left| \mathbb{E}_{X \sim \mathcal{P}_{\theta_{t,c}^*}} \left[ \frac{\pi_{t,c,j} f_{\mu_j^*}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_l^*}(X)} \right] - \mathbb{E}_{X \sim \mathcal{P}_{\theta_{t,c}^*}} \left[ \frac{\pi_{t,c,j} f_{\mu_{t+1,j}}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}}(X)} \right] \right| \\ &\quad + \left| \mathbb{E}_{X \sim \mathcal{P}_{\theta_{t,c}^*}} \left[ \frac{\pi_{t,c,j} f_{\mu_{t+1,j}}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}}(X)} \right] - \mathbb{E}_{X \sim \mathcal{P}_{\tilde{\theta}'_{t,c}}} \left[ \frac{\pi_{t,c,j} f_{\mu_{t+1,j}}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}}(X)} \right] \right| \end{aligned} \quad (\text{H.861})$$

$$\begin{aligned} &\leq \left| \mathbb{E}_{X \sim \mathcal{P}_{\theta_{t,c}^*}} \left[ \frac{\pi_{t,c,j} f_{\mu_{t+1,j}}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}}(X)} \right] - \mathbb{E}_{X \sim \mathcal{P}_{\tilde{\theta}'_{t,c}}} \left[ \frac{\pi_{t,c,j} f_{\mu_{t+1,j}}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}}(X)} \right] \right| \\ &\quad + \left| \mathbb{E}_{X \sim \mathcal{P}_{\theta_{t,c}^*}} \left[ \frac{\pi_{t,c,j} f_{\mu_j^*}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_l^*}(X)} - \frac{\pi_{t,c,j} f_{\mu_{t+1,j}}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}}(X)} \right] \right| \end{aligned} \quad (\text{H.862})$$

$$\begin{aligned} &\leq \max_{\mathbf{x} \in \mathbb{R}^d} \left\{ \frac{\pi_{t,c,j} f_{\mu_{t+1,j}}(\mathbf{x})}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}}(\mathbf{x})} \right\} \cdot \left\| \sum_{j=1}^m \tilde{\pi}_{t,c,j}^{(K+1)} \mathcal{N}(\boldsymbol{\mu}_{t+1,j}, \mathbf{I}_d) - \sum_{j=1}^m \pi_{t,c,j}^* \mathcal{N}(\boldsymbol{\mu}_j^*, \mathbf{I}_d) \right\|_{\text{TV}} \\ &\quad + \left| \mathbb{E}_{X \sim \mathcal{P}_{\theta_{t,c}^*}} \left[ \frac{\pi_{t,c,j} f_{\mu_j^*}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_l^*}(X)} - \frac{\pi_{t,c,j} f_{\mu_{t+1,j}}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}}(X)} \right] \right| \end{aligned} \quad (\text{H.863})$$

$$\leq \underbrace{\left| \mathbb{E}_{X \sim \mathcal{P}_{\theta_{t,c}^*}} \left[ \frac{\pi_{t,c,j} f_{\mu_{t+1,j}}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}}(X)} - \frac{\pi_{t,c,j} f_{\mu_j^*}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_l^*}(X)} \right] \right|}_{\triangleq A(X)}$$

$$+ \leq \left\| \sum_{j=1}^m \tilde{\pi}_{t,c,j}^{(K+1)} \mathcal{N}(\boldsymbol{\mu}_{t+1,j}, \mathbf{I}_d) - \sum_{j=1}^m \pi_{t,c,j}^* \mathcal{N}(\boldsymbol{\mu}_j^*, \mathbf{I}_d) \right\|_{\text{TV}}. \quad (\text{H.864})$$

The last part of the proof consist in bounding the first term of the RHS of (H.864). We use the same technique as in [KC20, Appendix E] and borrow their notation; for  $u \in [0, 1]$ , we define the function

$$\psi(u) = \frac{\pi_{t,c,j} f_{\mu_{t+1,j}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}^u}(X)} - \frac{\pi_{t,c,j} f_{\mu_j^*}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_l^*}(X)}, \quad (\text{H.865})$$

where  $\mu_{t+1,l}^u = \mu_{t+1,l} + u(\mu_l^* - \mu_{t+1,l})$  for  $l \in [m]$ . The function  $\psi$  is differentiable on the open interval  $(0, 1)$ . Furthermore, we observe that  $\psi(0) = 0$  and  $\psi(1) = A(X)$ . Using the mean value theorem, there exists  $u \in (0, 1)$  such that  $\psi'(u) = A(X)$ . Therefore,

$$\begin{aligned} A(X) &= -\frac{\pi_{t,c,j} f_{\mu_{t+1,j}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}^u}(X)} \left( 1 - \frac{\pi_{t,c,j} f_{\mu_{t+1,j}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}^u}(X)} \right) \cdot (X - \boldsymbol{\mu}_{t+1,j}^u)^\top (\boldsymbol{\mu}_{t+1,j} - \boldsymbol{\mu}_j^*) \\ &\quad + \sum_{q \neq j} \frac{\pi_{t,c,j} f_{\mu_{t+1,j}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}^u}(X)} \frac{\pi_{t,c,j} f_{\mu_{t+1,q}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}^u}(X)} \cdot (X - \boldsymbol{\mu}_{t+1,q}^u)^\top (\boldsymbol{\mu}_{t+1,q} - \boldsymbol{\mu}_q^*). \end{aligned} \quad (\text{H.866})$$

For  $r \neq j$ , we bound the first term of the RHS of (H.866).

$$e_{r,1} \triangleq \left| \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}_r, \mathbf{I}_d)} \left[ \frac{\pi_{t,c,j} f_{\mu_{t+1,j}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}^u}(X)} \left( 1 - \frac{\pi_{t,c,j} f_{\mu_{t+1,j}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}^u}(X)} \right) \right] \right| \quad (\text{H.867})$$

$$\times \left| (X - \boldsymbol{\mu}_{t+1,j}^u)^\top (\boldsymbol{\mu}_{t+1,j} - \boldsymbol{\mu}_j^*) \right| \quad (\text{H.868})$$

$$\leq 2 \left\| \boldsymbol{\mu}_{t+1,j} - \boldsymbol{\mu}_j^* \right\| \cdot \sup_{\|s\|=1} \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}_r, \mathbf{I}_d)} \left[ \frac{\pi_{t,c,j} f_{\mu_{t+1,j}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}^u}(X)} \langle X - \boldsymbol{\mu}_{t+1,j}^u, s \rangle \right] \quad (\text{H.869})$$

$$\leq \gamma \cdot \sup_q \left\| \boldsymbol{\mu}_q^* - \boldsymbol{\mu}_{t+1,q} \right\|, \quad (\text{H.870})$$

where we used Lemma H.2 to obtain the last inequality. Similarly, We bound now the second term the RHS of (H.866).

$$e_{r,2} \triangleq \left| \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}_r, \mathbf{I}_d)} \left[ \sum_{q \neq j} \frac{\pi_{t,c,j} f_{\mu_{t+1,j}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}^u}(X)} \frac{\pi_{t,c,j} f_{\mu_{t+1,q}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}^u}(X)} \right] \right| \quad (\text{H.871})$$

$$\times \left| (X - \boldsymbol{\mu}_{t+1,q}^u)^\top (\boldsymbol{\mu}_{t+1,q} - \boldsymbol{\mu}_q^*) \right| \quad (\text{H.872})$$

$$\leq 2 \sum_{q \neq j} \left\| \boldsymbol{\mu}_{t+1,q} - \boldsymbol{\mu}_q^* \right\| \quad (\text{H.873})$$

$$\times \left| \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}_r, \mathbf{I}_d)} \left[ \frac{\pi_{t,c,j} f_{\mu_{t+1,j}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}^u}(X)} \frac{\pi_{t,c,j} f_{\mu_{t+1,q}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\mu_{t+1,l}^u}(X)} \cdot (X - \boldsymbol{\mu}_{t+1,q}^u) \right] \right| \quad (\text{H.874})$$

$$\leq 2 \sup_q \left\| \boldsymbol{\mu}_q^* - \boldsymbol{\mu}_{t+1,q} \right\| \sqrt{\mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}_r, I_d)} \left[ \frac{\pi_{t,c,j} f_{\boldsymbol{\mu}_{t+1,j}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\boldsymbol{\mu}_{t+1,l}^u}(X)} \right]} \quad (\text{H.875})$$

$$\times \sum_{q \neq j} \sqrt{\sup_{\|\mathbf{s}\|=1} \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}_r, I_d)} \left[ \frac{\pi_{t,c,j} f_{\boldsymbol{\mu}_{t+1,q}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\boldsymbol{\mu}_{t+1,l}^u}(X)} \langle X - \boldsymbol{\mu}_{t+1,q}^u, \mathbf{s} \rangle^2 \right]} \quad (\text{H.876})$$

$$\leq \gamma \cdot \sup_q \left\| \boldsymbol{\mu}_q^* - \boldsymbol{\mu}_{t+1,q} \right\|, \quad (\text{H.877})$$

where we used Lemma H.2 and Lemma H.3 to obtain the last inequality.

Combining (H.870) and (H.877), we have

$$\mathbb{E}_{X \sim \mathcal{P}_{\theta_{t,c}^*}} [A(X)] = \gamma \cdot \sup_q \left\| \boldsymbol{\mu}_{t+1,q} - \boldsymbol{\mu}_q^* \right\|, \quad (\text{H.878})$$

for some small  $\gamma < 1$ .  $\square$

**Lemma H.2.** *There exists a sufficiently small constant  $\gamma$ , such that*

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}_r, I_d)} \left[ \frac{\pi_{t,c,j} f_{\boldsymbol{\mu}_{t+1,j}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\boldsymbol{\mu}_{t+1,l}^u}(X)} \right] \leq \gamma \quad (\text{H.879})$$

$$\sup_{\|\mathbf{s}\|=1} \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}_r, I_d)} \left[ \frac{\pi_{t,c,j} f_{\boldsymbol{\mu}_{t+1,j}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\boldsymbol{\mu}_{t+1,l}^u}(X)} \langle X - \boldsymbol{\mu}_{t+1,j}^u, \mathbf{s} \rangle \right] \leq \gamma \quad (\text{H.880})$$

*Proof.* The result follows from [KC20, Lemma 6] and [KC20, Corollary 5].  $\square$

**Lemma H.3.** [KC20, Lemma 27] *Let  $j \in [m]$ . There exists a small constant  $c_0 > 0$ , such that*

$$\sum_{q \neq j} \sqrt{\sup_{\|\mathbf{s}\|=1} \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}_r, I_d)} \left[ \frac{\pi_{t,c,j} f_{\boldsymbol{\mu}_{t+1,q}^u}(X)}{\sum_{l=1}^m \pi_{t,c,l} f_{\boldsymbol{\mu}_{t+1,l}^u}(X)} \langle X - \boldsymbol{\mu}_{t+1,q}^u, \mathbf{s} \rangle^2 \right]} \leq c_0. \quad (\text{H.881})$$

## H.4 Supporting Lemmas

**Lemma H.4.** *We have*

$$\sup_{\boldsymbol{\pi}, \boldsymbol{\pi}' \in \Delta^{m-1}} \sum_{j=1}^m \left\{ \pi_j \log(\pi_j) - \pi'_j \log(\pi'_j) \right\} = \log(m). \quad (\text{H.882})$$

*Proof.* We first remark that  $\sum_j \pi_j \log(\pi_j) \leq 0$  for every  $\boldsymbol{\pi} \in \Delta^{m-1}$ . Let  $\epsilon > 0$ , and consider  $\boldsymbol{\pi}_\epsilon = [1 - \epsilon, \epsilon/m-1, \dots, \epsilon/m-1] \in \Delta^{m-1}$ . Then,

$$\lim_{\epsilon \rightarrow 0} \sum_{j=1}^m \pi_{\epsilon,j} \log(\pi_{\epsilon,j}) = \lim_{\epsilon \rightarrow 0} (1 - \epsilon) \log(1 - \epsilon) + \epsilon \log\left(\frac{\epsilon}{m-1}\right) = 0. \quad (\text{H.883})$$

Therefore,  $\sup_{\boldsymbol{\pi}} \sum_j \pi_j \log(\pi_j) = 0$ . Using the the concavity of the  $\log$  function and Jensen's inequality, we prove that

$$-\sum_{j=1}^m \pi_j \log(\pi_j) = \sum_{j=1}^m \pi_j \log\left(\frac{1}{\pi_j}\right) \leq \log\left(\sum_{j=1}^m \frac{\pi_j}{\pi_j}\right) = \log(m), \quad (\text{H.884})$$

with equality if and only if  $\boldsymbol{\pi} = [1/m, \dots, 1/m]$ . Therefore,  $\sup_{\boldsymbol{\pi}} \left\{ -\sum_j \pi_j \log(\pi_j) \right\} = \log(m)$ .  $\square$

**Lemma H.5.** *Suppose that assumptions 27–30 hold. Suppose that  $n \geq \frac{C'd}{\beta\epsilon^2} \cdot \log^2\left(\frac{m^2TK}{\delta}\right)$  with sufficiently large universal constant  $C'$ , and  $K = \mathcal{O}(\log(1/\epsilon))$ . The iterates of Algorithm 15 verify  $\|\hat{\nabla}_t\|_\infty = \mathcal{O}(1)$ .*

*Proof.* Since  $\hat{\nabla}_t$  is a Monte-Carlo approximation of  $\tilde{\nabla}_t$ , it is enough to prove that  $\|\tilde{\nabla}_t\|_\infty = \mathcal{O}(1)$ . The result from the separation assumption (Assumption 27) using [KC20, Corollary 5].  $\square$

**Lemma H.6.** (Pinsker's inequality [Tsy08, Lemma 2.5]) *Let  $P$  and  $Q$  be two probability distributions on the same measurable space, then*

$$\|P - Q\|_{\text{TV}}^2 \leq \frac{D_{\text{KL}}(P\|Q)}{2} \quad (\text{H.885})$$

**Lemma H.7.** *Let  $\mu, \mu' \in \mathbb{R}^d$ . Then,*

$$D_{\text{KL}}(f_\mu\|f_{\mu'}) = \frac{\|\mu - \mu'\|^2}{2} \quad (\text{H.886})$$

*Proof.* We have

$$D_{\text{KL}}(f_\mu\|f_{\mu'}) = \int_{\mathbf{x} \in \mathbb{R}^d} f_\mu(\mathbf{x}) \log\left(\frac{f_\mu(\mathbf{x})}{f_{\mu'}(\mathbf{x})}\right) d\mathbf{x} \quad (\text{H.887})$$

$$= \int_{\mathbf{x} \in \mathbb{R}^d} f_\mu(\mathbf{x}) \cdot \frac{\|\mathbf{x} - \mu\|^2 - \|\mathbf{x} - \mu'\|^2}{2} d\mathbf{x} \quad (\text{H.888})$$

$$= \int_{\mathbf{x} \in \mathbb{R}^d} f_\mu(\mathbf{x}) \cdot \frac{2\langle \mu' - \mu, \mathbf{x} \rangle + \|\mu'\|^2 - \|\mu\|^2}{2} d\mathbf{x} \quad (\text{H.889})$$

$$= \frac{2\langle \mu' - \mu, \mu \rangle + \|\mu'\|^2 - \|\mu\|^2}{2} \quad (\text{H.890})$$

$$= \frac{\|\mu - \mu'\|^2}{2}. \quad (\text{H.891})$$

$\square$

**Lemma H.8.** *Let  $\mu, \mu' \in \mathbb{R}^d$ . Then,*

$$\|f_\mu - f_{\mu'}\|_{\text{TV}} \leq \frac{\|\mu - \mu'\|}{2}. \quad (\text{H.892})$$

*Proof.* The result follows using Lemma H.6 and Lemma H.7.  $\square$

**Lemma H.9.** *Let  $\mu_1, \dots, \mu_m, \mu'_1, \dots, \mu'_m \in \mathbb{R}^d$ , and  $\pi, \pi' \in \delta^{m-1}$ . Then,*

$$\left\| \sum_{i=1}^m \pi_i f_{\mu_i} - \sum_{i=1}^m \pi'_i f_{\mu'_i} \right\|_{\text{TV}} \leq \sum_{i=1}^m \pi_i \frac{\|\mu_i - \mu'_i\|}{2} + \|\pi - \pi'\|_1. \quad (\text{H.893})$$

*Proof.* We have

$$\left\| \sum_{i=1}^m \pi_i f_{\mu_i} - \sum_{i=1}^m \pi'_i f_{\mu'_i} \right\|_{\text{TV}} = \left\| \sum_{i=1}^m \pi_i f_{\mu_i} - \sum_{i=1}^m \pi_i f_{\mu'_i} + \sum_{i=1}^m \pi_i f_{\mu'_i} - \sum_{i=1}^m \pi'_i f_{\mu'_i} \right\|_{\text{TV}} \quad (\text{H.894})$$

$$= \left\| \sum_{i=1}^m \pi_i (f_{\mu_i} - f_{\mu'_i}) + \sum_{i=1}^m (\pi_i - \pi'_i) f_{\mu'_i} \right\|_{\text{TV}} \quad (\text{H.895})$$

$$\leq \sum_{i=1}^m \pi_i \|f_{\mu_i} - f_{\mu'_i}\|_{\text{TV}} + \sum_{i=1}^m |\pi_i - \pi'_i| \quad (\text{H.896})$$

$$\leq \sum_{i=1}^m \pi_i \frac{\|\mu_i - \mu'_i\|}{2} + \|\pi - \pi'\|_1. \quad (\text{H.897})$$

□

**Lemma H.10.** For  $\mu_1, \dots, \mu_m \in \mathbb{R}^d$ , the function  $\pi \mapsto \text{D}_{\text{KL}} \left( \sum_{j=1}^m \pi_{t,j}^* f_{\mu_j^*} \middle\| \sum_{j=1}^m \pi_j f_{\mu_j} \right)$ , defined on  $\Delta^{m-1}$  is convex.

*Proof.* The result follows from the convexity of the function  $x \mapsto -\log(x)$ . □







# Surmonter l'Hétérogénéité dans les Systèmes d'Apprentissage Fédéré

Othmane MARFOQ

## Résumé

L'apprentissage fédéré, qui provient de l'anglais "Federated Learning" (FL), se présente comme un cadre facilitant l'apprentissage collaboratif de modèles d'apprentissage automatique par des clients géographiquement répartis sans divulguer leurs données locales. Cette thèse se concentre sur la prise en charge de l'hétérogénéité, un défi majeur dans le domaine de l'apprentissage fédéré. L'hétérogénéité se manifeste par des variations entre les ensembles de données locaux des clients (hétérogénéité statistique), des disparités dans les capacités de stockage et de calcul (hétérogénéité système), et des fluctuations dans les ensembles de données locaux au fil du temps (hétérogénéité temporelle). Cette thèse explore différentes sources d'hétérogénéité dans le contexte de l'apprentissage fédéré et propose des algorithmes pratiques pour atténuer l'impact de l'hétérogénéité.

La première partie de la thèse se concentre sur la résolution des défis associés à l'hétérogénéité du système dans deux scénarios distincts : inter-silos et inter-appareils. Dans les environnements inter-silos, nous exploitons la théorie des systèmes linéaires dans l'algèbre max-plus pour modéliser le débit, c'est-à-dire le nombre de cycles complets par unité de temps, dans un système d'apprentissage fédéré entièrement décentralisé en inter-silos. Ensuite, nous proposons des algorithmes pratiques qui, en utilisant les caractéristiques mesurables du réseau, trouvent une topologie avec le débit le plus élevé ou avec des garanties de débit vérifiables. Dans les environnements inter-appareils, où les contraintes du système influencent la disponibilité et l'activité des clients, nous explorons différents niveaux de participation des clients, souvent présentant une corrélation au fil du temps et avec d'autres clients. Dans ce contexte, nous analysons un algorithme similaire à  $FedAvg$  sous une disponibilité hétérogène et corrélée des clients. L'analyse met en évidence comment la corrélation affecte négativement le taux de convergence de l'algorithme et comment la stratégie d'agrégation peut atténuer cet effet, même au prix de diriger l'entraînement vers un modèle biaisé. Guidé par l'analyse théorique, nous proposons "Correlation-Aware FL" ( $CA-Fed$ ), un nouvel algorithme FL qui tente d'équilibrer les objectifs contradictoires de maximiser la vitesse de convergence et de minimiser le biais du modèle. À cette fin,  $CA-Fed$  ajuste dynamiquement le poids attribué à chaque client et peut ignorer les clients avec une faible disponibilité et une forte corrélation.

La deuxième partie traite de l'hétérogénéité statistique grâce à deux algorithmes de personnalisation. Le premier algorithme, appelé  $FedEM$ , repose sur une hypothèse souple selon laquelle l'ensemble de données de chaque client est généré à partir d'un mélange de distributions sous-jacentes communes inconnues. Le deuxième algorithme, appelé  $kNN-Per$ , combine un modèle global entraîné collectivement avec un modèle local de plus proches voisins ( $kNN$ ) pour la personnalisation. Des garanties théoriques, notamment des bornes de convergence et de généralisation, sont fournies pour les deux algorithmes.

La troisième partie explore l'apprentissage fédéré pour les flux de données, en considérant deux scénarios : des échantillons indépendants tirés d'une distribution inconnue et des distributions de données composées de mélanges de distributions sous-jacentes inconnues. Pour le premier scénario, un meta-algorithme est proposé, offrant des informations sur la configuration et le compromis entre le temps d'entraînement et le biais du modèle appris. Pour le deuxième scénario, une variante fédérée de la descente du miroir séquentielle appelée  $FEM-OMD$  est introduite, avec un regret asymptotiquement sous-linéaire dans le cas des modèles de mélange Gaussien.

**Mots-clés :** Apprentissage fédéré, Personnalisation, Apprentissage séquentiel, Optimisation distribuée.