Robust Machine Learning Approaches to Wireless Communication Networks

Matteo Zecchin

SORBONNE UNIVERSITÉ

EURECOM
Sophia Antipolis

Robust Machine Learning Approaches to Wireless Communication Networks

Dissertation

submitted to

Sorbonne Université

in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Author:

Matteo Zecchin

Publicly defended on the 15th of December 2022, before a committee composed of:

Reviewers

Prof.	**Carlo Fischione**	KTH Royal Institute of Technology, Sweden
Prof.	**Petar Popovski**	Aalborg University, Denmark

Examiners

Dr.	**Jakob Hoydis**	NVIDIA Research, France
Dr.	**Monica Navarro**	CTTC, Spain
Prof.	**Andrea Zanella**	University of Padova, Italy
Prof.	**Motonobu Kanagawa**	EURECOM, France

Thesis Advisor

Prof.	**David Gesbert**	EURECOM, France

Thesis Co-advisor

Prof.	**Marios Kountouris**	EURECOM, France

SORBONNE UNIVERSITÉ

EURECOM
Sophia Antipolis

Approches robustes d'apprentissage automatique pour les réseaux de communication sans fil

Thèse

soumise à

Sorbonne Université

pour l'obtention du grade de docteur

présentée par:

Matteo Zecchin

Soutenance de thèse effectuée le 15 Décembre 2022 devant un jury composé de:

Rapporteurs

Prof.	**Carlo Fischione**	KTH Royal Institute of Technology, Sweden
Prof.	**Petar Popovski**	Aalborg University, Denmark

Examinateurs

Dr.	**Jakob Hoydis**	NVIDIA Research, France
Dr.	**Monica Navarro**	CTTC, Spain
Prof.	**Andrea Zanella**	University of Padova, Italy
Prof.	**Motonobu Kanagawa**	EURECOM, France

Directeur de thèse

Prof.	**David Gesbert**	EURECOM, France

Co-directeur de thèse

Prof.	**Marios Kountouris**	EURECOM, France

Ai miei genitori, Gabriella e Maurizio

Abstract

Artificial intelligence (AI) is widely viewed as a key enabler of sixth generation (6G) wireless systems. The main drivers of its adoption are the increasing complexity and specialization of the services offered by wireless networks to end users. The original premise towards the incorporation of AI in 6G networks is the possibility of establishing a mutually gainful synergy between wireless communication systems, and the tools belonging to the machine learning (ML) literature. Specifically, the edge of wireless networks caters an unprecedented data availability and computational power that ML algorithms can potentially tap into. At the same time, a plethora of wireless networking problems lacking analytical solutions can benefit from data-driven techniques belonging to the image and audio signals processing domain. This thesis targets fundamental problems arising in this domain, with the end goal of paving the way towards the adoption of reliable AI in future wireless networks.

The first part of this thesis is devoted to wireless communication for ML. It focuses on the development of distributed training algorithms that can be deployed at the edge of wireless networks to fully harness its potential. Future wireless networks are envisioned to be heavily reliant on device-to-device (D2D) communication. For that reason, we first investigate the implication of performing distributed optimization of ML models over wireless communication systems comprising unreliable computing devices restrained to intermittent peer-to-peer connectivity. We propose and formally analyze an implementation of distributed stochastic gradient descent that leverages asynchronous model updates and a time-varying consensus strategy to mitigate the detrimental effect of computational and communication impairments. While being in principle a challenge, we demonstrate that D2D communication brings a new degree of flexibility to the network infrastructure that can be exploited to speed up the training of ML models at the network edge. Specifically, given the increasingly important role of unmanned aerial vehicles (UAVs) in infrastructureless wireless networks, we introduce a UAV-aided training procedure in which the UAV trajectory is designed to promote the diffusion of locally optimized models across devices.

In the second part of this thesis, we switch focus to learning aspects associated with the distributed nature of the data generation processes, in particular data heterogeneity. Data heterogeneity entails the problem of producing ML models capable of generalizing to multiple and different data sources. We consider two different approaches to attain this desideratum. Our first solution is a user-centric federated learning protocol that sidesteps the issue of finding a universal ML solution by outputting tailored models

for different groups of devices that have similar learning goals.We then recommend an alternative approach, based on a distributionally robust reformulation of the learning problem, that has the goal of producing a unique and fair ML model with satisfactory worst-case performance. To achieve this, we develop an agnostic decentralized gradient descent-ascent algorithm that solves the underlying minimax optimization problem in a communication-efficient manner by employing a compressed consensus scheme.

In the third and final part of this thesis, we turn to the paradigm of ML for wireless communication. We take a critical look at frequentist learning and its applications to wireless communication problems. This stance is motivated by the unreliability of the frequentist framework under the challenging learning conditions that characterize wireless communication problems. The main contribution of this section is a novel robust Bayesian learning paradigm, that concurrently counteracts three prominent challenges arising in wireless communication learning: data scarcity, the presence of outliers and model misspecification. Finally, after over-viewing its main theoretical underpinnings and formally investigating its properties, we showcase the merits of the proposed robust Bayesian learning over a range of prototypical wireless communication problems.

Résumé

L'intelligence artificielle (IA) est largement considérée comme un élément fondamental des technologies de communication sans fil de sixième génération (6G). La hausse de la complexité et de la spécialisation des services offerts par les réseaux sans fil aux utilisateurs ont été des facteurs déterminants derrière son utilisation répandue. L'intégration de l'IA dans les réseaux 6G est motivée par la possibilité d'établir une relation mutuellement bénéfique entre les systèmes de communication sans fil et les outils appartenant à la littérature sur l'apprentissage automatique (AA). Plus précisément, la périphérie des réseaux sans fil offre une disponibilité de données et une puissance de calcul sans précédent que les algorithmes d'AA peuvent exploiter. En parallèle, une multitude de problèmes liés aux réseaux sans fil, pour lesquels il n'existe pas de solution analytique, peuvent bénéficier des techniques d'AA appartenant au domaine du traitement des images et des signaux audio. Cette thèse vise à résoudre les problèmes fondamentaux liés à ce domaine, afin de faciliter l'adoption d'une IA fiable dans les futurs réseaux sans fil.

La première partie de cette thèse est consacrée à la communication sans fil pour l'AA. Elle se concentre sur le développement d'algorithmes d'apprentissage distribués qui peuvent être déployés à la périphérie des réseaux sans fil afin d'en exploiter pleinement le potentiel. En effet, il est prévu que les futurs réseaux sans fil soient fortement tributaires de la communication de dispositif à dispositif (D2D) dans un futur proche. Pour cette raison, nous étudions l'optimisation distribuée des modèles d'AA sur les systèmes de communication sans fil comprenant des dispositifs informatiques non fiables limités à un système pair-à-pair intermittent. Nous proposons et analysons un algorithme de la descente de gradient stochastique distribuée qui exploite les mises à jour asynchrones des modèles et une stratégie de consensus variable dans le temps afin d'atténuer l'effet indésirable des dysfonctionnements en matière de calcul et de communication. Nous montrons ensuite que, même si la communication D2D est en principe un obstacle, elle apporte un nouveau degré de flexibilité à l'infrastructure du réseau qui peut être exploitée pour accélérer l'entraînement des modèles d'AA à la périphérie du réseau. Plus précisément, compte tenu du rôle de plus en plus important que tiennent les drones dans les réseaux sans fil ad hoc, nous proposons une procédure d'apprentissage assistée par drone dans laquelle la trajectoire de ce dernier est conçue pour favoriser la diffusion de modèles optimisés localement dans le réseau.

Dans la deuxième partie de cette thèse, nous nous concentrons sur les aspects d'apprentissage associés à la nature distribuée des processus de génération des données, et en particulier à l'hétérogénéité des données. L'hétérogénéité des données pose le

problème de la production de modèles d'AA avec une bonne capacité de généralisation sur des sources de données multiples et différentes. Nous proposons deux approches différentes pour atteindre ce desideratum. Notre première solution est une méthode d'apprentissage fédéré centrée sur l'utilisateur. Celle-ci contourne le problème de la recherche d'une solution d'AA universelle en produisant des modèles sur mesure pour différents groupes de dispositifs ayant des objectifs d'apprentissage similaires. Nous suggérons par la suite une approche alternative, basée sur une reformulation distributionnellement robuste du problème d'apprentissage, qui a pour but de produire un modèle d'AA unique et éthique avec une performance satisfaisante dans le pire des cas pour tous les dispositifs collaboratifs. Pour y parvenir, nous développons un algorithme de descente de gradient décentralisé agnostique qui résout le problème d'optimisation minimax sous-jacent d'une manière efficace en termes de communication en utilisant un schéma de consensus compressé.

Dans la troisième et dernière partie de cette thèse, nous nous tournons vers le paradigme de l'AA pour les communications sans fil. Nous jetons un regard critique sur l'apprentissage fréquentiste et son utilisation sur les problèmes de communication sans fil. Cette prise de position est motivée par le manque de fiabilité que le paradigme fréquentiste démontre lors de conditions d'apprentissage difficiles caractérisant les problèmes de communication sans fil. La principale contribution de cette section est un nouveau paradigme d'apprentissage bayésien robuste qui relève simultanément trois défis proéminents dans l'apprentissage des communications sans fil : la rareté des données, la présence de données aberrantes et la mauvaise spécification du modèle.Enfin, après avoir passé en revue les principaux fondements théoriques de l'apprentissage bayésien robuste et avoir étudié formellement ses propriétés, nous démontrons ses mérites sur une série de problèmes prototypiques de communication sans fil.

Acknowledgements

First, I would like to express my gratitude to David Gesbert and Marios Kountouris. Their crystalline view on problems helped me move my first steps in the world of research, and their constant support provided me with opportunities for which I am extremely grateful.

I want to also thank EURECOM and the great researchers that I had the chance to collaborate with. In particular, I would like to thank the past and present members of the M3 group, and Davit, Maurizio and Motonobu from the data science department.

I am very grateful to professor Osvaldo Simeone for welcoming me to his research group at King's College. During these six extraordinary months, I had the chance to meet fantastic new collaborators and friends. Thanks, Sangwoo, Mikolaj, Riccardo, Ivana, Nicolas, Sharu, Kfir, Yunchuan, Hari and Clement for all the good times spent in London and at The Vault!

I would like to thank the European Union for funding my research, and all the members of the ITN Windmill with whom I shared lots of unforgivable experiences.

I would also like to thank my friends that backed me up from Italy, Grazie! And all the ones that I have met here in France, Merci!

Last but not least, I would like to thank my family and Alexane for their unconditional support and love.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

5G Fifth Generation

6G Sixth Generation

B5G Beyond Fifth Generation

AI Artificial Intelligence

ML Machine Learning

D2D Device to Device

P2P Peer to Peer

UAV Unmanned Aerial Vehicle

DSGD Distributed Stochastic Gradient Descent

SGD Stochastic Gradient Descent

GDA Gradient Descent Ascent

AD-GDA Agnostic Decentralized Gradient Descent Ascent

CML Centralized Machine Learning

DML Distributed Machine Learning

IoT Internet of Things

BS Base Station

PS Parameter Server

AIV Autonomous Intelligent Vehicle

FL Federated Learning

RL Reinforcement Learning

IRS Intelligent Reflecting Surface

MS-FL Multi Stage Federated Learning

LoS Line-of-Sight

NLoS Non-Line-of-Sight

AirComp Over-the-air Computation

UL Uplink

DL Downlink

DRL Distributionally Robust Learning

AFL Agnostic Federated Learning

DRFA Distributionally Robust Federated Averaging

FedAvg Federated Averaging

CFL Clustered Federated Learning

EM Expectation Maximization

IID Independent and Identically Distributed

MAML Model-Agnostic Meta-Learning

Chapter 1

Introduction

As the fifth generation (5G) network roll out is ramping up around the world, research on sixth generation (6G) mobile systems is expected to deliver technological advancements that are able to sustain increasing demands for massive connectivity, increased reliability, reduced latency, while at the same time satisfying imperative energy-efficiency requirements [7, 8].

These, apparently contrasting, needs bring about complex engineering problems that frequently lack models that are concurrently well-descriptive and analytically tractable. Researchers have then considered complementing the classical model-based design with the data-driven one [9,10]. The major underpinning for this paradigm shift is the adoption of machine learning (ML) solutions. ML allows to sidestep the challenges of modelling and solving complex wireless communication problems by relying upon data-driven solutions obtained from the optimization of parametric models, i.e. neural networks, based on large amounts of data. This technology is expected to be sustained by the enormous data availability and computational power provided by 6G networks, and to penetrate at all the levels of the protocol stack [11].

The interaction between ML and wireless communication is not limited to an application of the tools of the former field to the problems arising in the latter. In fact, 6G networks also bears an opportunity to scale machine learning technologies up to an unprecedented level. Massive connectivity, combined with the advent of Internet of Things (IoT), will provide a new sheer amount of data and it will contribute to produce one of the largest and most powerful distributed computing platforms available [12, 13]. This opportunity calls for a novel network design that, instead of serving as a simple pipeline for data, can support distributed ML at the edge of the network.

Overall, form the intersection between ML and wireless communication stem two different and complementary research fields: *wireless communication for machine learning*, focusing on repurposing wireless communication systems for distributed training of ML models, and *machine learning for wireless communication*, mapping the tools from the ML literature to the problems emerging from the development of 6G networks.

1.1 Wireless Communication for Machine Learning

The current spring of AI has been fueled by the availability of powerful computing frameworks and the big data revolution [14]. These two ingredients are traditionally leveraged following the centralized machine learning (CML) paradigm. Accordingly, the training data is collected at a single processing unit, or a cluster of processing units interconnected by wired links, and the optimization process is run locally. However, as the amount of smartphones and IoT devices soar, data has started being generated in a distributed fashion at the edge of the network by increasingly powerful devices. In principle any distributed data sets can be collected at a central node and processed according to the CML framework; however, the volume of the data generated at the network edge, the unreliability of its wireless link and the privacy concerns associated to off-loading personal data, poses an insurmountable obstacle to the application of CML in 6G networks. This limitation of the CML approach motivates the search for novel communication protocols and physical layer technologies to enable efficient distributed machine learning (DML) at the wireless network edge [15].

1.1.1 Distributed Machine Learning

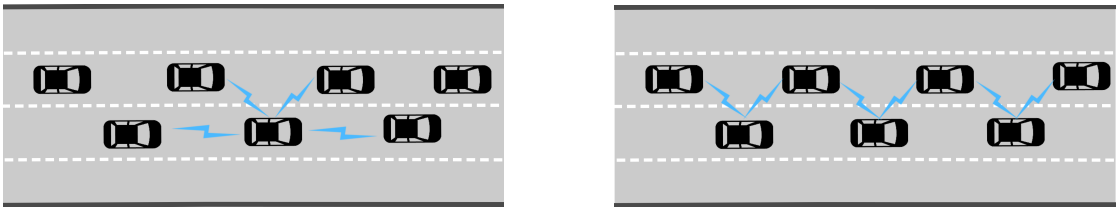
DML is a training technique that allows to scale out the training of ML models. Differently from the scale-up approach, which works by increasing the computational power and storage at a single device, DML parallelizes the optimization of a ML model by leveraging a group of computing nodes while keeping the training data distributed. The practical upshot is that devices with limited data and computing capabilities can aggregate their resources without off-loading sensitive data [16].

The archetypical DML protocol consists in an iterative and coordinated optimization scheme that encompasses multiple rounds, each comprising a computation and communication phase. During the former, a portion of the collaborating devices locally optimize the ML model based on the in-situ data and computing resources. It then follows a communication phase, during which devices share the result of the optimization and a new model is created through the aggregation of the received parameters. This procedure repeats until the validation performance of the ML model stabilizes.

There exists different types of distributed learning frameworks that differentiate depending on the number and reliability of their computing nodes, the way data is distributed, the speed of the communication links and the type of communication topology that interconnects the devices. In Table 1.1 we provide a taxonomy of the most popular DML schemes. The 6G network constitutes the ideal environment for the deployment of DML, in particular of the federated and decentralized strategies which are expected to be able to cope with the massive population of heterogeneous workers [17] (see Table 1.1).

1.1.2 Massive Device to Device Connectivity

According to the report of IoT Analytics of May 2022, the number of IoT devices in 2021 has reached a total of almost 12.2 billions, and this figure is expected to double by the



(a) Vehicle platoon federated learning

(b) Vehicle platoon decentralized learning

Figure 1.1: Comparison between the communication topology induced over a platoon of smart vehicles by the federated (left) and the decentralized learning protocols (right). Federated Learning cannot harness the full platoon resources when constrained to one-hop communication with the orchestrator. Decentralized learning allows to connect the entire platoon exploiting short range device-to-device links.

end of 2025 [18]. Similarly, the number of smartphones is steadily increasing since their advent [19].

Connectivity for this vast amount of devices has been traditionally provided using the cellular networks paradigm, with the base station (BS) serving all the devices within its coverage area. While this network design is compatible with the federated learning (FL) and the multi-stage federated learning (MS-FL) protocols, the star communication topology is affected by an inherent communication bottleneck hindering massive connectivity [20, 21]. To cope with this enormous amount of connected devices, the standard cellular design is then expected to be complemented by more flexible communication topologies based on device-to-device (D2D) communication [22, 23]. For this new communication paradigm, decentralized learning protocols are envisioned to play a crucial role in virtue of the absence of parameter servers (PS) and their flexibility with respect to the underlying communication topology. In fact, the communication phase of decentralized learning protocols requires nodes to communicate in a P2P fashion with other devices that are within range, and to perform aggregation locally [24, 25].

To illustrate the advantages of decentralized learning in D2D network deployments, consider the scenario in Figure 1.1. A platoon of autonomous intelligent vehicles (AIV) wish to collaboratively train a ML model relying upon the short-range D2D links. Federated learning can be applied, by either resorting to a BS at the side of the road or by defining a vehicle as a PS. The first solution is often infeasible due to the mobility of the platoon, the second limits the number of collaborative devices to the AIVs that are within range of the PS, unless one allows multi-hop links. In contrast, decentralized learning enables the platoon to organize in a line topology and to harness the entirety of the resource using only the available D2D links.

1.1.3 Challenges

From the above discussion it is clear the essential nature of decentralized learning protocols for the future smart edge. However, the application of DML to 6G communication systems brings about fundamental challenges that have to be addressed in order to scale out

ML and unleash the full potential of future IoT networks. The first challenge is the communication bottleneck introduced by unreliable wireless communication links. As the computing hardware develops, the communication phase of DML protocols becomes the most time-consuming step [26]. Devices can also become unavailable for long periods, e.g. to perform additional tasks other than training, causing delays or biases in the training results [27]. Furthermore, being an iterative procedure, the training of ML models can require numerous rounds of communication and large energy expenditures that are not always affordable by edge IoT devices [28]. These aspects related to the application of DML to unreliable wireless networks will be the focus of the first part of this thesis.

Another fundamental challenge derives from the collaboration among network devices that are heterogeneous. In fact, the 6G smart edge is expected to comprise devices with different sensing capabilities that sample processes influenced by geographical or user dependent factors [29]. Learning theory states that the aggregation of different data sources can heavily hinder the quality of the final model at testing time [30]. Therefore, decentralized learning procedures have to be carefully designed in order to make collaboration fruitful rather than detrimental. The development of such decentralized learning algorithms will constitute the content of the second part of this thesis.

1.2 Machine Learning for Wireless Communications

The design of 5G and previous generations networks has been reliant on mathematical models of communication systems, with the role of real-world data being integrative and limited to the fine-tuning of their parameters. However, the increased level of complexity and flexibility of 6G communication systems have rendered the model-based design ineffective [10]. The extent to which 6G networks will leverage complex physical layer technologies and adapt communication protocols based on user-centric and contextual information makes the mathematical description of communication models challenging [31]. As such, the role of data and ML algorithms has become prominent in the design of future wireless networks as it allows to bypass explicit problem formulations. This paradigm shift allows us to obtain solutions to wireless communication problems in a model-free fashion, leveraging large collections of network measurements and by optimizing expressive parametric models.

1.2.1 Complexity of Future Network Services

In virtue of its versatility, ML is envisioned to be employed at all layers of the wireless protocol stack.

Specifically, data-driven algorithms can reproduce the output of complex iterative optimization procedures at a smaller computational cost. For example, new physical layer technologies, such as intelligent reflecting surface (IRS) and millimeter wave communication, lead to complex optimization problems [32–35]. In these cases, classical solvers bear a large computational cost which is incompatible with 6G low-latency requirements. On the other hand, ML techniques such as deep unfolding are capable of efficiently providing

solutions to many high-dimensional signal processing problems by the means of a single forward pass [36, 37].

Similarly, the cross-layer design of communication protocols gives rise to complicated resource allocation problems. In this context, reinforcement learning (RL) has demonstrated its ability to cope with the large action space characterizing these sequential decision-making processes, and to cut down the delays of block-wise optimized solutions [38, 39].

At the same time, data-driven methods can also help tackle in a novel way problems that were out-of-reach or that were solved using costly iterative search procedures. Considering for example the case of millimeter wave beam alignment, ML can take advantage of high-dimensional contextual information to greatly reduce the search space compared to classical solutions [40, 41]. The data-driven design can also enable accurate localization and sensing with massive antenna arrays by extracting spatial information from high-dimensional channel fingerprints [42, 43]. These are key building blocks for context-aware networking protocols for which the model-based paradigm cannot provide general, yet adaptable, solutions. At the higher layer of the protocol stack, ML can be used to extract user-centric features from the application data and provide personalized services [44, 45].

1.2.2 Uncertainty Quantification and Robustness in Wireless Systems

As we have seen, ML naturally finds application in many wireless communication problems; however, in these scenarios, standard figures of merit such as the prediction accuracy have to be weighted against other performance indicators that are specific to communication systems.

The first is uncertainty quantification; namely, the ability of an ML model to faithfully quantify the uncertainty of its outputs [46]. This capability is essential in safety-critical decision-making processes, e.g. real-time control via the tactile internet [47], and it can be used to enhance the network performance, e.g. in the context of cognitive radio to adopt conservative behaviour upon uncertain spectrum sensing outcomes [48]. Uncertainty quantification also lays the foundation of network self-monitoring [49]. ML communication systems with good uncertainty quantification capabilities can detect the deterioration of their performance (low confidence) and trigger timely retraining of its modules, for example when the operating condition mutates.

A second important prerequisite for ML to be applied to wireless communication systems is the robustness against mismatches between the design assumptions and the real-world operating conditions [50–52]. When deployed in wireless communication systems, ML models need to be operational with little to no human intervention over a variety of application scenarios that are often outside the designer's control. As a result, data-driven solutions to communication problems are obtained based on model assumptions that frequently do not entail the testing condition, and therefore learning usually happens based on model classes that poorly approximate the phenomenon of interest. This learning condition, termed model misspecification, is known to greatly hamper the performance of ML solutions, and it renders the search for robust ML

algorithms extremely important in wireless communication problems [3].

1.2.3 Challenges

The major major obstacle to the application of ML solutions in 6G networks are the strict ML reliability requirements and the adverse learning conditions characterizing wireless communication problems.

In wireless systems, data generation processes often have short stationary intervals that impose strict upper bounds on the length of data acquisition procedures and the size of training data sets. In the limited data regime, the frequentist learning approach is known to perform poorly and lead to over-confident predictors [1]. Therefore, in spite of its popularity, the application of frequentist ML solutions in wireless communication systems is incompatible with most of the ML requirements.

On the other hand, the Bayesian learning approach provides a mathematically grounded framework to reason about epistemic uncertainty, the uncertainty due to the limited amount of data [53, 54]. This merit of the Bayesian framework is promising for the application of ML in 6G. However, the uncertainty quantification properties of Bayesian learning are reliant on two fundamental assumptions: the model class is well specified and the training data distribution matches the testing one. These two conditions are often violated in wireless communication systems. Strict energy efficiency and computation complexity requirements require the usage of simple models that are often rough approximations of the complex real-world phenomena the designer wishes to model. In this condition, the Bayesian learning rule does not retain good uncertainty quantification capabilities and is incapable of delivering calibrated models [3, 52, 55].

Additionally, the data collection procedures in realistic wireless systems are autonomous with little or no human interventions. Therefore, in stark contrast with the computer science domain, learning in wireless systems happens by the means of data sets that are small and frequently corrupted by outliers introduced by exogenous noise sources, and by malicious or inaccurate reporting [56, 57]. This mismatch between training and testing conditions is known to greatly degrade the performance of ML models.

The above limitations of the frequentist and standard Bayesian framework motivate the last part of this thesis and the development of a robust Bayesian framework able to concurrently counteract model misspecification and outliers.

1.3 Contributions and Thesis Outline

This thesis work comprises three separate parts, each focusing on particular challenges resulting from the integration of machine learning algorithms and wireless communication.

In Part I, we consider the challenges and opportunities deriving from the application of decentralized learning in D2D networks. In particular, in Chapter 2 we propose an implementation of the distributed stochastic gradient descent (DSGD) algorithm that is designed to cope with the inherent communication and computation impairments characterizing the edge of the wireless networks. The proposed algorithm leverages asynchronous updates and time-varying consensus strategies that allow it to tackle both

the presence of straggling computing nodes and unreliable communication links. For the proposed algorithm, we derive a convergence guarantee for non-convex objectives, which allows us to quantify the impact of key network performance indicators on the convergence properties of the algorithm. The main takeaway of this section is that asynchronicity can speed-up training in spite of the aforementioned network impairments. In Chapter 3 we consider the bright side of D2D connectivity, and we explore the role of UAVs as potential promoters of edge intelligence. We show that the sparse and local connectivity of IoT networks, which potentially hinder decentralized learning, can be mitigated by a UAV relay. We derive an optimized trajectory for the UAV that speeds up training by promoting the diffusion of the locally optimized model across the network.

This first part is based on the papers:

- E. Jeong[†], M. Zecchin[†], and M. Kountouris, "Asynchronous Decentralized Learning over Unreliable Wireless Networks," ICC 2022 - IEEE International Conference on Communications, 2022.
- M .Zecchin, D. Gesbert, and M. Kountouris, "UAV-Aided Decentralized Learning over Mesh Networks." EUSIPCO 2022 - European Signal Processing Conference, 2022.

In Part II, we focus on critical learning aspects deriving from the heterogeneity of the data generated at the edge. Specifically, in Chapter 4 we consider a D2D IoT network with devices sampling data from different processes, in turn leading to differently distributed local training data sets. In this scenario, we consider the task of producing a ML model that guarantees satisfactory performance for all devices. To this end, we formulate a decentralized distributionally robust problem and we propose AD-DGA, a decentralized learning algorithm to solve the associated minimax optimization problem in a communication-efficient manner. We establish non-asymptotic convergence guarantees both in the case of convex and non-convex objectives. The theoretical results are corroborated by experimental results highlighting the merits of the distributionally robust learning procedure.

In Chapter 5 we address statistical heterogeneity by adopting a different approach based on personalization. Starting from theoretical results derived from the domain adaptation literature, we replace the standard federated learning aggregation rule with a set of user-centric ones that serve groups of statistically homogeneous users. Each user-centric aggregation rule produces a model that is tailored for the target distribution associated with the group of users. Tuning the number of personalized rules allows trading personalization for communication resources. We show that the optimal number of user-centric rules can be obtained using clustering techniques. Our algorithm is shown to outperform state-of-the-art solutions both in terms of personalization capabilities and communication-efficiency.

This part is based on the works:

- M. Zecchin, M. Kountouris and D. Gesbert, "Communication-Efficient Distributionally Robust Decentralized Learning", submitted to Transactions on Machine Learning Research.

- M. Mestoukirdi†, M. Zecchin†, D. Gesbert, Q. Li, and N. Gresset. "User-Centric Federated Learning." 2021 IEEE Globecom Workshops (GC Wkshps), 2021.

Finally in Part III, we focus on the application of machine learning to wireless communications problems. We first highlight the fundamental challenges characterizing this application domain; namely, small training data sets affected by outliers and misspecified model classes. These learning conditions render the frequentist learning rule inadequate to deliver reliable models with good uncertainty quantification capabilities in 6G systems. Therefore, the main contribution of this part is the development of the (m, t)-Bayesian learning framework, an extension of the generalized Bayesian inference that is robust to outliers and misspecified model classes. The proposed learning rule is shown to enjoy advantageous theoretical properties and to provide better ML models on a range of supervised and unsupervised wireless communication problems.

This final part of the thesis is based on the papers:

- M. Zecchin, S. Park, O. Simeone, M. Kountouris and D. Gesbert, "Robust PAC^m : Training Ensemble Models Under Model Misspecification and Outliers", submitted to IEEE Transactions on Neural Networks and Learning Systems.

- M. Zecchin, S.Park, O. Simeone, M. Kountouris and D. Gesbert, "Robust Bayesian Learning for Reliable Wireless AI: Framework and Applications", submitted to IEEE Transactions on Cognitive Communications and Networking.

† indicates equal contribution

Table 1.1: Taxonomy of distributed machine learning (DML) paradigms.

DML Paradigm	**Data-center Learning**	**Cross-silo Federated Learning**	**Federated Learning**	**Multi-stage Federated Learning**	**Decentralized Learning**
# workers	$[10, 10^3]$	$[10, 10^2]$	$[10^2, 10^5]$		
Worker type	High performance computers.		Heterogeneous set of IoT devices and smartphones.		
Worker reliability	Workers are always available for computations.		Workers can become inactive or be dropped.		
Type of links	Reliable high-speed links.		Unreliable wireless links.	Wireless and wired links.	Wireless D2D links.
Communication topology	Star topology with a central orchestator.			Tree network with multiple orchestators.	General topology without orchestator.

Part I

Decentralized Learning over the Edge

Chapter 2

Asynchronous Decentralized Learning over Unreliable Wireless Networks

Decentralized learning enables edge users to collaboratively train models by exchanging information via device-to-device (D2D) communication. Prior works have been limited to the analysis of the performance of these algorithms over wireless networks with fixed topologies and reliable workers, which do not resemble realistic sixth generation (6G) network deployments. In this Chapter, we propose an asynchronous decentralized stochastic gradient descent (DSGD) algorithm, which is robust to the inherent computation and communication failures occurring at the wireless network edge. We theoretically analyze its performance and establish a non-asymptotic convergence guarantee. Experimental results corroborate our analysis, demonstrating the benefits of asynchronicity and outdated gradient information reuse in decentralized learning over unreliable wireless networks.

2.1 Introduction

Distributed learning algorithms empower devices in wireless networks to collaboratively optimize the model parameters by alternating between local optimization and communication phases. Leveraging the aggregated computational power available at the wireless network edge in a communication efficient [58] and privacy preserving manner [59], distributed learning is considered to be a key technology enabler for future intelligent networks. A promising paradigm, which enables collaborative learning among edge devices communicating in a peer-to-peer (P2P) manner, is decentralized learning [60]. Differently from federated learning, decentralized algorithms do not require a star topology with a central parameter server (PS), thus being more flexible with respect to the underlying connectivity [61]. This feature renders decentralized learning particularly appealing for future wireless networks with D2D communication. Several decentralized learning schemes over wireless networks have been proposed and analyzed [24, 25, 62, 63], highlighting the key role of over-the-air computation (AirComp) [64] for low-latency training at the edge. Prior works have mainly considered wireless networks of reliable workers communicating in a fixed topology throughout the entire training procedure. Nevertheless,

these assumptions are hardly met in practical systems, in which communication links can be intermittent or blocked, and devices may become temporarily unavailable due to computation impairments or energy saving reasons. Asynchronous distributed training has been shown to mitigate the effect of stragglers (slow workers) [65–67]. However, harnessing the potential benefits of asynchronism in decentralized learning over unreliable wireless networks remains elusive.

In this chapter, we propose an asynchronous implementation of decentralized stochastic gradient descent (DSGD) as a means to address the inherent communication and computation impairments of heterogeneous wireless networks. In particular, we study decentralized learning over a wireless network with a random time-varying communication topology, comprising unreliable devices that can become stragglers at any point of the learning process. To account for communication impairments, we propose a consensus strategy based on time-varying mixing matrices determined by the instantaneous network state. At the same time, we design the learning rates at the edge devices in such a way so as to preserve the stationary point of the original network objective in spite of the devices' heterogeneous computational capabilities. Finally, we provide a non-asymptotic convergence guarantee for the proposed algorithm, demonstrating that decentralized learning is possible even when outdated information from slow devices is used to locally train the models. Experimental results confirm our analysis and show that reusing stale gradient information can speed up convergence of asynchronous DSGD.

2.2 System Model

We consider a network consisting of m wireless edge devices, in which each node i is endowed with a local loss function $f_i : \mathbb{R}^d \to \mathbb{R}$ and local parameter estimate $\theta_i \in \mathbb{R}^d$. The network objective consists in minimizing the aggregate network loss subject to a consensus constraint

$$\begin{aligned} \underset{\theta_1,\dots,\theta_m}{\text{minimize}} \; & f(\theta_1,\dots,\theta_m) := \frac{1}{m}\sum_{i=1}^{m} f_i(\theta_i) \\ \text{s.t.} \quad & \theta_1 = \theta_2 = \cdots = \theta_m. \end{aligned} \tag{2.1}$$

This corresponds to the distributed empirical risk minimization problem whenever f_i is a loss term over a local dataset. In the following, we denote the network objective evaluated at a common parameter vector θ as

$$f(\theta) := f(\theta_1,\dots,\theta_m)\big|_{\theta_1=\cdots=\theta_m=\theta}, \tag{2.2}$$

and the mean parameter vector as

$$\bar{\theta} = 1/m \sum_{i=1}^{m} \theta_i. \tag{2.3}$$

To solve (2.1), we consider a DSGD algorithm according to which devices alternate between a local optimization based on gradient information (computation phase) and a communication phase.

2.2.1 Computation model

To locally optimize the model estimate θ_i, we assume that each device can query a stochastic oracle satisfying the following properties.

Assumption 1. *At each node i, the gradient oracle $g_i(\theta)$ satisfies the following properties for all $\theta \in \mathbb{R}^d$*

$$\mathbb{E}[g_i(\theta)] = \nabla_\theta f_i(\theta) \qquad \textit{(unbiasedness)} \tag{2.4}$$

$$\mathbb{E}\|g_i(\theta) - \nabla_\theta f_i(\theta)\|^2 \leq \sigma^2 \qquad \textit{(bounded variance)} \tag{2.5}$$

$$\mathbb{E}\|g_i(\theta)\| \leq G^2. \qquad \textit{(bounded magnitude)} \tag{2.6}$$

We admit the existence of straggling nodes and that a random subset of devices can become inactive or postpone local optimization procedures, e.g., due to computation impairments or energy constraints. As a result, devices may join the communication phase and disseminate a model that has been updated using gradient information computed using previous model estimates, or a model that has not been updated at all from the previous iteration(s). Formally, at every optimization round t, the local update rule is

$$\theta_i^{(t+\frac{1}{2})} = \begin{cases} \theta_i^{(t)}, & \text{if device } i \text{ is straggler at round } t \\ \theta_i^{(t)} - \eta_i^t g_i(\theta^{(t-\tau_i)}), & \text{otherwise} \end{cases} \tag{2.7}$$

where η_i^t is a local learning rate and the delay $\tau_i \geq 0$ accounts for the staleness of the gradient information at device i.

2.2.2 Communication model

The channel between any pair of device i and j follows a Rayleigh fading model. At every communication iteration t, devices can exchange information according to a connectivity graph $\mathcal{G}^{(t)} = (\mathcal{V}, \mathcal{E}^{(t)})$, where $\mathcal{V} = \{1, 2, \ldots, m\}$ indices the network nodes and $(i, j) \in \mathcal{E}^{(t)}$ if devices i and j can communicate during round t. We consider symmetric communication links; therefore the communication graph is undirected. While the connectivity graph is assumed to remain fixed within the optimization iteration, it may vary across optimization iterations due to deep fading, blockage, and/or synchronization failures.

2.3 Asynchronous Decentralized SGD

The proposed asynchronous DSGD procedure, which takes into account both computation and communication failures, is detailed in Algorithm 1.

At the beginning of each training iteration t, non straggling devices update the local estimate $\theta_i^{(t)}$ according to (2.7) using a potentially outdated gradient information. Subsequently, based on the current connectivity graph $\mathcal{G}^{(t)} = (\mathcal{V}, \mathcal{E}^{(t)})$, devices agree on a symmetric and doubly stochastic mixing matrix $W^{(t)}$ using a Metropolis-Hastings

Algorithm 1: Asynchronous Decentralized SGD

Input : Number of devices m, number of iterations T, learning rates η_θ and $\theta_i^{(0)} \in \mathbb{R}^d$.
Output : $\bar{\theta}^{(T)} = \frac{1}{T}\sum_{t=0}^{T-1} \bar{\boldsymbol{\theta}}^t$

```
for t in 0, ... T − 1 do
    for each non straggling devices do
        update local model as (2.7)
    end
    Determine matrix W^(t) based on G^(t)
    for s in [1, S_t] do
        if s ≡ 0 (mod 2) then
            // Broadcast phase
            for each device i scheduled in slot s do
                Device i transmits (2.12)
                Each device j ∈ N_i^(t) receives (2.13)
                Each device j ∈ N_i^(t) estimates (2.14)
            end
        else
            // AirComp phase
            for each star center i scheduled in slot s do
                Each device j ∈ N_i^(t) transmits (2.12)
                Device i receives (2.10)
                Device i estimates (2.11)
            end
        end
    end
    for each device do
        model consensus as in (2.15)
    end
end
```

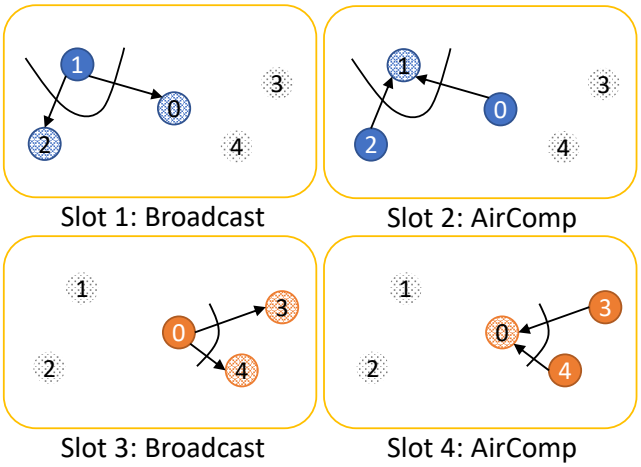


Figure 2.1: An example of the timeline for one training iteration composed of alternate Broadcast and AirComp slots.

weighting scheme [68]. In particular, the (i, j) entry of the mixing matrix $W^{(t)}$ is obtained as

$$w_{i,j}^{(t)} = \begin{cases} \frac{1}{1+\max\left\{d_i^{(t)}, d_j^{(t)}\right\}}, & \text{if } (i,j) \in \mathcal{E}^{(t)} \text{ and } i \neq j \\ 1 - \sum_j w_{i,j}^{(t)}, & \text{if } i = j \\ 0, & \text{otherwise.} \end{cases} \tag{2.8}$$

where $d_i^{(t)}$ is the degree on node i at the communication round t. These weights are very simple to compute and are amenable for distributed implementation. In particular, each device requires only knowledge of the degrees of its neighbors to determine the weights on its adjacent edges.

After that, it follows a communication phase in which devices exchange the updated estimates and employ a gossip scheme based on $W^{(t)}$. To leverage AirComp capabilities, devices employ analog transmission together with the scheduling scheme proposed in [63]. Accordingly, the communication phase is divided into multiple pairs of communication slots. Each pair consists of an *AirComp slot* and a *broadcast slot* as illustrated in Fig. 2.1. During the AirComp slot s, the star center i receives the superposition of the signals transmitted by its neighboring devices $\mathcal{N}^{(t)}(i) = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}^{(t)}\}$. In particular, each scheduled node $j \in \mathcal{N}^{(t)}(i)$ transmits to the star center i

$$x_j^{(s,t)} = \frac{\sqrt{\gamma_i^{(s,t)}}}{h_{i,j}^{(s,t)}} w_{i,j}^{(t)} \theta_j^{(t+\frac{1}{2})} \tag{2.9}$$

where $h_{i,j}^{(s,t)} \in \mathbb{C}^d$ is the channel coefficient between user i and j during slot s, $\gamma_i^{(s,t)} \in \mathbb{R}$ is a power alignment coefficient, and $w_{i,j}^{(t)}$ is the (i,j) entry of the mixing matrix $W^{(t)}$.

The star center i receives the aggregated signal

$$y_i^{(s,t)} = \sum_{j \in \mathcal{N}(i)} h_{i,j}^{(s,t)} x_j^{(s,t)} + z_i^{(s,t)} \tag{2.10}$$

where $z_i^{(s,t)} \sim \mathcal{N}(0, \sigma_w \mathbb{1}_d)$ is a noise vector, and estimates the aggregated model as

$$\hat{y}_i^{(s,t)} = \frac{y_i^{(s,t)}}{\sqrt{\gamma_i^{(s,t)}}} = \sum_{j \in \mathcal{N}(i)} w_{i,j}^{(t)} \theta_j^{(t+\frac{1}{2})} + \frac{z_i^{(s,t)}}{\sqrt{\gamma_i^{(s,t)}}}. \tag{2.11}$$

On the other hand, during a broadcast slot s, scheduled node i transmits using a power scaling factor $\alpha_i^{(s,t)}$ the signal

$$x_i^{(s,t)} = \sqrt{\alpha_i^{(s,t)}} \theta_i^{(t+\frac{1}{2})} \tag{2.12}$$

and all neighboring devices $j \in \mathcal{N}^{(t)}(i)$ receive

$$y_j^{(s,t)} = h_{j,i}^{(s,t)} x_i^{(s,t)} + z_j^{(s,t)} \tag{2.13}$$

and estimate the updated model as

$$\hat{y}_j^{(s,t)} = w_{j,i}^{(t)} \frac{y_j^{(s,t)}}{\sqrt{\alpha_i^{(s,t)}} h_{j,i}^{(s,t)}} = w_{j,i}^{(t)} \left(\theta_i^{(t+\frac{1}{2})} + \frac{z_j^{(s,t)}}{\sqrt{\alpha_i} h_{j,i}} \right). \tag{2.14}$$

At the end of the communication phase, each node i obtains the new estimate $\theta_i^{(t+1)}$ combining all received signals and using a consensus with step size $\zeta \in (0, 1]$

$$\theta_i^{(t+1)} = (1 - \zeta) \theta_i^{(t+\frac{1}{2})} + \zeta \left\{ \sum_{j=1}^{m} w_{i,j}^{(t)} \theta_j^{(t+\frac{1}{2})} + \tilde{n}_i{}^{(t)} \right\} \tag{2.15}$$

where $\tilde{n}_i^{(t)} \sim \mathcal{N}(0, \tilde{\sigma}_{w,i}^{(t)} \mathbb{1}_d)$ is a noise vector term that accounts for the aggregation of noise components during AirComp and broadcast transmissions at device i during communication phase t.

2.4 Convergence Analysis

In this section, we study the effect of communication and computation failures on the asynchronous DGSD procedure and prove its convergence.

2.4.1 Effect of Communication Failures

Communication impairments amount for a random connectivity graph with an edge set that differs at each different optimization iteration. From an algorithmic perspective, random communication impairments result in DSGD with stochastic mixing matrices. A particular class of stochastic mixing matrices are those that satisfy the expected consensus property.

Definition 1 (Expected Consensus Rate [61])**.** *A random matrix $W \in \mathbb{R}^{m\times m}$ is said to satisfy the expected consensus with rate p if for any $X \in \mathbb{R}^{d\times m}$*

$$\mathbb{E}_W \left[\left\| WX - \bar{X} \right\|_F^2 \right] \leq (1-p) \left\| X - \bar{X} \right\|_F^2 \tag{2.16}$$

where $\bar{X} = X\frac{\mathbf{1}\mathbf{1}^T}{m}$ and the expectation is w.r.t. the random matrix W.

Lemma 1. *If the event that the connectivity graph $\mathcal{G}^{(t)}$ is connected at round t has a probability $q > 0$ and the Metropolis-Hastings weighting is used to generated the mixing $W^{(t)}$, the expected consensus rate is satisfied with rate $p = q\delta > 0$, with δ being the expected consensus rate in case of a connected topology.*

Proof. See Appendix A.1. □

If the expected consensus is satisfied, it is then possible to establish a convergent behavior for the estimates generated by the proposed algorithm.

Lemma 2 (Consensus inequality)**.** *Under Assumption 1, after T iterations, decentralized SGD with a constant learning rate η and consensus step size ζ satisfies*

$$\sum_{i=1}^{m} \left\| \theta_i^{(T)} - \bar{\theta}^{(T)} \right\|_2 \leq \eta^2 \frac{12mG^2}{(p\zeta)^2} + \zeta \frac{2}{p} \sum_{i=1}^{m} \sigma_{w,i}^2 \tag{2.17}$$

where $\sigma_{w,i}^2 = \max_{t=0}^{T} \mathbb{E}\left\| \tilde{n}_i^{(t)} \right\|^2$.

Proof. See Appendix A.2. □

Overall, communication failures amount to a reduced expected consensus rate compared to the scenario with perfect communication. At the same time, dropping users that are delayed and are unable to synchronize and perform AirComp, renders the communication protocol more flexible. For instance, in Fig. 2.2, we consider a network of nine nodes organized according to different topologies and show the evolution of the average spectral gap of the mixing matrix with Metropolis-Hastings weights, whenever devices not satisfying a certain delay constraint are dropped. As expected, stricter delay requirements result in sparser effective communication graphs and mixing matrices with smaller spectral gaps.

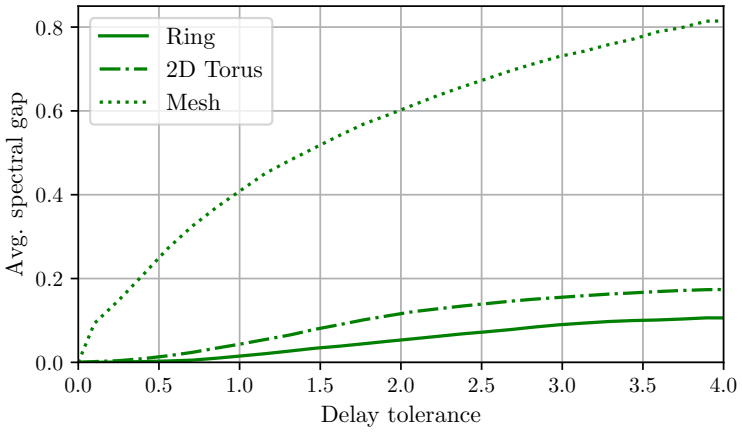


Figure 2.2: Average spectral gap under different delay constraints for mesh, ring, and two-dimensional torus topologies with 9 nodes. Each link is associated to a completion time $\sim Exp(1)$ and is dropped if it exceeds the delay tolerance value.

2.4.2 Effect of Computation Failures

Random computation impairments make the group of devices that effectively update the model parameter vary over time. To account for this in the analysis, we introduce a virtual learning rate that is zero in case of failed computation. Namely, the learning rate at device i during computation round t becomes

$$\tilde{\eta}_i^{(t)} = \begin{cases} 0, & \text{if } i \text{ is straggler at round } t \\ \eta_i^{(t)}, & \text{otherwise} \end{cases} \tag{2.18}$$

where $\eta_i^{(t)}$ is a specified learning rate value in case of successful computation. Furthermore, to ensure that the procedure converges to stationary points of the network objective even when edge devices have different computing capabilities, the expected learning rates have to be equalized. In particular, if $\mathbb{E}[\eta_i^{(t)}] = \eta,\ \forall i$, we have that stationary points are maintained in expectation, namely

$$\sum_{i=1}^{m} \mathbb{E}[\tilde{\eta}_i^{(t)}] \nabla f_i(\theta) = 0 \implies \nabla f(\theta) = 0. \tag{2.19}$$

Finally, the existence of straggling devices introduces asynchronicity in the decentralized optimization procedure. In particular, a device i that fails at completing the gradient computation at a given optimization iteration is allowed to apply the result in a later one, without discarding the computation results. While we do not specify the delay distribution, we rather introduce the following assumption regarding the staleness of gradients.

Assumption 2. *For all iteration t, there exists a constant $\gamma \leq 1$ such that*

$$\mathbb{E}\left\|\nabla f(\bar{\theta}^{(t)}) - \frac{\sum_{i=1}^{m} \nabla f_i(\theta_i^{(t-\tau_i)})}{m}\right\|^2 \leq \gamma \mathbb{E}\left\|\nabla f(\bar{\theta}^{(t)})\right\|^2 + L^2 \frac{\sum_{i=1}^{m} \mathbb{E}\left\|\theta_i^{(t)} - \bar{\theta}^{(t)}\right\|^2}{m}. \quad (2.20)$$

The above assumption is similar to the one in [65] with an additional consensus error term. Note that the value of γ is proportional to the staleness of the gradients and in case of perfect synchronization ($\gamma = 0$) the bound amounts to a standard consensus error term.

2.4.3 Convergence Guarantee

In this subsection, we demonstrate the convergence of the decentralized optimization procedure to a stationary point of the problem (2.1).

Theorem 1. *Consider a network of unreliable communicating devices in which the expected consensus rate is satisfied with constant p and each device can be a straggler with probability $\rho_i < 1$. If Assumptions 1 and 2 are satisfied, asynchronous DSGD with constant learning rate $\eta_i = \min_j(1-\rho_j)/(\sqrt{4LT}(1-\rho_i))$ and consensus rate $\zeta = 1/T^{3/8}$ satisfies the following stationary condition*

$$\frac{1}{T}\sum_{t=1}^{T}\left\|\nabla f(\bar{\theta}^{(t)})\right\|^2 \leq \frac{8\sqrt{L}(f(\bar{\theta}^{(T)}) - f^*)}{\gamma' \rho_{min}\sqrt{T}} + \frac{3G^2 L}{T^{1/4} p^2 \gamma'} + \sqrt{\frac{L}{4T}}\frac{\sigma^2}{m\gamma' \min_j(1-\rho_j)} + \sum_{i=1}^{m}\frac{\sigma_{w,i}^2}{m\gamma'}\left(\frac{2L^2\gamma}{pT^{3/8}} + \frac{4L\sqrt{L}}{mT^{1/4}\rho_{min}}\right) \quad (2.21)$$

where $\gamma' = 1-\gamma$, $\rho_{min} = \min_j(1-\rho_j)$ and $f^* = \min_{\theta\in\mathbb{R}^d} f(\theta)$.

Proof. See Appendix A.3. □

The above theorem establishes a vanishing bound on the stationarity of the returned solution, which involves quantities related to both communication and computation impairments. In particular, the constant of the slowest vanishing terms $T^{-1/4}$ contains the term p related to random connectivity, as well as γ' and ρ_{min} due to stragglers.

2.5 Numerical Results

The effectiveness of the proposed asynchronous DSGD scheme is assessed using a network of $m = 15$ devices that collaboratively optimize the parameters of a convolutional neural network (CNN) for image classification with Fashion-MNIST. Gradients are calculated using batches of 16 data samples and the performance is evaluated using a test set of 500 images. We model the channel gain between each device pair as Rayleigh fading and we assume a shifted exponential computation time at each device, i.e., $T_{comp} = T_{min} + Exp(\mu)$

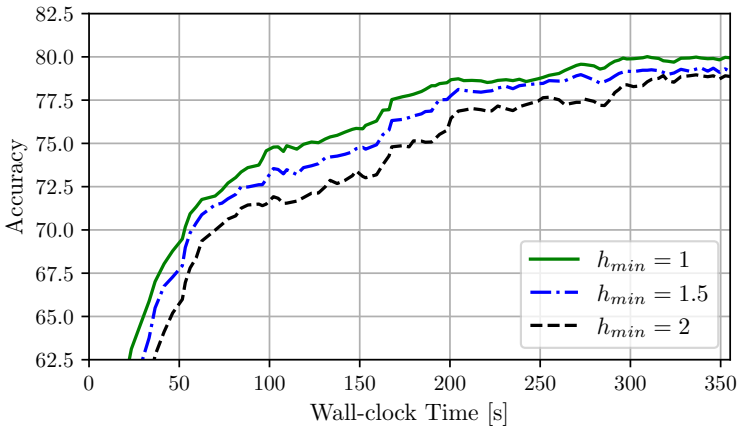


Figure 2.3: Test accuracy versus time under different channel gain thresholds. Smaller thresholds result in larger average consensus rates and therefore in faster convergence.

with $T_{min} = 0.25$s and $\mu = 1$. In Fig. 2.3, nodes communicate only when the channel is in favorable conditions, i.e., when the channel gain exceeds a certain minimum threshold h_{min}. This allows to save energy; however, while higher threshold values result into lower average energy consumption, they also produce mixing matrices with smaller consensus rate, thus increasing the convergence time.

To study the effect of computation impairments, our proposed asynchronous learning algorithm is compared with: (i) *synchronous DSGD*, which waits for all devices to finish their computations; and (ii) *synchronous DSGD with a delay barrier* T_{max}, which discards computation from users that violate the maximum computing time. Compared to the latter, our asynchronous procedure allows for slow devices to reuse stale gradient computations during later iterations. In Fig.2.4, we plot the evolution of the test accuracy of the aforementioned algorithms under two different values of T_{max}. For a moderate delay constraint $T_{max} = \mathbb{E}[T_{comp}]$, asynchronous DSGD and synchronous DSGD with delay barrier perform similarly as the fraction of slow users is modest. Nonetheless, imposing a delay constraint and discarding slow devices greatly reduces the training time compared to the synchronous DSGD case. On the other hand, for a stringent delay requirement, $T_{max} = \frac{4}{5}\mathbb{E}[T_{comp}]$, reusing stale gradients turns out to be beneficial and the proposed asynchronous DSGD attains higher accuracy faster compared to the synchronous DSGD with a delay barrier.

2.6 Conclusion

In this chapter, we have proposed and analyzed an asynchronous implementation of DSGD, which enables decentralized optimization over realistic wireless networks with unreliable communication and heterogeneous devices in terms of computation capabilities. We have studied the effect of both communication and computation failures on the training performance and proved non-asymptotic convergence guarantees for the proposed

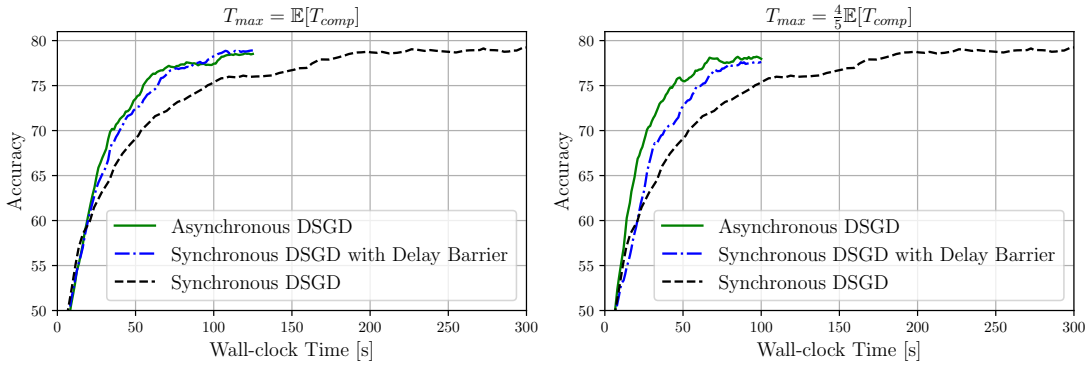


Figure 2.4: Test accuracy for the asynchronous, synchronous with delay barrier, and synchronous schemes under two different values of T_{max}.

algorithm. The main takeaway is that reusing outdated gradient information from slow devices is beneficial in asynchronous decentralized learning.

Chapter 3

UAV-Aided Decentralized Learning over Mesh Networks

In Chapter 2 we have shown that decentralized learning algorithm can be used to collaboratively train a machine learning (ML) over realistic wireless network affected by computation and communication impairments. As shown in Theorem 1 the convergence speed of the decentralized optimization algorithm severely depends on the degree of the network connectivity, with denser network topologies leading to shorter convergence time. Consequently, the local connectivity of real world mesh networks, due to the limited communication range of its wireless nodes, undermines the efficiency of decentralized learning protocols, rendering them potentially impracticable. In this chapter we investigate the role of an unmanned aerial vehicle (UAV), used as flying relay, in facilitating decentralized learning procedures in such challenging conditions. We propose an optimized UAV trajectory, that is defined as a sequence of waypoints that the UAV visits sequentially in order to transfer intelligence across sparsely connected group of users. We then provide a series of experiments highlighting the essential role of UAVs in the context of decentralized learning over mesh networks.

3.1 Introduction

Most of the decentralized learning schemes over wireless networks have been proposed and analyzed under the assumption that the network topology is strongly connected on average. [24, 25, 62, 63, 69]. However, real world mesh networks are characterized by local, rather than global connectivity, and groups of nodes are often isolated or sparsely connected to the rest of the network due to their limited communication range. In these scenarios, decentralized learning is either not possible or its performance is severely hampered.

At the same time, unmanned aerial vehicles (UAVs) represent an appealing solution to mitigate limited ground connectivity. UAVs have been used as smart flying relays to improve multi-hop routing capabilities [70], to self-organize in flying mesh networks [71] and to improve coverage to ground users [72]. In this chapter, we investigate the role of

UAVs in aiding decentralized learning protocols over ground mesh wireless networks.

The combination of FL and UAV assisted communication has recently been explored; however, these studies have been limited to scenarios in which the UAV has the role of a parameter server (PS), i.e., aggregating model estimates received from ground nodes and subsequently broadcasting the aggregated model back to the ground [73, 74].

The results presented in this chapter differ from these previous works as it considers the UAV serving as a relay and it assumes that ground nodes are able to carry out learning even in the absence of a UAV, by exploiting the already existing ground D2D links. This feature dramatically improves the convergence speed, versatility and fault tolerance of the proposed solution. We propose an optimized trajectory, given as a sequence of waypoints visited by the UAV, which is designed to intelligently provide relaying opportunities to ground nodes and to diffuse locally optimized model across subsets of users with limited connectivity. We provide experiments showing that with the aid of a UAV following the proposed trajectory, it is possible to harness the full potential of the mesh network in spite of sparse and local connectivity, and to accelerate learning compared to UAV-aided federated learning algorithms.

3.2 System Model

We consider a network of $m+1$ devices comprising m ground users plus a UAV serving as a flying relay. We index the ground user by $1, \dots, m$ and we denote the location of the i-th ground device by $\mathbf{p}_i = [x_i, y_i, z_i] \in \mathbb{R}^3$, where x_i and y_i are the horizontal coordinates while z_i denotes the elevation. We assume that ground devices are *static*, namely their position is not a function of time. On the other hand, the UAV location is denoted $\mathbf{p}_{uav} = [x, y, z] \in \mathbb{R}^3$ and is assumed to be *time-varying* in the horizontal coordinates x and y, but not in the vertical one z. Furthermore, the UAV elevation z is set to be larger than a safety altitude z_{min}.

3.2.1 Communication Model

Communication among network nodes takes place in rounds. At every communication round $t \in \{\tau, 2\tau, \dots\}$, the channel gain coefficient $g_{i,j}^{(t)} \in \mathbb{R}$, expressed in dB, between each pair of distinct user $(i, j) \in [1:m]^2$ is given by

$$g_{i,j}^{(t)} = g_{j,i}^{(t)} = \beta_g - \alpha_g 10 \log_{10} d_{i,j} + \eta_g^{(t)} \tag{3.1}$$

where α_g is the path loss exponent, β_g is the average channel gain in dB at a reference distance $d = 1$, $d_{i,j} = \|\mathbf{p}_i - \mathbf{p}_j\|_2$ is the distance between nodes i and j, and $\eta_g \sim \mathcal{N}(0, \sigma_g^2)$ models the shadowing effects. For simplicity, we assume that the link parameters α_g, β_g and σ_g are homogeneous across pairs of ground users; however, the proposed solution can easily accommodate heterogeneous channel parameters. At communication round t, the channel gain link between the UAV and a ground node i under Line-of-Sight (LoS) conditions is modeled as

$$g_{i,L}^{(t)} = \beta_L - \alpha_L 10 \log_{10} d_i^{(t)} + \eta_L^{(t)}, \tag{3.2}$$

while under Non-Line-of-Sight (NLoS) propagation it follows

$$g_{i,N}^{(t)} = \beta_N - \alpha_N 10 \log_{10} d_i^{(t)} + \eta_N^{(t)} \tag{3.3}$$

where $d_i^{(t)} = \left\| \mathbf{p}_i - \mathbf{p}_{uav}^{(t)} \right\|_2$ denotes the time-dependent distance between the UAV and user i, α_L, β_L and $\eta_L^{(t)} \sim \mathcal{N}(0, \sigma_L^2)$ are the channel parameters under LoS, while α_N, β_N and $\eta_N^{(t)} \sim \mathcal{N}(0, \sigma_N^2)$ describe the channel under NLoS propagation. These parameters are assumed to be homogeneous across users for simplicity.

The LoS probability between the UAV at a position $\mathbf{p}_{uav}^{(t)}$ and user i is modeled using the s-model [75]

$$\rho_i^{(t)} = \frac{1}{1 + e^{-a_i \theta_i^{(t)} + b_i}} \tag{3.4}$$

where a_i and b_i are model coefficients related to the propagation environment, and $\theta_i^{(t)}$ is the elevation angle between the UAV and ground user i. At every communication round t, a link between network nodes is modeled using a simple, yet classical, on-off channel model. Accordingly two nodes communicate if and only if the associated channel gain exceeds a threshold g_{th}. Therefore, the resulting ground connectivity matrix $A_{gr}^{(t)} \in [0,1]^{m \times m}$, is symmetric, has diagonal elements being equal to 1 and Bernoulli distributed off-diagonal entries

$$[A_{gr}^{(t)}]_{j,i} = [A_{gr}^{(t)}]_{i,j} \sim Bern\left(1 - \Phi\left(\frac{g_{th} - \bar{g}_{i,j}^{(t)}}{\sqrt{\sigma_g}}\right)\right) \tag{3.5}$$

where $\Phi(\cdot)$ denotes standard Gaussian cumulative distribution function and $\bar{g}_{i,j}^{(t)} = \mathbb{E}[g_{i,j}^{(t)}]$.

Similarly, the connectivity between the UAV and ground users is described by a vector $\mathbf{a}_{uav}^{(t)} \in [0,1]^{1 \times m}$ with entries

$$[\mathbf{a}_{uav}^{(t)}]_i \sim Bern\left(1 - \bar{\rho}_i \Phi\left(\frac{g_{th} - \bar{g}_{i,N}^{(t)}}{\sqrt{\sigma_N}}\right) - \rho_i \Phi\left(\frac{g_{th} - \bar{g}_{i,L}^{(t)}}{\sqrt{\sigma_L}}\right)\right) \tag{3.6}$$

where $\bar{\rho}_i = 1 - \rho_i$, $\bar{g}_{i,L}^{(t)} = \mathbb{E}[g_{i,L}^{(t)}]$ and $\bar{g}_{i,N}^{(t)} = \mathbb{E}[g_{i,N}^{(t)}]$.

Based on the instantaneous connectivity status, determined by the realization of $\mathbf{a}_{uav}^{(t)}$, the UAV serves as a one-hop relay for the communication among grounds users. The connectivity matrix resulting from the relaying opportunities offered by the UAV to ground users is obtained as

$$A_{uav}^{(t)} = (\mathbf{a}_{uav}^{(t)})^T \mathbf{a}_{uav}^{(t)}. \tag{3.7}$$

If follows that $A_{uav}^{(t)}$ is a symmetric random binary matrix whose entry (i,j) is 1 if an only if there exists a relaying opportunity between ground user i and j, and 0 otherwise.

Overall, the aggregated connectivity matrix, accounting for link existence either by D2D ground communication or thanks to UAV relaying, is given by

$$A^{(t)} = J_m - (J_m - A^{(t)}_{uav}) \odot (J_m - A^{(t)}_{gr}) \tag{3.8}$$

where J_m is the $m \times m$ all-one matrix and $\odot$ denotes the Hadamard product.

For every realization of the connectivity matrix $A^{(t)}$, the set of devices connected to node i is

$$\mathcal{N}^{(t)}(i) := \{j : [A^{(t)}]_{i,j} = 1\}. \tag{3.9}$$

Note that every ground user is connected to itself.

3.2.2 Learning procedure

We assume that the goal of ground devices is to collaboratively train a machine learning model in order to benefit from the aggregation of local computational resources and in-situ data. In particular, we assume that each ground device is endowed with a local loss function $f_i : \mathbb{R}^d \to \mathbb{R}$ and local parameter estimate $\theta_i \in \mathbb{R}^d$. This corresponds to the distributed empirical risk minimization problem, which entails most decentralized learning problems, whenever $\{f_i\}_{i=1}^m$ are loss terms defined over local datasets $\{\mathcal{D}_i\}_{i=1}^m$ and $\{\theta_i\}_{i=1}^m$ are local model estimates.

The network objective consists in the minimization of the aggregate network loss subject to a consensus constraint

$$\begin{aligned} \underset{\theta_1,\dots,\theta_m}{\text{minimize}} \; & f(\theta_1, \dots, \theta_m) := \frac{1}{m} \sum_{i=1}^{m} f_i(\theta_i) \\ \text{s.t.} \quad & \theta_1 = \theta_2 = \cdots = \theta_m. \end{aligned} \tag{3.10}$$

In the following, we denote the average network estimate as $\bar{\theta} = 1/m \sum_{i=1}^m \theta_i$.

To solve (3.10), we consider the asynchronous decentralized stochastic gradient descent (DSGD) algorithm proposed in [69]. According to this optimization scheme, ground devices alternate between a local optimization phase based on gradient information (computation phase) and a communication phase to exchange the updated local estimates with one-hop neighbours. To locally optimize the model estimate θ_i, we assume that each device i can query a stochastic oracle that is unbiased,

$$\mathbb{E}[g_i(\theta_i)] = \nabla_\theta f_i(\theta_i), \tag{3.11}$$

and has bounded variance and magnitude

$$\mathbb{E}\|g_i(\theta_i) - \nabla_\theta f_i(\theta_i)\|^2 \leq \sigma^2, \tag{3.12}$$

$$\mathbb{E}\|g_i(\theta_i)\| \leq G^2. \tag{3.13}$$

Furthermore, to account for computation impairments and energy constraints, we admit the existence of straggling ground users that can become inactive or postpone the local

optimization computation. At each communication round t, the local update rule at device i becomes

$$\theta_i^{(t+\frac{1}{2})} = \begin{cases} \theta_i^{(t)}, & \text{if device } i \text{ is straggler at round } t \\ \theta_i^{(t)} - \eta_i^{(t)} g_i(\theta^{(t-\tau_i)}) & \text{otherwise} \end{cases} \tag{3.14}$$

where $\eta_i^{(t)}$ is a local learning rate and the delay $\tau_i \geq 0$ accounts for the staleness of the gradient information at device i. Subsequently, each device i shares its updated local estimate $\theta_i^{(t+\frac{1}{2})}$ with its neighbours $\mathcal{N}^{(t)}(i)$ using either a digital or analog communication protocol [62, 63]. The received estimates are then averaged to obtain the new local estimate

$$\theta_i^{(t+1)} = \sum_{j \in \mathcal{N}(i)} w_{i,j} \theta_j^{(t+\frac{1}{2})} \tag{3.15}$$

where $w_{i,j}$ are the entries of the mixing matrix $W^{(t)}$ obtained using a Metropolis-Hastings weighting scheme [68] given in (2.8).

In [69], it has been shown that the performance of the asynchronous DSGD optimization procedure depends both on the activity of users and on the degree of wireless network connectivity. In particular, with more connected network topologies converging faster than sparser ones. This motivates the use of a UAV to facilitate the diffusion of locally optimized models, and to render the decentralized learning protocol more efficient in spite of sparse and local ground connectivity.

3.3 Trajectory Optimization

At every optimization round t, the connectivity matrix $A^{(t)}$ associated to the network of ground users depends on the UAV location $\mathbf{p}_{uav}^{(t)}$ and it can be enhanced thanks to the the relaying opportunities it provides to ground devices. In the following, we propose to optimize the UAV trajectory during the optimization process so that the distributed learning procedure is facilitated.

A key quantity that is used to measure the information diffusion capabilities of a network is the expected consensus rate [61]. While this quantity can be used to characterize the rate of convergence of DSGD procedure, it does not provide a tractable optimization objective to derive the UAV trajectory. For this reason we define a more tractable surrogate objective that yields the optimized trajectory as a sequence of waypoints $\{w_i\}_{i=1}^n$. In particular, we assume that the initial way point w_0 is equal to the initial UAV location $\mathbf{p}_{uav}^{(0)}$, and that the sequence of waypoints is determined on-the-fly, with the waypoint w_{i+1} being computed when the UAV reaches the location specified by the previous waypoint w_i. The waypoints are designed so as to hover the UAV on a position that maximizes the probability of creating a relaying opportunities between users that, up to communication round t, have not been able to communicate. To this end, we recursively define an link activity rate matrix $R^{(t)}$ as

$$R^{(0)} = 0 \tag{3.16}$$

$$R^{(t+1)} = \gamma R^{(t)} + (1-\gamma)\mathbb{E}[A^{(t)}], \tag{3.17}$$

for $\gamma \in (0,1)$. Denoting by t_i the communication round in which the UAV reaches the waypoint w_i, the subsequent waypoint is obtained solving the following optimization problem

$$\underset{p_t}{\text{maximize}} \left\| (1 - R^{(t_i)}) \odot \mathbb{E}[A_{uav}] \right\|_1 \tag{3.18}$$

where $\|\cdot\|_1$ denotes the entry-wise 1-norm; namely, the sum of absolute values of the matrix entries.

The optimization problem (3.18) determines the next waypoint so as to maximize the relaying opportunities between pair of users associated to links with a low activity rate. The main challenges in solving (3.18) are the lack of a close form expression for $\Phi(\cdot)$ and the non-convexity of the objective. In order to make the objective differentiable, we approximate $\Phi(\cdot)$ using the sigmoid function

$$S(x) := \frac{1}{1+e^{-\alpha x}} \tag{3.19}$$

where α is a fitting parameter set to $\alpha = -1.702$ as proposed in [76]. This approximation of the objective allows us to employ efficient gradient based solvers to generate the sequence of waypoints.

Nonetheless, the optimization objective (3.18) remains non-convex. In order to reduce the probability of obtaining a waypoint associated to poor local maxima, we employ gradient descent with restarts. The number of restart points is chosen to meet the UAV computation constraints and the restart points are sampled uniformly at random inside the convex hull determined by ground user locations.

3.4 Simulations

To test the proposed solution, we consider a network deployment of $30 \times 60\text{m}^2$ with 23 ground devices deployed at the ground level ($z = 0$m) as depicted in Figure 3.1. The propagation parameters describing the ground links channel gain are set to $\alpha_g = 3, \beta_g = -30$ dB and $\sigma_g^2 = 1$, and the channel gain threshold determining active/inactive links is fixed to $g_{th} = -60$ dB. In the considered deployment, ground users are naturally clustered together in 3 distinct groups and the path exponent is such that communication within each cluster is possible, but links between users belonging to different clusters are active with negligible probability. Furthermore, we consider an obstruction (gray vertical line) that amounts to a 35 dB attenuation for the links between users residing on opposite side of the line. A UAV flying at a fixed altitude of 10m serves as a relay to enhance ground connectivity. The channel gain parameters describing the link between ground users and the UAV under LoS propagation are $\alpha_L = 2.5, \beta_L = -30$ dB and $\sigma_L^2 = 1$, while under NLoS are $\alpha_N = 3, \beta_N = -30$ dB and $\sigma_N^2 = 1$.

We assume that the devices store only 10 data samples from the FashionMNIST dataset, which alone would not guarantee good inference capabilities. Therefore, they

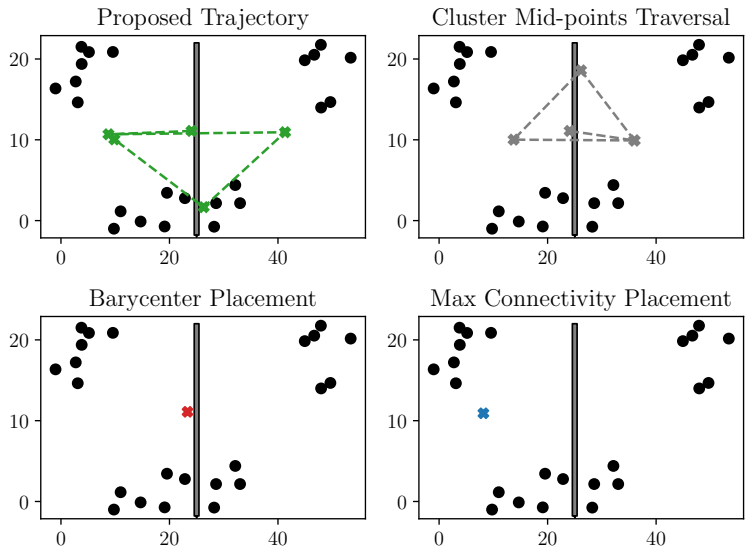


Figure 3.1: Different UAV trajectory and placements. Black dots represent ground users and the gray vertical line is a propagation obstacle that corresponds to a 35 dB attenuation.

wish to harness the distributed dataset to jointly train a machine learning model. In the following experiments we consider a fully connected neural network with one hidden layers comprising 25 neurons. Ground devices update the model employing a gradient descent optimizer and a geometrically decaying learning rate $\eta_i^{(t)} = 0.1 \cdot (0.995)^t$.

In this setting, the role of the UAV is to intelligently create relaying opportunities so to promote collaborative learning, and to facilitate the ground users to harness the entire distributed dataset by global diffusion of the locally optimized models, in spite of sparse and local connectivity.

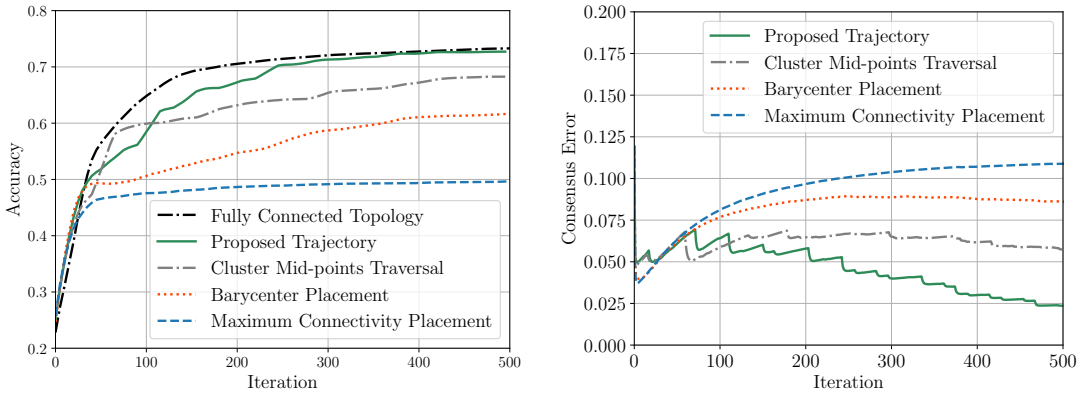
We benchmark the proposed solution, corresponding to a UAV visiting the sequence of waypoints returned by (3.18) and serving users for 20 optimization rounds each time a waypoint is reached, and compare it with alternative trajectory optimization schemes.

In particular, we consider the *cluster mid-points traversal* trajectory according to which the UAV first runs a k-means clustering algorithm to detect natural clusters of the ground devices positions, and then visits sequentially the mid-points between each pair of cluster centers. Once the mid-point is reached, the UAV serves ground users for 20 optimization rounds and then it flies to the next location.

We consider the *barycenter* placement, in which the UAV hovers at a fixed location $\mathbf{p}_{uav}^{bar}$ determined by the mean ground user location.

Finally, we consider the *maximum connectivity* placement in which the UAV location is fixed and set equal to the coordinate that maximizes the probability of creating a relay links. This is obtained solving the following maximization problem

$$\underset{p}{\text{maximize}} \|(1 - A_{gr}) \odot \mathbb{E}[A_{uav}]\|_1. \tag{3.20}$$



(a) Testing Accuracy

(b) Consensus Error

Figure 3.2: Testing accuracy averaged over 5 runs, obtained by the mean network estimate, using different UAV-Aided decentralized learning protocols. Evolution of the average consensus error (3.21) attained by the benchmarked trajectories. Smaller consensus error corresponds to disagreement between network nodes.

For both the barycenter and the maximum connectivity placements, the UAV serves ground users at every communication round as it hovers at a fixed location for the entire learning procedure. We consider the centralized learning solution, corresponding to a *fully connected* topology, as a performance upper bound. For all listed approaches we run the distributed optimization protocol for 500 rounds and we track the testing accuracy of the mean network estimate $\bar{\theta}$. We also consider the network consensus error metric

$$\varepsilon(\theta_1, \dots, \theta_m) = \frac{1}{dm} \sqrt{\sum_{i=1}^{m} \left\| \theta_i - \bar{\theta} \right\|_2^2}, \tag{3.21}$$

which measures the degree of disagreement among ground nodes estimates and it is a proxy to assess how effective is the UAV trajectory in satisfying the consensus constraint in (3.18).

In Figure 3.1 we plot the ground user locations (black dots) and we overlay the UAV trajectories during the training process. Specifically, in the top left corner, we report in green the UAV trajectory returned by the proposed scheme. The UAV frequently hovers between the disconnected components in order to relay information between the clusters and to diffuse model estimates between groups of interconnected ground nodes that rarely communicate using D2D ground links. In the top right corner, we provide the trajectory of the cluster mid-points traversal solution (gray). The UAV successfully identifies the clusters and it sequentially visits the mid-points. This strategy enhances the ground connectivity, but it fails at providing relaying opportunities across the two bottom components that are disconnected due to the propagation obstacle. In the bottom row, we plot the barycenter (left plot) and maximum connectivity (right plot) placementes. Both solutions yields a static UAV placement that is fixed throughout the entire training

phase.

In Figure 3.2a we report the testing accuracy attained by the mean network estimate $\bar{\theta}$ when decentralized learning is assisted by a UAV flying according to the different trajectories. The testing accuracy allows us to quantify the extent to which the UAV is beneficial to the collaborative learning process. The proposed solution (in green) is able to take full advantage of the distributed dataset, and it successfully enables fast distributed training with a final accuracy level that matches the accuracy of the centralized solution (in black). The barycenter solution (in red) converges slowly to a lower accuracy level, highlighting the necessity of a dynamic UAV placement to take full advantage of the network resources. The cluster mid-points traversal solution, despite enhancing ground connectivity, it is not able to connect all the disjoint components and therefore it converges to suboptimal solution. Similarly, the maximum connectivity placement is not able to connect all the disjoint network components and to transfer intelligence across different clusters.

In Figure 3.2b we report the network consensus error evolution attained by the different UAV trajectories. While the proposed trajectory is able to reduce the consensus error during training and it eventually ensures that the edge devices reach a common learning goal, the other baselines are not able to drive network nodes to a globally shared model estimate.

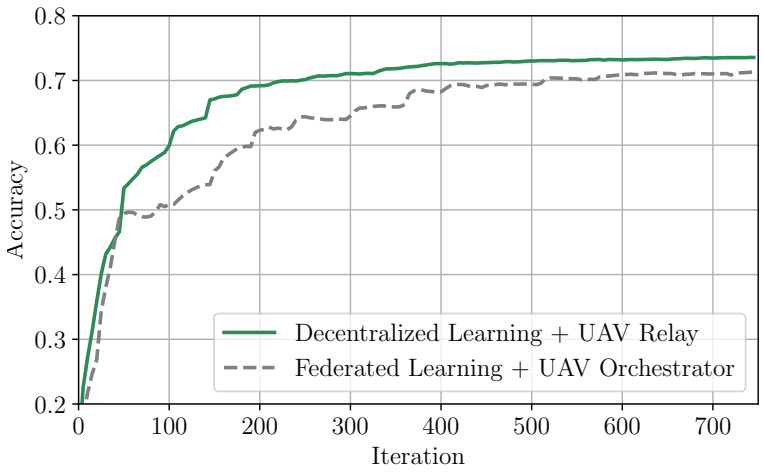


Figure 3.3: Testing accuracy, averaged over 5 experiments, obtained by the mean network estimate when training is aided by a UAV serving as a relay to assist the decentralized learning protocol (green), or as an orchestrator to perform federated learning (gray dashed).

Finally, we propose a comparison between the proposed decentralized learning scheme and a UAV-aided federated learning protocol, as in [73]. In particular, for the federated learning algorithm, we assume that the UAV serves as a PS, and it orchestrates the learning procedure by collecting locally optimized models by the network devices and broadcasting aggregated estimates back to the ground users. On the other hand, in case of decentralized learning, the UAV serves as a relay and the ground devices can also exploit

the available D2D ground links to exchange model estimates, in principle being able to perform learning without the presence of a UAV. As a result, the proposed protocol is more flexible with respect to the communication topology, it can easily accommodate multiple assisting UAVs, and converges faster. To compare these two approaches we study the same deployment as in Figure 3.1. We assume that the relaying UAV follows the proposed trajectory, while the UAV serving as orchestrator follows the trajectory obtained solving (3.18) setting $A_{gr}^{(t)} = 0$, trying to serve large groups of users prioritizing stale ones, akin to [73]. In Figure 3.3 we report the testing accuracies attained by the protocols. The proposed approach drastically reduces the training time, halving the number of iterations required to reach the final performance obtained by the federated learning protocol.

3.5 Conclusion

In this chapter, we have studied the benefits that a flying relay can bring to a network of wireless devices that are jointly training a machine learning model. We have proposed a trajectory optimization scheme that enhances the ground connectivity so as to facilitate the diffusion of locally optimized model estimates, and that enables ground users to take full advantage of network computational and data resources. We have also provided a series of experiments highlighting how a properly designed UAV trajectory can greatly promote decentralized training and outperform UAV-aided federated learning protocols.

Part II

Robust Learning for Heterogeneous Data

Chapter 4

Communication-Efficient Distributionally Robust Decentralized Learning

As shown in Chapter 2 and 3, decentralized learning algorithms empower interconnected edge devices to share data and computational resources to collaboratively train a machine learning model without the aid of a central coordinator (e.g. an orchestrating basestation). In the case of heterogeneous data distributions at the network devices, collaboration can yield predictors with unsatisfactory performance for a subset of the devices. For this reason, in this chapter we consider the formulation of a distributionally robust decentralized learning task and we propose a decentralized single loop gradient descent/ascent algorithm (AD-GDA) to solve the underlying minimax optimization problem. We render our algorithm communication efficient by employing a compressed consensus scheme and we provide convergence guarantees for smooth convex and non-convex loss functions. Finally, we corroborate the theoretical findings with empirical evidence of the ability of the proposed algorithm in providing unbiased predictors over a network of collaborating devices with highly heterogeneous data distributions.

4.1 Introduction

Decentralized learning algorithms have gained an increasing level of attention, mainly due to their ability to harness, in a fault tolerant and privacy preserving manner, the large computational power and data availability at the network edge exploiting device-to-device (D2D) communication [62, 63, 69]. According to this framework, a set of interconnected devices (e.g., smartphones, IoT devices, health monitors, research labs, etc.) collaboratively train a machine learning model alternating between local model updates, based on in situ data, and D2D type of communication to exchange model-related information. Compared to federated learning in which a swarm of edge devices communicates with a central parameter server (e.g., a shared access point) at each communication round, fully decentralized learning has the benefits of removing the single

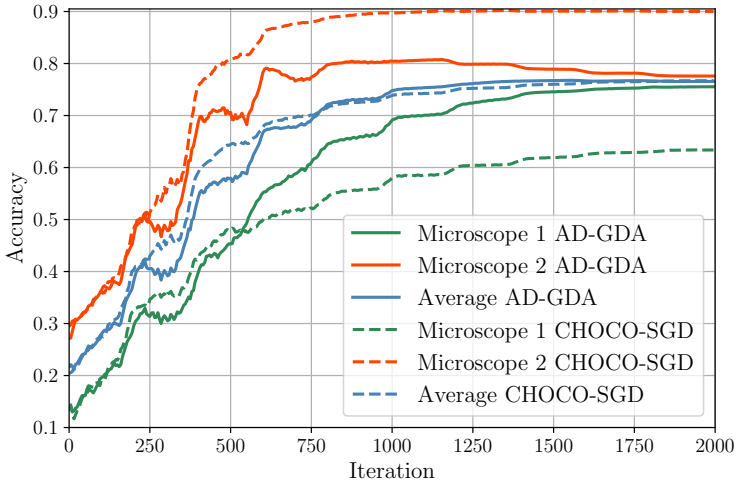


Figure 4.1: Validation accuracy of a mouse cell image classifier trained on the COOS7 dataset [2]. We consider a network of 5 devices with one device sampling images using a different microscope from the rest of the collaborating devices. CHOCO-SGD (solid lines), a not robust decentralized learning scheme, yields a model with highly imbalanced performance between the two type of instruments, while AD-GDA (dashed curves), the proposed distributionally robust algorithm, drastically reduces the accuracy gap and improve fairness among the collaborating devices.

point of failure and of alleviating the communication bottleneck inherent to the star topology.

The heterogeneity of distributedly generated data by the Internet-of-Things (IoT) entails a major challenge, represented by the notions of fairness [77] and robustness [78]. In the distributed setup, the customary global loss function is the weighted sum of the local empirical losses, with each term weighted by the fraction of samples that the associated device stores. However, in the case of data heterogeneity across participating parties, a model minimizing such definition of risk can lead to unsatisfactory and unfair [1] inference capabilities for certain subpopulations. Consider the example given in Fig.4.1 in which a network of IoT devices with different sensing capabilities (e.g., IoT devices with heterogeneous measuring instruments) wishes to collaboratively train a machine learning model. In this setting, a model obtained by myopically minimizing the standard notion of risk defined over the aggregated data can be severely biased towards some devices at the expense of others, leading to potentially dangerous or unfair decision making processes.

To tackle this issue, distributionally robust learning (DRL) aims at maximizing the worst-case performance over a set of distributions, termed as uncertainty set, which possibly contains the testing distribution of interest. Typical choices of the uncertainty sets are perturbed version of the training distribution [80] or, whenever the training

[1] In the machine learning community, the notion of fairness has many facets. In this chapter, we will use the term "fair" in accordance with the notion of good-intent fairness as introduced in [79].

samples come from a mixture of distributions, the set of potential subpopulations resulting from such mixture [81, 82]. Robust distributed learning with heterogeneous data in which different distributions exist at the various devices falls in the latter category, as the natural ambiguity set is the one represented by the convex combination of the local data distributions. In this case, minimizing the worst-case risk is equivalent to trying to ensure a minimum level of performance for each participating device. Specifically for the federated case, Mohri et al. [79] introduced agnostic federated learning (AFL) as a means to ensure fairness and proposed a gradient based algorithm to solve the underlying minimax optimization problem. Later, in [83], a communication-efficient version of the optimization algorithm, which avoids frequent retransmission of the dual variables, was proposed.

In virtue of the advantages of the fully decentralized setup and advocating the necessity for robust and fair predictors in future generation networks, in this chapter we propose and analyze a distributionally robust learning procedure for D2D communication networks. In contrast to previous works on collaborative distributional robust learning, our algorithm operates in the absence of a central aggregator and with devices limited to local and possibly sparse communication; therefore, it exhibits increased scalability, adaptability and tolerance against network failures. Despite the additional complexity stemming from the minimax nature of the distributionally robust decentralized optimization problem, our solution is computationally lightweight and communication-efficient as it alternates between local single-loop stochastic gradient descent/ascent model updates and compressed consensus steps in order to cope with local connectivity.

We establish convergence guarantees for the proposed algorithm both in the case of smooth convex and smooth non-convex local loss functions. In the former case, the algorithm returns an ϵ-optimal solution after $\mathcal{O}(1/\epsilon^2)$ iterations. In the latter, the output is guaranteed to be an ϵ-stationary solution after $\mathcal{O}(1/\epsilon^2)$ iterations whenever the stochastic gradient variance is also bounded by ϵ, otherwise the same guarantee can be obtained by increasing the number of calls to the stochastic gradient oracle. Furthermore, we demonstrate the effectiveness of the proposed algorithm in finding a robust predictor under different compression schemes, network topologies, and models architectures. We also compare the proposed approach against the distributionally robust federated learning counterpart and we the proposed solution attains higher worst-case distribution accuracy for the same number of transmitted bits, effectively reducing the communication burden of the distributionally robust learning procedure on the edge of the network.

4.2 Related work

Initiated in the 80s by the work of Tsitsiklis [60,84], the study of decentralized optimization algorithms was spurred by their adaptability to various network topologies, reliability to link failures, privacy-preserving capabilities, and potentially superior convergence properties compared to the centralized counterpart [59, 85–88]. This growing interest and the advent of large-scale machine learning brought forth an abundance of optimization algorithms both in the deterministic and stochastic settings [89–93]. With the intent of extending its applicability, a concurrent effort has been made to devise techniques

able to reduce the delay due to inter-device communication. Notable results in this direction are the introduction of message compression techniques, such as sparsification and quantization [94–99], and event-triggered communication to allow multiple local updates between communication rounds [100, 101]. Decentralized learning algorithms have also been studied in the context of wireless communication as an enabler of edge intelligence for beyond 5G (B5G) networks. [62, 63, 69].

Distributional robustness copes with the frequent mismatch between training and testing distributions by posing the training process as a game between a learner and an adversary, which has the ability to choose the testing distribution within an uncertainty set [102]. Restraining the decisional power of the adversary is crucial to obtain meaningful and tractable problems and a large body of the literature deals with uncertainty sets, represented by balls centered around the training distribution and whose radius are determined by f-divergences [103,104] or Wasserstein distance [80,105,106]. Distributional robustness is deeply linked with the notion of fairness as particular choices of uncertainty sets allows to guarantee uniform performance across the latent subpopulations in the data [81, 82]. In the case of federated learning, robust optimization ideas have been explored to ensure uniform performance across all participating devices [79] but, to the best of our knowledge, not in the context of fully decentralized learning.

Distributionally robust learning typically entails saddle point optimization problems. The convergence properties of saddle point optimization algorithms have also been studied in the decentralized scenario for the convex-concave setting [107, 108]. More recently, the assumptions on the convexity and concavity of the objective function have been relaxed. In [109] an algorithm for non-convex strongly-concave objective functions has been proposed; however, the double-loop nature of the solution requires to solve the inner maximization problem with increasing level of accuracy rendering it potentially slow. On the other hand, our algorithm is based on a single loop optimization scheme - with dual and primal variables being updated at each iteration in parallel - and, consequently, has a lower computational complexity. For the non-convex - non-concave case, [110] provides a proximal point algorithm, while a simpler gradient based algorithm is provided in [111] to train generative adversarial networks in a decentralized fashion. None of these works take into consideration communication efficiency in their algorithms.

4.3 System Model

We consider a network of m edge devices in which each device i is endowed with a local objective function $f_i : \mathbb{R}^d \to \mathbb{R}$ given by $\mathbb{E}_{\mathrm{z}\sim P_i}\ell(\theta, \mathrm{z})$, with P_i denoting the local distribution at device i and $\theta \in \mathbb{R}^d$ being the model parameter to be optimized. Whenever P_i is replaced by an empirical measure $\hat{P}_{i,n_i}$, the local objective function coincides with the empirical risk computed over n_i samples. Network devices are assumed to be interconnected according to a communication topology specified by a connected graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ in which $\mathcal{V} = \{1, \ldots, m\}$ indexes the devices and $(i, j) \in \mathcal{E}$ if and only if devices i and j can communicate. For each device $i \in \mathcal{V}$, we define its set of neighbors by $\mathcal{N}(i) := \{j : (i, j) \in \mathcal{E}\}$ and since we assume self-communication we have $(i, i) \in \mathcal{N}(i)$ for all i in $\mathcal{V}$. At each communication round, the network devices exchange messages

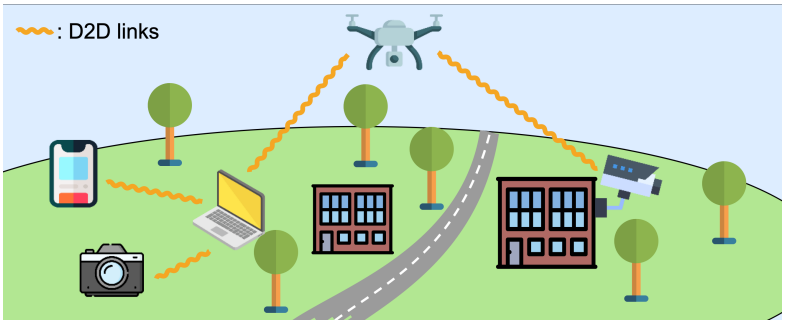


Figure 4.2: IoT network comprising edge devices with different sampling capabilities and operating in different conditions. The network goal consists in exploiting the heterogeneous distributed dataset and the D2D links to collaboratively train a robust and fair machine learning model.

with their neighbors and average the received messages according to a mixing matrix $W \in \mathbb{R}^{m\times m}$.

Assumption 3. *The mixing matrix $W \in \mathbb{R}^{m\times m}$ is symmetric and doubly-stochastic; we denote its eigengap by $\rho \in (0,1]$ and define $\beta = \|I - W\|_2 \in [0,2]$.*

Being the communication phase the major bottleneck of decentralized training, we assume that devices transmit only compressed messages instead of sharing uncompressed model updates. To this end, we define a, possibly randomized, compression operator $Q : \mathbb{R}^d \to \mathbb{R}^d$ that satisfies the following assumption.

Assumption 4. *For any $x \in \mathbb{R}^n$ and for some $\delta \in [0,1]$,*

$$\mathbb{E}_{\mathbb{Q}}\left[\|Q(x) - x\|^2\right] \leq (1-\delta)\|x\|^2. \tag{4.1}$$

The above definition is quite general as it entails both biased and unbiased compression operators. For instance, random quantization [97] falls into the former class and satisfies (4.1) with $\delta = \frac{1}{\tau}$. For a given vector $x \in \mathbb{R}^d$ and quantization levels 2^b, it yields a compressed message

$$x_b = \frac{\text{sign}(x)\|x\|}{2^b\tau}\left\lfloor 2^b \frac{|x|}{\|x\|} + \xi \right\rfloor \tag{4.2}$$

with $\tau = 1 + \min\left\{d/2^{2b}, \sqrt{d}/2^b\right\}$ and $\xi \sim \mathcal{U}[0,1]^{\otimes d}$. A notable representative of the biased category is the top-K sparsification [94], which for a given vector $x \in \mathbb{R}^d$ returns the K largest magnitude components and satisfies (4.1) with $\delta = \frac{K}{d}$. Operators of that type have been previously considered in the context of decentralized learning and the effect of compressed communication in decentralized stochastic optimization has been previously investigated [94, 99, 112]. The resulting communication cost savings have been showcased in the context of decentralized training of deep neural networks [112]. However, to the best of our knowledge, there are no applications of compressed communication to distributional robust training in the decentralized setup.

Algorithm 2: Agnostic Decentralized GDA with Compressed Communication (AD-GDA)

Input : Number of devices m, number of iterations T, learning rates η_θ and η_λ, mixing matrix W , initial values $\theta^0 \in \mathbb{R}^d$ and $\lambda^0 \in \Delta^{m-1}$.

Output : $\theta_o = \frac{1}{T}\sum_{t=0}^{T-1} \bar{\theta}^t$, $\lambda_o = \frac{1}{T}\sum_{t=0}^{T-1} \bar{\lambda}^t$

initialize $\theta_i^0 = \theta^0$, $\lambda_i^0 = \lambda^0$ and $s_i^0 = 0$ for $i = 1, \dots, m$

for t **in** $0, \dots T-1$ **do**

`// In parallel at each device` i

$\theta_i^{t+\frac{1}{2}} \leftarrow \theta_i^t - \eta_\theta \nabla_\theta g_i(\theta_i^t, \lambda_i^t, \xi_i^t)$ `// Descent Step`

$\lambda_i^{t+\frac{1}{2}} \leftarrow \mathcal{P}_\Lambda\left(\lambda_i^t + \eta_\lambda \nabla_\lambda g_i(\theta_i^t, \lambda_i^t, \xi_i^t)\right)$ `// Projected Ascent Step`

$\theta_i^{t+1} \leftarrow \theta_i^{t+\frac{1}{2}} + \gamma\left(s_i^{t+1} - \hat{\theta}_i^{t+1}\right)$ `// Gossip`

$q_i^t \leftarrow Q\left(\theta_i^{t+1} - \hat{\theta}_i^t\right)$ `// Compression`

send $(q_i^t, \lambda_i^{t+\frac{1}{2}})$ to $j \in \mathcal{N}(i)$ and receive $(q_j^t, \lambda_j^{t+\frac{1}{2}})$ from $j \in \mathcal{N}(i)$ `// Msgs exchange`

$\hat{\theta}_i^{t+1} \leftarrow q_i^t + \hat{\theta}_i^t$ `// Public variables update`

$s_i^{t+1} \leftarrow s_i^t + \sum_{j=1}^m w_{i,j} q_j$

$\lambda_i^{t+1} \leftarrow \sum_{j=1}^m w_{i,j} \lambda_j^{t+\frac{1}{2}}$ `// Dual variable averaging`

end

In order to obtain a final predictor with satisfactory performance for all local distributions $\{P_i\}_{i=1}^m$, the common objective is to learn global model which is distributionally robust with respect to the ambiguity set $\mathcal{P} := \left\{\sum_{i=1}^m \lambda_i P_i : \lambda_i \in \Delta^{m-1}\right\}$ where Δ^{m-1} denotes the $m-1$ probability simplex. As shown in [79], a network objective function that effectively works as proxy for this scope is given by

$$\min_{\theta \in \mathbb{R}^d} \max_{\lambda \in \Delta^{m-1}} \left(g(\theta, \lambda) := \frac{1}{m} \sum_{i=1}^m \underbrace{(\lambda_i f_i(\theta) + \alpha r(\lambda))}_{:= g_i(\theta, \lambda)} \right) \tag{4.3}$$

in which $r : \Delta^{m-1} \to \mathbb{R}$ is a strongly-concave regularizer and $\alpha \in \mathbb{R}^+$. For instance, in the empirical risk minimization framework in which each device i is endowed with a training set $\mathcal{D}_i \sim P_i^{\otimes n_i}$ and the overall number of training points is $n = \sum_i n_i$, a common choice of $r(\lambda)$ is $\chi^2(\lambda) := \sum_i \frac{(\lambda_i - n_i/n)^2}{n_i/n}$. In what follows, we refer to θ and λ as the primal and dual variables, respectively, and make the following fairly standard assumptions on the local functions g_i and the stochastic oracles available at the network devices.

Assumption 5. *Each function $g_i(\theta, \lambda)$ is differentiable in $\mathbb{R}^d \times \Delta^{m-1}$, L-smooth and μ-strongly concave in λ.*

Assumption 6. *Each device i has access to the stochastic gradient oracles $\nabla_\theta g_i(\theta, \lambda, \xi_i)$ and $\nabla_\lambda g_i(\theta, \lambda, \xi_i)$, with randomness w.r.t. ξ_i, which satisfy the following assumptions:*

- *Unbiasedness*

$$\mathbb{E}_{\xi_i}\left[\nabla_\theta g_i(\theta, \lambda, \xi_i)\right] = \nabla_\theta g_i(\theta, \lambda) \tag{4.4}$$

$$\mathbb{E}_{\xi_i}\left[\nabla_\lambda g_i(\theta,\lambda,\xi_i)\right] = \nabla_\lambda g_i(\theta,\lambda). \tag{4.5}$$

- *Bounded variance*

$$\mathbb{E}_{\xi_i}\left[\left\|\nabla_\theta g_i(\theta,\lambda,\xi_i) - \nabla_\theta g_i(\theta,\lambda)\right\|^2\right] \leq \sigma_\theta^2 \tag{4.6}$$

$$\mathbb{E}_{\xi_i}\left[\left\|\nabla_\lambda g_i(\theta,\lambda,\xi_i) - \nabla_\lambda g_i(\theta,\lambda)\right\|^2\right] \leq \sigma_\lambda^2. \tag{4.7}$$

- *Bounded magnitude*

$$\mathbb{E}_{\xi_i}\left[\left\|\nabla_\theta g_i(\theta,\lambda,\xi_i)\right\|^2\right] \leq G_\theta^2 \tag{4.8}$$

$$\mathbb{E}_{\xi_i}\left[\left\|\nabla_\lambda g_i(\theta,\lambda,\xi_i)\right\|^2\right] \leq G_\lambda^2. \tag{4.9}$$

The above assumption implies that each network device can query stochastic gradients that are unbiased, have finite variance, and have bounded second moment. The last assumption is rather strong but it is often made in distributed stochastic optimization [83, 94, 112].

4.4 Distributionally Robust Decentralized Learning Algorithm

Problem (4.3) entails solving a distributed minimax optimization problem in which, at every round, collaborating devices store a private value of the model parameters and the dual variable, which are potentially different from device to device. We denote the estimate of the primal and dual variables of device i at time t by θ_i^t and λ_i^t and the network estimates at time t as $\bar{\theta}^t = \frac{1}{m}\sum_{i=1}^m \theta_i^t$ and $\bar{\lambda}^t = \frac{1}{m}\sum_{i=1}^m \lambda_i^t$, respectively. The main challenge resulting from the decentralized implementation of the stochastic gradient descent/ascent algorithm consists in approaching a minimax solution or a stationary point (depending on the convexity assumption on the loss function) while concurrently ensuring convergence to a common global solution. To this end, the proposed procedure, given in Algorithm 2, alternates between a local update step and a consensus step. At each round, every device i queries the local stochastic gradient oracle and, in parallel, updates the model parameter θ_i by a gradient descent step with learning rate $\eta_\theta > 0$ and the dual variable λ_i by a projected gradient ascent one with learning rate $\eta_\lambda > 0$. Subsequently, a gossip strategy is used to share and average information between neighbors. In order to alleviate the communication burden of transmitting the vector of model parameters, which is typically high dimensional and contributes to the largest share of communication load, a compressed gossip step is employed. To implement the compressed communication, we consider the memory efficient version of CHOCO-GOSSIP [99] in which each device needs to store only two additional variables $\hat{\theta}_i$ and s_i, each of the same size as θ_i. The first one is a public version of θ_i, while the second is used to track the evolution of the weighted average, according to matrix W, of the public variables at the neighboring devices. Instead of transmitting θ_i, each device first computes an averaging step to update the value of the private value using the information about the public variables

encoded in $\hat{\theta}_i$ and s_i. It then computes q_i, a compressed representation of the difference between $\hat{\theta}_i$ and θ_i, and shares it with the neighboring devices to update the value of $\hat{\theta}_i$ and s_i used in the averaging step in the next round. As the number of participating devices is usually much smaller than the size of the model ($m \ll d$), the dual variable λ_i is updated sending uncompressed messages and then averaged according to matrix W. Note that AD-GDA implicitly assumes that collaborating parties are honest and for this reason it does not employ any countermeasure against malicious devices providing false dual variable information in order to steer the distributional robust network objective at their whim.

4.4.1 Convex loss function

We provide now a convergence guarantee for the solution output by Algorithm 2 for the case the loss function $\ell(\cdot)$ is convex in the model parameter θ. The result is given in the form of a sub-optimality gap bound for the function

$$\Phi(\theta) = g\left(\theta, \lambda^*(\theta)\right), \quad \lambda^*(\cdot) := \underset{\lambda \in \Delta^{m-1}}{\arg\max}\, g(\cdot, \lambda) \tag{4.10}$$

and it can be promptly derived from a primal-dual gap type of bound provided in the Appendix. In the bound we also refer to $\theta^*(\cdot) \in \arg\max_{\theta \in \mathbb{R}^d} g(\theta, \cdot)$.

Theorem 2. *Under Assumptions 5, 6, we have that for any $\theta^* \in \arg\min_\theta \Phi(\theta)$ the solution θ_o returned by Algorithm 2 with learning rates $\eta_\theta = \eta_\lambda = \frac{1}{\sqrt{T}}$ and consensus step size $\gamma = \frac{\rho^2\delta}{16\rho+\rho^2+4\beta^2+2\rho\beta^2-8\rho\delta}$ satisfies*

$$\begin{aligned}\mathbb{E}\left[\Phi(\theta_o) - \Phi(\theta^*)\right] \leq & \mathcal{O}\left(\frac{D_\theta + D_\lambda + G_\theta^2 + G_\lambda^2}{\sqrt{T}}\right) + \mathcal{O}\left(\frac{LD_\lambda G_\theta}{c\sqrt{T}} + \frac{LD_\theta G_\lambda}{\rho\sqrt{T}}\right) \\ & + \mathcal{O}\left(\frac{LG_\lambda^2}{\rho^2 T} + \frac{LG_\theta^2}{c^2 T}\right)\end{aligned} \tag{4.11}$$

where $D_\lambda := \max_t \mathbb{E}\left\|\bar{\lambda}^t - \lambda^*(\theta_o)\right\|$, $D_\theta := \max_t \mathbb{E}\left\|\bar{\theta}^t - \theta^*(\lambda_o)\right\|$ and $c = \frac{\rho^2\delta}{82}$.

The bound establishes a $\mathcal{O}(1/\sqrt{T})$ non-asymptotic optimality gap guarantee for the output solution. Compared to decentralized stochastic gradient descent (SGD) in the convex scenario, we obtain the same rate but with a dependency on the network topology and compression also in the lower order terms. Moreover, whenever θ and λ are constrained in convex sets, the diameter of the two can be used to explicitly bound D_θ and D_λ.

4.4.2 Non-convex loss function

We now focus on the case where the relation between the model parameters θ and the value of the loss function is non-convex. In this setting we provide a bound on the stationarity of the randomized solution, picked uniformly over time. In this setting,

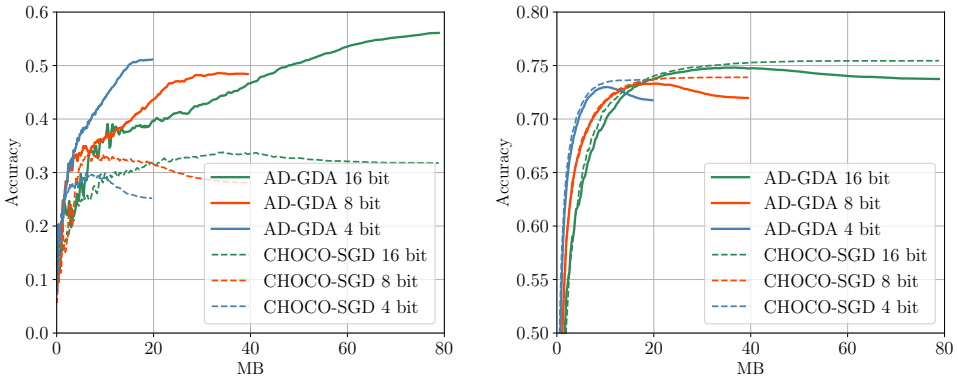
carefully tuning the relation between primal and dual learning rates is key to establish a convergent recursion (see B.2). This technical condition allows us to derive the following result.

Theorem 3. *Under Assumptions 5, 6, we have that the solution θ_o returned by Algorithm 2 with learning rates $\eta_\theta = \frac{\eta_\lambda}{16(\kappa+1)^2}$ and $\eta_\lambda = \frac{1}{2L\sqrt{T}}$ and consensus step size $\gamma = \frac{\rho^2\delta}{16\rho+\rho^2+4\beta^2+2\rho\beta^2-8\rho\delta}$ satisfies*

$$\frac{\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\Phi(\bar{\theta}^{t-1})\right\|^2\right]}{T} \leq \mathcal{O}\left(L\frac{\Delta\Phi^T}{\sqrt{T}} + \frac{L^2\kappa^2 D_\lambda^0}{2\sqrt{T}}\right) + \mathcal{O}\left(\frac{D_\lambda L G_\theta}{c\sqrt{T}} + \frac{\sigma_\theta^2+\kappa\sigma_\lambda^2}{m\sqrt{T}}\right) + \mathcal{O}\left(\frac{G_\theta^2}{c^2 T} + \frac{\kappa G_\lambda^2}{\rho^2 T}\right) + \frac{\sigma_\theta^2}{m} \tag{4.12}$$

where $\Delta\Phi^T = \mathbb{E}[\Phi(\bar{\theta}^0)] - \mathbb{E}[\Phi(\bar{\theta}^T)]$ and $c = \frac{\rho^2\delta}{82}$.

We note that the bound decreases at a rate $\mathcal{O}(1/\sqrt{T})$, except the last variance term which is non-vanishing. Nonetheless, whenever the variance of the stochastic gradient oracle for the primal variable is small or the number of participating devices is large, this term becomes negligible. Otherwise, at a cost of increased gradient complexity, each device can query the oracle $\mathcal{O}(1/\epsilon^2)$ times every round, average the results and make the stochastic gradient variance $\mathcal{O}(1/\epsilon^2)$. This procedure make the bound vanishing and leads to a gradient complexity matching the one of [113] given for the federated learning scenario.



(a) Worst-Case (b) Average

Figure 4.3: Average and worst-case accuracies of a fully connected neural network vs. number of transmitted bits using the random quantization compression scheme.

Table 4.1: Worst-case distribution accuracy attained by AD-GDA and CHOCO-SGD for different compression schemes.

	Quantization			Sparsification		
	16 bit	8 bit	4 bit	50%	25%	10%
Logistic AD-GDA	59.19 ± 2.05	57.43 ± 1.44	55.75 ± 2.09	57.05 ± 0.68	54.02 ± 1.14	51.51 ± 2.88
Logistic CHOCO-SGD	30.69 ± 0.96	30.06 ± 0.83	29.46 ± 0.05	30.28 ± 0.60	28.56 ± 0.54	26.39 ± 0.67
F.C. AD-GDA	54.99 ± 1.92	48.99 ± 2.30	47.08 ± 2.53	51.85 ± 2.11	43.65 ± 2.97	38.95 ± 3.21
F.C. CHOCO-SGD	30.83 ± 2.22	28.08 ± 2.50	28.01 ± 2.59	29.92 ± 2.54	27.11 ± 2.96	25.91 ± 3.20

4.5 Experiments

In this section, we empirically evaluate the capabilities of AD-GDA in producing a robust predictor for different learning models, communication network topologies, and message compression schemes. Lacking of a baseline for the distributionally robust fully decentralized setup, we compare the effectiveness of the proposed solution against the distributionally robust federated baseline (DRFA) [83] under similar communication constraints.

4.5.1 Setup

We perform our experiments using the Fashion-MNIST dataset [114] [2], a popular dataset made of images of 10 different clothing items, which is commonly used to test distributionally robust learners [79, 83]. In order to introduce data heterogeneity, samples are partitioned across the network devices using a class-wise split. Namely, we simulate a network of 10 devices, each storing data points coming from one of the 10 classes. In this setting, we train a logistic regression model and a two layer fully connected neural network with 25 hidden units in order to investigate both the convex and the non-convex case. In both cases, we use the SGD optimizer and, in order to ensure consensus at the end of the optimization process, we consider a geometrically decreasing learning rate $\eta_\theta^t = r^{-t}\eta_\theta^0$ with ratio $r = 0.995$ and initial value $\eta_\theta^0 = 1$. The metrics that we track are the final worst-device distribution accuracy and the average accuracy over the aggregated data samples of the network estimate $\bar{\theta}^t$.

4.5.2 Effect of compression

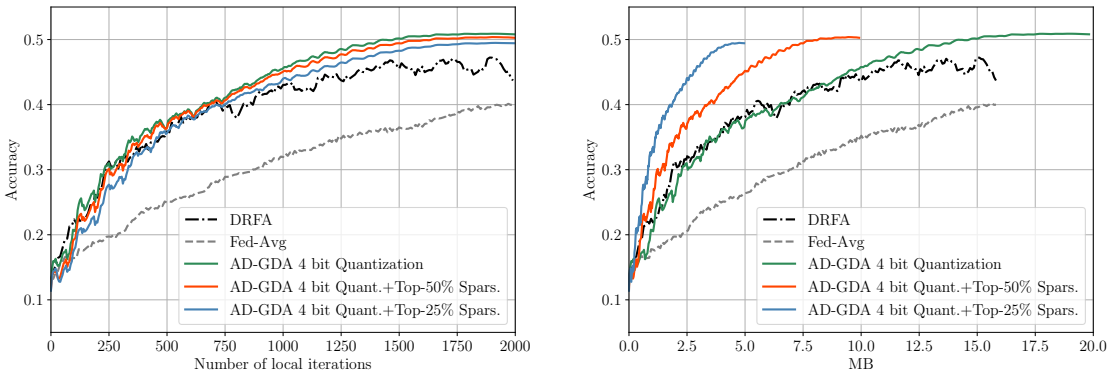
We assess the effect of compression with a fixed budget in terms of communication rounds by organizing devices in a ring topology and training the logistic model and the fully connected network for $T = 2000$ iterations. As representative of the unbiased compression operators, we consider the b-bit random quantization scheme for $b = \{16, 8, 4\}$ bit levels, while for the biased category we implement the top-K sparsification scheme saving $K = \{50\%, 25\%, 10\%\}$ of the original message components. For each compression scheme and compression level, we tune the consensus step size γ performing a grid search. We train the different model for 20 different random placements of the data shards across the devices using the distributionally robust and standard learning paradigms. In Table 4.1 we report the average worst-case accuracy attained by the final averaged model $\bar{\theta}^T$. AD-GDA almost doubles the worst-case accuracy compared to the not-robust baseline CHOCO-SGD [99]. This gain holds for both compression schemes and across different compression levels. For increased compression ratios, the worst-case accuracy degrades; however, for a comparable saving in communication bandwidth the unbiased quantization scheme results in superior performance than the biased sparsification compression operator. For a fixed optimization horizon compression degrades performance. Nonetheless, compression allows to obtain the same accuracy level with fewer transmitted bits as shown in Fig.

[2] The Fashion-MNIST dataset is released under the MIT License

Table 4.2: Worst-case distribution accuracy attained by AD-GDA and CHOCO-SGD for different network topologies.

	Top-10% Sparsification		4-bit Quantization	
	2D Torus	Mesh	2D Torus	Mesh
Log. AD-GDA	54.00 ± 0.61	54.07 ± 0.03	56.94 ± 0.38	57.11 ± 0.03
Log. CHOCO-SGD	26.82 ± 0.41	29.00 ± 0.02	30.82 ± 0.24	30.97 ± 0.03
F.C. AD-GDA	44.31 ± 2.47	45.21 ± 2.22	50.16 ± 1.85	50.80 ± 1.83
F.C. CHOCO-SGD	26.02 ± 2.29	26.38 ± 2.65	28.79 ± 2.22	28.96 ± 1.87

4.3a where we plot the average worst-case accuracy of the fully connected model as a function of the transmitted bits using the random quantization scheme. Furthermore, in Fig. 4.3b we compare the average accuracy of the robust predictor against the standard one. The price to pay in terms of average performance in order to ensure robustness of the predictor is modest and in the range of 2.5%.



(a) Worst-device accuracy vs. number of oracle calls

(b) Worst-device accuracy vs. number of transmitted bits

Figure 4.4: Comparison between distributionally robust federated averaging (DRFA), federated averaging (FedAvg) and the proposed algorithm (AD-GDA) for different compression techniques.

4.5.3 Effect of topology

We now turn to investigate the effect of device connectivity. Sparser communication topologies slow down the consensus process and therefore hamper the convergence of the algorithm. In the previous batch of experiments we considered a sparse ring topology, in which each device is connected to only two other devices. Here, we explore two other network configurations with a more favorable spectral gap. The communication topology with each device connected to other 4 devices and the mesh case, in which all devices communicate with each other. For these configurations we consider the 4-bit

Table 4.3: Testing accuracy attained at convergence for different regularization values α. The first two columns represent the accuracy when the model is tested on images produced by microscope 1 and microscope 2. The last column is the average accuracy when tested on a 50/50 test dataset.

	Microscope 1	Microscope 2	Mean
$\alpha = \infty$	65.86 ± 1.26	91.11 ± 0.63	78.48 ± 0.96
$\alpha = 1$	70.73 ± 1.33	84.30 ± 1.6	77.51 ± 1.51
$\alpha = 0.1$	73.30 ± 2.20	79.78 ± 2.30	76.54 ± 2.25
$\alpha = 0.01$	76.03 ± 1.45	79.02 ± 1.40	77.52 ± 1.43

quantization and top-10% sparsification compression schemes. In Table 4.2 we report the final worst-case performance for the different network configurations. As expected, network configurations with larger device degree lead to higher worst-case accuracy owing to smaller spectral gap which leads to the faster convergence rates.

4.5.4 Effect of regularization

Following a two-player game interpretation of the minimax optimization problem (4.3), the regularizer parameter $r(\lambda)$ reduces the freedom that an adversary has in choosing the weighting vector λ so as to maximize the training loss at every iteration t. As a result, the less constrained the adversary, the larger the emphasis on the worse performing devices. In the following we consider a regularizer of the form $\chi^2(\lambda) := \sum_i \frac{(\lambda_i - n_i/n)^2}{n_i/n}$ and study the effect of the regularization parameter α on the robustness of the yielded solution. For this specific experiment, we consider the biological dataset COOS7 [2] that contains images of mouse cells captured using two different microscopes. We consider a network of 5 collaborating parties (e.g. research labs) connected according to a ring topology. We endow 4 of these parties with images sampled using microscope 1, while we give to the remaining one images taken from microscope 2. We train the model using AD-GDA for $\alpha = \{\infty, 1, 0.1, 0.01\}$ and report the average accuracy along with the 95% confidence intervals in Table 4.3. For the case $\alpha \to \infty$, which corresponds to CHOCO-SGD, we observe a large test accuracy gap between images taken from microscope 1 and microscope 2, with the classifier attaining 25% higher accuracy on the latter. This accuracy mismatch showcases how standard decentralized optimization schemes are unable to guarantee uniform performance across participating parties. On the other hand, using AD-DGA and smaller regularization parameters, the gap between the two instruments is effectively reduced, eventually hitting a 3% performance mismatch for $\alpha = 0.01$. At the same time, the improved fairness brought by AD-GDA does not significantly hamper the average performance of the model when tested on both instruments.

4.5.5 Comparison with the federated baseline

Lacking of a term of comparison for the distributionally robust fully decentralized setting, we consider the communication-efficient distributionally robust federated learning scheme (DRFA) [83] and the standard federated averaging (FedAvg) [58]. In the federated scenario, network devices are connected according to a star topology, with the star center representing the central aggregator. Communication efficiency is obtained allowing network devices to perform multiple local updates of the primal variable between subsequent synchronization rounds at the central aggregator. We run DRFA allowing devices to perform 10 local gradient step before sending its local models for the distributionally robust averaging steps and we consider half user participation at each round. To have the same per round communication cost, we run FedAvg allowing 10 local gradient step between aggregations, but considering full user participation. Recall that the random sketching technique employed by DRFA requires devices to send two model updates to the central aggregator at each round therefore doubling the communication cost. To match this setting from a communication standpoint, we consider AD-GDA with a mesh network topology. Moreover, in order to have comparable communication cost per device, we consider the quantization compression operator with $b = 4$ in combination with the sparsification scheme saving $K = \{25\%, 50\%\}$ components. In Fig. 4.4a we compare the worst-case distribution accuracy attained by the different algorithms on the Fashion-MNIST dataset (with data split as in Sec.4.5.1) as a function of the number of stochastic gradients that each device needs to query. DRFA and AD-GDA have similar gradient complexity while FedAvg needs considerably more gradient calls to obtain the same worst-case performance. In Fig. 4.4b we compare the communication efficiency of the algorithms and we report the worst-case accuracy versus the average number of bits transmitted by each device. For the same communication budget, AD-GDA can attain higher worst-case distribution accuracy compared to DRFA transmitting only a fraction of bits compared to the federated counterparts.

4.6 Conclusion

We provided a provably convergent decentralized single-loop gradient descent/ascent algorithm to tackle the distributionally robust learning problem over a network of collaborating devices with heterogeneous local data distributions. Differently from previously proposed solutions, which are limited to the federated scenario with a central coordinator, our algorithm restrains devices to D2D communication and attains communication efficiency by employing compressed communication techniques. Experiments showed that the proposed solution produces distributionally robust predictors with higher worst-case accuracy, while it attains superior communication efficiency compared to the previously proposed algorithms that reduce the communication load allowing multiple local updates at participating devices. The proposed framework is a promising decentralized learning solution over edge devices in B5G IoT networks.

Chapter 5

User-Centric Federated Learning

Data heterogeneity across participating devices poses one of the main challenges in federated learning as it has been shown to greatly hamper its convergence time and generalization capabilities. In this chapter we address this limitation by enabling personalization using multiple user-centric aggregation rules at the parameter server. Our approach potentially produces a personalized model for each user at the cost of some extra downlink communication overhead. To strike a trade-off between personalization and communication efficiency, we propose a broadcast protocol that limits the number of personalized streams while retaining the essential advantages of our learning scheme. Through simulation results, our approach is shown to enjoy higher personalization capabilities, faster convergence, and better communication efficiency compared to other competing baseline solutions.

5.1 Introduction

Federated learning [58] has seen great success, being able to solve distributed learning problems in a communication-efficient and privacy-preserving manner. Specifically, federated learning provides to clients (e.g. smartphones, IoT devices, and organizations) the possibility of collaboratively train a model under the orchestration of a parameter server (PS) by iteratively aggregating locally optimized models and without off-loading local data [115]. The original aggregation policy was implemented by Federated Averaging (FedAvg) [58], has been devised under the assumption that clients' local datasets are statistically identical, an assumption that is hardly met in practice. In fact, clients typically store datasets that are statistically heterogeneous and different in size [116], and are mainly interested in learning models that generalize well over their local data distribution through collaboration. Generally speaking, FedAvg exhibits slow convergence and poor generalization capabilities in such non-IID setting [117]. To address these limitations, a large body of literature deals with personalization as a technique to reduce the detrimental effect of non-IID data. A straightforward solution consists in producing adapted models at a device scale by local fine-tuning procedures. Borrowing ideas from Model Agnostic Meta-Learning (MAML) [118], federated learning can be exploited in order to find a launch model that can be later personalized at each device using

few gradient iterations [119, 120]. Alternatively, local adaptation can be obtained by tuning only the last layer of a globally trained model [121] or by interpolating between a global model and locally trained ones [122, 123]. However, these methods can fail at producing models with an acceptable generalization performance even for synthetic datasets [124]. Adaptation can also be obtained leveraging user data similarity to personalize the training procedure. For instance, a Mixture of Experts formulation has been considered to learn a personalized mixing of the outputs of a commonly trained set of models [125]. Similarly, [126] proposed a distributed Expectation-Maximization (EM) algorithm concurrently converges to a set of shared hypotheses and a personalized linear combination of them at each device. Furthermore, [127] proposed a personalized aggregation rule at the user side based on the validation accuracy of the locally trained models at the different devices. In order to be applicable, these techniques need to strike a good balance between communication overhead and the amount of personalization in the system. In fact, if on one hand, the expressiveness of the mixture is proportional to the number of mixed components; on the other, the communication load is linear in this quantity. Clustered Federated Learning (CFL) measures the similarity among the model updates during the optimization process in order to lump together users in homogeneous groups. For example, [116, 128] proposed a hierarchical strategy in which the original set of users is gradually divided into smaller groups and, for each group, the federated learning algorithm is branched in a new decoupled optimization problem.

In this chapter, we propose a different approach to achieve personalization by allowing multiple user-centric aggregation strategies at the PS. The mixing strategies account for the existence of heterogeneous clients in the system and exploit estimates of the statistical similarity among clients that are obtained at the beginning of the federated learning procedure. Furthermore, the number of distinct aggregation rules — also termed personalized streams — can be fixed in order to strike a good trade-off between communication and learning efficiency. Finally, we provide simulation results for different scenarios and demonstrate that our approach exhibits faster convergence, higher personalization capabilities, and communication efficiency compared to other popular baseline algorithms.

5.2 Learning with heterogeneous data sources

In this section, we provide theoretical guarantees for learners that combine data from heterogeneous data distributions. The set-up mirrors the one of personalized federated learning and the results are instrumental to derive our user-centric aggregation rule. In the following, we limit our analysis to the discrepancy distance (5.4) but it can be readily extended to other divergences [129].

In the federated learning setting, the weighted combination of the empirical loss terms of the collaborating devices represents the customary training objective. Namely, in a distributed system with m nodes, each endowed with a dataset $\mathcal{D}_i$ of n_i IID samples from a local distribution P_i, the goal is to find a predictor $f : \mathcal{X} \to \hat{\mathcal{Y}}$ from a hypothesis

class $\mathcal{F}$ that minimizes

$$L(f, \vec{w}) = \sum_{i=1}^{m} \frac{w_i}{n_i} \sum_{(x,y)\in\mathcal{D}_i} \ell(f(x), y) \tag{5.1}$$

where $\ell : \hat{\mathcal{Y}} \times \mathcal{Y} \to \mathbb{R}^+$ is a loss function and $\vec{w} = (w_1, \ldots, w_m)$ is a weighting scheme. In case of identically distributed local datasets, the typical weighting vector is $\vec{w} = \frac{1}{\sum_i n_i}(n_1, \ldots, n_m)$, the relative fraction of data points stored at each device. This particular choice minimizes the variance of the aggregated empirical risk, which is also an unbiased estimate of the local risk at each node in this scenario. However, in the case of heterogeneous local distributions, the minimizer of $\vec{w}$-weighted risk may transfer poorly to certain devices whose target distribution differs from the mixture $P_{\vec{w}} = \sum_{i=1}^{m} w_i P_i$. Furthermore, it may not exists a single weighting strategy that yields a universal predictor with satisfactory performance for all participating devices. To address the above limitation of a universal model, personalized federated learning allows adapting the learned solution at each device. In order to better understand the potential benefits and drawbacks coming from the collaboration with statistically similar but not identical devices, let us consider the point of view of a generic node i that has the freedom of choosing the degree of collaboration with the other devices in the distributed system. Namely, identifying the degree of collaboration between node i and the rest of users by the weighting vector $\vec{w}_i = (w_{i,1}, \ldots, w_{i,m})$ (where $w_{i,j}$ defines how much node i relies on data from user j), we define the personalized objective for user i

$$L(f, \vec{w}_i) = \sum_{j=1}^{m} \frac{w_{i,j}}{n_j} \sum_{(x,y)\in\mathcal{D}_j} \ell(f(x), y) \tag{5.2}$$

and the resulting personalized model

$$\hat{f}_{\vec{w}_i} = \arg\min_{f\in\mathcal{F}} L(f, \vec{w}_i). \tag{5.3}$$

We now seek an answer to: *"What's the proper choice of $\vec{w}_i$ in order to obtain a personalized model $\hat{f}_{\vec{w}_i}$ that performs well on the target distribution P_i?"*. This question is deeply tied to the problem of domain adaptation, in which the goal is to successfully aggregate multiple data sources in order to produce a model that transfers positively to a different and possibly unknown target domain. In our context, the dataset $\mathcal{D}_i$ is made of data points drawn from the target distribution P_i and the other devices' datasets provide samples from the sources $\{P_j\}_{j\neq i}$. Leveraging results from domain adaptation theory [130], we provide learning guarantees on the performance of the personalized model $\hat{f}_{\vec{w}_i}$ to gauge the effect of collaboration that we later use to devise the weights for the user-centric aggregation rules.

In order to avoid negative transfer, it is crucial to upper bound the performance of the predictor w.r.t. to the target task. The discrepancy distance introduced in [30] provides a measure of similarity between learning tasks that can be used to this end. For

a hypothesis set of functions $\mathcal{F} : \mathcal{X} \to \hat{\mathcal{Y}}$ and two distributions P, Q on $\mathcal{X}$, the discrepancy distance is defined as

$$d_{\mathcal{F}}(P, Q) = \sup_{f,f' \in \mathcal{F}} \left| \mathbb{E}_{x \sim P} \left[\ell(f, f') \right] - \mathbb{E}_{x \sim Q} \left[\ell(f, f') \right] \right| \tag{5.4}$$

where we streamlined notation denoting $f(x)$ by f. For bounded and symmetric loss functions that satisfy the triangular inequality, the previous quantity allows to obtain the following inequality

$$\mathbb{E}_{(x,y)\sim P}[\ell(f, y)] \leq \mathbb{E}_{(x,y)\sim Q}[\ell(f, y)] + d_{\mathcal{F}}(P, Q) + \gamma \tag{5.5}$$

where $\gamma = \inf_{f \in \mathcal{F}} \left(\mathbb{E}_{(x,y)\sim P}[\ell(f, y)] + \mathbb{E}_{(x,y)\sim Q}[\ell(f, y)] \right)$. We can exploit the inequality to obtain the following risk guarantee for $\hat{f}_{\vec{w}_i}$ w.r.t the true minimizer f^* of the risk for the distribution P_i.

Theorem 4. *For a loss function ℓ B-bounded range, symmetric and satisfying the triangular inequality, with probability $1 - \delta$ the function $f_{\vec{w}_i}$ satisfies*

$$\begin{aligned} E_{z\sim P_i}[\ell(\hat{f}_{\vec{w}_i}, z)] - E_{z\sim P_i}[\ell(f^*, z)] \leq & \sqrt{\sum_{j=1}^{m} \frac{w_{i,j}^2}{n_j}} \left(\sqrt{\frac{2d}{\sum_i n_i} \log\left(\frac{e \sum_i n_i}{d} \right)} + \sqrt{\log\left(\frac{2}{\delta} \right)} \right) \\ & + 2 \sum_{j=1}^{m} w_{i,j} d_{\mathcal{F}}(P_i, P_j) + 2\gamma \end{aligned} \tag{5.6}$$

where $\gamma = \min_{f \in \mathcal{F}} \left(E_{z\sim P_i}[\ell(f, z)] + E_{z\sim P_{\vec{w}_i}}[\ell(f, z)] \right)$ and d is the VC-dimension of the function space resulting from the composition of $\mathcal{F}$ and ℓ.

Recently, an alternative bound based on an information theoretic notion of dissimilarity, the Jensen-Shannon divergence, has been proposed [131]. It is based on less restrictive constraints, as it only requires the loss function $\ell(f, Z)$ to be sub-Gaussian of some parameter σ for all $f \in \mathcal{F}$, and therefore whenever $\ell(\cdot)$ is bounded, the requirement is automatically satisfied. Measuring similarity by the Jensens-Shannon divergence the following inequality is available

$$E_{X\sim P}[X] \leq E_{X\sim Q}[X] + \beta\sigma^2 + \frac{D_{JS}(P||Q)}{\beta} \quad \text{for } \beta > 0 \tag{5.7}$$

where $D_{JS}(P\|Q) = \mathrm{KL}\left(P \middle\| \frac{P+Q}{2} \right) + \mathrm{KL}\left(Q \middle\| \frac{P+Q}{2} \right)$. Exploiting the above inequality we obtain the following estimation error bound.

Theorem 5. *For a loss function ℓ B-bounded range, the function $f_{\vec{w}_i}$ satisfies*

$$\begin{aligned} E_{z\sim P_i}[\ell(\hat{f}_{\vec{w}_i}, z)] - E_{z\sim P_i}[\ell(f^*, z)] \leq & B \sqrt{\sum_{j=1}^{m} \frac{w_{i,j}^2}{n_j}} \left(\sqrt{\frac{2d}{\sum_i n_i} \log\left(\frac{e \sum_i n_i}{d} \right)} + \sqrt{\log\left(\frac{2}{\delta} \right)} \right) \\ & + B \sqrt{2 \sum_{j=1}^{m} w_{i,j} D_{JS}(P_i||P_j)} \end{aligned} \tag{5.8}$$

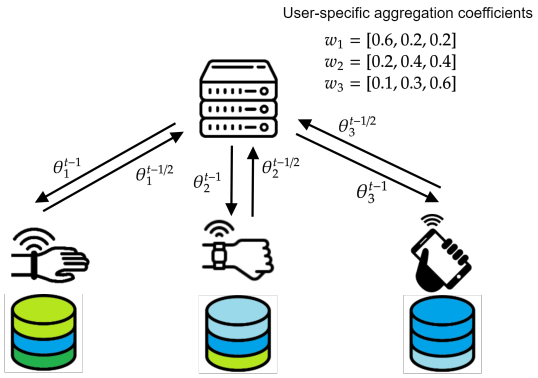


Figure 5.1: Personalized Federated Learning with user-centric aggregates at round t.

Proof of Theorem 4 and 5: In the Appendix C.1.
The theorems highlights that a fruitful collaboration should strike a balance between the bias terms due to dissimilarity between local distribution and the risk estimation gains provided by the data points of other nodes. Minimizing the upper bound in Th. 4,5 with respect to the user-specific weights, and using the optimal weights in our aggregation rule seems an appealing solution to tackle the data heterogeneity during training; however, the distance terms ($d_{\mathcal{F}}(P_i, P_k)$ and $D_{JS}(P_i||P_j)$) are difficult to compute, especially under the privacy constraints that federated learning imposes. For this reason, in the following we consider a heuristic method based on the similarity of the readily available users' model updates to estimate the collaboration coefficients.

5.3 User-centric aggregation

For a suitable hypothesis class parametrized by $\theta \in \mathbb{R}^d$, federated learning approaches use an iterative procedure to minimize the aggregate loss (5.1) with $\vec{w} = \frac{1}{\sum_i n_i}(n_1, \ldots, n_m)$. At each round t, the PS broadcasts the parameter vector θ^{t-1} and then combines the locally optimized models by the clients $\{\theta_i^{t-1}\}_{i=1}^m$ according to the following aggregation rule

$$\theta^t \leftarrow \sum_{i=1}^{m} \frac{n_i}{\sum_{j=1}^m n_j} \theta_i^{t-1}.$$

As mentioned in Sec. 5.2, this aggregation rule has two shortcomings: it does not take into account the data heterogeneity across users, and it is bounded to produce a single solution. For this reason, we propose a user-centric model aggregation scheme that takes into account the data heterogeneity across the different nodes participating in training and aims at neutralizing the bias induced by a universal model. Our proposal generalizes the naïve aggregation of FedAvg, by assigning a unique set of mixing coefficients $\vec{w}_i$ to each user i, and consequently, a user-specific model aggregation at the PS side. Namely,

at the PS side, the following set of user-centric aggregation steps are performed

$$\theta_i^t \leftarrow \sum_{j=1}^{m} w_{i,j}\theta_j^{t-1/2} \quad \text{for } i = 1, \dots, m \tag{5.9}$$

where now, $\theta_j^{t-1/2}$ is the locally optimized model at node j starting from θ_j^{t-1}, and θ_i^t is the user-centric aggregated model for user i at communication round t.

As we elaborate next, the mixing coefficients are heuristically defined based on a distribution similarity metric and the dataset size ratios. These coefficients are calculated before the start of federated training. The similarity score we propose is designed to favor collaboration among similar users and takes into account the relative dataset sizes, as more intelligence can be harvested from clients with larger data availability. Using these user-centric aggregation rules, each node ends up with its own personalized model that yields better generalization for the local data distribution. It is worth noting that the user-centric aggregation rule does not produce a minimizer of the user-centric aggregate loss given by (5.2). At each round, the PS aggregates model updates computed starting from a different set of parameters. Nonetheless, we find it to be a good approximation of the true update since personalized models for similar data sources tend to propagate in a close neighborhood. The aggregation in [127] capitalizes on the same intuition.

5.3.1 Computing the collaboration coefficients

Computing the discrepancy distance (5.4) can be challenging in high-dimension, especially under the communication and privacy constraints imposed by federated learning. For this reason, we propose to compute the mixing coefficient based on the relative dataset sizes and the distribution similarity metric given by

$$\begin{aligned}\Delta_{i,j}(\hat{\theta}) =& \left\| \frac{1}{n_i} \sum_{(x,y)\in\mathcal{D}_i} \nabla\ell(f_{\hat{\theta}}, y) - \frac{1}{n_j} \sum_{(x,y)\in\mathcal{D}_j} \nabla\ell(f_{\hat{\theta}}, y) \right\|^2 \\ \approx& \left\| \mathbb{E}_{z\sim P_i} \nabla\ell(f_{\hat{\theta}}, y) - \mathbb{E}_{z\sim P_j} \nabla\ell(f_{\hat{\theta}}, y) \right\|^2\end{aligned}$$

where the quality of the approximation depends on the number of samples n_i and n_j. The mixing coefficients for user i are then set to the following normalized exponential function

$$w_{i,j} = \frac{\frac{n_j}{n_i} e^{-\frac{1}{2\sigma_i\sigma_j}\Delta_{i,j}(\hat{\theta})}}{\sum_{j'=1}^{m} \frac{n_{j'}}{n_i} e^{-\frac{1}{2\sigma_i\sigma_{j'}}\Delta_{i,j'}(\hat{\theta})}} \quad \text{for } j = 1, \dots, m. \tag{5.10}$$

The mixture coefficients are calculated at the PS during a special round prior to federated training. During this round, the PS broadcasts a common model denoted $\hat{\theta}$ to the users, which compute the full gradient on their local datasets. At the same time, each node i locally estimates the value σ_i^2 partitioning the local data in K batches $\{\mathcal{D}_i^k\}_{k=1}^K$ of size

n_k and computing

$$\sigma_i^2 = \frac{1}{K}\sum_{k=1}^{K}\left\|\frac{1}{n_k}\sum_{(x,y)\in\mathcal{D}_i^k}\nabla\ell(f_{\hat{\theta}},y) - \frac{1}{n_i}\sum_{(x,y)\in\mathcal{D}_i}\nabla\ell(f_{\hat{\theta}},y)\right\|^2 \tag{5.11}$$

where σ_i^2 is an estimate of the gradient variance computed over local datasets $\mathcal{D}_i^k$ sampled from the same target distribution. Once all the necessary quantities are computed, they are uploaded to the PS, which proceeds to calculate the mixture coefficients and initiates the federated training using the custom aggregation scheme given by (5.9). Note that the proposed heuristic embodies the intuition provided by Th. 2. In fact, in the case of homogeneous users, it falls back to the standard FedAvg aggregation rule, while in the case of node i has an infinite amount of data it degenerates to the local learning rule which is optimal in that case.

5.3.2 Reducing the communication load

A full-fledged personalization by the means of the user-centric aggregation rule (5.9) would introduce a m-fold increase in communication load during the downlink phase as the original broadcast transmission is replaced by unicast ones. Although from a learning perspective the user-centric learning scheme is beneficial, it is also possible to consider overall system performance from a learning-communication trade-off point of view. The intuition is that, for small discrepancies between the user data distributions, the same model transfer positively to statistically similar devices. In order to strike a suitable trade-off between learning accuracy and communication overhead we hereby propose to adaptively limit the number of personalized downlink streams. In particular, for a number of personalized models m_t, we run a k-means clustering scheme with $k = m_t$ over the set of collaboration vectors $\{w_i\}_{i=1}^m$ and we select the centroids $\{\hat{w}_i\}_{i=1}^{m_t}$ to implement the m_t personalized streams. We then proceed to replace the unicast transmission with group broadcast ones, in which all users belonging to the same cluster c receive the same personalized model $\hat{w}_c$. Choosing the right value for the number of personalized streams is critical in order to save communication bandwidth but at the same time obtain satisfactory personalization capabilities. It can be experimentally shown that clustering quality indicators such as the Silhouette score over the user-centric weights can be used to guide the search for the suitable number of streams m_t.

5.3.3 Choosing the number of personalized streams

Choosing an insufficient number of personalized streams can yield unsatisfactory performance, while concurrently learning many models can prohibitively increase the communication load of personalized federated learning. Therefore, properly tuning this free parameter is essential in order to obtain a well-performing but still practical algorithm. Being agnostic w.r.t. the underlying data generating distributions at the devices, it does not exist a universal number of personalized streams that fits all problems. However, we now illustrate that the silhouette coefficient, a quality measure of the clustering, provides

Algorithm 3: Silhouette based scoring

Input : Collab. vectors $\{\vec{w}_i\}_{i=1}^m$ from (5.10) and trade-off function $c(k, s_k)$.
Output: Number of clusters m_t and personalized streams
for $k = 1, 2, \ldots, m$ **do**
  $\mathcal{C}_k \leftarrow K$-means clustering of $\{\vec{w}_i\}_{i=1}^m$
  $s_k \leftarrow$ the silhouette score of $s(\mathcal{C})$
end
return $m_t = \arg\max_{k=1,\ldots,m} c(k, s_k)$ and cluster centers of $\mathcal{C}_{m_t}$

a rule of thumb to choose the number of personalized streams. In order to compute the silhouette score of a clustering $\mathcal{C}_1, \ldots, \mathcal{C}_{m_t}$ of the clustering we define the intra-cluster similarity of the collaboration vector $\vec{w}_i \in \mathcal{C}_k$ as

$$a(\vec{w}_i) = \frac{1}{|\mathcal{C}_j| - 1} \sum_{\vec{w}_j \in \mathcal{C}_k, \vec{w}_j \neq \vec{w}_i} \|\vec{w}_j - \vec{w}_i\| \tag{5.12}$$

and the smallest mean distance between the collaboration vector $\vec{w}_i \in \mathcal{C}_k$ and the closest cluster

$$b(\vec{w}_i) = \min_{\mathcal{C}_j \neq \mathcal{C}_k} \frac{1}{|\mathcal{C}_j|} \sum_{\vec{w}_j \in \mathcal{C}_j} \|\vec{w}_j - \vec{w}_i\|. \tag{5.13}$$

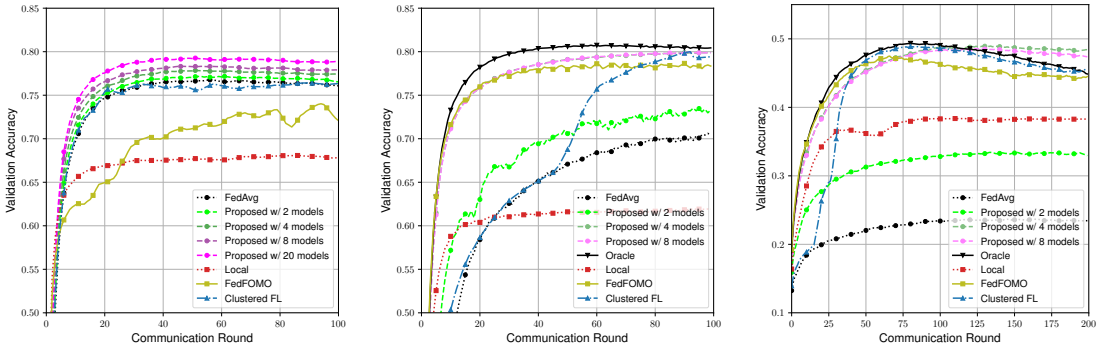
The average silhouette score s is then defined as

$$s(\mathcal{C}) = \frac{1}{m} \sum_{i=1}^{m} \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{5.14}$$

and it is a number in the range $[-1, 1]$, directly proportional to the quality of the clustering. In turn, a good clustering of the collaboration vectors $\{\vec{w}_i\}_{i=1}^m$ implies that users belonging to the same clusters are similar, and that the centroid $\vec{c}_j$ is a good approximation of the collaboration coefficient of users in $\mathcal{C}_j$. Consequently, whenever the silhouette score is large, the loss in terms of personalization performance resulting from the reduced number of aggregation rules compared to the full-fledged personalization system is modest. For this reason, the silhouette score provides a proxy to the inference performance and at the same time it allows to trade-off communication load and personalization capabilities in a principled way. In Algorithm 3 we provide the pseudocode of the procedure that autonomously chooses the optimal number of personalized streams m_t based on a communication-personalization trade-off function $c(k, s) : \mathbb{N} \times [-1, 1] \to \mathbb{R}^+$ scoring the utility of pairs of the systems based on the number of user-centric rules and the resulting silhouette scores. The function $c(k, s)$ is a system dependent function typically decreasing in k and increasing in s_k.

5.4 Experiments

We now provide a series of experiments to showcase the personalization capabilities and communication efficiency of the proposed algorithm.



(a) EMNIST + label shift

(b) EMNIST + label and covariate shift

(c) CIFAR10 + concept shift

Figure 5.2: Evolution of the average validation accuracy in the three simulation scenarios.

5.4.1 Set-up

In our simulation we consider a handwritten character/digit recognition task using the EMNIST dataset [132] and an image classification task using the CIFAR-10 dataset [133]. Data heterogeneity is induced by splitting and transforming the dataset in a different fashion across the group of devices. In particular, we analyze three different scenarios:

- **Character/digit recognition with user-dependent label shift** in which 10k EMNIST data points are split across 20 users according to their labels. The label distribution follows a Dirichlet distribution with parameter 0.4, as in [126, 134].
- **Character/digit recognition with user-dependent label shift and covariate shift** in which 100k samples from the EMNIST dataset are partitioned across 100 users each with a different label distribution, as in the previous scenario. Additionally, users are clustered in 4 group and at each group images are rotated of $\{0°, 90°, 180°, 270°\}$ respectively.
- **Image classification with user-dependent concept shift** in which the CIFAR-10 dataset is distributed across 20 users which are grouped in 4 clusters, for each group we apply a different random label permutation.

For each scenario, we aim at solving the task at hand by leveraging the distributed and heterogeneous datasets. We compare our algorithm against four different baselines: FedAvg, local learning, CFL [116] and FedFomo [127]. In all scenarios and for all algorithms, we train a LeNet-5 convolutional neural network [135] using a stochastic gradient descent optimizer with a fixed learning rate $\eta = 0.1$ and momentum $\beta = 0.9$.

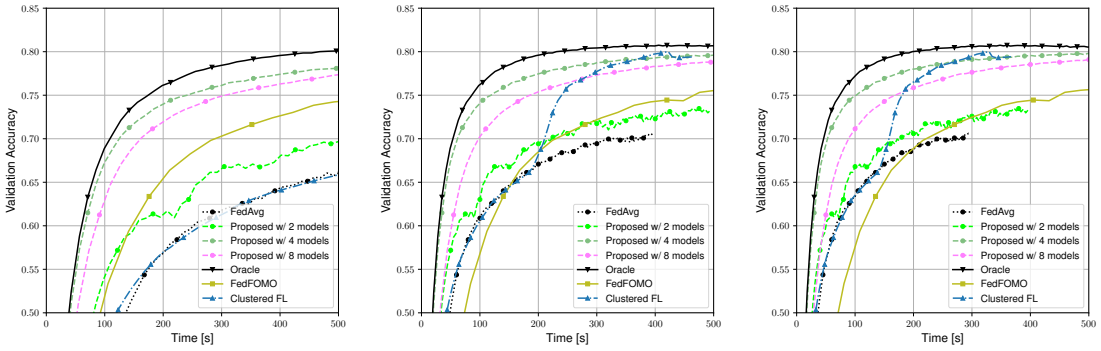
5.4.2 Personalization performance

We now report the average accuracy over 5 trials attained by the different approaches. We also study the personalization performance of our algorithm when we restrain the

Table 5.1: Worst user performance averaged over 5 experiments in the three simulation scenarios.

	Local	FedAvg	Oracle	CFL	FedFOMO	Proposed
EMNIST + label shift	58.8	68.9	-	70.3	70.0	**73.2** ($k = 20$)
EMNIST + cov. & label shift	56.0	67.5	77.4	76.1	73.6	**76.4** ($k = 4$)
CIFAR10 + concept shift	35.7	19.6	49.1	48.6	45.5	**48.8** ($k = 4$)

overall number of personalized streams, namely the number of personalized models that are concurrently learned. In Fig.5.2a we report the average validation accuracy in the EMNIST label shift scenario. We first notice that in the case of label shift, harvesting intelligence from the datasets of other users amounts to a large performance gain compared to the localized learning strategy. This indicates that data heterogeneity is moderate and collaboration is fruitful. Nonetheless, personalization can still provide gains compared to FedAvg. Our solution yields a validation accuracy which is increasing in the number of personalized streams. Allowing maximum personalization, namely a different model at each user, we obtain a 3% gain in the average accuracy compared to FedAvg. CFL is not able to transfer intelligence among different groups of users and attains performance similar to the FedAvg. This behavior showcases the importance of soft clustering compared to the hard one for the task at hand. We find that FedFOMO, despite excelling in the case of strong statistical heterogeneity, fails to harvest intelligence in the label shift scenario. In Fig.5.2b we report the personalization performance for the second scenario. In this case, we also consider the oracle baseline, which corresponds to running 4 different FedAvg instances, one for each cluster of users, as if the 4 groups of users were known beforehand. Different from the previous scenario, the additional shift in the covariate space renders personalization necessary in order to attain satisfactory performance. In fact, the oracle training largely outperforms FedAvg. Furthermore, as expected, our algorithm matches the oracle final performance when the number of personalized streams is 4 or more. Also, CLF and FedFOMO are able to correctly identify the 4 clusters. However, the former exhibits slower convergence due to the hierarchical clustering over time while the latter plateaus to a lower average accuracy level. We turn now to the more challenging CIFAR-10 image classification task. In Fig.5.2c we report the average accuracy of the proposed solution for a varying number of personalized streams, the baselines, and the oracle solution. As expected, the label permutation renders collaboration extremely detrimental as the different learning tasks are conflicting. As a result, local learning provides better accuracy than FedAvg. On the other hand, personalization can still leverage data among clusters and provide gains also in this case. Our algorithm matches the oracle performance for a suitable number of personalized streams. This scenario is particularly suitable for hard clustering, which isolates conflicting data distributions. As a result, CFL matches the proposed solution. FedFOMO promptly detects clusters and therefore quickly converges, but it attains lower average accuracy compared to the proposed solution.



(a) $\rho = 4, T_{min} = T_{dl} = \frac{1}{\mu}$ (b) $\rho = 2, T_{min} = T_{dl}, \frac{1}{\mu} = 0$ (c) $\rho = 1, T_{min} = T_{dl}, \frac{1}{\mu} = 0$

Figure 5.3: Evolution of the average validation accuracy against time normalized w.r.t. T_{dl} for the three different systems.

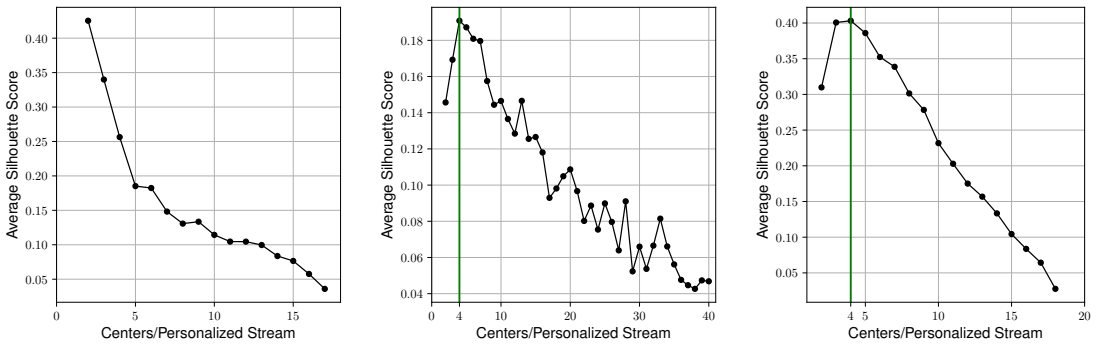
The performance reported so far is averaged over users and therefore fails to capture the existence of outliers performing worse than average. In order to assess the fairness of the training procedure, in Table 5.1 we report the worst user performance in the federated system. The proposed approach produces models with the highest worst case in all three scenarios.

5.4.3 Silhouette score

In Fig. 5.4 we plot the average silhouette score obtained by the k-means algorithm when clustering the federated users based on the procedure proposed in Sec. 5.3.3. In the labels shift scenario, for which we have seen that a universal model performs almost as good as the personalized ones, the silhouette scores monotonically decreases with k. In fact, in this simulation setting, a natural cluster-like structure among clients tasks does not exist. On the other hand, in the covariate shift and the concept shift scenarios, the silhouette score peaks around $k = 4$. In Sec. 5.4.2 this has has shown to be the minimum number of personalized models necessary to obtain satisfactory personalization performance in the system. This behaviour of the silhouette score is expected and desired, in this case the number of clusters matches exactly the number of underlying different tasks among the participants in FL that was induced by the rotation of the covariates and the permutation of the labels. We then conclude that the silhouette score provides meaningful information to tune the number of user-centric aggregation rules prior to training.

5.4.4 Communication Efficiency

Personalization comes at the cost of increased communication load in the downlink transmission from the PS to the federated user. In order to compare the algorithm convergence time, we parametrize the distributed system using two parameters. We define by $\rho = \frac{T_{ul}}{T_{dl}}$ the ratio between model transmission time in uplink (UL) and downlink (DL). Typical values of ρ in wireless communication systems are in the $[2, 4]$ range



(a) EMNIST label shift

(b) EMNIST label and covariate shift

(c) CIFAR10 concept shift

Figure 5.4: Average silhouette scores of the k-means clustering in the three scenarios. In the last two scenarios, in which user inherently belongs to 4 different cluster, the scores indicates the necessity of at least 4 personalized streams.

because of the larger transmitting power of the base station compared to the edge devices. Furthermore, to account for unreliable computing devices, we model the random computing time T_i at each user i by a shifted exponential r.v. with a cumulative distribution function

$$P[T_i > t] = 1 - \mathbb{1}(t \geq T_{min}) \left[1 - e^{-\mu(t - T_{min})}\right]$$

where T_{min} representing the minimum possible computing time and $1/\mu$ being the average additional delay due to random computation impairments. Therefore, for a population of m devices, we then have

$$T_{comp} = \mathbb{E}\left[\max\{T_1, \ldots, T_m\}\right] = T_{min} + \frac{H_m}{\mu}$$

where H_m is the m-th harmonic number. To study the communication efficiency we consider the simulation scenario with the EMNIST dataset with label and covariate shift. In Fig. 5.3 we report the time evolution of the validation accuracy in 3 different systems. A wireless systems with slow UL $\rho = 4$ and unreliable nodes $T_{min} = T_{dl} = \frac{1}{\mu}$, a wireless system with fast uplink $\rho = 2$ and reliable nodes $T_{min} = T_{dl}$, $\frac{1}{\mu} = 0$ and a wired system $\rho = 1$ (symmetric UL and DL) with reliable nodes $T_{min} = T_{dl}$, $\frac{1}{\mu} = 0$. The increased DL cost is negligible for wireless systems with strongly asymmetric UL/DL rates and in these cases, the proposed approach largely outperforms the baselines. In the case of more balanced UL and DL transmission times $\rho = [1, 2]$ and reliable nodes, it becomes instead necessary to properly choose the number of personalized streams in order to render the solution practical. Nonetheless, the proposed approach remains the best even in this case for $k = 4$. Note that FedFOMO incurs a large communication cost as personalized aggregation is performed at the client-side.

5.5 Conclusion

In this chapter, we presented a novel federated learning algorithm that exploits multiple user-centric aggregation rules to produce personalized models. The aggregation rules are based on user-specific mixture coefficients that can be computed during one communication round prior to federated training. Additionally, in order to limit the communication burden of personalization, we propose a simple strategy to effectively limit the number of personalized streams. We experimentally study the performance of the proposed solution across different tasks. Overall, our solution yields personalized models with higher testing accuracy while at the same time being more communication-efficient compared to the competing baselines.

Part III

Robust Bayesian Learning

Chapter 6

Robust Bayesian Learning

Bayesian learning provides a principled framework to account for uncertainty quantification, an essential enabler for reliable AI. However, standard Bayesian learning is known to have suboptimal generalization capabilities under model misspecification and in the presence of outliers. PAC-Bayes theory demonstrates that the free energy criterion minimized by Bayesian learning is a bound on the generalization error for Gibbs predictors (i.e., for single models drawn at random from the posterior) under the assumption of sampling distributions uncontaminated by outliers. This viewpoint provides a justification for the limitations of Bayesian learning when the model is misspecified, requiring ensembling, and when data is affected by outliers. In recent work, PAC-Bayes bounds – referred to as PAC^m – were derived to introduce free energy metrics that account for the performance of ensemble predictors, obtaining enhanced performance under misspecification. This chapter introduces a novel robust free energy criterion that combines the generalized logarithm score function with PAC^m ensemble bounds. The proposed free energy training criterion produces predictive distributions that are able to concurrently counteract the detrimental effects of model misspecification and outliers.

6.1 Introduction

Key assumptions underlying Bayesian inference and learning are that the adopted probabilistic model is well specified and that the training data set does not include outliers, so that training and testing distributions are matched [53]. Under these favorable conditions, the Bayesian posterior distribution provides an optimal solution to the inference and learning problems. In contrast, optimality does not extend to scenarios characterized by misspecification [136, 137] *or* outliers [51]. The framework developed in this chapter aims at addressing *both* problems by integrating the use of ensemble predictors [54] with generalized logarithm score functions [138] in Bayesian learning.

The proposed learning framework – termed (m, t)-robust Bayesian learning – is underpinned by a novel *free energy* learning criterion parameterized by integer $m \geq 1$ and scalar $t \in [0, 1]$. The parameter m controls robustness to misspecification by determining the size of the ensemble used for prediction. In contrast, parameter t controls robustness to outliers by dictating the degree to which the loss function penalizes low predictive

probabilities. The proposed learning criterion generalizes the standard free energy criterion underlying generalized Bayesian learning, which is obtained for $m = 1$ and $t = 1$ [139, 140]; as well as the m-free energy criterion, obtained for $t = 1$, which was recently introduced in [3].

6.1.1 Related Work

Recent work has addressed the problem of model misspecification for Bayesian learning. In particular, references [3, 52] have argued that the minimization of the standard free energy criterion – which defines generalized Bayesian learning [139,140] – yields predictive distributions that do not take advantage of ensembling, and thus have poor generalization capabilities for misspecified models.

To mitigate this problem, references [3, 52] introduced alternative free energy criteria that account for misspecification. The author of [52] leveraged a second-order Jensen's inequality to obtain a tighter bound on the cross entropy loss; while the work [3] proposed an m-free energy criterion that accounts for the performance of an ensemble predictor with m constituent models. Both optimization criteria were shown to be effective in overcoming the shortcomings of Bayesian learning under misspecification, by yielding posteriors that make better use of ensembling.

The free energy metrics introduced in [3,52] are defined by using the standard log-loss, which is known to be sensitive to outliers. This is because the log-loss grows unboundedly on data points that are unlikely under the model [141]. Free energy criteria metrics based on the log-loss amount to Kullback–Leibler (KL) divergence measures between data and model distributions. A number of papers have proposed to mitigate the effect of outliers by replacing the classical criteria based on the KL divergence in favor of more robust divergences, such as the β-divergences [142, 143] and the γ-divergence [144, 145]. These criteria can be interpreted as substituting the log-loss with generalized logarithmic scoring rules. To optimize such criteria, variational methods have been proposed that were shown to be robust to outliers, while not addressing model misspecification [146].

This chapter extends (generalized) Bayesian learning by tackling both model misspecification and the presence of outliers. Specifically, we propose the (m, t)-robust Bayesian learning framework, which is underpinned by a novel free energy criterion based on generalized logarithmic scoring rules and multi-sample objectives. The predictive distribution resulting from the minimization of the proposed objective takes full advantage of ensembling, while at the same time reducing the effect of outliers. The proposed robust m-free energy criterion is justified by following a PAC-Bayes approach, and its enhanced robustness is also proved through the lens of its influence function [147]. The theoretical findings are corroborated by experiments that highlight the enhanced generalization capabilities and calibration performance of the proposed learning criterion under model misspecification and with data sets corrupted by outliers.

6.1.2 Chapter Organization

The rest of chapter the is organized as follows. In Section 6.2, we review the generalized logarithm function, the associated entropy and divergence measures. In Section 6.3, we

define the learning setup and we provide a tutorial-style comparison between frequentist and Bayesian learning frameworks. In Section 6.4.1, we introduce the concept of model misspecification and we review the m-free energy criterion [3] as a tool to mitigate the effect of misspecified model classes. In Section 6.4.2, we define outliers and illustrate the role of robust losses to reduce the influence of outlying data samples. In Section 6.5, we introduce the (m, t)-robust Bayesian learning framework that tackles both model misspecification and the presence of outliers, and that overcomes the limitations of the standard Bayesian learning rule. We theoretically analyze the proposed learning criterion, providing PAC-Bayesian guarantees for the ensemble model with respect to the contaminated and the in-distribution measures. Finally, in Section 6.6, we provide regression and classification experiments to quantitatively and qualitatively measure the performance of the proposed learning criterion.

6.2 Preliminaries

6.2.1 Generalized Logarithms

The t-logarithm function, also referred to as generalized or tempered logarithm is defined as

$$\log_t(x) := \frac{1}{1-t}\left(x^{1-t} - 1\right) \quad \text{for } x > 0, \tag{6.1}$$

for $t \in [0, 1) \cup (1, \infty)$, and

$$\log_1(x) := \log(x) \quad \text{for } x > 0 \tag{6.2}$$

where the standard logarithm (6.2) is recovered from (6.1) in the limit $\lim_{t\to 1} \log_t(x) = \log(x)$. As shown in Figure 6.1, for $t \in [0, 1)$, the t-logarithms is a concave function, and for $t < 1$ is lower bounded as $\log_t(x) \geq -(1-t)^{-1}$.

Largely employed in classical and quantum physics, the t-logarithm has also been applied to machine learning problems. Specifically, t-logarithms have been used to define alternatives to the log-loss as a score function for probabilistic predictors with the aim of enhancing robustness to outliers [5, 138, 148]. Accordingly, the loss associated to a probabilistic model $q(x)$ is measured as $-\log_t q(x)$ instead of the standard log-loss $-\log q(x)$. Note that we have the upper bound $-\log_t q(x) \leq (1-t)^{-1}$ for $t < 1$.

In information theory, the t-logarithm was used by [149] to define the t-Tsallis entropy

$$H_t(p(x)) := -\int p(x)^t \log_t p(x) dx, \tag{6.3}$$

and the t-Tsallis divergence

$$D_t(p(x)||q(x)) := -\int p(x)^t [\log_t p(x) - \log_t q(x)] dx. \tag{6.4}$$

For $t = 1$, the t-Tsallis entropy and the t-Tsallis divergence coincide respectively with the Shannon (differential) entropy and with the Kullback–Leibler (KL) divergence.

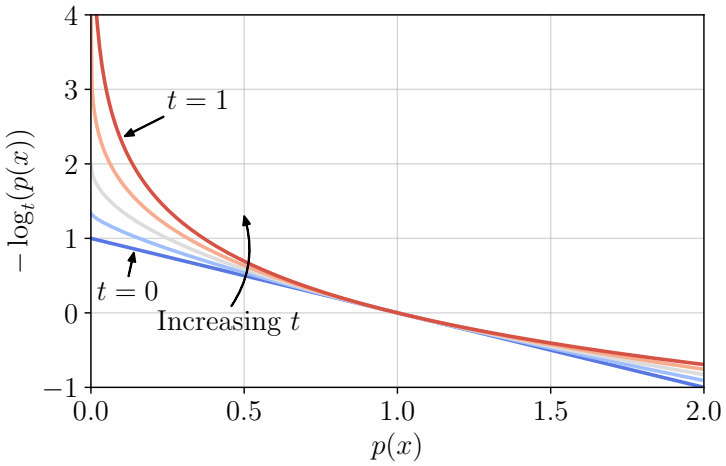


Figure 6.1: t-logarithm loss, or $\log_t$-loss, of a predictive distribution $p(x)$ for different values of t. For $t = 1$, the samples x corresponding to low predictive probability $p(x) \to 0$ have a potentially unbounded loss value. On the contrary, for $t < 1$, the t-logarithm loss is bounded by $(1-t)^{-1}$ and it limits their influence.

When using (6.4) as an optimization criterion in machine learning, the concept of escort distribution is often useful [150]. Given a probability density $p(x)$, the associated t-escort distribution is defined as

$$\mathcal{E}_t(p(x)) = \frac{p(x)^t}{\int p(x)^t dx}. \tag{6.5}$$

We finally note that t-logarithm does not satisfy the distributive property of the logarithm, i.e., $\log(xy) = \log(x) + \log(y)$. Instead, we have the equalities [151]

$$\log_t(xy) = \log_t x + \log_t y + (1-t)\log_t x \log_t y \tag{6.6}$$

and

$$\log_t\left(\frac{x}{y}\right) = y^{t-1}\left(\log_t x - \log_t y\right). \tag{6.7}$$

6.3 Frequentist vs. Bayesian Learning

Throughout this chapter we consider a standard learning set-up in which the learner has access to a data set $\mathcal{D}$ of n data points $\{z_i\}_{i=1}^n$ sampled in an independent and identically distributed (IID) fashion from a *sampling distribution* $\nu_s(z)$. As we will see, owing to the presence of outliers, the sampling distribution may differ from the *target distribution* $\nu(z)$. The general goal of learning is that of optimizing models that perform well on

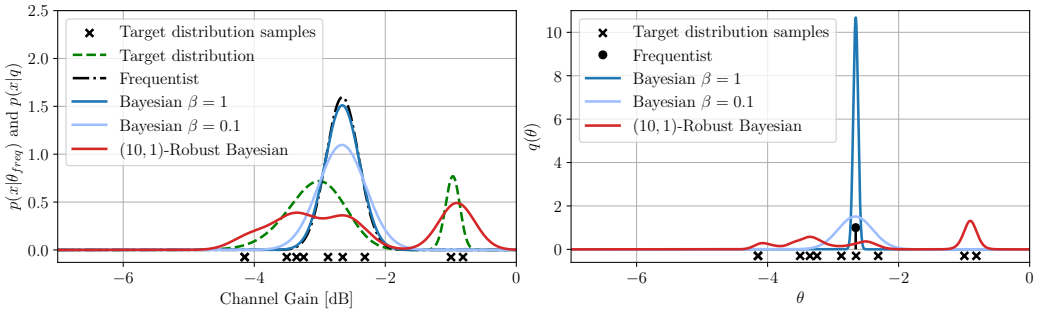


Figure 6.2: Estimated distribution over a scalar channel gain (left panel) and corresponding posterior distribution $q(\theta)$ over the model parameter θ (right panel) for frequentist learning, Bayesian learning with $\beta \in \{1, 0.1\}$ and $(m, 1)$-robust Bayesian learning with $m = 10$. The training data set, represented as crosses, is sampled from the target distribution $\nu(x)$.

average with respect to the target distribution $\nu(z)$. In this section, we assume that the sampling distribution $\nu_s(z)$ equals the target distribution $\nu(z)$, and we will address the problem of outliers – which arises when $\nu_s(z) \neq \nu(z)$ – in the next section.

We will consider both *supervised learning* problems and the *unsupervised learning* problem of density estimation, which, as we will see in Chapter 7, have many applications to wireless communications. In supervised learning, a data sample $z \in \mathcal{Z}$ corresponds to a pair $z = (x, y)$ that comprises a feature vector $x \in \mathcal{X}$ and a label $y \in \mathcal{Y}$. In contrast, for density estimation, each data point $z \in \mathcal{Z}$ corresponds to a feature vector $z = x \in \mathcal{X}$.

Supervised learning is formulated as an optimization over a family of discriminative models defined by a parameterized conditional distribution $p(y|x, \theta)$ of target y given input x. The conditional distribution, or model, $p(y|x, \theta)$ is parameterized by vector $\theta \in \Theta$ in some domain Θ. In contrast, density estimation amounts to an optimization over a model defined by parameterized densities $p(x|\theta)$. In both cases, optimization targets a real-valued *loss function*, which is used to score the model θ when tested on a data point z.

6.3.1 Frequentist Learning

The goal of frequentist learning consists in finding the model parameter vector θ that minimizes the *training loss* evaluated on the data set $\mathcal{D}$, i.e.,

$$\hat{\mathcal{L}}(\theta, \mathcal{D}) = \sum_{z \in \mathcal{D}} \ell(\theta, z). \tag{6.8}$$

This optimization follows the *empirical risk minimization* (ERM) principle. Accordingly, the frequentist solution is a *single* model parameter $\theta^{\text{freq}} \in \Theta$ that minimizes the training loss, i.e.,

$$\theta^{\text{freq}} = \arg\min_{\theta \in \Theta} \hat{\mathcal{L}}(\theta, \mathcal{D}). \tag{6.9}$$

To simplify the discussion, we assume that the solution to the ERM problem is unique, although this does not affect the generality of the presentation.

ERM is motivated by the fact that the training loss (6.8) is a finite-sample approximation of the true, unknown, *population loss*

$$\mathcal{L}(\theta) = \mathbb{E}_{\nu(z)}[\ell(\theta, z)], \tag{6.10}$$

which averages the loss over the target, and here also sampling, distribution $\nu(z)$. The discrepancy between the population loss and its approximation given by the training loss introduces uncertainty regarding the optimal model parameter

$$\theta^* = \arg\min_{\theta \in \Theta} \mathcal{L}(\theta), \tag{6.11}$$

which is also assumed to be unique to simplify the discussion. The error between the optimal solution θ^* and the frequentist solution θ^{freq} is a form of *epistemic uncertainty*, which can be reduced by increasing the size of the data set $\mathcal{D}$.

In practice, the short stationarity intervals of the data-generating distributions associated with wireless communications often limit the size of training data sets. In this scarce data regime, epistemic uncertainty may be significant. By selecting a single model, frequentist learning neglects epistemic uncertainty as it discards information about other plausible models that fit training data almost as well as the ERM solution (6.9). As a result, frequentist learning is known to lead to poorly calibrated decision [1,152], resulting in over- or under-confident outputs that may cause important reliability issues.

6.3.2 Bayesian Learning

Bayesian learning adopts a probabilistic reasoning framework by scoring all members in the model class by means of a distribution $q(\theta)$ over the model parameter space Θ. Through this distribution, Bayesian learning summarizes information obtained from data $\mathcal{D}$, as well as prior knowledge about the problem, e.g., about the scale of the optimal model parameter vector θ^* or about sparsity patterns in θ^*.

Mathematically, given a prior distribution $p(\theta)$ on the model parameter space, Bayesian learning can be formulated as the minimization of the *free energy criterion*

$$\hat{\mathcal{J}}(q) = \mathbb{E}_{q(\theta)}[\hat{\mathcal{L}}(\theta, \mathcal{D})] + \frac{1}{\beta}\text{KL}(q(\theta)||p(\theta)), \tag{6.12}$$

where $\text{KL}(q(\theta)||p(\theta))$ denotes the Kullback–Leibler (KL) divergence between the posterior distribution $q(\theta)$ and a *prior* distribution $p(\theta)$, i.e.

$$\text{KL}(q(\theta)||p(\theta)) = \mathbb{E}_{q(\theta)}\left[\log\left(\frac{q(\theta)}{p(\theta)}\right)\right], \tag{6.13}$$

while $\beta > 0$ is a constant, also known as inverse temperature. Accordingly, through problem

$$\underset{q}{\text{minimize}}\, \hat{\mathcal{J}}(q), \tag{6.14}$$

Bayesian learning minimizes a weighted sum of the average training loss and of the discrepancy with respect to the prior distribution $p(\theta)$.

The KL term in the free energy (6.12) plays an essential role in differentiating between Bayesian learning and frequentist learning for small data set sizes. In fact, the KL divergence term acts as a regularizer, whose influence on the solution of problem (6.14) is inversely proportional to the data set size n. When the regularizer is removed, i.e., when we set $\beta \to \infty$, the solution of the problem (6.14) reduces to the frequentist solution (6.9). More precisely, the distribution $q(\theta)$ that solves problem (6.14) reduces to a point distribution concentrated at θ^{freq}.

The optimization (6.14) of the free energy criterion (6.12) can be theoretically justified through the *PAC Bayes* generalization framework. In it, the KL term is proved to quantify an upper bound on the discrepancy between training loss and population loss on average with respect to the random draws of the model parameter vector $\theta \sim q(\theta)$. Mathematically, the free energy provides an upper bound on the average population loss (when neglecting constants that are inessential for optimization), i.e.,

$$\mathbb{E}_{q(\theta)}\left[\mathcal{L}(\theta)\right] \leq \hat{\mathcal{J}}(q) + \text{const.} \tag{6.15}$$

As we have discussed in the previous subsection, epistemic uncertainty is caused by the difference between training and population losses, and hence between the corresponding minimizers (6.11) and (6.9). By incorporating a bound on this error, the free energy criterion (6.12) unlike the frequentist training loss (6.8), provides a way to account for epistemic uncertainty.

Specializing the problem (6.14) to the *log-loss*

$$\ell(x, y, \theta) = -\log p(y|x, \theta) \tag{6.16}$$

for supervised learning, and

$$\ell(x, \theta) = -\log p(x|\theta) \tag{6.17}$$

for density estimation, the minimization of the free energy in (6.14) leads to the *β-tempered posterior distribution*

$$q^{Bayes}(\theta|\mathcal{D}) \propto \prod_{(x,y)\in\mathcal{D}} p(\theta)p(y|x,\theta)^{\beta} \tag{6.18}$$

for supervised learning, and a similar expression applies to unsupervised learning for density estimation. The distribution (6.18) reduces to the standard posterior distribution when $\beta = 1$. In practice, computing the posterior distribution, or more generally solving problem (6.14), are computationally prohibitive tasks. A common approach to address this issue is through *variational inference (VI)* [153]. VI limits the scope of the optimization over a tractable set of distributions $q(\theta)$, such as jointly Gaussian variables with free mean and covariance parameters.

Let us now assume that we have obtained a distribution $q(\theta)$ as a, generally approximate, solution of problem (6.14). We focus first on supervised learning. Given a test input x, the *ensemble predictor* obtained from distribution $q(\theta)$ is given by

$$p(y|x, q) = \mathbb{E}_{q(\theta)}[p(y|x, \theta)]. \tag{6.19}$$

The average in (6.19) is in practice approximated by drawing multiple, say m, samples $\theta \sim q(\theta)$ from distribution $q(\theta)$, obtaining the *m-sample predictor*

$$p(y|x, \theta_1, ..., \theta_m) = \frac{1}{m} \sum_{i=1}^{m} p(y|x, \theta_i), \tag{6.20}$$

where samples θ_i are generated from distribution $q(\theta)$ for $i = 1, ..., m$, which we write as $\theta_1, ..., \theta_m \sim q(\theta)^{\otimes m}$.

In the case of density estimation, the ensemble density $p(x|q)$ is similarly defined as

$$p(x|q) = \mathbb{E}_{q(\theta)}[p(x|\theta)], \tag{6.21}$$

which can be approximated as

$$p(x|\theta_1, \ldots, \theta_m) = \frac{1}{m} \sum_{i=1}^{m} p(x|\theta_i), \tag{6.22}$$

with $\theta_1, \ldots, \theta_m \sim q(\theta)^{\otimes m}$. Henceforth, when detailing expressions for supervised learning, it will be implied that the corresponding formulas for density estimation apply by replacing $p(y|x, \theta)$ with $p(x|\theta)$ as done above to define ensemble predictors.

Given a distribution $q(\theta)$, we define the log-loss of the ensemble model (6.21) as

$$\mathcal{R}(q, x, y) := -\log p(y|x, q) = -\log \mathbb{E}_{q(\theta)}[p(y|x, \theta)], \tag{6.23}$$

and the *m-sample* log-*loss* as

$$\hat{\mathcal{R}}^m(q, x, y) := \mathbb{E}_{q(\theta)^{\otimes m}}\left[-\log(p(y|x, \theta_1, ..., \theta_m))\right] = \mathbb{E}_{q(\theta)^{\otimes m}}\left[-\log\left(\frac{1}{m}\sum_{i=1}^{m} p(x|\theta_i)\right)\right], \tag{6.24}$$

which measures the log-loss of the m-sample predictor (6.20). Note that for $m = 1$ in (6.24), we obtain the log loss of the Gibbs predictor

$$\hat{\mathcal{R}}^1(q, x, y) = \hat{\mathcal{R}}(q, x, y) := \mathbb{E}_{q(\theta)}[-\log p(y|x, \theta)]. \tag{6.25}$$

Example 1: To illustrate the difference between the frequentist and Bayesian learning paradigms, let us consider the problem of estimating the probability distribution of the channel gain of a scalar wireless channel. This is an example of unsupervised learning for density estimation. Let us assume that the channel gain density follows a true, unknown, target distribution given by the mixture of two Gaussians $\nu(x) = 0.7\mathcal{N}(x|0.5, 0.05) + 0.3\mathcal{N}(x|0.8, 0.02)$. This is shown in the left part of Fig. 6.2 as a dashed green line. The two components may correspond to line-of-sight (LOS) and non-line-of-sight (NLOS) propagation conditions [154]. We fix a Gaussian model class $p(x|\theta) = \mathcal{N}(x|\theta, 0.25)$ and a prior distribution $p(\theta) = \mathcal{N}(\theta| - 5, 5)$. Given the data points represented as crosses in the left part of Figure 6.2, the estimated distribution obtained by frequentist learning is reported as a dash-dotted black curve in the left panel. In

contrast, Bayesian learning returns the posterior distribution (6.18), which in turn yields the ensemble density (6.19). The distributions are shown in the left and right parts of the Figure 6.2, respectively for inverse temperature parameters $\beta = \{1, 0.1\}$. The Bayesian predictive distribution is still unimodal but it has a larger variance, which results from the combination of multiple Gaussian models according to the Bayesian posterior that does not collapse to a point distribution in virtue of the KL regularization term whose influence is controlled by β. ■

6.4 Robust Bayesian Learning

As we have seen in the previous section, Bayesian learning optimizes the free energy by tackling problem (6.14). By (6.15), the free energy provides a bound on the population loss as a function of the training loss when averaging over the distribution $q(\theta)$ in the model parameter space [155]. This approach has two important limitations:

- *Model misspecification*: The bound (6.15) provided by the free energy is known to be loose in the presence of model misspecification. Model misspecification occurs when the assumed probabilistic model $p(y|x, \theta)$ cannot express the conditional target distribution $\nu(y|x) = \nu(x, y)/\nu(x)$, where $\nu(x) = \int \nu(x, y)dy$ [3, 156]. This causes the β-tempered posterior distribution to be generally suboptimal when the model is misspecified [139].

- *Discrepancy between sampling and target distributions*: The sampling distribution $\nu_s(z)$ that underlies the generation of the training data set $\mathcal{D}$ may not match the target distribution $\nu(x)$ used to test the trained model due to the presence of outliers in the training data. This discrepancy is not accounted for in the derivation of the free energy criterion, causing Bayesian learning to be suboptimal in the presence of outliers [139].

We observe that the two causes of suboptimality outlined in the previous paragraph are distinct. In fact, model misspecification may reflect the ignorance of the learner concerning the data generation process, or it may be caused by constraint on the computational resources of the device implementing the model. In contrast, the presence of outliers amounts to an inherent source of distortion in the data, which cannot be removed even if the learner acquired more information about the data generation process or more computing power. In this section, we review robust Bayesian learning solutions that address these two issues.

6.4.1 $(m, 1)$-Robust Bayesian Learning Against Model Misspecification

In this subsection, we describe a recently proposed method that robustifies Bayesian learning against model misspecification. We start by providing a formal definition of misspecification. Recall that we are focusing on supervised learning, but the presentation also applies to density estimation by replacing the discriminative model $p(x|y, \theta)$ with the density model $p(x|\theta)$.

Definition 2 (Misspecification)**.** *A model class $\mathcal{F} = \{p(y|x,\theta) : \theta \in \Theta\}$ is said to be misspecified with respect to the target distribution $\nu(x,y)$ whenever there is no model parameter vector $\theta \in \Theta$ such that $\nu(y|x) = p(y|x,\theta)$, where $\nu(y|x)$ is the conditional target distribution obtained from the joint target distribution $\nu(x,y)$.*

Under model misspecification, the free energy criterion has been shown to yield a loose bound (6.15) on the population loss obtained by the ensemble predictor (6.19) [3].

To address this problem, the m-sample free energy criterion was introduced in [3], whose minimization yields $(m,1)$-robust learning. The reason for the notation "$(m,1)$" will be made clear in the next two subsections. The key observation underlying this approach is that the training loss $\hat{\mathcal{L}}(\theta,\mathcal{D})$ in the standard free energy (6.12) does not properly account for the performance of ensemble predictors. In fact, the log-loss of an m-sample ensemble predictor is given by $\hat{\mathcal{R}}^m(q,x,y)$ in (6.24), and not by the Gibbs log-loss $\hat{\mathcal{R}}(q,x,y)$ in (6.25). By leveraging the results of [157] and [158], the multi-sample criterion $\hat{\mathcal{R}}^m(q,x,y)$ can be shown to provide a sharper bound to the ensemble risk $\mathcal{R}(q,x,y)$ in (6.23) as compared to the Gibbs risk $\hat{\mathcal{R}}(q,x,y)$ in (6.25), i.e.,

$$\mathcal{R}(q,x,y) \leq \hat{\mathcal{R}}^m(q,x,y) \leq \hat{\mathcal{R}}(q,x,y) \tag{6.26}$$

Furthermore, the first inequality in (6.26) becomes asymptotically tight as $m \to \infty$, i.e.,

$$\lim_{m\to\infty} \hat{\mathcal{R}}^m(q,x,y) = \mathcal{R}(q,x,y). \tag{6.27}$$

Using PAC-Bayes arguments, the m-sample free energy is obtained by replacing the training loss $\mathbb{E}_{q(\theta)}[\hat{\mathcal{L}}(\theta,\mathcal{D})]$ in the free energy (6.12) with the *m-sample training loss*

$$\begin{aligned}\hat{\mathcal{L}}(\theta_1,\ldots,\theta_m,\mathcal{D}) &= \sum_{(x,y)\in\mathcal{D}} \hat{\mathcal{R}}^m(q,x,y) \\ &= \sum_{(x,y)\in\mathcal{D}} \mathbb{E}_{q(\theta)^{\otimes m}}\left[-\log(p(y|x,\theta_1,...,\theta_m))\right].\end{aligned} \tag{6.28}$$

Furthermore, the *m-sample free energy* is defined as

$$\hat{\mathcal{J}}^m(q) = \hat{\mathcal{L}}(\theta_1,\ldots,\theta_m,\mathcal{D}) + \frac{m}{\beta}\mathrm{KL}(q(\theta)||p(\theta)), \tag{6.29}$$

in which the m-sample training loss is averaged over the distribution of the m samples $\theta_1,...,\theta_m \sim q(\theta)^{\otimes m}$ used in the ensemble predictor (6.20). We note that the m-sample free energy coincides with the standard free energy (6.12) for $m = 1$.

Finally, the *$(m,1)$-robust Bayesian learning* problem is defined by the optimization

$$\underset{q}{\text{minimize}}\, \hat{\mathcal{J}}^m(q). \tag{6.30}$$

Example 1 (continued): Let us return to Example 1. The problem is characterized by model misspecification since the target distribution $\nu(x)$ is a mixture of two Gaussian components, while the model class comprises only unimodal Gaussian models $p(x|\theta)$. In contrast to standard Bayesian learning, the ensemble density (6.20) obtained with the distribution $q(\theta)$ returned by $(m,1)$-robust Bayesian learning for $m = 10$ (red curve in the right panel) is able to take advantage of ensembling to approximate both the NLoS and LoS components of the target distribution. ■

6.4.2 $(1,t)$-Robust Bayesian Learning Against Outliers

We now turn to methods that robustify Bayesian learning against the presence of outliers in the training set. As in [159], we model the presence of outliers by assuming that the training data is generated from a sampling distribution $\nu_s(x,y)$ that is given by the contamination of the *in-distribution(ID) distribution* $\nu(x,y)$ by an *out-of-distribution (OOD) distribution* $\xi(x,y)$. A formal definition follows.

Assumption 7 (Outliers)**.** *The sampling distribution is given by*

$$\nu_s(x,y) = (1-\epsilon)\nu(x,y) + \epsilon\xi(x,y) \tag{6.31}$$

where $\nu(x,y)$ is the target distribution; $\xi(x,y)$ is the OOD distribution accounting for the presence of outliers; and $\epsilon \in [0,1]$ denotes the contamination ratio.

In order for model (6.31) to be meaningful, one typically assumes that the OOD measure $\xi(x,y)$ is large for pairs of (x,y) at which the target measure $\nu(x,y)$ is small. This ensures that outlying data points $(x,y) \sim \xi(x,y)$ tend to be in part of the domain that is not covered by the target distribution.

The performance of both frequentist and Bayesian learning is known to be sensitive to outliers when the log-loss is adopted to evaluate the training loss. This sensitivity is caused by the unbounded value of the log-loss (6.16) when evaluated on anomalous data points to which the model assigns low probabilities $p(y|x,\theta)$. This is illustrated in Figure 6.1 for a general conditional distribution $p(y|x)$. A number of papers have proposed to mitigate the effect of outliers by replacing the log-loss in favor of more robust losses [141–145].

A well-explored solution is to adopt the t-log-loss introduced in Section 6.2. Using the t-log-loss in lieu of the standard log-loss in the loss definitions (6.23) and (6.25), we obtain the $\log_t$-loss of the ensemble model (6.21) as

$$\mathcal{R}_t(q,x,y) := -\log_t p(y|x,q) = -\log_t \mathbb{E}_{q(\theta)}[p(y|x,\theta)], \tag{6.32}$$

and the $\log_t$ loss of the Gibbs predictor

$$\hat{\mathcal{R}}_t(q,x,y) := \mathbb{E}_{q(\theta)}[-\log_t p(y|x,\theta)]. \tag{6.33}$$

By (6.2), the above definitions generalize the ones based on the standard log-loss as these are obtained with $t=1$. On the other hand, for $t<1$ the associated loss function is bounded by $(1-t)^{-1}$, as shown in Figure 6.1.

Based on (6.33), we obtain the t-training loss as

$$\hat{\mathcal{L}}_t(\theta,\mathcal{D}) = -\sum_{(x,y)\in\mathcal{D}} \hat{\mathcal{R}}_t(q,x,y), \tag{6.34}$$

which leads to the corresponding *t-free energy*

$$\hat{\mathcal{J}}_t(q) = \hat{\mathcal{L}}_t(\theta,\mathcal{D}) + \frac{1}{\beta}\mathrm{KL}(q(\theta)||p(\theta)). \tag{6.35}$$

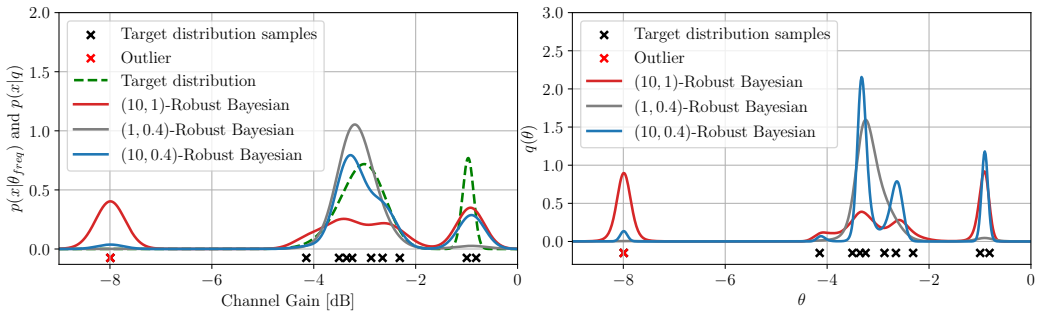


Figure 6.3: Estimated distribution over channel gains (left panel) and posterior distribution over the model parameter θ (right panel) of a density model trained following $(m, 1)$-robust Bayesian learning, the $(1, t)$-robust Bayesian learning and the (m, t)-robust Bayesian learning. The training data set, represented as crosses, comprises samples from the sampling distribution $\nu(x)$ (black) and an outlier (red).

Accordingly, $(1, t)$-*robust Bayesian learning* is defined by the minimization [3]

$$\underset{q}{\text{minimize}}\, \hat{\mathcal{J}}_t(q). \tag{6.36}$$

Example 2: To highlight the effect of outliers, we consider the same channel gain estimation problem described in Example 1, but we now assume that the original training data set (black crosses) is contaminated by an outlying data point (red cross). The $(m, 1)$-robust Bayesian learning solution (red curve with $m = 10$) is based on the standard log-loss and is observed to be significantly affected by the presence of the outliers. As a result, the estimated distribution for the $(m, 1)$-robust Bayesian learning concentrates a relevant fraction of its mass around the outlier. In contrast, the $(1, t)$-robust Bayesian solution (gray curve) with $t = 0.4$ is less influenced by the outlying data point. However, like Bayesian learning, it is not able to take advantage of ensembling and to approximate both LoS and NLoS components. This observation justifies the (m, t)-robust Bayesian learning approach described next. ■

6.5 (m, t)-Robust Bayesian Learning

In the previous section, we reviewed the m-free energy criterion introduced by [3], which was argued to produce predictive distributions that are more expressive, providing a closer match to the underlying sampling distribution $\nu(x)$. However, the approach is not robust to the presence of outliers. In this section, we introduce (m, t)-robust Bayesian learning and the associated novel free energy criterion that addresses both expressivity in the presence of misspecification and robustness in setting with outliers. To this end, we study the general setting described in Section 6.4 in which the sampling distribution $\tilde{\nu}(x)$ satisfies both Assumption 2 and Assumption 7, and we investigate the use of the $\log_t$-loss with $t \in [0, 1)$ as opposed to the standard log-loss as assumed in [3].

6.5.1 Robust m-free Energy

For a proposal posterior $q(\theta)$, generalizing (6.24), we define the multi-sample empirical $\log_t$-loss evaluated at a data point x as

$$\hat{\mathcal{R}}_t^m(q,x,y) := \mathbb{E}_{q(\theta)^{\otimes m}}\left[-\log_t(p(y|x,\theta_1,...,\theta_m))\right] = \mathbb{E}_{q(\theta)^{\otimes m}}\left[-\log_t(\frac{1}{m}\sum_{i=1}^{m} p(x|\theta_i))\right], \tag{6.37}$$

From the concavity of the t-logarithm with $t \in [0,1)$, in a manner similar to (6.26), the loss (6.37) provides an upper bound on the original $\log_t$-loss $\mathcal{R}_t(q,x,y)$ in (6.23)

$$\mathcal{R}_t(q,x,y) \leq \hat{\mathcal{R}}_t^m(q,x,y). \tag{6.38}$$

Furthermore, the bound becomes increasingly tighter as m increases, and we have the limit

$$\lim_{m\to\infty} \hat{\mathcal{R}}_t^m(q,x,y) = \mathcal{R}_t(q,x,y) \tag{6.39}$$

for $t \in [0,1)$.

The m-sample $\log_t$-loss (6.37) is used to define the (m,t) training loss

$$\hat{\mathcal{L}}_t(\theta_1,\ldots,\theta_m,\mathcal{D}) = \sum_{(x,y)\in\mathcal{D}} \hat{\mathcal{R}}_t^m(q,x,y), \tag{6.40}$$

based on which, for $\beta > 0$, is possible to derive the robust m-free energy as

$$\mathcal{J}_t^m(q) := \hat{\mathcal{L}}_t(\theta_1,\ldots,\theta_m,\mathcal{D}) + \frac{m}{\beta} KL(q(\theta)||p(\theta)). \tag{6.41}$$

The proposed free energy generalizes the standard free-energy criterion (6.12), which corresponds to the training criterion of (m,t)-robust Bayesian learning for $m=1$ and $t=1$, and the m-free energy criterion (6.29), which corresponds to the training criterion of (m,t)-robust Bayesian learning for $t=1$.

In the following we provide theoretical results for the proposed learning framework. The analysis is specialized for the unsupervised setting in order to simplify the notation. Nonetheless, the results can be readily extended to the supervised setting by replacing $p(x|\theta)$ with $p(y|x,\theta)$.

Following similar steps as in [3], the robust m-free energy can be proved to provide an upper bound on the population $\log_t$-risk as detailed in the following lemma.

Lemma 3. *With probability $1-\sigma$, with $\sigma \in (0,1)$, with respect to the random sampling of the data set $\mathcal{D}$, for all distributions $q(\theta)$ that are absolutely continuous with respect the prior $p(\theta)$, the following bound on the population risk of the ensemble model holds*

$$\mathbb{E}_{\tilde{\nu}(x)}[\mathcal{R}_t(q,x)] \leq \mathcal{J}_t^m(q) + \psi(\tilde{\nu},n,m,\beta,p,\sigma) \tag{6.42}$$

where

$$\psi(\tilde{\nu}, n, m, \beta, p, \sigma) := \frac{1}{\beta}\left(\log \mathbb{E}_{\mathcal{D},p(\theta)}\left[e^{\beta\Delta_{m,n}}\right] - \log\sigma\right) \tag{6.43}$$

and

$$\begin{aligned}\Delta_{m,n} :=& \frac{1}{n}\sum_{x\in\mathcal{D}} \log_t \mathbb{E}_{j\sim U[1:m]} p(x|\theta_j) \\ &- \mathbb{E}_{\tilde{\nu}(x)}\left[\log_t \mathbb{E}_{j\sim U[1:m]} p(x|\theta_j)\right].\end{aligned} \tag{6.44}$$

Furthermore, the risk with respect to the ID measure $\nu(x)$ can be bounded as

$$\begin{aligned}\mathbb{E}_{\nu(x)}[\mathcal{R}_t(q,x)] \leq& \frac{1}{1-\epsilon}\left(\mathcal{J}_t^m(q) + \psi(\tilde{\nu}, n, m, \beta, p, \sigma)\right) \\ &+ \frac{\epsilon(C^{1-t}-1)}{(1-\epsilon)(1-t)},\end{aligned} \tag{6.45}$$

if the contamination ratio satisfies the inequality $\epsilon < 1$.

Lemma 3 provides an upper bound on the $\log_t$-risk (6.32), which is defined with respect to the sampling distribution $\tilde{\nu}(x)$ corrupted by outliers, as well as on the ensemble $\log_t$-risk (6.23) evaluated with respect to the ID measure $\nu(x)$. Reflecting that the data set $\mathcal{D}$ contains samples from the corrupted measure $\tilde{\nu}(x)$, while the bound (6.42) vanishes as $n \to \infty$, a non-vanishing term appears in the bound (6.45).

6.5.2 Minimizing the Robust m-free Energy

Using standard tools from calculus of variations, it is possible to express the minimizer of the robust m-free energy

$$q_t^m(\theta) := \arg\min_q \mathcal{J}_t^m(q) \tag{6.46}$$

as fixed-point solution of an operator acting on the ensembling distribution $q(\theta)$.

Theorem 6. *The minimizer (6.46) of the robust m-free energy objective (6.41) is the fixed point of the operator*

$$T(q) := p(\theta_j)\exp\left(\beta\sum_{x\in\mathcal{D}} \mathbb{E}_{\{\theta_i\}_{i\neq j}}\left[\log_t\left(\frac{\sum_{i=1}^m p(x|\theta_i)}{m}\right)\right]\right) \tag{6.47}$$

where the average in (6.47) is taken with respect to the IID random vectors $\{\theta_i\}_{i\neq j} \sim q(\theta)^{\otimes m-1}$.

Theorem 6 is useful to develop numerical solutions to problem (6.46) for non-parametric posteriors, and it resembles standard mean-field variational inference iterations [160].

Alternatively, we can tackle the problem (6.46) over a parametric family of distribution using standard tools from variational inference [153].

To further characterize the posterior minimizing the robust m-free energy criterion, and to showcase the beneficial effect of the generalized logarithm, we now consider the asymptotic regime in which $m \to \infty$ and then $n \to \infty$. In this limit, the robust m-free energy (6.41) coincides with the $\log_t$-risk $\mathcal{R}_t(q)$. From the definition of t-Tsallis divergence (6.4), the $\log_t$-risk can be shown in turn to be equivalent to the minimization of the divergence

$$D_t\left(\mathcal{E}_t(\tilde{\nu}(x))||p(x|q)\right) \tag{6.48}$$

between the t-escort distribution (6.5) associated to the sampling distribution $\tilde{\nu}(x)$ and the ensemble predictive distribution $p(x|q)$. Therefore, unlike the standard Bayesian setup with $t = 1$, the minimizer of the robust m-free energy does not seek to approximate the sampling distribution $\tilde{\nu}(x)$. Instead, the minimizing ensembling posterior $q(\theta)$ aims at matching the t-escort version of the sampling distribution $\tilde{\nu}(x)$. In the case of corrupted data generation procedures, i.e., when $\nu(x) \neq \tilde{\nu}(x)$, recovering the sampling distribution $\tilde{\nu}(x)$ is not always the end goal, and, as shown by [138], escort distributions are particularly effective at reducing the contribution of OOD measures.

Example 2 (continued): Returning to Example 2, we now consider the performance of (m, t)-robust Bayesian learning for $m = 10$ and $t = 0.4$. The resulting distribution (blue line) with $m = 10$ and $t = 0.4$ seems to be able to better approximate the target distribution by reducing the effect of the outliers, while also taking advantage of ensembling to combat misspecification.

6.5.3 Influence Function Analysis

In this section, we study the robustness of the proposed free energy criterion by using tools from classical statistics. The robustness of an estimator is typically measured by the means of its influence function [147]. The influence function quantifies the extent to which an estimator derived from a data set $\mathcal{D}$ changes when a data point z is added to $\mathcal{D}$. We are specifically interested in quantifying the effect of data contamination, via the addition of a point z, on the ensembling distribution $q_t^m(\theta)$ that minimizes the proposed robust m-free energy objective (6.41). To this end, given a set $\mathcal{D}$ of n data points $\{x_1, \ldots, x_n\} \in \mathcal{X}^n$, we define the empirical measure

$$P^n(x) = \frac{1}{n}\sum_{i=1}^{n} \delta(x - x_i) \tag{6.49}$$

where $\delta(\cdot)$ denotes the Dirac function, and we introduce its γ-contaminated version for an additional data point $z \in \mathcal{X}$ as

$$P^n_{\gamma,z}(x) = \frac{(1-\gamma)}{n}\sum_{i=1}^{n} \delta(x - x_i) + \gamma\delta(x - z) \tag{6.50}$$

with $\gamma \in [0, 1]$.

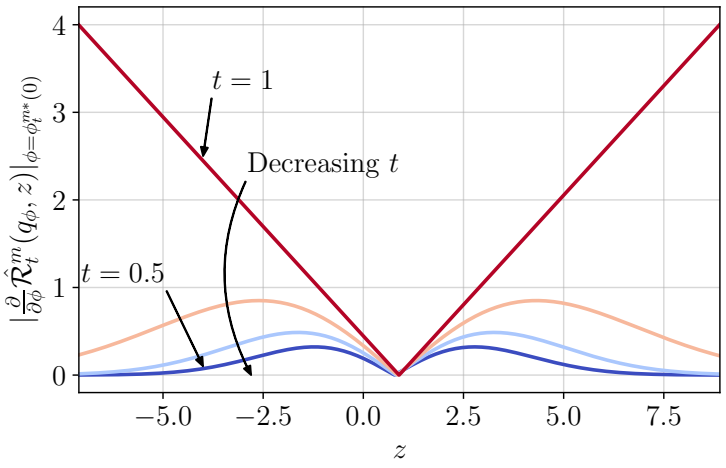


Figure 6.4: Absolute value of the contamination dependent term $\frac{\partial}{\partial\phi}\hat{\mathcal{R}}_t^m(q_\phi, z)$ evaluated at $\phi_t^{m*}(0)$ for different values of t. The predictive distribution of the ensemble model concentrates around 1.

The following analysis is inspired by [146], which considered Gibbs models trained using generalized free energy criteria based on the β-divergence and γ-divergence.

To compute the influence function we consider parametric ensembling distributions $q_\phi(\theta)$ defined by the parameter vector $\phi \in \Phi \subseteq \mathbb{R}^d$. We denote the robust m-free energy (6.41) evaluated using the empirical distribution (6.50) as

$$\mathcal{J}_t^m(\gamma, \phi) = \mathbb{E}_{P^n_{\gamma,z}(x)}\left[\hat{\mathcal{R}}_t^m(q_\phi, x)\right] + \frac{m}{\beta} D_1(q_\phi(\theta)||p(\theta)), \tag{6.51}$$

and its minimizer as

$$\phi_t^{m*}(\gamma) = \arg\min_{\phi\in\Phi} \mathcal{J}_t^m(\gamma, \phi). \tag{6.52}$$

The influence function is then defined as the derivative

$$IF_t^m(z, \phi, P^n) = \left.\frac{d\phi_t^{m*}(\gamma)}{d\gamma}\right|_{\gamma=0} \tag{6.53}$$

$$= \lim_{\gamma\to 0} \frac{\phi_t^{m*}(\gamma) - \phi_t^{m*}(0)}{\gamma}. \tag{6.54}$$

Accordingly, the influence function measures the extent to which the minimizer $\phi_t^{m*}(\gamma)$ changes for an infinitesimal perturbation of the data set.

Theorem 7. *The influence function of the robust m-free energy objective (6.51) is*

$$IF_t^m(z, \phi, P^n) = -\left[\frac{\partial^2 \mathcal{J}_t^m(\gamma, \phi)}{\partial\phi^2}\right]^{-1} \times \left.\frac{\partial^2 \mathcal{J}_t^m(\gamma, \phi)}{\partial\gamma\partial\phi}\right|_{\substack{\gamma=0 \\ \phi=\phi_t^{m*}(0)}}, \tag{6.55}$$

Table 6.1: Total variation (TV) distance between the ID measure $\nu(x)$ and the predictive distribution $p_q(x)$ obtained from the optimization of the different free energy criteria for the setting in Figure 6.5 (the TV values are scaled by 10^4).

	$t=1$ $\epsilon=0$	$t=1$ $\epsilon=0.1$	$t=0.9$ $\epsilon=0.1$	$t=0.8$ $\epsilon=0.1$
$\mathrm{TV}(\nu(x)\|\|p_q(x))$	1.38	2.15	1.88	1.79

where

$$\frac{\partial^2 \mathcal{J}_t^m(\gamma,\phi)}{\partial \phi^2} = \mathbb{E}_{P^n_{\gamma,z}(x)} \frac{\partial^2}{\partial \phi^2}\left[\hat{\mathcal{R}}_t^m(q_\phi, x)\right] \tag{6.56}$$

$$+ \frac{\partial^2}{\partial \phi^2}\left[\frac{m}{\beta} KL(q_\phi(\theta)||p(\theta))\right] \tag{6.57}$$

and

$$\frac{\partial^2 \mathcal{J}_t^m(\gamma,\phi)}{\partial \gamma \partial \phi} = \frac{\partial}{\partial \phi}\left[\mathbb{E}_{P^n(x)}\left[\hat{\mathcal{R}}_t^m(q_\phi, x)\right] - \hat{\mathcal{R}}_t^m(q_\phi, z)\right]. \tag{6.58}$$

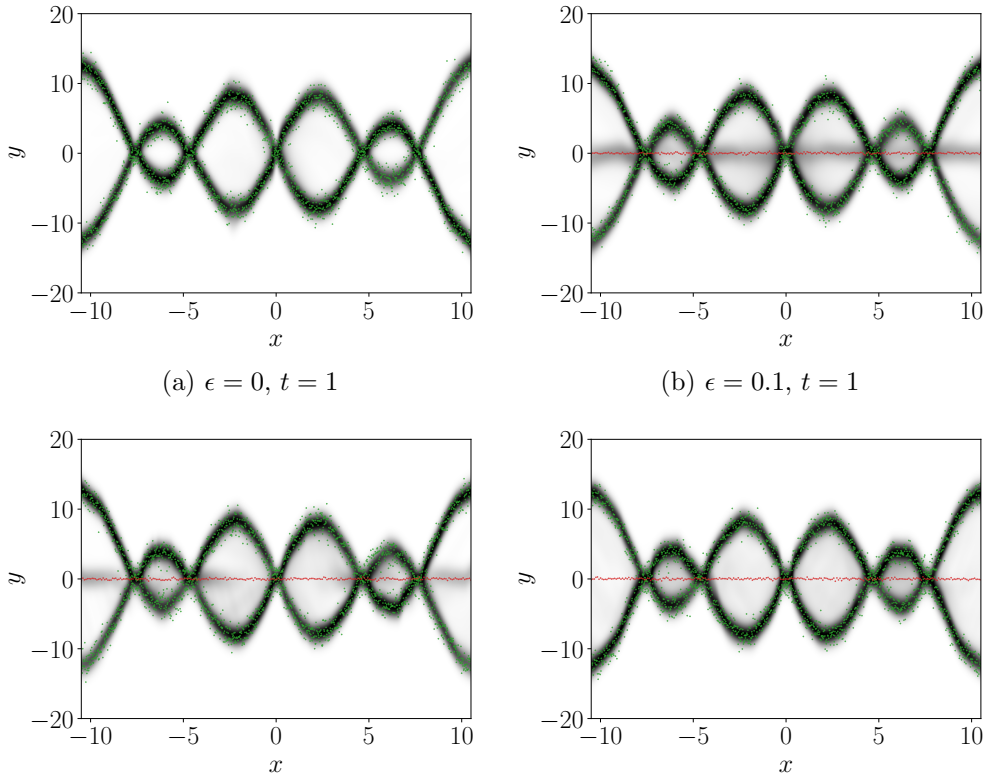
Theorem 7 quantifies the impact of the data point z through the contamination dependent term $\frac{\partial}{\partial \phi}\hat{\mathcal{R}}_t^m(q_\phi, z)$. We study the magnitude of this term to illustrate the enhanced robustness deriving from the proposed robust m-free energy objective. For ease of tractability, we consider the limit $m \to \infty$. In this case, the contamination dependent term can be expressed as

$$\frac{\partial}{\partial \phi} \lim_{m\to\infty} \hat{\mathcal{R}}_t^m(q_\phi, z) = \frac{\partial}{\partial \phi} \log_t \mathbb{E}_{q_\phi(\theta)}[p(z|\theta)] \tag{6.59}$$

$$= \left[\mathbb{E}_{q_\phi(\theta)}[p(z|\theta)]\right]^{-t} \frac{\partial \mathbb{E}_{q_\phi(\theta)}[p(z|\theta)]}{\partial \phi}. \tag{6.60}$$

The effect of the t-logarithm function thus appears in the first multiplicative term, and it is the one of reducing the influence of anomalous data points to which the ensemble predictive distribution $p_q(x)$ assigns low probability.

Example: To illustrate how the t-logarithm improves the robustness to outlying data points, we consider again the example of Figure 6.3 and we assume a parametrized ensembling posterior $q_\phi(\theta) = \mathcal{N}(\theta|\phi, 1)$. In Figure 6.4, we plot the magnitude of the contamination dependent term evaluated at the parameter $\phi_t^{m*}(0)$ that minimizes the robust m-free energy $\mathcal{J}_t^m(0,\phi)$ for $m = \infty$ and different values of t. For all values of t, the optimized predictive distribution concentrates around 0, where most of sampled data points lie. However, as the value of the contaminated data point z becomes smaller and moves towards regions where the ensemble assign low probability, the contamination dependent term grows linearly for $t = 1$, while it flattens for $t \in (0, 1)$. This showcases the role of the robust m-free energy criterion as a tool to mitigate the influence of outlying data points by setting $t < 1$.



(a) $\epsilon = 0$, $t = 1$

(b) $\epsilon = 0.1$, $t = 1$

(c) $\epsilon = 0.1$, $t = 0.9$

(d) $\epsilon = 0.1$, $t = 0.8$

Figure 6.5: Ensemble predictive distribution obtained minimizing different free energy criteria. The samples from the ID measure are represented as green dots, while data points sampled from the OOD component are in red. The optimized predictive distributions are displayed in shades of gray. In (a), we plot the predictive distribution associated to $(m, 1)$-robust Bayesian learning obtained minimizing the m-free energy criterion $\mathcal{J}^m$ of [3] with $m = 20$ by using only samples from the ID measure (i.e., there are no outliers). In (b), we show the predictive distribution obtained by minimizing the same criterion when using samples from the ID measure and OOD measure with a contamination ratio $\epsilon = 0.1$. In (c) and (d) we consider the same scenario as in (b), but we consider the proposed (m, t)-robust Bayesian based on the robust m-free energy criterion $\mathcal{J}_t^m$ with $m = 20$, when setting $t = 0.9$ and $t = 0.8$, respectively.

6.6 Experiments

In this section, we first describe a simple regression task with an unimodal likelihood, and then we present results for larger-scale classification and regression tasks. The main aim of these experiments is to provide qualitative and quantitative insights into the performance of $(m, 1)$-robust Bayesian learning of [3] and the proposed robust (m, t)-robust Bayesian learning. Both examples are characterized by model misspecification and outliers.

6.6.1 Multimodal Regression

For the first experiment, we modify the regression task studied by [52] and [3] in order to capture not only model misspecification but also the presence of outliers as in the contamination model (6.31). To this end, we assume that the ID distribution $\nu(x)$, with $x = (a, b)$, is given by $\nu(a, b) = p(a)\nu(b|a)$, where the covariate a is uniformly distributed in the interval $[-10.5, 10.5]$ – i.e., $p(a) = 1/21$ in this interval and $p(a) = 0$ otherwise – and by a response variable b that is conditionally distributed according to the two-component mixture

$$\nu(b|a) = \mathcal{N}(b|\alpha\mu_a, 1), \tag{6.61}$$

$$\alpha \sim \text{Rademacher}, \tag{6.62}$$

$$\mu_a = 7\sin\left(\frac{3a}{4}\right) + \frac{a}{2}. \tag{6.63}$$

The OOD component $\xi(x) = \xi(a, b) = p(a)\xi(b)$ also has a uniformly distributed covariate a in the interval $[-10.5, 10.5]$, but, unlike the ID measure, the response variable b is independent of a, with a distribution concentrated around $b = 0$ as

$$\xi(b) = \mathcal{N}(b|0, 0.1). \tag{6.64}$$

The parametric model is given by $p(x|\theta) = p(a, b|\theta) = p(a)\mathcal{N}(b|f_\theta(a), 1)$, where $f_\theta(a)$ is the output of a three-layer fully connected Bayesian neural network with 50 neurons and Exponential Linear Unit (ELU) activation functions [161] in the two hidden layers. We consider a Gaussian prior $p(\theta) = \mathcal{N}(0, I)$ over the neural network weights and use a Monte Carlo estimator of the gradient based on the reparametrization trick [162] as in [163].

Consider first only the effect of misspecification. The parametric model assumes a unimodal likelihood $\mathcal{N}(b|f_\theta(a), 1)$ for the response variable, and is consequently misspecified with respect to the ID measure (6.61). As a result, the standard Bayesian learning leads to a unimodal predictive distribution that approximates the mean value of the response variable, while $(m, 1)$-robust Bayesian learning can closely reproduce the data distribution [3, 52]. This is shown in Figure 6.5a, which depicts the predictive distribution obtained by minimizing the m-free energy criterion $\mathcal{J}^m$ with $m = 20$ when using exclusively samples from the ID measure (green dots). In virtue of ensembling, the resulting predictive distribution becomes multimodal, and it is seen to provide a good fit to the data from the ID measure.

Let us evaluate also the effect of outliers. To this end, in Figure 6.5b we consider $(m, 1)$-robust Bayesian learning and minimize again the m-free energy criterion, but this time using a data set contaminated with samples from the OOD component (red points) and with a contamination ratio $\epsilon = 0.1$. The predictive distribution is seen to cover not only the ID samples but also the outlying data points. In Figure 6.5c and 6.5d, we finally plot the predictive distributions obtained by (m, t)-robust Bayesian learning with $m = 20$, when setting $t = \{0.9, 0.8\}$, respectively. The proposed approach is able to mitigate the effect of the outlying component for $t = 0.9$, and, for $t = 0.8$, it almost completely suppresses it. As a result, the proposed energy criterion produces predictive distributions that match more closely the ID measure. This qualitative behavior is quantified in Table 6.1, where we report the total variation distance from the ID measure for the setting and predictors considered in Figure 4.

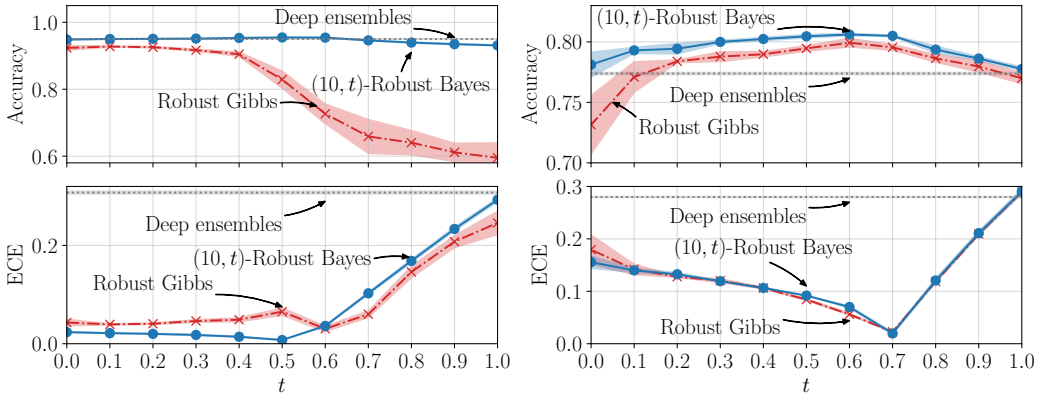
6.6.2 MNIST and CIFAR-10 Classification Tasks

We now address the problem of training Bayesian neural network classifiers in the presence of misspecification and outliers. We consider three different experimental setups entailing distinct data sets and model architectures:

- Classification of MNIST digits [164] based on a fully connected neural network comprising a single hidden layer with 25 neurons.
- Classification of Extended MNIST characters and digits [132] based on a fully connected neural network with two hidden layers with 25 neurons each.
- Classification of CIFAR-10 [133] images using a convolutional neural network (CNN) with two convolutional layers, the first with 8 filters of size 3×3 and the second with 4 filters of size 2×2, followed by a hidden layer with 25 neurons each.

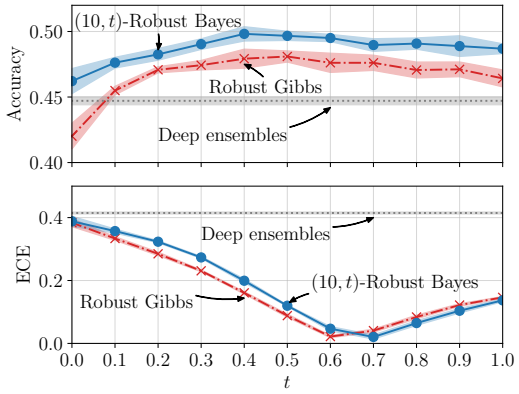
All hidden units use ELU activations [161] except the last, classifying, layer that implements the standard softmax function. Model misspecification is enforced by adopting neural network architectures with small capacity. As in [5], outliers are obtained by randomly modifying the labels for fraction ϵ of the data points in the training set. Additional details for the experiments can be found in the supplementary material.

We measure the accuracy of the trained models, as well as their calibration performance. Calibration refers to the capacity of a model to quantify uncertainty (see, e.g., [4]). We specifically adopt the expected calibration error (ECE) [1], a standard metric that compares model confidence to actual test accuracy (see supplementary material for the exact definition). We train the classifiers using corrupted data sets with a contamination ratio $\epsilon = 0.3$, and then we evaluate their accuracy and ECE as a function of $t \in [0, 1]$ based on a clean ($\epsilon = 0$) holdout data set. We compare the performance of (m, t)-robust Bayesian learning based on the minimization of the robust m-free energy $\mathcal{J}_t^m$, with $m = 10$, to: *(i) deep ensembles* [4], also with 10 models in the ensembles; and *(ii)* the robust Gibbs predictor of [5], which optimizes over a single predictor (not an ensemble) by minimizing the free energy metric $\mathcal{J}_t^1$. The inverse temperature parameter β is set to 0.1 in the (m, t)-robust Bayesian and the Gibbs predictor objectives.



(a) MNIST data set

(b) Extended MNIST data set



(c) CIFAR-10 data set

Figure 6.6: Test accuracy (top) and expected calibration error (ECE) (bottom) as a function of t under the contamination ratio $\epsilon = 0.3$ for: *(i)* deep ensembles [4]; *(ii)* robust Gibbs predictor, which minimizes the free energy criterion $\mathcal{J}_t^1$ [5]; and *(iii)* (m, t)-robust Bayesian learning, which minimizes the free energy criterion $\mathcal{J}_t^{10}$.

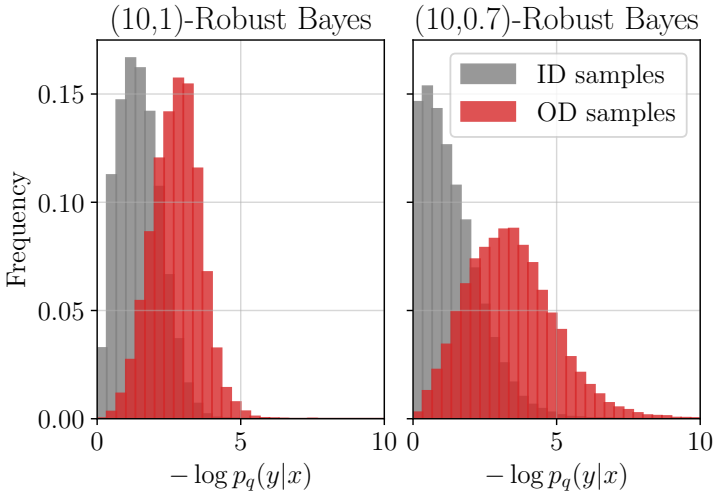


Figure 6.7: Distribution of the negative log-likelihood of ID and OOD training data samples for an ensemble model minimizing (on the left) the log-loss based criterion $\mathcal{J}_1^{10}$, and (on the right) the proposed robust objective $\mathcal{J}_{0.7}^{10}$ based on the $\log_t$-loss with $t = 0.7$.

In Figure 6.6 we report the performance metrics attained by the trained models in the three different setups listed above. From the top panels we conclude that (m, t)-robust Bayesian learning is able to mitigate model misspecification by improving the final accuracy as compared to the robust Gibbs predictor and the deep ensemble models. Furthermore, the use of the robust loss for a properly chosen value of t leads to a reduction of the detrimental effect of outliers and to an increase in the model accuracy performance as compared to the standard log-loss ($t = 1$). In terms of calibration performance, the lower panels demonstrate the capacity of robust ensemble predictors with $t < 1$ to drastically reduce the ECE as compared to deep ensembles. In this regard, it is also observed that the accuracy and ECE performance levels depend on the choice of parameter t. In practice, the selection of t may be addressed using validation or meta-learning methods in a manner akin to [165]. Additional results on calibration in the form of reliability diagrams [166] can be found in supplementary material.

As shown shown theoretically in Section 6.5.3, the effect of the $\log_t$-loss is to reduce the influence of outliers during training for $t < 1$. We empirically investigate the effect of the robust loss in Figure 6.7, in which we compare the distribution of the negative log-likelihood for ID and OOD training data samples. We focus on the CIFAR-10 data set, and we compare the histogram of the negative log-likelihood under a CNN model trained based on the m-free energy $\mathcal{J}_1^m$, with $m = 10$ and standard logarithmic loss, and a CNN minimizing the proposed robust m-free energy $\mathcal{J}_t^m$, with $m = 10$ and $t = 0.7$. The $(m, 1)$-robust Bayesian based on the standard log-loss tries to fit both ID and OOD samples and, as a result, the two components have similar likelihoods. In contrast, (m, t)-robust Bayesian learning is able to downweight the influence of outliers and to better fit the ID component.

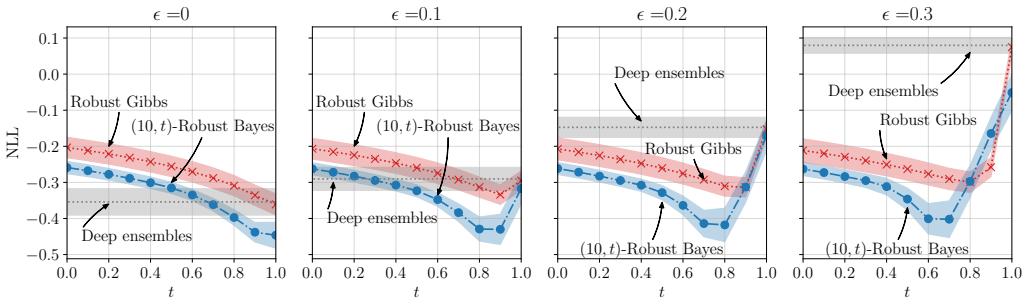


Figure 6.8: Negative log-likelihood computed on a uncorrupted data set for: *(i)* deep ensembles [4]; *(ii)* robust Gibbs predictor, which minimizes $\mathcal{J}_t^1$ [5]; and *(iii)* the (m, t)-robust Bayesian learning, which minimizes $\mathcal{J}_t^{10}$. The models are trained on ϵ-contaminated data set for $\epsilon \in \{0, 0.1, 0.2, 0.3\}$

6.6.3 California Housing Regression Task

Finally, we consider the problem of training a robust regressor based on training data sets corrupted by outliers and in the presence of model misspecification. We consider the California housing dataset, which is characterized by response variables y normalized in the $[0, 1]$ interval, and we fix a unimodal likelihood $p(y|x, \theta) = \mathcal{N}(y|f_\theta(x), 0.1)$, where $f_\theta(x)$ is the output of a three-layer neural network with hidden layers comprising 10 units with ELU activation functions [161]. The model class is misspecified since the response variable is bounded and hence not Gaussian. Outliers are modeled by replacing the label of fraction ϵ of the training sample with random labels picked uniformly at random within the $[0, 1]$ interval.

We consider training based on data sets with different contamination ratios $\epsilon \in \{0, 0.1, 0.2, 0.3\}$, and measure the trained model ability to approximate the ID data by computing the negative log-likelihood on a clean holdout data set ($\epsilon = 0$). As in the previous subsection, we compare models trained using (m, t)-robust Bayesian learning, with $m = 5$, to: *(i) deep ensembles* [4], also with 5 models in the ensembles; and *(ii)* the robust Gibbs predictor of [5] minimizing the free energy metric $\mathcal{J}_t^1$. The inverse temperature parameter β is set to 0.1 in the (m, t)-robust Bayesian and the Gibbs predictor objectives.

In Figure 6.8 we report the negative log-likelihood of an uncontaminated data set for models trained according to the different learning criteria. The leftmost panel ($\epsilon = 0$) corresponds to training based on an uncontaminated data set. For this case, the best performance is obtained for $t = 1$ – an expected result due to the absence of outliers – and the proposed criterion outperforms both the Gibbs predictor and deep ensembles, as it is capable of counteracting misspecification by the means of ensembling. In the remaining panels, training is performed based on ϵ-contaminated data sets, with the contamination ϵ increasing from left to right. In these cases, learning criteria based on robust losses are able to retain similar performance to the uncontaminated case for suitable chosen values of t. Furthermore, the optimal value of t is observed to increase with the fraction

of outliers in the training data set.

6.7 Conclusion

In this work, we addressed the problem of training ensemble models under model misspecification and in the presence of outliers. We proposed the (m, t)-robust Bayesian learning framework that leverages generalized logarithm score functions in combination with multi-sample bounds, with the goal of deriving posteriors that are able to take advantage of ensembling, while at the same time being robust with respect to outliers. The proposed learning framework is shown to lead to predictive distributions characterized by better generalization capabilities and calibration performance in scenarios in which the standard Bayesian posterior fails.

The proposed robust Bayesian learning framework can find application to learning scenarios that can benefit from uncertainty quantification in their decision making processes and are characterized by the presence of outliers and model misspecification. Examples include inference in wireless communication systems [167], medical imaging [168] and text sentiment analysis [169, 170].

We conclude by suggesting a number of directions for future research. The (m, t)-robust Bayesian learning has been shown to lead to the largest performance gains for properly chosen values of t. The optimal values of t depend on the particular task at hand, and deriving rules to automate the tuning of these parameters represents a practical and important research question. Furthermore, (m, t)-robust Bayesian learning can be extended to reinforcement learning, as well as to meta-learning, for which Bayesian methods have recently been investigated (see, e.g., [171, 172] and references therein).

Chapter 7

Robust Bayesian Learning Applications to Wireless Communication

This chapter takes a critical look at the application of conventional machine learning methods to wireless communication problems through the lens of reliability and robustness. Deep learning techniques adopt a frequentist framework, and are known to provide poorly calibrated decisions that do not reproduce the true uncertainty caused by limitations in the size of the training data. Bayesian learning, while in principle capable of addressing this shortcoming, is in practice impaired by model misspecification and by the presence of outliers. Both problems are pervasive in wireless communication settings, in which the capacity of machine learning models is subject to resource constraints and training data is affected by noise and interference. In this context, we explore the application of the framework of *robust* Bayesian learning developed in Chapter 6. We showcase the merits of robust Bayesian learning on several important wireless communication problems in terms of accuracy, calibration, and robustness to outliers and misspecification.

7.1 Introduction

Artificial intelligence (AI) is widely viewed as a key enabler of 6G wireless systems. Research on this topic has mostly focused on identifying use cases and on mapping techniques from the vast literature on machine learning to given problems [173–175]. At a more fundamental level, there have been efforts to integrate well-established communication modules, e.g., for channel encoding and decoding, with data-driven designs, notably via tools such as model unrolling [176, 177]. All these efforts have largely relied on *deep learning* libraries and tools. The present paper takes a critical look at the use of this conventional methodology through the lens of *reliability* and *robustness* . To this end, we explore the potential benefits of the alternative design framework of *robust Bayesian learning* by focusing on several key wireless communication applications, namely modulation classification, indoor and outdoor localization, and channel modeling and

simulation.

7.1.1 Frequentist vs. Bayesian Learning

In *frequentist* learning, the output of the training process is a single model – typically, a single vector of weights for a neural network – obtained by minimizing the training loss. This approach is justified by the use of the training loss as an estimate of the population loss, whose computation would require averaging over the true, unknown distribution of the data. This estimate is only accurate in the presence of sufficiently large data sets. While abundant data is common in the benchmark tasks studied in the computers science literature, the reality of many engineering applications is that data are often scarce. In wireless communications, the problem is particularly pronounced at the physical layer, in which fading dynamics imply short stationary intervals for data collection and training [178–181].

The practical upshot of the reliance on frequentist learning is that, in the presence of limited data, decisions made by AI models tend to be *poorly calibrated*, providing confidence levels that do not match their true accuracy [1, 152]. As a result, an AI model may output a decision with some level of confidence, say 95%, while the accuracy of the decision is significantly lower. This is an issue problem in many engineering applications, including emerging communication networks (e.g., 5G and beyond), in which a more or less confident decision should be treated differently by the end user [182].

The framework of *Bayesian learning* addresses the outlined shortcomings of frequentist learning [183, 184]. At its core, Bayesian learning optimizes over a *distribution* over the model parameter space, which enables it to quantify uncertainty arising from limited data. In fact, if several models fit the data almost equally well, Bayesian learning does not merely select one of the models, disregarding uncertainty; rather it assigns similar distribution values to all such models [185]. This way, decisions produced by AI modules trained via Bayesian learning can account for the "opinions" of multiple models by averaging their outputs using the optimized distribution [54, 186]. Bayesian learning has recently been applied in [152] by focusing on the problem of demodulation over fading channels; as well as in [187] for detection over multiple-antenna channels.

7.1.2 Robust Bayesian Learning

Like frequentist learning, Bayesian learning assumes that the distribution underlying training data generation is the same as that producing test data. Furthermore, Bayesian learning implicitly assumes that the posited model – namely likelihood and prior distribution – is sufficiently close to the true, unknown data-generating distribution to justify the use of the posterior distribution as the optimized distribution in the model parameter space. As a result, the benefit of Bayesian learning is degraded when data is affected by outliers and/or when the model is misspecified.

In Chapter 6 we have addressed both of these limitations, introducing a generalized framework that we will refer to as *robust Bayesian learning*. Robust Bayesian learning aims at providing well-calibrated, and hence reliable, decisions even in the presence of model misspecification and of discrepancies between training and testing conditions.

Model misspecification has been addressed in [3, 156]. These papers start from two observations. The first is that Bayesian learning can be formulated as the minimization of a *free energy* metric, which involves the average of the training loss, as well as an information-theoretic regularizing term dependent on a prior distribution. The conventional free energy metric can be formally derived as an upper bound on the population loss within the theoretical framework of *PAC Bayes theory* [188–190]. The second observation is that, in the presence of model misspecification, *model ensembling* can be useful in combining the decisions of different models that may be specialized to distinct parts of the problem space. Using these two observations, references [3, 156] introduced alternative free energy criteria that are based on a tighter bound of the population loss for ensemble predictors.

To address the problem of outliers (see e.g. [139]), different free energy criteria have been introduced, which are less sensitive to the presence of outliers. These metrics are based on divergences, such as β-divergences [142, 143] and γ-divergence [144, 145], which generalize the Kullback-Liebler divergence underlying the standard free energy metric. Finally, a unified framework has been introduced in [167] that generalizes the free energy metrics introduced in [3, 156]. The approach is robust to misspecification, while also addressing the presence of outliers.

7.1.3 Main Contributions

In the following, we explore the application of robust Bayesian learning to wireless communication systems. We detail applications of robust Bayesian learning to communication systems, focusing on automated modulation classification (AMC), received signal strength indicator (RSSI)-based localization, as well as channel modeling and simulation. These applications have been selected in order to highlight the importance of considering uncertainty quantification, in addition to accuracy, while also emphasizing the problems of model misspecification and outliers in wireless communications [55–57].

Our specific contributions are as follows.

- As a first application, we focus on the AMC problem for intelligent spectrum sensing [191]. In this setting, the necessity of deploying lightweight models that satisfy the strict computational requirements of network edge devices can give rise to model misspecification. At the same time, the training data sets often contain non-informative outliers due to interfering transmissions from other devices. We demonstrate that robust Bayesian learning yields classifiers with good calibration performance despite model misspecification and the presence of outliers.

- As a second application, we study node localization based on crowdsourced RSSI data sets [192]. Such data sets typically contain inaccurately reported location measurements due to imprecise or malicious devices. Furthermore, owing to the complex relation between RSSI measurements and device locations, learning often happens using misspecified model classes. In this context, we demonstrate that robust Bayesian is able to properly estimate residual uncertainty about the transmitters' locations in spite of the presence of outliers and misspecified model classes.

- Finally, we apply robust Bayesian learning to the problem of channel modeling and simulation. We show via experiments that robust Bayesian learning produces accurate and well-calibrated generative models even in the presence of outlying data points.

7.2 Robust and Calibrated Automatic Modulation Classification

As a first application of robust Bayesian learning we consider the AMC problem. This is the task of classifying received baseband signals in terms of the modulation scheme underlying their generation. The relation between the received signal and the chosen modulation scheme is often mediated by complex propagation phenomena, as well as hardware non-idealities at both the receiver and the transmitter side. As a result, model-based AMC methods often turn out to be inaccurate because of the overly simplistic nature of the assumed models [193]. In contrast, machine learning based AMC has been shown to be extremely effective in correctly classifying received signals based on signal features autonomously extracted from data [6]. We refer to [194] and references therein for a comprehensive overview.

All prior works on learning-based AMC, reviewed in [194], have adopted frequentist learning. In this section, we consider the practical setting in which AMC must be implemented on resource-constrained devices, entailing the use of small, and hence mismatched, models; and the training data sets are characterized by the presence of outliers due to interference.

7.2.1 Problem Definition and Performance Metrics

The AMC problem can be framed as an instance of supervised classification, with the training data set $\mathcal{D}$ comprising pairs (x, y) of discrete-time received baseband signal x and modulation label y, with $\mathcal{Y}$ being the set of possible modulation schemes. Each training data point $(x, y) \in \mathcal{D}$ is obtained by transmitting a signal with a known modulation $y \in \mathcal{Y}$ over the wireless channel, and then recording the received discrete-time vector x at the receiver end. The outlined procedure determines the unknown sampling distribution $\nu_s(x, y)$.

We evaluate the performance of AMC on a testing data set $\mathcal{D}_{te}$ in terms of accuracy and *calibration*. To describe calibration performance metrics, let us consider a predictive distribution $p(y|x)$, which may be the frequentist distribution $p(y|x, \theta^{\text{freq}})$, or the ensemble distribution (6.20) in the cases of Bayesian learning and robust Bayesian learning. A hard prediction $\hat{y}$ is obtained as the maximum-probability solution

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} p(y|x). \tag{7.1}$$

The corresponding *confidence score* assigned by the predictor $p(y|x)$ is the probability $p(\hat{y}|x) \in [0, 1]$. The calibration of a classifier measures the degree to which the confidence score $p(\hat{y}|x) \in [0, 1]$ reflects the true probability of correct classification $P[\hat{y} = y|x]$ conditioned on the input x.

We adopt the standard reliability diagrams [166] and the expected calibration error as diagnostic tools for the calibration performance [1]. Both metrics require binning the output of the classifier confidence score $p(\hat{y}|x)$ into M intervals of equal size, and then grouping the testing data points $(x, y) \in \mathcal{D}_{te}$ based on the index of the bin for the confidence score $p(\hat{y}|x)$. For each bin $\mathcal{B}_m$, the *within-bin accuracy* is defined as

$$\text{Acc}(\mathcal{B}_m) = \frac{1}{|\mathcal{B}_m|} \sum_{(x,y)\in\mathcal{B}_m} \mathbb{1}\{\hat{y} = y\}, \tag{7.2}$$

which measures the fraction of test samples within the bin that are correctly classified; and the *within-bin confidence* as

$$\text{Conf}(\mathcal{B}_m) = \frac{1}{|\mathcal{B}_m|} \sum_{(x,y)\in\mathcal{B}_m} p(\hat{y}|x), \tag{7.3}$$

which is the average confidence level for the test samples within the bin.

The *reliability diagram* plots within-bin accuracy and within-bin confidence as a function of the bin index m. As a result, a reliability diagram visualizes the relation between confidence and accuracy of a predictor, establishing whether a classifier is over-confident ($\text{Conf}(\mathcal{B}_m) > \text{Acc}(\mathcal{B}_m)$), under-confident ($\text{Conf}(\mathcal{B}_m) < \text{Acc}(\mathcal{B}_m)$) or well-calibrated ($\text{Conf}(\mathcal{B}_m) \approx \text{Acc}(\mathcal{B}_m)$).

The *expected calibration error (ECE)* summarizes the calibration performance of a classifier as a single number obtained as the weighted sum of the absolute difference between within-bin accuracy and within-bin confidence, namely

$$\text{ECE} = \sum_{m=1}^{M} \frac{|\mathcal{B}_m|}{\sum_{m=1}^{M} |\mathcal{B}_m|} \left|\text{Conf}(\mathcal{B}_m) - \text{Acc}(\mathcal{B}_m)\right|. \tag{7.4}$$

By this definition, one can generally conclude that a lower ECE indicates a better calibrated predictor.

7.2.2 Data Set

We adopt the *DeepSIG: RadioML 2016.10A* data set [6]. This is a synthetic data set that contains 220K vectors of I/Q samples of signals comprising 8 digital modulation schemes (BPSK, QPSK, 8PSK, 16QAM, 64QAM, BFSK, CPFSK) and 3 analog modulations (WB-FM, AM-SSB, AM-DSB). We focus on the problem of classifying the 8 digital modulation schemes using received signals recorded at different SNR levels ranging from 0 dB to 18 dB. Furthermore, we model the presence of *interference* during training by generating an ϵ-contaminated version of the original data set. In it, with probability $\epsilon \in [0, 1)$, the original training sample x is summed to an interfering signal x' picked uniformly at random from the data set. Note that the interfering signal can be possibly generated from a different modulation scheme. Using Definition 2, the samples affected by interference represent *outliers*, since no interference is assumed during testing. We consider 30% of the available samples for training; 20% of the samples for validation; and the remaining 50% for testing. The use of a small training data set is intentional, as we wish to focus on a regime characterized by data scarcity.

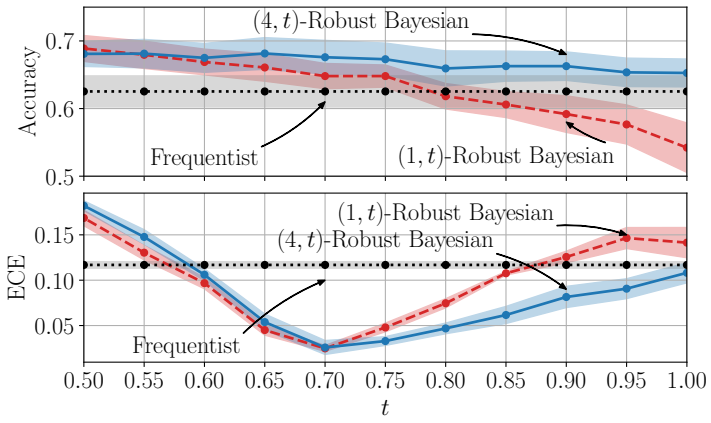


Figure 7.1: Average test accuracy and ECE for AMC over the DeepSIG: RadioML 2016.10A data set [6] for frequentist and (m, t)-robust Bayesian learning as a function of the parameter t. The test set is free from interference, while the training set is subject to interference ($\epsilon = 0.5$).

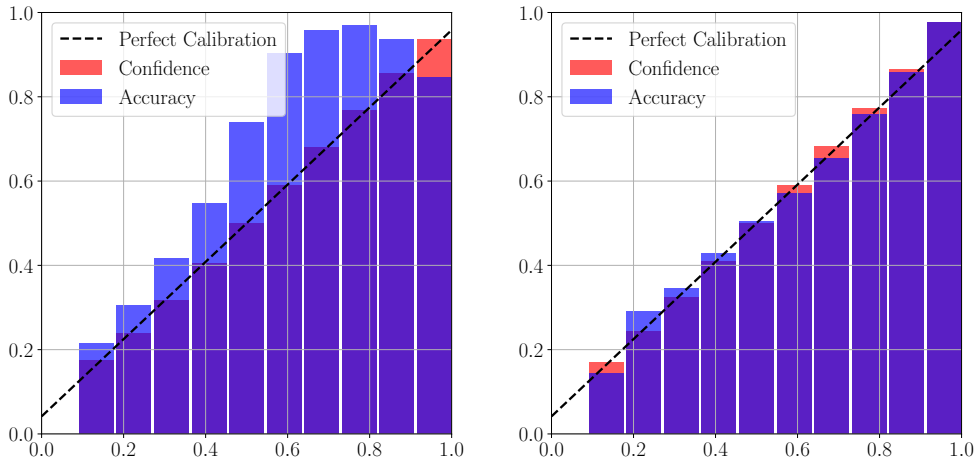
7.2.3 Implementation

We implement a lightweight convolutional neural network (CNN) architecture comprising of two convolutional layers followed by two linear layers with 30 neurons each. The first convolutional layer has 16 filters of size 2×3, and the second layer has 4 filters of size 1×2. We adopt the Exponential Linear Unit (ELU) activation with parameter $\alpha = 1$. The lightweight nature of the architecture is motivated by the strict computational and memory requirements at network edge devices. As a result, the CNN model is generally *misspecified*, in the sense that, following Definition 1, the complex relation between received signal and chosen modulation scheme cannot be exactly represented using the model.

In the training data set, half of the samples are affected by interference, i.e., $\epsilon = 0.5$. For Bayesian learning, we adopt a Gaussian variational distribution $q(\theta) = \mathcal{N}(\theta|\mu, \Sigma)$ over the CNN model parameter vector θ. Accordingly, the mean μ and diagonal covariance matrix Σ are optimized, while we fix the prior $p(\theta) = \mathcal{N}(\theta|0, I)$. Optimization for both frequentist and Bayesian methods is carried out via Adam with a learning rate $\eta = 0.001$, and the reparametrization trick is implemented for Bayesian learning [162]. In our experiments we set $\beta = 0.01$. The number of samples used to evaluate the ensemble prediction (6.20) is $m = 10$. Note that this may differ from the value of m used to define the training criterion.

7.2.4 Results

In Figure 7.1 we report the average test accuracy and ECE for frequentist and (m, t)-robust Bayesian with different values of m as a function of t. The main observation is



(a) Frequentist Learning (b) Robust Bayesian Learning

Figure 7.2: Reliability diagrams for frequentist (left) and (m, t)-robust Bayesian learning for $m = 4$ and $t = 0.7$ (right) for AMC over the DeepSIG: RadioML 2016.10A data set [6].

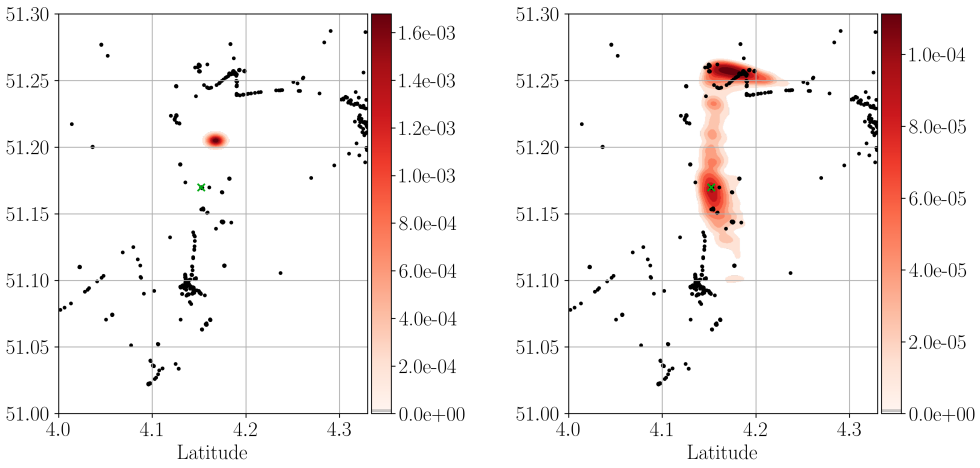
that, with suitably chosen parameters (m, t), robust Bayesian learning can outperform standard frequentist learning both in terms of accuracy and calibration for $t < 1$. The smallest ECE is obtained by robust Bayesian learning for $t = 0.7$, and it is five times smaller compared to the one obtained using conventional Bayesian learning ($t = 1$). Overall, (m, t)-robust Bayesian paradigm is able to improve the final accuracy by 5% and to reduce the ECE by five times via suitable choice of parameters (m, t).

To further elaborate on the calibration performance, in Figure 7.2 we compare the reliability diagrams obtained via frequentist and (m, t)-robust Bayesian learning for $m = 4$ and $t = 0.7$. While frequentist learning provides under-confident predictions, robust Bayesian learning offers well-calibrated predictions that consistently offer a small discrepancy between accuracy and confidence levels.

7.3 Robust and Calibrated RSSI-Based Localization

Table 7.1: Test negative log-likelihood for RSSI localization (7.6) with $t = 1$ and no outliers ($\epsilon = 0$). The case $m = 1$ corresponds to conventional Bayesian learning.

	$m = 1$	$m = 2$	$m = 10$
SigfoxRural	1.70 ± 1.03	-0.43 ± 0.61	$\mathbf{-1.59 \pm 0.36}$
UTSIndoor	4.33 ± 2.32	2.25 ± 1.69	$\mathbf{2.17 \pm 1.76}$
UJIIndoor	4.86 ± 1.02	2.74 ± 0.46	$\mathbf{1.44 \pm 0.33}$



(a) Bayesian Learning

(b) Robust Bayesian Learning

Figure 7.3: Predictive distribution $p(y|x)$ as a function of the estimated position of the transmitter y, where x is the RSSI vector associated to the true location shown as a green cross. The black dots correspond to the locations recorded in the *SigfoxRural* data set. The left panel shows the predictive distribution for Bayesian learning, while the right panel depicts the predictive distribution for (m, t)-robust Bayesian learning with $m = 10$ and $t = 1$. No outliers are considered in the training set, i.e., $\epsilon = 0$.

In this section, we turn to the problem of localization. In outdoor environments, accurate localization information of a wireless device can be obtained leveraging the global navigation satellite system (GNSS). However, the performance of satellite-based positioning is severely degraded in indoor environments [195], and its power requirements are not compatible with IoT application characterized by ultra-low power consumption [196]. For this reason, alternative techniques have been investigated that rely on so-called *channel fingerprints*, i.e., feature extracted from the received wireless signals [197].

Among such methods, the use of *received signal strength indicators* (RSSI) measured at multiple wireless access points has been shown to provide an accessible, yet informative, vector of features. Owing to the complexity of defining explicit models relating the device location $y \in \mathcal{Y}$ with the RSSI-measurements vector $x \in \mathcal{X}$, data-driven RSSI-based localization techniques have been recently explored [198, 199]. The outlined prior work in this area has focused on machine learning models trained using the conventional frequentist approach.

In this section, we study a setting in which the training data set is collected using noisy, e.g., *crowd-sourced*, fingerprints. As such, the training set contains outliers. Furthermore, we aim at developing strategies, based on robust Bayesian learning, which can offer accurate localization, while also properly quantifying residual uncertainty.

7.3.1 Problem Definition and Performance Metrics

The RSSI-based localization problem is a supervised regression task. In it, a training sample (x, y) is obtained by measuring the RSSI fingerprint y corresponding to the transmission of a reference signal at a device located at a known position x. The general goal is to train a machine learning model $p(y|x)$ to predict the location y associated to a RSSI vector x so as to optimize accuracy and uncertainty quantification.

Given a test data set $\mathcal{D}_{te}$ and assuming that the predictive location is the mean of the predictive distribution, i.e. $\bar{y} = \mathbb{E}_{p(y|x)}[y]$, we adopt the *mean squared error (MSE)* metric

$$\mathrm{MSE}(\mathcal{D}_{te}, p) = \frac{1}{|\mathcal{D}_{te}|} \sum_{(x,y)\in\mathcal{D}_{te}} \|y - \bar{y}\|_2 \tag{7.5}$$

as a measure of accuracy. Furthermore, in order to estimate the residual uncertainty about y predicted by the model, we adopt the *negative test log-likelihood* [200]

$$\mathrm{NLL}(\mathcal{D}_{te}, p) = -\frac{1}{|\mathcal{D}_{te}|} \sum_{(x,y)\in\mathcal{D}_{te}} \log(p(y|x)). \tag{7.6}$$

Note that the negative log-likelihood is large if the model assigns a small probability density $p(y|x)$ to the correct output y.

7.3.2 Data Sets

We experiment on different publicly available RSSI fingerprint data sets, encompassing both outdoor and indoor conditions:

- The *SigfoxRural* data set [196] comprises $25,638$ Sigfox messages measured at 137 base stations and emitted from vehicles roaming around a large rural area (1068 km^2) between Antwerp and Gent.
- The *UTSIndoorLoc* data set [201] contains 9494 WiFi fingerprints sampled from 589 access points inside the FEIT Building at the University of Technology of Sydney, covering an area of $44,000$ m^2.
- The *UJIIndoorLoc* data set [202] contains $21,049$ WiFi fingerprints measured at 520 access points and collected from 3 building of the Jaume I University, spanning a total area of $108,703$ m^2.

To model the presence of *outliers*, we modify the training data sets described above, producing ϵ-contaminated data sets $\mathcal{D}$ as per Definition 2. This is done by replacing the target variable y for a fraction ϵ of the data points $(x, y) \in \mathcal{D}$ with a uniformly random location y within the deployment area.

7.3.3 Implementation

We consider a model class specified by a Gaussian likelihood $p(y|x, \theta) = \mathcal{N}(y|f_\theta(x), 0.01)$, where the mean $f_\theta(x)$ is the output of a neural network with two hidden layers, each with

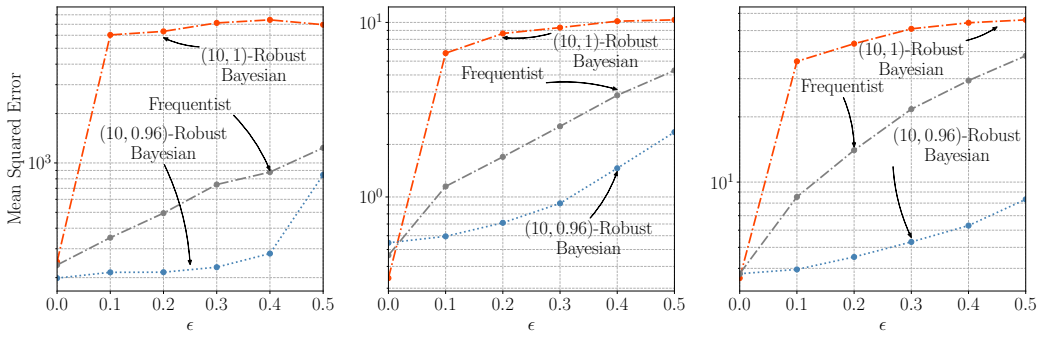
50 neurons with ELU activations. Despite the expressive power of the neural network model, each model $p(y|x, \theta)$ in this class can only account for unimodal, Gaussian distributed, residual uncertainties around the estimated position $f_\theta(x)$. Therefore, whenever the residual uncertainty about the receiver location is multimodal, the model class is *misspecified* by Definition 1. As we will see, given the complex relation between RSSI vector and location, particularly when the number of RSSI measurements is sufficiently small, residual uncertainty tends to be multimodal, making this an important problem. Training for frequentist and Bayesian learning is carried out as described in the previous section, and ensembling uses $m = 50$ samples during testing time.

7.3.4 Results

We start by considering the case in which there are no outliers, i.e., $\epsilon = 0$, thus focusing solely on the problem of misspecification. In Figure 7.3, we plot the predictive distribution obtained via Bayesian learning ($m = 1$, left panel) and robust Bayesian learning with $m = 10$ and $t = 1$ (right panel) for a testing sample x corresponding to the position shown as a green cross. The black dots correspond to the positions covered by the training set in the *SigfoxRural* data set. The resulting predictive distribution for conventional Bayesian learning provides a poor estimation of the true device position, and is unable to properly quantify uncertainty. In contrast, robust Bayesian learning is able to counteract model misspecification, producing a more informative predictive distribution. The distribution correctly suggests that the receiver can be in two possible areas, one of which indeed containing the true node location.

To further elaborate on the capacity of robust Bayesian learning for uncertainty quantification, in Table 7.1 we report the negative log-likelihood (7.6) attained by Bayesian learning ($m = 1$), as well as by robust Bayesian learning with $t = 1$ and $m = 2$ or $m = 10$ on the three data sets. Increasing the value of m is seen to yield lower negative log-likelihood scores, confirming that robust Bayesian learning provides a more precise quantification of uncertainty.

We now introduce outliers by carrying out training on contaminated data sets with different levels of contamination ϵ. Recall that the trained models are tested on a clean ($\epsilon = 0$) test data set $\mathcal{D}_{te}$. In Figure 7.4, we plot the test MSE (7.5) for frequentist and the (m, t)-robust Bayesian learning with $m = 10$ and $t \in \{1, 0.96\}$ as a function of ϵ. The MSE of frequentist learning and $(10, 1)$-robust Bayesian learning are seen to degrade significantly for increasing values of ϵ. The performance loss is particularly severe for $(m, 1)$-robust Bayesian learning. This is due to the mass-covering behavior entailed by the use of m-sample training loss, which in this case becomes detrimental due to the presence of outliers. In contrast, robust Bayesian learning with $t = 0.96$ is able to counteract the effect of outliers, retaining good predictive performance even in case of largely corrupted data sets.



(a) *SigfoxRural* (b) *UTSIndoor* (c) *UJIIndoor*

Figure 7.4: Test mean squared error (7.5) for frequentist and the (m, t)-robust Bayesian learning with $m = 10$ and $t = \{1, 0.96\}$ as a function of the corruption level ϵ for RSSI-based localization. As ϵ increases, the training data sets are increasingly affected by outliers.

7.4 Robust and Calibrated Channel Simulation

The design of communication systems has traditionally relied on analytical channel models obtained via measurements campaigns. Due to the complexity of multipath propagation scenarios, in recent years generative machine learning models have introduced as an alternative to analytical models. Generative models can be trained to produce samples that mimic hard-to-model channel conditions. Applications of deep generative models in the form of variational autoencoders (VAEs) [162] and generative adversarial networks (GANs) [203] were specifically reported in the context of end-to-end simulation of wireless systems in [204, 205] and for channel modeling in [206–209] for earlier applications to satellite communications.

The outlined prior work has focused on frequentist methods and has assumed the availability of clean data sets that are free from outliers. In this section, we explore the use of robust Bayesian learning to account for both outliers and model misspecification.

7.4.1 Problem Definition and Performance Metrics

Generative models are trained in an unsupervised manner by assuming the availability of a training set $\mathcal{D}$ of examples x corresponding to channel impulse responses. We focus on VAEs, i.e., on generative models with latent variables. VAEs comprise a parameterized *encoder* $q(h|x, \theta_e)$, mapping an input $x \in \mathcal{X}$ into a lower-dimensional latent vector $h \in \mathcal{H}$; as well as a parameterized *decoder* $p(x|h, \theta_d)$ that reconstructs the input sample $x \in \mathcal{X}$ from the latent representation $h \in \mathcal{H}$. Note that the vector of model parameters encompasses both encoding and decoding parameters as $\theta = (\theta_e, \theta_d)$.

Let us define as $p(h)$ a fixed *prior* distribution on the latent variables h. Once training is complete, samples x of channel responses can be generated from the model as follows. For frequentist learning, given the trained model θ^{freq}, one generates a sample $h \sim p(h)$

for the latent vector, and then produces a channel sample $x \sim p(x|h, \theta^{\text{freq}})$. For Bayesian learning, given the optimized distribution $q(\theta)$, we produce a random sample $\theta \sim q(\theta)$ and then generate channel sample $x \sim p(x|h, \theta_d)$. The role of the encoder $q(h|x, \theta_e)$ will be made clear in Section 7.4.3 when discussing the training method.

According to the discussion in the previous paragraph, the channel distribution implemented by the model is given by

$$p(x) = \mathbb{E}_{p(h)}[p(x|h, \theta_d^{\text{freq}})] \tag{7.7}$$

for frequentist learning; and by

$$p(x) = \mathbb{E}_{p(h)q(\theta_d)}[p(x|h, \theta_d)] \tag{7.8}$$

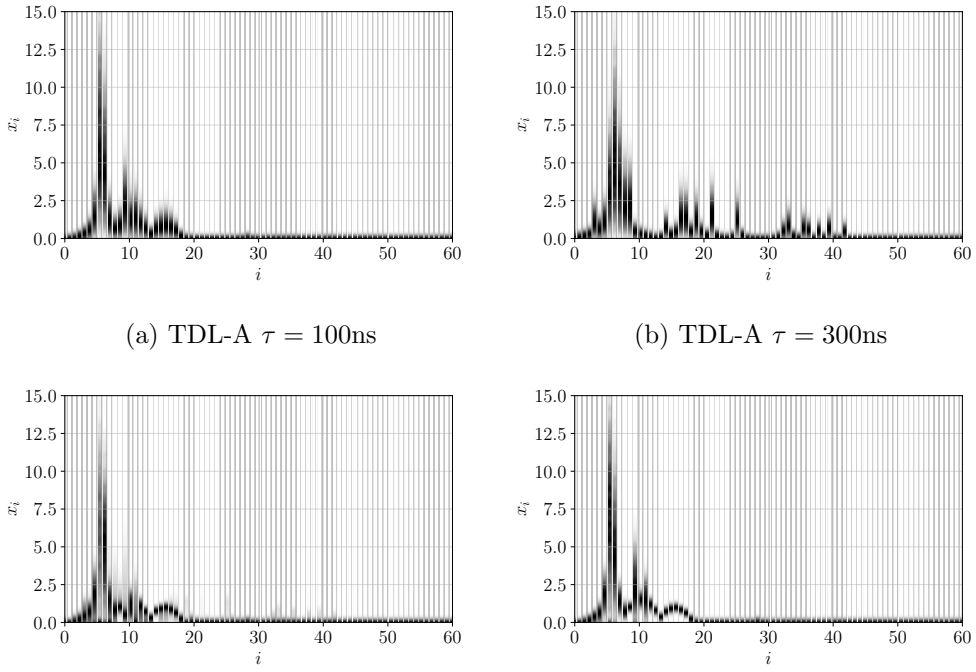
for Bayesian learning. Note that the average is taken only over the latent vector $h \sim p(h)$ for frequentist learning; while in Bayesian learning the expectation is also taken over the optimized distribution $q(\theta_d)$ for the decoder's parameters θ_d.

To evaluate the performance of the generative model, we consider two different metrics accounting for accuracy and uncertainty quantification. Accuracy is measured by the "distance" between the target distribution $\nu(x)$ and the distribution $p(x)$ produced by the model. We measure the "distance" between $\nu(x)$ and $p(x)$ via the *maximum-mean discrepancy* (MMD) [210], which is defined as

$$\begin{aligned}\text{MMD}(p, \nu) =& \mathbb{E}_{x,x' \sim p(x)}[k(x, x')] + \mathbb{E}_{x,x' \sim \nu(x)}[k(x, x')] \\ &- 2\mathbb{E}_{x \sim \nu(x), x' \sim p(x)}[k(x, x')]\end{aligned} \tag{7.9}$$

where $k(x, x')$ is a positive definite kernel function. In the experiments reported below, we have approximated the MMD based on empirical averages. These are evaluated using samples from distribution $p(x)$, which are generated as explained above, as well as samples from the sampling distribution $\nu(x)$, i.e., examples from the training set $\mathcal{D}$. Moreover, we use the Gaussian kernel $k(x, x') = \mathcal{N}(\|x - x'\||0, 1)$.

To evaluate the performance in terms of uncertainty quantification, we focus on the problem of *out-of-distribution (OOD) detection* (see, e.g., [211]). A well-calibrated model $p(x)$, when fed with an input x, should return a small value if x is an OOD sample, that is, if it has a low target distribution $\nu(x)$. To obtain a quantitative measure, we consider the task of distinguishing between samples drawn from the target distribution $\nu(x)$ and from the OOD distribution $\xi(x)$. Specifically, we adopt the model probability distribution $p(x)$ as the test statistic, classifying x as in-distribution (ID) if $p(x)$ is larger than some threshold γ and as OOD otherwise. As in ([212]), we take the area under the receiver operating characteristic curve (AUROC) score for this test as a measure of how distinguishable the two samples are. The AUROC metric is obtained by integrating the ROC traced by probability of detection versus probability of false alarm as the threshold γ is varied. A larger AUROC indicates that the model provides a better quantification of uncertainty, as reflected in its capacity to detect OOD samples against ID samples.



(a) TDL-A $\tau = 100$ns

(b) TDL-A $\tau = 300$ns

(c) Frequentist Learning

(d) $(4, 0.7)$-Robust Bayesian Learning

Figure 7.5: The top row shows a sample of the magnitude for the TDL-A channel response given a delay spread $\tau = 100$ns in panel (a), while an outlier sample corresponding to the larger delay spread $\tau = 300$ ns is depicted in panel (b). The bottom row reports a sample from the trained model for frequentist learning in panel (c) and for $(4, 0.7)$-robust Bayesian learning in panel (d).

7.4.2 Data Set

We consider the simulation of the magnitudes of a frequency-selective channel response $x \in \mathbb{R}^{128}$ that mimics the target distribution $\nu(x)$ defined by the 3GPP TDL-A channel model distribution [213] with a delay spread of $\tau = 100$ ns. *Outliers* are accounted for by constructing an ϵ-contaminated training set $\mathcal{D}$ that contains a fraction $\epsilon = 0.2$ of samples distributed according to the same channel model but with a larger delay spread $\tau = 300$ ns (see the top row in Fig. 7.5).

7.4.3 Implementation

For models with latent variables, the direct adoption of the log-loss generally yields intractable optimization problems (see, e.g., [186]). To address this problem, training of VAEs replaces the training loss (6.8) with the *variational lower bound*

$$\hat{\mathcal{L}}^{VAE}(\theta, \mathcal{D}) = \sum_{x \in \mathcal{D}} \mathbb{E}_{p(h|x,\theta_e)}[\log p(x|h, \theta_d)] = -\sum_{x \in \mathcal{D}} \mathrm{KL}(p(h|x, \theta_d)||p(h)), \tag{7.10}$$

which involves the use of the encoder model $p(h|x, \theta_e)$. Accordingly, the frequentist training objective is modified as

$$\underset{\theta}{\text{minimize}}\, \hat{\mathcal{L}}_{VAE}(\theta, \mathcal{D}), \tag{7.11}$$

while Bayesian learning addresses the problem

$$\underset{q(\theta)}{\text{minimize}}\, \mathbb{E}_{q(\theta)} \left[\hat{\mathcal{L}}_{VAE}(\theta_e, \theta_d, \mathcal{D})\right] + \frac{1}{\beta}\mathrm{KL}(q(\theta)||p(\theta)). \tag{7.12}$$

The robust free energy metrics are obtained in a similar manner, yielding the following formulation for (m, t)-robust Bayesian learning

$$\hat{\mathcal{L}}_t^{VAE}(\theta_1, \ldots, \theta_m, \mathcal{D}) = \sum_{x \in \mathcal{D}} \mathbb{E}_{p(h|x,\theta_e)} \log_t \left(\sum_{i=1}^{m} \frac{p(x|h, \theta_{d,i})}{m} \right) - \sum_{x \in \mathcal{D}} \mathrm{KL}(p(h|x, \theta_d)||p(h)). \tag{7.13}$$

The prior latent variable distribution is $p(h) = \mathcal{N}(h|0, \mathbb{I}_5)$. We implement both the encoder and the decoder by using fully connected neural networks with a single hidden layer with 10 units. Specifically, the encoder distribution $p(h|x, \theta_e) = \mathcal{N}(h|\mu_{\theta_e}(x), \Sigma_{\theta_e}(x))$ has mean vector $\mu_{\theta_e}(x) \in \mathbb{R}^5$ and diagonal covariance matrix $\Sigma_{\theta_e}(x) \in \mathbb{R}^{5\times5}$ obtained from the output of the neural network. The decoder $p(x|h, \theta_d) = \mathcal{N}(\hat{x}|\mu_{\theta_d}(h), \sigma\mathbb{I}_{128})$ has mean vector $\mu_{\theta_d}(h)$ obtained as the output of the neural network with a fixed variance value $\sigma = 0.1$. For Bayesian learning, we optimize distribution $q(\theta_d)$ as in the previous sections, while we consider a distribution $q(\theta_e)$ concentrated at a single vector θ_e. Ensembling during testing time is carried out with $m = 50$ samples.

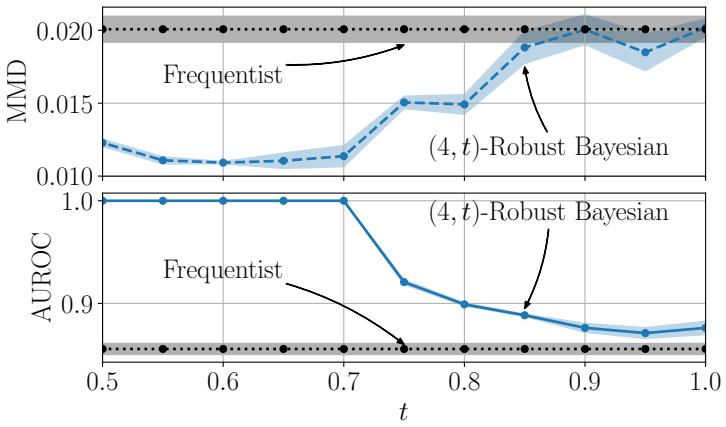


Figure 7.6: Maximum mean discrepancy (MMD) and area under receiving operating curve (AUROC) for frequentist learning and $(4, t)$-robust Bayesian learning. Both models are trained on a corrupted data set with ($\epsilon = 0.2$).

7.4.4 Results

To start, in Figure 7.5 we illustrate a sample of the magnitude for the TDL-A channel response given a delay spread $\tau = 100$ ns in panel (a), while an outlier sample corresponding to the larger delay spread $\tau = 300$ ns is depicted in panel (b). The bottom row of Figure 7.5 reports a sample from the trained model for frequentist learning in panel (c) and for $(4, 0.7)$-robust Bayesian learning in panel (d). Visual inspection of the last two panels confirms that (m, t)-robust Bayesian learning can mitigate the effect of outliers as it reduces the spurious multipath components associated with larger delays.

For a numerical comparison, Figure 7.6 compares frequentist and $(4, t)$-robust Bayesian learning in terms of both accuracy – as measured by the MMD – and uncertainty quantification – as evaluated via the AUROC. For $t < 0.85$ robust Bayesian learning is confirmed to have the capacity to mitigate the effect of the outlying component, almost halving the MMD obtained by frequentist learning. Furthermore, robust Bayesian learning has a superior uncertainty quantification performance, with gain increasing for decreasing values of t.

7.5 Conclusion

This chapter has focused on the problem of ensuring that AI models trained for wireless communications satisfy reliability and robustness requirements. We have specifically addressed two important problems: model misspecification, arising from limitations on the available knowledge about the problem and on the complexity of the AI models that can be implemented on network devices; and outliers, which cause a mismatch between training and testing conditions. We have argued that standard frequentist

learning, as well as Bayesian learning, are not designed to address these requirements, and we have explored the application of *robust Bayesian learning* to achieve robustness to model misspecification and to the presence of outliers in the training data set. Robust Bayesian learning has been shown to consistently provide better accuracy and uncertainty estimation capabilities in a range of important wireless communication problems. These results motivate a range of extension of robust Bayesian learning and applications. For instance, the integration of robust Bayesian learning to the meta-learning framework, in order to enable robust and sample effective learning, or the application of robust Bayesian learning to higher layers of the protocol stack as a tool to empower semantic communication.

Chapter 8

Conclusion

The content of this thesis illustrates how vast and heterogeneous the range of problems arising from the application of machine learning in wireless communication networks is. For some of these challenges, we provided solutions that we hope will contribute to the adoption of reliable machine learning solutions in 6G networks. In the following, we summarize the contributions and potential directions for future work.

In Part I of this manuscript, we focused on decentralized training of machine learning models over device-to-device 6G networks. In Chapter 2, we proved that wireless networks, even when characterized by straggling nodes and unreliable communication links, are a suitable infrastructure for the training of machine learning models in a decentralized manner. In particular, we showed how asynchronous updates can greatly reduce the convergence time of optimization procedures without hampering the quality of the final model. As shown by our analysis, the training procedure converges despite the dissemination of outdated updates and sparse communication between workers. This achievability result provides us with the flexibility of designing energy-efficiency optimization procedures in which devices communicate only in opportune slots; for example, when the wireless channel is in favourable conditions or when the model updates are relevant. Studying the trade-off between energy-efficiency performance indicators and the convergence properties of decentralized optimization represents an interesting research direction. In Chapter 3, we have investigated the potential role of UAVs in decentralized learning procedures. While the literature on UAV-aided communication is vast, the applications of UAVs in the context of edge learning are mostly unexplored. Our results are presented for a single drone and its optimized trajectory is obtained accordingly. Analyses of multi-drone scenarios and the derivations of a jointly optimized trajectory are natural extensions of the results presented in this chapter.

Part II has been devoted to addressing one of the fundamental limitations of collaborative learning procedures: data heterogeneity. We provided two possible algorithms to mitigate the detrimental effects due to the aggregation of heterogeneous data. In Chapter 4 we formulated the learning problem as a distributionally robust optimization problem and provided a communication-efficient algorithm to solve it. The outcome of this procedure is a single machine learning model that is fair, i.e. it has satisfactory performance on all devices. The concept of fairness is fundamental in wireless commu-

nication protocols, therefore we expect that the tools derived in this chapter can find application in many networking problems. In Chapter 5 we tackled data heterogeneity by proposing a training procedure that outputs personalized models to serve groups of users with different needs. The main underpinning of the algorithm is the estimation of the similarity between users' learning tasks. Evaluating similarity scores between users poses a threat to the privacy guarantees of federated algorithms, therefore investigating the trade-off between personalization and privacy may shed light on the fundamental limitations of this approach.

In Part III, motivated by the necessity of quantifying uncertainty in wireless communication learning problems, we proposed the (m, t)-robust Bayesian learning framework, a Bayesian learning procedure capable of addressing both model misspecification and the presence of outliers. The proposed methodology produced well-calibrated and robust predictive posteriors over a range of wireless communication problems. In Chapter 7, we showed that (m, t)-robust Bayesian learning greatly outperforms the frequentist and standard Bayesian learning approaches. Despite the superior uncertainty quantification capabilities, the (m, t)-robust Bayesian learning relies on ensembling, which comes with a potentially large computational cost. This is due to the necessity of sampling and aggregating the output of multiple components to perform inference. Therefore, it become essential to validate the merits of the (m, t)-robust Bayesian learning when applied to more computationally efficient ensembling approaches. Furthermore, (m, t)-robust Bayesian learning can be directly applied to reinforcement learning, as well as to meta-learning, for which Bayesian methods have recently been investigated.

Appendices

Appendix A

Appendix of Chapter 2

A.1 Proof of Lemma 1

Define the event $E^{(t)} := \{\mathcal{G}^{(t)} \text{ is connected}\}$ and its complementary event $\bar{E}^{(t)}$. Whenever the Metropolis-Hasting weights are obtained from a connected graph, the resulting mixing matrix $W^{(t)}$ has a consensus rate greater than zero. Therefore, there exists $\delta > 0$ such that

$$\mathbb{E}_{W^{(t)}|E^{(t)}} \left\| W^{(t)} X - \bar{X} \right\|_F^2 \leq (1-\delta) \left\| W^{(t)} X - \bar{X} \right\|_F^2$$

It follows that, for any $X \in \mathbb{R}^{d \times m}$

$$\begin{aligned}
\mathbb{E}_{W^{(t)}} \left\| W^{(t)} X - \bar{X} \right\|_F^2 =& q \mathbb{E}_{W^{(t)}|E^{(t)}} \left\| W^{(t)} X - \bar{X} \right\|_F^2 \\
&+ (1-q) \mathbb{E}_{W^{(t)}|\bar{E}^{(t)}} \left\| X - \bar{X} \right\|_F^2 \\
\leq& q(1-\delta) \left\| W^{(t)} X - \bar{X} \right\|_F^2 \\
&+ (1-q) \left\| X - \bar{X} \right\|_F^2
\end{aligned}$$

where we have lower bounded the consensus rate by zero in case of disconnected topologies. Grouping terms and having assumed $q > 0$, we obtain that the expected consensus is satisfied with rate $(1 - q\delta) > 0$.

A.2 Proof of Lemma 2

Similarly to [63, 99] we establish the following recursive inequality

$$\begin{aligned}
\sum_{i=1}^{m} \mathbb{E} \left\| \theta^{(t)} - \bar{\theta}^{(t)} \right\|^2 \leq& \left(1 - \frac{p\zeta}{2} \right) \sum_{i=1}^{m} \mathbb{E} \left\| \theta^{(t-1)} - \bar{\theta}^{(t-1)} \right\|^2 \\
&+ \frac{\eta^2}{p\zeta} \left(6mG^2 \right) + \zeta^2 \sum_{i=1}^{m} \mathbb{E} \left\| \tilde{n}_i^{(t)} \right\|^2.
\end{aligned}$$

Defining $\sigma_{w,i}^2 = \max_{t=0}^{T} \mathbb{E}\left\|\tilde{n}_i^{(t)}\right\|^2$ and then solving the recursion we obtain the final expression.

A.3 Proof of Theorem 1

We denote stale gradients by $g_i(\tilde{\theta}_i^{(t)}) = g_i(\theta_i^{(t-\tau_i)})$. According to the update rule, at each iteration $t+1$, we have

$$\mathbb{E}[f(\bar{\theta}^{t+1})] = \mathbb{E}\left[f\left(\bar{\theta}^t - \frac{1}{m}\sum_{i=1}^{m}\left(\tilde{\eta}_i^{(t)} g_i(\tilde{\theta}_i^{(t)}) + \zeta \tilde{n}_i^{(t)}\right)\right)\right]$$

where the expectation is w.r.t. the stochastic gradients, the communication noise $\Xi^{(t)}$, and the computation and communication failures at iteration $t+1$. For an L-smooth objective function, we have

$$\mathbb{E}[f(\bar{\theta}^{(t+1)})] \leq f(\bar{\theta}^{(t)}) \underbrace{-\frac{1}{m}\sum_{i=1}^{m}\left\langle \nabla f(\bar{\theta}^{(t)}), \mathbb{E}[\tilde{\eta}_i^{(t)} g_i(\tilde{\theta}_i^{(t)})]\right\rangle}_{:=T_1}$$
$$+ \underbrace{\frac{L}{2m^2}\mathbb{E}\left\|\sum_{i=1}^{m}\tilde{\eta}_i^{(t)} g_i(\tilde{\theta}_i^{(t)}))\right\|^2}_{:=T_2} + \frac{L}{2m^2}\zeta^2 \sum_{i=1}^{m}\mathbb{E}\left\|\tilde{n}_i^{(t)}\right\|^2$$

where we used the fact that the communication noise has zero mean and is independent across users.

Adding and subtracting $\nabla f_i(\bar{\theta}^{(t)})$ to each summand of T_1 and since $\mathbb{E}[\tilde{\eta}_i^{(t)} g_i(\tilde{\theta}_i^{(t)})] = \eta \nabla f_i(\tilde{\theta}_i^{(t)})$, with $\eta = \min_j(1-\rho_j)/(\sqrt{4LT})$, we obtain

$$\begin{aligned}
T_1 =& -\eta\left\langle \nabla f(\bar{\theta}^{(t)}), \frac{1}{m}\sum_{i=1}^{m}\nabla f_i(\tilde{\theta}_i^{(t)})\right\rangle \\
=& \frac{\eta}{2}\left\|\nabla f(\bar{\theta}^{(t)}) - \frac{1}{m}\sum_{i=1}^{m}\nabla f_i(\tilde{\theta}_i^{(t)})\right\|^2 \\
&- \frac{\eta}{2}\left\|\nabla f(\bar{\theta}^{(t)})\right\|^2 - \frac{\eta}{2m^2}\left\|\sum_{i=1}^{m}\nabla f_i(\tilde{\theta}_i^{(t)})\right\|^2 \\
\leq& \frac{\eta\gamma}{2}\left\|\nabla f(\bar{\theta}^{(t)})\right\|^2 + \frac{\eta L^2}{2m}\sum_{i=1}^{m}\left\|\theta_i^{(t)} - \bar{\theta}^{(t)}\right\|^2 \\
&- \frac{\eta}{2}\left\|\nabla f(\bar{\theta}^{(t)})\right\|^2 - \frac{\eta}{2m^2}\left\|\sum_{i=1}^{m}\nabla f_i(\tilde{\theta}_i^{(t)})\right\|^2
\end{aligned}$$

where we have used the staleness assumption. The last term can be bounded using the property of the stochastic gradient and the fact that $\tilde{\eta}_i^{(t)} \leq 1/(\sqrt{4LT}) \leq 1/(\sqrt{4L})$ as

$$\begin{aligned}
T_2 \leq & \frac{L}{2m^2}\mathbb{E}\left\|\sum_{i=1}^{m}\tilde{\eta}_i^{(t)}[g_i(\tilde{\theta}_i^{(t)}) - \nabla f_i(\tilde{\theta}_i^{(t)})]\right\|^2 \\
& + \frac{L}{2m^2}\mathbb{E}\left\|\sum_{i=1}^{m}\tilde{\eta}_i^{(t)}\nabla f_i(\tilde{\theta}_i^{(t)})\right\|^2 \\
\leq & \frac{\sigma^2}{8mT} + \frac{\eta}{8m^2}\mathbb{E}\left\|\sum_{i=1}^{m}\nabla f_i(\tilde{\theta}_i^{(t)})\right\|^2.
\end{aligned}$$

Summing T_1 and T_2 we obtain

$$\begin{aligned}
T_1 + T_2 \leq & -\frac{\eta}{2}(1-\gamma)\left\|\nabla f(\bar{\theta}^{(t)})\right\|^2 + \frac{\sigma^2}{8mT} \\
& + \frac{\eta L^2}{2m}\sum_{i=1}^{m}\left\|\theta_i^{(t)} - \bar{\theta}^{(t)}\right\|^2 \\
& - \frac{\eta}{4m^2}\left\|\sum_{i=1}^{m}\nabla f_i(\tilde{\theta}_i^{(t)})\right\|^2.
\end{aligned}$$

Defining $\gamma' = (1-\gamma)$, telescoping and taking expectations we obtain

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\left\|\nabla f(\bar{\theta}^{(t)})\right\|^2 \leq & 2\frac{f(\bar{\theta}^0) - f(\bar{\theta}^T)}{\eta T\gamma'} + \frac{\sigma^2}{4\eta\gamma' mT} \\
& + \frac{1}{T}\sum_{t=1}^{T}\frac{L^2}{m\gamma'}\sum_{i=1}^{m}\mathbb{E}\left\|\theta_i^{(t)} - \bar{\theta}^{(t)}\right\|^2 \\
& + \frac{1}{T}\sum_{t=1}^{T}\frac{L\zeta^2}{\eta m^2\gamma'}\sum_{i=1}^{m}\mathbb{E}\left\|\tilde{n}_i^{(t)}\right\|^2.
\end{aligned}$$

Defining $\sigma_{w,i}^2 = \max_{t=0}^{T}\mathbb{E}\left\|\tilde{n}_i^{(t)}\right\|^2$ and bounding the consensus term by Lemma 2, we obtain

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\left\|\nabla f(\bar{\theta}^{(t)})\right\|^2 \leq & 2\frac{f(\bar{\theta}^0) - f(\bar{\theta}^T)}{\eta T\gamma'} \\
& + \frac{L^2}{m\gamma'}\left(\eta^2\frac{12mG^2}{(p\zeta)^2} + \zeta\frac{2}{p}\sum_{i=1}^{m}\sigma_{w,i}^2\right) \\
& + \frac{\sigma^2}{4\eta\gamma' mT} + \frac{L\zeta^2}{\eta m^2\gamma'}\sum_{i=1}^{m}\sigma_{w,i}^2.
\end{aligned}$$

The final result is obtained setting $\eta = \frac{1}{\sqrt{4LT}}$ and $\zeta = \frac{1}{T^{3/8}}$.

Appendix B

Appendix of Chapter 4

B.1 Useful inequalities

This section contains a collection of ancillary results that are useful for the subsequent proofs.

Proposition 1. *A differentiable and L-smooth function $f(x)$ satisfies*

$$\big\|\nabla f(x) - \nabla f(x')\big\| \leq L\big\|x - x'\big\|. \tag{B.1}$$

Furthermore, if $f(x)$ is convex

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|y - x\|^2 \tag{B.2}$$

and if x^* is a minimizer

$$\frac{1}{2L}\|\nabla f(x)\|^2 \leq f(x) - f(x^*). \tag{B.3}$$

Otherwise, if $f(x)$ concave

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle - \frac{L}{2}\|y - x\|^2 \tag{B.4}$$

and if x^* is a maximizer

$$\frac{1}{2L}\|\nabla f(x)\|^2 \leq f(x^*) - f(x). \tag{B.5}$$

Proposition 2. *A differentiable and μ-strongly convex function $f(x)$ satisfies*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\mu}{2}\|y - x\|^2 \tag{B.6}$$

and a differentiable and μ-strongly concave function $g(x)$ satisfies

$$g(y) \leq g(x) + \langle \nabla g(x), y - x\rangle - \frac{\mu}{2}\|y - x\|^2. \tag{B.7}$$

Proposition 3. *Given two vectors $a, b \in \mathbb{R}^d$, for $\beta > 0$ we have*

$$2\langle a, b\rangle \leq \beta^{-1}\|a\|^2 + \beta\|b\|^2 \tag{B.8}$$

and

$$\|a+b\| \leq (1+\beta^{-1})\|a\|^2 + (1+\beta)\|b\|^2. \tag{B.9}$$

Proposition 4. *Given two matrices $A \in \mathbb{R}^{p\times q}, B \in \mathbb{R}^{q\times r}$, we have*

$$\|AB\|_F \leq \|A\|_F \|B\|_2 \tag{B.10}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

Proposition 5. *Given a set of vectors $\{a_i\}_{i=1}^n$ we have*

$$\left\|\sum_{i=1}^n a_i\right\|^2 \leq n\sum_{i=1}^n \|a_i\|^2. \tag{B.11}$$

Consensus inequalities

To streamline the notation we define $\tilde{\nabla} g_i(\theta_i^t, \lambda_i^t) = \nabla g_i(\theta_i^t, \lambda_i^t, \xi_i^t)$ and introduce the following matrices

$$\Theta^t = \left[\theta_1^t, \ldots, \theta_m^t\right] \in \mathbb{R}^{d\times m}, \tag{B.12}$$

$$\hat{\Theta}^t = \left[\hat{\theta}_1^t, \ldots, \hat{\theta}_m^t\right] \in \mathbb{R}^{d\times m}, \tag{B.13}$$

$$\Lambda^t = \left[\lambda_1^t, \ldots, \lambda_m^t\right] \in \mathbb{R}^{m\times m}, \tag{B.14}$$

$$\tilde{\nabla}_\theta G(\Theta^t, \Lambda^t) = \left[\tilde{\nabla}_\theta g_1(\theta_1^t, \lambda_1^t), \ldots, \tilde{\nabla}_\theta g_m(\theta_m^t, \lambda_m^t)\right] \in \mathbb{R}^{d\times m} \tag{B.15}$$

$$\tilde{\nabla}_\lambda G(\Theta^t, \Lambda^t) = \left[\tilde{\nabla}_\lambda g_1(\theta_1^t, \lambda_1^t), \ldots, \tilde{\nabla}_\lambda g_m(\theta_m^t, \lambda_m^t)\right] \in \mathbb{R}^{m\times m} \tag{B.16}$$

and for a matrix X we define $\bar{X} = X\frac{\mathbf{1}\mathbf{1}^T}{m}$.

The local update rule of Algorithm 2 can be rewritten as

$$\Theta^{t+\frac{1}{2}} = \Theta^t - \eta_\theta \tilde{\nabla}_\theta G(\Theta^t, \Lambda^t) \tag{B.17}$$

$$\Lambda^{t+\frac{1}{2}} = \mathcal{P}_\Lambda\left(\Lambda^t + \eta_\lambda \tilde{\nabla}_\lambda G(\Theta^t, \Lambda^t)\right) \tag{B.18}$$

where $\mathcal{P}_\Lambda$ is applied column-wise. The compressed gossip algorithm CHOCO-GOSSIP [99] used to share model parameters preserves averages and satisfies the following recursive inequality with $c = \frac{\rho^2\delta}{82}$

$$\mathbb{E}\left[\left\|\Theta^{t+1} - \bar{\Theta}^{t+1}\right\|_F^2 + \left\|\Theta^{t+1} - \hat{\Theta}^{t+1}\right\|_F^2\right] \leq (1-c)\,\mathbb{E}\left[\left\|\Theta^{t+\frac{1}{2}} - \bar{\Theta}^{t+\frac{1}{2}}\right\|_F^2 + \left\|\Theta^{t+\frac{1}{2}} - \hat{\Theta}^t\right\|_F^2\right]. \tag{B.19}$$

The uncompressed gossip scheme used to communicate Λ satisfies

$$\mathbb{E}\left[\left\|\Lambda^{t+1} - \bar{\Lambda}^{t+1}\right\|_F^2\right] \leq (1-\rho)\,\mathbb{E}\left[\left\|\Lambda^{t+\frac{1}{2}} - \bar{\Lambda}^{t+\frac{1}{2}}\right\|_F^2\right]. \tag{B.20}$$

Lemma 4. *(Consensus inequality for compressed communication [112]) For a fixed $\eta_\theta > 0$ and $\gamma = \frac{\rho^2\delta}{16\rho+\rho^2+4\beta^2+2\rho\beta^2-8\rho\delta}$ the iterates of Algorithm 2 satisfy*

$$\mathbb{E}\left[\Xi_\theta^t\right] = \mathbb{E}\left[\sum_{i=1}^m \left\|\theta_i^t - \bar{\theta}^t\right\|^2\right] \leq 12\eta_\theta^2 \frac{mG_\theta^2}{c^2} \tag{B.21}$$

Lemma 5. *(Consensus Inequality for uncompressed communication [99]) For a fixed $\eta_\lambda > 0$ the iterates of Algorithm 2 satisfy*

$$\mathbb{E}\left[\Xi_\lambda^t\right] = \mathbb{E}\left[\sum_{i=1}^m \left\|\lambda_i^t - \bar{\lambda}^t\right\|^2\right] \leq 4\eta_\lambda^2 \frac{mG_\lambda^2}{\rho^2} \tag{B.22}$$

B.2 Proof of Theorem 2: Convex case

Define

$$\Phi(\cdot) = \max_{\lambda\in\Delta^{m-1}} g(\cdot,\lambda);$$

under assumptions 5, 6 and if the local objective functions $\{f_i(\theta)\}_{i=1}^m$ are convex, Theorem 2 guarantees that the output solution (θ_o, λ_o) satisfies

$$\begin{aligned}\mathbb{E}\left[\Phi(\theta_o) - \min_{\theta\in\Theta}\Phi(\theta)\right] \leq& \frac{4}{T}\left(\frac{LG_\lambda^2}{\rho^2} + 3\frac{LG_\theta^2}{c^2}\right) + \frac{1}{\sqrt{T}}\left(\sqrt{12}\frac{D_\lambda LG_\theta}{c} + 2\frac{D_\theta LG_\lambda}{\rho}\right)\\ &+ \frac{1}{\sqrt{T}}\left(\frac{D_\theta + D_\lambda}{2} + \frac{G_\theta^2 + G_\lambda^2}{2}\right).\end{aligned} \tag{B.23}$$

The proof starts from the following decomposition of the sub-optimality gap

$$\mathbb{E}\left[\max_\lambda g(\theta_o,\lambda) - \min_\theta \max_\lambda g(\theta,\lambda)\right] \leq \mathbb{E}\left[\max_\lambda g(\theta_o,\lambda) - \max_\lambda \min_\theta g(\theta,\lambda))\right] \tag{B.24}$$

$$\leq \mathbb{E}\left[\max_\lambda g(\theta_o,\lambda) - \min_\theta g(\theta,\lambda_o))\right] \tag{B.25}$$

$$\leq \mathbb{E}\left[\max_{\lambda,\theta} g(\theta_o,\lambda) - g(\theta,\lambda_o))\right] \tag{B.26}$$

$$\leq \mathbb{E}\left[\max_{\lambda,\theta} \frac{1}{T}\sum_{t=0}^{T-1} g(\bar{\theta}^t,\lambda) - g(\theta,\bar{\lambda}^t))\right] \tag{B.27}$$

$$\begin{aligned}\leq& \mathbb{E}\left[\max_\lambda \frac{1}{T}\sum_{t=0}^{T-1} g(\bar{\theta}^t,\lambda) - g(\bar{\theta}^t,\bar{\lambda}^t)\right]\\ &+ \mathbb{E}\left[\max_\theta \frac{1}{T}\sum_{t=0}^{T-1} g(\bar{\theta}^t,\bar{\lambda}^t) - g(\theta,\bar{\lambda}^t)\right].\end{aligned} \tag{B.28}$$

Thanks to Lemmas (6) and (7) proved below, the two summands can be bounded to obtain

$$\mathbb{E}\left[\Phi(\theta_o) - \min_{\theta\in\Theta}\Phi(\theta)\right] \leq \frac{D_\theta}{2\eta_\theta T} + \frac{\eta_\theta}{2}\left(G_\theta^2 + \sqrt{48}\frac{D_\lambda LG_\theta}{c}\right) + 12\eta_\theta^2\frac{LG_\theta^2}{c^2}$$

$$+\frac{D_\lambda}{2\eta_\lambda T}+\frac{\eta_\lambda}{2}\left(G_\lambda^2+4\frac{D_\theta L G_\lambda}{\delta}\right)+4\eta_\lambda^2\frac{LG_\lambda^2}{\rho^2}. \tag{B.29}$$

Setting $\eta_\lambda=\eta_\theta=\frac{1}{\sqrt{T}}$, the final result is obtained. □

Lemma 6. *For $T>0$ and any θ, the sequence $\{\bar\theta^t,\bar\lambda^t\}_{t=0}^T$ generated by Algorithm 2 satisfies*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}g(\bar\theta^t,\bar\lambda^t)-g(\theta,\bar\lambda^t)\right]\leq\frac{D_\theta}{2\eta_\theta T}+\frac{\eta_\theta}{2}G_\theta^2+12\eta_\theta^2\frac{LG_\theta^2}{c^2}+2\eta_\lambda\frac{D_\theta L G_\lambda}{\rho} \tag{B.30}$$

where $D_\theta=\max_{t=0\dots,T}\mathbb{E}\left\|\bar\theta^t-\theta\right\|$.

Proof: From the update rule of the primal variable and the assumptions 6 on the stochastic gradient we have, that for any θ

$$\begin{aligned}
\mathbb{E}_{\xi^t}\left\|\bar\theta^{t+1}-\theta\right\|^2=&\mathbb{E}_{\xi^t}\left\|\bar\theta^t-\theta-\frac{\eta_\theta}{m}\sum_{i=1}^m\tilde\nabla_\theta g_i(\theta_i^t,\lambda_i^t)\right\|^2 && \text{(B.31)}\\
=&\left\|\bar\theta^t-\theta\right\|^2-2\frac{\eta_\theta}{m}\sum_{i=1}^m\langle\bar\theta^t-\theta;\mathbb{E}_{\xi^t}\left[\tilde\nabla_\theta g_i(\theta_i^t,\lambda_i^t)\right]\rangle\\
&+\mathbb{E}_{\xi^t}\left\|\frac{\eta_\theta}{m}\sum_{i=1}^m\tilde\nabla_\theta g_i(\theta_i^t,\lambda_i^t)\right\|^2 && \text{(B.32)}\\
\leq&\left\|\bar\theta^t-\theta\right\|^2\underbrace{-2\frac{\eta_\theta}{m}\sum_{i=1}^m\langle\bar\theta^t-\theta;\nabla_\theta g_i(\theta_i^t,\lambda_i^t)\rangle}_{:=T_2}+\eta_\theta^2G_\theta^2. && \text{(B.33)}
\end{aligned}$$

Denoting with $D_\theta^t=\left\|\bar\theta^t-\theta\right\|$ we have that for T_2 the following holds

$$\begin{aligned}
T_2=&-2\frac{\eta_\theta}{m}\left(\sum_{i=1}^m\langle\bar\theta^t-\theta;\nabla_\theta g_i(\theta_i^t,\bar\lambda^t)\rangle+\sum_{i=1}^m\langle\bar\theta^t-\theta;\nabla_\theta g_i(\theta_i^t,\lambda_i^t)-\nabla_\theta g_i(\theta_i^t,\bar\lambda^t)\rangle\right) && \text{(B.34)}\\
\leq&-2\frac{\eta_\theta}{m}\sum_{i=1}^m\langle\bar\theta^t-\theta;\nabla_\theta g_i(\theta_i^t,\bar\lambda^t)\rangle+2\eta_\theta LD_\theta^t\sqrt{\frac{\Xi_\lambda^t}{m}} && \text{(B.35)}\\
\leq&-2\frac{\eta_\theta}{m}\sum_{i=1}^m\left(\langle\bar\theta^t-\theta_i^t;\nabla_\theta g_i(\theta_i^t,\bar\lambda^t)\rangle+\langle\theta_i^t-\theta;\nabla_\theta g_i(\theta_i^t,\bar\lambda^t)\rangle\right)+2\eta_\theta LD_\theta^t\sqrt{\frac{\Xi_\lambda^t}{m}} && \text{(B.36)}\\
\overset{(B.4)}{\leq}&-2\frac{\eta_\theta}{m}\sum_{i=1}^m\left(g_i(\bar\theta^t,\bar\lambda^t)-g_i(\theta,\bar\lambda^t)-\frac{L}{2}\left\|\bar\theta^t-\theta_i^t\right\|^2\right)+2\eta_\theta LD_\theta^t\sqrt{\frac{\Xi_\lambda^t}{m}} && \text{(B.37)}\\
=&-2\frac{\eta_\theta}{m}\sum_{i=1}^m\left(g_i(\bar\theta^t,\bar\lambda^t)-g_i(\theta,\bar\lambda^t)\right)+\frac{2\eta_\theta L}{m}\Xi_\theta^t+2\eta_\theta LD_\theta^t\sqrt{\frac{\Xi_\lambda^t}{m}}. && \text{(B.38)}
\end{aligned}$$

Plugging it back in (B.33), rearranging the terms and taking the expectation over the previous iterate we get

$$\mathbb{E}\left[g(\bar{\theta}^t, \bar{\lambda}^t) - g(\theta, \bar{\lambda}^t)\right] = \frac{1}{m}\mathbb{E}\left[\sum_{i=1}^{m} g_i(\bar{\theta}^t, \bar{\lambda}^t) - g_i(\theta, \bar{\lambda}^t)\right] \tag{B.39}$$

$$\leq \frac{\mathbb{E}\|\bar{\theta}^t - \theta\|^2 - \mathbb{E}\|\bar{\theta}^{t+1} - \theta\|^2}{2\eta_\theta} + \frac{\eta_\theta}{2}G_\theta^2 + \frac{L}{m}\mathbb{E}\left[\Xi_\theta^t\right] + L\mathbb{E}\left[D_\theta^t\right]\sqrt{\frac{\mathbb{E}\left[\Xi_\lambda^t\right]}{m}}. \tag{B.40}$$

Telescoping from $t = 0$ to $t = T - 1$ and plugging the consensus inequalities (B.21) and (B.22), we get

$$\frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1} g(\bar{\theta}^t, \bar{\lambda}^t) - g(\theta, \bar{\lambda}^t)\right] \leq \frac{D_\theta}{2\eta_\theta T} + \frac{\eta_\theta}{2}G_\theta^2 + 12\eta_\theta^2\frac{LG_\theta^2}{c^2} + 2\eta_\lambda\frac{D_\theta L G_\lambda}{\rho} \tag{B.41}$$

where $D_\theta = \max_{t=0\ldots,T}\mathbb{E}[D_\theta^t] = \max_{t=0\ldots,T}\mathbb{E}\left\|\bar{\theta}^t - \theta\right\|$. □

Lemma 7. *For $T > 0$ and any λ, the sequence $\{\bar{\theta}^t, \bar{\lambda}^t\}_{t=0}^T$ generated by Algorithm 2 satisfies*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} g(\bar{\theta}^t, \lambda) - g(\bar{\theta}^t, \bar{\lambda}^t)\right] \leq \frac{D_\lambda}{2\eta_\lambda T} + \frac{\eta_\lambda}{2}G_\lambda^2 + 4\eta_\lambda^2\frac{LG_\lambda^2}{\rho^2} + \sqrt{1}2\eta_\theta\frac{D_\lambda L G_\theta}{c} \tag{B.42}$$

where $D_\lambda = \max_{t=0\ldots,T}\mathbb{E}\left\|\bar{\lambda}^t - \lambda\right\|$.

Proof The proof follows similarly as in Lemma (6)

$$\mathbb{E}_{\xi^t}\left\|\bar{\lambda}^{t+1} - \lambda\right\|^2 = \mathbb{E}_{\xi^t}\left\|\lambda - \bar{\lambda}^t + \frac{\eta_\lambda}{m}\sum_{i=1}^{m}\tilde{\nabla}_\lambda g_i(\theta_i^t, \lambda_i^t)\right\|^2 \tag{B.43}$$

$$= \left\|\bar{\lambda}^t - \lambda\right\|^2 - 2\frac{\eta_\lambda}{m}\sum_{i=1}^{m}\langle\lambda - \bar{\lambda}^t; \mathbb{E}_{\xi^t}\left[\tilde{\nabla}_\lambda g_i(\theta_i^t, \lambda_i^t)\right]\rangle + \mathbb{E}_{\xi^t}\left\|\frac{\eta_\lambda}{m}\sum_{i=1}^{m}\tilde{\nabla}_\lambda g_i(\theta_i^t, \lambda_i^t)\right\|^2 \tag{B.44}$$

$$= \mathbb{E}\left\|\bar{\lambda}^t - \lambda\right\|^2 \underbrace{-2\frac{\eta_\lambda}{m}\sum_{i=1}^{m}\langle\lambda - \bar{\lambda}^t; \nabla_\lambda g_i(\theta_i^t, \lambda_i^t)\rangle}_{:=T_3} + \eta_\lambda^2 G_\lambda^2. \tag{B.45}$$

Denoting with $D_\lambda^t = \left\|\bar{\lambda}^t - \lambda\right\|$ we have that for T_3 the following holds

$$T_3 = -2\frac{\eta_\lambda}{m}\left(\sum_{i=1}^{m}\langle\lambda - \bar{\lambda}^t; \nabla_\lambda g_i(\bar{\theta}^t, \lambda_i^t)\rangle + \sum_{i=1}^{m}\langle\lambda - \bar{\lambda}^t; \nabla_\lambda g_i(\theta_i^t, \lambda_i^t) - \nabla_\lambda g_i(\bar{\theta}^t, \lambda_i^t)\rangle\right) \tag{B.46}$$

$$\leq -2\frac{\eta_\lambda}{m}\sum_{i=1}^{m}\left(\langle \lambda - \bar{\lambda}^t; \nabla_\lambda g_i(\bar{\theta}^t, \lambda_i^t)\rangle\right) + 2\eta_\lambda L D_\lambda^t \sqrt{\frac{\Xi_\theta^t}{m}} \tag{B.47}$$

$$\leq -2\frac{\eta_\lambda}{m}\sum_{i=1}^{m}\left(\langle \lambda - \lambda_i^t; \nabla_\lambda g_i(\bar{\theta}^t, \lambda_i^t)\rangle + \langle \lambda_i^t - \bar{\lambda}^t; \nabla_\lambda g_i(\bar{\theta}^t, \lambda_i^t)\rangle\right) + 2\eta_\lambda L D_\lambda^t \sqrt{\frac{\Xi_\theta^t}{m}} \tag{B.48}$$

$$\overset{(B.4)}{\leq} -2\frac{\eta_\lambda}{m}\sum_{i=1}^{m}\left(g_i(\bar{\theta}^t, \lambda) - g_i(\bar{\theta}^t, \bar{\lambda}^t) - \frac{L}{2}\left\|\bar{\lambda}^t - \lambda_i^t\right\|^2\right) + 2\eta_\lambda L D_\lambda^t \sqrt{\frac{\Xi_\theta^t}{m}} \tag{B.49}$$

$$= -2\frac{\eta_\lambda}{m}\sum_{i=1}^{m}\left(g_i(\bar{\theta}^t, \lambda) - g_i(\bar{\theta}^t, \bar{\lambda}^t)\right) + \frac{2\eta_\lambda L}{m}\Xi_\lambda^t + 2\eta_\lambda L D_\lambda^t \sqrt{\frac{\Xi_\theta^t}{m}}. \tag{B.50}$$

Plugging it back in (B.45), rearranging the terms and taking the expectation over the previous iterate we get

$$\mathbb{E}\left[g(\bar{\theta}^t, \lambda) - g(\bar{\theta}^t, \bar{\lambda}^t)\right] = \frac{1}{m}\mathbb{E}\left[\sum_{i=1}^{m} g_i(\bar{\theta}^t, \lambda) - g_i(\bar{\theta}^t, \bar{\lambda}^t)\right] \tag{B.51}$$

$$\leq \frac{\mathbb{E}\|\bar{\lambda}^t - \lambda\|^2 - \mathbb{E}\|\bar{\lambda}^{t+1} - \lambda\|^2}{2\eta_\lambda} + \frac{\eta_\lambda}{2}G_\lambda^2 + \frac{L}{m}\mathbb{E}\left[\Xi_\lambda^t\right] + L\mathbb{E}\left[D_\lambda^t\right]\sqrt{\frac{\mathbb{E}\left[\Xi_\theta^t\right]}{m}}. \tag{B.52}$$

Telescoping from $t = 0$ to $t = T - 1$ and plugging the consensus inequalities (B.21) and (B.22) we get

$$\frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1} g(\bar{\theta}^t, \lambda) - g(\bar{\theta}^t, \bar{\lambda}^t)\right] \leq \frac{D_\lambda}{2\eta_\lambda T} + \frac{\eta_\lambda}{2}G_\lambda^2 + 4\eta_\lambda^2\frac{LG_\lambda^2}{\rho^2} + \sqrt{1}2\eta_\theta\frac{D_\lambda L G_\theta}{c}. \tag{B.53}$$

where $D_\lambda = \max_{t=0\ldots,T}\mathbb{E}[D_\lambda^t] = \max_{t=0\ldots,T}\mathbb{E}\left\|\bar{\lambda}^t - \lambda\right\|$. □

B.3 Proof of Theorem 3: Non-convex case

In the case of non-convex functions $\{f_i\}_{i=1}^m$, Theorem 3 provides the following ϵ-stationarity guarantee on the randomized solution of Algorithm 2 :

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\Phi(\bar{\theta}^{t-1})\right\|^2\right] \leq \frac{2L}{\sqrt{T}}\left(256\left(\mathbb{E}[\Phi(\bar{\theta}^0)] - \mathbb{E}[\Phi(\bar{\theta}^T)]\right) + \frac{45L\kappa^2 D_\lambda^0}{2}\right) + \frac{1}{\sqrt{T}}\left(5D_\lambda L\frac{G_\theta}{c} + \frac{\sigma_\theta^2}{2m} + \frac{45\kappa\sigma_\lambda^2}{4m}\right) + \frac{1}{T}\left(\frac{G_\theta^2}{4c^2} + 171\frac{\kappa G_\lambda^2}{\rho^2}\right) + \frac{\sigma_\theta^2}{m}. \tag{B.54}$$

The proof is inspired from recent results in [214]. Specifically, Lemma 8, stated and proved below, provides a descent inequality of the type

$$\begin{aligned}\mathbb{E}[\Phi(\bar{\theta}^t)] \leq & \mathbb{E}[\Phi(\bar{\theta}^{t-1})] + \frac{\eta_\theta^2 \kappa L \sigma_\theta^2}{m} - \left(\frac{\eta_\theta}{2} - 2\eta_\theta^2 \kappa L\right) \mathbb{E}\left[\nabla \|\Phi(\bar{\theta}^{t-1})\|^2\right] \\ & + L^2 \left(\frac{\eta_\theta}{2} + 2\eta_\theta^2 \kappa L\right) \left(\frac{\mathbb{E}[\Xi_\theta^t]}{m} + \frac{2\mathbb{E}[\Xi_\lambda^t]}{m} + 2\mathbb{E}[\delta_\lambda^t]\right). \end{aligned} \tag{B.55}$$

Setting $\eta_\theta = \frac{\eta_\lambda}{16(\kappa+1)^2}$ and $\eta_\lambda \leq \frac{1}{2L}$ expression (B.55) can be simplified thanks to the following chain of inequalities

$$\frac{7\eta_\theta}{16} \leq \eta_\theta(\frac{1}{2} - 2\eta_\theta \kappa L) \leq \eta_\theta(\frac{1}{2} + 2\eta_\theta \kappa L) \leq \frac{9\eta_\theta}{16}. \tag{B.56}$$

Telescoping the simplified expression from $t = 1$ to T we obtain

$$\begin{aligned}\mathbb{E}[\Phi(\bar{\theta}^T)] \leq & \mathbb{E}[\Phi(\bar{\theta}^0)] + T\frac{\eta_\theta^2 \kappa L \sigma_\theta^2}{m} - \frac{7\eta_\theta}{16} \sum_{t=1}^{T} \mathbb{E}\left[\nabla \|\Phi(\bar{\theta}^{t-1})\|^2\right] \\ & + L^2 \frac{9\eta_\theta}{16} \sum_{t=1}^{T} \left(\frac{\mathbb{E}[\Xi_\theta^t]}{m} + \frac{2\mathbb{E}[\Xi_\lambda^t]}{m}\right) + \frac{9\eta_\theta L^2}{8} \mathbb{E}\left[\sum_{t=1}^{T} \delta_\lambda^t\right] \end{aligned} \tag{B.57}$$

where $\delta_\lambda^t := \left\|\lambda^*(\bar{\theta}^t) - \bar{\lambda}^t\right\|^2$ represents the squared distance between the optimal value of the dual variable for the current averaged network belief and the current averaged value of the dual variable.

Lemma 9, reported below, provides a bound on $\sum_{t=1}^{T} \delta_\lambda^t$ that plugged in (B.57) yields

$$\begin{aligned}\mathbb{E}[\Phi(\bar{\theta}^T)] \leq & \mathbb{E}[\Phi(\bar{\theta}^0)] + \eta_\theta \frac{45L\kappa^2 \delta_\lambda^0}{8\eta_\lambda} + \eta_\theta \left(\frac{45\kappa^4 \eta_\theta^2}{\eta_\lambda^2} - \frac{7}{16}\right) \sum_{t=1}^{T} \mathbb{E}\left[\nabla \left\|\Phi(\bar{\theta}^{t-1})\right\|^2\right] \\ & + T\eta_\theta \left(\frac{\eta_\theta \kappa L \sigma_\theta^2}{m} + \frac{45\kappa L \eta_\lambda \sigma_\lambda^2}{4m} + \frac{45\sigma_\theta^2}{2 \cdot 16^2 m}\right) \\ & + L^2 \frac{9\eta_\theta}{16} \sum_{t=1}^{T} \left(\frac{\mathbb{E}[\Xi_\theta^t]}{m} + \frac{2\mathbb{E}[\Xi^t]_\lambda}{m} + \frac{30\kappa \mathbb{E}[\Xi_\theta^{t-1}]}{m} + \frac{70\kappa \mathbb{E}[\Xi_\lambda^{t-1}]}{m}\right) \\ & + L^2 \frac{9\eta_\theta}{16} \sum_{t=1}^{T} \left(40\kappa D_\lambda^{t-1} \sqrt{\frac{1}{m}\mathbb{E}[\Xi_\theta^{t-1}]}\right). \end{aligned} \tag{B.58}$$

Moreover, the relation between the two step-sizes established above ensures that

$$\left(\frac{45\kappa^4 \eta_\theta^2}{\eta_\lambda^2} - \frac{7}{16}\right) \leq -\frac{1}{4} \tag{B.59}$$

and therefore rearranging terms, dividing by $\frac{4}{T\eta_\theta}$ and recalling that $\kappa \geq 1$

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\nabla \left\|\Phi(\bar{\theta}^{t-1})\right\|^2\right] \leq \frac{4}{\eta_\theta T} \left(\mathbb{E}[\Phi(\bar{\theta}^0)] - \mathbb{E}[\Phi(\bar{\theta}^T)]\right) + \frac{45L\kappa^2 \delta_\lambda^0}{2T\eta_\lambda}$$

$$+ 4\left(\frac{\eta_\theta \kappa L \sigma_\theta^2}{m} + \frac{45\kappa L \eta_\lambda \sigma_\lambda^2}{4m} + \frac{45\sigma_\theta^2}{2 \cdot 16^2 m}\right)$$
$$+ \frac{9L^2}{4T}\sum_{t=1}^{T}\left(\frac{31\kappa \mathbb{E}[\Xi_\theta^{t-1}]}{m} + \frac{72\mathbb{E}[\kappa \Xi_\lambda^{t-1}]}{m}\right)$$
$$+ \frac{9L^2}{4T}\sum_{t=1}^{T}\left(40\kappa D_\lambda^{t-1}\sqrt{\frac{1}{m}\mathbb{E}[\Xi_\theta^{t-1}]}\right). \tag{B.60}$$

Exploiting consensus inequalities (B.21), (B.22) and the fact that $\kappa \geq 1$ and $\eta_\theta = \frac{\eta_\lambda}{16(\kappa+1)^2} \leq 1/2L$ we can simplify and obtain

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\nabla\left\|\Phi(\bar{\theta}^{t-1})\right\|^2\right] \leq \frac{64(\kappa+1)^2}{\eta_\lambda T}\left(\mathbb{E}[\Phi(\bar{\theta}^0)] - \mathbb{E}[\Phi(\bar{\theta}^T)]\right) + \frac{45L\kappa^2\delta_\lambda^0}{2T\eta_\lambda}$$
$$+ 2\left(\eta_\lambda \frac{L\sigma_\theta^2}{m} + \eta_\lambda \frac{45\kappa L \sigma_\lambda^2}{2m} + \frac{45\sigma_\theta^2}{16^2 m}\right)$$
$$+ L^2 \frac{9}{4T}\sum_{t=1}^{T}\left(40\kappa D_\lambda^{t-1}\sqrt{12}\eta_\theta \frac{G_\theta}{c} + 372\eta_\theta^2 \frac{\kappa G_\theta^2}{c^2} + 288\eta_\lambda^2 \frac{\kappa G_\lambda^2}{\rho^2}\right). \tag{B.61}$$

Simplifying and defining $D_\lambda = \max_{t=0,\dots,T} D_\lambda^t$

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\nabla\left\|\Phi(\bar{\theta}^{t-1})\right\|^2\right] \leq \frac{64(\kappa+1)^2}{\eta_\lambda T}\left(\mathbb{E}[\Phi(\bar{\theta}^0)] - \mathbb{E}[\Phi(\bar{\theta}^T)]\right) + \frac{45L\kappa^2\delta_\lambda^0}{2T\eta_\lambda}$$
$$+ 2\left(\eta_\lambda \frac{L\sigma_\theta^2}{m} + \eta_\lambda \frac{45\kappa L \sigma_\lambda^2}{2m} + \frac{45\sigma_\theta^2}{16^2 m}\right)$$
$$+ L^2\left(10 D_\lambda \eta_\lambda \frac{G_\theta}{c} + \eta_\lambda^2 \frac{G_\theta^2}{c^2} + 684\eta_\lambda^2 \frac{\kappa G_\lambda^2}{\rho^2}\right). \tag{B.62}$$

Grouping

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\nabla\left\|\Phi(\bar{\theta}^{t-1})\right\|^2\right] \leq \frac{1}{\eta_\lambda T}\left(256\left(\mathbb{E}[\Phi(\bar{\theta}^0)] - \mathbb{E}[\Phi(\bar{\theta}^T)]\right) + \frac{45L\kappa^2\delta_\lambda^0}{2}\right)$$
$$+ \eta_\lambda\left(10 D_\lambda L^2 \frac{G_\theta}{c} + \frac{L\sigma_\theta^2}{m} + \frac{45\kappa L\sigma_\lambda^2}{2m}\right)$$
$$+ \eta_\lambda^2\left(\frac{L^2 G_\theta^2}{c^2} + 684\frac{L^2\kappa G_\lambda^2}{\rho^2}\right) + \frac{\sigma_\theta^2}{m}. \tag{B.63}$$

Setting $\eta_\lambda = \frac{1}{2L\sqrt{T}}$ we get

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\nabla\left\|\Phi(\bar{\theta}^{t-1})\right\|^2\right] \leq \frac{2L}{\sqrt{T}}\left(256\left(\mathbb{E}[\Phi(\bar{\theta}^0)] - \mathbb{E}[\Phi(\bar{\theta}^T)]\right) + \frac{45L\kappa^2\delta_\lambda^0}{2}\right)$$

$$+\frac{1}{\sqrt{T}}\left(5D_\lambda L\frac{G_\theta}{c}+\frac{\sigma_\theta^2}{2m}+\frac{45\kappa\sigma_\lambda^2}{4m}\right)$$
$$+\frac{1}{T}\left(\frac{G_\theta^2}{4c^2}+171\frac{\kappa G_\lambda^2}{\rho^2}\right)+\frac{\sigma_\theta^2}{m}. \tag{B.64}$$

Lemma 8. *For each $t=1,\ldots,T$ the iterates generated by Algorithm 2 satisfies*

$$\mathbb{E}[\Phi(\bar{\theta}^t)] \leq \mathbb{E}[\Phi(\bar{\theta}^{t-1})]+\frac{\eta_\theta^2\kappa L\sigma_\theta^2}{m}-\left(\frac{\eta_\theta}{2}-2\eta_\theta^2\kappa L\right)\mathbb{E}\left[\nabla\|\Phi(\bar{\theta}^{t-1})\|^2\right]$$
$$+L^2\left(\frac{\eta_\theta}{2}+2\eta_\theta^2\kappa L\right)\left(\frac{\mathbb{E}[\Xi_\theta^t]}{m}+\frac{2\mathbb{E}[\Xi_\lambda^t]}{m}+2\mathbb{E}[\delta_\lambda^t]\right). \tag{B.65}$$

Proof: From the $2\kappa L$-smoothness of $\Phi(\cdot)$ (Lemma 4.3 of [214]) and the update rule we have:

$$\mathbb{E}_{\xi^{t-1}}\left[\Phi(\bar{\theta}^t)\right] \leq \Phi(\bar{\theta}^{t-1})+\mathbb{E}_{\xi^{t-1}}\left[\langle\nabla_\theta\Phi(\bar{\theta}^{t-1}),\bar{\theta}^t-\bar{\theta}^{t-1}\rangle\right]+\kappa L\mathbb{E}_{\xi^{t-1}}\left\|\bar{\theta}^t-\bar{\theta}^{t-1}\right\|^2 \tag{B.66}$$
$$\leq \Phi(\bar{\theta}^{t-1})-\eta_\theta\langle\nabla_\theta\Phi(\bar{\theta}^{t-1}),\frac{1}{m}\sum_{i=1}^m\mathbb{E}_{\xi^{t-1}}\left[\tilde{\nabla}_\theta g_i(\theta_i^{t-1},\lambda_i^{t-1})\right]\rangle$$
$$+\frac{\eta_\theta^2\kappa L}{m^2}\mathbb{E}_{\xi^{t-1}}\left\|\sum_{i=1}^m\tilde{\nabla}_\theta g_i(\theta_i^{t-1},\lambda_i^{t-1})\right\|^2 \tag{B.67}$$
$$\leq \Phi(\bar{\theta}^{t-1})+\underbrace{\eta_\theta\langle\nabla\Phi(\bar{\theta}^{t-1}),\nabla\Phi(\bar{\theta}^{t-1})-\frac{1}{m}\sum_{i=1}^m\nabla_\theta g_i(\theta_i^{t-1},\lambda_i^{t-1})\rangle}_{:=T_4}$$
$$-\eta_\theta\nabla\|\Phi(\bar{\theta}^{t-1})\|^2+\frac{\eta_\theta^2\kappa L}{m^2}\underbrace{\mathbb{E}_{\xi^t}\left\|\sum_{i=1}^m\tilde{\nabla}_\theta g_i(\theta_i^{t-1},\lambda_i^{t-1})\right\|^2}_{:=T_5}. \tag{B.68}$$

We now turn bounding term T_4

$$T_4=\eta_\theta\langle\nabla\Phi(\bar{\theta}^{t-1}),\nabla\Phi(\bar{\theta}^{t-1})-\frac{1}{m}\sum_{i=1}^m\nabla_\theta g_i(\theta_i^{t-1},\lambda_i^{t-1})\rangle \tag{B.69}$$
$$\overset{(B.8)}{\leq}\frac{\eta_\theta}{2}\left(\left\|\nabla\Phi(\bar{\theta}^{t-1})\right\|^2+\left\|\nabla\Phi(\bar{\theta}^{t-1})-\frac{1}{m}\sum_{i=1}^m\nabla_\theta g_i(\theta_i^{t-1},\lambda_i^{t-1})\right\|^2\right) \tag{B.70}$$
$$\leq\frac{\eta_\theta}{2}\left(\left\|\nabla\Phi(\bar{\theta}^{t-1})\right\|^2+\left\|\frac{1}{m}\sum_{i=1}^m\nabla_\theta g_i(\bar{\theta}^{t-1},\lambda^*(\bar{\theta}^{t-1}))-\nabla_\theta g_i(\theta_i^{t-1},\lambda_i^{t-1})\right\|^2\right) \tag{B.71}$$
$$\overset{(B.1)}{\leq}\frac{\eta_\theta}{2}\left(\left\|\nabla\Phi(\bar{\theta}^{t-1})\right\|^2+\frac{L^2}{m}\sum_{i=1}^m\left\|\bar{\theta}^{t-1}-\theta_i^{t-1}\right\|^2+\frac{L^2}{m}\sum_{i=1}^m\left\|\lambda^*(\bar{\theta}^{t-1})-\lambda_i^{t-1})\right\|^2\right) \tag{B.72}$$

$$\overset{(B.9)}{\leq} \frac{\eta_\theta}{2}\left(\left\|\nabla\Phi(\bar{\theta}^{t-1})\right\|^2 + \frac{L^2\Xi_\theta^{t-1}}{m} + \frac{2L^2\Xi_\lambda^{t-1}}{m} + \frac{2L^2}{m}\sum_{i=1}^{m}\underbrace{\left\|\lambda^*(\bar{\theta}^{t-1}) - \bar{\lambda}^{t-1}\right\|^2}_{=\delta_\lambda^{t-1}}\right) \tag{B.73}$$

$$\tag{B.74}$$

and from stochastic gradient assumptions 6 we can bound T_5 as follows

$$T_5 = \mathbb{E}_{\xi^{t-1}}\left\|\sum_{i=1}^{m}\tilde{\nabla}_\theta g_i(\theta_i^{t-1}, \lambda_i^{t-1})\right\|^2 \tag{B.75}$$

$$= \mathbb{E}_{\xi^{t-1}}\left[\left\|\sum_{i=1}^{m}\left(\tilde{\nabla}_\theta g_i(\theta_i^{t-1}, \lambda_i^{t-1}) - \nabla_\theta g_i(\theta_i^{t-1}, \lambda_i^{t-1})\right)\right\|^2\right] + \left\|\sum_{i=1}^{m}\nabla_\theta g_i(\theta_i^{t-1}, \lambda_i^{t-1})\right\|^2 \tag{B.76}$$

$$\overset{(B.9)}{\leq} m\sigma_\theta^2 + 2\left\|\sum_{i=1}^{m}\nabla_\theta g_i(\theta_i^{t-1}, \lambda_i^{t-1}) - \nabla_\theta g_i(\bar{\theta}^{t-1}, \lambda^*(\bar{\theta}^{t-1}))\right\|^2 + 2\left\|m\nabla\Phi(\bar{\theta}^{t-1})\right\|^2 \tag{B.77}$$

$$\overset{(B.1)}{\leq} m\sigma_\theta^2 + 2L^2 m\sum_{i=1}^{m}\left\|\bar{\theta}^{t-1} - \theta_i^{t-1}\right\|^2 + 2L^2 m\sum_{i=1}^{m}\left\|\lambda^*(\bar{\theta}^{t-1}) - \lambda_i^{t-1}\right\|^2 + 2m^2\left\|\nabla\Phi(\bar{\theta}^{t-1})\right\|^2 \tag{B.78}$$

$$\leq m\sigma_\theta^2 + 2L^2 m\Xi_\theta^{t-1} + 4L^2 m\Xi_\lambda^{t-1} + 4L^2 m\sum_{i=1}^{m}\underbrace{\left\|\lambda^*(\bar{\theta}^{t-1}) - \bar{\lambda}^{t-1}\right\|^2}_{=\delta_\lambda^{t-1}} + 2m^2\left\|\nabla\Phi(\bar{\theta}^{t-1})\right\|^2. \tag{B.79}$$

Recombining, grouping, and taking the expectation over the previous iterates we get the desired result. □

Lemma 9. *The sequence of $\{\delta_\lambda^t\}_{t=1}^T$ generated by Algorithm 2 satisfies*

$$\sum_{t=1}^{T}\mathbb{E}\left[\delta_\lambda^t\right] \leq \frac{5\delta_\lambda^0\kappa}{\eta_\lambda\mu} + \sum_{t=1}^{T}5\kappa\left(4D_\lambda^{t-1}\sqrt{\frac{1}{m}\mathbb{E}\left[\Xi_\theta^{t-1}\right]} + \frac{3\mathbb{E}\left[\Xi_\theta^{t-1}\right]}{m} + \frac{7\mathbb{E}\left[\Xi_\lambda^{t-1}\right]}{m}\right) + \sum_{t=1}^{T}5\left(\frac{8\kappa^2\eta_\theta^2}{\eta_\lambda^2\mu^2}\mathbb{E}\left[\left\|\nabla\Phi(\bar{\theta}^{t-1})\right\|^2\right]\right) + 5T\left(\frac{2\eta_\lambda\sigma_\lambda^2}{m\mu} + \frac{4\sigma_\theta^2}{16^2 m(\kappa+1)^2\mu^2}\right) \tag{B.80}$$

where $D_\lambda^{t-1} = \left\|\bar{\lambda}^{t-1} - \lambda\right\|$.

Proof: From (B.9), for $b > 0$, we have

$$\mathbb{E}_{\xi^{t-1}}[\delta_\lambda^t] \leq \left(1 + \frac{1}{b}\right)\underbrace{\mathbb{E}_{\xi^{t-1}}\left\|\lambda^*(\bar{\theta}^{t-1}) - \bar{\lambda}^t\right\|^2}_{:=T_6} + (1+b)\underbrace{\mathbb{E}_{\xi^{t-1}}\left\|\lambda^*(\bar{\theta}^t) - \lambda^*(\bar{\theta}^{t-1})\right\|^2}_{:=T_7}. \tag{B.81}$$

Bounding T_6 similarly

$$T_6 = \mathbb{E}_{\xi^{t-1}} \left\| \lambda^*(\bar{\theta}^{t-1}) - \bar{\lambda}^t \right\|^2 \tag{B.82}$$

$$= \mathbb{E}_{\xi^{t-1}} \left\| \lambda^*(\bar{\theta}^{t-1}) - \bar{\lambda}^{t-1} - \frac{\eta_\lambda}{m} \sum_{i=1}^m \left(\tilde{\nabla}_\lambda g_i(\theta_i^{t-1}, \lambda_i^{t-1}) \pm \nabla_\lambda g_i(\theta_i^{t-1}, \lambda_i^{t-1}) \right) \right\|^2 \tag{B.83}$$

$$\leq \left\| \lambda^*(\bar{\theta}^{t-1}) - \bar{\lambda}^{t-1} - \frac{\eta_\lambda}{m} \sum_{i=1}^m \nabla_\lambda g_i(\theta_i^{t-1}, \lambda_i^{t-1}) \right\|^2 + \frac{\eta_\lambda^2 \sigma_\lambda^2}{m} \tag{B.84}$$

$$= \left\| \lambda^*(\bar{\theta}^{t-1}) - \bar{\lambda}^{t-1} \right\|^2 + \underbrace{\left\| \frac{\eta_\lambda}{m} \sum_{i=1}^m \nabla_\lambda g_i(\theta_i^{t-1}, \lambda_i^{t-1}) \right\|^2}_{T_{6,1}} + \frac{\eta_\lambda^2 \sigma_\lambda^2}{m}$$

$$\underbrace{-2 \langle \lambda^*(\bar{\theta}^{t-1}) - \bar{\lambda}^{t-1}; \frac{\eta_\lambda}{m} \sum_{i=1}^m \nabla_\lambda g_i(\theta_i^{t-1}, \lambda_i^{t-1}) \rangle}_{T_{6,2}} . \tag{B.85}$$

Estimating $T_{6,1}$

$$T_{6,1} = 2\eta_\lambda^2 \left\| \frac{1}{m} \sum_{i=1}^m \nabla_\lambda g_i(\theta_i^{t-1}, \lambda_i^{t-1}) \pm \nabla_\lambda g_i(\bar{\theta}^{t-1}, \bar{\lambda}^{t-1}) - \nabla_\lambda g_i(\bar{\theta}^{t-1}, \lambda^*(\bar{\theta}^{t-1})) \right\|^2 \tag{B.86}$$

$$\leq \frac{2\eta_\lambda^2}{m} \sum_{i=1}^m \left\| \nabla_\lambda g_i(\theta_i^{t-1}, \lambda_i^{t-1}) - \nabla_\lambda g_i(\bar{\theta}^{t-1}, \bar{\lambda}^{t-1}) \right\|^2 + 2\eta_\lambda^2 \left\| \frac{1}{m} \sum_{i=1}^m \nabla_\lambda g_i(\bar{\theta}^{t-1}, \bar{\lambda}^{t-1}) - \nabla_\lambda g_i(\bar{\theta}^{t-1}, \lambda^*(\bar{\theta}^{t-1})) \right\|^2 \tag{B.87}$$

$$\overset{(B.1, B.5)}{\leq} \frac{2\eta_\lambda^2}{m} \sum_{i=1}^m L^2 \left\| \lambda_i^{t-1} - \bar{\lambda}^{t-1} \right\|^2 + L^2 \left\| \theta_i^{t-1} - \bar{\theta}^{t-1} \right\|^2 + \frac{4\eta_\lambda^2 L}{m} \sum_{i=1}^m \left[g_i(\bar{\theta}^{t-1}, \lambda^*(\bar{\theta}^{t-1})) - g_i(\bar{\theta}^{t-1}, \bar{\lambda}^{t-1}) \right] \tag{B.88}$$

$$= \frac{2\eta_\lambda^2 L^2}{m} \Xi_\lambda^{t-1} + \frac{2\eta_\lambda^2 L^2}{m} \Xi_\theta^{t-1} + \frac{4\eta_\lambda^2 L}{m} \sum_{i=1}^m \left[g_i(\bar{\theta}^{t-1}, \lambda^*(\theta_i^{t-1})) - g_i(\bar{\theta}^{t-1}, \bar{\lambda}^{t-1}) \right] . \tag{B.89}$$

Estimating $T_{6,2}$

$$T_{6,2} = -2 \frac{\eta_\lambda}{m} \sum_{i=1}^m \langle \lambda^*(\bar{\theta}^{t-1}) - \bar{\lambda}^{t-1}; \nabla_\lambda g_i(\theta_i^{t-1}, \lambda_i^{t-1}) \rangle \tag{B.90}$$

$$= -2 \frac{\eta_\lambda}{m} \sum_{i=1}^m \langle \lambda^*(\bar{\theta}^{t-1}) - \bar{\lambda}^{t-1}; \nabla_\lambda g_i(\theta_i^{t-1}, \lambda_i^{t-1}) \pm \nabla_\lambda g_i(\bar{\theta}^{t-1}, \lambda_i^{t-1}) \rangle \tag{B.91}$$

$$= -2\frac{\eta_\lambda}{m}\sum_{i=1}^{m}\langle\lambda^*(\bar{\theta}^{t-1}) - \bar{\lambda}^{t-1}; \nabla_\lambda g_i(\bar{\theta}^{t-1}, \lambda_i^{t-1}\rangle$$
$$+ 2\frac{\eta_\lambda}{m}\sum_{i=1}^{m}\langle\bar{\lambda}^{t-1} - \lambda^*(\bar{\theta}^{t-1}); \nabla_\lambda g_i(\theta_i^{t-1}, \lambda_i^{t-1}) - \nabla_\lambda g_i(\bar{\theta}^{t-1}, \lambda_i^{t-1})\rangle \tag{B.92}$$
$$\leq -2\frac{\eta_\lambda}{m}\sum_{i=1}^{m}\langle\lambda^*(\bar{\theta}^{t-1}) - \bar{\lambda}^{t-1}; \nabla_\lambda g_i(\bar{x}^{t-1}, \lambda_i^{t-1}\rangle + 2\eta_\lambda L D_\lambda^{t-1}\sqrt{\frac{1}{m}\Xi_\theta^{t-1}} \tag{B.93}$$
$$= -2\frac{\eta_\lambda}{m}\sum_{i=1}^{m}\langle\lambda^*(\bar{\theta}^{t-1}) - \lambda_i^{t-1}; \nabla_\lambda g_i(\bar{\theta}^{t-1}, \lambda_i^{t-1})\rangle + \langle\lambda_i^{t-1} - \bar{\lambda}^{t-1}); \nabla_\lambda g_i(\bar{\theta}^{t-1}, \lambda_i^{t-1})\rangle$$
$$+ 2\eta_\lambda L D_\lambda^{t-1}\sqrt{\frac{1}{m}\Xi_\theta^{t-1}} \tag{B.94}$$
$$\overset{(B.4,B.7)}{\leq} 2\frac{\eta_\lambda}{m}\sum_{i=1}^{m}\left(g_i(\bar{\theta}^{t-1}, \bar{\lambda}^{t-1}) - g_i(\theta_i^{t-1}, \lambda^*(\bar{\theta}^{t-1}))\right) + 2\eta_\lambda L D_\lambda^{t-1}\sqrt{\frac{1}{m}\Xi_\theta^{t-1}}$$
$$- 2\frac{\eta_\lambda}{m}\sum_{i=1}^{m}\left(\frac{\mu}{2}\left\|\lambda^*(\bar{\theta}^{t-1}) - \lambda_i^{t-1}\right\|^2 + \frac{L}{2}\left\|\bar{\lambda}^{t-1} - \lambda_i^{t-1}\right\|^2\right) \tag{B.95}$$
$$\overset{(B.9)}{\leq} 2\frac{\eta_\lambda}{m}\sum_{i=1}^{m}\left(g_i(\bar{\theta}^{t-1}, \bar{\lambda}^{t-1}) - g_i(\bar{\theta}^{t-1}, \lambda^*(\bar{\theta}^{t-1}))\right) + 2\eta_\lambda L D_\lambda^{t-1}\sqrt{\frac{1}{m}\Xi_\theta^{t-1}}$$
$$- 2\frac{\eta_\lambda}{m}\sum_{i=1}^{m}\left(\frac{\mu}{4}\left\|\lambda^*(\bar{\theta}^{t-1}) - \bar{\lambda}^{t-1}\right\|^2 - \frac{L+\mu}{2}\left\|\bar{\lambda}^{t-1} - \lambda_i^{t-1}\right\|^2\right) \tag{B.96}$$
$$= -\frac{\mu\eta_\lambda}{2}\left\|\lambda^*(\bar{\theta}^{t-1}) - \bar{\lambda}^{t-1}\right\|^2 - 2\frac{\eta_\lambda}{m}\sum_{i=1}^{m} g_i(\bar{\theta}^{t-1}, \lambda^*(\bar{\theta}^{t-1})) - g_i(\bar{\theta}^{t-1}, \bar{\lambda}^{t-1})$$
$$+ \frac{2L\eta_\lambda}{m}\Xi_\lambda^{t-1} + 2\eta_\lambda L D_\lambda^{t-1}\sqrt{\frac{1}{m}\Xi_\theta^{t-1}} \tag{B.97}$$

where the last inequality follows from choosing $\eta_\lambda \leq 1/(2L)$. Substituting the expressions we get

$$T_6 = \left(1 - \frac{\mu\eta_\lambda}{2}\right)\left\|\lambda^*(\bar{\theta}^{t-1}) - \bar{\lambda}^{t-1}\right\|^2 + \frac{\eta_\lambda^2\sigma_\lambda^2}{m} + \frac{L\eta_\lambda}{m}\Xi_\theta^{t-1}$$
$$+ \frac{3L\eta_\lambda}{m}\Xi_\lambda^{t-1} + 2\eta_\lambda L D_\lambda^{t-1}\sqrt{\frac{1}{m}\Xi_\theta^{t-1}}. \tag{B.98}$$

Being $\lambda^*(\cdot)$ is κ-smooth (Lemma 4.3 [214]) we can bound T_7 as follows

$$T_7 = \mathbb{E}_{\xi^{t-1}}\left\|\lambda^*(\bar{\theta}^t) - \lambda^*(\bar{\theta}^{t-1})\right\|^2 \tag{B.99}$$
$$\leq \kappa^2\mathbb{E}_{\xi^{t-1}}\left\|\bar{\theta}^t - \bar{\theta}^{t-1}\right\|^2 \tag{B.100}$$
$$= \frac{\kappa^2\eta_\theta^2}{m^2}\mathbb{E}_{\xi^{t-1}}\left\|\sum_{i=1}^{m}\tilde{\nabla}_\theta g_i(\theta_i^{t-1}, \lambda_i^{t-1})\right\|^2 \tag{B.101}$$

$$=\frac{\kappa^2\eta_\theta^2}{m^2}\left(m\sigma_\theta^2+\left\|\sum_{i=1}^m\nabla_\theta g_i(\theta_i^{t-1},\lambda_i^{t-1})\pm m\nabla\Phi(\bar{\theta}^{t-1})\right\|^2\right) \tag{B.102}$$

$$\overset{(B.9)}{\leq}\frac{\kappa^2\eta_\theta^2}{m^2}\Bigg(m\sigma_\theta^2+2\left\|m\nabla\Phi(\bar{\theta}^{t-1})\right\|^2+2L^2m\sum_{i=1}^m\left\|\bar{\theta}^{t-1}-\theta_i^{t-1}\right\|^2$$
$$+2L^2m\sum_{i=1}^m\left\|\lambda^*(\bar{\theta}^{t-1})-\lambda_i^{t-1}\right\|^2\Bigg) \tag{B.103}$$

$$=\frac{\kappa^2\eta_\theta^2}{m^2}\Bigg(m\sigma_\theta^2+2m^2\left\|\nabla\Phi(\bar{\theta}^{t-1})\right\|^2+2L^2m\Xi_\theta^{t-1}$$
$$+2L^2m\sum_{i=1}^m\left\|\lambda^*(\bar{\theta}^{t-1})-\bar{\lambda}^{t-1}+\bar{\lambda}^{t-1}-\lambda_i^{t-1}\right\|^2\Bigg) \tag{B.104}$$

$$\overset{(B.9)}{\leq}\kappa\eta_\theta^2\left(\frac{\sigma_\theta^2}{m}+2\left\|\nabla\Phi(\bar{\theta}^{t-1})\right\|^2+\frac{2L^2\Xi_\theta^{t-1}}{m}+\frac{4L^2\Xi_\lambda^{t-1}}{m}+4L^2\left\|\lambda^*(\bar{\theta}^{t-1})-\bar{\lambda}^{t-1}\right\|^2\right) \tag{B.105}$$

$$=\kappa^2\eta_\theta^2\left(\frac{\sigma_\theta^2}{m}+2\left\|\nabla\Phi(\bar{\theta}^{t-1})\right\|^2+\frac{2L^2\Xi_\theta^{t-1}}{m}+\frac{4L^2\Xi_\lambda^{t-1}}{m}+4L^2\delta_\lambda^{t-1}\right). \tag{B.106}$$

Recombining and grouping we get

$$\delta_\lambda^t\leq\left(\left(1+\frac{1}{b}\right)\left(1-\frac{\mu\eta_\lambda}{2}\right)+4(1+b)\kappa^2\eta_\theta^2L^2\right)\delta_\lambda^{t-1}+2\left(1+\frac{1}{b}\right)\eta_\lambda LD_\lambda^{t-1}\sqrt{\frac{1}{m}\Xi_\theta^{t-1}}$$
$$+\left(\left(1+\frac{1}{b}\right)L\eta_\lambda+2(1+b)\kappa^2\eta_\theta^2L^2\right)\frac{\Xi_\theta^{t-1}}{m}+\left(1+\frac{1}{b}\right)\frac{\eta_\lambda^2\sigma_\lambda^2}{m}+(1+b)\,\kappa^2\eta_\theta^2\frac{\sigma_\theta^2}{m}$$
$$+\left(\left(1+\frac{1}{b}\right)3L\eta_\lambda+4(1+b)\kappa^2\eta_\theta^2L^2\right)\frac{\Xi_\lambda^{t-1}}{m}+2\kappa^2\eta_\theta^2(1+b)\left\|\nabla\Phi(\bar{\theta}^{t-1})\right\|^2. \tag{B.107}$$

Setting $b=2\left(\frac{2}{\eta_\lambda\mu}-1\right)>0$ we get the following inequalities

$$\left(1+\frac{1}{b}\right)\left(1-\frac{\eta_\lambda\mu}{2}\right)\leq\left(1-\frac{\eta_\lambda\mu}{4}\right), \tag{B.108}$$

$$(1+b)\leq\frac{4}{\eta_\lambda\mu}, \tag{B.109}$$

$$\left(1+\frac{1}{b}\right)\leq 2. \tag{B.110}$$

that allows to simplify (B.107) as follows

$$\delta_\lambda^t\leq\left(1-\frac{\eta_\lambda\mu}{4}+\frac{16\kappa^2\eta_\theta^2L^2}{\eta_\lambda\mu}\right)\delta_\lambda^{t-1}+4\eta_\lambda LD_\lambda^{t-1}\sqrt{\frac{1}{m}\Xi_\theta^{t-1}}$$
$$+\left(2L\eta_\lambda+\frac{8\kappa^2\eta_\theta^2L^2}{\eta_\lambda\mu}\right)\frac{\Xi_\theta^{t-1}}{m}+2\frac{\eta_\lambda^2\sigma_\lambda^2}{m}+\frac{4\kappa^2\eta_\theta^2\sigma_\theta^2}{m\eta_\lambda\mu}$$

$$+\left(6L\eta_\lambda+\frac{16\kappa^2\eta_\theta^2L^2}{\eta_\lambda\mu}\right)\frac{\Xi_\lambda^{t-1}}{m}+\frac{8\kappa^2\eta_\theta^2}{\eta_\lambda\mu}\left\|\nabla\Phi(\bar{x}^{t-1})\right\|^2. \tag{B.111}$$

Fixing $\eta_x=\frac{\eta_\lambda}{16(\kappa+1)^2}$ we get that

$$\nu=1-\frac{\eta_\lambda\mu}{4}+\frac{16\kappa^2\eta_x^2L^2}{\eta_\lambda\mu}\leq\left(1-\frac{\eta_\lambda\mu}{5}\right). \tag{B.112}$$

Taking the expectation over the current iterate and applying recursively the inequality we obtain

$$\begin{aligned}\mathbb{E}_{\xi^{t-1}}[\delta_\lambda^t]\leq&\nu^t\delta_\lambda^0+\sum_{i=0}^{t-1}\nu^{t-1-i}\left(\frac{8\kappa^2\eta_\theta^2}{\eta_\lambda\mu}\mathbb{E}_{\xi^{t-1}}[\left\|\nabla\Phi(\bar{\theta}^{t-1})\right\|^2]+2\frac{\eta_\lambda^2\sigma_\lambda^2}{m}+\frac{4\kappa^2\eta_\theta^2}{\eta_\lambda\mu}\frac{\sigma_\theta^2}{m}\right)\\&+\sum_{i=0}^{t-1}\nu^{t-1-i}\left(4\eta_\lambda LD_\lambda^{t-1}\sqrt{\frac{1}{m}\mathbb{E}_{\xi^{t-1}}[\Xi_\theta^{t-1}]}\right)\\&+\sum_{i=0}^{t-1}\nu^{t-1-i}\left(\frac{3L\eta_\lambda\mathbb{E}_{\xi^{t-1}}[\Xi_\theta^{t-1}]}{m}+\frac{7L\eta_\lambda\mathbb{E}_{\xi^{t-1}}[\Xi_\lambda^{t-1}]}{m}\right).\end{aligned} \tag{B.113}$$

Summing from $t=1$ to T and from (B.112) we get

$$\begin{aligned}\sum_{t=1}^{T}\mathbb{E}_{\xi^{t-1}}[\delta_\lambda^t]\leq&\frac{5\delta_\lambda^0}{\eta_\lambda\mu}+\sum_{t=1}^{T}\frac{5}{\eta_\lambda\mu}\left(\frac{8\kappa^2\eta_\theta^2}{\eta_\lambda\mu}\mathbb{E}_{\xi^{t-1}}[\left\|\nabla\Phi(\bar{\theta}^{t-1})\right\|^2\right)+\frac{5T}{\eta_\lambda\mu}\left(2\frac{\eta_\lambda^2\sigma_\lambda^2}{m}+\frac{4\kappa^2\eta_\theta^2}{\eta_\lambda\mu}\frac{\sigma_\theta^2}{m}\right)\\&+\sum_{t=1}^{T}\frac{5}{\eta_\lambda\mu}\left(4\eta_\lambda LD_\lambda^{t-1}\sqrt{\frac{1}{m}\mathbb{E}_{\xi^{t-1}}[\Xi_\theta^{t-1}]}\right)\\&+\sum_{t=1}^{T}\frac{5}{\eta_\lambda\mu}\left(\frac{3L\eta_\lambda\mathbb{E}_{\xi^{t-1}}[\Xi_\theta^{t-1}]}{m}+\frac{7L\eta_\lambda\mathbb{E}_{\xi^{t-1}}[\Xi_\lambda^{t-1}]}{m}\right).\end{aligned} \tag{B.114}$$

□

Appendix C

Appendix of Chapter 5

C.1 Proof of Theorem 4

Denote by f^* the $\arg\min_{f\in\mathcal{F}} E_{z\sim P_i}[\ell(f,z)]$ and bound the estimation error of $\hat{f}_{\vec{w}_i}$ as

$$\begin{aligned}
&Exc(\hat{f}_{\vec{w}_i}, P_i) = E_{z\sim P_i}[\ell(\hat{f}_{\vec{w}_i}, z)] - E_{z\sim P_i}[\ell(f^*, z)] \\
&\leq E_{z\sim P_{\vec{w}_i}}[\ell(\hat{f}_{\vec{w}_i}, z)] - E_{z\sim P_{\vec{w}_i}}[\ell(f^*, z)] + 2d_{\mathcal{F}}(P_i, P_{\vec{w}_i}) + 2\lambda \\
&\leq E_{z\sim P_{\vec{w}_i}}[\ell(\hat{f}_{\vec{w}_i}, z)] - \inf_{f\in\mathcal{F}} E_{z\sim P_{\vec{w}_i}}[\ell(f, z)] \\
&+ 2\sum_{j=1}^{m} w_{i,j} d_{\mathcal{F}}(P_i, P_j) + 2\lambda
\end{aligned}$$

where $\lambda = \arg\min_{f\in\mathcal{F}} \Big(E_{z\sim P_i}[\ell(f,z)] + E_{z\sim P_{\vec{w}_i}}[\ell(f,z)]\Big)$. We recognize the estimation error of $\hat{f}_{\vec{w}_i}$ w.r.t to the measure $P_{\vec{w}_i}$ that can be bounded following fairly standard approaches. In particular,

$$E_{z\sim P_{\vec{w}_i}}[\ell(\hat{f}_{\vec{w}_i}, z)] - \inf_{f\in\mathcal{F}} E_{z\sim P_{\vec{w}_i}}[\ell(f, z)] \leq 2\Delta(\mathcal{G}, Z)$$

where

$$\Delta(\mathcal{G}, Z) = \sup_{g\in\mathcal{G}} \left| E_{P_{\vec{w}_i}}[g(Z)] - \sum_{j=1}^{m} \frac{w_{i,j}}{n_i} \sum_{z\in\mathcal{D}_i} g(z) \right|.$$

is the uniform deviation term and

$$\mathcal{G} = \{Z \to \ell(f, Z) : f \in \mathcal{F}\}.$$

is the class resulting from the composition of the loss function $\ell(\cdot)$ and $\mathcal{F}$. The uniform deviation bound can be bounded in different ways, depending on the type of knowledge about the random variable $g(Z)$, in the following we assume that the loss function is bounded with range B and we exploit Azuma's inequality. In particular, the Doob's Martingale associated to the weighted loss will still have increments bounded by $\frac{w_{i,j}}{n_i} B$

depending to which loss term the increment is associated. Recognizing this, we can then directly apply Azuma's concentration bound and state that w.p. $1-\delta$ the following holds

$$\Delta(\mathcal{G}, Z) \leq E_P[\Delta(\mathcal{G}, Z)] + B\sqrt{\sum_{j=1}^{m} \frac{w_{i,j}^2}{n_j} \log\left(\frac{2}{\delta}\right)}$$

Finally, the expected uniform deviation can be bounded by the Rademacher complexity as follows

$$E_P[\Delta(\mathcal{G}, Z)] \leq 2\mathrm{Rad}(\mathcal{G})$$

where

$$\mathrm{Rad}(\mathcal{G}) = E_{\vec{\sigma},\mathcal{D}_1,\ldots,\mathcal{D}_j}\left[\sup_{g\in\mathcal{G}} \sum_{j=1}^{m} \frac{w_{i,j}}{n_i} \sum_{i=1}^{n_i} \sigma_{i,j} g(Z_{i,j})\right]$$

By a direct application of Massart's and Sauer's Lemma we obtain

$$\mathrm{Rad}(\mathcal{G}) \leq \sqrt{\sum_{j=1}^{m} \frac{w_{i,j}^2}{n_j}} \times \sqrt{\frac{2\mathrm{VCdim}(\mathcal{G})\left(\log\left(e\sum_j n_j\right) + \log(\mathrm{VCdim}(\mathcal{G}))\right)}{\sum_j n_j}}$$

combining everything together, we get the final result.

C.2 Proof of Theorem 5

Thanks to the upper bound on the target domain risk and the fact that the sum of two sub-Gaussian random variables of parameter σ is also sub-Gaussian with parameter 2σ, we can decompose the excess risk as

$$\begin{aligned} Exc(\hat{f}_{\vec{w}_i}, P_i) &= E_{z\sim P_i}[\ell(\hat{f}_{\vec{w}_i}, z)] - \inf_{f\in\mathcal{F}} E_{z\sim P_i}[\ell(f, z)] \\ &= E_{z\sim P_i}[\ell(\hat{f}_{\vec{w}_i}, z) - \ell(f^*, z)] \\ &\leq E_{z\sim P_{\vec{w}_i}}[\ell(\hat{f}_{\vec{w}_i}, z) - \ell(f^*, z)] + 2\beta\sigma^2 + \frac{D_{JS}(P_i||P_{\vec{w}_i})}{\beta} \end{aligned}$$

From the convexity of the KL-divergence we can bound the Jensen-Shannon divergence as follows

$$\begin{aligned} D_{JS}(P_i||P_{\vec{w}_i}) &= \frac{1}{2}KL\left(P_i||\frac{P_i + P_{\vec{w}_i}}{2}\right) + \frac{1}{2}KL\left(P_{\vec{w}_i}||\frac{P_i + P_{\vec{w}_i}}{2}\right) \\ &= \frac{1}{2}KL\left(P_i||\frac{\sum_j w_{i,j}(P_i + P_j)}{2}\right) \\ &+ \frac{1}{2}KL\left(\sum_j w_{i,j}P_j||\frac{\sum_j w_{i,j}(P_i + P_j)}{2}\right) \end{aligned}$$

$$\leq \quad \frac{1}{2}\sum_j w_{i,j}\left(KL\left(P_i||\frac{(P_i+P_j)}{2}\right)+KL\left(P_j||\frac{(P_i+P_j)}{2}\right)\right)$$
$$=\sum_j w_{i,j}D_{JS}(P_i||P_j)$$

Plugging it back into the previous expression and minimizing with respect to β we obtain

$$\begin{aligned}Exc(\hat{f}_{\vec{w}_i},P_i) &\leq E_{z\sim P_{\vec{w}_i}}[\ell(\hat{f}_{\vec{w}_i},z)]-\inf_{f\in\mathcal{F}}E_{z\sim P_{\vec{w}_i}}[\ell(f,z)]\\ &+2\beta\sigma^2+\frac{\sum_j \vec{w}_{i,j}D_{JS}(P_i||P_j)}{\beta}\\ &\leq E_{z\sim P_{\vec{w}_i}}[\ell(\hat{f}_{\vec{w}_i},z)]-\inf_{f\in\mathcal{F}}E_{z\sim P_{\vec{w}_i}}[\ell(f,z)]\\ &+2\sigma\sqrt{2\sum_{j=1}^{m}D_{JS}(P_i||P_j)}\end{aligned}$$

We identify the estimation error and we bound as previously done for Theorem 4 to obtain the final result. Moreover, for B-bounded random variables, $\sigma = B/2$

Appendix D

Appendix of Chapter 6

D.1 Proof of Lemma 3

Lemma. *With probability $1-\sigma$, with $\sigma \in (0,1)$, with respect to the random sampling of the data set $\mathcal{D}$, for all distributions $q(\theta)$ that are absolutely continuous with respect the prior $p(\theta)$, the following bound on the population risk of the ensemble model holds*

$$\mathbb{E}_{\tilde{\nu}(x)}[\mathcal{R}_t(q,x)] \leq \mathcal{J}_t^m(q) + \psi(\tilde{\nu}, n, m, \beta, p, \sigma), \tag{D.1}$$

where

$$\psi(\tilde{\nu}, n, m, \beta, p, \sigma) := \frac{1}{\beta}\left(\log \mathbb{E}_{\mathcal{D},p(\theta)}\left[e^{\beta \Delta_{m,n}}\right] - \log \sigma\right) \tag{D.2}$$

and

$$\Delta_{m,n} := \frac{1}{n}\sum_{x\in\mathcal{D}} \log_t \mathbb{E}_{j\sim U[1:m]} p(x|\theta_j) - \mathbb{E}_{\tilde{\nu}(x)}\left[\log_t \mathbb{E}_{j\sim U[1:m]} p(x|\theta_j)\right]. \tag{D.3}$$

Furthermore, the risk with respect to the ID measure $\nu(x)$ can be bounded as

$$\mathbb{E}_{\nu(x)}[\mathcal{R}_t(q,x)] \leq \frac{1}{1-\epsilon}\left(\mathcal{J}_t^m(q) + \psi(\tilde{\nu}, n, m, \beta, p, \sigma)\right) + \frac{\epsilon(C^{1-t}-1)}{(1-\epsilon)(1-t)}, \tag{D.4}$$

if the contamination ratio satisfies the inequality $\epsilon < 1$.

Proof: The proof follows in a manner similar to [3]. For a data set size n, and for an ensemble of models $\Theta = \{\theta\}_{i=1}^m$, we define the quantity

$$\Delta_{m,n}(\Theta, \mathcal{D}) := \frac{1}{n}\sum_{x\in\mathcal{D}} \log_t \mathbb{E}_{j\sim U[1:m]} p(x|\theta_j) - \frac{1}{n}\sum_{x\in\mathcal{D}} \mathbb{E}_{\tilde{\nu}(x)}\left[\log_t \mathbb{E}_{j\sim U[1:m]} p(x|\theta_j)\right]. \tag{D.5}$$

From the compression lemma [215], we have that for any distribution $q(\theta)$ which is absolutely continuous with respect to the prior $p(\theta)$, and for any $\beta < 0$, the following holds

$$\mathbb{E}_{q(\theta)^{\otimes m}}\left[\beta \Delta_{m,n}\right] \leq D_1(q(\theta)^{\otimes m} || p(\theta)^{\otimes m}) + \log \mathbb{E}_{p(\theta)^{\otimes m}}\left[e^{\beta \Delta_{m,n}}\right] \tag{D.6}$$

$$= m D_1(q(\theta)||p(\theta)) + \log \mathbb{E}_{p(\theta)^{\otimes m}} \left[e^{\beta \Delta_{m,n}} \right], \tag{D.7}$$

where we have used the simplified notation $\Delta_{m,n} = \Delta_{m,n}(\Theta, \mathcal{D})$, and the equality follows from the basic properties of the KL divergence.

A direct application of Markov's inequality is then used to bound the last term of (D.7) with high probability. Namely, with probability greater then $1 - \sigma$ with respect to the random drawn of the data set $\mathcal{D} \sim \tilde{\nu}(x)^{\otimes n}$, the following holds

$$\mathbb{E}_{p(\theta)^{\otimes m}} \left[e^{\Delta_{m,n}} \right] \leq \frac{\mathbb{E}_{\tilde{\nu}(x)^{\otimes n}, p(\theta)^{\otimes m}} \left[e^{\Delta_{m,n}} \right]}{\sigma}, \tag{D.8}$$

or, equivalently,

$$\log \mathbb{E}_{p(\theta)^{\otimes m}} \left[e^{\Delta_{m,n}} \right] \leq \log \mathbb{E}_{\tilde{\nu}(x)^{\otimes n}, p(\theta)^{\otimes m}} \left[e^{\Delta_{m,n}} \right] - \log \sigma. \tag{D.9}$$

Combining (D.7) with (D.9), the following upper bound on the predictive risk holds with probability $1 - \sigma$

$$\mathcal{R}_t(q) \leq \mathbb{E}_{\tilde{\nu}(x), q(\theta)^{\otimes m}} \left[-\log_t \mathbb{E}_{j \sim U[1:m]} p(x|\theta_j) \right] \tag{D.10}$$

$$\leq \mathbb{E}_{q(\theta)^{\otimes m}} \left[\frac{1}{n} \sum_{x \in \mathcal{D}} \log_t \mathbb{E}_{j \sim U[1:m]} p(x|\theta_j) \right] + \frac{m}{\beta} D_1(q(\theta)||p(\theta)) + \frac{\log \mathbb{E}_{\tilde{\nu}(x)^{\otimes n}} \mathbb{E}_{p(\theta)^{\otimes m}} \left[e^{\Delta_{m,n}} \right] - \log \sigma}{\beta}. \tag{D.11}$$

Finally, the result above can be translated to a guarantee with respect to the ID measure $\nu(x) = \frac{\tilde{\nu}(x)}{1-\epsilon} - \frac{\epsilon}{1-\epsilon} \xi(x)$ via the sequence of inequalities

$$\mathbb{E}_{\nu(x), q(\theta)^{\otimes m}} \left[-\log_t \mathbb{E}_{j \sim U[1:m]} p(x|\theta_j) \right] = \frac{\mathbb{E}_{\tilde{\nu}(x), q(\theta)^{\otimes m}} \left[-\log_t \mathbb{E}_{j \sim U[1:m]} p(x|\theta_j) \right]}{1 - \epsilon} + \epsilon \frac{\mathbb{E}_{\epsilon(x), q(\theta)^{\otimes m}} \left[-\log_t \mathbb{E}_{j \sim U[1:m]} p(x|\theta_j) \right]}{1 - \epsilon} \tag{D.12}$$

$$\leq \frac{\mathbb{E}_{\tilde{\nu}(x), q(\theta)^{\otimes m}} \left[-\log_t \mathbb{E}_{j \sim U[1:m]} p(x|\theta_j) \right]}{1 - \epsilon} + \epsilon \frac{\left(C^{1-t} - 1 \right)}{(1 - \epsilon)(1 - t)}, \tag{D.13}$$

where the last inequality follows by having assumed the probabilistic model being uniformly upper bounded by C (Assumption 2).

■

Finally, with regard to the comparison between the PAC^m bound in Theorem 1 in [3] and the guarantee with respect to the ID measure, we observe that it is not in general possible to translate a guarantee on the $\log_t$-risk to one on the log-risk. This can be illustrated by the following counter-example. Consider the following discrete target distribution parametrized by integer k, which defines the size of its support, as

$$\nu_k(x) = \begin{cases} 1 - \frac{1}{k}, & \text{for } x = 0 \\ \frac{1}{k} 2^{-k^2}, & \text{for } x = 1, \ldots, 2^{k^2}, \end{cases} \tag{D.14}$$

and the optimization of the $\log_t$-loss over a predictive distribution $p(x)$. The following limit holds

$$\lim_{k\to\infty} \min_{p} \mathbb{E}_{\nu_k(x)}[\log_t p(x)] = \begin{cases} 0, & \text{for } t \in [0,1) \\ \infty, & \text{for } t = 1 \end{cases}, \tag{D.15}$$

and therefore that an ensemble optimized for a value of t in the range $[0,1)$ can incur in an unboundedly large loss when scored using the log-loss.

D.2 Proof of Theorem 6

Theorem. *The minimizer of the robust m-free energy objective*

$$\mathcal{J}_t^m(q) := \hat{\mathcal{L}}_t^m(\theta, \mathcal{D}) + \frac{m}{\beta} D_1(q(\theta)||p(\theta)). \tag{D.16}$$

is the fixed point of the operator

$$T(q) := p(\theta_j) \exp\left(\beta \sum_{x\in\mathcal{D}} \mathbb{E}_{\{\theta_i\}_{i\neq j}}\left[\log_t\left(\frac{\sum_{i=1}^m p(x|\theta_i)}{m}\right)\right]\right) \tag{D.17}$$

where the average in (6.47) is taken with respect to the i.i.d. random vectors $\{\theta_i\}_{i\neq j} \sim q(\theta)^{\otimes m-1}$.

Proof: The functional derivative of the multi-sample risk is instrumental to computation of the minimizer of the robust m-free energy objective (6.41). This is given as

$$\frac{d\hat{\mathcal{R}}_t^m(q,x)}{dq} = \frac{d}{dq}\mathbb{E}_{\theta_1,\dots,\theta_m\sim q(\theta)^{\otimes m}}\left[-\log_t \mathbb{E}_{j\sim U[1:m]} p(x|\theta_j)\right] \tag{D.18}$$

$$= -\frac{d}{dq}\int_{\Theta^m} \log_t \mathbb{E}_{j\sim U[1:m]} p(x|\theta_j) \prod_{i=1}^m q(\theta_i) d\theta_i \tag{D.19}$$

$$\overset{(a)}{=} -\sum_{k=1}^m \int_{\Theta^{m-1}} \log_t \mathbb{E}_{j\sim U[1:m]} p(x|\theta_j) \prod_{i\neq k} q(\theta_i) d\theta_i \tag{D.20}$$

$$\overset{(b)}{=} -m \int_{\Theta^{m-1}} \log_t \mathbb{E}_{j\sim U[1:m]} p(x|\theta_j) \prod_{i=1}^{m-1} q(\theta_i) d\theta_i, \tag{D.21}$$

$$= -m\mathbb{E}_{\theta_1,\dots,\theta_{m-1}\sim q(\theta)^{\otimes m-1}}\left[\log_t \mathbb{E}_{j\sim U[1:m]} p(x|\theta_j)\right], \tag{D.22}$$

where (a) follows from the derivative of a nonlocal functional of m functions, and (b) holds since the integrand is invariant under the permutation of $\{\theta_i\}_{i\neq k}$.

The functional derivative of the robust m-free energy then follows as

$$\frac{d\mathcal{J}_t^m(q)}{dq} = \frac{d\hat{\mathcal{R}}_t^m(q,x)}{dq} + \frac{m}{\beta}\frac{dD_1(q(\theta)||p(\theta)}{dq} \tag{D.23}$$

$$= -m\mathbb{E}_{\theta_1,\ldots,\theta_{m-1}\sim q(\theta)^{\otimes m-1}}\left[\log_t \mathbb{E}_{j\sim U[1:m]} p(x|\theta_j)\right] + \frac{m}{\beta}\left(1 + \log(q(\theta)) - \log(p(\theta))\right). \tag{D.24}$$

Imposing the functional derivative equals to zero function it follows that the optimized posterior must satisfy

$$q(\theta_m) = p(\theta_m) \cdot \exp\left\{\beta \mathbb{E}_{\theta_1,\ldots,\theta_{m-1}\sim q(\theta)^{\otimes m-1}}\left[\log_t \mathbb{E}_{j\sim U[1:m]} p(x|\theta_j)\right]\right\}. \tag{D.25}$$

■

D.3 Proof of Theorem 7

Theorem. *The influence function of the robust m-free energy objective (6.51) is*

$$IF_t^m(z,\phi,P^n) = -\left[\frac{\partial^2 \mathcal{J}_t^m(\gamma,\phi)}{\partial\phi^2}\right]^{-1} \times \frac{\partial^2 \mathcal{J}_t^m(\gamma,\phi)}{\partial\gamma\partial\phi}\Bigg|_{\substack{\gamma=0 \\ \phi=\phi_t^{m*}(0)}}, \tag{D.26}$$

where

$$\frac{\partial^2 \mathcal{J}_t^m(\gamma,\phi)}{\partial\phi^2} = \mathbb{E}_{P^n_{\gamma,z}(x)} \frac{\partial^2}{\partial\phi^2}\left[\hat{\mathcal{R}}_t^m(q_\phi, x)\right] + \frac{\partial^2}{\partial\phi^2}\left[\frac{m}{\beta} KL(q_\phi(\theta)||p(\theta))\right] \tag{D.27}$$

and

$$\frac{\partial^2 \mathcal{J}_t^m(\gamma,\phi)}{\partial\gamma\partial\phi} = \frac{\partial}{\partial\phi}\left[\mathbb{E}_{P^n(x)}\left[\hat{\mathcal{R}}_t^m(q_\phi, x)\right] - \hat{\mathcal{R}}_t^m(q_\phi, z)\right]. \tag{D.28}$$

The proof of Theorem 7 directly follows from the Cauchy implicit function theorem stated below.

Theorem 8 (Cauchy implicit function theorem)**.** *Given a continuously differentiable function $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$, with domain coordinates (x, y), and a point $(x^*, y^*) \in \mathbb{R}^n \times \mathbb{R}^m$ such that $F(x^*, y^*) = 0$, if the Jacobian $J_{F,y}(x^*, y^*) = \left[\frac{\partial F_1(x^*,y^*)}{\partial y_1}, \ldots, \frac{\partial F_m(x^*,y^*)}{\partial y_m}\right]$ is invertible, then there exists an open set U that contains x^* and a function $g : U \to Y$ such that $g(x^*) = y^*$ and $F(x, g(x)) = 0$, $\forall x \in U$. Moreover the partial derivative of $g(x)$ in U are given by*

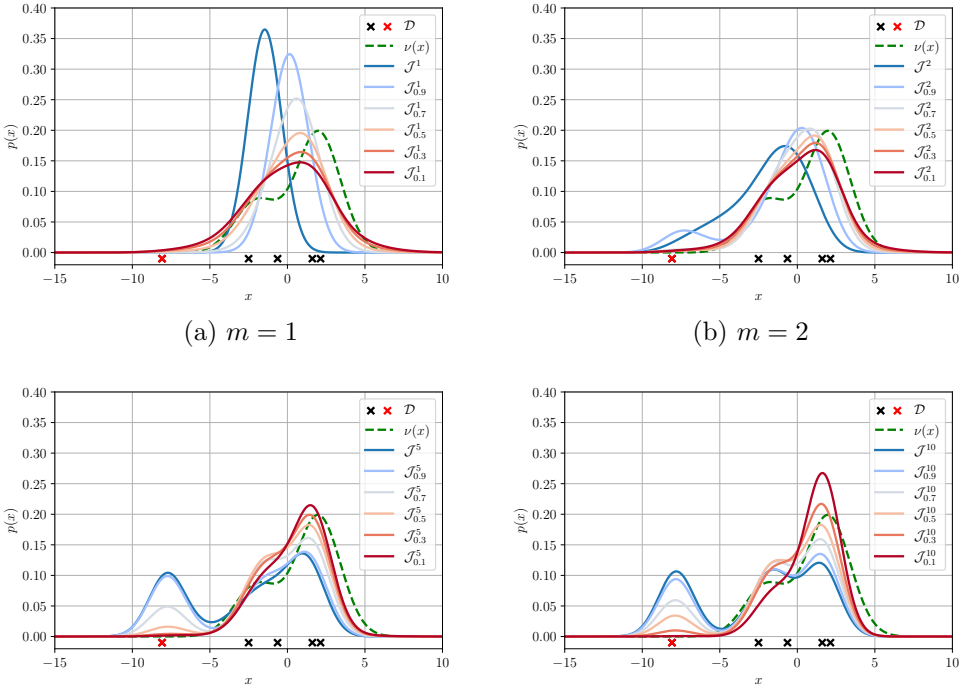
$$\frac{\partial g}{\partial x_i}(x) = -\left[J_{F,y}(x, g(x))\right]^{-1}\left[\frac{\partial F}{\partial x_i}(x, g(x))\right] \tag{D.29}$$

Proof: Replacing $F(x, y)$ with $\frac{\partial \mathcal{J}_t^m(\gamma,\phi)}{\partial\phi}$ and $g(x)$ with $\phi_t^{m*}(\gamma)$ and accordingly rewriting $(D.29)$, we obtain

$$\frac{d\phi_t^{m*}(\gamma)}{d\gamma} = -\left[\frac{\partial^2 \mathcal{J}_t^m(\gamma,\phi_t^{m*}(\gamma))}{\partial\phi^2}\right]^{-1} \times \frac{\partial^2 \mathcal{J}_t^m(\gamma,\phi_t^{m*}(\gamma))}{\partial\gamma\partial\phi}. \tag{D.30}$$

The influence function (D.26) is then obtained evaluating (D.30) at $\gamma = 0$.

■



(a) $m = 1$ (b) $m = 2$

(c) $m = 5$ (d) $m = 20$

Figure D.1: Ensemble predictive distribution obtained minimizing different free energy criteria and different values of m. The samples from the ID measure are represented as green dots, while data points sampled from the OOD component are in red. The optimized predictive distributions. The predictive distribution obtained minimizing the standard m-free energy is denoted by $\mathcal{J}^m$, while the predictive distribution yielded by the minimization of the robust m-free energy are denoted by $\mathcal{J}^m_{0.9}, \mathcal{J}^m_{0.7}, \mathcal{J}^m_{0.5}, \mathcal{J}^m_{0.3}$ and $\mathcal{J}^m_{0.1}$ for $t = \{1, 0.9, 0.7, 0.5, 0.3, 0.1\}$ respectively.

Table D.1: Total variation (TV) distance between the ID measure $\nu(x)$ and the predictive distribution $p_q(x)$ obtained from the optimization of the different free energy criteria.

	$t = 1$	$t = 0.9$	$t = 0.7$	$t = 0.5$	$t = 0.3$	$t = 0.1$
$m = 1$	0.59	0.42	0.27	0.18	**0.16**	0.18
$m = 2$	0.44	0.32	0.22	0.17	**0.15**	0.15
$m = 5$	0.34	0.32	0.23	0.18	0.15	**0.14**
$m = 10$	0.34	0.30	0.24	0.19	**0.15**	0.16

D.4 Simulation Details

D.5 Details on the Toy Example of Figure D.1

In the toy example of Figure D.1, the ID distribution $\nu(x)$ is a two component Gaussian mixture with means $\{-2, 2\}$, variance equal to 2, and mixing coefficients $\{0.3, 0.7\}$, respectively. The OOD distribution $\xi(x)$ is modelled using a Gaussian distribution with mean -8 and variance equal to 1.

The probabilistic model is a Gaussian unit variance $p(x|\theta) = \mathcal{N}(x|\theta, 1)$, the ensembling distribution $q(\theta)$ is represented by a discrete probability supported on 500 evenly spaced values in the interval $[-30, 30]$, and the prior is $p(\theta) = \mathcal{N}(\theta|0, 9)$. For a given m, β and t, the optimized ensembling distribution is obtained applying the fixed-point iteration in Theorem 6, i.e.,

$$q^+(\theta) = p(\theta) \exp\left\{\beta \sum_{\theta_1,\ldots,\theta_{m-1}} \prod_{i=1}^{m-1} q^t(\theta_i) \log_t \left(\frac{\sum_{j=1}^{m-1} p(x|\theta_j) + p(x|\theta)}{m}\right)\right\}, \tag{D.31}$$

$$q^{t+1}(\theta) = (1-\alpha)q^t(\theta) + \alpha \frac{q^+(\theta)}{\sum_\theta q^+(\theta)}, \tag{D.32}$$

for $\alpha \in (0, 1)$.

In Figure D.1 we report the optimized predictive distributions produced by the above procedure for $\beta = 1$, $m = \{1, 2, 5, 20\}$ and $t = \{1, 0.9, 0.7, 0.5, 0.3, 0.1\}$. As m grows larger, the multi-sample bound on the predictive risk becomes tighter. As a result, the predictive distribution becomes more expressive, and it covers all the data points. The use of generalized logarithms offers increased robustness against the outlier data point, and leads to predictive distributions that are more concentrated around the ID measure. In Table D.1 we report the total variation distance between the ID measure and the predictive distribution $p_q(x)$. The proposed robust m-free energy criterion consistently outperforms the standard criterion by halving the total variation distance form the ID measure for $t = 0.3$.

D.6 Details and Further Results for the Classification Example in Sec. 6.6.2

In Figure 6.6, we used expected calibration error (ECE) [1] to assess the quality of uncertainty quantification of the classifier. In this section, we formally define the ECE, along with the related visual tool of reliability diagrams [166], and present additional results using reliability diagrams.

Consider a probabilistic parametric classifier $p(b|a, \theta)$, where $b \in \{1, \ldots, C\}$ represents the label and a the covariate. The *confidence* level assigned by the model to the predicted label

$$\hat{b}(a) = \arg\max_b p(b|a, \theta) \tag{D.33}$$

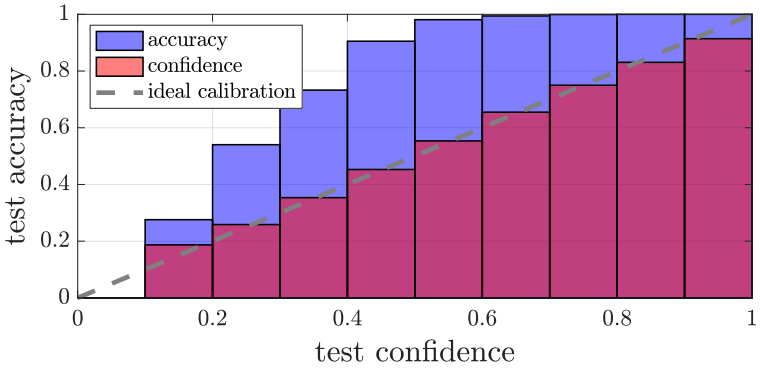


Figure D.2: Reliability diagram of deep ensembles [4].

given the covariate a is given as [1]

$$\hat{p}(a) = \max_b p(b|a, \theta). \tag{D.34}$$

Perfect calibration corresponds to the equality [1]

$$\mathbb{P}(\hat{b}(a) = b|\hat{p}(a) = p) = p, \ \forall p \in [0, 1], \tag{D.35}$$

where the probability is taken over the ID sampling distribution $\nu(a, b)$. This equality expresses the condition that the probability of a correct decision for inputs with confidence level p equals p for all $p \in [0, 1]$. In words, confidence equals accuracy.

The ECE and reliability diagram provide means to quantify the extent to which the perfect calibration condition (D.35) is satisfied. To start, the probability interval $[0, 1]$ is divided into K bins, with the k-th bin being interval $(\frac{k-1}{K}, \frac{k}{K}]$. Assume that we have access to test data from the ID distribution. Denote as $\mathcal{B}_k$ the set of data points (a, b) in such test set for which the confidence $\hat{p}(a)$ lies within the k-th bin, i.e., $\hat{p}(a) \in (\frac{k-1}{K}, \frac{k}{K}]$. The average accuracy of the predictions for data points in $\mathcal{B}_k$ is defined as

$$\text{acc}(\mathcal{B}_k) = \frac{1}{|\mathcal{B}_k|} \sum_{a \in \mathcal{B}_k} \mathbf{1}(\hat{b}(a) = b), \tag{D.36}$$

with $\mathbf{1}(\cdot)$ being indicator function, b being the label corresponding to a in the given data point (a, b), and $|\mathcal{B}_k|$ denoting the number of total samples in the k-th bin $\mathcal{B}_k$. Similarly, the average confidence of the predictions for covariates in $\mathcal{B}_k$ can be written as

$$\text{conf}(\mathcal{B}_k) = \frac{1}{|\mathcal{B}_k|} \sum_{a \in \mathcal{B}_k} \hat{p}(a). \tag{D.37}$$

Note that perfectly calibrated model $p(b|a, \theta)$ would have $\text{acc}(\mathcal{B}_k) = \text{conf}(\mathcal{B}_k)$ for all $k \in \{1, \ldots, K\}$ in the limit of a sufficiently large data set.

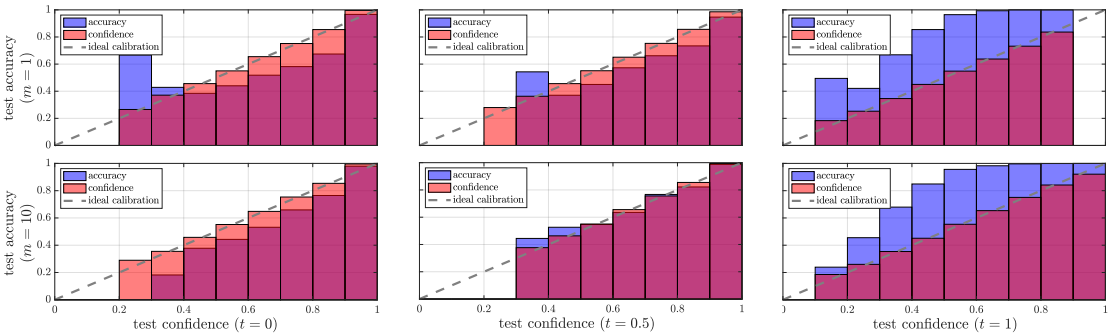


Figure D.3: Reliability diagrams of robust Gibbs predictor that optimizes $\mathcal{J}_t^1$ (top); and proposed robust ensemble predictor that optimizes $\mathcal{J}_t^{10}$ (bottom) under contamination ratio $\epsilon = 0.3$ for different $t = 0, 0.5, 1$.

D.6.1 Expected Calibration Error (ECE) [1]

ECE quantifies the amount of miscalibration by computing the weighted average of the differences between accuracy and confidence levels across the bins, i.e.,

$$\text{ECE} = \sum_{k=1}^{K} \frac{|\mathcal{B}_k|}{\sum_{k=1}^{K} |\mathcal{B}_k|} \Big| \text{acc}(\mathcal{B}_k) - \text{conf}(\mathcal{B}_k) \Big|. \tag{D.38}$$

D.6.2 Reliability Diagrams

Since the ECE quantifies uncertainty by taking an average over the bins, it cannot provide insights into the individual calibration performance per bin. In contrast, reliability diagrams plot the accuracy $\text{acc}(\mathcal{B}_k)$ versus the confidence $\text{conf}(\mathcal{B}_k)$ as a function of the bin index k, hence offering a finer-grained understanding of the calibration of the predictor.

D.6.3 Additional Results

For the MNIST image classification problem considered in Section 6.6.2, Figure D.2 plots for reference the reliability diagrams for deep ensembles [4], while Figure D.3 reports reliability diagrams for the proposed classifiers with different values of m and t. The figures illustrate that using the standard log-loss ($t = 1$) tends to yield poorly calibrated decisions (Figure D.2 and Figure D.3 (right)), while the proposed robust ensemble predictor can accurately quantify uncertainty using $t = 0.5$ (Figure D.3 (bottom, middle)). It is also noted that setting $t = 1$ is seen to yield underconfident predictions due to the presence of outliers, while a decrease in t leads to overconfident decision due to the reduced expressiveness of t-logarithms. A proper choice of t leads to well-calibrated, robust prediction.

Bibliography

[1] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1321–1330.

[2] A. X. Lu, A. X. Lu, W. Schormann, M. Ghassemi, D. W. Andrews, and A. M. Moses, "The cells out of sample (coos) dataset and benchmarks for measuring out-of-sample generalization of image classifiers," *arXiv preprint arXiv:1906.07282*, 2019.

[3] W. R. Morningstar, A. A. Alemi, and J. V. Dillon, "PAC^m-Bayes: narrowing the empirical risk gap in the misspecified Bayesian regime," *arXiv preprint arXiv:2010.09629*, 2020.

[4] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural information processing systems*, vol. 30, 2017.

[5] E. Amid, M. K. Warmuth, R. Anil, and T. Koren, "Robust bi-tempered logistic loss based on Bregman divergences," *arXiv preprint arXiv:1906.03361*, 2019.

[6] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International conference on engineering applications of neural networks*. Springer, 2016, pp. 213–226.

[7] W. Saad, M. Bennis, and M. Chen, "A vision of 6g wireless systems: Applications, trends, technologies, and open research problems," *IEEE network*, vol. 34, no. 3, pp. 134–142, 2019.

[8] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6g wireless communication systems: Applications, requirements, technologies, challenges, and research directions," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 957–975, 2020.

[9] T. Wang, S. Wang, and Z.-H. Zhou, "Machine learning for 5g and beyond: From model-based to data-driven mobile wireless networks," *China Communications*, vol. 16, no. 1, pp. 165–175, 2019.

[10] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, ai-based, or both?" *IEEE Transactions on Communications*, vol. 67, no. 10, pp. 7331–7376, 2019.

[11] J. Du, C. Jiang, J. Wang, Y. Ren, and M. Debbah, "Machine learning for 6g wireless networks: Carrying forward enhanced bandwidth, massive access, and ultrareliable/low-latency service," *IEEE Vehicular Technology Magazine*, vol. 15, no. 4, pp. 122–134, 2020.

[12] L. Lovén, T. Leppänen, E. Peltonen, J. Partala, E. Harjula, P. Porambage, M. Ylianttila, and J. Riekki, "Edgeai: A vision for distributed, edge-native artificial intelligence in future 6g networks," *The 1st 6G wireless summit*, pp. 1–2, 2019.

[13] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6g: Vision, enabling technologies, and applications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 5–36, 2021.

[14] B. G. Buchanan, "A (very) brief history of artificial intelligence," *Ai Magazine*, vol. 26, no. 4, pp. 53–53, 2005.

[15] H. Hellström, J. M. B. d. Silva Jr, V. Fodor, and C. Fischione, "Wireless for machine learning," *arXiv preprint arXiv:2008.13492*, 2020.

[16] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, "A survey on distributed machine learning," *Acm computing surveys (csur)*, vol. 53, no. 2, pp. 1–33, 2020.

[17] Z. Yang, M. Chen, K.-K. Wong, H. V. Poor, and S. Cui, "Federated learning for 6g: Applications, challenges, and opportunities," *Engineering*, 2021.

[18] M. Hasan, "State of IoT 2022: Number of connected IoT devices growing 18% to 14.4 billion globally," https://iot-analytics.com/number-connected-iot-devices/, 2022.

[19] Ericsoon, "Ericsson Mobility Report - June 2022," https://www.ericsson.com/49d3a0/assets/local/reports-papers/mobility-report/documents/2022/ericsson-mobility-report-june-2022.pdf, 2022.

[20] M. S. H. Abad, E. Ozfatura, D. Gunduz, and O. Ercetin, "Hierarchical federated learning across heterogeneous cellular networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8866–8870.

[21] S. Hosseinalipour, S. S. Azam, C. G. Brinton, N. Michelusi, V. Aggarwal, D. J. Love, and H. Dai, "Multi-stage hybrid federated learning over large-scale d2d-enabled fog networks," *IEEE/ACM Transactions on Networking*, 2022.

[22] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[23] M. N. Tehrani, M. Uysal, and H. Yanikomeroglu, "Device-to-device communication in 5g cellular networks: challenges, solutions, and future directions," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 86–92, 2014.

[24] H. Xing, O. Simeone, and S. Bi, "Decentralized federated learning via sgd over wireless d2d networks," in *IEEE 21st Inter. Workshop on Sig. Proc. Adv. in Wirel. Commun. (SPAWC)*, 2020.

[25] Y. Shi, Y. Zhou, and Y. Shi, "Over-the-air decentralized federated learning," *arXiv preprint arXiv:2106.08011*, 2021.

[26] C. Van Berkel, "Multi-core for mobile phones," in *2009 Design, Automation & Test in Europe Conference & Exhibition*. IEEE, 2009, pp. 1260–1265.

[27] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, B. McMahan *et al.*, "Towards federated learning at scale: System design," *Proceedings of Machine Learning and Systems*, vol. 1, pp. 374–388, 2019.

[28] X. Liu and N. Ansari, "Toward green iot: Energy solutions and key challenges," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 104–110, 2019.

[29] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[30] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," *arXiv preprint arXiv:0902.3430*, 2009.

[31] N. Kato, B. Mao, F. Tang, Y. Kawamoto, and J. Liu, "Ten challenges in advancing machine learning technologies toward 6G," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 96–103, 2020.

[32] Y. Liu, X. Liu, X. Mu, T. Hou, J. Xu, M. Di Renzo, and N. Al-Dhahir, "Reconfigurable intelligent surfaces: Principles and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1546–1577, 2021.

[33] R. Alghamdi, R. Alhadrami, D. Alhothali, H. Almorad, A. Faisal, S. Helal, R. Shalabi, R. Asfour, N. Hammad, A. Shams *et al.*, "Intelligent surfaces for 6g wireless networks: A survey of optimization and performance analysis techniques," *IEEE access*, 2020.

[34] W. Roh, J.-Y. Seol, J. Park, B. Lee, J. Lee, Y. Kim, J. Cho, K. Cheun, and F. Aryanfar, "Millimeter-wave beamforming as an enabling technology for 5g

cellular communications: Theoretical feasibility and prototype results," *IEEE communications magazine*, vol. 52, no. 2, pp. 106–113, 2014.

[35] S. Kutty and D. Sen, "Beamforming for millimeter wave communications: An inclusive survey," *IEEE communications surveys & tutorials*, vol. 18, no. 2, pp. 949–973, 2015.

[36] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv preprint arXiv:1409.2574*, 2014.

[37] A. Balatsoukas-Stimming and C. Studer, "Deep unfolding for communications systems: A survey and some new directions," in *2019 IEEE International Workshop on Signal Processing Systems (SiPS)*. IEEE, 2019, pp. 266–271.

[38] A. Musaddiq, Z. Nain, Y. Ahmad Qadri, R. Ali, and S. W. Kim, "Reinforcement learning-enabled cross-layer optimization for low-power and lossy networks under heterogeneous traffic patterns," *Sensors*, vol. 20, no. 15, p. 4158, 2020.

[39] J. Mei, X. Wang, K. Zheng, G. Boudreau, A. B. Sediq, and H. Abou-Zeid, "Intelligent radio access network slicing for service provisioning in 6g: A hierarchical deep reinforcement learning approach," *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 6063–6078, 2021.

[40] A. Klautau, N. González-Prelcic, and R. W. Heath, "Lidar data for deep learning-based mmwave beam-selection," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 909–912, 2019.

[41] M. Zecchin, M. B. Mashhadi, M. Jankowski, D. Gündüz, M. Kountouris, and D. Gesbert, "Lidar and position-aided mmwave beam selection with non-local cnns and curriculum training," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 2979–2990, 2022.

[42] K. S. V. Prasad, E. Hossain, and V. K. Bhargava, "Machine learning methods for rss-based user positioning in distributed massive mimo," *IEEE Transactions on Wireless Communications*, vol. 17, no. 12, pp. 8402–8417, 2018.

[43] C. J. Vaca-Rubio, P. Ramirez-Espinosa, K. Kansanen, Z.-H. Tan, E. De Carvalho, and P. Popovski, "Assessing wireless sensing potential with large intelligent surfaces," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 934–947, 2021.

[44] L. Fiorini, F. Cavallo, P. Dario, A. Eavis, and P. Caleb-Solly, "Unsupervised machine learning for developing personalised behaviour models using activity data," *Sensors*, vol. 17, no. 5, p. 1034, 2017.

[45] D. Goldenberg, K. Kofman, J. Albert, S. Mizrachi, A. Horowitz, and I. Teinemaa, "Personalization in practice: Methods and applications," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021, pp. 1123–1126.

[46] K. Beven and A. Binley, "The future of distributed models: model calibration and uncertainty prediction," *Hydrological processes*, vol. 6, no. 3, pp. 279–298, 1992.

[47] G. P. Fettweis, "The tactile internet: Applications and challenges," *IEEE Vehicular Technology Magazine*, vol. 9, no. 1, pp. 64–70, 2014.

[48] B. Wang and K. R. Liu, "Advances in cognitive radio networks: A survey," *IEEE Journal of selected topics in signal processing*, vol. 5, no. 1, pp. 5–23, 2010.

[49] P. H. Masur, J. H. Reed, and N. Tripathi, "Artificial intelligence in open-radio access network," *IEEE Aerospace and Electronic Systems Magazine*, 2022.

[50] J. Steinhardt, M. Charikar, and G. Valiant, "Resilience: A criterion for learning in the presence of arbitrary outliers," *arXiv preprint arXiv:1703.04940*, 2017.

[51] R. Martinez-Cantin, K. Tee, and M. McCourt, "Practical Bayesian optimization in the presence of outliers," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2018, pp. 1722–1731.

[52] A. R. Masegosa, "Learning under model misspecification: Applications to variational and ensemble methods," *arXiv preprint arXiv:1912.08335*, 2019.

[53] S. Theodoridis, *Machine learning: a Bayesian and optimization perspective*. Academic Press, 2015.

[54] D. Madigan, A. E. Raftery, C. Volinsky, and J. Hoeting, "Bayesian model averaging," in *Proceedings of the AAAI Workshop on Integrating Multiple Learned Models, Portland, OR*, 1996, pp. 77–83.

[55] S. Kalyani and K. Giridhar, "OFDM channel estimation in the presence of NBI and the effect of misspecified NBI model," in *2007 IEEE 8th Workshop on Signal Processing Advances in Wireless Communications*. IEEE, 2007, pp. 1–5.

[56] A. Fawzy, H. M. Mokhtar, and O. Hegazy, "Outliers detection and classification in wireless sensor networks," *Egyptian Informatics Journal*, vol. 14, no. 2, pp. 157–164, 2013.

[57] R. Jin, Z. Che, H. Xu, Z. Wang, and L. Wang, "An RSSI-based localization algorithm for outliers suppression in wireless sensor networks," *Wireless Networks*, vol. 21, no. 8, pp. 2561–2569, 2015.

[58] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[59] F. Yan, S. Sundaram, S. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2483–2493, 2012.

[60] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE transactions on automatic control*, vol. 31, no. 9, pp. 803–812, 1986.

[61] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized sgd with changing topology and local updates," in *Inter. Conf. on Machine Learning (ICML)*, 2020, pp. 5381–5393.

[62] E. Ozfatura, S. Rini, and D. Gündüz, "Decentralized SGD with over-the-air computation," in *IEEE Global Communications Conference*, 2020.

[63] H. Xing, O. Simeone, and S. Bi, "Federated learning over wireless device-to-device networks: Algorithms and convergence analysis," *arXiv preprint arXiv: 2101.12704v1*, 2021.

[64] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. on Signal Processing*, vol. 68, pp. 2155–2169, 2020.

[65] S. Dutta, J. Wang, and G. Joshi, "Slow and stale gradients can win the race," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 3, pp. 1012–1024, 2021.

[66] G. Nadiradze, A. Sabour, P. Davies, I. Markov, S. Li, and D. Alistarh, "Decentralized SGD with asynchronous, local and quantized updates," *arXiv preprint arXiv:1910.12308*, 2019.

[67] T. Adikari and S. Draper, "Decentralized optimization with non-identical sampling in presence of stragglers," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3702–3706.

[68] L. Xiao, S. Boyd, and S. Lall, "Distributed average consensus with time-varying Metropolis weights," *Automatica*, vol. 1, 2006.

[69] E. Jeong, M. Zecchin, and M. Kountouris, "Asynchronous decentralized learning over unreliable wireless networks," *arXiv preprint arXiv:2202.00955*, 2022.

[70] O. Esrafilian, R. Gangula, and D. Gesbert, "Autonomous UAV-aided mesh wireless networks," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 634–640.

[71] D. Behnke, K. Daniel, and C. Wietfeld, "Comparison of distributed ad-hoc network planning algorithms for autonomous flying robots," in *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*. IEEE, 2011, pp. 1–6.

[72] S. Sabino and A. Grilo, "Topology control of unmanned aerial vehicle UAV mesh networks: A multi-objective evolutionary algorithm approach," in *Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, 2018, pp. 45–50.

[73] I. Donevski, N. Babu, J. J. Nielsen, P. Popovski, and W. Saad, "Federated learning with a drone orchestrator: Path planning for minimized staleness," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 1000–1014, 2021.

[74] I. Mrad, L. Samara, A. A. Abdellatif, A. Al-Abbasi, R. Hamila, and A. Erbad, "Federated learning for UAV swarms under class imbalance and power consumption constraints," *arXiv preprint arXiv:2108.10748*, 2021.

[75] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[76] G. J. Lieberman and F. S. Hillier, *Introduction to operations research.* McGraw-Hill New York, NY, USA, 2005, vol. 8.

[77] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd innovations in theoretical computer science conference*, 2012, pp. 214–226.

[78] J. Quiñonero-Candela, M. Sugiyama, N. D. Lawrence, and A. Schwaighofer, *Dataset shift in machine learning.* Mit Press, 2009.

[79] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *International Conference on Machine Learning.* PMLR, 2019, pp. 4615–4625.

[80] P. M. Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations," *Mathematical Programming*, vol. 171, no. 1, pp. 115–166, 2018.

[81] J. C. Duchi, T. Hashimoto, and H. Namkoong, "Distributionally robust losses against mixture covariate shifts," *Under review*, 2019.

[82] J. Duchi and H. Namkoong, "Learning models with uniform performance via distributionally robust optimization," *arXiv preprint arXiv:1810.08750*, 2018.

[83] Y. Deng, M. M. Kamani, and M. Mahdavi, "Distributionally robust federated averaging," *arXiv preprint arXiv:2102.12660*, 2021.

[84] J. N. Tsitsiklis, "Problems in decentralized decision making and computation." Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, Tech. Rep., 1984.

[85] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, 2012.

[86] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[87] Q. Ling, Z. Wen, and W. Yin, "Decentralized jointly sparse optimization by reweighted ℓ_q minimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 5, pp. 1165–1170, 2012.

[88] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," *arXiv preprint arXiv:1705.09056*, 2017.

[89] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[90] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 5445–5450.

[91] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic control*, vol. 57, no. 3, pp. 592–606, 2011.

[92] O. Shamir and N. Srebro, "Distributed stochastic optimization and learning," in *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2014, pp. 850–857.

[93] M. Rabbat, "Multi-agent mirror descent for decentralized stochastic optimization," in *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2015, pp. 517–520.

[94] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified sgd with memory," *arXiv preprint arXiv:1809.07599*, 2018.

[95] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," *arXiv preprint arXiv:1704.05021*, 2017.

[96] D. Alistarh, T. Hoefler, M. Johansson, S. Khirirat, N. Konstantinov, and C. Renggli, "The convergence of sparsified gradient methods," *arXiv preprint arXiv:1809.10505*, 2018.

[97] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," *Advances in Neural Information Processing Systems*, vol. 30, pp. 1709–1720, 2017.

[98] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signsgd: Compressed optimisation for non-convex problems," in *International Conference on Machine Learning*. PMLR, 2018, pp. 560–569.

[99] A. Koloskova, S. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3478–3487.

[100] S. U. Stich, "Local sgd converges fast and communicates little," *arXiv preprint arXiv:1805.09767*, 2018.

[101] H. Yu, S. Yang, and S. Zhu, "Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 5693–5700.

[102] H. E. Scarf, "A min-max solution of an inventory problem," RAND CORP SANTA MONICA CALIF, Tech. Rep., 1957.

[103] H. Namkoong and J. C. Duchi, "Stochastic gradient methods for distributionally robust optimization with f-divergences." in *NIPS*, vol. 29, 2016, pp. 2208–2216.

[104] Z. Hu and L. J. Hong, "Kullback-leibler divergence constrained distributionally robust optimization," *Available at Optimization Online*, 2013.

[105] D. Wozabal, "A framework for optimization under ambiguity," *Annals of Operations Research*, vol. 193, no. 1, pp. 21–47, 2012.

[106] R. Jiang and Y. Guan, "Data-driven chance constrained stochastic program," *Mathematical Programming*, vol. 158, no. 1, pp. 291–327, 2016.

[107] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5149–5164, 2015.

[108] D. Mateos-Núnez and J. Cortés, "Distributed subgradient methods for saddle-point problems," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 5462–5467.

[109] I. Tsaknakis, M. Hong, and S. Liu, "Decentralized min-max optimization: Formulations, algorithms and applications in network poisoning attack," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5755–5759.

[110] W. Liu, A. Mokhtari, A. Ozdaglar, S. Pattathil, Z. Shen, and N. Zheng, "A decentralized proximal point-type method for saddle point problems," *arXiv preprint arXiv:1910.14380*, 2019.

[111] M. Liu, W. Zhang, Y. Mroueh, X. Cui, J. Ross, T. Yang, and P. Das, "A decentralized parallel algorithm for training generative adversarial nets," *arXiv preprint arXiv:1910.12999*, 2019.

[112] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression," *arXiv preprint arXiv:1907.09356*, 2019.

[113] P. Sharma, R. Panda, G. Joshi, and P. K. Varshney, "Federated minimax optimization: Improved convergence analyses and algorithms," *arXiv preprint arXiv:2203.04850*, 2022.

[114] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[115] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[116] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[117] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, 2018.

[118] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1126–1135.

[119] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3557–3568, 2020.

[120] Y. Jiang, J. Konečnỳ, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," *arXiv preprint arXiv:1909.12488*, 2019.

[121] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.

[122] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," *arXiv preprint arXiv:2003.13461*, 2020.

[123] F. Hanzely and P. Richtárik, "Federated learning of a mixture of global and local models," *arXiv preprint arXiv:2002.05516*, 2020.

[124] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.

[125] M. Reisser, C. Louizos, E. Gavves, and M. Welling, "Federated mixture of experts," *arXiv preprint arXiv:2107.06724*, 2021.

[126] O. Marfoq, G. Neglia, A. Bellet, L. Kameni, and R. Vidal, "Federated multi-task learning under a mixture of distributions," *International Workshop on Federated Learning for User Privacy and Data Confidentiality in conjunction with ICML 2021 (FL-ICML'21)*, 2021.

[127] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez, "Personalized federated learning with first order model optimization," *arXiv preprint arXiv:2012.08565*, 2020.

[128] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–9.

[129] M. Mestoukirdi, M. Zecchin, D. Gesbert, Q. Li, and N. Gresset, "User-centric federated learning: Trading off wireless resources for personalization," To be submitted to: *IEEE Transactions on Wireless Communications*, 2021.

[130] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1, pp. 151–175, 2010.

[131] C. Shui, Q. Chen, J. Wen, F. Zhou, C. Gagné, and B. Wang, "Beyond h-divergence: Domain adaptation theory with jensen-shannon divergence," 2020.

[132] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: Extending mnist to handwritten letters," in *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 2921–2926.

[133] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Toronto, Ontario, Tech. Rep. 0, 2009.

[134] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *arXiv preprint arXiv:2007.07481*, 2020.

[135] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[136] S. G. Walker, "Bayesian inference with misspecified models," *Journal of Statistical Planning and Inference*, vol. 143, no. 10, pp. 1621–1633, 2013.

[137] P. Grünwald and T. Van Ommen, "Inconsistency of Bayesian inference for misspecified linear models, and a proposal for repairing it," *Bayesian Analysis*, vol. 12, no. 4, pp. 1069–1103, 2017.

[138] T. Sypherd, M. Diaz, J. K. Cava, G. Dasarathy, P. Kairouz, and L. Sankar, "A loss function for robust classification: Calibration, landscape, and generalization," *arXiv preprint arXiv:1906.02314*, 2019.

[139] J. Knoblauch, J. Jewson, and T. Damoulas, "Generalized variational inference: Three arguments for deriving new posteriors," *arXiv preprint arXiv:1904.02063*, 2019.

[140] O. Simeone, *Machine Learning for Engineers*. Cambridge University Press, 2022.

[141] J. Jewson, J. Q. Smith, and C. Holmes, "Principles of Bayesian inference using general divergence criteria," *Entropy*, vol. 20, no. 6, p. 442, 2018.

[142] A. Basu, I. R. Harris, N. L. Hjort, and M. Jones, "Robust and efficient estimation by minimising a density power divergence," *Biometrika*, vol. 85, no. 3, pp. 549–559, 1998.

[143] A. Ghosh and A. Basu, "Robust Bayes estimation using the density power divergence," *Annals of the Institute of Statistical Mathematics*, vol. 68, no. 2, pp. 413–437, 2016.

[144] H. Fujisawa and S. Eguchi, "Robust parameter estimation with a small bias against heavy contamination," *Journal of Multivariate Analysis*, vol. 99, no. 9, pp. 2053–2081, 2008.

[145] T. Nakagawa and S. Hashimoto, "Robust Bayesian inference via γ-divergence," *Communications in Statistics-Theory and Methods*, vol. 49, no. 2, pp. 343–360, 2020.

[146] F. Futami, I. Sato, and M. Sugiyama, "Variational inference based on robust divergences," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2018, pp. 813–822.

[147] F. R. Hampel, "The influence curve and its role in robust estimation," *Journal of the American Statistical Association*, vol. 69, no. 346, pp. 383–393, 1974.

[148] E. Amid and M. K. Warmuth, "A more globally accurate dimensionality reduction method using triplets." *arXiv preprint arXiv:1803.00854*, 2018.

[149] C. Tsallis, "Possible generalization of Boltzmann-Gibbs statistics," *Journal of statistical physics*, vol. 52, no. 1, pp. 479–487, 1988.

[150] T. Sears *et al.*, "Generalized maximum entropy, convexity and machine learning," Ph.D. dissertation, Australian National University, 2008.

[151] S. Umarov, C. Tsallis, and S. Steinberg, "On a q-central limit theorem consistent with nonextensive statistical mechanics," *Milan Journal of Mathematics*, vol. 76, no. 1, pp. 307–328, 2008.

[152] K. M. Cohen, S. Park, O. Simeone, and S. Shamai, "Learning to learn to demodulate with uncertainty quantification via Bayesian meta-learning," *arXiv preprint arXiv:2108.00785*, 2021.

[153] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[154] Z. Xiao, H. Wen, A. Markham, N. Trigoni, P. Blunsom, and J. Frolik, "Identification and mitigation of non-line-of-sight conditions using received signal strength," in *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2013, pp. 667–674.

[155] O. Catoni, "PAC-bayesian supervised classification: the thermodynamics of statistical learning," *arXiv preprint arXiv:0712.0248*, 2007.

[156] A. Masegosa, "Learning under model misspecification: Applications to variational and ensemble methods," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5479–5491, 2020.

[157] Y. Burda, R. B. Grosse, and R. Salakhutdinov, "Importance weighted autoencoders," in *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[158] A. Mnih and D. Rezende, "Variational inference for Monte Carlo objectives," in *International Conference on Machine Learning*. PMLR, 2016, pp. 2188–2196.

[159] P. J. Huber, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964. [Online]. Available: http://www.jstor.org/stable/2238020

[160] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[161] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[162] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[163] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1613–1622.

[164] Y. LeCun, "The MNIST database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[165] R. Zhang, Y. Li, C. De Sa, S. Devlin, and C. Zhang, "Meta-learning divergences for variational inference," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 4024–4032.

[166] M. H. DeGroot and S. E. Fienberg, "The comparison and evaluation of forecasters," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 32, no. 1-2, pp. 12–22, 1983.

[167] M. Zecchin, S. Park, O. Simeone, M. Kountouris, and D. Gesbert, "Robust bayesian learning for reliable wireless ai: Framework and applications," *arXiv preprint arXiv:2207.00300*, 2022.

[168] S. Liu, R. Cao, Y. Huang, T. Ouypornkochagorn, and J. Jia, "Time sequence learning for electrical impedance tomography using bayesian spatiotemporal priors," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 6045–6057, 2020.

[169] A. Onan, S. Korukoğlu, and H. Bulut, "A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification," *Information Processing & Management*, vol. 53, no. 4, pp. 814–833, 2017.

[170] A. Onan, "Biomedical text categorization based on ensemble pruning and optimized topic modelling," *Computational and Mathematical Methods in Medicine*, vol. 2018, 2018.

[171] J. Yoon, T. Kim, O. Dia, S. Kim, Y. Bengio, and S. Ahn, "Bayesian model-agnostic meta-learning," *Advances in neural information processing systems*, vol. 31, 2018.

[172] S. T. Jose, S. Park, and O. Simeone, "Information-theoretic analysis of epistemic uncertainty in bayesian meta-learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 9758–9775.

[173] O. Simeone, "A very brief introduction to machine learning with applications to communication systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648–664, 2018.

[174] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3072–3108, 2019.

[175] D. Gündüz, P. de Kerret, N. D. Sidiropoulos, D. Gesbert, C. R. Murthy, and M. van der Schaar, "Machine learning in the air," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2184–2199, 2019.

[176] Y. Jiang, H. Kim, H. Asnani, S. Kannan, S. Oh, and P. Viswanath, "Learn codes: Inventing low-latency codes via recurrent neural networks," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 207–216, 2020.

[177] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, "Deepcode: Feedback codes via deep learning," *Advances in neural information processing systems*, vol. 31, 2018.

[178] S. Park, O. Simeone, and J. Kang, "Meta-learning to communicate: Fast end-to-end training for fading channels," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5075–5079.

[179] S. Park, H. Jang, O. Simeone, and J. Kang, "Learning to demodulate from few pilots via offline and online meta-learning," *IEEE Transactions on Signal Processing*, vol. 69, pp. 226–239, 2020.

[180] O. Simeone, S. Park, and J. Kang, "From learning to meta-learning: Reduced training overhead and complexity for communication systems," in *2020 2nd 6G Wireless Summit (6G SUMMIT)*. IEEE, 2020, pp. 1–5.

[181] Y. Yuan, G. Zheng, K.-K. Wong, B. Ottersten, and Z.-Q. Luo, "Transfer learning and meta learning-based fast downlink beamforming adaptation," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1742–1755, 2020.

[182] P. H. Masur and J. H. Reed, "Artificial intelligence in Open Radio Access Network," *arXiv preprint arXiv:2104.09445*, 2021.

[183] D. J. C. MacKay, *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

[184] K. Osawa, S. Swaroop, M. E. E. Khan, A. Jain, R. Eschenhagen, R. E. Turner, and R. Yokota, "Practical deep learning with Bayesian principles," *Advances in neural information processing systems*, vol. 32, 2019.

[185] I. Nikoloska and O. Simeone, "BAMLD: Bayesian Active Meta-Learning by Disagreement," *arXiv preprint arXiv:2110.09943*, 2021.

[186] O. Simeone, *Machine Learning for Engineers*. Cambridge university press, 2022.

[187] N. Zilberstein, C. Dick, R. Doost-Mohammady, A. Sabharwal, and S. Segarra, "Annealed Langevin dynamics for massive mimo detection," *arXiv preprint arXiv:2205.05776*, 2022.

[188] S. T. Jose and O. Simeone, "Free energy minimization: A unified framework for modeling, inference, learning, and optimization [lecture notes]," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 120–125, 2021.

[189] O. Catoni, "A PAC-Bayesian approach to adaptive classification," *preprint*, vol. 840, 2003.

[190] P. Alquier, "User-friendly introduction to PAC-Bayes bounds," *arXiv preprint arXiv:2110.11216*, 2021.

[191] Y.-C. Liang, K.-C. Chen, G. Y. Li, and P. Mahonen, "Cognitive radio networking and communications: An overview," *IEEE transactions on vehicular technology*, vol. 60, no. 7, pp. 3386–3407, 2011.

[192] E. S. Lohan, J. Torres-Sospedra, H. Leppäkoski, P. Richter, Z. Peng, and J. Huerta, "Wi-Fi crowdsourced fingerprinting dataset for indoor positioning," *Data*, vol. 2, no. 4, p. 32, 2017.

[193] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[194] R. Zhou, F. Liu, and C. W. Gravelle, "Deep learning for modulation recognition: A survey with a demonstration," *IEEE Access*, vol. 8, pp. 67 366–67 376, 2020.

[195] R. Mautz, "Overview of current indoor positioning systems," *Geodezija ir kartografija*, vol. 35, no. 1, pp. 18–22, 2009.

[196] M. Aernouts, R. Berkvens, K. Van Vlaenderen, and M. Weyn, "Sigfox and LoRaWAN datasets for fingerprint localization in large urban and rural areas," *Data*, vol. 3, no. 2, p. 13, 2018.

[197] G. Pecoraro, S. Di Domenico, E. Cianca, and M. De Sanctis, "CSI-based fingerprinting for indoor localization using LTE signals," *EURASIP Journal on Advances in Signal Processing*, vol. 2018, no. 1, pp. 1–18, 2018.

[198] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent neural networks for accurate RSSI indoor localization," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 639–10 651, 2019.

[199] R. S. Sinha and S.-H. Hwang, "Comparison of CNN applications for RSSI-based fingerprint indoor localization," *Electronics*, vol. 8, no. 9, p. 989, 2019.

[200] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[201] X. Song, X. Fan, C. Xiang, Q. Ye, L. Liu, Z. Wang, X. He, N. Yang, and G. Fang, "A novel convolutional neural network based indoor localization framework with WiFi fingerprinting," *IEEE Access*, vol. 7, pp. 110 698–110 709, 2019.

[202] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and J. Huerta, "Ujiindoorloc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems," in *2014 international conference on indoor positioning and indoor navigation (IPIN)*. IEEE, 2014, pp. 261–270.

[203] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[204] F. A. Aoudia and J. Hoydis, "End-to-end learning of communications systems without a channel model," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 298–303.

[205] H. Ye, L. Liang, G. Y. Li, and B.-H. Juang, "Deep learning-based end-to-end wireless communication systems with conditional GANs as unknown channels," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3133–3143, 2020.

[206] T. J. O'Shea, T. Roy, and N. West, "Approximating the void: Learning stochastic channel models from observation with variational generative adversarial networks,"

in *2019 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2019, pp. 681–686.

[207] T. Orekondy, A. Behboodi, and J. B. Soriaga, "MIMO-GAN: Generative MIMO channel modeling," *arXiv preprint arXiv:2203.08588*, 2022.

[208] Y. Yang, Y. Li, W. Zhang, F. Qin, P. Zhu, and C.-X. Wang, "Generative-adversarial-network-based wireless channel modeling: Challenges and opportunities," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 22–27, 2019.

[209] M. Ibnkahla, "Applications of neural networks to digital communications–a survey," *Signal processing*, vol. 80, no. 7, pp. 1185–1215, 2000.

[210] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel method for the two-sample-problem," *Advances in neural information processing systems*, vol. 19, 2006.

[211] E. Daxberger and J. M. Hernández-Lobato, "Bayesian variational autoencoders for unsupervised out-of-distribution detection," *arXiv preprint arXiv:1912.05651*, 2019.

[212] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[213] D. 3rd Generation Partnership Project (3GPP), "Study on channel model for frequencies from 0.5 to 100 GHz," *3GPP TR 38.901*, 2020.

[214] T. Lin, C. Jin, and M. Jordan, "On gradient descent ascent for nonconvex-concave minimax problems," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6083–6093.

[215] A. Banerjee, "On Bayesian bounds," in *Proceedings of the 23rd International Conference on Machine learning*, 2006, pp. 81–88.