Online Distributed Learning : A Projection-Free Approach

Tuan-Anh Nguyen Huu

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique
Spécialité : Mathématiques et Informatique
Unité de recherche : Laboratoire d'Informatique de Grenoble

Apprentissage Distribué en Ligne : Une Approche Sans Projection

Online Distributed Learning : A Projection-Free Approach

Présentée par :

Tuan-Anh NGUYEN HUU

Direction de thèse :

Denis TRYSTRAM PROFESSEUR DES UNIVERSITES, GRENOBLE INP - UGA	Directeur de thèse
Kim Thang NGUYEN PROFESSEUR DES UNIVERSITES, GRENOBLE INP - UGA	Co-directeur de thèse

Rapporteurs :

AYMERIC DIEULEVEUT
PROFESSEUR, ECOLE POLYTECHNIQUE
JEAN-MARC NICOD
PROFESSEUR DES UNIVERSITES, ECOLE NAT SUP MECANIQUE MICROTECHNIQUES

Thèse soutenue publiquement le **23 octobre 2024**, devant le jury composé de :

ADELINE LECLERCQ-SAMSON, PROFESSEURE DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES	Présidente
DENIS TRYSTRAM, PROFESSEUR DES UNIVERSITES, GRENOBLE INP - UGA	Directeur de thèse
KIM THANG NGUYEN, PROFESSEUR DES UNIVERSITES, GRENOBLE INP - UGA	Co-directeur de thèse
AYMERIC DIEULEVEUT, PROFESSEUR, ECOLE POLYTECHNIQUE	Rapporteur
AURELIEN BELLET, DIRECTEUR DE RECHERCHE, ANTENNE INRIA UNIVERSITE DE MONTPELLIER	Examinateur
SONIA BEN MOKHTAR, DIRECTRICE DE RECHERCHE, CNRS DELEGATION RHONE AUVERGNE	Examinatrice

Acknowledgements

Time flies, especially when the journey is filled with joy and excitement. As I conclude this meaningful chapter of my life, I find myself deeply grateful to all those who have been part of this journey and made it so memorable. It is with heartfelt appreciation that I express my gratitude to everyone who supported me throughout my PhD, both professionally and personally.

First and foremost, I extend my deepest thanks to my supervisors, Denis Trystram and Nguyen Kim Thang, for their invaluable guidance, encouragement, and for fostering an inspiring and supportive research environment over the past three years. Their mentorship has been pivotal to my growth as a researcher, and I am profoundly grateful for their dedication and expertise.

I would also like to sincerely thank the members of my thesis jury—Aurélien Bellet, Aymeric Dieuleveut, Jean-Marc Nicod, Sonia Ben-Mokhtar, and Adeline Samson—for taking the time to review my manuscript, provide thoughtful feedback, and participate in my thesis defense. Their expertise and constructive suggestions have been invaluable in refining my work.

My sincere thanks go to the MIAI Institute for funding my PhD research and for offering opportunities to collaborate with leading researchers in the field of machine learning. The institute's financial support, resources, and networking opportunities have played a critical role in the success of my academic journey.

I am deeply grateful to the DATAMOVE team, whose insightful discussions and collaborative spirit greatly enriched my research experience. I am equally thankful to my fellow office-mates in IMAG 468—Paul, Eniko, Mathilde, Louis, Yoann, and Yannick—for their friendship, support, and for creating a vibrant and welcoming environment. The camaraderie and shared moments of laughter made even the most challenging days enjoyable. A special mention goes to my Vietnamese friends in Grenoble, who turned my time away from home into a truly fulfilling experience. Their unwavering companionship, from my first day in the city to my last, was a source of immense comfort and joy.

Lastly, I owe everything to my family and close friends, whose unconditional love and support have been the foundation of my strength throughout this journey. This thesis is as much a testament to their belief in me as it is to my own efforts.

Completing this PhD has been a transformative journey—one that has shaped me both scientifically and personally. It has been an honor to learn from brilliant minds, collaborate with incredible individuals, and grow through challenges. I will carry the lessons and memories of this journey with me as I embark on the next phase of my career.

Abstract

Distributed learning has been studied intensively in recent years due to its practicality for a wide range of applications where data transfer incurs high costs in terms of privacy and communication bandwidth. In this context, it is crucial to design algorithms that are suitable for edge devices with limited computational and communication capabilities, while still achieving optimal performance in a distributed setting. However, this is a challenging task as the algorithm's performance is dependent on multiple factors such as the overlay communication network, the computational capabilities, and the nature of the data on each device. The majority of research in distributed learning has focused on the offline setting, where data is stored locally and the objective function remains static throughout the learning process. However, this offline setting becomes unrealistic for many machine learning applications as data is generated in a continuous manner. In this thesis, we study the problem of distributed online learning, where multiple agents learn from streams of data generated at local devices to reach a consensus on a global objective function. We propose projection-free algorithms that are well-suited for a distributed setting. These algorithms are carefully designed to achieve optimal regret bounds for various scenarios of online and distributed learning, including delayed feedback, zeroth-order feedback for convex and non-convex functions. We conduct an extensive theoretical study and experimentally validate the performance of our algorithms by comparing them with existing ones on real-world problems. Furthermore, we provide an empirical study on the energy consumption of training federated learning (FL) on edge devices, taking into account data heterogeneity and the computation/communication trade-off when varying the number of devices and data partition.

L'apprentissage distribué a fait l'objet de recherches intensives ces dernières années en raison de sa praticité pour une large gamme d'applications où le transfert de données entraîne des coûts élevés en termes de confidentialité et de bande passante de communication. Dans ce contexte, il est crucial de concevoir des algorithmes qui conviennent aux périphériques de bord avec des capacités de calcul et de communication limitées, tout en atteignant des performances optimales dans un environnement distribué. Cependant, c'est une tâche difficile car les performances de l'algorithme dépendent de plusieurs facteurs tels que le réseau de communication, les capacités de calcul et la nature des données sur chaque périphérique. La majorité des recherches en apprentissage distribué ont porté sur le paramètre hors ligne, où les données sont stockées localement et la fonction objective reste statique tout au long du processus d'apprentissage. Cependant, ce paramètre hors ligne devient irréaliste pour de nombreuses applications d'apprentissage automatique car les données sont générées de manière continues. Dans cette thèse, nous étudions le problème de l'apprentissage en ligne distribué, où plusieurs agents apprennent à partir de flux de données générés sur des périphériques locaux pour atteindre un consensus sur une fonction objective globale. Nous proposons des algorithmes sans projection qui sont bien adaptés à un environnement distribué. Ces algorithmes sont conçus avec soin pour atteindre des bornes de regret optimales pour divers scénarios d'apprentissage en ligne et distribué, y compris le delai de feedback, le bandit feedback pour les fonctions convexes et non convexes. Nous menons une étude théorique approfondie et validons expérimentalement les performances de nos algorithmes en les comparant à des algorithmes existants sur des problèmes du monde réel. De plus, nous fournissons une étude empirique sur la consommation d'énergie de l'apprentissage fédéré sur des périphériques de bord, en tenant compte de l'hétérogénéité des données et du compromis entre calcul et communication lors de la variation du nombre de périphériques et de la partition des données.

Contents

1 Introduction

Contents

1.1 Introduction

Over the past decade, there has been a significant surge in the growth of machine learning and AI applications. From image recognition to natural language processing, and now the proliferation of generative AI, this growth has been fueled by advancements in computational power, research breakthroughs, and most importantly, the ever-increasing volume of data. Data sourced from various origins has emerged as valuable asset for training machine learning models, but it also presents a major challenge for storage and processing. Typically, a machine learning application gathers data from diverse sources, consolidates in a central database for processing and training, and subsequently delivers it to end-users for inference. However, data collection and storage in a centralized manner raises significant concerns about privacy as it may contain sensitive information such as personal, financial, or medical data. Furthermore, data collection and storage create a significant cost in terms of infrastructure and maintenance. To address these challenges, it is imperative to develop machine learning techniques that can address the following constraints:

- *"Data: How to train machine learning models without exposing sensitive data to third parties?"*
- *"Resources: How can we reduce the cost of data storage and processing?"*

The resolution to these challenges can be found within the framework of federated learning (FL) [Kairouz2021], where multiple clients collaboratively train a machine learning model under the coordination of a central server without disclosing the raw data. Essentially, clients train the model on their individual datasets and send their model updates to the server for aggregation. This method effectively addresses privacy concerns by keeping the data at sources. However, scalability issues arise with a large number of clients since the communication between clients and server can become a bottlenect, especially in slow or unreliable network conditions. Consequently, decentralized learning (DL) has emerged as an alternative to FL, eliminating the need for a central server. In DL, clients communicate directly with each other in a peer-to-peer fashion to exchange model parameters and updates on their local datasets, ensuring global convergence of the objective function without central server. A few significant works in this area are Decentralized Parallel Stochastic Gradient Descent (D-PSGD) algorithm and its variants [Tang2018,Nedic2009,Ram2010, Lian2017, Duchi2012, Yuan2016], adopts a decentralized approach to stochastic gradient descent algorithm, treating a weighted sum of local objectives as the global objective. Clients are expected to have an overlay communication network to facilitate communication with neighboring clients. In each communication round, clients update their local models using locally stored data and exchange model parameters with their immediate neighbors in the communication graph. This process continues until global model achieves convergence. Accordingly, decentralized learning encompasses two crucial aspects : 1) *locally stored data on the client side*; 2) *a local optimization process*;

The first aspect raises numerous challenges for today's distributed learning systems. The learning process is conducted on user-end devices such as smartphones, IoT devices, or edge servers, which have limited storage capacity. As data is continuously generated on these devices, requiring data storage might not be a pratical choice for an efficient learning system. In this context, the concept of online learning is a natural consideration to not only alleviate storage constraint but also to adapt to the dynamic nature of the data. Regarding the second aspect of the optimization process, Stochastic Gradient Descent (SGD) and its variants are the preferred algorithms in many machine learning tasks. However, they might not be the optimal choice for for constrained optimization environments, such as decentralized learning, when it comes to small edge and IoT devices. This is because the additional operation, such as the projection step, can introduce substantial computational overhead and potentially hinder the learning process. In such circumstances, the Frank-Wolfe algorithm, also known as conditional gradient, emerges as a more attractive alternative to the family of projected gradient descent. Its projection-free nature helps reduce the computational costs, making it a more suitable choice for resource-constrained devices, such as edge and IoT devices.

In this thesis, we address the challenges of learning in a distributed and dynamic environment by approaching it through the framework of online optimization. We aim to develop algorithms solving constrained optimization problem that is suitable for distributed setting where the participant in the learning process are primarily edge and IoT devices with limited storage capacity and

computational power. To this end, we focus on algorithms that belong to the Frank-Wolfe family, which are projection-free by nature. This feature reduces the computational cost of the optimization process while ensuring competitive performance compared to projection-based algorithms.

In the following sections, we will provide a formal description of the problem and review the concept of online optimization in both centralized and distributed settings, using Online Gradient Descent as the illustration algorithm. We will then introduce the vanilla Frank-Wolfe algorithm and its online variant. Additionally, we will give a brief review of related work in the field of online optimization. We will present our contributions and the outline of the thesis, along with some common lemmas and notations that will be used throughout the thesis.

1.2 Notations and Preliminaries

We use boldface lower case and upper case letter to denote vector and matrix e.g $\boldsymbol{x}$ and $\mathbf{X}$, respectively . Given an undirected graph $\mathbb{G} = (\mathcal{V}, \mathcal{E})$, we let $|\mathcal{V}| = n$ the total number of nodes.By abuse of notation, we denote the agents by indice $i \in [n]$ where $[n] = [1, \dots, n]$. The set of neighbors of an agent $i \in [n]$ is denoted $\mathcal{N}(i) := \{j \in [n] : (i,j) \in E\}$. Consider a symmetric matrix $\mathbf{W} \in \mathbb{R}_+^{n \times n}$. We call by $\tau^i = |\mathcal{N}(i)|$, the degree of of vertex i, the entries w_{ij} defined as follows.

$$w_{ij} = \begin{cases} \dfrac{1}{1 + \max\{\tau^i, \tau^j\}} & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{if } (i,j) \notin \mathcal{E}, i \neq j \\ 1 - \sum_{j \in \mathcal{N}(i)} w_{ij} & \text{if } i = j \end{cases}$$

We suppose the matrix $\mathbf{W}$ is doubly stochastic, i.e $\mathbf{W1} = \mathbf{W}^T\mathbf{1} = \mathbf{1}$ and denote by $\lambda(\mathbf{W})$ the absolute second largest eigenvalue of $\mathbf{W}$, the spectral gap of $\mathbf{W}$ is denoted by $\rho(\mathbf{W}) = 1 - \lambda(\mathbf{W})$. We denote by t the time step and T the time horizon. In the distributed setting, we add a superscript i to denote the dependance on the agent i e.g $\boldsymbol{x}_t^i$ is the decision vector of agent i at time t. Then we note by $\overline{\boldsymbol{x}}_t := \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_t^i$ the average decision vector of all agents at time step t and finally, we use the notation $\mathcal{P}_{\mathcal{K}}(\boldsymbol{x})$ to denote the projection of $\boldsymbol{x}$ onto the set $\mathcal{K}$. In the following, we recall some basic denitions that are commonly used in convex optimization and machine learning.

Gradient Tracking A common scenario in distributed optimization is that the agents do not have access to the global objective function but only to their local functions f^i. This problem is more pronounced when the data are heterogeneous between agents as the update direction of each agent may not be aligned with the global objective. To mitigate this problem, one popular technique in the literature is the use of gradient tracking [Lorenzo2016, Pu2018]. Let $f : \mathcal{K} \to \mathbb{R}$, we define the tracking variable as $\boldsymbol{g}_t^i$. The gradient tracking mechanism can be formulated as, for all $i \in [n]$ and $t \in [T]$:

$$\boldsymbol{g}_{t+1}^i = \nabla f^i(\boldsymbol{x}_{t+1}^i) - \nabla f^i(\boldsymbol{x}_t^i) + \sum_{j \in \mathcal{N}(i)} w_{ij} \boldsymbol{g}_t^j$$

From the above formulation, the tracking variable $\boldsymbol{g}_{t+1}^i$ is updated by the local gradient $\nabla f^i(\boldsymbol{x}_{t+1}^i)$ and a bias correction term which helps aligning the update direction of each agent with the global objective. By induction on t, one can easily verify that $\frac{1}{n}\sum_{i=1}^n \boldsymbol{g}_t^i = \frac{1}{n}\sum_{i=1}^n \nabla f^i(\boldsymbol{x}_t^i)$

Definition 1 (Convexity)**.** *Let $\mathcal{K} \subset \mathbb{R}^d$. For all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{K}$ and $\lambda \in [0,1]$, $\mathcal{K}$ is said to be convex if*

$$\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y} \in \mathcal{K}$$

Let f be a function defined over the convex set $\mathcal{K}$, f is convex if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{K}$ and $\forall \lambda \in [0,1]$, we have

$$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y})$$

If f is differentiable, then f is convex if its verify $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{K}$,

$$\langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle \leq f(\boldsymbol{y}) - f(\boldsymbol{x})$$

Definition 2 (Lipschitz Continuity). *Let $\mathcal{K} \subset \mathbb{R}^d$. A function f is said to be G-Lipschitz over $\mathcal{K}$ if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{K}$ and $\forall t \in [T]$, we have*

$$|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq G \|\boldsymbol{x} - \boldsymbol{y}\|$$

Equivalently, the gradient of f is upper-bounded by G i.e $\forall \boldsymbol{x} \in \mathcal{K}$, $\|\nabla f(\boldsymbol{x})\| \leq G$.

Definition 3 (Smoothness). *Let $\mathcal{K} \subset \mathbb{R}^d$. A function f is said to be β-smooth over $\mathcal{K}$ if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{K}$ and $\forall t \in [T]$, we have*

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\beta}{2} \|\boldsymbol{y} - \boldsymbol{x}\|^2$$

More over, if f is convex and the gradient of f is Lipschitz continuous i.e

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq \beta \|\boldsymbol{x} - \boldsymbol{y}\|$$

Then f is β-smooth an the two definitions are equivalents.

Definition 4 (ℓ_p-norm). *For $p \geq 1$, we define the ℓ_p-norm of a vector $\boldsymbol{x} \in \mathbb{R}^d$ as*

$$\|\boldsymbol{x}\|_p := \left(\sum_{i=1}^{d} |x_i|^p \right)^{1/p}$$

For most chapters, if not specified otherwise, we will consider the ℓ_2-norm and denote it by $\|\cdot\|$ for simplicity of notation. We make the following assumptions on the constraint set $\mathcal{K}$, the functions $f_t, t \in [T]$ and the adjacency matrix that will be used throughout this manuscript.

Assumption 1.2.1. *The constraint set $\mathcal{K}$ is a compact convex set with diameter D and radius R, i.e, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{K}$, we have*

$$D := \sup_{\boldsymbol{x}, \boldsymbol{y} \in \mathcal{K}} \|\boldsymbol{x} - \boldsymbol{y}\| \qquad R := \sup_{\boldsymbol{x} \in \mathcal{K}} \|\boldsymbol{x}\|$$

Assumption 1.2.2. *For all $t \in [T]$, the functions f_t are G-Lipschitz and β-smooth over $\mathcal{K}$ for some constants G and β.*

Assumption 1.2.3. *Let $\mathbf{W}$ be the adjacency matrix of the communication graph $\mathcal{G}$. We assume that the graph G is connected and that the matrix $\mathbf{W}$ is doubly stochastic with $\lambda(\mathbf{W})$ the second largest eigenvalue of $\mathbf{W}$. Then, there exist a smallest integer k_0 that verifies :*

$$\lambda(\mathbf{W}) \leq \left(\frac{k_0}{k_0 + 1} \right)^2$$

1.3 Online Optimization

The concept of online optimization can be cast into a repeated game between a player and an environment. At each round, the player takes an action and subsequentially receive a loss/reward from the environment. As the game goes on, the player tries to improves its decision mechanism based on the outcomes of previous actions and the goal is to minimize (or maximize) the cumulative loss (reward) over time. As opposed to the concept of batch learning where the data has an underlying distribution. In online learning, we allow the sequenc to be deterministic, stochastic or even adversarial. The performance of the player in a long run is often measured by a notion called regret, which quantifies the gap between the player cumulative loss and the one of the best player or in other word, how regret the player is for not following the best players actions. This notion of regret can find its simlarity to the estimation error in the statistical learning theory where we take the difference between the error of a hypothesis with the best one the the hypothesis class. The

regret can also be interpreted as a measure of robustness as small regret guarentees the performance of the player is as good as the best player in hindsight.

Formally, let $f_1, \ldots, f_t$, $f_t : \mathcal{K} \to \mathbb{R}$ be a sequence of adversarial convex functions, defined over a compact convex set $\mathcal{K} \subset \mathbb{R}^d$. At each time t, the agent selects a point $\boldsymbol{x}_t \in \mathcal{K}$ and incurs a loss $f_t(\boldsymbol{x}_t)$. The agent then updates its strategy based on the incurred loss. We define regret as the difference between the cumulative loss of the agent and the one of the optimal decision in hindsight, formally, we note.

$$\mathcal{R}_T = \sum_{t=1}^{T} f_t(\boldsymbol{x}_t) - \min_{\boldsymbol{x} \in \mathcal{K}} \sum_{t=1}^{T} f_t(\boldsymbol{x}) \tag{1.1}$$

Algorithm 1 Online Learning

1: **Input:** $\mathcal{K}$, T, $\boldsymbol{x}_1$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Choose a point $\boldsymbol{x}_t \in \mathcal{K}$
4: Incurs loss $f_t(\boldsymbol{x}_t)$
5: Update internal state
6: **end for**

The study of OCO dates back to the work of [Zinkevich2003] where the author proposed an online variant of gradient descent algorithm 2. At each time t, the algorithm chooses the next decision by taking a step in the direction of the negative gradient of the incurred loss. The resulting decision may fall outside the feasible set $\mathcal{K}$ and therefore must be projected back into the set. This projection operation is often computationally expensive and may not be practical for certain problems. Another popular family of algorithms for OCO is the Follow-The-Leader [Kalai2005]. This algorithm is based on the simple idea of choosing a decision in $\mathcal{K}$ that minimizes the cumulatives of previous losses. Although this algorithm fails to achieve a sublinear regret bound in many cases [Hazan2016a], it is considered a reference point for many regularized algorithms, namely Follow-Regularized Leader (FTRL) [Shalev-Shwartz2012, Abernethy2008] and Follow-Perturbed Leader (FTPL) [Kalai2005, Hannan1958], which have served as one of the underlying mechanisms for many of the algorithms proposed in this thesis. It should be noted that [Shalev-Shwartz2012] also offers an online variant of the mirror descent algorithm, which is equivalent to FTRL [Shalev-Shwartz2007] for lazy update with linear loss function, and both algorithms achieve the same regret bound. For a comprehensive survey of OCO, we refer the reader to [Hazan2016a].

Algorithm 2 Online Gradient Descent

1: **Input:** $\mathcal{K}, T, \boldsymbol{x}_1$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Choose a point $\boldsymbol{x}_t \in \mathcal{K}$
4: Incurs loss $f_t(\boldsymbol{x}_t)$
5: Compute gradient $\nabla f_t(\boldsymbol{x}_t)$
6: Update $\boldsymbol{x}_{t+1} = \mathcal{P}_{\mathcal{K}}\left(\boldsymbol{x}_t - \eta \nabla f_t(\boldsymbol{x}_t)\right)$
7: **end for**

1.3.1 Online Optimization Oracle

An online optimization oracle is a black-box optimization algorithm that solves the online optimization problem. At each time t, the oracle receives a feedback function $\boldsymbol{g}_t$ and a constraint set $\mathcal{K}$, and it must output a decision $\boldsymbol{v}_t$ that satisfies the constraint and minimizes (resp. maximizes) the given optimization problem. One common candidate of an online optimization oracle is the OGD (algorithm 2). In this case, the feedback function is the gradient of the loss function and the decision is the result of projection onto the constraint set of the gradient descent step.
A particular case of the online optimization oracle is the online linear oracle (OLO), where the feedback function is linear. Let $\langle \boldsymbol{g}_1, \cdot \rangle, \ldots, \langle \boldsymbol{g}_t, \cdot \rangle$ be the sequence of linear loss functions. The goal of the OLO is to select an extreme point $\boldsymbol{v}_t \in \mathcal{K}$ that minimizes the cumulative loss, i.e., $\boldsymbol{v}_t = \arg\min_{\boldsymbol{v} \in \mathcal{K}} \sum_{\tau=1}^{t} \langle \boldsymbol{g}_\tau, \boldsymbol{v} \rangle$. In the algorithms proposed in this thesis, we will use the OLO as a

subroutine to solve the online optimization problem. The specificity of each OLO will be detailed in the corresponding sections.

1.4 Distributed Online Optimization

We consider a network of cooperative agents connected via an undirected graph $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ and n sequences of functions $f_{i,1}, \ldots, f_{i,T}$. We suppose $f_t^i : \mathcal{K} \to \mathbb{R}$ is convex and defined on a compact convex set $\mathcal{K} \subset \mathbb{R}^d$ and define the global objective function as :

$$F_t(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} f_t^i(\boldsymbol{x}) \quad \forall t \in [T], \boldsymbol{x} \in \mathcal{K} \tag{1.2}$$

where we are interested in finding a sequence generation strategy that minimizes the cumulative global loss over the time horizon T, which can be formulated as:

$$\min_{\boldsymbol{x}_t \in \mathcal{K}} \sum_{t=1}^{T} F_t(\boldsymbol{x}_t) \tag{1.3}$$

At each round t, every agent i selects a decision $\boldsymbol{x}_t^i$ from a feasible set $\mathcal{K}$ and incurs a loss $f_t^i(\boldsymbol{x}_t^i)$, where f_t^i revealed by the environment and is adversarial to the agents. Subsequently, each agent adjusts its strategy on the basis of the history and current information received from neighboring agents. At first glance, the problem of Distributed Online Optimization (DOO) may appear similar to Online Convex Optimization (OCO). However, DOO presents additional challenges. Firstly, agents are not merely attempting to minimize their individual cumulative loss; they need to cooperate to optimize the global objective F_t. This is challenging because agents do not have direct access to the global loss function and only receive partial information on F_t through their local functions f_t^i. Secondly, the individual decision variable $\left\{\boldsymbol{x}_t^i\right\}_{i=1}^n$ often differs for $i \neq j$ as each agent's decision-making process is based on its own local history and information exchange with neighboring agents. This necessitates careful algorithm design to ensure that the agents reach a consensus on the strategy that minimizes the cumulative global loss as defined in equation (1.3). As a result, we consider the following regret formulation, termed individual regret for each agent $i \in [n]$:

$$\mathcal{R}_T^i = \sum_{t=1}^{T} F_t(\boldsymbol{x}_t^i) - \min_{\boldsymbol{x} \in \mathcal{K}} \sum_{t=1}^{T} F_t(\boldsymbol{x}) \tag{1.4}$$

which compares the cumulative global loss on agent's strategy to that of the optimal decision in hindsight.The goal derived from this formula is to design algorithms that achieve sublinear regret for all agents i.e $\forall i \in [n], \mathcal{R}_T^i = o(T)$. It's also worth noting that other regret formulation existe in the context of DOO such as the network regret, which is defined as:

$$\mathcal{R}_T^{network} = \frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{T} f_t^i(\boldsymbol{x}_t^i) - \min_{\boldsymbol{x} \in \mathcal{K}} \sum_{t=1}^{T} F_t(\boldsymbol{x}) \tag{1.5}$$

while this formulation have similarities to the previous one, the underlying objective is quite different since it seeks is to minimize the cumulative loss of all agents in the network. Notably, this does not inherently necessitates cooperation, as agents can simply choosing strategy that minimize their own cumulative loss. In this thesis, if not stated otherwise, we will focus on the individual regret formulation as defined in equation (1.4).

The field of distributed online optimization has witnessed substantial research in recent years, finding applications in machine learning, game theory, control, and multi-agent systems. Previous works in this domain include a distributed online projected subgradient descent algorithm introduced by [Yan2013], and a distributed dual averaging technique extended to the online setting by [Hosseini2013]. [Shahrampour2018] propose a distributed version of the online mirror descent

algorithm and analyze its convergence properties for both exact and stochastic gradients. [Li2022] also study distributed gradient descent methods with an additional gradient tracking step, providing theoretical guarantees for both exact and stochastic gradients. For a comprehensive survey on recent developments in distributed online optimization, we refer the reader to [Li2023]. In each chapter, we will present a dedicated section to the related works in the context of the problem at hand.

1.5 Conditional Gradient Algorithm

The Conditional Gradient Algorithm, also known as the Frank-Wolfe [Frank1956], is a projection-free method for solving constrained optimization problems. In many machine learning applications, it is common to requires the model parameters to satisfy certain constraints. For example, to improve memory efficiency and prevent overfitting, one may require the weights of a machine learning model to be sparse, which involves constraining the weights to belong to an L_1 norm ball. Matrix completion is another popular constrained optimization problem in recommender systems where the matrix of user-item ratings is often incomplete. The goal is to fill in the missing entries for recommendation and a common assumption in matrix completion is that the matrix has low rank which requires a constraint set to be the nuclear norm ball.

The common approach for solving theses problems is projected gradient descent, where the solution is projected onto the constraint set at each iteration. However, the projection step is usually computationally expensive and may not be feasible in many large-scale problems. On the other hand, the Conditional Gradient Algorithm (Algorithm 3) is a better alternative when the constraint has a fast linear optimization. The idea of the Frank-Wolfe is to iteratively minimize the

Algorithm 3 Frank-Wolfe Algorithm

1: **Input**: A constraint set $\mathcal{K}$, $\boldsymbol{x}_0 \in \mathcal{K}$, T.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: $\quad \boldsymbol{v}_t = \arg\min_{\boldsymbol{x} \in \mathcal{K}} \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{x} \rangle$.
4: $\quad$ Choose a step size $\gamma_t \in [0, 1]$.
5: $\quad \boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \gamma_t(\boldsymbol{v}_t - \boldsymbol{x}_t)$.
6: **end for**

first order approximation of the objective function and subsequently moving toward an extreme point, solution of the minimization problem. Specifically, let $f : \mathcal{K} \to \mathbf{R}$ be a differentiable convex function,the Taylor's expansion of f at a point $\boldsymbol{x} \in \mathcal{K}$ as follows:

$$f(\boldsymbol{x}) \simeq f(\boldsymbol{y}) + \langle \nabla f(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle \quad \forall \boldsymbol{y} \in \mathcal{K} \tag{1.6}$$

Minimizing the right hand side of (1.6) over $\boldsymbol{x} \in \mathcal{K}$ consist of solving a linear problem $\langle \nabla f(\boldsymbol{y}), \boldsymbol{x} \rangle$ that yields a solution v as the extreme point of the constraint set that is mostly correlated with the negative gradient $-\nabla f(\boldsymbol{y})$. By moving the current iterate on that direction, the FW algorithm ensures that all the points are feasible without the need for projection. The details of vanilla FW is shown in Algorithm 3.

1.5.1 Online Frank-Wolfe

The study of Frank-Wolfe algorithm in online optimization has gained a lot of interest in recent year since its projection-free nature makes it suitable for designing computational efficient algorithm. Online Frank-Wolfe is an online variant of FW that was proposed by [Hazan2012]. In constrast to the online variant of Gradient Descent [Zinkevich2003] where the previous gradient is sufficient to update the iterate. The adaptation of vanilla frank-wolfe to the online setting requires a surrogate objective function F_t that take into account the historical information. As a consequence, OFW apply the FW steps on this surrogate objective and solve the linear problem that still computational efficient in compared to the projection step in Online Gradient Descent.

Algorithm 4 Online Frank-Wolfe

Input: A constraint set $\mathcal{K}$, $\boldsymbol{x}_0 \in \mathcal{K}$, T.
for $t = 1, 2, \ldots, T$ **do**
  Play $\boldsymbol{x}_t$
  Observe $f_t(\boldsymbol{x}_t)$.
  $F_t(\boldsymbol{x}) = \eta \sum_{\tau=1}^{t} \langle \nabla f_\tau(\boldsymbol{x}_\tau), \boldsymbol{x} \rangle + \|\boldsymbol{x} - x_1\|^2$
  $\boldsymbol{v}_t = \arg\min_{\boldsymbol{x} \in \mathcal{K}} \langle \nabla F_t(\boldsymbol{x}_t), \boldsymbol{x} \rangle$.
  Choose a step size $\gamma_t \in [0, 1]$.
  $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \gamma_t(\boldsymbol{v}_t - \boldsymbol{x}_t)$.
end for

1.5.2 Motivation Example

To illustrate the efficient of Frank-Wolfe and Projection algorithm in Online setting, we run two algorithm on a matrix completion problem [Hazan2012]. We are given a matrix $\mathbf{Y} \in \mathbb{R}^{m \times n}$ with y_{ij} being the rating of user i on item j. The matrix $\mathbf{Y}$ is sparse and the goal is to fill the missing entries by estimating a low rank approximation of the matrix. This problem can formulated as constraint optimization problem where the feasible region is the nuclear norm ball defined as :

$$\mathcal{K} = \{\mathbf{X} \in \mathbb{R}^{m \times n} : \|\mathbf{X}\|_* \leq \alpha\}$$

where α is the maximal rank of the matrix. We let the objective function is the square loss function defined as:

$$f(\mathbf{X}) = \frac{1}{2} \sum_{(i,j)} (x_{ij} - y_{ij})^2$$

To project the matrix to the nuclear norm ball, the OGD need to do a full singular value decomposition of the matrix which is usually take $O(mn \min(m, n))$. On the other hand, the solving the linear problem in OGD only need to compute the top-left and right singular vector which take only linear time. To illustrate the running time of the two algorithm, we run the two algorithm on the MovieLens100k dataset that has 100k observed entries, 943 users and 1682 items, we set $\alpha = 100$ and run the two algorithm 6000 iterations. We can see on the right figure of 1.1, the running time of OGD is significantly higher than OFW. This is due to the fact that OGD takes in average 0.3 seconds to complete one round of computation while the amount for OFW is only 0.01 seconds which is 30 times faster. More over, the running average loss also so show a better performance of OFW in compared to OGD.

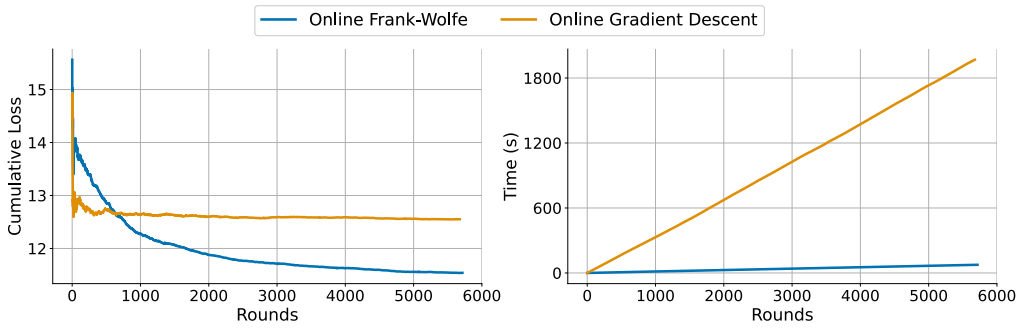


Figure 1.1: Running time of OGD and OFW on MovieLens100k dataset. Left figure shows the running average loss of the two algorithms and the right figure show the running time of the two algorithms.

1.6 Contributions

In online optimization, an essential component is the feedback from the environment that is received immediately after a decision is made. For instance, in a game, the player anticipates the reward of an action to be revealed right after taking the action, or in a classification model, we would like to receive the true label of the classified example promptly. However, this immediate feedback might not be available in many real-world problems. A typical scenario for this problem is online advert placement, where the advertiser only observes the user's action of clicking or purchasing the product after a few days or even months. Another typical scenario is in distributed learning, where devices might have different network or computation configurations. Consequently, some devices might take longer to receive their feedback at the local level, leading to a delay in sending or receiving information with neighboring agents. In this scenario, we pose the following question:

"Can we still achieve an optimal regret bound in a delayed setting with a projection-free algorithm?"

We address this question in Chapter 2 by proposing a variant of the Meta Frank-Wolfe algorithm [Chen2018a] that can adapt to delayed feedback in both centralized and distributed settings. In the centralized setting, we analyze the behavior of the online linear oracle when subjected to delayed feedback and provide an upper bound on the Euclidean distance between the prediction of the Online Linear Optimization (OLO) with and without delay. We structure the analysis of Algorithm 5 around this result. We extend this idea to the distributed setting to derive an analysis for Algorithm 6, where we demonstrate the dependence of the effect of delay on the network topology. Finally, we provide numerical experiments on simulated and real-world data to illustrate the performance of our algorithms.

Submodular functions are a class of functions that have extensive applications in machine learning and optimization problems. One such application is in influential marketing, where the objective is to identify a subset of influencers that can maximize the spread of information in the network [Kempe2003]. Another popular application of submodularity is data subset selection, where the goal is to find an optimal subset of data that is informative enough to train a model accurately. Finding an optimal subset is NP-hard in general, but an approximate solution can be obtained by a reformulation as submodular maximization problem [Wei2015]. In Chapter 3, we study the problem the problem of submodular maximization via a continuous relaxation and convex minimization in a distributed online setting. We propose a Frank-Wolfe-like algorithm that requires each agent to compute only one gradient evaluation of the objective function per round, which is a significant improvement over existing algorithms that require at least $O(T^{3/2})$ gradient computations per agent per round. To achieve this result, we use a blocking procedure to partition the horizon into blocks and consider the average of the objective function over each block as a virtual objective function to be optimized. Consequently, each function f_t becomes an estimate of the virtual objective that needs to be queried only once. This idea was first developed by [Zhang2019] in a centralized manner. We first consider the full information setting for convex and DR-Submodular functions, where the agent has access to the first-order feedback of the objective function. We then extend our analysis to the bandit setting, where the agent has access only to zeroth-order feedback of the objective function. We provide a theoretical guarantee for our algorithms and demonstrate their performance on a movie recommendation problem.

The era of learning on the edge has opened up many applications in the field of distributed learning. One application field that has captured our attention is smart buildings, due to their high environmental implications such as energy consumption, air quality and comfort, and greenhouse gas emissions. Smart buildings are complex systems that consist of multiple sensors that capture surrounding environment data, process, and adjust the overall system, on-device and in real-time, to optimize the energy consumption and occupant comfort. This concept aligns with the framework of distributed online optimization, where each sensor can be seen as an agent that optimizes its own objective function (which may be non-convex by nature) while exchanging information with neighboring agents to achieve a global consensus. Inspired by this application, we propose in Chapter 4 a Frank-Wolfe distributed online algorithm that minimizes non-convex loss functions under exact and stochastic gradient settings. As opposed to the well-studied online convex optimization problem, the study of online non-convex optimization is still an open research question in the field of machine learning. In the online convex setting, the performance of an algorithm is often measured

by the regret (or normalized regret), which is the difference between the cumulative loss of the algorithm and that of the best fixed decision in hindsight. However, this measure is not suitable for the non-convex problem, as finding a global minimum is NP-hard in general. To address this challenge, we propose a generalized measure of the Frank-Wolfe gap [Lacoste-Julien2016, Jaggi2013] to the online setting, which we call call the *convergence-gap*. We provide a convergence rate of $O(T^{-1/2})$ and $O(T^{-1/4})$ for the exact and stochastic gradient,respectively. To validate the performance of the algorithm, we run numerical experiments on a time-series forecasting problem using a real-life smart building dataset and measure the performance on various network size and topology.

Federated learning has gained significant attention in recent years due to its potential to train machine learning models on edge devices without the need to transfer data to a central server. This approach is particularly useful in scenarios where data privacy is a concern, such as in healthcare, finance, and smart cities. However, training machine learning model on multiple edge devices can be computationally expensive and energy-consuming, which can be a significant challenge for devices with limited computational resources. In Chapter 5, we investigate the energy consumption of federated learning on edge devices. We consider various federated learning algorithms, including FedAvg, FedAdam, FedYogi, and FedAdaGrad, with differents local optimizer and hyperparameters settings and measure their energy consumption on a real-world dataset. Our experiments show that the energy consumption of federated learnings varies depending on the time to convergence, which is influenced by the training configuration. We also highlight the important of client-sampling in designing energy-efficient FL algorithm, especially in a cross-silo setting. Our research provides valuable insights into the energy consumption of federated learning algorithms and sets the stage for future explorations in this field.

1.7 Publications

The material of this manuscript are selected from following publications :

- Tuan-Anh Nguyen, Nguyen Kim Thang et Denis Trystram. *Handling Delayed Feedback in Distributed Online Optimization : A Projection-Free Approach.* In Machine Learning and Knowledge Discovery in Databases: Research Track. ECML PKDD, 2024
- Tuan-Anh Nguyen, Nguyen Kim Thang et Denis Trystram. *One Gradient Frank-Wolfe for Decentralized Online Convex and Submodular Optimization.* In Proceedings of The 14th Asian Conference on Machine Learning, volume 189 of *Proceedings of Machine Learning Research*, pages 802–815. PMLR, 12–14 Dec 2023
- Angan Mitra, Nguyen Kim Thang, Tuan-Anh Nguyen, Denis Trystram et Paul Youssef. *Online Decentralized Frank-Wolfe: From Theoretical Bound to Applications in Smart-Building.* In Internet of Things: 5th The Global IoT Summit, GIoTS 2022, Dublin, Ireland, June 20–23, 2022, Revised Selected Papers, page 43–54, Berlin, Heidelberg, 2023. Springer-Verlag

I also have the opportunity to contribute to the following publication :

- Akash Dhasade, Anne-Marie Kermarrec, Tuan-Anh Nguyen, Rafael Pires et Martijn de Vos. *Harnessing Increased Client Participation with Cohort-Parallel Federated Learning*, 2024

Tableau 1.1: Notations

Notation	Description
$\mathbb{G}$	Communication graph
n	Total number of agents/devices/users/nodes
$\mathbf{W}$	Adjadcency matrix of $\mathcal{G}$
$\lambda(\mathbf{W})$	Second largest eigen value of $\mathbb{W}$
$\mathcal{K}$	Constraint set
$\mathcal{P}_{\mathcal{K}}$	Projection operator on $\mathcal{K}$
D	Diameters of $\mathcal{K}$
R	Radius of $\mathcal{K}$
T	Time horizon
K	Sub-iteration
f_t^i	Loss function of agent i at time t
$\boldsymbol{x}_t^i$	Decision of agent i at time t
$\overline{\boldsymbol{x}}_t$	Average decision at time t
F_t	Global loss function at time t
$\delta_{t,k}^i$	$f_t^i(\boldsymbol{x}_{t,k}^i) - f_t^i(\boldsymbol{x}_{t,k-1}^i)$
d_t	Delay at time t
$\mathcal{G}$	Duality gap
$\mathcal{R}_{i,T}$	The individual regret of agent i at time T
$\mathcal{R}_T$	The network regret at time T
R	The rating matrix where R_{ij} is the rating of user i for item j

2

Distributed Online Optimization with Delayed Feedback

Contents

2.1 Introduction

Many machine learning (ML) applications owe their success to factors such as efficient optimization methods, effective system design, robust computation, and the availability of enormous amounts of data. In a typical situation, ML models are trained in an offline and centralized manner. However, in real-life scenarios, significant portions of data are continuously generated locally at the user level. Learning at the edge naturally emerges as a new paradigm to address such issues. In this new paradigm, the development of suitable learning techniques has become a crucial research objective. Responding to the requirements (of this new paradigm), online learning has been intensively studied in recent years. Its efficient use of computational resources, adaptability to changing environments, scalability, and robustness against uncertainty show promise as an effective approach for edge devices.

However, online learning/online convex optimization (OCO) problems typically assume that the feedback is immediately received after a decision is made, which is too restrictive in many real-world scenarios. For example, a common problem in online advertising is the delay that occurs between clicking on an ad and taking subsequent action, such as buying or selling a product. In distributed systems, the previous assumption is clearly a real issue. Wireless sensor/mobile networks that exchange information sequentially may experience delays in feedback due to several problems: connectivity reliability, varying processing/computation times, heterogeneous data and infrastructures, and unaware-random events. This can lead to difficulties in maintaining coordination and efficient data exchange, eventually affecting network performance and responsiveness. Given these scenarios, the straightforward application of traditional OCO algorithms often results in inefficient resource utilization because one must wait for feedback before starting another round. To address this need, this paper focuses on developing algorithms that can adapt to adversarial delayed feedback in both centralized and distributed settings.

Model. We first describe the delay model in a centralized setting. Given a compact convex set $\mathcal{K} \subseteq \mathbb{R}^d$, at every time step t, the decision maker/agent chooses a decision $\boldsymbol{x}_t \in \mathcal{K}$ and suffers from a loss function $f_t : \mathcal{K} \to \mathbb{R}$. We denote by $d_t \geq 1$ an arbitrary delay value of time t. In contrast to the classical OCO problem, the feedback of iteration t is revealed at time $t + d_t - 1$. The agent does not know d_t in advance and is only aware of the feedback of iteration t at time $t + d_t - 1$. Consequently, at time t, the agent receives feedback from the previous iterations $s \in \mathcal{F}_t$, where $\mathcal{F}_t = \{s : s + d_s - 1 = t\}$. In other words, $\mathcal{F}_t$ is the set of moments before time t such that the corresponding feedbacks are released at time t. Moreover, the corresponding feedbacks are not necessarily released in the order of their iterations. The goal is to minimize regret, which is defined as:

$$\mathcal{R}_T := \sum_{t=1}^{T} f_t(\boldsymbol{x}_t) - \min_{\boldsymbol{x} \in \mathcal{K}} \sum_{t=1}^{T} f_t(\boldsymbol{x})$$

In a distributed setting, we have additionally a set of agents connected over a network, represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $n = |\mathcal{V}|$ is the number of agents. Each agent $i \in [n]$ can communicate with (and only with) its immediate neighbors, that is, adjacent agents in $\mathcal{G}$. At each time $t \geq 1$, agent i takes a decision $\boldsymbol{x}_t^i \in \mathcal{K}$ and suffers a partial loss function $f_t^i : \mathcal{K} \to \mathbb{R}$, which is revealed adversarially and locally to the agent at time $(t + d_t^i - 1)$ — again, that is unknown to the agent. Similarly, denote $\mathcal{F}_t^i = \{s : s + d_s^i - 1 = t\}$ as the set of feedbacks revealed to agent i at time t where d_s^i is the delay of iteration s to agent i. Although the limitation in communication and information, the agent i is interested in the global loss $F_t(.)$ where $F_t(.) = \frac{1}{n} \sum_{i=1}^{n} f_t^i(.)$. In particular, at time t, the loss of agent i for chosen $\boldsymbol{x}_t^i$ is $F_t(\boldsymbol{x}_t^i)$. Note that each agent i does not know F_t but has only knowledge of f_t^i — its observed cost function. The objective here is to minimize regret for all agents:

$$\mathcal{R}_T := \max_i \left(\sum_{t=1}^{T} F_t(\boldsymbol{x}_t^i) - \min_{\boldsymbol{x} \in \mathcal{K}} \sum_{t=1}^{T} F_t(\boldsymbol{x}) \right)$$

2.1.1 Our contribution

The challenge in designing robust and efficient algorithms for these problems is to address the following issues simultaneously:

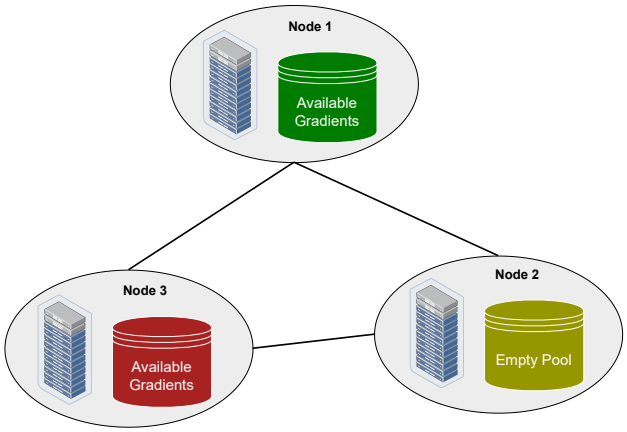


Figure 2.1: Illustration of delayed feedback in distributed system. Given a time t, each agent holds a distinct pool of available gradient feedback from $s < t$ that is ready for computation at the current time. The pool can also be empty if no feedback is provided.

- Uncertainty (online setting, agents observe their loss functions only after selecting their decisions).
- Asynchronous (distributed setting with different delayed feedback between agents)
- Partial information (distributed setting, agents know only its local loss functions while attempting to minimize the cumulative loss).
- Low computation/communication resources of agents (so it is desirable that each agent performs a small number of gradient computations and communications).

We introduce performance-guaranteed algorithms in solving the centralized and distributed constraint online convex optimization problem with adversarial delayed feedback. Our algorithms achieve an *optimal* regret bound for centralized and distributed settings. Specifically, we obtain the regret bound of $O(\sqrt{B})$ where B is the total delay in the centralized setting and B is the average total delay over all agents in the distributed setting. Note that, if d is a maximum delay of each feedback then our regret bound becomes $O(\sqrt{dT})$. This result recovers the regret bound of $O(\sqrt{T})$ in the classic setting without delay (i.e., $d = 1$). Additionally, the algorithms can be made projection-free by selecting appropriate oracles, allowing them to be implemented in different contexts based on the computational capacity of local devices. Finally, we illustrate the practical potential of our algorithms and provide a thorough analysis of their performance which is predictably explained by our theoretical results. The experiments demonstrate that our proposed algorithms outperform existing solutions in both synthetic and real-world datasets.

Tableau 2.1: Comparisons to previous algorithms DGD [Quanrud2015] and DOFW [Wan2022a] on centralized online convex optimization with delays bounded by d. Our algorithms are in bold.

Algorithm	Centralized	Distributed	Adversarial Delay	Projection-free	Regret
DGD	✓	-	✓	-	$O(\sqrt{dT})$
DOFW	✓	-	✓	✓	$O(T^{3/4} + dT^{1/4})$
DeLMFW	✓	-	✓	✓	$O(\sqrt{dT})$
De2MFW	-	✓	✓	✓	$O(\sqrt{dT})$

2.1.2 Related Work

Online Optimization with delayed feedback Over the years, studies on online optimization with delayed feedback have undergone a swift evolution. [Zinkevich2009] shed light on the field by focusing on the convergence properties of online stochastic gradient descent with delays. They provide a regret bound of $O(\sqrt{dT})$ with d the delay value if $d^2 \leq T$. Later on, [Quanrud2015] proposes a centralized (single-agent) gradient descent algorithm under adversarial delays. The theoretical analysis of [Quanrud2015] entails a regret bound of $O(\sqrt{B})$, where B is the total delay. This bound becomes $O(\sqrt{dT})$ if d is the upper bound of delays. [Joulani2013] provided a black-box style method to learn under delayed feedback. They showed that for any non-delayed online algorithms, the additional regret in the presence of delayed feedback depends on its prediction drifts. [Cao2021] developed an online saddle point algorithm for convex optimization with feedback delays. They achieved a sublinear regret $O(\sqrt{dT})$ where d is a fixed constant delay value. Recently, [Wan2022a] proposed a first Frank-Wolfe-type online algorithm with delayed feedback. They modified the Online Frank-Wolfe (OFW) for the unknown delays setting and provided a regret bound of $O(T^{3/4} + dT^{1/4})$. This is the current state of the art for projection-free (Frank-Wolfe-type) algorithms with delays. Our bound of $O(\sqrt{dT})$ improves over the aforementioned results.

Distributed Online Optimization. [Yan2013] introduced decentralized online projected sub-gradient descent and showed vanishing regret for convex and strongly convex functions. In contrast, [Hosseini2013] extended distributed dual averaging technique to the online setting, using a general regularized projection for both unconstrained and constrained optimization. A distributed variant of online conditional gradient [Hazan2016a] was designed and analyzed in [Zhang2017] that requires linear minimizers and uses exact gradients. Computing exact gradients may be prohibitively expensive for moderately sized data and intractable when a closed form does not exist. [Thang2022] proposes a decentralized online algorithm for convex function using stochastic gradient estimate and multiple optimization oracles. This work achieves the optimal regret bound of $O(T^{1/2})$ and requires multiple gradient evaluation and communication rounds. Later on, [Nguyen2023] provide a decentralized algorithm that uses stochastic gradient estimate and reduces communication by using only one gradient evaluation. More recent work on distributed online optimization with feedback delays is proposed in [Cao2022]. The authors consider a distributed projected gradient descent algorithm where each agent has a fixed known amount of delay d_i. They provide a regret bound of $O(\sqrt{dT})$ where $d = \max_i d_i$ but the delays d_i must be fixed (non-adversarial).

Despite the growing number of studies on decentralized online learning in recent years, there is a lack of research that accounts for the *adversarial/online* delayed feedback. In this paper, we first present a centralized online algorithm and then extend it to a distributed online variant that takes an adversarial delay setting into consideration.

2.2 Preliminaries

Online Linear Optimization Oracles In the context of the Frank-Wolfe (FW) algorithm, we utilize multiple optimization oracles to approximate the gradient of the upcoming loss function by solving an online linear problem. This approach was first introduced in [Chen2018a]. Specifically, the online linear problem involves selecting a decision $\boldsymbol{v}_t \in \mathcal{K}$ at every time $t \in [T]$. The adversary then reveals a vector $\boldsymbol{g}_t$ and loss function $\langle \boldsymbol{g}_t, \cdot \rangle$ to the oracle. The objective is to minimize the oracle's regret. A possible candidate for an online linear oracle is the Follow the Perturbed Leader algorithm (FTPL) [Kalai2005]. Given a sequence of historical loss functions $\langle \boldsymbol{g}_\ell, \cdot \rangle, \ell \in [1, t]$ and a random vector $\boldsymbol{n}$ drawn uniformly from a probability distribution $\mathcal{D}$, FTPL makes the following update.

$$\hat{\boldsymbol{v}}_{t+1} = \arg\min_{\boldsymbol{v} \in \mathcal{K}} \left\{ \zeta \sum_{\ell=1}^{t} \langle \boldsymbol{g}_\ell, \boldsymbol{v} \rangle + \langle \boldsymbol{n}, \boldsymbol{v} \rangle \right\} \tag{2.1}$$

Lemma 2.2.1 (Theorem 5.8 [Hazan2016a]). *Given a sequence of linear loss function $f_1, \ldots, f_T$. Suppose that Assumptions 1.2.1 and 1.2.2 hold true. Let $\mathcal{D}$ be a the uniform distribution over hypercube $[0,1]^m$. The regret of FTPL is*

$$\mathcal{R}_{T,\mathcal{O}} \leq \zeta D G^2 T + \frac{1}{\zeta}\sqrt{m} D$$

where ζ is learning rate of algorithm.

Delay Mechanism We consider the following delay mechanism. At round t, the agent receives a set of delayed gradient $\nabla f_s(\boldsymbol{x}_s)$ from previous rounds $s \leq t$ such that $s + d_s - 1 = t$, where d_s is the delay value of iteration s. We denote by $\mathcal{F}_t = \{s : s + d_s - 1 = t\}$ the set of indices released at round t. Following this setting, the feedback of round t is released at time $t + d_t - 1$, and the case $d_t = 1$ is considered as no delay. We suppose that the delay value is unknown to the agent and make no assumption about the set $\mathcal{F}_t$. Consequently it is possible for the set to be empty at any particular round. We extend the aforementioned mechanism to the distributed setting by assuming that each agent has a unique delay value at each round $t \in [T]$. The delay value of agent i at round t is denoted by d_t^i, and the set of delayed feedbacks of agent i at round t is denoted by $\mathcal{F}_t^i = \left\{s : s + d_s^i - 1 = t\right\}$, which is distinct between agents.

2.3 Centralized Algorithm

We describe the procedure of Algorithm 5 in details. At each round t, the agent performs two blocks of operations: prediction and update. During the prediction block, the agent performs K iterations of FW updates by querying solutions from the oracles $\mathcal{O}_k$, $k \in [K]$ and updates the sub-iterate vector $\boldsymbol{x}_{t,k+1}$ using a convex combination of the previous one and the oracle's output. The agent then plays the final decision $\boldsymbol{x}_t = \boldsymbol{x}_{t,K+1}$ and incurs a loss $f_t(\boldsymbol{x}_t)$ which may not be revealed at t due to delay. From the mechanism described in Section 2.2, there exists a set of gradient feedbacks from the previous rounds revealed at t whose indices are in $\mathcal{F}_t$. The update block involves observing the delayed gradients evaluated at K sub-iterates of rounds $s \in \mathcal{F}_t$, computing surrogate gradients $\{\boldsymbol{g}_{t,k}, k \in [K]\}$ by summing the delayed gradients and feeding them back to the oracles $\{\mathcal{O}_k, k \in K\}$.

In our algorithm, the agent employs a suite of online linear optimization oracles, denoted $\mathcal{O}_1, \ldots, \mathcal{O}_K$. These oracles utilize feedbacks accumulated from previous rounds to estimate the gradient of the upcoming loss function. However, in the delay setting, these estimations may be perturbed owing to a lack of information. For example, if there is no feedback from rounds t to t', that is, $\mathcal{F}_s = \emptyset$ for $s \in [t, t']$, the oracles will resort to the information available in round $t-1$ to estimate the gradient of all rounds from $t+1$ to $t'+1$. As a result, the oracle's output remains unchanged for these rounds, and decisions $\{\boldsymbol{x}_s : s \in [t+1, t'+1]\}$ are not improved. Our analysis for Algorithms 5 and 6 will be focused on assessing the impact of delayed feedback on the oracle's output.

The proof of Theorem 2.3.1 necessitates Lemmas 2.3.1 and 2.3.2. The former provides a bound on the difference between the predictions of FTPL with and without delayed feedback at round t, which depends on the number of unrevealed feedbacks. The latter establishes a bound on the primal at a sub-iterate k in terms of the previous sub-iterate $k-1$ and the gradient feedback. This bound is crucial for the analysis of the regret of Algorithm 5. We will first prove Lemma 2.3.1 and Lemma 2.3.2 before moving on to the proof of Theorem 2.3.1.

Lemma 2.3.1. *Let $\hat{\boldsymbol{v}}_t$ be the FTPL prediction defined in Equation* (2.1) *and*

$$\boldsymbol{v}_t = \arg\min_{\boldsymbol{v} \in \mathcal{K}} \left\{ \zeta \sum_{\ell=1}^{t-1} \left\langle \sum_{s \in \mathcal{F}_\ell} \boldsymbol{g}_s, \boldsymbol{v} \right\rangle + \langle \boldsymbol{n}, \boldsymbol{v} \rangle \right\}$$

the prediction of FTPL with delayed feedback. For all $t \in [T]$, we have:

$$\|\boldsymbol{v}_t - \hat{\boldsymbol{v}}_t\| \leq \zeta DG \sum_{s<t} \mathbb{I}_{\{s+d_s>t\}}$$

Proof. Recall from Lemma 2.2.1 that $\boldsymbol{n}$ is drawn from $\mathcal{D}$, the uniform distribution over the hypercube $[0,1]^m$. Then $\mathcal{D}$ is (σ, L)-stable with respect to the Euclidean norm such that $\sigma \leq \sqrt{m}$ and $L \leq 1$ [Hazan2016a]. Let f_t be a linear function defined as $f_t = \langle \boldsymbol{g}_t, \cdot \rangle$. For ease of analysis, we call $\hat{\boldsymbol{u}}_t = \zeta \sum_{\ell=1}^{t-1} \boldsymbol{g}_\ell$ and $\boldsymbol{u}_t = \zeta \sum_{\ell=1}^{t-1} \sum_{s \in \mathcal{F}_\ell} \boldsymbol{g}_s$. We define $h_t(\boldsymbol{n}) = \arg\min_{\boldsymbol{v} \in \mathcal{K}} \{\langle \boldsymbol{n}, \boldsymbol{v} \rangle\}$. By definition of $\hat{\boldsymbol{v}}_t$ and $\boldsymbol{v}_t$, we have

$$\hat{\boldsymbol{v}}_t = \mathbb{E}\left[h_t(\boldsymbol{n} + \hat{\boldsymbol{u}}_t)\right] = \int_{\boldsymbol{n}} h_t(\boldsymbol{n} + \hat{\boldsymbol{u}}_t) p(\boldsymbol{n}) \mathrm{d}\boldsymbol{n} = \int_{\boldsymbol{n}} h_t(\boldsymbol{n}) p(\boldsymbol{n} - \hat{\boldsymbol{u}}_t) \mathrm{d}\boldsymbol{n} \tag{2.2}$$

and

$$\boldsymbol{v}_t = \mathbb{E}\left[h_t(\boldsymbol{n} + \boldsymbol{u})\right] = \int_{\boldsymbol{n}} h_t(\boldsymbol{n}) p(\boldsymbol{n} - \boldsymbol{u}) \mathrm{d}\boldsymbol{n} \tag{2.3}$$

where p is the density function. From linearity of expectation and Cauchy-Schwarz inequality,

$$\begin{aligned}
\|\boldsymbol{v}_t - \hat{\boldsymbol{v}}_t\| &\leq \int_{\boldsymbol{n}} \|h_t(\boldsymbol{n})\| \, |p(\boldsymbol{n} - \boldsymbol{u}) - p(\boldsymbol{n} - \hat{\boldsymbol{u}}_t)| \mathrm{d}\boldsymbol{n} \\
&= \int_{\boldsymbol{n}} \|h_t(\boldsymbol{n}) - h_t(\boldsymbol{0})\| \, |p(\boldsymbol{n} - \boldsymbol{u}) - p(\boldsymbol{n} - \hat{\boldsymbol{u}}_t)| \mathrm{d}\boldsymbol{n} \\
&\leq D \int_{\boldsymbol{n}} |p(\boldsymbol{n} - \boldsymbol{u}) - p(\boldsymbol{n} - \hat{\boldsymbol{u}}_t)| \mathrm{d}\boldsymbol{n} \\
&\leq DL \, \|\boldsymbol{u} - \hat{\boldsymbol{u}}_t\| \\
&\leq \zeta DLG \sum_{s<t} \mathbb{I}_{\{s+d_s>t\}}
\end{aligned} \tag{2.4}$$

The first inequality follows from the fact that $h_t(\boldsymbol{n})$ and $h_t(\boldsymbol{0})$ are in $\mathcal{K}$. The second inequality is due to the stability of the distribution $\mathcal{D}$. Since each function is G-Lipschitz, the distance between $\boldsymbol{u}$ and $\hat{\boldsymbol{u}}_t$ is bounded by G multiplied by the number of functions whose feedback is not received at time t, leading to the last inequality. □

Algorithm 5 DeLMFW

Input: Constraint set $\mathcal{K}$, number of iterations T, sub-iteration K, online oracles $\{\mathcal{O}_k\}_{k=1}^K$, step sizes $\eta_k \in (0, 1]$

for $t = 1$ to T **do**
  Initialize arbitrarily $\boldsymbol{x}_{t,1} \in \mathcal{K}$
  for $k = 1$ to K **do**
    Query $\boldsymbol{v}_{t,k}$ from oracle $\mathcal{O}_k$.
    $\boldsymbol{x}_{t,k+1} \leftarrow (1 - \eta_k)\boldsymbol{x}_{t,k} + \eta_k \boldsymbol{v}_{t,k}$.
  end for
  $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_{t,K+1}$, play $\boldsymbol{x}_t$ and incurs loss $f_t(\boldsymbol{x}_t)$
  Receive $\mathcal{F}_t = \{s \in [T] : s + d_s - 1 = t\}$
  if $\mathcal{F}_t = \emptyset$ **then**
    do nothing
  else
    for $k = 1$ to K **do**
      $\boldsymbol{g}_{t,k} \leftarrow \sum_{s \in \mathcal{F}_t} \nabla f_s(\boldsymbol{x}_{s,k})$
      Feedback $\langle \boldsymbol{g}_{t,k}, \cdot \rangle$ to oracles $\mathcal{O}_k$.
    end for
  end if
end for

Lemma 2.3.2 ([Thang2022]). *For every $t \in [T]$ and $k \in [K]$. Define $h_{t,k} = f_t(\boldsymbol{x}_{t,k+1}) - f_t(\boldsymbol{x}^*)$ let $A = \max(3, \frac{G}{\beta D})$ and $\eta_k = \min(1, \frac{A}{k})$, it holds that*

$$h_{t,k} = f_t(\boldsymbol{x}_{t,k+1}) - f_t(\boldsymbol{x}^*) \leq \frac{2\beta AD^2}{k} + \sum_{k'=1}^{k} \eta_{k'} \left(\prod_{\ell=k'+1}^{k} (1-\eta_\ell) \right) \langle \nabla f_{t,k'}, \boldsymbol{v}_{t,k'} - \boldsymbol{x}^* \rangle \tag{2.5}$$

Proof. See Lemma 2.7.1. □

Theorem 2.3.1. *Given a constraint set $\mathcal{K}$. Let $A = \max\left\{3, \frac{G}{\beta D}\right\}$, $\eta_k = \min\left\{1, \frac{A}{k}\right\}$, and $K = \sqrt{T}$. Suppose that Assumptions 1.2.1 and 1.2.2 hold true. If we choose FTPL as the underlying oracle and set $\zeta = \frac{1}{G\sqrt{B}}$, the regret of Algorithm 5 is*

$$\sum_{t=1}^{T} [f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}^*)] \leq 2\beta AD^2\sqrt{T} + 3(A+1)\left(DG\sqrt{B} + \mathcal{R}_{T,\mathcal{O}}\right) \tag{2.6}$$

where $B = \sum_{t=1}^{T} d_t$, the sum of all delay values and $\mathcal{R}_{T,\mathcal{O}}$ is the regret of FTPL with respect to the current choice of ζ.

Proof. Let $\boldsymbol{x}_{t,1}, \ldots, \boldsymbol{x}_{t,K+1}$ be the sequence of sub-iterate for a fixed time step t. Using Frank-Wolfe updates and smoothness of f_t, we have

$$\begin{aligned}
f_t(\boldsymbol{x}_{t,k+1}) - f_t(\boldsymbol{x}^*) &= f_t(\boldsymbol{x}_{t,k} + \eta_k(\boldsymbol{v}_{t,k} - \boldsymbol{x}_{t,k})) - f_t(\boldsymbol{x}^*) \\
&\leq f_t(\boldsymbol{x}_{t,k}) - f_t(\boldsymbol{x}^*) + \eta_k \langle \nabla f_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}_{t,k} \rangle + \eta_k^2 \frac{\beta}{2} \|\boldsymbol{v}_{t,k} - \boldsymbol{x}_{t,k}\|^2 \\
&\leq f_t(\boldsymbol{x}_{t,k}) - f_t(\boldsymbol{x}^*) + \eta_k \langle \nabla f_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}_{t,k} \rangle + \eta_k^2 \frac{\beta D^2}{2} \quad (\mathcal{K} \text{ is bounded}) \\
&\leq f_t(\boldsymbol{x}_{t,k}) - f_t(\boldsymbol{x}^*) + \eta_k [\langle \nabla f_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^* \rangle + \langle \nabla f_{t,k}, \boldsymbol{x}^* - \boldsymbol{x}_{t,k} \rangle] + \eta_k^2 \frac{\beta D^2}{2} \\
&\leq f_t(\boldsymbol{x}_{t,k}) - f_t(\boldsymbol{x}^*) + \eta_k [\langle \nabla f_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^* \rangle + f_t(\boldsymbol{x}^*) - f_t(\boldsymbol{x}_{t,k})] + \eta_k^2 \frac{\beta D^2}{2} \\
&\leq (1-\eta_k) [f_t(\boldsymbol{x}_{t,k}) - f_t(\boldsymbol{x}^*)] + \eta_k \langle \nabla f_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^* \rangle + \eta_k^2 \frac{\beta D^2}{2}
\end{aligned} \tag{2.7}$$

Let $h_{t,k} = f_t(\boldsymbol{x}_{t,k+1}) - f_t(\boldsymbol{x}^*)$, equation (2.7) becomes

$$h_{t,k} \leq (1-\eta_k) h_{t,k-1} + \eta_k \langle \nabla f_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^* \rangle + \eta_k^2 \frac{\beta D^2}{2} \tag{2.8}$$

A direct application of Lemma 2.3.2 for $k = K$ yields

$$f_t(\boldsymbol{x}_{t,K+1}) - f_t(\boldsymbol{x}^*) \leq \frac{2\beta AD^2}{K} + \sum_{k'=1}^{K} \eta_{k'} \left[\prod_{\ell=k+1}^{K} (1-\eta_\ell) \right] \langle \nabla f_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^* \rangle \tag{2.9}$$

Following the notation from *Algorithm* 5 and Lemma 2.3.1. For a fixed time t and any sub-iterate k, $\boldsymbol{v}_{t,k}$ and $\hat{\boldsymbol{v}}_{t,k}$ are respectively the predictions of the oracle $\mathcal{O}_k$ under delayed and non-delayed feedback, the scalar product of equation (2.9) over T-round is written as

$$\begin{aligned}
\sum_{t=1}^{T} \langle \nabla f_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^* \rangle &= \sum_{t=1}^{T} \langle \nabla f_{t,k}, \boldsymbol{v}_{t,k} - \hat{\boldsymbol{v}}_{t,k} \rangle + \sum_{t=1}^{T} \langle \nabla f_{t,k}, \hat{\boldsymbol{v}}_{t,k} - \boldsymbol{x}^* \rangle \\
&\leq \sum_{t=1}^{T} \langle \nabla f_{t,k}, \boldsymbol{v}_{t,k} - \hat{\boldsymbol{v}}_{t,k} \rangle + \sum_{t=1}^{T} \langle \nabla f_{t,k}, \hat{\boldsymbol{v}}_{t,k} - \boldsymbol{x}^* \rangle
\end{aligned} \tag{2.10}$$

In the first term on the right hand side of Equation (2.10), using the Cauchy-Schwartz inequality and Lemma 2.3.1, we have

$$\sum_{t=1}^{T} \langle \nabla f_{t,k}, \boldsymbol{v}_{t,k} - \hat{\boldsymbol{v}}_{t,k} \rangle \leq \sum_{t=1}^{T} \|\nabla f_{t,k}\| \, \|\boldsymbol{v}_{t,k} - \hat{\boldsymbol{v}}_{t,k}\| \leq \zeta DG^2 \sum_{t=1}^{T} \sum_{s<t} \mathbb{I}_{s+d_s>t} \leq \zeta DG^2 B \tag{2.11}$$

Recall that the objective function of the oracle $\mathcal{O}_k$ in non-delay setting is $\langle \nabla f_{t,k}, \cdot \rangle$ and $\hat{\boldsymbol{v}}_{t,k}$ is its prediction at time t, the second term of Equation (2.10) is bounded by the regret of $\mathcal{O}_k$ in non-delay setting, $\mathcal{R}_{T,\mathcal{O}}$. Specifically, we have

$$\sum_{t=1}^{T} \langle \nabla f_{t,k}, \hat{\boldsymbol{v}}_{t,k} \rangle \leq \min_{\boldsymbol{x} \in \mathcal{K}} \sum_{t=1}^{T} \langle \nabla f_{t,k}, \boldsymbol{x} \rangle + \mathcal{R}_{T,\mathcal{O}} \leq \sum_{t=1}^{T} \langle \nabla f_{t,k}, \boldsymbol{x}^* \rangle + \mathcal{R}_{T,\mathcal{O}} \tag{2.12}$$

Recall that $\boldsymbol{x}_t = \boldsymbol{x}_{t,K+1}$, combining Equations (2.10) to (2.12) and summing Equation (2.9) over T-rounds yields

$$\sum_{t=1}^{T} [f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}^*)] \leq \frac{2\beta A D^2}{K} T + \sum_{k'=1}^{K} \eta_{k'} \left[\prod_{\ell=k+1}^{K} (1 - \eta_\ell) \right] \left[\zeta D G^2 B + \mathcal{R}_{T,\mathcal{O}} \right] \tag{2.13}$$

Let $\eta_k = \frac{A}{k}$, we have

$$\prod_{k'=k+1}^{K} (1 - \eta_{k'}) \leq e^{-\sum_{k'=k+1}^{K} \eta_{k'}} \leq e^{-\sum_{k'=\ell}^{K} \frac{A}{k'}} \leq e^{-A \int_{k+2}^{K} ds/s} \leq \left(\frac{k+2}{K} \right)^A \tag{2.14}$$

We have then,

$$\begin{aligned}
&\sum_{k=1}^{K} \eta_k \left[\prod_{k'=k+1}^{K} (1 - \eta_{k'}) \right] \leq \min \left\{ 1, \frac{A}{K} \right\} + \min \left\{ 1, \frac{A}{K-1} \right\} + \min \left\{ 1, \frac{A}{K-2} \right\} + \sum_{k=1}^{K-3} \frac{A}{k} \left[\frac{k+2}{K} \right]^A \\
\leq & 3 \min \left\{ 1, \frac{A}{K-2} \right\} + \frac{A}{K} \sum_{k=1}^{K-3} \frac{k+2}{k} \left[\frac{k+2}{K} \right]^{A-1} \leq 3 + \frac{3A}{K} \sum_{k=1}^{K-3} \left[\frac{k+2}{K} \right]^{A-1} \leq 3(A+1)
\end{aligned} \tag{2.15}$$

From Equation (2.15), we deduce that

$$\begin{aligned}
\sum_{t=1}^{T} [f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}^*)] &\leq \frac{2\beta A D^2}{K} T + \sum_{k'=1}^{K} \eta_{k'} \left(\prod_{\ell=k+1}^{K} (1 - \eta_\ell) \right) \left(\zeta D G^2 B + \mathcal{R}_{T,\mathcal{O}} \right) \\
&\leq \frac{2\beta A D^2}{K} T + 3(A+1) \left(\zeta D G^2 B + \mathcal{R}_{T,\mathcal{O}} \right)
\end{aligned} \tag{2.16}$$

The theorem follows by letting $\zeta = \frac{1}{G\sqrt{B}}$, $K = \sqrt{T}$ and choosing the oracle as FTPL with regret $\mathcal{R}_{T,\mathcal{O}}$. □

Discussion The regret bound of Theorem 2.3.1 differs from that of the non-delayed MFW [Chen2018a] by the additive term $DG\sqrt{B}$ which represents the total cost of sending delayed feedback to the oracles over T rounds (Lemma 2.3.1). If we assume that there exists a maximum value d such that $d_t \leq d$ for all $t \in [T]$. Our regret bound becomes $O(\sqrt{dT})$ which coincides with the setting in [Wan2022a], a delayed-feedback FW algorithm that achieves $O(T^{3/4} + dT^{1/4})$. Another line of work is from [Joulani2013], a framework that addresses delayed feedback for any base algorithm. By considering MFW as the base algorithm, their theoretical analysis suggests that the algorithm also achieves $O(\sqrt{dT})$ regret bound. However, their delay value is not completely unknown to the agent because it is time-stamped by maintaining multiple copies of the base algorithm. We empirically show in Section 2.5 that this algorithm is highly susceptible to high delay values. Instead of using FTPL, our algorithm has the flexibility to select any online algorithm as an oracle, for example, Online Gradient Descent [Hazan2016a].

2.4 Distributed Algorithm

In this section, we extend Algorithm 5 to a distributed setting in which multiple agents collaboratively optimize a global model. Our setting considers a fully distributed framework, characterized

by the absence of a server to coordinate the learning process. The setup is outlined as described in Section 1.2.

At a high level, each agent maintains K copies of the oracles $\mathcal{O}_1^i, \cdots, \mathcal{O}_K^i$ while performing prediction and update at every round t. The prediction block consists of performing K FW-steps while incorporating the neighbors' information. Specifically, the agent computes at its local level during the K steps a local average decision $\boldsymbol{y}_{t,k}^i$ representing a weighted aggregation of its neighbor's current sub-iterates. The update vector is convex combination of the local average decision and the oracle's output. The final decision of agent $\boldsymbol{x}_t^i$ is disclosed at the end of K steps. Lemma 2.7.2 shows that $\boldsymbol{y}_{t,k}^i$ is a local estimation of the global average $\overline{\boldsymbol{x}}_{t,k} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_{t,k}^i$ as K increases.

Following the K FW-steps, the update block employs K gradient updates utilizing the delayed feedback from previous rounds. The agent observes the delayed gradients evaluated on theirs corresponding subiterates and computes the local average gradient $\boldsymbol{d}_{t,k}^i$ through a weighted aggregation of the neighbors' current surrogates (18). The agent updates the surrogate gradient via a gradient-tracking step (19) to ensure that it approaches the global gradient as K increases. It is worth noting that feedback provided to the oracle contains information about delays experienced by all neighboring agents. Consequently, the oracle $\mathcal{O}_k^i$ observes delayed feedback from $\cup_{j\in\mathcal{N}(i)}\mathcal{F}_t^j$ instead of $\mathcal{F}_t^i$. This result highlights the dependency on the connectivity of the communication graph when considering the effect of delayed feedback to the oracle's output.

Theorem 2.4.1. *Given a constraint set $\mathcal{K}$. Let $A = \max\left\{3, \frac{3G}{2\beta D}, \frac{2\beta C_d + C_g}{\beta D}\right\}$, $\eta_k = \min\left\{1, \frac{A}{k}\right\}$, and $K = \sqrt{T}$. Suppose that Assumptions 1.2.1 and 1.2.2 hold true. If we choose FTPL as the underlying oracle and set $\zeta = \frac{1}{G\sqrt{B}}$, the regret of Algorithm 6 is*

$$\sum_{t=1}^{T}\left[F_t(\boldsymbol{x}_t^i) - F_t(\boldsymbol{x}^*)\right] \leq \left(GC_d + 2\beta AD^2\right)\sqrt{T} + 3(A+1)\left(2\sqrt{n}DG\left(\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})}+1\right)\sqrt{B} + \mathcal{R}_{T,\mathcal{O}}\right)$$

where $B = \frac{1}{n}\sum_{i=1}^n B_i$ such that B_i is the sum of all delay values of agent i. $C_d = k_0\sqrt{n}D$ and $C_g = \sqrt{n}\max\left\{\lambda(\mathbf{W})\left(G + \frac{\beta D}{1-\lambda(\mathbf{W})}\right), k_0\beta\left(4C_d + AD\right)\right\}$ and $\mathcal{R}_{T,\mathcal{O}}$ is the regret of FTPL with respect to the current choice of ζ.

2.4.1 Technical Analysis

Before proceeding to analysis, we introduce some additional notations that will be used specifically for distributed setting. For any vector $\boldsymbol{x}^i \in \mathbb{R}^d, \forall i \in [n]$, we note $\boldsymbol{x}^{cat} \in \mathbb{R}^{dn}$ a column vector defined as $\boldsymbol{x}^{cat} := \left[\boldsymbol{x}^{1\top}, \ldots, \boldsymbol{x}^{n\top}\right]^\top$. Let $\overline{\boldsymbol{x}}$ be the average of $\boldsymbol{x}^i$ over $i \in [n]$, the vector $\overline{\boldsymbol{x}}^{cat}$ is a dn-vector where we stack n-times $\overline{\boldsymbol{x}}$ i.e $\overline{\boldsymbol{x}}^{cat} := \left[\overline{\boldsymbol{x}}^\top, \ldots, \overline{\boldsymbol{x}}^\top\right]$. For simplicity of notation, we note $\nabla f_t^i(\boldsymbol{x}_{t,k}^i) := \nabla f_{t,k}^i$ and $\nabla F_{t,k} := \frac{1}{n}\sum_{i=1}^n \nabla f_{t,k}^i$. In order to incorporate the delay of agents at each time-step t, we define $\nabla f_{t,k}^{cat}$ as described above using the sum of agent's delay feedback, we note then

$$\nabla f_{t,k}^{cat} = \left[\sum_{s\in\mathcal{F}_t^1}\nabla f_{s,k}^{1\top}, \ldots, \sum_{s\in\mathcal{F}_t^n}\nabla f_{s,k}^{n\top}\right]^\top \tag{2.17}$$

and its homologous in the non-delay setting by $\nabla \hat{f}_{t,k}^{cat} = \left[\nabla f_{t,k}^{1\top}, \ldots, \nabla f_{t,k}^{n\top}\right]^\top$. The variables $\boldsymbol{d}_{t,k}^{cat}, \hat{\boldsymbol{d}}_{t,k}^{cat}$ and $\boldsymbol{g}_{t,k}^{cat}, \hat{\boldsymbol{g}}_{t,k}^{cat}$ are defined similarly as described. Lastly, we define the slack variable $\boldsymbol{\delta}_{t,k}^i := \nabla f_{t,k}^i - \nabla f_{t,k-1}^i$, then the definition of $\overline{\boldsymbol{\delta}}_{t,k}, \boldsymbol{\delta}_{t,k}^{cat}$ and $\overline{\boldsymbol{\delta}}_{t,k}^{cat}$ followed.

The proof of Theorem 2.4.1 is structured as follows:

- In Lemma 2.4.3, we establish a bound on the distance between the output of the FTPL oracle with delayed feedback and its counterpart in the non-delayed setting in the distributed case.
- We utilize two important results, Lemmas 2.4.1 and 2.4.2, to bound the decreasing distance between the decision/gradient local estimates and the global average.

- The main derivation of the proof combines Lemmas 2.4.1 and 2.4.2 and uses Lemma 2.3.2 to establish an upper bound on the primal gap at time t (2.22). We then apply Lemma 2.4.3 and the regret of the oracle in the non-delay setting to complete the main part (2.23).
- The final derivation directly applies proposition 2.7.1 to bound the agent regret (2.24).

The detailed proof of the theorem is provided in the following section. We postpone some missing proofs to the end of this chapter.

Lemma 2.4.1. *Define $C_d = k_0\sqrt{n}D$, for all $t \in [T]$, $k \in [K]$, we have*

$$\max_{i\in[1,n]} \left\|\boldsymbol{y}_{t,k}^i - \overline{\boldsymbol{x}}_{t,k}\right\| \leq \frac{C_d}{k} \tag{2.18}$$

Proof. See Lemma 2.7.2. □

Lemma 2.4.2. *Define $C_g = \sqrt{n}\max\left\{\lambda(\mathbf{W})\left(G + \frac{\beta D}{1-\lambda(\mathbf{W})}\right), k_0\beta\left(4C_d + AD\right)\right\}$ and recall the definition of $\nabla F_{t,k} := \frac{1}{n}\sum_{i=1}^n \nabla f_{t,k}^i$. For all $t \in [T]$, $k \in [K]$, we have*

$$\max_{i\in[1,n]} \left\|\boldsymbol{d}_{t,k}^i - \nabla F_{t,k}\right\| \leq \frac{C_g}{k} \tag{2.19}$$

Proof. See Lemma 2.7.3. □

Lemma 2.4.3. *For all $t \in [T]$, $k \in [K]$ and $i \in [n]$. Let $\boldsymbol{v}_{t,k}^i$ be the output of the oracle $\mathcal{O}_k^i$ with delayed feedback and $\hat{\boldsymbol{v}}_{t,k}^i$ its homologous in non-delay case. Suppose that Assumptions 1.2.1 and 1.2.2 hold true. Choosing FTPL as the oracle, we have:*

$$\left\|\boldsymbol{v}_{t,k}^i - \hat{\boldsymbol{v}}_{t,k}^i\right\| \leq 2\zeta\sqrt{n}DG\left[\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})} + 1\right]\frac{1}{n}\sum_{i=1}^n\sum_{s\leq t}\mathbb{I}_{\{s+d_s^i>t\}} \tag{2.20}$$

where ζ is the learning rate, $\lambda(\mathbf{W})$ is the second-largest eigenvalue of $\mathbf{W}$.

Algorithm 6 De2MFW

Input: Constraint set $\mathcal{K}$, number of iterations T, sub-iterations K, online linear optimization oracles $\left\{\mathcal{O}_k^i : k \in [K]\right\}$ for each agent $i \in [n]$, step sizes $\eta_k \in (0, 1]$

for $t = 1$ to T **do**
  for every agent $i = 1$ to n **do**
    Initialize arbitrarily $\boldsymbol{x}_{t,1}^i \in \mathcal{K}$
    for $k = 1$ to K **do**
      Query $\boldsymbol{v}_{t,k}^i$ from oracle $\mathcal{O}_k^i$
      Exchange $\boldsymbol{x}_{t,k}^i$ with neighbours $\mathcal{N}(i)$
      $\boldsymbol{y}_{t,k}^i \leftarrow \sum_j w_{ij}\boldsymbol{x}_{t,k}^j$
      $\boldsymbol{x}_{i,k+1}^t \leftarrow (1-\eta_k)\boldsymbol{y}_{t,k}^i + \eta_k\boldsymbol{v}_{t,k}^i$
    end for
    $\boldsymbol{x}_t^t \leftarrow \boldsymbol{x}_{t,K+1}^t$, play $\boldsymbol{x}_t^i$ and incurs loss $f_t^i(\boldsymbol{x}_t^i)$
    Receive $\mathcal{F}_t^i = \left\{s \in [T] : s + d_s^i - 1 = t\right\}$
    if $\mathcal{F}_t^i = \emptyset$ **then**
      do nothing
    else
      $\boldsymbol{g}_{t,1}^i \leftarrow \sum_{s\in\mathcal{F}_t^i}\nabla f_s^i(\boldsymbol{x}_{s,1}^i)$
      for $k = 1$ to K **do**
        Exchange $\boldsymbol{g}_{t,k}^i$ with neighbours $\mathcal{N}(i)$
        $\boldsymbol{d}_{t,k}^i \leftarrow \sum_{j\in\mathcal{N}(i)} w_{ij}\boldsymbol{g}_{t,k}^j$
        $\boldsymbol{g}_{t,k+1}^i \leftarrow \sum_{s\in\mathcal{F}_t^i}\left(\nabla f_s^i(\boldsymbol{x}_{s,k+1}^i) - \nabla f_s^i(\boldsymbol{x}_{s,k}^i)\right) + \boldsymbol{d}_{t,k}^i$
        Feedback $\langle \boldsymbol{d}_{t,k}^i, \cdot\rangle$ to oracles $\mathcal{O}_{i,k}$
      end for
    end if
  end for
end for

Proof. See Lemma 2.7.4. □

2.4.2 Proof of Theorem 2.4.1

Proof. Using smoothness, convexity of F_t and Frank-Wolfe updates, we have

$$
\begin{aligned}
&F_t(\overline{\boldsymbol{x}}_{t,k+1}) - F_t(\boldsymbol{x}^*) = F_t\left(\overline{\boldsymbol{x}}_{t,k} + \eta_k\left(\frac{1}{n}\sum_{i=1}^n \boldsymbol{v}_{t,k}^i - \overline{\boldsymbol{x}}_{t,k}\right)\right) - F_t(\boldsymbol{x}^*)\\
\leq& F_t(\overline{\boldsymbol{x}}_{t,k}) - F_t(\boldsymbol{x}^*) + \eta_k\left\langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \frac{1}{n}\sum_{i=1}^n \boldsymbol{v}_{t,k}^i - \overline{\boldsymbol{x}}_{t,k}\right\rangle + \eta_k^2\frac{\beta}{2}\left\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{v}_{t,k}^i - \overline{\boldsymbol{x}}_{t,k}\right\|^2\\
\leq& F_t(\overline{\boldsymbol{x}}_{t,k}) - F_t(\boldsymbol{x}^*) + \frac{\eta_k}{n}\left\langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \boldsymbol{v}_{t,k}^i - \overline{\boldsymbol{x}}_{t,k}\right\rangle + \eta_k^2\frac{\beta D^2}{2} \qquad (2.21)\\
\leq& F_t(\overline{\boldsymbol{x}}_{t,k}) - F_t(\boldsymbol{x}^*) + \frac{\eta_k}{n}\sum_{i=1}^n\left[\left\langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \boldsymbol{v}_{t,k}^i - \boldsymbol{x}^*\right\rangle + \left\langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \boldsymbol{x}^* - \overline{\boldsymbol{x}}_{t,k}\right\rangle\right] + \eta_k^2\frac{\beta D^2}{2}\\
\leq& (1-\eta_k)\left[F_t(\overline{\boldsymbol{x}}_{t,k}) - F_t(\boldsymbol{x}^*)\right] + \frac{\eta_k}{n}\sum_{i=1}^n\left[\left\langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \boldsymbol{v}_{t,k}^i - \boldsymbol{x}^*\right\rangle\right] + \eta_k^2\frac{\beta D^2}{2}\\
\leq& (1-\eta_k)\left[F_t(\overline{\boldsymbol{x}}_{t,k}) - F_t(\boldsymbol{x}^*)\right] + \frac{\eta_k}{n}\sum_{i=1}^n\left[\left\langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \boldsymbol{v}_{t,k}^i - \hat{\boldsymbol{v}}_{t,k}^i\right\rangle + \left\langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \hat{\boldsymbol{v}}_{t,k}^i - \boldsymbol{x}^*\right\rangle\right] + \eta_k^2\frac{\beta D^2}{2}\\
\leq& (1-\eta_k)\left[F_t(\overline{\boldsymbol{x}}_{t,k}) - F_t(\boldsymbol{x}^*)\right] + \frac{\eta_k}{n}\sum_{i=1}^n\left[\left\langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \boldsymbol{v}_{t,k}^i - \hat{\boldsymbol{v}}_{t,k}^i\right\rangle + \left\langle \hat{\boldsymbol{d}}_{t,k}^i, \hat{\boldsymbol{v}}_{t,k}^i - \boldsymbol{x}^*\right\rangle\right] + \eta_k\frac{2\beta C_d + C_g}{k}D + \eta_k^2\frac{\beta D^2}{2}
\end{aligned}
$$

where the last inequality followed by observing that

$$
\begin{aligned}
&\left\langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \hat{\boldsymbol{v}}_{t,k}^i - \boldsymbol{x}^*\right\rangle = \left\langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}) - \nabla F_{t,k}, \hat{\boldsymbol{v}}_{t,k}^i - \boldsymbol{x}^*\right\rangle + \left\langle \nabla F_{t,k}, \hat{\boldsymbol{v}}_{t,k}^i - \boldsymbol{x}^*\right\rangle\\
&= \left\langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}) - \nabla F_{t,k}, \hat{\boldsymbol{v}}_{t,k}^i - \boldsymbol{x}^*\right\rangle + \left\langle \nabla F_{t,k} - \hat{\boldsymbol{d}}_{t,k}^i, \hat{\boldsymbol{v}}_{t,k}^i - \boldsymbol{x}^*\right\rangle + \left\langle \hat{\boldsymbol{d}}_{t,k}^i, \hat{\boldsymbol{v}}_{t,k}^i - \boldsymbol{x}^*\right\rangle\\
&\leq \left(\beta\left\|\overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}_{t,k}^i\right\| + \left\|\nabla F_{t,k} - \hat{\boldsymbol{d}}_{t,k}^i\right\|\right)D + \left\langle \hat{\boldsymbol{d}}_{t,k}^i, \hat{\boldsymbol{v}}_{t,k}^i - \boldsymbol{x}^*\right\rangle\\
&\leq \frac{2\beta C_d + C_g}{k}D + \left\langle \hat{\boldsymbol{d}}_{t,k}^i, \hat{\boldsymbol{v}}_{t,k}^i - \boldsymbol{x}^*\right\rangle
\end{aligned}
$$

where the last two steps are direct application of Lemmas 2.4.1 and 2.4.2. Let $A = \max\left\{3, \frac{3G}{2\beta D}, \frac{2\beta C_d + Cg}{\beta D}\right\}$ and $\eta_k = \frac{A}{k}$, from Lemma 2.3.2 we have

$$
F_t(\overline{\boldsymbol{x}}_{t,K+1}) - F_t(\boldsymbol{x}^*) \leq \frac{2\beta A D^2}{K} + \frac{1}{n}\sum_{i=1}^n\sum_{k=1}^K \eta_k\left[\prod_{\ell=k+1}^K(1-\eta_\ell)\right]\left[\left\langle \hat{\boldsymbol{d}}_{t,k}^i, \hat{\boldsymbol{v}}_{t,k}^i - \boldsymbol{x}^*\right\rangle + \left\langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \boldsymbol{v}_{t,k}^i - \hat{\boldsymbol{v}}_{t,k}^i\right\rangle\right] \qquad (2.22)
$$

Summing equation (2.22) over T-rounds yields,

$$
\begin{aligned}
&\sum_{t=1}^T\left[F_t(\overline{\boldsymbol{x}}_t) - F_t(\boldsymbol{x}^*)\right] \leq \frac{2\beta A D^2 T}{K} + \frac{1}{n}\sum_{i=1}^n\sum_{k=1}^K \eta_k\prod_{\ell=k+1}^K(1-\eta_\ell)\sum_{t=1}^T\left[\left\langle \hat{\boldsymbol{d}}_{t,k}^i, \hat{\boldsymbol{v}}_{t,k}^i - \boldsymbol{x}^*\right\rangle + \left\langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \boldsymbol{v}_{t,k}^i - \hat{\boldsymbol{v}}_{t,k}^i\right\rangle\right]\\
\leq& \frac{2\beta A D^2 T}{K} + \frac{1}{n}\sum_{i=1}^n\sum_{k=1}^K \eta_k\prod_{\ell=k+1}^K(1-\eta_\ell)\left[\mathcal{R}_{T,\mathcal{O}} + 2\zeta\sqrt{n}DG^2\left(\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})}+1\right)\frac{1}{n}\sum_{i=1}^n\sum_{t=1}^T\sum_{s\leq t}\mathbb{I}_{\{s+d_s^i>t\}}\right]\\
\leq& \frac{2\beta A D^2 T}{K} + 3(A+1)\left[\mathcal{R}_{T,\mathcal{O}} + 2\zeta\sqrt{n}DG^2\left(\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})}+1\right)\frac{1}{n}\sum_{i=1}^n\sum_{t=1}^T\sum_{s\leq t}\mathbb{I}_{\{s+d_s^i>t\}}\right] \qquad (2.23)
\end{aligned}
$$

From equation (2.23), we deduce that, for all $i \in [n]$,

$$\begin{aligned}
&\sum_{t=1}^{T}\left[F_t(\boldsymbol{x}_t^i) - F_t(\boldsymbol{x}^*)\right] \leq \sum_{t=1}^{T}\left[F_t(\boldsymbol{x}_t^i) - F_t(\overline{\boldsymbol{x}}_t)\right] + \sum_{t=1}^{T}\left[F_t(\overline{\boldsymbol{x}}_t) - F_t(\boldsymbol{x}^*)\right] \\
\leq &\sum_{t=1}^{T} G\left\|\boldsymbol{x}_t^i - \overline{\boldsymbol{x}}_t\right\| + \sum_{t=1}^{T}\left[F_t(\overline{\boldsymbol{x}}_t) - F_t(\boldsymbol{x}^*)\right] \\
\leq &\frac{GC_dT}{K} + \sum_{t=1}^{T}\left[F_t(\overline{\boldsymbol{x}}_t) - F_t(\boldsymbol{x}^*)\right] \\
\leq &\frac{GC_d + 2\beta AD^2}{K}T + 3(A+1)\left[\mathcal{R}_{T,\mathcal{O}} + 2\zeta\sqrt{n}DG^2\left(\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})} + 1\right)\frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\sum_{s<t}\mathbb{I}_{\{s+d_s^i>t\}}\right] \\
\leq &\frac{GC_d + 2\beta AD^2}{K}T + 3(A+1)\left[\mathcal{R}_{T,\mathcal{O}} + 2\zeta\sqrt{n}DG^2\left(\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})} + 1\right)\frac{1}{n}\sum_{i=1}^{n}B_i\right]
\end{aligned} \tag{2.24}$$

The theorem follows by letting $\zeta = \frac{1}{G\sqrt{B}}$, $K = \sqrt{T}$ and $B = \frac{1}{n}\sum_{i=1}^{n} B_i$. This concludes the proof. □

2.5 Numerical Experiments

We evaluated the performance of our algorithms on the online multiclass logistic regression problem using two datasets: MNIST and FashionMNIST. MNIST is a well-known hand digit dataset containing 60000 grayscale images of size (28×28), divided into 10 classes, and FashionMNIST includes images of fashion products with the same configuration. We conducted the experiment using Julia 1.7 on MacOS 13.3 with 16GB of memory.

Centralized Setting Given an iteration t, the agent receives a subset $\mathcal{B}_t$ of the form $\boldsymbol{b}_t = \{\boldsymbol{a}_t, y_t\} \in \mathbb{R}^m \times \{1, \ldots, C\}$, consisting of the features vector $\boldsymbol{a}_t$ and the corresponding label y_t. We define the loss function f_t as

$$f_t(\boldsymbol{x}) = -\sum_{\boldsymbol{b}_t \in \mathcal{B}_t}\sum_{c=1}^{C}\left\{y_t^i = c\right\}\log\frac{\exp\left\langle \boldsymbol{x}_c, \boldsymbol{a}_t^i\right\rangle}{\sum_{\ell=1}^{C}\exp\left\langle \boldsymbol{x}_\ell, \boldsymbol{a}_t^i\right\rangle} \tag{2.25}$$

where $\boldsymbol{x}$ must satisfy the constraint $\boldsymbol{x} \in \mathcal{K}$ such that $\mathcal{K} = \left\{\boldsymbol{x} \in \mathbb{R}^{m\times C}, \|\boldsymbol{x}\|_1 \leq r\right\}$. Using the MNIST dataset, we note $m = 784$, $C = 10$, $r = 8$, $|\mathcal{B}_t| = 60$ and a total of $T = 1000$ rounds. To evaluate the performance of the algorithm under different delay regimes, we generated a random sequence of delays d_t such that $d_t \leq d$ for $d \in \{21, 41, 61, 81, 101\}$. We compared the performance of DeLMFW against DOFW [Wan2022a], a projection-free algorithm with adversarial delay, and BOLD-MFW [Joulani2013], an online learning framework designed to handle delayed feedback. Figure 2.2 displays the performance of the three algorithms under various delay regimes. In the absence of delay, that is, $d = 1$ (left figure), DeLMFW and BOLD-MFW have the same performance since both algorithms reduce to MFW [Chen2018a] with a regret of $O(\sqrt{T})$. Meanwhile, DOFW is the classical OFW [Hazan2012] that guarantees a regret of $O(T^{3/4})$. The analysis in Theorem 2.3.1 suggests that DeLMFW achieves a regret of $O(\sqrt{dT})$ when the delay is upper-bounded by d. In the case where $d \leq T^{1/2}$ (middle figure, $d = 21$), the dominant term in DOFW is $T^{3/4}$ whereas DeLMFW takes advantage by incurring a regret of order $\sqrt{dT} \leq T^{3/4}$. For $d \geq T^{1/2}$ (right figure, $d = 101$), DOFW's regret is dominated by the term $dT^{1/4}$, which is outperformed by DeLMFW, particularly for high values of d. This result confirms our theoretical analysis in Section 2.3.

Figure 2.3 illustrates the total loss of DeLMFW and the other two algorithms when increasing d to show the sensitivity of each algorithm in the presence of delays. As BOLD is a general framework

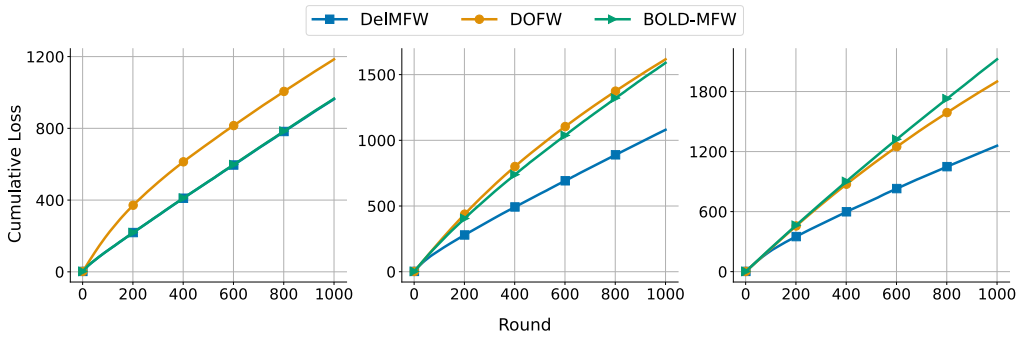


Figure 2.2: Cumulative Loss Comparison for Different Delays Regimes. Left : Without delay. Middle : Maximal delay 21. Right : Maximal delay 101

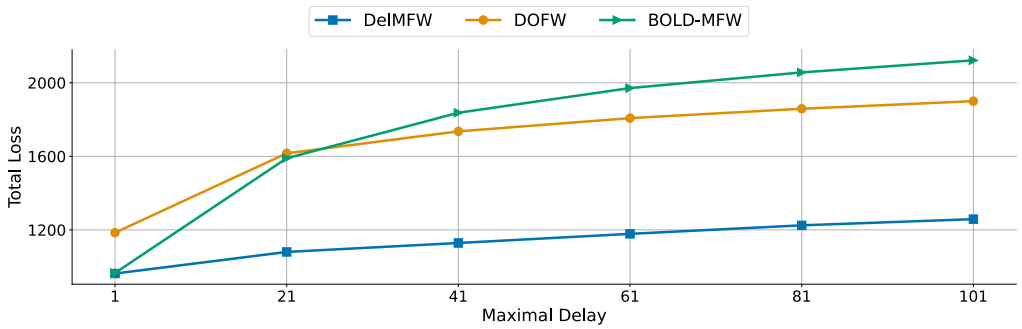


Figure 2.3: Total loss of BOLD-MFW, DOFW and DeLMFW when varying delay value.

that can be applied to any base algorithm, it is noticeable that it is susceptible to high levels of delays. This phenomenon has also been observed in [Wan2022a] when utilizing BOLD with OFW, highlighting the need for a customized design algorithms in the context of delayed feedback.

Distributed Setting In the second experiment, we examined the distributed online multiclass logistic regression problem on the FashionMNIST dataset, using a network of 30 agents. The algorithm was run on four different topologies, including Erdos-Renyi, Complete, Grid, and Cycle. At each iteration $t \in [T]$, each agent i received a subset $\mathcal{B}_t^i$ of the form $\{\boldsymbol{a}_t^i, y_t^i\} \in \mathbb{R}^d \times \{1, \ldots, C\}$, which consisted of the feature vector $\boldsymbol{a}_t^i$ and its corresponding label y_t^i. The goal was to collaboratively optimize the global loss function $F_t(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^n f_t^i(\boldsymbol{x})$, where the local loss f_t^i was defined in Equation (2.25).

For this experiment, we set $m = 784$, $C = 10$, $r = 32$, $|\mathcal{B}_t^i| = 2$ and $T = 1000$ rounds. We are interested in examining the effect of delays on network performance, and thus randomly select $f < n$ agents to have delayed feedback with a maximum value of 501. We compared the total loss on each topology under these conditions, and present the result in Figure 2.4. We observe that the presence of delayed agents has a significant impact on the network performance of Cycle graph as the number of delayed agents increases, while the Complete graph is less affected. This result is consistent with the analysis in Section 2.4 because the delay term in the regret bound depends on the connectivity of the communication graph.

In Table 2.2, we report the change in total loss when increasing the number of delayed agents. We observe that the average percentage change is smaller for Grid than for Erdos-Renyi when compared with the network of non-delayed agents ($f = 0$). This result indicate that the generated Erdos-Renyi graph is more sensitive to the presence of delayed agents.

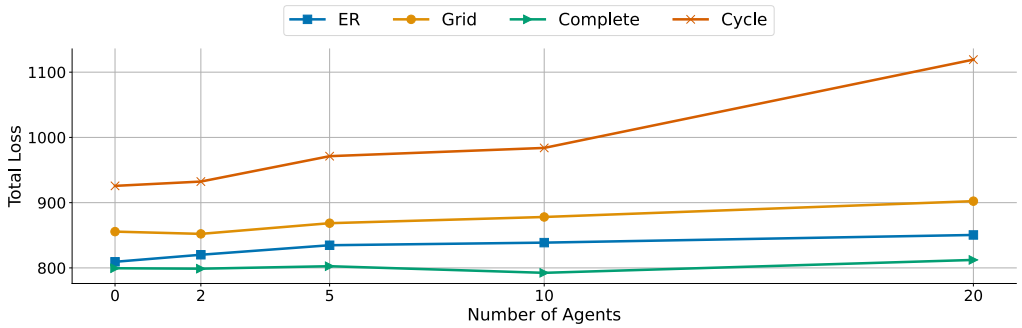


Figure 2.4: Total Loss with varying numbers of agents experiencing delayed feedback in the network. ($f = 0$) for no delayed-agents.

Tableau 2.2: Total Loss of the algorithm running on 4 different topology. We randomly select $f < n$ agents to have delay with maximal value to be 501. In parenthesis, the percentage of total loss compared that of no delayed agents in the network (i.e $f = 0$).

Topology / f	**Erdős-Rényi**	**Grid**	**Complete**	**Cycle**
0	809.37	855.62	799.49	925.72
2	820.15 (+1.3%)	852.15 (-0.4%)	798.79 (-0.08%)	932.34 (+0.7%)
5	834.74 (+3.0%)	868.52 (+1.4%)	802.59 (+0.3%)	971.24 (+4.7%)
10	838.74 (+3.5%)	878.04 (+2.5%)	792.45 (-0.8%)	983.89 (+6.0%)
20	850.49 (+4.9%)	902.30 (+5.3%)	812.21 (+1.5%)	1119.24 (+18.9%)

2.6 Concluding Remarks

In this chapter, we propose two algorithms for solving the online convex optimization problem with adversarial delayed feedback in both centralized and decentralized settings. These algorithms achieve optimal $O(\sqrt{dT})$ regret bounds, where d is the upper bound of the delays. The experimental results show that our algorithms outperform existing solutions in both centralized and decentralized settings, which are predictable by our theoretical analysis. Although the algorithms achieve good performance guarantees for the online convex optimization problem with adversarial delays, they currently rely on exact gradients, which may not be feasible for many real-world applications. Therefore, future research could explore the use of stochastic gradients with variance reduction techniques. Additionally, in decentralized settings, communication delays can be practically challenging, and further improvements are needed in this area. Nevertheless, our work demonstrates the potential of using Frank-Wolfe-type algorithms for solving constraint convex optimization problems under adversarial delays, which is beneficial for learning on edge devices.

2.7 Missing proofs of Chapter 2

Lemma 2.7.1 ([Thang2022]). *For every $t \in [T]$ and $k \in [K]$. Define $h_{t,k} = f_t(\boldsymbol{x}_{t,k+1}) - f_t(\boldsymbol{x}^*)$ let $A = \max(3, \frac{G}{\beta D})$ and $\eta_k = \min(1, \frac{A}{k})$, it holds that*

$$h_{t,k} = f_t(\boldsymbol{x}_{t,k+1}) - f_t(\boldsymbol{x}^*) \leq \frac{2\beta AD^2}{k} + \sum_{k'=1}^{k} \eta_{k'} \left(\prod_{\ell=k'+1}^{k} (1-\eta_\ell) \right) \langle \nabla f_{t,k'}, \boldsymbol{v}_{t,k'} - \boldsymbol{x}^* \rangle \tag{2.26}$$

Proof. The proof is based on an induction on k. For $k = 1$, $\eta_1 = 1$, we have $h_{t,1} = f_t(\boldsymbol{x}_{t,2}) - f_t(\boldsymbol{x}^*) \leq GD$ since f_t is G-Lipschitz and the constraint set $\mathcal{K}$ is bounded by D (Assumptions 1.2.1 and 1.2.2). More over, we have $2\beta AD^2 + \langle \nabla f_{t,1}, \boldsymbol{x}_{t,1} - \boldsymbol{x}^* \rangle \geq 2\beta AD^2 - \langle \nabla f_{t,1}, \boldsymbol{x}_{t,1} - \boldsymbol{x}^* \rangle \geq 2\beta AD^2 - GD \geq GD$ by assuming $A \geq \frac{G}{\beta D}$. We have then $h_{t,1} \leq 2\beta AD^2 + \langle \nabla f_{t,1}, \boldsymbol{x}_{t,1} - \boldsymbol{x}^* \rangle$ Assume that the inequality holds for $k-1$, we now prove for k. By definition of $h_{t,k}$, we have

$$\begin{aligned}
h_{t,k} &\leq (1-\eta_k) h_{t,k-1} + \eta_k \langle \nabla f_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^* \rangle + \eta_k^2 \frac{\beta D^2}{2} \\
&\leq (1-\eta_k) \left[\frac{2\beta AD^2}{k-1} + \sum_{k'=1}^{k-1} \eta_{k'} \left(\prod_{\ell=k'+1}^{k-1} (1-\eta_\ell) \right) \langle \nabla f_{t,k'}, \boldsymbol{v}_{t,k'} - \boldsymbol{x}^* \rangle \right] \\
&\quad + \eta_k \langle \nabla f_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^* \rangle + \eta_k^2 \frac{\beta D^2}{2} \\
&\leq (1-\eta_k) \left[\frac{2\beta AD^2}{k-1} + \sum_{k'=1}^{k-1} \eta_{k'} \left[\prod_{\ell=k'+1}^{k-1} (1-\eta_\ell) \right] \langle \nabla f_{t,k'}, \boldsymbol{v}_{t,k'} - \boldsymbol{x}^* \rangle \right] \\
&\quad + \eta_k \prod_{\ell=k+1}^{k} (1-\eta_\ell) \langle \nabla f_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^* \rangle + \eta_k^2 \frac{\beta D^2}{2} \\
&\leq (1-\eta_k) \frac{2\beta AD^2}{k-1} + \sum_{k'=1}^{k-1} \eta_{k'} \left[\prod_{\ell=k'+1}^{k} (1-\eta_\ell) \right] \langle \nabla f_{t,k'}, \boldsymbol{v}_{t,k'} - \boldsymbol{x}^* \rangle \\
&\quad + \eta_k \prod_{\ell=k+1}^{k} (1-\eta_\ell) \langle \nabla f_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^* \rangle + \eta_k^2 \frac{\beta D^2}{2} \\
&\leq (1-\eta_k) \frac{2\beta AD^2}{k-1} + \sum_{k'=1}^{k} \eta_{k'} \left[\prod_{\ell=k'+1}^{k} (1-\eta_\ell) \right] \langle \nabla f_{t,k'}, \boldsymbol{v}_{t,k'} - \boldsymbol{x}^* \rangle + \eta_k^2 \frac{\beta D^2}{2} \\
&\leq \frac{2\beta AD^2}{k-1} - \frac{2\beta A^2 D^2}{k\,(k-1)} + \frac{\beta A^2 D^2}{2k^2} + \sum_{k'=1}^{k} \eta_{k'} \left[\prod_{\ell=k'+1}^{k} (1-\eta_\ell) \right] \langle \nabla f_{t,k'}, \boldsymbol{v}_{t,k'} - \boldsymbol{x}^* \rangle \\
&\leq \frac{2\beta AD^2}{k-1} - \frac{2\beta A^2 D^2}{k(k-1)} + \frac{\beta A^2 D^2}{2k(k-1)} + \sum_{k'=1}^{k} \eta_{k'} \left[\prod_{\ell=k'+1}^{k} (1-\eta_\ell) \right] \langle \nabla f_{t,k'}, \boldsymbol{v}_{t,k'} - \boldsymbol{x}^* \rangle \\
&\leq \frac{2\beta AD^2}{k-1} - \frac{2\beta A^2 D^2 - \frac{\beta}{2} A^2 D^2}{k(k-1)} + \sum_{k'=1}^{k} \eta_{k'} \left[\prod_{\ell=k'+1}^{k} (1-\eta_\ell) \right] \langle \nabla f_{t,k'}, \boldsymbol{v}_{t,k'} - \boldsymbol{x}^* \rangle \\
&\leq \frac{2\beta AD^2}{k-1} - \frac{\beta A^2 D^2}{k(k-1)} + \sum_{k'=1}^{k} \eta_{k'} \left[\prod_{\ell=k'+1}^{k} (1-\eta_\ell) \right] \langle \nabla f_{t,k'}, \boldsymbol{v}_{t,k'} - \boldsymbol{x}^* \rangle \\
&\leq \frac{2\beta AD^2}{k} + \sum_{k'=1}^{k} \eta_{k'} \left[\prod_{\ell=k'+1}^{k} (1-\eta_\ell) \right] \langle \nabla f_{t,k'}, \boldsymbol{v}_{t,k'} - \boldsymbol{x}^* \rangle
\end{aligned}$$

where the last inequality follows from the fact that $\beta A^2 \geq 2\beta A$ for $A \geq 3$ and $\frac{1}{k-1} - \frac{1}{k(k-1)} \leq \frac{1}{k}$. □

Proposition 2.7.1. *In the analysis, we make use of the following bounds*

$$\left\| \overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}_{t,k}^i \right\| \leq \frac{2C_d}{k}$$

$$\left\|\boldsymbol{x}_{t,k+1}^i - \boldsymbol{x}_{t,k}^i\right\| \leq \frac{4C_d + AD}{k}$$

Proof of claim. For the first bound, recall the definition of FW-update in Algorithm 6 and using Lemma 2.7.2, we have

$$\begin{aligned}
\left\|\overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}_{t,k}^i\right\| &= \left\|(1-\eta_{k-1})\left(\overline{\boldsymbol{x}}_{t,k-1} - \boldsymbol{y}_{t,k-1}^i\right) + \eta_{k-1}\left(\overline{\boldsymbol{v}}_{t,k-1} - \boldsymbol{v}_{t,k-1}^i\right)\right\| \\
&\leq \frac{C_d}{k-1} - \frac{AC_d}{(k-1)^2} + \frac{AD}{k-1} \leq \frac{C_d}{k-1} - \left[\frac{AC_d}{(k-1)^2} - \frac{AD}{k-1}\right] \\
&\leq \frac{C_d}{k-1} - \left[\frac{AC_d - AD}{(k-1)^2}\right] \leq \frac{C_d}{k-1} \leq \frac{2C_d}{k}
\end{aligned}$$

Applying the first bound on the second one yields

$$\begin{aligned}
\left\|\boldsymbol{x}_{t,k+1}^i - \boldsymbol{x}_{t,k}^i\right\| &\leq \left\|\boldsymbol{x}_{t,k+1}^i - \overline{\boldsymbol{x}}_{t,k+1}\right\| + \left\|\overline{\boldsymbol{x}}_{t,k+1} - \overline{\boldsymbol{x}}_{t,k}\right\| + \left\|\overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}_{t,k}^i\right\| \\
&\leq \frac{2C_d}{k+1} + \frac{AD}{k} + \frac{2C_d}{k} \\
&\leq \frac{4C_d + AD}{k}
\end{aligned}$$

□

Lemma 2.7.2 (Lemma 2.4.1)**.** *Define $C_d = k_0\sqrt{n}D$, for all $t \in [T]$, $k \in [K]$, we have*

$$\max_{i\in[1,n]} \left\|\boldsymbol{y}_{t,k}^i - \overline{\boldsymbol{x}}_{t,k}\right\| \leq \frac{C_d}{k} \tag{2.27}$$

Proof. We prove the lemma by induction, we first note that

$$\begin{aligned}
\left\|\boldsymbol{y}_{t,k}^{cat} - \overline{\boldsymbol{x}}_{t,k}^{cat}\right\| &= \left\|(\mathbf{W}\otimes I_d)\,\boldsymbol{x}_{t,k}^{cat} - \left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\boldsymbol{x}_{t,k}^{cat}\right\| \\
&= \left\|\left[\left(\mathbf{W} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\boldsymbol{x}_{t,k}^{cat}\right\| \\
&= \left\|\left[\left(\mathbf{W} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\left(\boldsymbol{x}_{t,k}^{cat} - \overline{\boldsymbol{x}}_{t,k}^{cat}\right)\right\| \\
&\leq \lambda(\mathbf{W})\left\|\boldsymbol{x}_{t,k}^{cat} - \overline{\boldsymbol{x}}_{t,k}^{cat}\right\|
\end{aligned} \tag{2.28}$$

Let $C_d = k_0\sqrt{n}D$, the base case is verified for $k \in [1, k_0]$ since $\left\|\boldsymbol{x}_{t,k}^{cat} - \overline{\boldsymbol{x}}_{t,k}^{cat}\right\| \leq \sqrt{n}D \leq \frac{C_d}{k}$. Suppose that the hypothesis is verified for $k-1 \geq k_0$, we have

$$\begin{aligned}
\left\|\boldsymbol{y}_{t,k}^{cat} - \overline{\boldsymbol{x}}_{t,k}^{cat}\right\| &\leq \lambda(\mathbf{W})\left\|\boldsymbol{x}_{t,k}^{cat} - \overline{\boldsymbol{x}}_{t,k}^{cat}\right\| \\
&= \lambda(\mathbf{W})\left\|(1-\eta_{k-1})\left(\boldsymbol{y}_{t,k-1}^{cat} - \overline{\boldsymbol{x}}_{t,k-1}^{cat}\right) + \eta_{k-1}\left(\boldsymbol{v}_{t,k-1}^{cat} - \overline{\boldsymbol{v}}_{t,k-1}^{cat}\right)\right\| \\
&\leq \lambda(\mathbf{W})\left\|\boldsymbol{y}_{t,k-1}^{cat} - \overline{\boldsymbol{x}}_{t,k-1}^{cat}\right\| + \lambda(\mathbf{W})\frac{\sqrt{n}D}{k-1} \\
&\leq \lambda(\mathbf{W})\left(\frac{C_d + \sqrt{n}D}{k-1}\right) \\
&\leq \lambda(\mathbf{W})C_d\frac{k_0+1}{k_0(k-1)} \\
&\leq \frac{C_d}{k}
\end{aligned} \tag{2.29}$$

where we use the induction hypothesis in the third inequality and the last inequality follows the fact that $\lambda(\mathbf{W})\frac{k_0+1}{k_0(k-1)} \leq \frac{k-1}{k}\cdot\frac{1}{k-1} \leq \frac{1}{k}$. We conclude the proof by noting that

$$\max_{i\in[1,n]} \left\|\boldsymbol{y}_{t,k}^i - \overline{\boldsymbol{x}}_{t,k}\right\| \leq \sqrt{\sum_{i=1}^n \left\|\boldsymbol{y}_{t,k}^i - \overline{\boldsymbol{x}}_{t,k}\right\|^2} = \left\|\boldsymbol{y}_{t,k}^{cat} - \overline{\boldsymbol{x}}_{t,k}^{cat}\right\| \leq \frac{C_d}{k}$$

□

Lemma 2.7.3 (Lemma 2.4.2)**.** *Define* $C_g = \sqrt{n}\max\left\{\lambda(\mathbf{W})\left(G + \frac{\beta D}{1-\lambda(\mathbf{W})}\right), k_0\beta\left(4C_d + AD\right)\right\}$ *and recall the definition of* $\nabla F_{t,k} := \frac{1}{n}\sum_{i=1}^{n}\nabla f_{t,k}^i$*. For all* $t \in [T]$*,* $k \in [K]$*, we have*

$$\max_{i\in[1,n]}\left\|\boldsymbol{d}_{t,k}^i - \nabla F_{t,k}\right\| \leq \frac{C_g}{k} \tag{2.30}$$

Proof. We prove the lemma by induction. Following the idea from [Xie2019], we have

$$\begin{aligned}
\left\|\hat{\boldsymbol{d}}_{t,k}^{cat} - \nabla F_{t,k}^{cat}\right\| &= \left\|\left(\mathbf{W}\otimes I_d\right)\hat{\boldsymbol{g}}_{t,k}^{cat} - \left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\hat{\boldsymbol{g}}_{t,k}^{cat}\right\| \\
&= \left\|\left[\left(\mathbf{W} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\hat{\boldsymbol{g}}_{t,k}^{cat}\right\| \\
&= \left\|\left[\left(\mathbf{W} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\left(\hat{\boldsymbol{g}}_{t,k}^{cat} - \nabla F_{t,k}^{cat}\right)\right\| \\
&\leq \lambda(\mathbf{W})\left\|\hat{\boldsymbol{g}}_{t,k}^{cat} - \nabla F_{t,k}^{cat}\right\|
\end{aligned} \tag{2.31}$$

where the third equality and the last inequality are verified since

$$\mathbf{W}\cdot\nabla F_{t,k}^{cat} = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\cdot\nabla F_{t,k}^{cat} = \nabla F_{t,k}^{cat} \quad \text{and} \quad \left\|\mathbf{W} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\| \leq \lambda(\mathbf{W})$$

by [Koloskova2019, Lemma 16, (see appendix A)]. Using the gradient tracking step and Equation (2.31), we have

$$\begin{aligned}
\left\|\hat{\boldsymbol{d}}_{t,k}^{cat} - \nabla F_{t,k}^{cat}\right\| &\leq \lambda(\mathbf{W})\left\|\hat{\boldsymbol{g}}_{t,k}^{cat} - \nabla F_{t,k}^{cat}\right\| \\
&= \lambda(\mathbf{W})\left\|\boldsymbol{\delta}_{t,k}^{cat} + \hat{\boldsymbol{d}}_{t,k-1}^{cat} - \nabla F_{t,k}^{cat} + \nabla F_{t,k-1}^{cat} - \nabla F_{t,k-1}^{cat}\right\| \\
&\leq \lambda(\mathbf{W})\left(\left\|\hat{\boldsymbol{d}}_{t,k-1}^{cat} - \nabla F_{t,k-1}^{cat}\right\| + \left\|\boldsymbol{\delta}_{t,k}^{cat} - \overline{\boldsymbol{\delta}}_{t,k}^{cat}\right\|\right) \\
&\leq \lambda(\mathbf{W})\left(\left\|\hat{\boldsymbol{d}}_{t,k-1}^{cat} - \nabla F_{t,k-1}^{cat}\right\| + \left\|\boldsymbol{\delta}_{t,k}^{cat}\right\|\right)
\end{aligned} \tag{2.32}$$

where the last inequality holds since

$$\left\|\boldsymbol{\delta}_{t,k}^{cat} - \overline{\boldsymbol{\delta}}_{t,k}^{cat}\right\|^2 = \sum_{i=1}^{n}\left\|\boldsymbol{\delta}_{t,k}^i - \overline{\boldsymbol{\delta}}_{t,k}\right\|^2 = \sum_{i=1}^{n}\left\|\boldsymbol{\delta}_{t,k}^i\right\|^2 - n\left\|\overline{\boldsymbol{\delta}}_{t,k}\right\|^2 \leq \sum_{i=1}^{n}\left\|\boldsymbol{\delta}_{t,k}^i\right\|^2 = \left\|\boldsymbol{\delta}_{t,k}^{cat}\right\|^2 \tag{2.33}$$

Moreover, using the smoothness of f_t and Proposition 2.7.1, we have

$$\begin{aligned}
\left\|\boldsymbol{\delta}_{t,k}^{cat}\right\|^2 = \sum_{i=1}^{n}\left\|\boldsymbol{\delta}_{t,k}^i\right\|^2 &\leq \sum_{i=1}^{n}\left\|\nabla f_{t,k}^i - \nabla f_{t,k-1}^i\right\|^2 \leq \sum_{i=1}^{n}\beta^2\left\|\boldsymbol{x}_{t,k}^i - \boldsymbol{x}_{t,k-1}^i\right\|^2 \\
&\leq n\beta^2\left(\frac{4C_d + AD}{k-1}\right)^2
\end{aligned} \tag{2.34}$$

Thus, we have $\left\|\boldsymbol{\delta}_{t,k}^{cat}\right\| \leq \sqrt{n}\beta\frac{4C_d+AD}{k-1}$. For the base case $k = 1$, we have

$$\begin{aligned}
\left\|\hat{\boldsymbol{d}}_{t,1}^{cat} - \nabla F_{t,1}^{cat}\right\|^2 &= \left\|\left[\left(\mathbf{W} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\hat{\boldsymbol{g}}_{t,1}^{cat}\right\|^2 \\
&\leq \lambda^2(\mathbf{W})\left\|\hat{\boldsymbol{g}}_{t,1}^{cat}\right\|^2 \leq \lambda^2(\mathbf{W})\sum_{i=1}^{n}\left\|\nabla f_{t,1}^i\right\|^2 \leq n\lambda^2(\mathbf{W})G^2
\end{aligned} \tag{2.35}$$

We have then $\left\|\hat{\boldsymbol{d}}_{t,1}^{cat} - \nabla F_{t,1}^{cat}\right\| \leq \sqrt{n}\lambda(\mathbf{W})G$. For $k \in (1, k_0]$, by Equation (2.32)

$$\begin{aligned}
\left\|\hat{\boldsymbol{d}}_{t,k}^{cat} - \nabla F_{t,k}^{cat}\right\| &\leq \lambda(\mathbf{W})\left(\left\|\hat{\boldsymbol{d}}_{t,k-1}^{cat} - \nabla F_{t,k-1}^{cat}\right\| + \left\|\boldsymbol{\delta}_{t,k}^{cat}\right\|\right) \\
&\leq \lambda(\mathbf{W})\left(\left\|\hat{\boldsymbol{d}}_{t,k-1}^{cat} - \nabla F_{t,k-1}^{cat}\right\| + \sqrt{n\beta^2 D^2}\right) \\
&\leq \lambda^{k-1}(\mathbf{W})\left\|\hat{\boldsymbol{d}}_{t,1}^{cat} - \nabla F_{t,1}^{cat}\right\| + \sum_{\tau=1}^{k}\lambda^{\tau}(\mathbf{W})\sqrt{n}\beta D \\
&\leq \lambda^{k}(\mathbf{W})\sqrt{n}G + \frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})}\sqrt{n}\beta D \\
&\leq \lambda(\mathbf{W})\sqrt{n}\left(G + \frac{\beta D}{1-\lambda(\mathbf{W})}\right)
\end{aligned}$$

where in the second inequality, we use smoothness of f_t and bound the distance $\left\|\boldsymbol{x}_{t,k}^i - \boldsymbol{x}_{t,k-1}^i\right\|$ by the diameters D. The third inequality resulted from applying the previous one recursively for $k \in \{1, \ldots, k-1\}$. Using Taylor's expansion of $\lambda(\mathbf{W})$ and the bound in Equation (2.35), we obtain the fourth inequality.

Let $C_g = \sqrt{n}\max\left\{\lambda(\mathbf{W})\left(G + \frac{\beta D}{1-\lambda(\mathbf{W})}\right), k_0\beta\left(4C_d + AD\right)\right\}$. We claim that

$$\left\|\hat{\boldsymbol{d}}_{t,k-1}^{cat} - \nabla F_{t,k-1}^{cat}\right\| \leq \frac{C_g}{k-1}$$

for all $k - 1 \geq k_0$. We prove the claim for round k. Using Equation (2.32), we have

$$\begin{aligned}
\left\|\hat{\boldsymbol{d}}_{t,k}^{cat} - \nabla F_{t,k}^{cat}\right\| &\leq \lambda(\mathbf{W})\left(\left\|\hat{\boldsymbol{d}}_{t,k-1}^{cat} - \nabla F_{t,k-1}^{cat}\right\| + \left\|\boldsymbol{\delta}_{t,k}^{cat}\right\|\right) \\
&\leq \lambda(\mathbf{W})\left(\frac{C_g}{k-1} + \sqrt{n}\beta\frac{4C_d + AD}{k-1}\right) \\
&\leq \lambda(\mathbf{W})\left(\frac{C_g + \sqrt{n}\beta\left(4C_d + AD\right)}{k-1}\right) \\
&\leq \lambda(\mathbf{W})\left(C_g\frac{k_0+1}{k_0(k-1)}\right) \\
&\leq \frac{C_g}{k} \qquad (2.36)
\end{aligned}$$

where the second inequality followed by the induction hypothesis and Equation (2.34). The fourth inequality is a consequence of the definition of C_g and the final inequality resulted from the fact that $\lambda(\mathbf{W})\frac{k_0+1}{k_0(k-1)} \leq \frac{1}{k}$ as $k > k_0$. We conclude the proof by noting that

$$\max_{i\in[1,n]}\left\|\hat{\boldsymbol{d}}_{t,k}^i - \nabla F_{t,k}\right\| \leq \sqrt{\sum_{i=1}^{n}\left\|\hat{\boldsymbol{d}}_{t,k}^i - \nabla F_{t,k}\right\|^2} = \left\|\hat{\boldsymbol{d}}_{t,k}^{cat} - \nabla F_{t,k}^{cat}\right\| \leq \frac{C_g}{k}$$

□

Lemma 2.7.4 (Lemma 2.4.3)**.** *For all $t \in [T]$, $k \in [K]$ and $i \in [n]$. Let $\boldsymbol{v}_{t,k}^i$ be the output of the oracle $\mathcal{O}_k^i$ with delayed feedback and $\hat{\boldsymbol{v}}_{t,k}^i$ its homologous in non-delay case. Suppose that Assumptions 1.2.1 and 1.2.2 hold true. Choosing FTPL as the oracle, we have:*

$$\left\|\boldsymbol{v}_{t,k}^i - \hat{\boldsymbol{v}}_{t,k}^i\right\| \leq 2\zeta\sqrt{n}DG\left[\frac{\lambda\left(\mathbf{W}\right)}{1-\lambda(\mathbf{W})} + 1\right]\frac{1}{n}\sum_{i=1}^{n}\sum_{s\leq t}\mathbb{I}_{\{s+d_s^i>t\}} \qquad (2.37)$$

where ζ is the learning rate, $\lambda(\mathbf{W})$ is the second-largest eigenvalue of $\mathbf{W}$.

Proof. We call $\boldsymbol{u}_t^i = \zeta\sum_{\ell=1}^{t}\boldsymbol{d}_\ell^i$ the accumulated delayed feedback of the oracle of agent i and $\hat{\boldsymbol{u}}_t^i = \zeta\sum_{\ell=1}^{t}\hat{\boldsymbol{d}}_\ell^i$ its homologous in non-delay setting. Using the same computation in the proof of

Lemma 2.3.1, we have

$$\left\|\boldsymbol{v}_{t,k}^i - \hat{\boldsymbol{v}}_{t,k}^i\right\| \leq D\left\|\boldsymbol{u}_{t,k}^i - \hat{\boldsymbol{u}}_{t,k}^i\right\| \leq D\left\|\boldsymbol{u}_{t,k}^{cat} - \hat{\boldsymbol{u}}_{t,k}^{cat}\right\| \leq \zeta D\left\|\sum_{\ell=1}^{t}\left[\boldsymbol{d}_{\ell,k}^{cat} - \hat{\boldsymbol{d}}_{\ell,k}^{cat}\right]\right\| \tag{2.38}$$

By the definition $\nabla f_{t,k}^{cat}$ and $\nabla \hat{f}_{t,k}^{cat}$ from equation (2.17), we have

$$\sum_{\ell=1}^{t}\left[\nabla f_{\ell,k}^{cat} - \nabla \hat{f}_{\ell,k}^{cat}\right] = \sum_{s<t}\left[\nabla f_{s,k}^{1\top}\mathbb{I}_{\{s+d_s^1>t\}}, \ldots, \nabla f_{s,k}^{n\top}\mathbb{I}_{\{s+d_s^n>t\}},\right]^\top \tag{2.39}$$

and using the expansion

$$\begin{aligned}\boldsymbol{d}_{t,k}^{cat} = &\sum_{\tau=1}^{k-1}\left[\left[\left(\mathbf{W}^{k-\tau} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\left(\nabla f_{t,\tau+1}^{cat} - \nabla f_{t,\tau}^{cat}\right)\right] \\ &+\left[\left(\mathbf{W}^k - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\nabla f_{t,1}^{cat} + \left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\otimes I_d\right)\nabla f_{t,k}^{cat}\end{aligned} \tag{2.40}$$

from proposition 5, we bound the RHS of equation (2.38) as follows:

$$\begin{aligned}&\left\|\sum_{\ell=1}^{t}\left[\boldsymbol{d}_{\ell,k}^{cat} - \hat{\boldsymbol{d}}_{\ell,k}^{cat}\right]\right\| \leq \sum_{\tau=1}^{k-1}\left\|\left(\mathbf{W}^{k-\tau} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right\|\left\|\sum_{\ell=1}^{t}\left[\nabla f_{\ell,\tau+1}^{cat} - \hat{f}_{\ell,\tau+1}^{cat} + \hat{f}_{\ell,\tau}^{cat} - \nabla f_{t,\tau}^{cat}\right]\right\| \\ &\quad+\left\|\left(\mathbf{W}^k - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right\|\left\|\sum_{\ell=1}^{t}\left[\nabla f_{\ell,1}^{cat} - \hat{f}_{\ell,1}^{cat}\right]\right\| + \left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\otimes I_d\right)\left\|\sum_{\ell=1}^{t}\left[\nabla f_{\ell,k}^{cat} - \hat{f}_{\ell,k}^{cat}\right]\right\| \\ &\leq 2\sum_{\tau=1}^{k}\lambda^{k-\tau}(\mathbf{W})\left\|\sum_{s\leq t}\left[\nabla f_{s,\tau}^{1\top}\mathbb{I}_{\{s+d_s^1>t\}}, \ldots, \nabla f_{s,\tau}^{n\top}\mathbb{I}_{\{s+d_s^n>t\}}\right]^\top\right\| \\ &\quad+\lambda^k(\mathbf{W})\left\|\sum_{s\leq t}\left[\nabla f_{s,k}^{1\top}\mathbb{I}_{\{s+d_s^1>t\}}, \ldots, \nabla f_{s,k}^{n\top}\mathbb{I}_{\{s+d_s^n>t\}}\right]^\top\right\| + \left\|\sum_{s\leq t}\left[\nabla f_{s,1}^{1\top}\mathbb{I}_{\{s+d_s^1>t\}}, \ldots, \nabla f_{s,1}^{n\top}\mathbb{I}_{\{s+d_s^n>t\}}\right]^\top\right\| \\ &\leq 2\sum_{\tau=1}^{k}\lambda^{k-\tau}(\mathbf{W})\sum_{s\leq t}\left\|\left[\nabla f_{s,\tau}^{1\top}\mathbb{I}_{\{s+d_s^1>t\}}, \ldots, \nabla f_{s,\tau}^{n\top}\mathbb{I}_{\{s+d_s^n>t\}}\right]^\top\right\| \\ &\quad+\lambda^k(\mathbf{W})\sum_{s\leq t}\left\|\left[\nabla f_{s,k}^{1\top}\mathbb{I}_{\{s+d_s^1>t\}}, \ldots, \nabla f_{s,k}^{n\top}\mathbb{I}_{\{s+d_s^n>t\}}\right]^\top\right\| + \sum_{s\leq t}\left\|\left[\nabla f_{s,1}^{1\top}\mathbb{I}_{\{s+d_s^1>t\}}, \ldots, \nabla f_{s,1}^{n\top}\mathbb{I}_{\{s+d_s^n>t\}}\right]^\top\right\| \\ &\leq 2\sum_{\tau=1}^{k}\lambda^{k-\tau}(\mathbf{W})\sum_{s\leq t}\sqrt{\sum_{i=1}^{n}\left\|\nabla f_{s,\tau}^i\mathbb{I}_{\{s+d_s^i>t\}}\right\|^2} + \lambda^k(\mathbf{W})\sum_{s\leq t}\sqrt{\sum_{i=1}^{n}\left\|\nabla f_{s,k}^i\mathbb{I}_{\{s+d_s^i>t\}}\right\|^2} + \sum_{s\leq t}\sqrt{\sum_{i=1}^{n}\left\|\nabla f_{s,1}^i\mathbb{I}_{\{s+d_s^i>t\}}\right\|^2} \\ &\overset{(a)}{\leq} 2G\sum_{\tau=1}^{k}\lambda^{k-\tau}(\mathbf{W})\sum_{s\leq t}\sqrt{\sum_{i=1}^{n}\mathbb{I}_{\{s+d_s^i>t\}}} + G\left[\lambda^k(\mathbf{W})+1\right]\sum_{s\leq t}\sqrt{\sum_{i=1}^{n}\mathbb{I}_{\{s+d_s^i>t\}}} \\ &\leq 2G\left[\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})}+1\right]\sum_{s\leq t}\sqrt{\sum_{i=1}^{n}\mathbb{I}_{\{s+d_s^i>t\}}} \\ &\leq 2\sqrt{n}G\left[\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})}+1\right]\sum_{s\leq t}\frac{1}{n}\sum_{i=1}^{n}\sum_{i=1}^{n}\mathbb{I}_{\{s+d_s^i>t\}}\end{aligned}$$

where the (a) resulted from Lipschitzness of f_t. □

3

Distributed Online Algorithm for DR-Submodular Optimization

Contents

3.1 Introduction

Learning over data generated by sensors and mobile devices has gained a high interest in recent years due to the continual interaction with users and the environment on a timely basis. The patterns related to user's behavior, preference, and the surrounding stochastic events become a promising source for machine learning applications to be more and more reliable. However, collecting such data in a centralized location has become problematic due to privacy concerns and the high cost of data transfer over the network. Consequently, the learning methods that can leave the data locally while efficiently exploiting data patterns, such as decentralized learning, are emerging as an alternative to traditional centralized learning.

Under the optimization scheme, learning in a decentralized manner consists of multiple interconnected agents cooperating to optimize a global objective function where each agent detains partial information of the interested function. Several works [Deori2016, Reisizadeh2019, Yuan2016, Duchi2012, Zheng2018] have considered this setting for convex and strongly convex functions. [Wai2017] also study the problem when the objective function is generally non-convex whereas [Mokhtari2018b, Xie2019] proposes a decentralized algorithm to maximize monotone submodular functions for both continuous and discrete domains. However, these works only consider the offline setting which is not realistic since data constantly evolve in many real-world applications. In this chapter, we study decentralized online algorithms for optimizing both convex and submodular functions.

Problem definition. Formally, we are given a compact convex set $\mathcal{K} \subseteq \mathbb{R}^d$ (w.l.o.g one can assume that $\mathcal{K} \subseteq [0,1]^d$) and a set of agents connected over a network as introduced in Section 1.2. At every time $t \in [T]$, each agent $i \in V$ can communicate with (and only with) its immediate neighbors and takes a decision $\boldsymbol{x}_t^i \in \mathcal{K}$. Subsequently, a cost/reward function $f_t^i : \mathcal{K} \to \mathbb{R}$ is revealed adversarially and locally to agent i. Note that in the *bandit* setting, agent i observes only the value $f_t^i(\boldsymbol{x}_t^i)$ instead of the whole function f_t^i. Although each agent i observes only function f_t^i (or the value $f_t^i(\boldsymbol{x}_t^i)$ in the bandit setting), agent i is interested in the cumulating cost/reward $F_t(\cdot) = \frac{1}{n}\sum_{j=1}^n f_t^j(\cdot)$. In particular, at time t, the cost/reward of agent i with the its chosen $\boldsymbol{x}_t^i$ is $F_t(\boldsymbol{x}_t^i)$.

In the context of convex minimization, the functions f_t^i's are convex and the goal of each agent i is to minimize the total cumulating cost $\sum_{t=1}^T F_t(\boldsymbol{x}_t^i)$ via local communication with its immediate neighbors. Our objective is to design an algorithm with small regret. An online algorithm is $\mathcal{R}_T$-*regret* if for every agent $1 \leq i \leq n$,

$$\sum_{t=1}^{T} F_t(\boldsymbol{x}_t^i) - \min_{\boldsymbol{x}\in\mathcal{K}} \sum_{t=1}^{T} F_t(\boldsymbol{x}) \leq \mathcal{R}_T$$

In the context of monotone DR-submodular maximization, the functions f_t^i's are monotone DR-submodular. Roughly speaking, a bounded differentiable and non-negative function $F : [0,1]^d \to \mathbb{R}_+$ is *DR-submodular* if for every $\boldsymbol{x}, \boldsymbol{y} \in [0,1]^d$ satisfying $x_i \leq y_i, \forall i \in [d]$, we have $\nabla F(\boldsymbol{x}) \geq \nabla F(\boldsymbol{y})$. The goal of each agent i is to maximize the total cumulating reward $\sum_{t=1}^T F_t(\boldsymbol{x}_t^i)$, again via local communication with its immediate neighbors. Our objective is to design an algorithm with an approximation ratio as close to 1 as possible and together with a small regret. An online algorithm has a ρ-*regret* of $\mathcal{R}_T$ if for every agent $1 \leq i \leq n$,

$$\rho \cdot \max_{\boldsymbol{x}\in\mathcal{K}} \sum_{t=1}^{T} F_t(\boldsymbol{x}) - \sum_{t=1}^{T} F_t(\boldsymbol{x}_t^i) \leq \mathcal{R}_T$$

3.1.1 Our contribution

The challenge in designing robust and efficient algorithms for these problems is to simultaneously address the following issues:

- Uncertainty (online setting, agents observe their loss functions only after selecting their decisions).
- Partial information (decentralized setting, agents know only its local loss functions while attempting to minimize the cumulated cost).

- Low computation/communication resources of agents (so it is desirable that each agent performs a small number of gradient computations and communications).
- Additionally, in the bandit setting, one has only limited feedback (agents can only observe the function value of their decisions).

We present performance-guaranteed algorithms for solving the constraint convex and continuous DR-submodular optimization problem in the decentralized and online setting with *only* one gradient evaluation and low communications per agent per time step on average. Specifically, our algorithms achieve the regret and the $\left(1-\frac{1}{e}\right)$-regret bounds of $O(T^{4/5})$ for both convex and monotone continuous DR-submodular functions. Using a one-point gradient estimator [Flaxman2005], we extend the algorithms to the bandit setting in which the gradient is unavailable to the agents. We obtain the $\left(1-\frac{1}{e}\right)$-regret bound of $O(T^{8/9})$ for the bandit setting. It should be noted that the $\left(1-\frac{1}{e}\right)$-regret of $O(T^{4/5})$ and $O(T^{8/9})$ matches the regret guarantees in the centralized online settings. Besides, one can convert the algorithm to be projection-free (by selecting suitable oracles). This property allows the algorithm to be implemented in various contexts based on the computing capacity of local devices. We demonstrate the practical application of our algorithm on a Movie Recommendation problem and present a thorough analysis of different aspects of the performance guarantee, the effects of network topology, and decentralization, which are predictably explained by our theoretical results.

Algorithm	Stochastic Gradient	$(1-1/e)$-Regret	Communications	Gradient Evaluations
DMFW	Yes	$\mathcal{O}(T^{1/2})$	$2\cdot T^{3/2}$	$T^{3/2}$
Monode-FW	Yes	$\mathcal{O}(T^{4/5})$	$2\cdot$ #neighbors	1
Bandit Monode-FW	-	$\mathcal{O}(T^{8/9})$	$2\cdot$ #neighbors	-

Tableau 3.1: Comparison of previous work on adversarial *decentralized online monotone DR-submodular maximization (DMFW [Zhu2021]) and our proposed algorithms (in bold). The communications and gradient evaluations are mesured per agent per time step.*

3.1.2 Related Works

Distributed Online Optimization. [Zhang2017] introduces a distributed variant of the online conditional gradient, which is designed and analyzed in their work. Another study by [Wan2022b] proposes a distributed online conditional gradient algorithm that achieves the same regret bound as [Zhang2017] but requires only sublinear communication rounds. However, computing exact gradients may be prohibitively expensive for moderately sized data and intractable when a closed-form solution does not exist. Many works propose stochastic variants, but only for gradient descent methods, such as those presented by [Shahrampour2018] and [Li2022]. For conditional gradient algorithms, [Zhu2021] proposes a decentralized online algorithm for maximizing monotone submodular functions on a time-varying network using stochastic gradient estimates and multiple optimization oracles. This work achieves the optimal regret bound of $O(T^{1/2})$ but requires $O(T^{3/2})$ gradient evaluations and communications per function. [Thang2022] also proposes a decentralized online algorithm for convex functions using stochastic gradient estimates and multiple optimization oracles, achieving the optimal regret bound for static networks. In this work, we advance further by designing a distributed algorithm that uses stochastic gradient estimates and requires only one gradient evaluation.

Monotone DR-submodular Maximization. The maximization of monotone DR-submodular functions has been investigated in both offline and online settings. For the offline case, [Bian2017] examined the problem where the constraint set is a down-closed convex set and demonstrated that the greedy method [Calinescu2011], a variation of the Frank-Wolfe algorithm, ensures a $(1-1/e)$-approximation. [Hassani2017] demonstrated the restriction of the greedy method in a stochastic

environment where only unbiased gradient estimates are available. Later, [Mokhtari2018a] introduced an algorithm for maximizing monotone DR-submodular function over the general convex set using new variance reduction techniques to accomplish $(1-1/e)$-approximation in a stochastic setting. [Chen2018a] suggested a method that achieves $(1-1/e, O(\sqrt{T}))$-regret for maximizing monotone Dr-submodular over a general convex set in an online setting. Subsequently, [Zhang2019] introduced an approach that reduces the number of per-function gradient evaluations from $T^{3/2}$ to 1, while maintaining the same approximation ratio of $(1-1/e)$. They also presented a bandit approach that achieves an expected $(1-1/e)$-approximation ratio with regret $T^{8/9}$ to tackle the same problem.

3.2 Preliminaries and Notations

Following the notations defined in Section 1.2, we use boldface letter e.g $\boldsymbol{x}$ to represent vectors and we denote by $\boldsymbol{x}_{q,k}^i$ the decision vector of agent i at time step k of phase q. If not specified otherwise, we suppose that the constraint set $\mathcal{K}$ is a compact convex set with diameters D and radius R i.e (Assumption 1.2.1). For two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, we note $\boldsymbol{x} \leq \boldsymbol{y}$ if $x_i \leq y_i \ \forall i$. We note $\mathbb{B}_d$ and $\mathbb{S}_d$ the d-dimensional unit ball and the unit sphere, respectively.

A continuous function $F : [0,1]^d \to \mathbb{R}_+$ is *DR-submodular* if for any vectors $\boldsymbol{x}$, $\boldsymbol{y} \in [0,1]^d$ such that $\boldsymbol{x} \leq \boldsymbol{y}$, for a constant $\alpha > 0$ and any basis vectors $\boldsymbol{e}_i = (0, \ldots, 1, \ldots, 0)$ such that $\boldsymbol{x} + \alpha \boldsymbol{e}_i \in [0,1]^d$ and $\boldsymbol{y} + \alpha \boldsymbol{e}_i \in [0,1]^d$.

$$F(\boldsymbol{x} + \alpha \boldsymbol{e}_i) - F(\boldsymbol{x}) \geq F(\boldsymbol{y} + \alpha \boldsymbol{e}_i) - F(\boldsymbol{y}) \tag{3.1}$$

For a differentiable function, the DR-property is equivalent to $\nabla F(\boldsymbol{x}) \geq \nabla F(\boldsymbol{y})$, $\forall \boldsymbol{x} \leq \boldsymbol{y} \in [0,1]^d$. More over, if F is twice-differentiable, the DR-property is equivalent to all entries of the Hessian matrix being non-positive i.e $\forall 1 \leq i, j \leq d$, $\frac{\partial^2 F}{\partial \boldsymbol{x}_i \partial \boldsymbol{x}_j} \leq 0$. A function F is *monotone* if $\forall \boldsymbol{x} \leq \boldsymbol{y} \in [0,1]^d$, we have $F(\boldsymbol{x}) \leq F(\boldsymbol{y})$.

In this chapter, we employ optimization oracles in our algorithm to solve an online linear optimization problem given a feedback function and a constraint set. In particular, in the online linear optimization problem, one must choose $\boldsymbol{u}^t \in \mathcal{K}$ at every time $1 \leq t \leq T$. The adversary then discloses a vector $\boldsymbol{d}^t$ and feedbacks the cost function $\langle \cdot, \boldsymbol{d}^t \rangle$ where the goal is to minimize the regret of the linear objective. Several algorithms [Hazan2016a], including the projection-free follow-the-perturbed-leader algorithm offer an optimum regret bound of $\mathcal{R}_T = O(\sqrt{T})$ for the online linear optimization problem. One of these methods can be used as an oracle to solve the online linear optimization problem.

In practice, it may not be possible to use a full gradient due to the vast quantity of data and processing restrictions. To address this issue, our approach utilizes an unbiased stochastic gradient in place of the gradient and proposes a variance reduction technique for distributed optimization based on a rigorous analysis that may be applied to problems of independent interest. Additional to the assumptions defined in Section 1.2, we make the following assumptions for the next two sections.

Assumption 3.2.1. *The function f_t verifies Assumption 1.2.2 and its stochastic gradient $\widetilde{\nabla} f_t(\boldsymbol{x})$ is unbiased, uniformly upper-bounded and has a bounded variance, i.e., $\mathbb{E}\left[\widetilde{\nabla} f_t(\boldsymbol{x})\right] = \nabla f_t(\boldsymbol{x})$, $\left\|\widetilde{\nabla} f_t(\boldsymbol{x})\right\| \leq G_0$, and $\mathbb{E}\left[\left\|\widetilde{\nabla} f_t(\boldsymbol{x}) - \nabla f_t(\boldsymbol{x})\right\|^2\right] \leq \sigma_0^2$.*

Assumption 3.2.2. *For all $t \in [T]$ and $i \in [n]$, $\exists B \in \mathbb{R}_+$ s.t $\sup_{\boldsymbol{x} \in \mathcal{K}} |f_t^i(\boldsymbol{x})| \leq B$*

Assumption 3.2.3. *There exist a number $r \geq 0$ such that $r\mathbb{B}_d \subseteq \mathcal{K}$*

3.3 Full Information Setting

This section thoroughly describes the algorithm for both convex and DR-submodular optimization. Recall that each agent receives a function f_t^i at every time $t \in [T]$. We partition time steps into Q blocks, each of size K so that $T = QK$. For each block $q \in [Q]$, we define f_q^i as the average of the K functions within the block. Additionally, each agent $1 \leq i \leq n$ maintains K online linear optimization oracles $\mathcal{O}_{i,1}, \ldots, \mathcal{O}_{i,K}$. Let $\sigma_q \in \mathfrak{S}_K$ be a random permutation of function indexes for all agents.

At a high level, at each block q, the agent i performs K-steps of Frank-Wolfe algorithm, where the update vector is a combination of the oracles' outputs and the aggregate of its neighbors' current decisions. The final decision $\boldsymbol{x}_q^i$ for the block q is disclosed at the end of K steps, such that at each time step in the block, agent i plays the same decision $\boldsymbol{x}_q^i$.

More specifically, following the Frank-Wolfe steps, agent i performs K gradient updates using the estimators $f^i_{\sigma_q(k)}$. It calculates the stochastic gradient of the permuted function $f^i_{\sigma_q(k)}$ evaluated at the corresponding decision vector $\boldsymbol{x}^i_{q,k}$ and thereafter exchanges information with its neighbors. It then computes a variance reduction version $\widetilde{\boldsymbol{a}}^i_{q,k}$ of the vector $\widetilde{\boldsymbol{d}}^i_{q,k}$ and returns $\langle \widetilde{\boldsymbol{a}}^i_{q,k}, \cdot \rangle$ as the cost function at time $\sigma_q^{-1}(k)$ to the oracle $\mathcal{O}^i_k$. The vectors $\widetilde{\boldsymbol{d}}^i_{q,k}$ are subtly constructed to capture progressively more information on the accumulating cost functions.

Note that the use of random permutation σ_q is crucial here. By that, all the permuted functions $f^i_{\sigma_q(k)}$ become an estimation of f^i_q, i.e., $\mathbb{E}[f^i_{\sigma_q(k)}] = f^i_q$. Therefore the gradient of $f^i_{\sigma_q(k)}$ is likewise an estimation of the gradient of f^i_q. One can think of f^i_q as an artificial objective function for which we have access to its gradient estimates, where each estimation is one gradient evaluation per function within the block. As a result, conducting K gradient updates of f^i_q turns out to be executing one gradient update for each of the K functions. Using this approach, initiated in [Zhang2019], we can effectively reduce the gradient evaluation number to 1 for each arriving function f^i_t.

Since we deal with both convex and submodular, there are modifications to adapt for both kinds of optimization problem. The online optimization oracle's objective function should be minimized for convex optimization and maximized for submodular optimization. The decision update for convex problems is a convex combination of the aggregated neighbors' decisions $\boldsymbol{y}^i_{q,k}$ and the oracle's output $\boldsymbol{v}^i_{q,k}$, i.e.,

$$\boldsymbol{x}^i_{q,k+1} = (1-\eta_k)\boldsymbol{y}^i_{q,k} + \eta_k \boldsymbol{v}^i_{q,k}, \quad \eta_k \in [0,1] \tag{3.2}$$

whereas the update for the submodular optimization problem is achieved by shifting the aggregated decisions towards the direction of the oracle's output by a step-size η_k, i.e.,

$$\boldsymbol{x}^i_{q,k+1} = \boldsymbol{y}^i_{q,k} + \eta_k \boldsymbol{v}^i_{q,k}, \quad \eta_k \in [0,1] \tag{3.3}$$

For convex functions, the initialization can be any random point inside the constraint set, however for submodular functions, this value should be set to 0. We give a formal description in Algorithm 9.

Theorem 3.3.1 (Convex Case)**.** *Given a convex set $\mathcal{K}$ and assume that F_t is convex. Setting $Q = T^{2/5}, K = T^{3/5}, T = QK$ and step-size $\eta_k = \frac{1}{k}$. Let $\rho_k = \frac{2}{(k+3)^{2/3}}$ and $\rho_k = \frac{1.5}{(K-k+2)^{2/3}}$ when $k \in \left[1, \frac{K}{2}\right]$ and $k \in \left[\frac{K}{2}+1, K\right]$ respectively. Then, the expected regret of Algorithm 9 is at most*

$$\mathbb{E}\left[\mathcal{R}_T\right] \leq \left(GD + 2\beta D^2\right) T^{2/5} + \left(C + 6D\left(N + \sqrt{M}\right)\right) T^{4/5} + \frac{3}{5}\beta D^2 T^{2/5} \log(T) \tag{3.4}$$

where $N = k_0 \cdot nG \max\{\lambda_2\left(1 + \frac{2}{1-\lambda_2}\right), 2\}$ and $M = \max\{M_1, M_2\}$ where $M_0 = 4\left(V_{\boldsymbol{d}}^2 + \sigma_1^2\right) + 128V_{\boldsymbol{d}}^2$, $M_1 = \max\left\{5^{2/3}\left(V_{\boldsymbol{d}} + \frac{2}{4^{2/3}}G_0\right)^2, M_0\right\}$ and $M_2 = 2.55\left(V_{\boldsymbol{d}}^2 + \sigma_1^2\right) + \frac{28V_{\boldsymbol{d}}^2}{3}$

Theorem 3.3.2 (Submodular Case)**.** *Given a convex set $\mathcal{K}$ and assume that the function F_t is monotone continuous DR-submodular. Setting $Q = T^{2/5}, K = T^{3/5}, T = QK$ and step-size $\eta_k = \frac{1}{K}$. Let $\rho_k = \frac{2}{(k+3)^{2/3}}$ and $\rho_k = \frac{1.5}{(K-k+2)^{2/3}}$ when $k \in \left[1, \frac{K}{2}+1\right]$ and $k \in \left[\frac{K}{2}+2, K\right]$ respectively. Then, the expected $\left(1 - \frac{1}{e}\right)$-regret is at most*

$$\mathbb{E}\left[\mathcal{R}_T\right] \leq \frac{3}{2}\beta D^2 T^{2/5} + \left(C + 3D(N + \sqrt{M}\right) T^{4/5} \tag{3.5}$$

where the constants are defined in Theorem 3.3.1

Algorithm 7 Monode Frank-Wolfe

Input: A convex set $\mathcal{K}$, a time horizon T, a block size K, online linear optimization oracles $\mathcal{O}_{i,1}, \dots, \mathcal{O}_{i,K}$ for each agent $1 \le i \le n$, step sizes $\eta_k \in (0,1)$ for all $1 \le k \le K$, number of blocks $Q = T/K$

1: Initialize linear optimizing oracle $\mathcal{O}_k^i$ for all $1 \le k \le K$
2: **for** $q = 1$ to Q **do**
3: **for** every agent $1 \le i \le n$ **do**
4: Initialize $\boldsymbol{x}_{q,1}^i$ and set $\widetilde{\boldsymbol{a}}_{i,0}^t \leftarrow 0$
5: **for** $1 \le k \le K$ **do**
6: Let $\boldsymbol{v}_{q,k}^i$ be the output of oracle $\mathcal{O}_k^i$ at phase q.
7: Send $\boldsymbol{x}_{q,k}^i$ to all neighbours $N(i)$
8: Once receiving $\boldsymbol{x}_{q,k}^j$ from all neighbours $j \in N(i)$, set $\boldsymbol{y}_{q,k}^i \leftarrow \sum_j W_{ij} \boldsymbol{x}_{q,k}^j$.
9: Update $\boldsymbol{x}_{q,k+1}^i$ as (3.2) or (3.3).
10: **end for**
11: Choose $\boldsymbol{x}_q^i \leftarrow \boldsymbol{x}_{q,K+1}^i$ and agent i plays the same $\boldsymbol{x}_q^i$ for every time t in phase q.
12: Let σ_q be a random permutation of $1, \dots, K$ — times in phase q.
13: **for** $1 \le k \le K$ **do**
14: Let $s = \sigma_q^{-1}(k)$
15: Query the values of $\widetilde{\nabla} f_k^i(\boldsymbol{x}_{q,s}^i)$
16: **end for**
17: Set $\widetilde{\boldsymbol{g}}_{q,1}^i \leftarrow \widetilde{\nabla} f_{\sigma_q(1)}^i(\boldsymbol{x}_{q,1}^i)$
18: **for** $1 \le k \le K$ **do**
19: Send $\widetilde{\boldsymbol{g}}_{q,k}^i$ to all neighbours $N(i)$.
20: After receiving $\widetilde{\boldsymbol{g}}_{q,k}^j$ from all neighbours $j \in \mathcal{N}(i)$, compute $\widetilde{\boldsymbol{d}}_{q,k}^i \leftarrow \sum_{j \in N(i)} W_{ij} \widetilde{\boldsymbol{g}}_{q,k}^j$ and $\widetilde{\boldsymbol{g}}_{q,k+1}^i \leftarrow \big(\widetilde{\nabla} f_{\sigma_q(k+1)}^i(\boldsymbol{x}_{q,k+1}^i) - \widetilde{\nabla} f_{\sigma_q(k)}^i(\boldsymbol{x}_{q,k}^i)\big) + \widetilde{\boldsymbol{d}}_{q,k}^i$
21: $\widetilde{\boldsymbol{a}}_{q,k}^i \leftarrow (1-\rho_k) \cdot \widetilde{\boldsymbol{a}}_{q,k-1}^i + \rho_k \cdot \widetilde{\boldsymbol{d}}_{q,k}^i$.
22: Feedback function $\langle \widetilde{\boldsymbol{a}}_{q,k}^i, \cdot \rangle$ to oracles $\mathcal{O}_k^i$. (The cost of the oracle $\mathcal{O}_k^i$ at block q is $\langle \widetilde{\boldsymbol{a}}_{q,k}^i, \boldsymbol{v}_{q,k}^i \rangle$.)
23: **end for**
24: **end for**
25: **end for**

As stated in the preceding paragraph, the distinction between convex and submodular optimization can be found in line 9 of Algorithm 9 and in the oracle optimization subroutine. To achieve the regret bound mentioned in Theorems 3.3.1 and 3.3.2, we use follow-the-perturbed-leader as the oracle with regret $\mathcal{R}_T = C\sqrt{T}$. In the case of convex optimization, one may use online gradient descent to obtain the same outcome, but this method is more computationally intensive because it involves a projection step onto the constraint set.

3.3.1 Technical Analysis

For the ease of analysis, we note $\sigma_q(k)$ to be the permutation of k at phase q. We define the average function of the remaining $(K-k)$ functions as

$$\bar{F}_{q,k}(\boldsymbol{x}) = \frac{1}{K-k} \sum_{\ell=k+1}^{K} F_{\sigma_q(\ell)}(\boldsymbol{x}) = \frac{1}{K-k} \sum_{\ell=k+1}^{K} \frac{1}{n} \sum_{i=1}^{n} f_{\sigma_q(\ell)}^i(\boldsymbol{x}) \tag{3.6}$$

where $F_{\sigma_q(\ell)}(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^n f_{\sigma_q(\ell)}^i(\boldsymbol{x})$. We also define

$$\hat{f}_{q,k}^i = \frac{1}{K-k} \sum_{\ell=k+1}^{K} f_{\sigma_q(\ell)}^i(\boldsymbol{x}_{q,\ell}^i), \quad \nabla \hat{f}_{q,k}^i = \frac{1}{K-k} \sum_{\ell=k+1}^{K} \nabla f_{\sigma_q(\ell)}^i(\boldsymbol{x}_{q,\ell}^i) \tag{3.7}$$

as the average of the remaining $(K-k)$ functions and stochastic gradients of $f^i_{\sigma_q(\ell)}(\boldsymbol{x}^i_{q,\ell})$ respectively. Then we note,

$$\hat{F}_{q,k} = \frac{1}{n}\sum_{i=1}^{n} \hat{f}^i_{q,k}, \quad \nabla\hat{F}_{q,k} = \frac{1}{n}\sum_{i=1}^{n} \nabla\hat{f}^i_{q,k}, \tag{3.8}$$

In the same spririt of $\hat{f}^i_{q,k}$, we define

$$\hat{\boldsymbol{g}}^i_{q,k} = \frac{1}{K-k}\sum_{\ell=k+1}^{K} \boldsymbol{g}^i_{q,\ell}, \quad \hat{\boldsymbol{d}}^i_{q,k} = \frac{1}{K-k}\sum_{\ell=k+1}^{K} \boldsymbol{d}^i_{q,\ell} \tag{3.9}$$

We let $\mathcal{F}_{q,1} \subset \cdots \subset \mathcal{F}_{q,k}$ to be the σ-field generated by the permutation up to time k and $\mathcal{H}_{q,1} \subset \cdots \subset \mathcal{H}_{q,k}$ another σ-field generated by the randomness of the stochastic gradient estimate up to time k.

Assumption 3.3.1. *Let $\left\{\widetilde{\boldsymbol{d}}_t\right\}_1^T$ be a sequence such that $\mathbb{E}\left[\widetilde{\boldsymbol{d}}_t \mid \mathcal{H}_{t-1}\right] = \boldsymbol{d}_t$ where $\mathcal{H}_{t-1}$ is the filtration of the stochastic estimate up to $t-1$.*

The proof of Theorem 3.3.1 and Theorem 3.3.2 are proceeded as follows :

- We begin by deriving upper bounds on the expected distance between the local gradient average $\hat{\boldsymbol{d}}^i_{q,k}$ and the remaining global average $\nabla\hat{F}q,k$, as detailed in Lemma 3.3.1. We also establish a bound between $\hat{\boldsymbol{d}}^i q, k-1$ and its variance reduced estimates $\widetilde{\boldsymbol{a}}^i_{q,k}$ in Lemma 3.3.4.
- The proof of Lemma 3.3.4 involves two additional lemmas, 3.3.3 and 3.3.2, which provide bounds on the variance of the stochastic local gradient $\widetilde{\boldsymbol{d}}^i_{q,k}$ and the norm of the local gradient $\boldsymbol{d}^i_{q,k}$, respectively.
- These two lemmas are instrumental in proving Proposition 3.3.2, where we bound the sum of expected distances between the full block average gradient $\nabla\overline{F}q,k-1$ and the variance reduced estimates $\widetilde{\boldsymbol{a}}^i q,k$ over K sub-iterations.
- The results from Proposition 3.3.2 are then utilized to derive the regret bounds in Theorems 3.3.1 and 3.3.2.
- For the proof of Theorem 3.3.2, we also employ Lemma 3.3.5 to derive an upper bound on the primal gap for DR-submodular function.

The detailed proofs will be presented in the subsequent section, with some remaining proofs postponed to the end of this chapter.

Lemma 3.3.1. *Suppose that each of $f^i_{\sigma_q(k)}$ is β-smooth. Using the Frank-Wolfe update of $\boldsymbol{x}^i_{q,k}$, the average of the remaining $(K-k)$ gradient approximation $\hat{\boldsymbol{d}}^i_{q,k}$ satisfies*

$$\max_{i\in[1,n]} \mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k} - \nabla\hat{F}_{q,k}\right\|\right] \leq \begin{cases} \dfrac{N}{k} & k \in \left[1, \dfrac{K}{2}\right] \\ \dfrac{N}{K-k+1} & k \in \left[\dfrac{K}{2}+1, K\right] \end{cases}$$

where $N = nGk_0 \max\{\lambda(\mathbf{W})\left(1 + \frac{2}{1-\lambda(\mathbf{W})}\right), 2\}$.

Proof. See Lemma 3.7.1. □

Lemma 3.3.2. *Let $V_d = 2nG\left(\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})} + 1\right)$, the local gradient is uniformly upper-bounded, i.e, $\forall i \in [n], \forall k \in [K]$. $\left\|\boldsymbol{d}^i_{q,k}\right\| \leq V_d$.*

Proof. See Lemma 3.7.2. □

Lemma 3.3.3. *Under Assumption 3.2.1 and let* $\sigma_1^2 = 4n\left[\left(\frac{G+G_0}{\frac{1}{\lambda(\mathbf{W})}-1}\right)^2 + 2\sigma_0^2\right]$. *For* $i \in [n], k \in [K]$, *the variance of the local stochastic gradient is uniformly bounded i.e*

$$\mathbb{E}\left[\left\|\boldsymbol{d}_{q,k}^i - \widetilde{\boldsymbol{d}}_{q,k}^i\right\|^2\right] \leq \sigma_1^2$$

Proof. See Lemma 3.7.3. □

Lemma 3.3.4 (Lemma 6, [Zhang2019])**.** *Under Assumption 3.3.1, Lemma 3.3.2, Lemma 3.3.3 and setting* $\rho_k = \frac{2}{(k+3)^{2/3}}$ *and* $\rho_k = \frac{1.5}{(K-k+2)^{2/3}}$ *for* $k \in \left[\frac{K}{2}\right]$ *and* $k \in \left[\frac{K}{2}+1, K\right]$ *respectively, we have*

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k}^i\right\|\right] \leq \begin{cases} \dfrac{\sqrt{M}}{(k+4)^{1/3}} & k \in \left[\dfrac{K}{2}\right] \\ \dfrac{\sqrt{M}}{(K-k+1)^{1/3}} & i \in \left[\dfrac{K}{2}+1, K\right] \end{cases} \tag{3.10}$$

where $M = \max\{M_1, M_2\}$ *where* $M_1 = \max\{5^{2/3}\left(V_{\boldsymbol{d}} + L_0\right)^2, M_0\}$, $M_0 = 4\left(V_{\boldsymbol{d}}^2 + \sigma^2\right) + 32\sqrt{2}V_{\boldsymbol{d}}$ *and* $M_2 = 2.55\left(V_{\boldsymbol{d}}^2 + \sigma^2\right) + \dfrac{7\sqrt{2}V_{\boldsymbol{d}}}{3}$ *and* $L_0 = \frac{2}{4^{2/3}}\left\|\widetilde{\boldsymbol{d}}_{q,1}^i\right\|$

Proof. See Lemma 3.7.4 □

Proposition 3.3.1. *Let* $\overline{F}_{q,k-1}$ *and* $\hat{F}_{q,k-1}$ *defined as in equations* (3.6) *and* (3.8)*,respectively. Under boundedness and smoothness assumptions (1.2.1,1.2.2), we have*

$$\mathbb{E}\left[\left\|\nabla\overline{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \nabla\hat{F}_{q,k-1}\right\|\right] \leq \beta D \tag{3.11}$$

Proof. Following the definition of $\overline{F}_{q,k-1}$ and $\hat{F}_{q,k-1}$, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla\overline{F}_{q,k-1}(\overline{\boldsymbol{x}}_q^k) - \nabla\hat{F}_{q,k-1}\right\|\right] \\
&= \mathbb{E}\left[\left\|\frac{1}{K-k+1}\cdot\frac{1}{n}\sum_{\ell=k}^{K}\sum_{i=1}^{n}\left(\nabla f_{\sigma_q(\ell)}^i\left(\overline{\boldsymbol{x}}_{q,k}\right) - \nabla f_{\sigma_q(\ell)}^i\left(\boldsymbol{x}_{q,\ell}^i\right)\right)\right\|\right] \\
&\leq \mathbb{E}\left[\frac{1}{K-k+1}\cdot\frac{1}{n}\sum_{\ell=k}^{K}\sum_{i=1}^{n}\left\|\nabla f_{\sigma_q(\ell)}^i\left(\overline{\boldsymbol{x}}_{q,k}\right) - \nabla f_{\sigma_q(\ell)}^i\left(\boldsymbol{x}_{q,\ell}^i\right)\right\|\right] \\
&\leq \mathbb{E}\left[\frac{1}{K-k+1}\cdot\frac{1}{n}\sum_{\ell=k}^{K}\sum_{i=1}^{n}\beta\left\|\overline{\boldsymbol{x}}_{q,k} - \boldsymbol{x}_{q,\ell}^i\right\|\right] \qquad \text{(by } \beta\text{-smoothness)} \\
&\leq \beta D
\end{aligned} \tag{3.12}$$

□

Proposition 3.3.2. *Let* $\overline{F}_{q,k}(\overline{\boldsymbol{x}}_{q,k})$ *be defined as in equation* (3.6) *and* $\widetilde{\boldsymbol{a}}_{q,k}^i$ *the variance reduction estimates. Under boundedness and smoothness assumptions, for all* $q \in [Q]$, $i \in [n]$, *we have*

$$\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla\overline{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \widetilde{\boldsymbol{a}}_{q,k}^i\right\|\right] \leq \beta D + \left(N + \sqrt{M}\right)3K^{2/3} \tag{3.13}$$

where N *and* M *are defined in lemma 3.3.1 and lemma 3.3.4 respectively.*

Proof.

$$\begin{aligned}
\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla\overline{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \tilde{\boldsymbol{a}}_{q,k}^i\right\|\right] &\leq \sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla\overline{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \nabla\hat{F}_{q,k-1}\right\|\right] + \sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla\hat{F}_{q,k-1} - \widetilde{\boldsymbol{a}}_{q,k}^i\right\|\right] \\
&\leq \beta D + \sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla\hat{F}_{q,k-1} - \hat{\boldsymbol{d}}_{q,k-1}^i\right\|\right] + \sum_{k=1}^{K}\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k}^i\right\|\right]
\end{aligned} \tag{3.14}$$

where we have used Proposition 3.3.1 and triangle inequality in the last inequality. Using Lemma 3.3.1, we have

$$\begin{aligned}
&\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla\hat{F}_{q,k-1}-\hat{\boldsymbol{d}}^i_{q,k-1}\right\|\right]\\
&\quad=\sum_{k=1}^{K/2}\mathbb{E}\left[\left\|\nabla\hat{F}_{q,k-1}-\hat{\boldsymbol{d}}^i_{q,k-1}\right\|\right]+\sum_{k=K/2+1}^{K}\mathbb{E}\left[\left\|\nabla\hat{F}_{q,k-1}-\hat{\boldsymbol{d}}^i_{q,k-1}\right\|\right]\\
&\quad\leq\sum_{k=1}^{K/2}\frac{N}{k}+\sum_{k=K/2+1}^{K}\frac{N}{K-k+1}
\end{aligned}\tag{3.15}$$

By Lemma 3.3.4, we also have

$$\begin{aligned}
&\sum_{k=1}^{K}\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1}-\widetilde{\boldsymbol{a}}^i_{q,k}\right\|\right]\\
&\quad=\sum_{k=1}^{K/2}\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1}-\widetilde{\boldsymbol{a}}^i_{q,k}\right\|\right]+\sum_{k=K/2+1}^{K}\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1}-\widetilde{\boldsymbol{a}}^i_{q,k}\right\|\right]\\
&\quad\leq\sum_{k=1}^{K/2}\frac{\sqrt{M}}{(k+4)^{1/3}}+\sum_{k=K/2+1}^{K}\frac{\sqrt{M}}{(K-k+1)^{1/3}}
\end{aligned}\tag{3.16}$$

Combining equation (3.15) and equation (3.16), equation (3.14) is written as

$$\begin{aligned}
&\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla\overline{F}_{q,k-1}(\overline{\boldsymbol{x}}^k_q)-\tilde{\boldsymbol{a}}^i_{q,k}\right\|\right]\\
&\quad\leq\beta D+\sum_{k=1}^{K/2}\left(\frac{N}{k}+\frac{\sqrt{M}}{(k+4)^{1/3}}\right)+\sum_{k=K/2+1}^{K}\left(\frac{N}{K-k+1}+\frac{\sqrt{M}}{(K-k+1)^{1/3}}\right)\\
&\quad\leq\beta D+\left(N+\sqrt{M}\right)\sum_{k=1}^{K/2}\frac{1}{(k+4)^{1/3}}+\left(N+\sqrt{M}\right)\sum_{k=K/2+1}^{K}\frac{1}{(K-k+1)^{1/3}}\\
&\quad\leq\beta D+\left(N+\sqrt{M}\right)\sum_{k=1}^{K/2}\frac{1}{k^{1/3}}+\left(N+\sqrt{M}\right)\sum_{l=1}^{K/2}\frac{1}{l^{1/3}}\\
&\quad\leq\beta D+2\left(N+\sqrt{M}\right)\int_0^{K/2}\frac{1}{s^{1/3}}ds\\
&\quad\leq\beta D+2\left(N+\sqrt{M}\right)\frac{3}{2}\left(\frac{K}{2}\right)^{2/3}\\
&\quad\leq\beta D+\left(N+\sqrt{M}\right)3K^{2/3}
\end{aligned}\tag{3.17}$$

□

3.3.2 Proof of Theorem 3.3.1

Proof. Using smoothness of $F_{\sigma_q(k)}$ and the convexity of $F_{\sigma_q(k)}$, we have

$$\begin{aligned}
&\mathbb{E}\left[\bar{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k+1})-\bar{F}_{q,k-1}(\boldsymbol{x}^*)\right]\\
&\quad\leq(1-\eta_k)\mathbb{E}\left[\bar{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k})-\bar{F}_{q,k-1}(\boldsymbol{x}^*)\right]+\frac{\eta_k}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left\langle\widetilde{\boldsymbol{a}}^i_{q,k},\boldsymbol{v}^i_{q,k}-\boldsymbol{x}^*\right\rangle\right]\\
&\qquad+\frac{\eta_k}{n}D\sum_{i=1}^{n}\mathbb{E}\left[\left\|\nabla\bar{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k})-\widetilde{\boldsymbol{a}}^i_{q,k}\right\|\right]+\frac{\beta}{2}\eta_k^2D^2
\end{aligned}\tag{3.18}$$

As $\mathbb{E}\left[\bar{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \bar{F}_{q,k-1}(\boldsymbol{x}^*)\right] = \mathbb{E}\left[\bar{F}_{q,k-2}(\overline{\boldsymbol{x}}_{q,k}) - \bar{F}_{q,k-2}(\boldsymbol{x}^*)\right]$, we can apply equation (3.18) recursively for $k \in \{1, \dots, K\}$, thus

$$\begin{aligned}
&\mathbb{E}\left[\bar{F}_{q,0}(\overline{\boldsymbol{x}}_q) - \bar{F}_{q,0}(\boldsymbol{x}^*)\right] \\
&\quad\leq \prod_{k=1}^{K}(1-\eta_k)\mathbb{E}\left[\bar{F}_{q,0}(\overline{\boldsymbol{x}}_{q,1}) - \bar{F}_{q,0}(\boldsymbol{x}^*)\right] + \sum_{k=1}^{K}\prod_{k'=k+1}^{K}(1-\eta_{k'})\frac{\eta_k}{n}\sum_{i=1}^{n}\mathbb{E}\left[\langle \widetilde{\boldsymbol{a}}_{q,k}^i, \boldsymbol{v}_{q,k}^i - \boldsymbol{x}^*\rangle\right] \\
&\qquad + \sum_{k=1}^{K}\prod_{k'=k+1}^{K}(1-\eta_{k'})\frac{\eta_k}{n}D\sum_{i=1}^{n}\mathbb{E}\left[\left\|\nabla\bar{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \widetilde{\boldsymbol{a}}_{q,k}^i\right\|\right] + \frac{\beta}{2}D^2\sum_{k=1}^{K}\prod_{k'=k+1}^{K}(1-\eta_{k'})\eta_k^2
\end{aligned} \tag{3.19}$$

Choosing $\eta_k = \frac{1}{k}$, we have

$$\prod_{k=r}^{K}(1-\eta_k) \leq \exp\left(-\sum_{k=r}^{K}\frac{1}{k}\right) \leq \frac{r}{K}$$

We have then,

$$\begin{aligned}
&\mathbb{E}\left[\bar{F}_{q,0}(\overline{\boldsymbol{x}}_q) - \bar{F}_{q,0}(\boldsymbol{x}^*)\right] \\
&\quad\leq \frac{1}{K}\mathbb{E}\left[\bar{F}_{q,0}(\overline{\boldsymbol{x}}_{q,1}) - \bar{F}_{q,0}(\boldsymbol{x}^*)\right] + \sum_{k=1}^{K}\frac{k+1}{K}\cdot\frac{1}{k}\cdot\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\langle \widetilde{\boldsymbol{a}}_{q,k}^i, \boldsymbol{v}_{q,k}^i - \boldsymbol{x}^*\rangle\right] \\
&\qquad + \sum_{k=1}^{K}\frac{k+1}{K}\cdot\frac{1}{k}\cdot\frac{1}{n}D\sum_{i=1}^{n}\mathbb{E}\left[\left\|\nabla\bar{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \widetilde{\boldsymbol{a}}_{q,k}^i\right\|\right] + \frac{\beta}{2}D^2\sum_{k=1}^{K}\frac{k+1}{K}\cdot\frac{1}{k^2}
\end{aligned} \tag{3.20}$$

Which maybe simplified by using $\frac{k+1}{K}\cdot\frac{1}{k} \leq \frac{2}{K}$.

$$\begin{aligned}
&\mathbb{E}\left[\bar{F}_{q,0}(\overline{\boldsymbol{x}}_q) - \bar{F}_{q,0}(\boldsymbol{x}^*)\right] \\
&\quad\leq \frac{1}{K}\mathbb{E}\left[\bar{F}_{q,0}(\overline{\boldsymbol{x}}_{q,1}) - \bar{F}_{q,0}(\boldsymbol{x}^*)\right] + \frac{2}{K}\cdot\frac{1}{n}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathbb{E}\left[\langle \widetilde{\boldsymbol{a}}_{q,k}^i, \boldsymbol{v}_{q,k}^i - \boldsymbol{x}^*\rangle\right] \\
&\qquad + \frac{2}{K}\cdot\frac{1}{n}D\sum_{k=1}^{K}\sum_{i=1}^{n}\mathbb{E}\left[\left\|\nabla\bar{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \widetilde{\boldsymbol{a}}_{q,k}^i\right\|\right] + \frac{\beta D^2}{2}\frac{2}{K}\sum_{k=1}^{K}\frac{1}{k} \\
&\quad\leq \frac{GD}{K} + \frac{2}{K}\cdot\frac{1}{n}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathbb{E}\left[\langle \widetilde{\boldsymbol{a}}_{q,k}^i, \boldsymbol{v}_{q,k}^i - \boldsymbol{x}^*\rangle\right] \\
&\qquad + \frac{2}{K}\cdot D\left(\beta D + \left(N + \sqrt{M}\right)3K^{2/3}\right) + \frac{\beta D^2}{K}\log K
\end{aligned} \tag{3.21}$$

where we have used Proposition 3.3.2, G-Lipschitz property of $\bar{F}_{q,0}$ and boundedness of $\mathcal{K}$. Since $T = QK$ and assume that the oracle at round k has a regret of order $\mathcal{O}\left(\sqrt{Q}\right)$, i.e

$$\mathbb{E}\left[\sum_{q=1}^{Q}\langle \widetilde{\boldsymbol{a}}_{q,k}^i, \boldsymbol{v}_{q,k}^i - \boldsymbol{x}^*\rangle\right] \leq C\sqrt{Q}$$

then, the expected regret of the algorithm upper bounded by

$$\begin{aligned}
\mathbb{E}\left[\mathcal{R}_T\right] &= \mathbb{E}\left[\sum_{q=1}^{Q}K\left(\bar{F}_{q,0}(\overline{\boldsymbol{x}}_q) - \bar{F}_{q,0}(\boldsymbol{x}^*)\right)\right] \\
&\leq QGD + CKQ^{1/2} + 2QD\left(\beta D + \left(N + \sqrt{M}\right)3K^{2/3}\right) + Q\beta D^2\log K \\
&\leq QGD + CKQ^{1/2} + 2Q\beta D^2 + 6D\left(N + \sqrt{M}\right)QK^{2/3} + Q\beta D^2\log K \\
&\leq \left(GD + 2\beta D^2\right)Q + CKQ^{1/2} + 6D\left(N + \sqrt{M}\right)QK^{2/3} + Q\beta D^2\log K
\end{aligned} \tag{3.22}$$

Setting $Q = T^{2/5}$ and $K = T^{3/5}$, we have

$$\mathbb{E}\left[\mathcal{R}_T\right] \leq \left(GD + 2\beta D^2\right) T^{2/5} + \left(C + 6D\left(N + \sqrt{M}\right)\right) T^{4/5} + \frac{3}{5}\beta D^2 T^{2/5} \log(T) \tag{3.23}$$

□

Lemma 3.3.5. *If F_t is monotone continous DR-submodular and β-smoothness, $\boldsymbol{x}_{t,k+1} = \boldsymbol{x}_{t,k} + \frac{1}{K}\boldsymbol{v}_{t,k}$ for $k \in [K]$, then*

$$\begin{aligned} F_t(\boldsymbol{x}^*) - F_t(\boldsymbol{x}_{t,k+1}) \leq{}& (1 - 1/K)\left[F_t(\boldsymbol{x}^*) - F_t\left(\boldsymbol{x}_{t,k}\right)\right] \\ &- \frac{1}{K}\left[-\left\|\nabla F_t(\boldsymbol{x}_{t,k}) - \boldsymbol{d}_{t,k}\right\| D + \langle \boldsymbol{d}_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^*\rangle\right] + \frac{\beta D^2}{2K^2} \end{aligned} \tag{3.24}$$

Proof. See Lemma 3.7.5 □

3.3.3 Proof of Theorem 3.3.2

Proof. We apply Lemma 3.3.5 with $F_t = \bar{F}_{q,k-1}$, $\boldsymbol{x}_{t,k} = \overline{\boldsymbol{x}}_{q,k}$ and $\boldsymbol{d}_{t,k} = \frac{1}{n}\sum_{i=1}^{n} \widetilde{\boldsymbol{a}}^i_{q,k}$, we have

$$\begin{aligned} \bar{F}_{q,k-1}(\boldsymbol{x}^*) - \bar{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k+1}) \leq{}& \left(1 - \frac{1}{K}\right)\left[\bar{F}_{q,k-1}(\boldsymbol{x}^*) - \bar{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k})\right] \\ &+ \frac{1}{K}\cdot\frac{1}{n}\sum_{i=1}^{n}\left[\left\|\nabla \bar{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \widetilde{\boldsymbol{a}}^i_{q,k}\right\| D + \left\langle \widetilde{\boldsymbol{a}}^i_{q,k}, \boldsymbol{x}^* - \boldsymbol{v}^i_{q,k}\right\rangle\right] + \frac{\beta}{2}\frac{D^2}{K^2} \end{aligned} \tag{3.25}$$

As $\mathbb{E}\left[\bar{F}_{q,k-1}(\boldsymbol{x}^*) - \bar{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k})\right] = \mathbb{E}\left[\bar{F}_{q,k-2}(\boldsymbol{x}^*) - \bar{F}_{q,k-2}(\overline{\boldsymbol{x}}_{q,k})\right]$, we can apply equation (3.25) recursively for $k \in \{1, \ldots, K\}$, thus

$$\begin{aligned} \mathbb{E}\left[\bar{F}_{q,0}(\boldsymbol{x}^*) - \bar{F}_{q,0}(\overline{\boldsymbol{x}}_q)\right] \leq{}& \left(1 - \frac{1}{K}\right)^K \mathbb{E}\left[\bar{F}_{q,0}(\boldsymbol{x}^*) - \bar{F}_{q,0}(\overline{\boldsymbol{x}}_{q,1})\right] \\ &+ \frac{1}{K}\cdot\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla \bar{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \widetilde{\boldsymbol{a}}^i_{q,k}\right\| D\right] + \frac{1}{K}\cdot\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{E}\left[\left\langle \widetilde{\boldsymbol{a}}^i_{q,k}, \boldsymbol{x}^* - \boldsymbol{v}^i_{q,k}\right\rangle\right] + \frac{\beta}{2}\frac{D^2}{K} \end{aligned} \tag{3.26}$$

Note that $\left(1 - \frac{1}{K}\right)^K \leq \frac{1}{e}$ and $\bar{F}_{q,0}(\overline{\boldsymbol{x}}_{q,1}) \geq 0$, we have

$$\begin{aligned} \mathbb{E}\left[\left(1 - \frac{1}{e}\right)\bar{F}_{q,0}(\boldsymbol{x}^*) - \bar{F}_{q,0}(\overline{\boldsymbol{x}}_q)\right] \leq{}& \frac{1}{K}\cdot\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla \bar{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \widetilde{\boldsymbol{a}}^i_{q,k}\right\| D\right] \\ &+ \frac{1}{K}\cdot\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{E}\left[\left\langle \widetilde{\boldsymbol{a}}^i_{q,k}, \boldsymbol{x}^* - \boldsymbol{v}^i_{q,k}\right\rangle\right] + \frac{\beta}{2}\frac{D^2}{K} \end{aligned} \tag{3.27}$$

Let $T = QK$, using Proposition 3.3.2 and note that the oracle has a regret $\mathcal{R}_{\mathcal{Q}} \leq C\sqrt{Q}$. We have

$$\begin{aligned} \mathbb{E}\left[\mathcal{R}_T\right] &= \mathbb{E}\left[\sum_{q=1}^{Q} K\left[\left(1 - \frac{1}{e}\right)\bar{F}_{q,0}(\boldsymbol{x}^*) - \bar{F}_{q,0}(\overline{\boldsymbol{x}}_q)\right]\right] \\ &\leq \frac{D}{n}\sum_{q=1}^{Q}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla \bar{F}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \widetilde{\boldsymbol{a}}^i_{q,k}\right\|\right] + \frac{1}{n}\sum_{q=1}^{Q}\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{E}\left[\left\langle \widetilde{\boldsymbol{a}}^i_{q,k}, \boldsymbol{x}^* - \boldsymbol{v}^i_{q,k}\right\rangle\right] + \frac{\beta}{2}QD^2 \\ &\leq QD\left(\beta D + \left(N + \sqrt{M}\right)3K^{2/3}\right) + KC\sqrt{Q} + \frac{\beta Q D^2}{2} \end{aligned} \tag{3.28}$$

Setting $Q = T^{2/5}$ and $K = T^{3/5}$, the expected regret of the algorithm is upper bounded by

$$\begin{aligned} \mathbb{E}\left[\mathcal{R}_T\right] &\leq T^{2/5}\left(\beta D^2 + \left(N + \sqrt{M}\right)3T^{2/5}\right) + CT^{4/5} + \frac{\beta D^2 T^{2/5}}{2} \\ &\leq \frac{3}{2}\beta D^2 T^{2/5} + \left(C + 3D(N + \sqrt{M}\right)T^{4/5} \end{aligned} \tag{3.29}$$

□

3.4 Bandit Setting

This section describes a bandit algorithm for a decentralized submodular maximization. We let $\mathcal{K}$ be a down-closed convex set. A major difference between this algorithm and the previous one is the function's value $f_t^i(\boldsymbol{x}_t^i)$ being the only information provided to the agent. It does not know of the value incurred if it had chosen another decision in the constraint set. As a consequence, this setting makes access to the gradient impossible for the agent. To circumvent this limitation, we use the one-point gradient estimate [Flaxman2005] and adapt the biphasic bandit setting [Zhang2019] to our decentralized algorithm.

We recall that for a function f_t defined on $\mathcal{K} \subset \mathbb{R}^d$, it admits a δ-smoothed version for any $\delta > 0$, given as

$$\hat{f}_{t,\delta}(\boldsymbol{x}_t) = \mathbb{E}_{\boldsymbol{v}\sim\mathbb{B}_d}\left[f_t(\boldsymbol{x}_t + \delta\boldsymbol{v})\right]$$

where $\boldsymbol{v}$ is drawn uniformly random from the d-dimensional unit ball. The value of $\hat{f}_{t,\delta}$ at a point $\boldsymbol{x}$ is the average of f_t evaluated across the d-dimensional ball of radius δ centered at $\boldsymbol{x}$. This function inherits various functional properties from f_t, therefore becomes a suitable approximation for f_t, as shown in the following lemma.

Lemma 3.4.1 (Lemma 2 [Chen2020], Lemma 6.6 [Hazan2016a])**.** *Let f be a monotone continuous DR-submodular function. If f is β-smooth, G-Lipschitz, then so is $\hat{f}_\delta$ and we have $\left\|\hat{f}_\delta(\boldsymbol{x}) - f(\boldsymbol{x})\right\| \leq \delta G$. More over, if we choose $\boldsymbol{u}$ uniformly from the unit sphere $\mathbb{S}^{d-1}$, the following equation holds*

$$\nabla\hat{f}_{t,\delta}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{u}\sim\mathbb{S}_{d-1}}\left[\frac{d}{\delta}f_t(\boldsymbol{x}+\delta\boldsymbol{u})\boldsymbol{u}\right] \tag{3.30}$$

Lemma 3.4.1 shows that a decision that maximizes $\hat{f}_{t,\delta}$ can also maximizes f_t approximately. The δ-smooth version additionally provides a one-point gradient estimate that can be used to estimate the gradient of f_t by evaluating the function at a random point on the $(d-1)$-dimensional sphere of radius δ. It is important to note that the point $\boldsymbol{x} + \delta\boldsymbol{u}$ may be outside the set when $\boldsymbol{x}$ is near to the constraint set's boundary. For this reason, we let $\mathcal{K}' \subset \mathcal{K}$ be the δ-interior of $\mathcal{K}$ that verifies : $\forall \boldsymbol{x} \in \mathcal{K}'$, $\mathbb{B}(\boldsymbol{x}, \delta) \subset \mathcal{K}$, and solve the optimization problem on the new set $\mathcal{K}'$. By shrinking the constraint set down to $\mathcal{K}'$, we assure that the point $\boldsymbol{x} + \delta\boldsymbol{u}$ is in $\mathcal{K}$ for any point $\boldsymbol{x}$ in $\mathcal{K}$'. More over, if the distance $d(\mathcal{K}', \mathcal{K})$ between $\mathcal{K}'$ and $\mathcal{K}$ is small enough, we can approximately get the optimal regret bound on the original constraint set $\mathcal{K}$ by running the bandit algorithm on $\mathcal{K}'$. The detail on the construction of $\mathcal{K}'$ is given is Lemma 3.4.2

The biphasic setting consist of partitioning T into Q blocks of size L, with each block consisting of two phases: exploration and exploitation. Each agent i performs $K < L$ steps of exploration by updating the decision vector $\boldsymbol{x}_{q,k}^i$ using equation (3.3). During the exploration phase, rather than playing the final decision as in Algorithm 9, the agent draws uniformly a random vector $\boldsymbol{u}_{q,k}^i$ from $\mathbb{S}_{d-1}$ and plays $\boldsymbol{x}_{q,k}^i + \delta\boldsymbol{u}_{q,k}^i$ for the function $f_{\sigma_q(k)}^i$, as it can only estimate the gradient at the point it plays. The gradient estimate $\widetilde{\boldsymbol{h}}_{q,k}^i$ is then computed using equation (3.30), followed by a local aggregation and variance reductions steps, the final step consisting of feeding the variance reduction vector $\widetilde{\boldsymbol{a}}_{q,k}^i$ back to the oracle $\mathcal{O}_k^i$. The remaining $L-K$ iterations are used for exploitation, where each agent plays the final decision x_q^i to obtain a high reward. We give the detail in Algorithm 8.

Lemma 3.4.2 (Lemma 1, [Zhang2019])**.** *Let $\mathcal{K}$ is down-closed convex set and δ is is sufficiently small such that $\alpha = \frac{\sqrt{d}+1}{r}\delta < 1$. The set $\mathcal{K}' = (1-\alpha)\mathcal{K} + \delta\mathbf{1}$ is convex, compact and down-closed δ-interior of $\mathcal{K}$ satisfies $d(\mathcal{K}, \mathcal{K}') \leq \left(\sqrt{d}\left(\frac{R}{e}+1\right) + \frac{R}{r}\right)\delta$*

Theorem 3.4.1. *Let $\mathcal{K}$ be a down-closed convex and compact set. We suppose the δ-interior $\mathcal{K}'$ verify Lemma 3.4.2. Let $Q = T^{2/9}, L = T^{7/9}, K = T^{2/3}$, $\delta = \frac{r}{\sqrt{d}+2}T^{-1/9}$ and $\rho_k = \frac{2}{(k+2)^{2/3}}$, $\eta_k = \frac{1}{K}$. Then the expected $\left(1-\frac{1}{e}\right)$-regret is upper bounded*

$$\mathbb{E}\left[\mathcal{R}_T\right] \leq ZT^{8/9} + \frac{\beta D^2}{2}T^{1/9} + \frac{3}{2}D\frac{d\left(\sqrt{d}+2\right)}{r}P_{n,\lambda(\mathbf{W})}T^{2/9} + \beta D^2 T^{3/9} \tag{3.31}$$

Algorithm 8 Bandit Monode Frank-Wolfe

Input: Smoothing radius δ, δ-interior $\mathcal{K}'$ with lower bound $\underline{u}$, a time horizon T, a block size L, number of exploration step K. Online linear optimization oracles $\mathcal{O}_{i,1}, \ldots, \mathcal{O}_{i,K}$ for each player $1 \leq i \leq n$, step sizes $\eta_k, \rho_k \in (0,1)$ for all $1 \leq k \leq K$, number of blocks $Q = T/L$

1: Initialize linear optimizing oracle $\mathcal{O}_k^i$ for all $1 \leq k \leq K$
2: **for** $q = 1$ to Q **do**
3: **for** every agent $1 \leq i \leq n$ **do**
4: Initialize $\boldsymbol{x}_{q,1}^i \leftarrow \underline{u}$ and set $\widetilde{\boldsymbol{a}}_{i,0}^t \leftarrow 0$
5: Update $\boldsymbol{x}_{q,k}^i$ using line 5 to 10 of Algorithm 9. Choose $\boldsymbol{x}_q^i \leftarrow \boldsymbol{x}_{q,K+1}^i$
6: Let σ_q be a random permutation of $1, \ldots, L$ — times in phase q.
7: **for** $1 \leq \ell \leq L$ **do**
8: Let $s = \sigma_q^{-1}(\ell)$
9: **if** $\ell \leq K$ **then**
10: play $f_{q,\ell}^i\left(\boldsymbol{x}_{q,s}^i + \delta \boldsymbol{u}_{q,s}^i\right)$ where $\boldsymbol{u}_{q,s}^i \in \mathbb{S}^{d-1}$. - *Exploration*
11: **else**
12: play $f_{q,\ell}^i\left(\boldsymbol{x}_q^i\right)$. - *Exploitation*
13: **end if**
14: **end for**
15: Set $\widetilde{\boldsymbol{g}}_{q,1}^i \leftarrow \frac{d}{\delta} f_{\sigma_q(1)}^i\left(\boldsymbol{x}_{q,1}^i + \delta \boldsymbol{u}_{q,1}^i\right) \boldsymbol{u}_{q,1}^i$
16: **for** $1 \leq k \leq K$ **do**
17: Let $\widetilde{\boldsymbol{h}}_{q,k}^i = \frac{d}{\delta} f_{\sigma_q(k)}^i\left(\boldsymbol{x}_{q,k}^i + \delta \boldsymbol{u}_{q,k}^i\right) \boldsymbol{u}_{q,k}^i$
18: Send $\widetilde{\boldsymbol{g}}_{q,k}^i$ to all neighbours $N(i)$.
19: After receiving $\widetilde{\boldsymbol{g}}_{q,k}^j$ from all neighbours $j \in N(i)$, compute $\widetilde{\boldsymbol{d}}_{q,k}^i \leftarrow \sum_{j \in N(i)} W_{ij} \widetilde{\boldsymbol{g}}_{q,k}^j$
20: $\widetilde{\boldsymbol{g}}_{q,k+1}^i \leftarrow \widetilde{\boldsymbol{h}}_{q,k+1}^i - \widetilde{\boldsymbol{h}}_{q,k}^i + \widetilde{\boldsymbol{d}}_{q,k}^i$
21: $\widetilde{\boldsymbol{a}}_{q,k}^i \leftarrow (1-\rho_k) \cdot \widetilde{\boldsymbol{a}}_{q,k-1}^i + \rho_k \cdot \widetilde{\boldsymbol{d}}_{q,k}^i$.
22: Feedback function $\langle \widetilde{\boldsymbol{a}}_{q,k}^i, \cdot \rangle$ to oracles $\mathcal{O}_k^i$. (The cost of the oracle $\mathcal{O}_k^i$ at block q is $\langle \widetilde{\boldsymbol{a}}_{q,k}^i, \boldsymbol{v}_{q,k}^i \rangle$.)
23: **end for**
24: **end for**
25: **end for**

where we note $Z = \left(1 - \frac{1}{e}\right)\left(\sqrt{d}\left(\frac{R}{e} + 1\right) + \frac{R}{r}\right) G \frac{r}{\sqrt{d}+2} + \left(2 - \frac{1}{e}\right) G \frac{r}{\sqrt{d}+2} + 2\beta + C$ *and* $P_{n,\lambda(\mathbf{W})} = k_0 \cdot nB \max\left\{\lambda(\mathbf{W})\left(1 + \frac{2}{1-\lambda(\mathbf{W})}\right), 2\right\} + 4^{1/3}\left(24n^2\left(\frac{1}{\frac{1}{\lambda(\mathbf{W})}-1} + 1\right)^2 + 8n\left(\frac{1}{\left(\frac{1}{\lambda(\mathbf{W})}-1\right)^2} + 2\right)\right)^{1/2}$

3.4.1 Technical Analysis

Let $f_t^\delta(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{v} \in \mathbb{B}^d}\left[f_t\left(\boldsymbol{x} + \delta \boldsymbol{v}\right)\right]$ and recall its gradient $\nabla f_t^\delta(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{u} \in \mathbb{S}^{d-1}}\left[\frac{d}{\delta} f_t\left(\boldsymbol{x} + \delta \boldsymbol{u}\right) \boldsymbol{u}\right]$. We define the average function

$$\bar{F}_{q,k}^\delta(\boldsymbol{x}) = \frac{1}{L-k} \sum_{\ell=k+1}^{L} F_{\sigma_q(\ell)}^\delta(\boldsymbol{x}) = \frac{1}{L-k} \sum_{\ell=k+1}^{L} \frac{1}{n} \sum_{i=1}^{n} f_{\sigma_q(\ell)}^{i,\delta}(\boldsymbol{x}) \tag{3.32}$$

and the average of the remaining $(L-k)$ functions of $f_{\sigma_q(\ell)}^{i,\delta}(\boldsymbol{x}_{q,\ell}^i)$ over n agents as

$$\hat{F}_{q,k}^\delta = \frac{1}{n} \sum_{i=1}^{n} \hat{f}_{q,k}^{i,\delta} = \frac{1}{L-k} \sum_{\ell=k+1}^{L} \frac{1}{n} \sum_{i=1}^{n} f_{\sigma_q(\ell)}^{i,\delta}(\boldsymbol{x}_{q,\ell}^i) \tag{3.33}$$

where $F_{\sigma_q(\ell)}^\delta(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^n f_{\sigma_q(\ell)}^{i,\delta}(\boldsymbol{x})$ and $\hat{f}_{q,k}^{i,\delta} = \frac{1}{L-k}\sum_{\ell=k+1}^L f_{\sigma_q(\ell)}^{i,\delta}(\boldsymbol{x}_{q,\ell}^i)$. Then, the one-point gradient $\nabla \bar{F}_{q,k}^\delta$ and $\nabla \hat{F}_{q,k}^\delta$ come naturally with the above definitions. Let $\mathcal{H}_{q,1} \subset \cdots \subset \mathcal{H}_{q,k}$ be the σ-fields generated by the randomness of the stochastic gradient estimate up to time k.

$$\boldsymbol{g}_{q,k}^{i,\delta} = \mathbb{E}\left[\widetilde{\boldsymbol{g}}_{q,k}^i | \mathcal{H}_{q,k-1}\right], \quad \boldsymbol{d}_{q,k}^{i,\delta} = \mathbb{E}\left[\widetilde{\boldsymbol{d}}_{q,k}^i | \mathcal{H}_{q,k-1}\right], \quad \nabla f_{\sigma_q(k)}^{i,\delta}(\boldsymbol{x}_{q,k}^i) = \mathbb{E}\left[\widetilde{\boldsymbol{h}}_{q,k}^i\right] \tag{3.34}$$

and

$$\hat{\boldsymbol{g}}_{q,k}^{i,\delta} = \frac{1}{L-k}\sum_{\ell=k+1}^{L} \boldsymbol{g}_{q,\ell}^{i,\delta}, \quad \hat{\boldsymbol{d}}_{q,k}^{i,\delta} = \frac{1}{L-k}\sum_{\ell=k+1}^{L} \boldsymbol{d}_{q,\ell}^{i,\delta}, \tag{3.35}$$

The roadmap for the proof of Theorem 3.4.1 is structured as follows:

- We start by setting bounds on the expected distance between the local gradient average $\hat{\boldsymbol{d}}^{i,\delta}q,k$ and the global gradient $\nabla\hat{F}^{\delta}q,k$, as well as between $\hat{\boldsymbol{d}}^{i,\delta}q,k$ and the variance reduction estimate $\widetilde{\boldsymbol{a}}^{i}q,k$, as detailed in Lemmas 3.4.5 and 3.4.6.
- To establish the bound in Lemma 3.4.6, we utilize auxiliary results from Lemmas 3.4.3 and 3.4.4, which provide measures for the norm and variance of the local gradient estimate.
- Leveraging the results from Lemmas 3.4.5 and 3.4.6, we then derive an upper bound on the total expected distance between the remaining global gradient and the variance reduction estimate, as shown in Proposition 3.4.1. This allows us to further derive an upper bound on the expected regret for the one-point estimate global function $F_{\sigma_q(\ell)}^{\delta}$ in terms of Q, L, and the optimal point in $\mathcal{K}'$, as outlined in Proposition 3.4.2.
- Finally, the proof of Theorem 3.4.1 is completed by applying Lemma 3.4.2 and Proposition 3.4.2 to Proposition 3.4.3.

Lemma 3.4.3. *For $i \in [n], k \in [K]$. Let $V_d^{\delta} = 2n\frac{d}{\delta}B\left(\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})}+1\right)$, the local gradient is upper-bounded, i.e $\left\|\boldsymbol{d}_{q,k}^{i,\delta}\right\| \leq V_d^{\delta}$*

Proof. See Lemma 3.7.2. □

Lemma 3.4.4. *Under Assumption 3.2.2, the variance of the local gradient estimate is uniformly bounded, i.e*

$$\mathbb{E}\left[\left\|\boldsymbol{d}_{q,k}^{i,\delta} - \widetilde{\boldsymbol{d}}_{q,k}^{i,\delta}\right\|^2\right] \leq 4n\left(\frac{d}{\delta}B\right)^2\left[\frac{1}{\left(\frac{1}{\lambda(\mathbf{W})}-1\right)^2}+2\right] \tag{3.36}$$

Proof. See Lemma 3.7.6. □

Lemma 3.4.5. *Let $N = k_0 \cdot nB\frac{d}{\delta}\max\left\{\lambda(\mathbf{W})\left(1+\frac{2}{1-\lambda(\mathbf{W})}\right), 2\right\}$. Under Assumptions 1.2.3 and 3.2.2, for $k \in [K]$, we have*

$$\max_{i\in[1,n]} \mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k}^{i,\delta} - \nabla\hat{F}_{q,k}^{\delta}\right\|\right] \leq \frac{N}{k} \tag{3.37}$$

Proof. See Lemma 3.7.7. □

Lemma 3.4.6 (Lemma 10, Lemma 11 [Zhang2019])**.** *Under Lemma 3.4.3 and lemma 3.4.4 and setting $\rho_k = \frac{2}{(k+3)^{2/3}}$, we have*

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta} - \widetilde{\boldsymbol{a}}_{q,k}^{i}\right\|\right] \leq \frac{\sqrt{M_0}}{(k+3)^{1/3}}, \qquad k \in [K] \tag{3.38}$$

where $M_0 = 4^{2/3}\frac{d^2}{\delta^2}B^2\left[24n^2\left(\frac{1}{\frac{1}{\lambda(\mathbf{W})}-1}+1\right)^2 + 8n\left(\frac{1}{\left(\frac{1}{\lambda(\mathbf{W})}-1\right)^2}+2\right)\right]$

Proof. See Lemma 3.7.8. □

Proposition 3.4.1. *Let $\overline{F}^{\delta}_{q,k-1}$ be the average of the remaining functions at iteration k (see equation (3.32)) and $\widetilde{\boldsymbol{a}}^{i}_{q,k}$ be the gradient variance reduction estimates. Let N and M_0 be defined as in Lemmas 3.4.5 and 3.4.6. Recall that K is the number of exploration steps, using Proposition 3.3.1, Assumption 1.2.1 and smoothness of $F^{\delta}_{\sigma_q(\ell)}$, for all $q \in [Q]$, we have*

$$\sum_{k=1}^{K} \mathbb{E}\left[\left\|\nabla \bar{F}^{\delta}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \widetilde{\boldsymbol{a}}^{i}_{q,k}\right\|\right] \leq \beta D + \frac{3}{2}\left(N + \sqrt{M_0}\right) K^{2/3} \tag{3.39}$$

Proof of claim.

$$\begin{aligned}
\sum_{k=1}^{K} \mathbb{E}\left[\left\|\nabla \bar{F}^{\delta}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \tilde{\boldsymbol{a}}^{i}_{q,k}\right\|\right] &\leq \sum_{k=1}^{K} \mathbb{E}\left[\left\|\nabla \bar{F}^{\delta}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \nabla \hat{F}^{\delta}_{q,k-1}\right\|\right] + \sum_{k=1}^{K} \mathbb{E}\left[\left\|\nabla \hat{F}^{\delta}_{q,k-1} - \widetilde{\boldsymbol{a}}^{i}_{q,k}\right\|\right] \\
&\leq \beta D + \sum_{k=1}^{K} \mathbb{E}\left[\left\|\nabla \hat{F}^{\delta}_{q,k-1} - \hat{\boldsymbol{d}}^{i,\delta}_{q,k-1}\right\|\right] + \sum_{k=1}^{K} \mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^{i,\delta}_{q,k-1} - \widetilde{\boldsymbol{a}}^{i}_{q,k}\right\|\right] \\
&\leq \beta D + \sum_{k=1}^{K} \frac{N}{k} + \sum_{k=1}^{K} \frac{\sqrt{M_0}}{(k+3)^{1/3}} \\
&\leq \beta D + \left(N + \sqrt{M_0}\right) \sum_{k=1}^{K} \frac{1}{(k+3)^{1/3}} \\
&\leq \beta D + \frac{3}{2}\left(N + \sqrt{M_0}\right) K^{2/3}
\end{aligned} \tag{3.40}$$

where Proposition 3.3.1 still verified in the second inequality since $f^{i,\delta}_{\sigma_q(\ell)}$ is β-smooth and the third inequality is the result of Lemma 3.4.5 and Lemma 3.4.6 □

Proposition 3.4.2. *Let $F^{\delta}_{\sigma_q(\ell)}$ be the average over n agents of the permuted one-point estimates (see equation (3.32)). Let N and M_0 be defined as in Lemmas 3.4.5 and 3.4.6 and C a constant verifies $\mathcal{R}_Q \leq C\sqrt{Q}$. By Assumption 1.2.1 and smoothness of $F^{\delta}_{\sigma_q(\ell)}$, the expected primal gap over Q blocks and L iterations is bounded by :*

$$\sum_{q=1}^{Q}\sum_{\ell=1}^{L} \mathbb{E}\left[\left(1 - \frac{1}{e}\right) F^{\delta}_{\sigma_q(\ell)}(\boldsymbol{x}^*_{\delta}) - F^{\delta}_{\sigma_q(\ell)}(\overline{\boldsymbol{x}}_q)\right] \leq \frac{L\beta D^2}{K} + \frac{3LD\left(N + \sqrt{M_0}\right)}{2K^{1/3}} + LC\sqrt{Q} + \frac{\beta Q L D^2}{2K} \tag{3.41}$$

Proof of claim. Using Lemma 3.7.5 with $F_t = \bar{F}^{\delta}_{q,k-1}$, $\boldsymbol{x}_{t,k} = \overline{\boldsymbol{x}}_{q,k}$ and $\boldsymbol{d}_{t,k} = \frac{1}{n}\sum_{i=1}^{n} \widetilde{\boldsymbol{a}}^{i}_{q,k}$, we have

$$\begin{aligned}
\bar{F}^{\delta}_{q,k-1}(\boldsymbol{x}^*_{\delta}) - \bar{F}^{\delta}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k+1}) \leq \left(1 - \frac{1}{K}\right) &\left[\bar{F}^{\delta}_{q,k-1}(\boldsymbol{x}^*_{\delta}) - \bar{F}^{\delta}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k})\right] \\
&+ \frac{1}{K}\cdot\frac{1}{n}\sum_{i=1}^{n}\left[\left\|\nabla \bar{F}^{\delta}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \widetilde{\boldsymbol{a}}^{i}_{q,k}\right\| D + \left\langle \widetilde{\boldsymbol{a}}^{i}_{q,k}, \boldsymbol{x}^*_{\delta} - \boldsymbol{v}^{i}_{q,k}\right\rangle\right] + \frac{\beta}{2}\frac{D^2}{K^2}
\end{aligned} \tag{3.42}$$

Similarly to the proof of Theorem 3.3.2 and using Proposition 3.4.1, we note

$$\begin{aligned}
&\mathbb{E}\left[\left(1 - \frac{1}{e}\right) \bar{F}^{\delta}_{q,0}(\boldsymbol{x}^*_{\delta}) - \bar{F}^{\delta}_{q,0}(\overline{\boldsymbol{x}}_q)\right] \\
&\leq \frac{1}{K}\cdot\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla \bar{F}^{\delta}_{q,k-1}(\overline{\boldsymbol{x}}_{q,k}) - \widetilde{\boldsymbol{a}}^{i}_{q,k}\right\| D\right] + \frac{1}{K}\cdot\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{E}\left[\left\langle \widetilde{\boldsymbol{a}}^{i}_{q,k}, \boldsymbol{x}^*_{\delta} - \boldsymbol{v}^{i}_{q,k}\right\rangle\right] + \frac{\beta}{2}\frac{D^2}{K} \\
&\leq \frac{D}{K}\left(\beta D + \frac{3}{2}\left(N + \sqrt{M_0}\right) K^{2/3}\right) + \frac{1}{K}\cdot\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{E}\left[\left\langle \widetilde{\boldsymbol{a}}^{i}_{q,k}, \boldsymbol{x}^*_{\delta} - \boldsymbol{v}^{i}_{q,k}\right\rangle\right] + \frac{\beta}{2}\frac{D^2}{K}
\end{aligned} \tag{3.43}$$

Thus, we can write

$$\sum_{q=1}^{Q}\sum_{\ell=1}^{L}\mathbb{E}\left[\left(1-\frac{1}{e}\right)F^{\delta}_{\sigma_q(\ell)}(\boldsymbol{x}^*_\delta)-F^{\delta}_{\sigma_q(\ell)}(\overline{\boldsymbol{x}}_q)\right]=\sum_{q=1}^{Q}L\mathbb{E}\left[\left(1-\frac{1}{e}\right)\bar{F}^{\delta}_{q,0}(\boldsymbol{x}^*_\delta)-\bar{F}^{\delta}_{q,0}(\overline{\boldsymbol{x}}_q)\right]$$
$$\leq\frac{LD}{K}\left(\beta D+\frac{3}{2}\left(N+\sqrt{M_0}\right)K^{2/3}\right)+LC\sqrt{Q}+\frac{\beta}{2}\frac{QLD^2}{K}$$
$$\leq\frac{L\beta D^2}{K}+\frac{3LD\left(N+\sqrt{M_0}\right)}{2K^{1/3}}+LC\sqrt{Q}+\frac{\beta QLD^2}{2K} \tag{3.44}$$

□

Proposition 3.4.3 (Theorem 4 [Zhang2019])**.** *Let f_t a monotone DR-Submodular function and $\hat{f}_t$ its one-point estimation. Let $\boldsymbol{x}^*_\delta$ the optimal solution in $\mathcal{K}'$ and σ_q the permutation function of block $q\in[Q]$. Using lemma 3.4.2,assumptions 1.2.2 and 3.2.2, we have*

$$\mathbb{E}\left[\mathcal{R}_T\right]\leq\left(1-\frac{1}{e}\right)d\left(\mathcal{K},\mathcal{K}'\right)GT+\left(2-\frac{1}{e}\right)GT\delta+2BQK$$
$$+\mathbb{E}\left[\sum_{q=1}^{Q}\sum_{\ell=1}^{L}\left[\left(1-\frac{1}{e}\right)\hat{f}_{\sigma_q(\ell)}(\boldsymbol{x}^*_\delta)-\hat{f}_{\sigma_q(\ell)}(\boldsymbol{x}_q)\right]\right]$$

Proof.

$$\mathbb{E}\left[\mathcal{R}_T\right]=\mathbb{E}\left[\left(1-\frac{1}{e}\right)\sum_{t=1}^{T}f_t(\boldsymbol{x}^*)-\sum_{t=1}^{T}f_t(\boldsymbol{x}_t)\right]=\mathbb{E}\left[\sum_{t=1}^{T}\left[\left(1-\frac{1}{e}\right)f_t(\boldsymbol{x}^*)-f_t(\boldsymbol{x}_t)\right]\right]$$
$$=\sum_{t=1}^{T}\mathbb{E}\left[\left(1-\frac{1}{e}\right)f_t(\boldsymbol{x}^*)-f_t(\boldsymbol{x}_t)+\left(1-\frac{1}{e}\right)f_t(\boldsymbol{x}^*_\delta)-\left(1-\frac{1}{e}\right)f_t(\boldsymbol{x}^*_\delta)\right]$$
$$=\sum_{t=1}^{T}\mathbb{E}\left[\left(1-\frac{1}{e}\right)f_t(\boldsymbol{x}^*)-\left(1-\frac{1}{e}\right)f_t(\boldsymbol{x}^*_\delta)+\left(1-\frac{1}{e}\right)f_t(\boldsymbol{x}^*_\delta)-f_t(\boldsymbol{x}_t)\right]$$
$$=\left(1-\frac{1}{e}\right)\sum_{t=1}^{T}\left[f_t(\boldsymbol{x}^*)-f_t(\boldsymbol{x}^*_\delta)\right]+\sum_{t=1}^{T}\mathbb{E}\left[\left(1-\frac{1}{e}\right)\hat{f}_t(\boldsymbol{x}^*_\delta)-\hat{f}_t(\boldsymbol{x}_t)\right]$$
$$+\left(1-\frac{1}{e}\right)\sum_{t=1}^{T}\mathbb{E}\left[f_t(\boldsymbol{x}^*_\delta)-\hat{f}_t(\boldsymbol{x}^*_\delta)\right]+\sum_{t=1}^{T}\mathbb{E}\left[\hat{f}_t(\boldsymbol{x}_t)-f_t(\boldsymbol{x}_t)\right]$$
$$\leq\left(1-\frac{1}{e}\right)d\left(\mathcal{K},\mathcal{K}'\right)GT+\sum_{t=1}^{T}\left[\left(1-\frac{1}{e}\right)\hat{f}_t(\boldsymbol{x}^*_\delta)-\hat{f}_t(\boldsymbol{x}_t)\right]+\left(1-\frac{1}{e}\right)GT\delta+GT\delta$$
$$=\left(1-\frac{1}{e}\right)d\left(\mathcal{K},\mathcal{K}'\right)GT+\left(2-\frac{1}{e}\right)GT\delta+\sum_{t=1}^{T}\mathbb{E}\left[\left(1-\frac{1}{e}\right)\hat{f}_t(\boldsymbol{x}^*_\delta)-\hat{f}_t(\boldsymbol{x}_t)\right]$$
$$=\left(1-\frac{1}{e}\right)d\left(\mathcal{K},\mathcal{K}'\right)GT+\left(2-\frac{1}{e}\right)GT\delta$$
$$+\sum_{q=1}^{Q}\sum_{\ell=1}^{L}\mathbb{E}\left[\left(1-\frac{1}{e}\right)\hat{f}_{\sigma_q(\ell)}(\boldsymbol{x}^*_\delta)-\hat{f}_{\sigma_q(\ell)}(\boldsymbol{x}_q)\right]+\sum_{q=1}^{Q}\sum_{k=1}^{K}\mathbb{E}\left[\hat{f}_{\sigma_q(\ell)}(\boldsymbol{x}_q)-\hat{f}_{\sigma_q(k)}(\boldsymbol{x}_{\sigma_q(k)})\right]$$
$$\leq\left(1-\frac{1}{e}\right)d\left(\mathcal{K},\mathcal{K}'\right)GT+\left(2-\frac{1}{e}\right)GT\delta+\sum_{q=1}^{Q}\sum_{\ell=1}^{L}\mathbb{E}\left[\left(1-\frac{1}{e}\right)\hat{f}_{\sigma_q(\ell)}(\boldsymbol{x}^*_\delta)-\hat{f}_{\sigma_q(\ell)}(\boldsymbol{x}_q)\right]+2BQK$$

□

3.4.2 Proof of Theorem 3.4.1

Proof. Recall the values of N and M_0 from Lemma 3.4.5 and Lemma 3.4.6, we have

$$N=k_0\cdot nB\frac{d}{\delta}\max\left\{\lambda(\mathbf{W})\left(1+\frac{2}{1-\lambda(\mathbf{W})}\right),2\right\}$$

$$M_0 = 4^{2/3}\frac{d^2}{\delta^2}B^2\left[24n^2\left(\frac{1}{\frac{1}{\lambda(\mathbf{W})}-1}+1\right)^2+8n\left(\frac{1}{\left(\frac{1}{\lambda(\mathbf{W})}-1\right)^2}+2\right)\right]$$

$$P_{n,\lambda(\mathbf{W})} = k_0\cdot nB\max\left\{\lambda(\mathbf{W})\left(1+\frac{2}{1-\lambda(\mathbf{W})}\right),2\right\} + 4^{1/3}\left(24n^2\left(\frac{1}{\frac{1}{\lambda(\mathbf{W})}-1}+1\right)^2+8n\left(\frac{1}{\left(\frac{1}{\lambda(\mathbf{W})}-1\right)^2}+2\right)\right)^{1/2}$$

Then one can see that $N+\sqrt{M_0} = \frac{d}{\delta}BP_{n,\lambda(\mathbf{W})}$. We let $\delta = \frac{r}{\sqrt{d}+2}T^{-1/9}$, then $\frac{d}{\delta} = \frac{d\left(\sqrt{d}+2\right)}{r}T^{1/9}$, $Q = T^{2/9}, L = T^{7/9}$ and $K = T^{2/3}$. We apply Proposition 3.4.3 with $f_t = F_t$, $\hat{f}_{\sigma_q(\ell)} = F^{\delta}_{\sigma_q(\ell)}$ and $\boldsymbol{x}_q = \overline{\boldsymbol{x}}_q$; using Lemma 3.4.2 and Proposition 3.4.2, we have

$$\begin{aligned}
\mathbb{E}\left[\mathcal{R}_T\right] &\leq \left(1-\frac{1}{e}\right)\left(\sqrt{d}\left(\frac{R}{e}+1\right)+\frac{R}{r}\right)GT\delta^{\gamma}+\left(2-\frac{1}{e}\right)GT\delta+2BQK \\
&\quad+\sum_{q=1}^{Q}\sum_{\ell=1}^{L}\mathbb{E}\left[\left(1-\frac{1}{e}\right)F^{\delta}_{\sigma_q(\ell)}(\boldsymbol{x}^*_{\delta})-F^{\delta}_{\sigma_q(\ell)}(\overline{\boldsymbol{x}}_q)\right] \\
&\leq \left(1-\frac{1}{e}\right)\left(\sqrt{d}\left(\frac{R}{e}+1\right)+\frac{R}{r}\right)GT\delta^{\gamma}+\left(2-\frac{1}{e}\right)GT\delta+2BQK \\
&\quad+\frac{L\beta D^2}{K}+\frac{3LD\left(N+\sqrt{M_0}\right)}{2K^{1/3}}+LC\sqrt{Q}+\frac{\beta QLD^2}{2K} \\
&\leq \left(1-\frac{1}{e}\right)\left(\sqrt{d}\left(\frac{R}{e}+1\right)+\frac{R}{r}\right)GT\delta^{\gamma}+\left(2-\frac{1}{e}\right)GT\delta+2BQK \\
&\quad+\frac{L\beta D^2}{K}+\frac{3LD\frac{d}{\delta}P_{n,\lambda(\mathbf{W})}}{2K^{1/3}}+LC\sqrt{Q}+\frac{\beta QLD^2}{2K} \\
&\leq \left(1-\frac{1}{e}\right)\left(\sqrt{d}\left(\frac{R}{e}+1\right)+\frac{R}{r}\right)GT\frac{r}{\sqrt{d}+2}T^{-1/9}+\left(2-\frac{1}{e}\right)GT\frac{r}{\sqrt{d}+2}T^{-1/9} \\
&\quad+2\beta T^{2/9}T^{2/3}+T^{7/9}\beta D^2T^{-2/3}+\frac{3}{2}T^{7/9}D\frac{d\left(\sqrt{d}+2\right)}{r}T^{1/9}P_{n,\lambda(\mathbf{W})}T^{-2/3} \\
&\quad+T^{7/9}CT^{1/9}+\frac{\beta}{2}T^{2/9}T^{7/9}D^2T^{-2/3} \\
&\leq \left[\left(1-\frac{1}{e}\right)\left(\sqrt{d}\left(\frac{R}{e}+1\right)+\frac{R}{r}\right)G\frac{r}{\sqrt{d}+2}+\left(2-\frac{1}{e}\right)G\frac{r}{\sqrt{d}+2}+C\right]T^{8/9} \\
&\quad+\frac{\beta D^2}{2}T^{6/9}+\left[2\beta+\frac{3}{2}D\frac{d\left(\sqrt{d}+2\right)}{r}P_{n,\lambda(\mathbf{W})}\right]T^{5/9}+\beta D^2T^{4/9} \\
&\leq \left[\left(1-\frac{1}{e}\right)\left(\sqrt{d}\left(\frac{R}{e}+1\right)+\frac{R}{r}\right)G\frac{r}{\sqrt{d}+2}+\left(2-\frac{1}{e}\right)G\frac{r}{\sqrt{d}+2}+2\beta+C\right]T^{8/9} \\
&\quad+\frac{\beta D^2}{2}T^{1/9}+\frac{3}{2}D\frac{d\left(\sqrt{d}+2\right)}{r}P_{n,\lambda(\mathbf{W})}T^{2/9}+\beta D^2T^{3/9}
\end{aligned}$$

□

3.5 Experiments

We run the algorithm on a movie recommendation problem, with the goal of identifying a set of k movies that satisfy all users. Our setting is closely related to the one in [Mokhtari2018b]

and [Xie2019]. We use the MovieLens dataset, which contains one million ratings ranging from 1 to 5 from 6000 users on 3883 movies. We divided the data set into T batches $B_1, \ldots, B_T$, with each batch B_t containing ratings from 50 users. We chose Complete, Line, Grid, and Erdos-Renyi graphs with linked probability 0.2. We set the number of nodes/agents equals to 10, 25, and 50. At each iteration t, the agent i receives a subset of ratings $B_t^i \subset B_t$. Let $\mathcal{M}$ be the set of movies and $\mathcal{U}$ the set of users; we note $r(u, m)$ the rating of user $u \in \mathcal{U}$ for movies $m \in \mathcal{M}$. Let $S \subset \mathcal{M}$ a collection of movies such that $|S| = k$, the facility location function associated to each agent i denoted,

$$f(B_t^i, S) = f_t^i(S) = \frac{1}{|B_t^i|} \sum_{u \in B_t^i} \max_{m \in S} r(u, m) \tag{3.45}$$

We denote by $\mathcal{K} = \left\{ \boldsymbol{x} \in [0,1]^d \,|\, \sum_{j=1}^d \boldsymbol{x}_j = k \right\}$. The multilinear extension of f_i^t is defined as,

$$F_t^i(\boldsymbol{x}) = \sum_{S \subset \mathcal{M}} f_t^i(S) \prod_{j \in S} \boldsymbol{x}_j \prod_{\ell \notin S} (1 - \boldsymbol{x}_\ell), \quad \forall \boldsymbol{x} \in \mathcal{K} \tag{3.46}$$

The goal is to maximize the global objective function $F_t(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^n F_t^i(\boldsymbol{x})$, subject to $\boldsymbol{x} \in \mathcal{K}$ while using only local communication and partial information for each local functions.

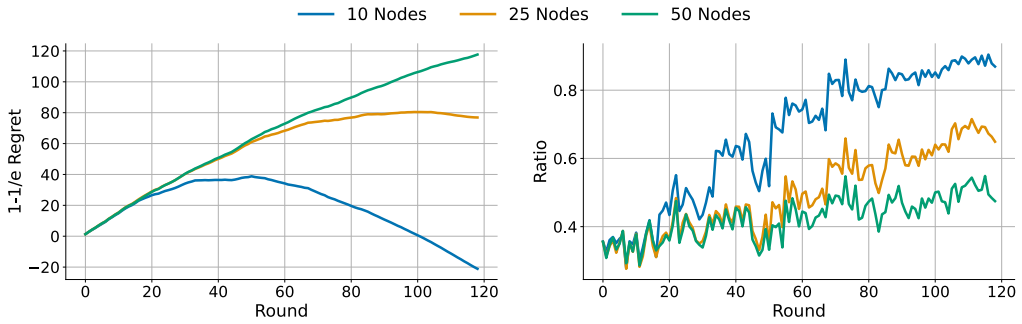


Figure 3.1: Performance of the algorithm on complete graphs with varying nodes (10, 25, 50) - Left: $(1 - 1/e)$-Regret, Right: Ratio of the algorithm's objective value to an offline centralized Frank-Wolfe.

Figure 3.1 shows the $\left(1 - \frac{1}{e}\right)$-regret of the algorithm for $k = 20$ on a complete graph with different node's configuration. We observe that increasing network size leads to a decrease in regret value, which is expected in a decentralized setting because information distributed across a larger set of nodes makes reaching consensus more difficult. Recall that the algorithm uses the same value for each function f_t in block q. If we set $K = 17$ and $Q = 6$, we can expect a stepwise-like curve since the objective function's value changes significantly at each round $t \mod 17$. In a small graph configuration, this value change is more pronounced, bringing the cumulative sum of the objective function closer to the $\left(1 - \frac{1}{e}\right)$-optimal value. Figure 3.1 depicts the ratio of our algorithm's objective value on a complete graph to an offline centralized Frank-Wolfe. As t increases, the ratio approaches one, demonstrating that our algorithm's performance is comparable to that of an offline setting if we run the algorithm for many rounds, particularly in a 10-nodes configuration. Thus, the results validate our theoretical analysis in the previous section.

Figure 3.2 shows the average value of the objective function over T rounds for all graph types when the number of movie suggestions k is varied in a 50-node configuration. The average degree for Erdos-Renyi, Complete, Grid, and Line is $5.8, 51, 5.4$, and 4, respectively. As a result, we observe lower performance on less connected graphs when compared to other graph settings. We also notice that increasing the value of k is equivalent to relaxing the cardinality constraint, which results in better performance on the objective function.

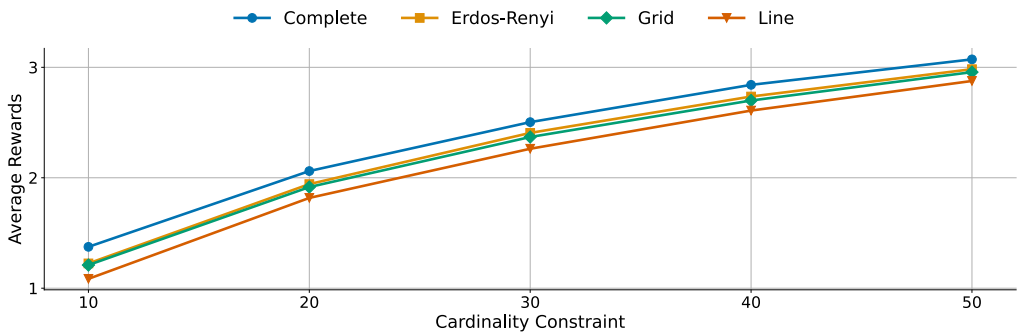


Figure 3.2: Average rewards over T rounds as function of cardinality constraint.

3.6 Concluding remarks

In this chapter, we propose a decentralized online algorithm for optimizing convex and monotone continuous DR-submodular functions with regret and $\left(1-\frac{1}{e}\right)$-regret bound of $O(T^{4/5})$. The extension of the algorithm to the bandit setting ensures a $\left(1-\frac{1}{e}\right)$-regret bound of $O(T^{8/9})$. A detailed analysis is given when the constraint set is either a general convex set or a downward-closed convex set under full information and bandit settings, respectively. In addition, the experiment results on a real-life movie recommendation problem assess the interest of the proposed algorithm for learning in decentralized settings.

3.7 Missing proofs of Chapter 3

3.7.1 Section 3.3 : Full Information Setting

Lemma 3.7.1 (Lemma 3.3.1)**.** *Suppose that each of $f^i_{\sigma_q(k)}$ is β-smooth. Using the Frank-Wolfe update of $\boldsymbol{x}^i_{q,k}$, the average of the remaining $(K-k)$ gradient approximation $\hat{\boldsymbol{d}}^i_{q,k}$ satisfies*

$$\max_{i\in[1,n]} \mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k} - \nabla \hat{F}_{q,k}\right\|\right] \leq \begin{cases} \dfrac{N}{k} & k \in \left[1, \dfrac{K}{2}\right] \\ \dfrac{N}{K-k+1} & k \in \left[\dfrac{K}{2}+1, K\right] \end{cases}$$

where $N = nGk_0 \max\{\lambda(\mathbf{W})\left(1 + \frac{2}{1-\lambda(\mathbf{W})}\right), 2\}$.

Proof. We will prove the lemma by induction using similar technique in Lemma 2.7.3. Recall the following notations

$$\hat{\boldsymbol{d}}^{cat}_{q,k} = \left[\hat{\boldsymbol{d}}^{1\top}_{q,k}, \ldots, \hat{\boldsymbol{d}}^{n\top}_{q,k}\right]^\top, \quad \hat{\boldsymbol{g}}^{cat}_{q,k} = \left[\hat{\boldsymbol{g}}^{1\top}_{q,k}, \ldots, \hat{\boldsymbol{g}}^{n\top}_{q,k}\right]^\top, \quad \nabla \hat{F}^{cat}_{q,k} = \left[\nabla \hat{F}^\top_{q,k}, \ldots, \nabla \hat{F}^\top_{q,k}\right]^\top \tag{3.47}$$

and let the slack variables as

$$\delta^i_{q,k} := \nabla \hat{f}^i_{q,k} - \nabla \hat{f}^i_{q,k-1}, \quad \bar{\delta}_{q,k} := \frac{1}{n}\sum_{i=1}^n \left(\nabla \hat{f}^i_{q,k} - \nabla \hat{f}^i_{q,k-1}\right) = \nabla \hat{F}_{q,k} - \nabla \hat{F}_{q,k-1} \tag{3.48}$$

then, following the definition in 2.17, we note

$$\delta^{cat}_{q,k} = \left[\delta^{1\top}_{q,k}, \ldots, \delta^{n\top}_{q,k}\right]^\top, \quad \bar{\delta}^{cat}_{q,k} = \left[\bar{\delta}^\top_{q,k}, \ldots, \bar{\delta}^\top_{q,k}\right]^\top$$

By similar analysis to equation (2.31), we have

$$\begin{aligned} \left\|\hat{\boldsymbol{d}}^{cat}_{q,k} - \nabla \hat{F}^{cat}_{q,k}\right\|^2 &= \sum_{i=1}^n \left\|\hat{\boldsymbol{d}}^i_{q,k} - \nabla \hat{F}_{q,k}\right\|^2 \\ &\leq \lambda(\mathbf{W})^2 \sum_{i=1}^n \left\|\hat{\boldsymbol{g}}^i_{q,k} - \nabla \hat{F}_{q,k}\right\|^2 \\ &= \lambda(\mathbf{W})^2 \left\|\hat{\boldsymbol{g}}^{cat}_{q,k} - \nabla \hat{F}^{cat}_{q,k}\right\|^2 \end{aligned} \tag{3.49}$$

We can deduce that

$$\begin{aligned} \mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^{cat}_{q,k} - \nabla \hat{F}^{cat}_{q,k}\right\|\right] &\leq \lambda(\mathbf{W})\mathbb{E}\left[\left\|\hat{\boldsymbol{g}}^{cat}_{q,k} - \nabla \hat{F}^{cat}_{q,k}\right\|\right] \\ &= \lambda(\mathbf{W})\mathbb{E}\left[\left\|\delta^{cat}_{q,k} + \hat{\boldsymbol{d}}^{cat}_{q,k-1} - \nabla \hat{F}^{cat}_{q,k} + \nabla \hat{F}^{cat}_{q,k-1} - \nabla \hat{F}^{cat}_{q,k-1}\right\|\right] \\ &\leq \lambda(\mathbf{W})\left(\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^{cat}_{q,k-1} - \nabla \hat{F}^{cat}_{q,k-1}\right\|\right] + \mathbb{E}\left[\left\|\delta^{cat}_{q,k} - \bar{\delta}^{cat}_{q,k}\right\|\right]\right) \\ &\leq \lambda(\mathbf{W})\left(\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^{cat}_{q,k-1} - \nabla \hat{F}^{cat}_{q,k-1}\right\|\right] + \mathbb{E}\left[\left\|\delta^{cat}_{q,k}\right\|\right]\right) \end{aligned} \tag{3.50}$$

since

$$\left\|\delta^{cat}_{q,k} - \bar{\delta}^{cat}_{q,k}\right\|^2 = \sum_{i=1}^n \left\|\delta^i_{q,k} - \bar{\delta}_{q,k}\right\|^2 \leq \sum_{i=1}^n \left\|\delta^i_{q,k}\right\|^2 - n\left\|\bar{\delta}_{q,k}\right\|^2 \leq \sum_{i=1}^n \left\|\delta^i_{q,k}\right\|^2 = \left\|\delta^{cat}_{q,k}\right\|^2 \tag{3.51}$$

Notice that we can bound the expected value of δ^{cat} by

$$\begin{aligned}
\mathbb{E}\left[\left\|\delta_{q,k}^{cat}\right\|^2\right] &= \mathbb{E}\left[\sum_{i=1}^{n}\left\|\delta_{q,k}^{i}\right\|^2\right] = \sum_{i=1}^{n}\mathbb{E}\left[\left\|\nabla\hat{f}_{q,k}^{i} - \nabla\hat{f}_{q,k-1}^{i}\right\|^2\right] \\
&= \sum_{i=1}^{n}\mathbb{E}\left[\mathbb{E}\left[\left\|\nabla\hat{f}_{q,k}^{i} - \nabla\hat{f}_{q,k-1}^{i}\right\|^2 \mid \mathcal{F}_{q,k-1}\right]\right] \\
&= \sum_{i=1}^{n}\mathbb{E}\left[\mathbb{E}\left[\left\|\frac{\sum_{\ell=k+1}^{K}\nabla f_{\sigma_q(\ell)}^{i}(\boldsymbol{x}_{q,\ell}^{i})}{K-k} - \frac{\sum_{\ell=k}^{K}\nabla f_{\sigma_q(\ell)}^{i}(\boldsymbol{x}_{q,\ell}^{i})}{K-k+1}\right\|^2 \mid \mathcal{F}_{q,k-1}\right]\right] \\
&= \sum_{i=1}^{n}\mathbb{E}\left[\mathbb{E}\left[\left\|\frac{\sum_{\ell=k+1}^{K}\nabla f_{\sigma_q(\ell)}^{i}(\boldsymbol{x}_{q,\ell}^{i})}{(K-k)(K-k+1)} - \frac{\nabla f_{\sigma_q(k)}^{i}(\boldsymbol{x}_{q,k}^{i})}{K-k+1}\right\|^2 \mid \mathcal{F}_{q,k-1}\right]\right] \\
&\leq n\left(\frac{2G}{K-k+1}\right)^2
\end{aligned} \tag{3.52}$$

using Jensen's inequality, we can deduce that

$$\mathbb{E}\left[\left\|\delta_{q,k}^{cat}\right\|\right] \leq \sqrt{\mathbb{E}\left[\left\|\delta_{q,k}^{cat}\right\|^2\right]} \leq \frac{2\sqrt{n}G}{K-k+1} \tag{3.53}$$

We are now proving the lemma by induction, when $k = 1$, we have

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,1}^{cat} - \nabla\hat{F}_{q,1}^{cat}\right\|^2\right] &= \mathbb{E}\left[\sum_{i=1}^{n}\left\|\hat{\boldsymbol{d}}_{q,1}^{i} - \nabla\hat{F}_{q,1}\right\|^2\right] \leq \lambda(\mathbf{W})^2\mathbb{E}\left[\sum_{i=1}^{n}\left\|\hat{\boldsymbol{g}}_{q,1}^{i} - \nabla\hat{F}_{q,1}\right\|^2\right] \\
\leq \lambda(\mathbf{W})^2\mathbb{E}\left[\sum_{i=1}^{n}\left\|\nabla\hat{f}_{q,1}^{i} - \nabla\hat{F}_{q,1}\right\|^2\right] &\leq \lambda(\mathbf{W})^2\mathbb{E}\left[\sum_{i=1}^{n}\left\|\nabla\hat{f}_{q,1}^{i}\right\|^2\right] \leq n\lambda(\mathbf{W})^2G^2
\end{aligned}$$

where we have used Lipschitzness of f in the last inequality. We now suppose that $1 \leq k \leq k_0$, from equations (2.32) and (3.53)

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k}^{cat} - \nabla\hat{F}_{q,k}^{cat}\right\|\right] &\leq \lambda(\mathbf{W})\left(\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{cat} - \nabla\hat{F}_{q,k-1}^{cat}\right\|\right] + \mathbb{E}\left[\left\|\delta_{q,k}^{cat}\right\|\right]\right) \\
&\leq \lambda(\mathbf{W})^{k-1}\sqrt{n}G + 2\sum_{\tau=1}^{k}\lambda(\mathbf{W})^{\tau}\sqrt{n}G \\
&\leq \lambda(\mathbf{W})\sqrt{n}G + 2\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})}\sqrt{n}G \\
&= \lambda(\mathbf{W})\sqrt{n}G\left(1 + \frac{2}{1-\lambda(\mathbf{W})}\right)
\end{aligned} \tag{3.54}$$

We set $N_0 = k_0\sqrt{n}G\max\{\lambda(\mathbf{W})\left(1+\frac{2}{1-\lambda(\mathbf{W})}\right), 2\}$, we claim that $\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k}^{cat} - \nabla\hat{F}_{q,k}^{cat}\right\|\right] \leq \frac{N_0}{k}$ for $k \in \left[k_0, \frac{K}{2}+1\right]$. Recall that $K-k+1 \geq k-1$, by equations (2.32) and (3.53) and induction hypothesis, we have

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k}^{cat} - \nabla\hat{F}_{q,k}^{cat}\right\|\right] &\leq \lambda(\mathbf{W})\left(\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{cat} - \nabla\hat{F}_{q,k-1}^{cat}\right\|\right] + \mathbb{E}\left[\left\|\delta_{q,k}^{cat}\right\|\right]\right) \\
&\leq \lambda(\mathbf{W})\left(\frac{N_0}{k-1} + \frac{2\sqrt{n}G}{K-k+1}\right) \\
&\leq \lambda(\mathbf{W})\left(\frac{N_0}{k-1} + \frac{2\sqrt{n}G}{k-1}\right) \\
&\leq \lambda(\mathbf{W})\left(\frac{N_0 + 2\sqrt{n}G}{k-1}\right) \\
&\leq \lambda(\mathbf{W})\left(N_0\frac{k_0+1}{k_0(k-1)}\right) \\
&\leq \frac{N_0}{k}
\end{aligned} \tag{3.55}$$

where we have used the fact that $\lambda(\mathbf{W})(\mathbf{W})\frac{k_0+1}{k_0(k-1)} \leq \frac{1}{k}$ in the last inequality. When $k \in \left[\frac{K}{2}+1, K\right]$, we claim that $\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k}^{cat} - \nabla \hat{F}_{q,k}^{cat}\right\|\right] \leq \frac{N_0}{K-k+1}$. The base case $k = \frac{K}{2}+1$ is verified by equation (3.55),

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k}^{cat} - \nabla \hat{F}_{q,k}^{cat}\right\|\right] \leq \frac{N_0}{\frac{K}{2}+1} \leq \frac{N_0}{\frac{K}{2}} \leq \frac{N_0}{K-(\frac{K}{2}+1)+1} \tag{3.56}$$

For $k \geq \frac{K}{2}+2$, using equations (2.32) and (3.53) and the induction hypothesis, we have

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k}^{cat} - \nabla \hat{F}_{q,k}^{cat}\right\|\right] &\leq \lambda(\mathbf{W})\left(\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{cat} - \nabla \hat{F}_{q,k-1}^{cat}\right\|\right] + \mathbb{E}\left[\left\|\delta_{q,k}^{cat}\right\|\right]\right) \\
&\leq \lambda(\mathbf{W})\left(\frac{N_0}{K-k+2} + \frac{2\sqrt{n}G}{K-k+1}\right) \\
&\leq \lambda(\mathbf{W})\left(\frac{N_0+2G}{K-k+1}\right) \\
&\leq \lambda(\mathbf{W})\left(N_0\frac{k_0+1}{k_0(K-k+1)}\right) \\
&\leq \frac{N_0}{K-k+1}
\end{aligned} \tag{3.57}$$

Recall that

$$\frac{1}{\sqrt{n}}\mathbb{E}\left[\sum_{i=1}^{n}\left\|\hat{\boldsymbol{d}}_{q,k}^{i} - \nabla \hat{F}_{q,k}\right\|\right] \leq \mathbb{E}\left[\left(\sum_{i=1}^{n}\left\|\hat{\boldsymbol{d}}_{q,k}^{i} - \nabla \hat{F}_{q,k}\right\|^2\right)^{1/2}\right] = \mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k}^{cat} - \nabla \hat{F}_{q,k}^{cat}\right\|\right] \tag{3.58}$$

The desired result followed from equations (3.55), (3.57) and (3.58) where $N = \sqrt{n}N_0$

$$\max_{i\in[1,n]}\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k}^{i} - \nabla \hat{F}_{q,k}\right\|\right] \leq \begin{cases} \frac{N}{k} & k \in \left[1, \frac{K}{2}\right] \\ \frac{N}{K-k+1} & k \in \left[\frac{K}{2}+1, K\right] \end{cases} \tag{3.59}$$

□

Lemma 3.7.2 (Lemma 3.3.2)**.** *Let $V_d = 2nG\left(\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})}+1\right)$, the local gradient is uniformly upper-bounded, i.e, $\forall i \in [n], \forall k \in [K]$. $\left\|\boldsymbol{d}_{q,k}^{i}\right\| \leq V_d$.*

Proof. We use the same notation introduced in equation (2.17). Let's define

$$\boldsymbol{d}_{q,k}^{cat} = \left[\boldsymbol{d}_{q,k}^{1\top}, \ldots, \boldsymbol{d}_{q,k}^{n\top}\right]^\top \in \mathbb{R}^{nd}, \quad \nabla f_{\sigma_q(k)}^{cat} = \left[\nabla f_{\sigma_q(k)}^{1}(\boldsymbol{x}_{q,k}^{1})^\top, \ldots, \nabla f_{\sigma_q(k)}^{n}(\boldsymbol{x}_{q,k}^{n})^\top\right]^\top \in \mathbb{R}^{nd} \tag{3.60}$$

and

$$\nabla F_{\sigma_q(k)}^{cat} = \left[\nabla F_{\sigma_q(k)}^{\top}, \ldots, \nabla F_{\sigma_q(k)}^{\top}\right]^\top = \left[\frac{1}{n}\sum_{i=1}^{n}\nabla f_{\sigma_q(k)}^{i}(\boldsymbol{x}_{q,k}^{i})^\top, \ldots, \frac{1}{n}\sum_{i=1}^{n}\nabla f_{\sigma_q(k)}^{i}(\boldsymbol{x}_{q,k}^{i})^\top\right]^\top \tag{3.61}$$

Using the local gradient expansion in proposition 5, we have

$$\begin{aligned}
\boldsymbol{d}_{q,k}^{cat} = &\sum_{\tau=1}^{k-1}\left[\left(\mathbf{W}^{k-\tau} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\left(\nabla f_{\sigma_q(\tau+1)}^{cat} - \nabla f_{\sigma_q(\tau)}^{cat}\right) \\
&+ \left[\left(\mathbf{W}^{k} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\nabla f_{\sigma_q(1)}^{cat} + \left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T \otimes I_d\right)\nabla f_{\sigma_q(k)}^{cat}
\end{aligned} \tag{3.62}$$

Recall that $\|\mathbf{W} \otimes I_d\| = \|\mathbf{W}\|$. Taking the norm on equation (3.62), we have

$$\left\|\boldsymbol{d}_{q,k}^{cat}\right\| \leq 2\sqrt{n}G\sum_{\tau=1}^{k-1}\lambda(\mathbf{W})^{k-\tau} + \sqrt{n}G\left(\lambda(\mathbf{W})^k + 1\right) \leq 2\sqrt{n}G\left(\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})} + 1\right) \tag{3.63}$$

where we have used $\left\|\nabla f_{\sigma_q(\tau+1)}^{cat} - \nabla f_{\sigma_q(\tau)}^{cat}\right\| \leq 2\sqrt{n}G$, $\left\|\mathbf{W}^k - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\| \leq \lambda(\mathbf{W})^k$ and $\left\|\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\| \leq 1$ in the first inequality. We have $\forall i \in [n]$

$$\left\|\boldsymbol{d}_{q,k}^{i}\right\| \leq \sum_{i=1}^{n}\left\|\boldsymbol{d}_{q,k}^{i}\right\| \leq \sqrt{n}\left(\sum_{i=1}^{n}\left\|\boldsymbol{d}_{q,k}^{i}\right\|^2\right)^{1/2} = \sqrt{n}\left\|\boldsymbol{d}_{q,k}^{cat}\right\| \tag{3.64}$$

one can obtain the desired result. □

Lemma 3.7.3 (Lemma 3.3.3)**.** *Under Assumption 3.2.1 and let* $\sigma_1^2 = 4n\left[\left(\frac{G+G_0}{\frac{1}{\lambda(\mathbf{W})}-1}\right)^2 + 2\sigma_0^2\right]$. *For* $i \in [n], k \in [K]$*, the variance of the local stochastic gradient is uniformly bounded i.e*

$$\mathbb{E}\left[\left\|\boldsymbol{d}_{q,k}^{i} - \widetilde{\boldsymbol{d}}_{q,k}^{i}\right\|^2\right] \leq \sigma_1^2$$

Proof. We denote $\widetilde{\boldsymbol{d}}^{cat}$ the stochastique version of $\boldsymbol{d}^{cat}$, following equation (3.62), we have

$$\begin{aligned}\widetilde{\boldsymbol{d}}_{q,k}^{cat} = &\sum_{\tau=1}^{k-1}\left[\left(\mathbf{W}^{k-\tau} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\left(\widetilde{\nabla} f_{\sigma_q(\tau+1)}^{cat} - \widetilde{\nabla} f_{\sigma_q(\tau)}^{cat}\right)\\ &+\left[\left(\mathbf{W}^{k} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\widetilde{\nabla} f_{\sigma_q(1)}^{cat} + \left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T \otimes I_d\right)\widetilde{\nabla} f_{\sigma_q(k)}^{cat}\end{aligned} \tag{3.65}$$

Then, we have

$$\begin{aligned}\boldsymbol{d}_{q,k}^{cat} - \widetilde{\boldsymbol{d}}_{q,k}^{cat} = &\sum_{\tau=1}^{k-1}\left[\left(\mathbf{W}^{k-\tau} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\left(\nabla f_{\sigma_q(\tau+1)}^{cat} - \widetilde{\nabla} f_{\sigma_q(\tau+1)}^{cat} + \widetilde{\nabla} f_{\sigma_q(\tau)}^{cat} - \nabla f_{\sigma_q(\tau)}^{cat}\right)\\ &+\left[\left(\mathbf{W}^{k} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\left(\nabla f_{\sigma_q(1)}^{cat} - \widetilde{\nabla} f_{\sigma_q(1)}^{cat}\right) + \left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T \otimes I_d\right)\left(\nabla f_{\sigma_q(k)}^{cat} - \widetilde{\nabla} f_{\sigma_q(k)}^{cat}\right)\end{aligned} \tag{3.66}$$

By Assumption 3.2.1 and Jensen's inequality, we have

$$\begin{aligned}\mathbb{E}\left[\left\|\nabla f_{\sigma_q(\tau)}^{cat} - \widetilde{\nabla} f_{\sigma_q(\tau)}^{cat}\right\|^2\right] &= \mathbb{E}\left[\sum_{i=1}^{n}\left\|\nabla f_{\sigma_q(\tau)}^{i}\left(\boldsymbol{x}_{q,\tau}^{i}\right) - \widetilde{\nabla} f_{\sigma_q(\tau)}^{i}\left(\boldsymbol{x}_{q,\tau}^{i}\right)\right\|^2\right]\\ &\leq \sqrt{\sum_{i=1}^{n}\mathbb{E}\left[\left\|\nabla f_{\sigma_q(\tau)}^{i}\left(\boldsymbol{x}_{q,\tau}^{i}\right) - \widetilde{\nabla} f_{\sigma_q(\tau)}^{i}\left(\boldsymbol{x}_{q,\tau}^{i}\right)\right\|^2\right]} \leq \sqrt{n}\sigma_0\end{aligned} \tag{3.67}$$

Taking the second moment of equation (4.12), we have

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\boldsymbol{d}_{q,k}^{cat}-\widetilde{\boldsymbol{d}}_{q,k}^{cat}\right\|^2\right]\\
\leq&\mathbb{E}\left[\left(\sum_{\tau=1}^{k-1}\left\|\mathbf{W}^{k-\tau}-\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\|\left\|\nabla f_{\sigma_q(\tau+1)}^{cat}-\widetilde{\nabla} f_{\sigma_q(\tau+1)}^{cat}+\widetilde{\nabla} f_{\sigma_q(\tau)}^{cat}-\nabla f_{\sigma_q(\tau)}^{cat}\right\|\right)^2\right]\\
&+\mathbb{E}\left[\left\|\left(\mathbf{W}^k-\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\otimes I_d\right)\left(\nabla f_{\sigma_q(1)}^{cat}-\widetilde{\nabla} f_{\sigma_q(1)}^{cat}\right)+\left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\otimes I_d\right)\left(\nabla f_{\sigma_q(k)}^{cat}-\widetilde{\nabla} f_{\sigma_q(k)}^{cat}\right)\right\|^2\right]\\
\leq&\mathbb{E}\left[\left(\sum_{\tau=1}^{k-1}\left\|\mathbf{W}^{k-\tau}-\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\|\left\|\nabla f_{\sigma_q(\tau+1)}^{cat}-\widetilde{\nabla} f_{\sigma_q(\tau+1)}^{cat}+\widetilde{\nabla} f_{\sigma_q(\tau)}^{cat}-\nabla f_{\sigma_q(\tau)}^{cat}\right\|\right)^2\right]\\
&+4\left(\mathbb{E}\left[\left\|\mathbf{W}^k-\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\|^2\left\|\nabla f_{\sigma_q(1)}^{cat}-\widetilde{\nabla} f_{\sigma_q(1)}^{cat}\right\|^2\right]+\mathbb{E}\left[\left\|\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\|^2\left\|\nabla f_{\sigma_q(k)}^{cat}-\widetilde{\nabla} f_{\sigma_q(k)}^{cat}\right\|^2\right]\right)\\
\leq&4n\left(G+G_0\right)^2\left(\sum_{\tau=1}^{k-1}\lambda(\mathbf{W})^{k-\tau}\right)^2+4n\sigma_0^2\left(\lambda(\mathbf{W})^{2k}+1\right)\\
\leq&4n\left(G+G_0\right)^2\left(\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})}\right)^2+4n\sigma_0^2(\lambda(\mathbf{W})+1)\leq 4n\left[\left(\frac{G+G_0}{\frac{1}{\lambda(\mathbf{W})}-1}\right)^2+2\sigma_0^2\right]
\end{aligned}
\tag{3.68}
$$

where the first inequality holds since $\mathbb{E}\left[\nabla f_{\sigma_q(\tau+1)}^{cat}-\widetilde{\nabla} f_{\sigma_q(\tau+1)}^{cat}+\widetilde{\nabla} f_{\sigma_q(\tau)}^{cat}-\nabla f_{\sigma_q(\tau)}^{cat}\right]=0$. The second inequality follows the fact that $\|a+b\|^2\leq 4\left(\|a\|^2+\|b\|^2\right)$. The third inequality comes from Assumption 3.2.1 and the analysis in Lemma 3.3.2. Finally, one can obtain the desired result by noticing $\mathbb{E}\left[\left\|\boldsymbol{d}_{q,k}^i-\widetilde{\boldsymbol{d}}_{q,k}^i\right\|^2\right]\leq\sum_{i=1}^n\mathbb{E}\left[\left\|\boldsymbol{d}_{q,k}^i-\widetilde{\boldsymbol{d}}_{q,k}^i\right\|^2\right]=\mathbb{E}\left[\left\|\boldsymbol{d}_{q,k}^{cat}-\widetilde{\boldsymbol{d}}_{q,k}^{cat}\right\|^2\right]$ □

Lemma 3.7.4 (Lemma 3.3.4)**.** *Under Assumption 3.3.1, Lemma 3.3.2, Lemma 3.3.3 and setting* $\rho_k=\frac{2}{(k+3)^{2/3}}$ *and* $\rho_k=\frac{1.5}{(K-k+2)^{2/3}}$ *for* $k\in\left[\frac{K}{2}\right]$ *and* $k\in\left[\frac{K}{2}+1,K\right]$ *respectively, we have*

$$
\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^i-\widetilde{\boldsymbol{a}}_{q,k}^i\right\|\right]\leq\begin{cases}\dfrac{\sqrt{M}}{(k+4)^{1/3}} & k\in\left[\dfrac{K}{2}\right]\\ \dfrac{\sqrt{M}}{(K-k+1)^{1/3}} & i\in\left[\dfrac{K}{2}+1,K\right]\end{cases}
\tag{3.69}
$$

where $M=\max\{M_1,M_2\}$ *where* $M_1=\max\{5^{2/3}\left(V_{\boldsymbol{d}}+L_0\right)^2,M_0\}$, $M_0=4\left(V_{\boldsymbol{d}}^2+\sigma^2\right)+32\sqrt{2}V_{\boldsymbol{d}}$ *and* $M_2=2.55\left(V_{\boldsymbol{d}}^2+\sigma^2\right)+\dfrac{7\sqrt{2}V_{\boldsymbol{d}}}{3}$ *and* $L_0=\frac{2}{4^{2/3}}\left\|\widetilde{\boldsymbol{d}}_{q,1}^i\right\|$

Proof. In order to prove the lemma, we only need to bound $\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^i-\widetilde{\boldsymbol{a}}_{q,k}^i\right\|^2\right]$, following the decomposition in [Zhang2019], we have

$$
\begin{aligned}
\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^i-\widetilde{\boldsymbol{a}}_{q,k}^i\right\|^2\right]&=\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^i-(1-\rho_k)\widetilde{\boldsymbol{a}}_{q,k-1}^i-\rho_k\widetilde{\boldsymbol{d}}_{q,k}^i\right\|^2\right]\\
&=\rho_k^2\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^i-\widetilde{\boldsymbol{d}}_{q,k}^i)\right\|^2\right]+(1-\rho_k)^2\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^i-\hat{\boldsymbol{d}}_{q,k-2}^i\right\|^2\right] \qquad (3.70)\\
&\quad+(1-\rho_k)^2\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-2}^i-\widetilde{\boldsymbol{a}}_{q,k-1}^i\right\|^2\right]\\
&\quad+2\rho_k(1-\rho_k)\mathbb{E}\left[\left\langle\hat{\boldsymbol{d}}_{q,k-1}^i-\widetilde{\boldsymbol{d}}_{q,k}^i,\hat{\boldsymbol{d}}_{q,k-1}^i-\hat{\boldsymbol{d}}_{q,k-2}^i\right\rangle\right]\\
&\quad+2\rho_k(1-\rho_k)\mathbb{E}\left[\left\langle\hat{\boldsymbol{d}}_{q,k-1}^i-\widetilde{\boldsymbol{d}}_{q,k}^i,\hat{\boldsymbol{d}}_{q,k-2}^i-\widetilde{\boldsymbol{a}}_{q,k-1}^i\right\rangle\right]\\
&\quad+2(1-\rho_k)^2\mathbb{E}\left[\left\langle\hat{\boldsymbol{d}}_{q,k-1}^i-\hat{\boldsymbol{d}}_{q,k-2}^i,\hat{\boldsymbol{d}}_{q,k-2}^i-\widetilde{\boldsymbol{a}}_{q,k-1}^i\right\rangle\right] \qquad (3.71)
\end{aligned}
$$

The first part of the above equation is written as

$$\begin{aligned}
&\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1}-\widetilde{\boldsymbol{d}}^i_{q,k})\right\|^2\right]=\mathbb{E}\left[\mathbb{E}_\sigma\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1}-\widetilde{\boldsymbol{d}}^i_{q,k})\right\|^2\mid\mathcal{F}_{q,k-1}\right]\right]\\
&=\mathbb{E}\left[\mathbb{E}_\sigma\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1}-\boldsymbol{d}^i_{q,k}+\boldsymbol{d}^i_{q,k}-\widetilde{\boldsymbol{d}}^i_{q,k})\right\|^2\mid\mathcal{F}_{q,k-1}\right]\right]\\
&\leq\mathbb{E}\left[\mathbb{E}_\sigma\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1}-\boldsymbol{d}^i_{q,k}\right\|^2+\left\|\boldsymbol{d}^i_{q,k}-\widetilde{\boldsymbol{d}}^i_{q,k})\right\|^2+2\langle\hat{\boldsymbol{d}}^i_{q,k-1}-\boldsymbol{d}^i_{q,k},\boldsymbol{d}^i_{q,k}-\widetilde{\boldsymbol{d}}^i_{q,k}\rangle\mid\mathcal{F}_{q,k-1}\right]\right] \quad (3.72)
\end{aligned}$$

Using the definition of $\hat{\boldsymbol{d}}^i_{q,k-1}$, Lemma 3.7.2 and Lemma 4.4.1 and law of total expectation, we have

$$\mathbb{E}\left[\mathbb{E}_\sigma\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1}-\boldsymbol{d}^i_{q,k}\right\|^2\mid\mathcal{F}_{q,k-1}\right]\right]=\mathbb{E}\left[Var_\sigma\left(\boldsymbol{d}^i_{q,k}\mid\mathcal{F}_{q,k-1}\right)\right]\leq\mathbb{E}\left[\mathbb{E}_\sigma\left[\left\|\boldsymbol{d}^i_{q,k}\right\|^2\mid\mathcal{F}_{q,k-1}\right]\right]\leq V_{\boldsymbol{d}}^2 \quad (3.73)$$

$$\mathbb{E}\left[\mathbb{E}_\sigma\left[\left\|\boldsymbol{d}^i_{q,k}-\widetilde{\boldsymbol{d}}^i_{q,k})\right\|^2\right]|\mathcal{F}_{q,k-1}\right]\leq\sigma_1^2 \quad (3.74)$$

Recall that $\mathcal{H}_{q,k}$ is the filtration related to the randomness of $\widetilde{\boldsymbol{d}}^i_{q,k}$ and $\hat{\boldsymbol{d}}^i_{q,k-1}$ and $\boldsymbol{d}^i_{q,k}$ is $\mathcal{F}_{q,k}$-measurable, then one can write

$$\begin{aligned}
&\mathbb{E}\left[\mathbb{E}_\sigma\left[\left\langle\hat{\boldsymbol{d}}^i_{q,k-1}-\boldsymbol{d}^i_{q,k},\boldsymbol{d}^i_{q,k}-\widetilde{\boldsymbol{d}}^i_{q,k}\right\rangle\mid\mathcal{F}_{q,k-1}\right]\right]\\
=&\mathbb{E}\left[\left\langle\hat{\boldsymbol{d}}^i_{q,k-1}-\boldsymbol{d}^i_{q,k},\boldsymbol{d}^i_{q,k}-\widetilde{\boldsymbol{d}}^i_{q,k}\right\rangle\right]\\
=&\mathbb{E}\left[\mathbb{E}_\sigma\left[\left\langle\hat{\boldsymbol{d}}^i_{q,k-1}-\boldsymbol{d}^i_{q,k},\boldsymbol{d}^i_{q,k}-\widetilde{\boldsymbol{d}}^i_{q,k}\right\rangle\mid\mathcal{F}_{q,k}\right]\right]\\
=&\mathbb{E}\left[\left\langle\hat{\boldsymbol{d}}^i_{q,k-1}-\boldsymbol{d}^i_{q,k},\mathbb{E}_\sigma\left[\boldsymbol{d}^i_{q,k}-\widetilde{\boldsymbol{d}}^i_{q,k}\mid\mathcal{F}_{q,k}\right]\right\rangle\right] && \text{(by } \mathcal{F}_{q,k}\text{-measurability)}\\
=&\mathbb{E}\left[\mathbb{E}_{\widetilde{\boldsymbol{d}}}\left[\left\langle\hat{\boldsymbol{d}}^i_{q,k-1}-\boldsymbol{d}^i_{q,k},\mathbb{E}_\sigma\left[\boldsymbol{d}^i_{q,k}-\widetilde{\boldsymbol{d}}^i_{q,k}\mid\mathcal{F}_{q,k}\right]\right\rangle\right]\mid\mathcal{H}_{q,k-1}\right]\\
=&\mathbb{E}\left[\left\langle\hat{\boldsymbol{d}}^i_{q,k-1}-\boldsymbol{d}^i_{q,k},\mathbb{E}_{\widetilde{\boldsymbol{d}}}\left[\mathbb{E}_\sigma\left[\boldsymbol{d}^i_{q,k}-\widetilde{\boldsymbol{d}}^i_{q,k}\mid\mathcal{F}_{q,k}\right]\mid\mathcal{H}_{q,k-1}\right]\right\rangle\right]\\
=&\mathbb{E}\left[\left\langle\hat{\boldsymbol{d}}^i_{q,k-1}-\boldsymbol{d}^i_{q,k},\mathbb{E}_\sigma\left[\mathbb{E}_{\widetilde{\boldsymbol{d}}}\left[\boldsymbol{d}^i_{q,k}-\widetilde{\boldsymbol{d}}^i_{q,k}\mid\mathcal{H}_{q,k-1}\right]\mid\mathcal{F}_{q,k}\right]\right\rangle\right] && \text{(by Fubini's theorem)}\\
=&0 \quad (3.75)
\end{aligned}$$

where the last equation holds since $\mathbb{E}_{\widetilde{\boldsymbol{d}}}\left[\widetilde{\boldsymbol{d}}^i_{q,k}\mid\mathcal{H}_{q,k-1}\right]=\boldsymbol{d}^i_{q,k}$. Combining equations (3.73) to (3.75), equation (3.72) is upper bounded by

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1}-\widetilde{\boldsymbol{d}}^i_{q,k})\right\|^2\right]\leq V_{\boldsymbol{d}}^2+\sigma_1^2\triangleq V \quad (3.76)$$

We are now bounding $\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1}-\hat{\boldsymbol{d}}^i_{q,k-2}\right\|^2\right]$, using the definition of $\hat{\boldsymbol{d}}^i_{q,k}$ and Lemma 3.7.2. We

have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1}-\hat{\boldsymbol{d}}^i_{q,k-2}\right\|^2\right]\\
=&\mathbb{E}\left[\mathbb{E}_\sigma\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1}-\hat{\boldsymbol{d}}^i_{q,k-2}\right\|^2 \mid \mathcal{F}_{q,k-2}\right]\right]\\
=&\mathbb{E}\left[\mathbb{E}_\sigma\left[\left\|\frac{\sum_{\ell=k}^K \boldsymbol{d}^i_{q,\ell}}{K-k+1}-\frac{\sum_{\ell=k-1}^K \boldsymbol{d}^i_{q,\ell}}{K-k+2}\right\|^2 \mid \mathcal{F}_{q,k-2}\right]\right]\\
=&\mathbb{E}\left[\mathbb{E}_\sigma\left[\left\|\frac{\sum_{\ell=k}^K \boldsymbol{d}^i_{q,\ell}}{K-k+1}-\frac{\sum_{\ell=k}^K \boldsymbol{d}^i_{q,\ell}}{K-k+2}-\frac{\boldsymbol{d}^i_{q,k-1}}{K-k+2}\right\|^2 \mid \mathcal{F}_{q,k-2}\right]\right]\\
=&\mathbb{E}\left[\mathbb{E}_\sigma\left[\left\|\frac{\sum_{\ell=k}^K \boldsymbol{d}^i_{q,\ell}}{(K-k+1)(K-k+2)}-\frac{\boldsymbol{d}^i_{q,k-1}}{K-k+2}\right\|^2 \mid \mathcal{F}_{q,k-2}\right]\right]\\
\leq&\mathbb{E}\left[\mathbb{E}_\sigma\left[\left(\frac{\sum_{\ell=k}^K \left\|\boldsymbol{d}^i_{q,\ell}\right\|}{(K-k+1)(K-k+2)}+\frac{\left\|\boldsymbol{d}^i_{q,k-1}\right\|}{K-k+2}\right)^2 \mid \mathcal{F}_{q,k-2}\right]\right]\\
\leq&\frac{4V_{\boldsymbol{d}}^2}{(K-k+2)^2}\triangleq\frac{L}{(K-k+2)^2}
\end{aligned}\tag{3.77}$$

More over, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\langle\hat{\boldsymbol{d}}^i_{q,k-1}-\widetilde{\boldsymbol{d}}^i_{q,k},\hat{\boldsymbol{d}}^i_{q,k-1}-\hat{\boldsymbol{d}}^i_{q,k-2}\right\rangle\right]\\
=&\mathbb{E}\left[\mathbb{E}_{\sigma,\widetilde{\boldsymbol{d}}}\left[\left\langle\hat{\boldsymbol{d}}^i_{q,k-1}-\widetilde{\boldsymbol{d}}^i_{q,k},\hat{\boldsymbol{d}}^i_{q,k-1}-\hat{\boldsymbol{d}}^i_{q,k-2}\right\rangle \mid \mathcal{F}_{q,k-1},\mathcal{H}_{q,k-1}\right]\right]\\
=&\mathbb{E}\left[\left\langle\mathbb{E}_{\sigma,\widetilde{\boldsymbol{d}}}\left[\hat{\boldsymbol{d}}^i_{q,k-1}-\widetilde{\boldsymbol{d}}^i_{q,k} \mid \mathcal{F}_{q,k-1},\mathcal{H}_{q,k-1}\right],\hat{\boldsymbol{d}}^i_{q,k-1}-\hat{\boldsymbol{d}}^i_{q,k-2}\right\rangle\right]\\
=&0
\end{aligned}\tag{3.78}$$

since $\mathbb{E}_{\widetilde{\boldsymbol{d}}}\left[\widetilde{\boldsymbol{d}}^i_{q,k} \mid \mathcal{H}_{q,k-1}\right]=\boldsymbol{d}^i_{q,k}$ and $\mathbb{E}_\sigma\left[\boldsymbol{d}^i_{q,k} \mid \mathcal{F}_{q,k-1}\right]=\hat{\boldsymbol{d}}^i_{q,k}$. Using the same argument, we can deduce

$$\begin{aligned}
&\mathbb{E}\left[\left\langle\hat{\boldsymbol{d}}^i_{q,k-1}-\widetilde{\boldsymbol{d}}^i_{q,k},\hat{\boldsymbol{d}}^i_{q,k-2}-\widetilde{\boldsymbol{a}}^i_{q,k-1}\right\rangle\right]\\
=&\mathbb{E}\left[\mathbb{E}_{\sigma,\widetilde{\boldsymbol{d}}}\left[\left\langle\hat{\boldsymbol{d}}^i_{q,k-1}-\widetilde{\boldsymbol{d}}^i_{q,k},\hat{\boldsymbol{d}}^i_{q,k-2}-\widetilde{\boldsymbol{a}}^i_{q,k-1}\right\rangle \mid \mathcal{F}_{q,k-1},\mathcal{H}_{q,k-1},\right]\right]\\
=&\mathbb{E}\left[\left\langle\mathbb{E}_{\sigma,\widetilde{\boldsymbol{d}}}\left[\hat{\boldsymbol{d}}^i_{q,k-1}-\widetilde{\boldsymbol{d}}^i_{q,k} \mid \mathcal{F}_{q,k-1},\mathcal{H}_{q,k-1}\right],\hat{\boldsymbol{d}}^i_{q,k-2}-\widetilde{\boldsymbol{a}}^i_{q,k-1}\right\rangle\right]\\
=&0
\end{aligned}\tag{3.79}$$

where we have use law of total expectation and conditional unbiasness of $\widetilde{\boldsymbol{d}}^i_{q,k}$. Using Young's inequality and equation (3.77), one can write

$$\begin{aligned}
&\mathbb{E}\left[\left\langle\hat{\boldsymbol{d}}^i_{q,k-1}-\hat{\boldsymbol{d}}^i_{q,k-2},\hat{\boldsymbol{d}}^i_{q,k-2}-\widetilde{\boldsymbol{a}}^i_{q,k-1}\right\rangle\right]\\
&\quad\leq\mathbb{E}\left[\frac{1}{2\alpha_k}\left\|\hat{\boldsymbol{d}}^i_{q,k-1}-\hat{\boldsymbol{d}}^i_{q,k-2}\right\|^2+\frac{\alpha_k}{2}\left\|\hat{\boldsymbol{d}}^i_{q,k-2}-\widetilde{\boldsymbol{a}}^i_{q,k-1}\right\|^2\right]\\
&\quad\leq\frac{L}{2\alpha_k(K-k+2)^2}+\frac{\alpha_k}{2}\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-2}-\widetilde{\boldsymbol{a}}^i_{q,k-1}\right\|^2\right]
\end{aligned}\tag{3.80}$$

With the above analysis, we can deduce that

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1}-\widetilde{\boldsymbol{a}}^i_{q,k}\right\|^2\right]\leq&\rho_k^2V+(1-\rho_k)^2\frac{L}{(K-k+2)^2}+(1-\rho_k)^2\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-2}-\widetilde{\boldsymbol{a}}^i_{q,k-1}\right\|^2\right]\\
&+(1-\rho_k)^2\left(\frac{L}{\alpha_k(K-k+2)^2}+\alpha_k\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-2}-\widetilde{\boldsymbol{a}}^i_{q,k-1}\right\|^2\right]\right)
\end{aligned}\tag{3.81}$$

Setting $\alpha_k = \frac{\rho_k}{2}$, we have

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1} - \widetilde{\boldsymbol{a}}^i_{q,k}\right\|^2\right] &\leq \rho_k^2 V + (1-\rho_k)^2\left(1+\frac{2}{\rho_k}\right)\frac{L}{(K-k+2)^2} \\ &\quad + (1-\rho_k)^2\left(1+\frac{\rho_k}{2}\right)\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-2} - \widetilde{\boldsymbol{a}}^i_{q,k-1}\right\|^2\right] \\ &\leq \rho_k^2 V + \left(1+\frac{2}{\rho_k}\right)\frac{L}{(K-k+2)^2} + (1-\rho_k)\,\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-2} - \widetilde{\boldsymbol{a}}^i_{q,k-1}\right\|^2\right]\end{aligned} \tag{3.82}$$

For $k \leq \frac{K}{2}+1$, we set $\rho_k = \frac{2}{(k+3)^{2/3}}$ and recall that $K-k+2 \geq k$, equation (3.82) is written as:

$$\begin{aligned}&\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1} - \widetilde{\boldsymbol{a}}^i_{q,k}\right\|^2\right] \\ &\leq \frac{4}{(k+3)^{4/3}}V + \left(1+(k+3)^{2/3}\right)\frac{L}{k^2} + \left(1-\frac{2}{(k+3)^{2/3}}\right)\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-2} - \widetilde{\boldsymbol{a}}^i_{q,k-1}\right\|^2\right] \\ &\leq \frac{4}{(k+3)^{4/3}}V + \left(1+(k+3)^{2/3}\right)\frac{16L}{(k+3)^2} + \left(1-\frac{2}{(k+3)^{2/3}}\right)\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-2} - \widetilde{\boldsymbol{a}}^i_{q,k-1}\right\|^2\right] \\ &\leq \frac{4}{(k+3)^{4/3}}V + \frac{16L}{(k+3)^{4/3}} + \frac{16L}{(k+3)^{4/3}} + \left(1-\frac{2}{(k+3)^{2/3}}\right)\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-2} - \widetilde{\boldsymbol{a}}^i_{q,k-1}\right\|^2\right] \\ &\leq \frac{4V+32L}{(k+3)^{4/3}} + \left(1-\frac{2}{(k+3)^{2/3}}\right)\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-2} - \widetilde{\boldsymbol{a}}^i_{q,k-1}\right\|^2\right] \\ &\triangleq \frac{M_0}{(k+3)^{4/3}} + \left(1-\frac{2}{(k+3)^{2/3}}\right)\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-2} - \widetilde{\boldsymbol{a}}^i_{q,k-1}\right\|^2\right]\end{aligned} \tag{3.83}$$

We consider the base step where $k=1$,

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,0} - \widetilde{\boldsymbol{a}}^i_{q,1}\right\|^2\right] &= \mathbb{E}\left[\left\|\frac{1}{K}\sum_{\ell=1}^{K}\boldsymbol{d}^i_{q,\ell} - \frac{2}{4^{2/3}}\widetilde{\boldsymbol{d}}^i_{q,1}\right\|^2\right] \\ &\leq \left(V_{\boldsymbol{d}} + \frac{2}{4^{2/3}}\left\|\widetilde{\boldsymbol{d}}^i_{q,1}\right\|\right)^2 \\ &\leq \left(V_{\boldsymbol{d}} + \frac{2}{4^{2/3}}G_0\right)^2 \\ &\triangleq \left(V_{\boldsymbol{d}} + L_0\right)^2\end{aligned} \tag{3.84}$$

Set $M_1 = \max\left\{5^{2/3}\left(V_{\boldsymbol{d}}+L_0\right)^2, M_0\right\}$. For $k \in \left[\frac{K}{2}+1\right]$, we claim that $\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1} - \widetilde{\boldsymbol{a}}^i_{q,k}\right\|^2\right] \leq \frac{M_1}{(k+4)^{2/3}}$. Suppose the claim holds for $k-1$, we have

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-1} - \widetilde{\boldsymbol{a}}^i_{q,k}\right\|^2\right] &\leq \frac{M_1}{(k+3)^{4/3}} + \mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^i_{q,k-2} - \widetilde{\boldsymbol{a}}^i_{q,k-1}\right\|^2\right]\left(1-\frac{2}{(k+3)^{2/3}}\right) \\ &\leq \frac{M_1}{(k+3)^{4/3}} + \frac{M_1}{(k+3)^{2/3}}\cdot\frac{(k+3)^{2/3}-2}{(k+3)^{2/3}} \\ &\leq \frac{M_1\left((k+3)^{2/3}-1\right)}{(k+3)^{4/3}} \\ &\leq \frac{M_1}{(k+4)^{2/3}}\end{aligned} \tag{3.85}$$

since $\frac{(k+3)^{2/3}-1}{k+3)^{4/3}} \leq \frac{1}{(k+4)^{2/3}}$. For $k \in \left[\frac{K}{2}+1, K\right]$, we set $\rho_k = \frac{1.5}{(K-k+2)^{2/3}}$, thus

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k}^i\right\|^2\right] &\leq \frac{2.55V}{(K-k+2)^{4/3}} + \left(1+\frac{4}{3}(K-k+2)^{2/3}\right)\frac{L}{(K-k+2)^2} \\
&\quad + \mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-2}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\|^2\right]\left(1-\frac{1.5}{(K-k+2)^{2/3}}\right) \\
&\leq \frac{2.55V}{(K-k+2)^{4/3}} + \frac{L}{(K-k+2)^{4/3}} + \frac{4}{3}\frac{L}{(K-k+2)^{4/3}} \\
&\quad + \mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-2}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\|^2\right]\left(1-\frac{1.5}{(K-k+2)^{2/3}}\right) \\
&\leq \frac{2.55V+7L/3}{(K-k+2)^{4/3}} + \mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-2}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\|^2\right]\left(1-\frac{1.5}{(K-k+2)^{2/3}}\right) \\
&\triangleq \frac{M_2}{(K-k+2)^{4/3}} + \mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-2}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\|^2\right]\left(1-\frac{1.5}{(K-k+2)^{2/3}}\right) \qquad (3.86)
\end{aligned}$$

Let $M = \max\{M_1, M_2\}$ and $k \in \left[\frac{K}{2}+1, K\right]$, we claim that $\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k}^i\right\|^2\right] \leq \frac{M}{(K-k+1)^{2/3}}$. The base step is verified by equation (3.85). We now suppose the claim holds for $k-1$, let's prove for k.

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k}^i\right\|^2\right] &\leq \frac{M}{(K-k+2)^{4/3}} + \frac{M}{(K-k+2)^{2/3}}\cdot\frac{(K-k+2)^{2/3}-1.5}{(K-k+2)^{2/3}} \\
&= \frac{M\left((K-k+2)^{2/3}-0.5\right)}{(K-k+2)^{4/3}} \\
&\leq \frac{M}{(K-k+1)^{2/3}} \qquad (3.87)
\end{aligned}$$

Thus, from equation (3.85) and equation (3.87), we have

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k}^i\right\|^2\right] \leq \begin{cases} \frac{M}{(k+4)^{2/3}} & k \in \left[1, \frac{K}{2}\right] \\ \frac{M}{(K-k+1)^{2/3}} & k \in \left[\frac{K}{2}+1, K\right] \end{cases} \qquad (3.88)$$

Thus, using Jensen inequality, we have

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k}^i\right\|\right] &\leq \sqrt{\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k}^i\right\|^2\right]} \\
&= \sqrt{\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k}^i\right\|^2\right]} \\
&\leq \begin{cases} \frac{\sqrt{M}}{(k+4)^{1/3}} & k \in \left[1, \frac{K}{2}\right] \\ \frac{\sqrt{M}}{(K-k+1)^{1/3}} & k \in \left[\frac{K}{2}+1, K\right] \end{cases} \qquad (3.89)
\end{aligned}$$

□

Lemma 3.7.5 (Lemma 3.3.5)**.** *For F_t a monotone continous DR-submodular and β-smoothness, $\boldsymbol{x}_{t,k+1} = \boldsymbol{x}_{t,k} + \frac{1}{K}\boldsymbol{v}_{t,k}$ for $k \in [K]$, we*

$$\begin{aligned}
F_t(\boldsymbol{x}^*) - F_t(\boldsymbol{x}_{t,k+1}) &\leq (1-1/K)\left[F_t(\boldsymbol{x}^*) - F_t(\boldsymbol{x}_{t,k})\right] \qquad (3.90) \\
&\quad - \frac{1}{K}\left[-\left\|\nabla F_t(\boldsymbol{x}_{t,k}) - \boldsymbol{d}_{t,k}\right\| D + \langle \boldsymbol{d}_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^*\rangle\right] + \frac{\beta D^2}{2K^2}
\end{aligned}$$

Proof. Followng the idea of [Chen2018a] and using β-smoothness of F_t,

$$\begin{aligned}
F_t(\boldsymbol{x}_{t,k+1}) &\geq F_t(\boldsymbol{x}_{t,k}) + \langle F_t(\boldsymbol{x}_{t,k}), \boldsymbol{x}_{t,k+1} - \boldsymbol{x}_{t,k}\rangle - \frac{\beta}{2}\left\|\boldsymbol{x}_{t,k+1} - \boldsymbol{x}_{t,k}\right\|^2 \\
&\geq F_t(\boldsymbol{x}_{t,k}) + \frac{1}{K}\left\langle F_t(\boldsymbol{x}_{t,k}), \boldsymbol{v}_{t,k}^i\right\rangle - \frac{\beta}{2}\frac{D^2}{K^2} && \text{(since } \|\boldsymbol{v}_{t,k}\| \leq D\text{)} \\
&\geq F_t(\boldsymbol{x}_{t,k}) + \frac{1}{K}\left[\langle \nabla F_t(\boldsymbol{x}_{t,k}) - \boldsymbol{d}_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^*\rangle + \langle \nabla F_t(\boldsymbol{x}_{t,k}), \boldsymbol{x}^*\rangle + \langle \boldsymbol{d}_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^*\rangle\right] - \frac{\beta}{2}\frac{D^2}{K^2}
\end{aligned} \tag{3.91}$$

By Cauchy-Schwarz's inequality, note that,

$$\langle \nabla F_t(\boldsymbol{x}_{t,k}) - \boldsymbol{d}_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^*\rangle \geq -\left\|\nabla F_t(\boldsymbol{x}_{t,k}) - \boldsymbol{t}_{t,k}\right\| D$$

Using concavity along non-negative direction and monotonicity of F_t, we have,

$$\begin{aligned}
F_t(\boldsymbol{x}^*) - F_t(\boldsymbol{x}_{t,k}) &\leq F_t(\boldsymbol{x}^* \vee \boldsymbol{x}_{t,k}) - F_t(\boldsymbol{x}_{t,k}) \\
&\leq \langle \nabla F_t(\boldsymbol{x}_{t,k}), (\boldsymbol{x}^* \vee \boldsymbol{x}_{t,k}) - \boldsymbol{x}_{t,k}\rangle \\
&= \langle \nabla F_t(\boldsymbol{x}_{t,k}), (\boldsymbol{x}^* - \boldsymbol{x}_{t,k}) \vee 0\rangle \\
&\leq \langle \nabla F_t(\boldsymbol{x}_{t,k}), \boldsymbol{x}^*\rangle
\end{aligned} \tag{3.92}$$

then, equation (3.91) becomes

$$\begin{aligned}
F_t(\boldsymbol{x}_{t,k+1}) &\geq F_t(\boldsymbol{x}_{t,k}) + \langle F_t(\boldsymbol{x}_{t,k}), \boldsymbol{x}_{t,k+1} - \boldsymbol{x}_{t,k}\rangle - \frac{\beta}{2}\left\|\boldsymbol{x}_{t,k+1} - \boldsymbol{x}_{t,k}\right\|^2 \\
&\geq F_t(\boldsymbol{x}_{t,k}) + \frac{1}{K}\left[-\left\|\nabla F_t(\boldsymbol{x}_{t,k}) - \boldsymbol{t}_{t,k}\right\| D + F_t(\boldsymbol{x}^*) - F_t(\boldsymbol{x}_{t,k}) + \langle \boldsymbol{d}_{t,k}, \boldsymbol{v}_{t,k} - \boldsymbol{x}^*\rangle\right] - \frac{\beta}{2}\frac{D^2}{K^2}
\end{aligned} \tag{3.93}$$

Adding and substracting $F_t(\boldsymbol{x}^*)$ and multiply both side by -1 yields Lemma 3.7.5. □

3.7.2 Section 3.4 : Bandit Setting

Lemma 3.7.6 (Lemma 3.4.4)**.** *Under Assumption 3.2.2, the variance of the local gradient estimate is uniformly bounded, i.e*

$$\mathbb{E}\left[\left\|\boldsymbol{d}_{q,k}^{i,\delta} - \widetilde{\boldsymbol{d}}_{q,k}^{i,\delta}\right\|^2\right] \leq 4n\left(\frac{d}{\delta}B\right)^2\left[\frac{1}{\left(\frac{1}{\lambda(\mathbf{W})} - 1\right)^2} + 2\right] \tag{3.94}$$

Proof. By Assumption 3.2.2, we have

$$\mathbb{E}\left[\left\|\nabla f_{\sigma_q(\tau)}^{cat} - \widetilde{\boldsymbol{h}}_{q,\tau}^{cat}\right\|^2\right] = \mathbb{E}\left[\sum_{i=1}^{n}\left\|\nabla f_{\sigma_q(\tau)}^{i,\delta}\left(\boldsymbol{x}_{q,\tau}^i\right) - \widetilde{\boldsymbol{h}}_{q,\tau}^i\right\|^2\right] \leq n\left(\frac{d}{\delta}B\right)^2 \tag{3.95}$$

Following the same analysis in equation (4.14), we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\boldsymbol{d}_{q,k}^{cat} - \widetilde{\boldsymbol{d}}_{q,k}^{cat}\right\|^2\right] \\
\leq&\mathbb{E}\left[\left(\sum_{\tau=1}^{k-1}\left\|\mathbf{W}^{k-\tau} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\|\left\|\nabla f_{\sigma_q(\tau+1)}^{cat} - \widetilde{\boldsymbol{h}}_{q,\tau+1}^{cat} + \widetilde{\boldsymbol{h}}_{q,\tau}^{cat} - \nabla f_{\sigma_q(\tau)}^{cat}\right\|\right)^2\right] \\
&+ 4\left(\mathbb{E}\left[\left\|\mathbf{W}^k - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\|^2\left\|\nabla f_{\sigma_q(1)}^{cat} - \widetilde{\boldsymbol{h}}_{q,1}^{cat}\right\|^2\right] + \mathbb{E}\left[\left\|\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\|^2\left\|\nabla f_{\sigma_q(k)}^{cat} - \widetilde{\boldsymbol{h}}_{q,k}^{cat}\right\|^2\right]\right) \\
\leq&4n\left(\frac{d}{\delta}B\right)^2\left(\sum_{\tau=1}^{k-1}\lambda(\mathbf{W})^{k-\tau}\right)^2 + 4n\left(\frac{d}{\delta}B\right)^2\left(\lambda(\mathbf{W})^{2k} + 1\right) \\
\leq&4n\left(\frac{d}{\delta}B\right)^2\left(\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})}\right)^2 + 4n\left(\frac{d}{\delta}B\right)^2(\lambda(\mathbf{W}) + 1) \leq 4n\left(\frac{d}{\delta}B\right)^2\left[\frac{1}{\left(\frac{1}{\lambda(\mathbf{W})} - 1\right)^2} + 2\right]
\end{aligned} \tag{3.96}$$

The lemma follows by remarking that $\mathbb{E}\left[\left\|\boldsymbol{d}_{q,k}^{i,\delta} - \widetilde{\boldsymbol{d}}_{q,k}^{i}\right\|^2\right] \leq \mathbb{E}\left[\left\|\boldsymbol{d}_{q,k}^{cat} - \widetilde{\boldsymbol{d}}_{q,k}^{cat}\right\|^2\right]$ □

Lemma 3.7.7 (Lemma 3.4.5)**.** *Let $N = k_0 \cdot nB\frac{d}{\delta}\max\left\{\lambda(\mathbf{W})\left(1+\frac{2}{1-\lambda(\mathbf{W})}\right), 2\right\}$. Under Assumptions 1.2.3 and 3.2.2, for $k \in [K]$, we have*

$$\max_{i\in[1,n]} \mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k}^{i,\delta} - \nabla\hat{F}_{q,k}^{\delta}\right\|\right] \leq \frac{N}{k} \tag{3.97}$$

Proof. The proof is essentially based on the one of Lemma 3.7.1. Note that we keep the same notation with a superscipt δ to indicate the smooth version of f and related variables. By definition of the one-point gradient estimator and Assumption 3.2.2, equation (3.52) becomes

$$\begin{aligned}
\mathbb{E}\left[\left\|\delta_{q,k}^{cat,\delta}\right\|^2\right] &= \mathbb{E}\left[\sum_{i=1}^{n}\left\|\delta_{q,k}^{i,\delta}\right\|^2\right] = \sum_{i=1}^{n}\mathbb{E}\left[\left\|\nabla\hat{f}_{q,k}^{i,\delta} - \nabla\hat{f}_{q,k-1}^{i,\delta}\right\|^2\right] \\
&= \sum_{i=1}^{n}\mathbb{E}\left[\mathbb{E}\left[\left\|\nabla\hat{f}_{q,k}^{i,\delta} - \nabla\hat{f}_{q,k-1}^{i,\delta}\right\|^2 \mid \mathcal{F}_{q,k-1}\right]\right] \\
&= \sum_{i=1}^{n}\mathbb{E}\left[\mathbb{E}\left[\left\|\frac{\sum_{\ell=k+1}^{L}\nabla f_{\sigma_q(\ell)}^{i,\delta}(\boldsymbol{x}_{q,\ell}^{i})}{L-k} - \frac{\sum_{\ell=k}^{L}\nabla f_{\sigma_q(\ell)}^{i,\delta}(\boldsymbol{x}_{q,\ell}^{i})}{L-k+1}\right\|^2 \mid \mathcal{F}_{q,k-1}\right]\right] \\
&= \sum_{i=1}^{n}\mathbb{E}\left[\mathbb{E}\left[\left\|\frac{\sum_{\ell=k+1}^{L}\nabla f_{\sigma_q(\ell)}^{i,\delta}(\boldsymbol{x}_{q,\ell}^{i})}{(L-k)(L-k+1)} - \frac{\nabla f_{\sigma_q(k)}^{i,\delta}(\boldsymbol{x}_{q,k}^{i})}{L-k+1}\right\|^2 \mid \mathcal{F}_{q,k-1}\right]\right] \\
&\leq n\left(\frac{2B\frac{d}{\delta}}{L-k+1}\right)^2
\end{aligned} \tag{3.98}$$

By Jensen's inequality, we deduce that

$$\mathbb{E}\left[\left\|\delta_{q,k}^{cat,\delta}\right\|\right] \leq \sqrt{\mathbb{E}\left[\left\|\delta_{q,k}^{cat,\delta}\right\|^2\right]} \leq \frac{2\sqrt{n}B\frac{d}{\delta}}{L-k+1} \tag{3.99}$$

When $k = 1$, following the same derivation in equation (3.54), we have

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,1}^{cat,\delta} - \nabla\hat{F}_{q,1}^{cat,\delta}\right\|^2\right] \leq \lambda(\mathbf{W})^2\mathbb{E}\left[\sum_{i=1}^{n}\left\|\hat{\boldsymbol{g}}_{q,1}^{i,\delta} - \nabla\hat{F}_{q,1}^{\delta}\right\|^2\right] \leq n\lambda(\mathbf{W})^2\frac{d^2}{\delta^2}B^2$$

Let $k \in [2, k_0]$, from equation (2.32) and equation (3.99)

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k}^{cat,\delta} - \nabla\hat{F}_{q,k}^{cat,\delta}\right\|\right] &\leq \lambda(\mathbf{W})\left(\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{cat,\delta} - \nabla\hat{F}_{q,k-1}^{cat,\delta}\right\|\right] + \mathbb{E}\left[\left\|\delta_{q,k}^{cat,\delta}\right\|\right]\right) \\
&\leq \lambda(\mathbf{W})^{k-1}\sqrt{n}\frac{d}{\delta}B + 2\sum_{\tau=1}^{k}\lambda(\mathbf{W})^{\tau}\sqrt{n}\frac{d}{\delta}B \\
&\leq \lambda(\mathbf{W})\sqrt{n}\frac{d}{\delta}B + 2\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})}\sqrt{n}\frac{d}{\delta}B \\
&= \lambda(\mathbf{W})\sqrt{n}\frac{d}{\delta}B\left(1+\frac{2}{1-\lambda(\mathbf{W})}\right)
\end{aligned} \tag{3.100}$$

Let $N_0 = k_0 \cdot \sqrt{n}\max\left\{\lambda(\mathbf{W})B\frac{d}{\delta}\left(1+\frac{2}{1-\lambda(\mathbf{W})}\right), 2B\frac{d}{\delta}\right\}$. We claim that $\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k}^{cat,\delta} - \nabla\hat{F}_{q,k}^{cat,\delta}\right\|\right] \leq \frac{N_0}{k}$ when $k \in [k_0, K]$. Let $L \geq 2K$, we have then $\frac{1}{L-k+1} \leq \frac{1}{2K-k+1} \leq \frac{1}{K+1} \leq \frac{1}{k+1}$. Thus, using

the induction hypothesis, we have

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k}^{cat,\delta}-\nabla\hat{F}_{q,k}^{cat,\delta}\right\|\right] &\leq \lambda(\mathbf{W})\left(\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{cat,\delta}-\nabla\hat{F}_{q,k-1}^{cat,\delta}\right\|\right]+\mathbb{E}\left[\left\|\delta_{q,k}^{cat,\delta}\right\|\right]\right)\\ &\leq \lambda(\mathbf{W})\left(\frac{N_0}{k-1}+\frac{2\sqrt{n}B\frac{d}{\delta}}{L-k+1}\right)\\ &\leq \lambda(\mathbf{W})\left(\frac{N_0}{k-1}+\frac{2\sqrt{n}B\frac{d}{\delta}}{k+1}\right)\\ &\leq \lambda(\mathbf{W})\left(N_0\frac{k_0+1}{k_0(k-1)}\right)\\ &\leq \frac{N_0}{k}\end{aligned} \tag{3.101}$$

Using the inequality in equation (3.58) and the above result, the lemma is then proven. □

Lemma 3.7.8 (Lemma 3.4.6)**.** *Under Lemma 3.4.3 and lemma 3.4.4 and setting* $\rho_k=\frac{2}{(k+3)^{2/3}}$*, we have*

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta}-\widetilde{\boldsymbol{a}}_{q,k}^{i}\right\|\right]\leq\frac{\sqrt{M_0}}{(k+3)^{1/3}},\qquad k\in[K] \tag{3.102}$$

where $M_0=4^{2/3}\frac{d^2}{\delta^2}B^2\left[24n^2\left(\frac{1}{\frac{1}{\lambda(\mathbf{W})}-1}+1\right)^2+8n\left(\frac{1}{\left(\frac{1}{\lambda(\mathbf{W})}-1\right)^2}+2\right)\right]$

Proof. The proof follows the same idea in Lemma 10 and Lemma 11 of [Zhang2019] with different constants. We will evoques in details in the following section. Following the same decomposition in the proof of Lemma 3.7.4, we have

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta}-\widetilde{\boldsymbol{a}}_{q,k}^{i}\right\|^2\right] &= \mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta}-(1-\rho_k)\widetilde{\boldsymbol{a}}_{q,k-1}^{i}-\rho_k\widetilde{\boldsymbol{d}}_{q,k}^{i,\delta}\right\|^2\right]\\ &= \rho_k^2\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta}-\widetilde{\boldsymbol{d}}_{q,k}^{i,\delta})\right\|^2\right]+(1-\rho_k)^2\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta}-\hat{\boldsymbol{d}}_{q,k-2}^{i,\delta}\right\|^2\right]\\ &\quad+(1-\rho_k)^2\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-2}^{i,\delta}-\widetilde{\boldsymbol{a}}_{q,k-1}^{i}\right\|^2\right]\\ &\quad+2\rho_k(1-\rho_k)\mathbb{E}\left[\left\langle\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta}-\widetilde{\boldsymbol{d}}_{q,k}^{i,\delta},\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta}-\hat{\boldsymbol{d}}_{q,k-2}^{i,\delta}\right\rangle\right]\\ &\quad+2\rho_k(1-\rho_k)\mathbb{E}\left[\left\langle\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta}-\widetilde{\boldsymbol{d}}_{q,k}^{i,\delta},\hat{\boldsymbol{d}}_{q,k-2}^{i,\delta}-\widetilde{\boldsymbol{a}}_{q,k-1}^{i}\right\rangle\right]\\ &\quad+2(1-\rho_k)^2\mathbb{E}\left[\left\langle\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta}-\hat{\boldsymbol{d}}_{q,k-2}^{i,\delta},\hat{\boldsymbol{d}}_{q,k-2}^{i,\delta}-\widetilde{\boldsymbol{a}}_{q,k-1}^{i}\right\rangle\right]\end{aligned}$$

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta}-\widetilde{\boldsymbol{d}}_{q,k}^{i,\delta})\right\|^2\right] &= \mathbb{E}\left[\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta}-\widetilde{\boldsymbol{d}}_{q,k}^{i,\delta})\right\|^2\mid\mathcal{F}_{q,k-1}\right]\right]\\ &\leq \mathbb{E}\left[\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta}-\boldsymbol{d}_{q,k}^{i,\delta}\right\|^2+\left\|\boldsymbol{d}_{q,k}^{i,\delta}-\widetilde{\boldsymbol{d}}_{q,k}^{i,\delta})\right\|^2+2\langle\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta}-\boldsymbol{d}_{q,k}^{i,\delta},\boldsymbol{d}_{q,k}^{i,\delta}-\widetilde{\boldsymbol{d}}_{q,k}^{i,\delta}\rangle\mid\mathcal{F}_{q,k-1}\right]\right]\end{aligned} \tag{3.103}$$

By the definition in equation (3.35), we have $\mathbb{E}\left[\boldsymbol{d}_{q,k}^{i,\delta}|\mathcal{F}_{q,k-1}\right]=\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta}$, using Lemma 3.4.3, we have

$$\mathbb{E}\left[\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta}-\boldsymbol{d}_{q,k}^{i,\delta}\right\|^2\mid\mathcal{F}_{q,k-1}\right]\right]\leq(V_{\boldsymbol{d}}^{\delta})^2 \tag{3.104}$$

Invoking Lemma 3.4.4, we have

$$\mathbb{E}\left[\left\|\boldsymbol{d}_{q,k}^{i,\delta}-\widetilde{\boldsymbol{d}}_{q,k}^{i,\delta})\right\|^2\right]\leq\sigma_2^2 \tag{3.105}$$

and

$$\mathbb{E}\left[\mathbb{E}\left[\left\langle \hat{\boldsymbol{d}}_{q,k-1}^{i,\delta} - \boldsymbol{d}_{q,k}^{i,\delta}, \boldsymbol{d}_{q,k}^{i,\delta} - \widetilde{\boldsymbol{d}}_{q,k}^{i,\delta}\right\rangle \mid \mathcal{F}_{q,k-1}\right]\right] = 0 \tag{3.106}$$

by following the same analysis in equation (3.75). We now claim that equation (3.103) is bounded above by

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta} - \widetilde{\boldsymbol{d}}_{q,k}^{i,\delta})\right\|^2\right] \leq (V_{\boldsymbol{d}}^{\delta})^2 + \sigma_2^2 \triangleq V^{\delta} \tag{3.107}$$

More over, taking the idea from equations (3.77) to (3.80), we have

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta} - \hat{\boldsymbol{d}}_{q,k-2}^{i,\delta}\right\|^2\right] \leq \frac{4(V_{\boldsymbol{d}}^{\delta})^2}{(L-k+2)^2} \triangleq \frac{L^{\delta}}{(L-k+2)^2} \tag{3.108}$$

$$\mathbb{E}\left[\left\langle \hat{\boldsymbol{d}}_{q,k-1}^{i,\delta} - \widetilde{\boldsymbol{d}}_{q,k}^{i,\delta}, \hat{\boldsymbol{d}}_{q,k-1}^{i,\delta} - \hat{\boldsymbol{d}}_{q,k-2}^{i,\delta}\right\rangle\right] = 0 \tag{3.109}$$

$$\mathbb{E}\left[\left\langle \hat{\boldsymbol{d}}_{q,k-1}^{i,\delta} - \widetilde{\boldsymbol{d}}_{q,k}^{i,\delta}, \hat{\boldsymbol{d}}_{q,k-2}^{i,\delta} - \widetilde{\boldsymbol{a}}_{q,k-1}^{i}\right\rangle\right] = 0 \tag{3.110}$$

and

$$\mathbb{E}\left[\left\langle \hat{\boldsymbol{d}}_{q,k-1}^{i,\delta} - \hat{\boldsymbol{d}}_{q,k-2}^{i,\delta}, \hat{\boldsymbol{d}}_{q,k-2}^{i,\delta} - \widetilde{\boldsymbol{a}}_{q,k-1}^{i}\right\rangle\right] \leq \frac{L^{\delta}}{2\alpha_k(L-k+2)^2} + \frac{\alpha_k}{2}\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-2}^{i,\delta} - \widetilde{\boldsymbol{a}}_{q,k-1}^{i}\right\|^2\right] \tag{3.111}$$

by using Young's inequality. Setting $\alpha_k = \frac{\rho_k}{2}$ similarly to Lemma 3.3.4, we have

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta} - \widetilde{\boldsymbol{a}}_{q,k}^{i}\right\|^2\right] \leq \rho_k^2 V^{\delta} + \left(1 + \frac{2}{\rho_k}\right)\frac{L^{\delta}}{(L-k+2)^2} + (1-\rho_k)\,\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-2}^{i,\delta} - \widetilde{\boldsymbol{a}}_{q,k-1}^{i}\right\|^2\right] \tag{3.112}$$

Setting $L \geq 2K$ and $\rho_k = \frac{2}{(k+2)^{2/3}}$, we have then $\frac{1}{L-k+2} \leq \frac{1}{2K-k+2} \leq \frac{1}{K+2} \leq \frac{1}{k+2}$. Following the derivation from Lemma 11 of [Zhang2019].Equation (3.112) can be bounded above by

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta} - \widetilde{\boldsymbol{a}}_{q,k}^{i}\right\|^2\right] &\leq \rho_k^2 V^{\delta} + \left(1 + \frac{2}{\rho_k}\right)\frac{L^{\delta}}{(k+2)^2} + (1-\rho_k)\,\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-2}^{i,\delta} - \widetilde{\boldsymbol{a}}_{q,k-1}^{i}\right\|^2\right] \\
&\leq \frac{4^{2/3}\left(2V^{\delta} + L^{\delta}\right)}{(k+2)^{4/3}} + \left(1 - \frac{2}{(k+2)^{2/3}}\right)\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-2}^{i,\delta} - \widetilde{\boldsymbol{a}}_{q,k-1}^{i}\right\|^2\right] \\
&\triangleq \frac{M_0}{(k+2)^{4/3}} + \left(1 - \frac{2}{(k+2)^{2/3}}\right)\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-2}^{i,\delta} - \widetilde{\boldsymbol{a}}_{q,k-1}^{i}\right\|^2\right]
\end{aligned} \tag{3.113}$$

Assume that $\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,k-1}^{i,\delta} - \widetilde{\boldsymbol{a}}_{q,k}^{i}\right\|^2\right] \leq \frac{M_0}{(k+3)^{2/3}}$ for $k \in [K]$. When $k = 1$, by definition of $\widetilde{\boldsymbol{a}}_{q,1}^{i}$ and $\hat{\boldsymbol{d}}_{q,0}^{i,\delta}$, we have

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,0}^{i,\delta} - \widetilde{\boldsymbol{a}}_{q,1}^{i}\right\|^2\right] \leq \left(V_{\boldsymbol{d}}^{\delta} + \frac{2}{3^{2/3}}\frac{d}{\delta}B\right)^2 \tag{3.114}$$

Thus, since $\sigma_2 \geq \frac{2}{3^{2/3}}\frac{d}{\delta}B$, one can observe that

$$\frac{M_0}{(1+2)^{2/3}} = 2V^{\delta} + L^{\delta} \geq 2V^{\delta} = 2\left((V_{\boldsymbol{d}}^{\delta})^2 + \sigma_2^2\right) \geq \left(V_{\boldsymbol{d}}^{\delta}) + \sigma_2\right)^2 \geq \mathbb{E}\left[\left\|\hat{\boldsymbol{d}}_{q,0}^{i,\delta} - \widetilde{\boldsymbol{a}}_{q,1}^{i}\right\|^2\right] \tag{3.115}$$

Suppose that the induction hypothesis holds for $k-1$, one can easily verify for k since

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^{i,\delta}_{q,k-1} - \widetilde{\boldsymbol{a}}^{i}_{q,k}\right\|^2\right] &\leq \frac{M_0}{(k+2)^{4/3}} + \left(1 - \frac{2}{(k+2)^{2/3}}\right)\mathbb{E}\left[\left\|\hat{\boldsymbol{d}}^{i,\delta}_{q,k-2} - \widetilde{\boldsymbol{a}}^{i}_{q,k-1}\right\|^2\right] \\
&\leq \frac{M_0}{(k+2)^{4/3}} + \left(1 - \frac{2}{(k+2)^{2/3}}\right)\frac{M_0}{(k+2)^{2/3}} \\
&\leq M_0 \frac{(k+2)^{2/3} - 1}{(k+3)^{4/3}} \\
&\leq \frac{M_0}{(k+3)^{2/3}} \qquad (3.116)
\end{aligned}$$

□

4

Distributed Online Algorithm for Non-Convex Optimization

Contents

4.1 Introduction

The popularity of sensors and IoT devices has the potential to generate and, equivalently, accumulate data in order of Zeta bytes [MacCarthy2018] annually. High throughput, low latency, data consumption, and network dependencies are often the key metrics in designing high-performance learning algorithms under the constraint of low-power computing. In recent times, there has been an alternate trend to process data in the cloud or dump into a centralized database. Commonly known as edge computing, the new paradigm embraces the idea of using interconnected computing nodes to reduce high bandwidth-consuming data uploads, privacy preservation of data, and knowledge on the fly. Smart building applications typically have a profound implication on the environment in terms of energy savings, reduction of green house emissions, etc. Predicting the future often forms the basis of corrective actions taken by such apps and can be regarded as a predominant use case of machine learning. Typically, data is generated in various zones by heterogeneous sensors, creating a distributed learning environment. Recently, improvements in network communication and edge computing have enhanced the hardware-software interface. Consequently, the practical option of implementing a machine learning model on-site and analyzing data in real time has emerged as an alternative to transmitting data to a centralized database. Optimizing problems to maintain robust solutions under the uncertainty of the future is a nice feature for such cyber-physical systems. Contrary to the classical train-test-deploy framework, online learning offers continuous learning where, during run-time, a batch of sensor data has the potential to update an AI model on site. This work aligns with the edge computing paradigm by proposing a distributed and online learning algorithm. Online learning helps to better adapt to the uncertainty of the future, where the data pattern continually changes over time. The designed algorithm repeatedly chooses a high-performance strategy given a set of actions compared to the best-fixed action in hindsight. Instead of having a centralized mediator, the distributed environment promotes peer-to-peer knowledge exchanges while prohibiting data sharing between learners. Many proposed distributed online algorithms use gradient descent-based methods to solve constraint problems. Such an approach requires projection into the constraint set, which usually involves intensive computation, which is not best suited in the context of sensors and IoT. We aim to design a competitive, robust algorithm in the distributed and online setting that has the flexibility of being projection-free.

Problem setting We are given a convex set $\mathcal{K} \subseteq \mathbb{R}^d$ and a set of agents connected over a network represented by a graph $G = (V, E)$ where $n = |V|$ is the number of agents. At every time $t \in [T]$, each agent $i \in V$ can communicate with (and only with) its immediate neighbors, ie, adjacent agents in G and makes a decision $\boldsymbol{x}_i^t \in \mathcal{K}$. Subsequently, a batch of new data is revealed exclusively to the agent i and, from its own batch, a non-convex cost function $f_t^i : \mathcal{K} \to \mathbb{R}$ is induced locally. Although each agent i observes only a function f_t^i, agent i is interested in the cumulative cost $F_t(\,\cdot\,)$ where $F_t(\,\cdot\,) := \frac{1}{n}\sum_{i=1}^n f_t^i(\,\cdot\,)$. In particular, at time t, the cost of agent i with its chosen $\boldsymbol{x}_t^i$ is $F_t(\boldsymbol{x}_t^i)$. The objective of each agent i is to minimize the total cumulative cost $\sum_{t=1}^T F_t(\boldsymbol{x}_t^i)$ through local communication with its immediate neighbors.

In the online convex optimization setting, a standard measure of performance is the *regret* (or the normalized version) which compares the total cost of every agent to that of the best solution in hindsight, that is, for all $i \in [n]$,

$$\mathcal{R}_T = \frac{1}{T}\left(\sum_{t=1}^T F_t(\boldsymbol{x}_t^i) - \min_{\boldsymbol{x}\in\mathcal{K}} \sum_{t=1}^T F_t(\boldsymbol{x})\right)$$

However, the concept of regret is ill-defined in the for non-convex objectives, as there is no guarantee of a global minimum's existence. This makes it challenging to evaluate the performance of agents against an optimal solution. In such cases, an alternative performance measure in the offline setting is the distance to stationary points, which often provides a sufficient condition for algorithm convergence to a local minimum. In the setting Frank-Wolfe algorithm, a natural stationarity measure for non-convex settings is the duality gap [Jaggi2013, Lacoste-Julien2016], defined as :

$$\mathcal{G}_t(\boldsymbol{x}) = \max_{\boldsymbol{v}\in\mathcal{K}} \langle \nabla F_t(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{v}\rangle$$

It is important to note that $\mathcal{G}_t(\boldsymbol{x})$ is non-negative, and $\mathcal{G}_t(\boldsymbol{x}) = 0$ if and only if $\boldsymbol{x}$ is a stationary point of the function F_t. We extend this duality gap to the online setting by defining the convergence

gap as:

$$\max_{\boldsymbol{v}\in\mathcal{K}} \frac{1}{T}\sum_{t=1}^{T} \langle \nabla F_t(\boldsymbol{x}_t), \boldsymbol{x}_t - \boldsymbol{v}\rangle$$

In the same spirit as regret, the convergence measures the tocal cost of agents to that of the best stationary point in hindsight. When the function F_t are convex, the convergence gap is always upper bounded by the regret. Moreover, when the problem becomes offline, that is, all F_t are the same, the convergence gap measures the speed of convergence to a stationary solution.

4.1.1 Our contribution

The challenge in designing robust and efficient algorithms for the problem is to resolve the following issues together: the uncertainty (online setting, agents observe their own loss functions only after choosing their decisions), the partial information (distributed setting, agents know only its own loss functions while aiming to minimize the cumulating cost), and the nonconvexity of the loss functions. As a starting point, we consider the Meta Frank-Wolfe (MFW) algorithm [Chen2018b] in the (centralized, convex) online setting and the distributed Frank-Wolfe (DFW) algorithm [Wai2017] in the distributed (offline) setting. However, these algorithms work either in the online setting or in the distributed one but not both together. The difficulty in our problem, as mentioned earlier, is to resolve all issues together.

In the paper, we present algorithms, subtly built on MFW and DFW algorithms, that achieve the convergence gap of $O(T^{-1/2})$ and $O(T^{-1/4})$ in cases where the exact gradients or only stochastic gradients of loss functions are available, respectively. Note that in the former, the convergence gap of $O(T^{-1/2})$ asymptotically matches the best-regret guarantee even in the centralized offline settings with convex functions. In addition, one can convert the algorithms to be projection-free by choosing the appropriate oracles used in the algorithm. This property provides flexibility to apply the algorithms to different settings depending on the computing capacity of local devices. Our work applies to online neural network optimization amongst a group of autonomous learners. We demonstrate the practical utility of our algorithm in a smart building application where zones mimic learners optimizing a temperature forecasting problem. We provide a thorough analysis of our algorithms in different angles of the performance guarantee (quality of solutions), the effects of network topology, and decentralization, which are predictably explained by our theoretical results.

4.1.2 Related Work

Distributed Online Optimization. Authors [Yan2013] introduced decentralized online projected subgradient descent and showed vanishing regret for convex and strongly convex functions. In contrast, Hosseini et al. [Hosseini2013] extended distributed dual averaging technique to the online setting using a general regularized projection for both unconstrained and constrained optimization. A distributed variant of online conditional gradient [Hazan2016a] was designed and analyzed in [Zhang2017] that requires linear minimizers and uses exact gradients. However, computing exact gradients may be prohibitively expensive for moderately sized data and intractable when a closed-form does not exist. In this work, we go a step ahead in designing a distributed algorithm that uses stochastic gradient estimates and provides a better regret bound than in [Zhang2017].

Learning on the edge. Over the year, edge computing has become an exciting alternative for cloud-based learning by processing the data closer to end devices while ensuring data confidentiality and reducing transmission. [Wang2018] proposes a distributed framework for non-i.i.d data using multiple gradient descent-based algorithms to update local models and a dedicated edge unit for global aggregation. Another popular approach is to reduce the memory size of classical machine learning models to meet edge resource constraints. [Shotton2013] and [Nan2016] similarly takes this idea by building a tree-based learning framework with a considerable reduction in memory using compression and pruning. At the same time, [Gupta2017] introduce an edge-friendly version of k-nearest neighbor [Cover1967] by projecting the data into a lower-dimensional space. Besides traditional machine learning algorithms, adapting deep learning models to work on edge devices is an emerging research domain. In [Chiliang2019, Lin2017], the authors propose a pruning technique on convolutional network for faster computation while preserving the model ability. Another

approach using weight quantization is proposed in [Simons2019]. The current dominant paradigm is federated learning [McMahan2017, Kairouz2021], where offline centralized training is performed through a star network with multiple devices connected to a central server. However, decentralized training is more efficient than centralized one when operating on networks with low bandwidth or high latency [Lian2017, He2018]. In this paper, we go one step further by studying arbitrary communication networks without a central coordinator and the local data (so local cost functions) evolve.

Thermal Profiling a Building. Usually, building monitoring sensors are distributed across a building and thus acts as a scattered data lake with potentially heterogeneous patterns. Indoor temperature is an important factor in controlling Heating Ventilation Air Conditioning systems that maintain ambient comfort within a building [Gupta2015]. Typically such embedded systems run in anticipatory mode where temperature prediction [Cai2019] of controlled building zones helps in maintaining thermal consistency. A multitude of factors effect the thermal profile like outdoor environment, opening/closing of windows, number of occupants, etc, which are hard to get and often rely on intrusive mechanisms to gather the data. Researchers have utilized deep learning models [Zamora-Martinez2014] in the context of online learning of temperature, but lack the benefit of interacting with multiple similar sensors. This study seeks to generate a thermal profile of a building by only utilizing temperature data from multiple zones of a building in order to extract patterns about thermal variation. The proposed methodology not only processes data on the fly [Abdel-Aziz2019], but also identifies meaningful topological data exchange networks that can best predict multi zonal temperature settings.

4.2 Preliminaries

We recall the notations and concepts defined in Section 1.2. We denote the convex set by $\mathcal{K}$ and the decision vector of agent i at time step k of phase t by $\boldsymbol{x}_{t,k}^i$. We let $\mathbb{G}$ be an undirected graph with adjacency matrix $\mathbf{W} \in \mathbb{R}_+^{n\times n}$, where $n = |\mathcal{V}|$ is the number of agents. We assume that the matrix $\mathbf{W}$ is doubly stochastic, meaning that $\mathbf{W1} = \mathbf{W}^T\mathbf{1} = \mathbf{1}$. Boldface letters, such as $\boldsymbol{x}$, represent vectors. We denote the decision vector of agent i at time step t by $\boldsymbol{x}_t^i$. We assume that the constraint set $\mathcal{K}$ satisfies Assumption 1.2.1 and the functions f_t^i satisfy Assumption 1.2.2 with constants β, G. The stochastic estimates $\widetilde{f}_t^i$ satisfy Assumption 3.2.1 with constants σ_0 and G_0. The functions f_t^i are not necessarily convex, and we denote the global loss function by $F_t(\,\cdot\,) = \frac{1}{n}\sum_{i=1}^n f_t^i(\,\cdot\,)$. For further details on notation, we refer to Section 1.2.

In our algorithm, we make use of linear optimization oracles, whose role is to resolve an online linear optimization problem given a feedback function and a constraint set. Specifically, in the online linear optimization problem, at each time $1 \le t \le T$, one has to select $\boldsymbol{u}_t \in \mathcal{K}$. Subsequently, the adversary reveals a vector $\boldsymbol{d}^t$ and feedbacks the cost function $\langle \boldsymbol{d}_t, \cdot \rangle$. The objective is to minimize regret, that is, $\frac{1}{T}\left(\sum_{t=1}^T \langle \boldsymbol{u}_t, \boldsymbol{d}_t\rangle - \min_{\boldsymbol{u}\in\mathcal{K}} \sum_{t=1}^T \langle \boldsymbol{u}, \boldsymbol{d}_t\rangle\right)$. Several algorithms [Hazan2016a] provide an optimal regret bound of $\mathcal{R}_T = O(1/\sqrt{T})$ for the online linear optimization problem. These algorithms include the online gradient descent algorithm or the follow-the-perturbed-leader algorithm (projection-free). One can pick one of such algorithms to be an oracle that solves the online linear optimization problem.

In the next section, after introducing and recalling useful notions, we will first provide an algorithm for the setting with exact gradients. Subsequently, building on the salient ideas of that algorithm, we extend to the more realistic setting with stochastic gradients.

4.3 An Algorithm with Exact Gradients

Assume that the exact gradients of the loss functions f_t^i are available (or can be computed). The high-level idea of the algorithm is the following. In the algorithm, at every time t, each agent i executes the steps K of the Frank-Wolfe algorithm where every update vector (for iterations $k \in [K]$ where the parameter K will be chosen later) is constructed by combining the output of the linear optimization oracles $\mathcal{O}_k^j$ and the current vectors of its neighbors $j \in N(i)$. During this execution, a set of feasible solutions $\{\boldsymbol{x}_{t,k}^i : k \in [K]\}$ is computed. The solution $\boldsymbol{x}_t^i$ for each agent $i \in [n]$ is then randomly chosen uniformly among $\{\boldsymbol{x}_{t,k}^i : k \in [K]\}$. Subsequently, after communicating and aggregating the information related to functions f_j^t for $j \in N(i)$, the algorithm computes a vector $\boldsymbol{d}_{t,k}^i$ and feedbacks $\langle \boldsymbol{d}_{t,k}^i, \cdot \rangle$ as the cost function at time t to the oracle $\mathcal{O}_k^i$ for $k \in [K]$. The vectors $\boldsymbol{d}_{t,k}^i$' are subtly built so that they capture step by step more and more information on the cumulating cost functions. The formal description is given in Algorithm 9.

Algorithm 9 Distributed Online Algorithm

Input: A convex set $\mathcal{K}$, a time horizon T, a parameter K, online linear optimization oracles $\mathcal{O}_1^i, \ldots, \mathcal{O}_K^i$ for each agent $i \in [n]$, step sizes $\eta_k \in (0,1)$ for all $k \in [K]$

for $t = 1$ to T **do**
  for every agent $i \in [n]$ **do**
    Initialize arbitrarily $\boldsymbol{x}_{t,1}^i \in \mathcal{K}$
    for $k \in [K]$ **do**
      Let $\boldsymbol{v}_{t,k}^i$ be the output of oracle $\mathcal{O}_k^i$ at time step t.
      Send $\boldsymbol{x}_{t,k}^i$ to all neighbors $N(i)$
      Once receiving $\boldsymbol{x}_{j,k}^t$ from all neighbors $j \in N(i)$, set $\boldsymbol{y}_{t,k}^i \leftarrow \sum_j W_{ij} \boldsymbol{x}_{j,k}^t$.
      Compute $\boldsymbol{x}_{t,k+1}^i \leftarrow (1-\eta_k)\boldsymbol{y}_{t,k}^i + \eta_k \boldsymbol{v}_{t,k}^i$.
    end for
    Choose $\boldsymbol{x}_i^t \leftarrow \boldsymbol{x}_{t,k}^i$ for $1 \le k \le K$ with probability $\frac{1}{K}$ and play $\boldsymbol{x}_t^i$
    Receive function f_t^i
    Set $\boldsymbol{g}_{t,1}^i \leftarrow \nabla f_t^i(\boldsymbol{x}_{i,1}^t)$
    for $k \in [K]$ **do**
      Exchange $\boldsymbol{g}_{t,k}^i$ with neighbours $\mathcal{N}(i)$
      $\boldsymbol{d}_{t,k}^i \leftarrow \sum_{j \in \mathcal{N}(i)} w_{ij} \boldsymbol{g}_{t,k}^j$ and
      $\boldsymbol{g}_{t,k+1}^i \leftarrow \left(\nabla f_t^i(\boldsymbol{x}_{t,k+1}^i) - \nabla f_t^i(\boldsymbol{x}_{t,k}^i)\right) + \boldsymbol{d}_{t,k}^i$.
      Feedback function $\left\langle \boldsymbol{d}_{t,k}^i, \cdot \right\rangle$ to oracles $\mathcal{O}_k^i$.
    end for
  end for
end for

Theorem 4.3.1. *Let $\mathcal{K}$ be a convex set with diameter D. Assume that functions F_t (possibly non convex) verify Assumption 1.2.2 with β and G. Choosing step size $\eta_k = \min\left(1, \frac{A}{k^\alpha}\right)$ where $A \in \mathbb{R}_+$ and $\alpha \in (0,1)$. Then, Algorithm 1 guarantees that for all $i \in [n]$:*

$$\max_{\boldsymbol{x} \in \mathcal{K}} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{x}_t^i} \left[\left\langle \nabla F_t(\boldsymbol{x}_t^i), \boldsymbol{x}_t^i - \boldsymbol{x} \right\rangle \right] \le \frac{GDA^{-1}}{K^{1-\alpha}} + \frac{M}{K^\alpha (1-\alpha)} + \mathcal{R}_T$$

where we note

$$M = (\beta C_d + C_g) D + AD^2\beta/2 + 2C_d(\beta D + G)$$

$C_d = k_0\sqrt{n}D$; $C_g = \sqrt{n}\max\left\{\lambda(\mathbf{W})\left(G + \frac{\beta D}{1-\lambda(\mathbf{W})}\right), k_0\beta(4C_d + AD)\right\}$ (see Lemmas 2.4.1 and 2.4.2) and $\mathcal{R}_T$ is the regret of online linear minimization oracles. Choosing $K = T$, $\alpha = 1/2$ and oracles as gradient descent or follow-the-perturbed-leader with regret $\mathcal{R}_T = O\left(T^{-1/2}\right)$, we obtain the gap convergence rate of $O\left(T^{-1/2}\right)$.

4.3.1 Technical Analysis

Before presenting the proof of Theorem 4.3.1, we first introduce some specific notations. Recall the definition of duality gap as :

$$\mathcal{G}(\boldsymbol{x}) = \max_{\boldsymbol{y}\in\mathcal{K}} \langle \nabla F(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{y}\rangle$$

We denote by $\mathcal{G}_t$ and $\mathcal{G}_{t,k}$ the duality gap at time t and at sub-iteration k as:

$$\mathcal{G}_t = \max_{\boldsymbol{x}\in\mathcal{K}} \langle \nabla F_t(\overline{\boldsymbol{x}}_t), \overline{\boldsymbol{x}}_t - \boldsymbol{x}\rangle \qquad \mathcal{G}_{t,k} = \max_{\boldsymbol{x}\in\mathcal{K}} \langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}\rangle$$

where we note by $\overline{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}^i$ as the global decision. If we let $\boldsymbol{x}_{t,k} = \arg\min_{\boldsymbol{x}\in\mathcal{K}} \langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \boldsymbol{x}\rangle$ be the optimal solution at time t and sub-iterate k. Then, we have that $\mathcal{G}_{t,k} = \langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}_{t,k}\rangle$. Finally, we denote by $\mathbb{E}_{\boldsymbol{x}_t}[\,\cdot\,]$ the expectation over the random choice in $\{1,\dots,K\}$ with probability $\frac{1}{K}$.

The structured approach for the proof of Theorem 4.3.1 is as follows:

- We start by deriving an upper bound on the distance between the global gradient $\nabla F_t(\overline{\boldsymbol{x}}_{t,k})$ and the local gradient $\boldsymbol{d}^i_{t,k}$ using results from Lemmas 2.4.1 and 2.4.2.
- The above distance bound, combined with β-smoothness of F_t, allows us to bound the duality gap $\mathcal{G}_{t,k}$, as shown in equation (4.5).
- The proof then proceeds by setting an upper bound on the expectated duality gap $\mathbb{E}_{\boldsymbol{x}_t}[\mathcal{G}_t]$ over K random choices. We employ Lemma 4.3.1 to establish a connection between the expected duality gap at the agent level $\mathcal{G}^i t,k$ and $\mathcal{G}_{t,k}$, as indicated in equation (4.8).
- We finalize the proof by averaging over T iterations and applying Jensen's inequality.

The detailed proof of Theorem 4.3.1 will be presented in the subsequent section. The proofs of lemmas 2.4.1 and 2.4.2 are provided in Section 2.7 respectively.

Lemma 4.3.1. *For every $i \in [n]$ and $k \in [K]$, it holds that*

$$\mathcal{G}^i_{t,k} = \max_{\boldsymbol{x}\in\mathcal{K}} \left\langle \nabla F_t(\boldsymbol{x}^i_{t,k}), \boldsymbol{x}^i_{t,k} - \boldsymbol{x}\right\rangle \leq \max_{\boldsymbol{x}\in\mathcal{K}} \langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}\rangle + (\beta D + G)\,\frac{2C_d}{k^\alpha}$$

Proof. Fix $i \in [n]$ and $k \in [K]$. We have

$$\left\langle \nabla F_t(\boldsymbol{x}^i_{t,k}), \boldsymbol{x}^i_{t,k} - \boldsymbol{x}\right\rangle = \langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}\rangle + \left\langle \nabla F_t(\boldsymbol{x}^i_{t,k}) - \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}\right\rangle + \langle \nabla F_t(\boldsymbol{x}^i_{t,k}), \boldsymbol{x}^i_{t,k} - \overline{\boldsymbol{x}}_{t,k}\rangle$$

Using Lemma 2.4.1, we have

$$\begin{aligned}
\max_{\boldsymbol{x}\in\mathcal{K}} \left\langle \nabla F_t(\boldsymbol{x}^i_{t,k}), \boldsymbol{x}^i_{t,k} - \boldsymbol{x}\right\rangle &\leq \max_{\boldsymbol{x}\in\mathcal{K}} \langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}\rangle + \max_{\boldsymbol{x}\in\mathcal{K}} \left\langle \nabla F_t(\boldsymbol{x}^i_{t,k}) - \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}\right\rangle \\
&\quad + \left\langle \nabla F_t(\boldsymbol{x}^i_{t,k}), \boldsymbol{x}^i_{t,k} - \overline{\boldsymbol{x}}_{t,k}\right\rangle\rangle \\
&\leq \max_{\boldsymbol{x}\in\mathcal{K}} \langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}\rangle + (\beta D + G)\left\|\boldsymbol{x}^i_{t,k} - \overline{\boldsymbol{x}}_{t,k}\right\| \\
&\leq \max_{\boldsymbol{x}\in\mathcal{K}} \langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}\rangle + (\beta D + G)\frac{2C_d}{k^\alpha}
\end{aligned}$$

□

4.3.2 Proof of Theorem 4.3.1

Proof. By β-smoothness, for $k \in [K]$:

$$F_t\left(\overline{\boldsymbol{x}}_{t,k+1}\right) - F_t\left(\overline{\boldsymbol{x}}_{t,k}\right) \leq \left\langle \nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right), \overline{\boldsymbol{x}}_{t,k+1} - \overline{\boldsymbol{x}}_{t,k}\right\rangle + \frac{\beta}{2}\left\|\overline{\boldsymbol{x}}_{t,k+1} - \overline{\boldsymbol{x}}_{t,k}\right\|^2 \tag{4.1}$$

Using proposition 7, the inner product in (4.1) can be written as :

$$\begin{aligned}\left\langle \nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right), \overline{\boldsymbol{x}}_{t,k+1} - \overline{\boldsymbol{x}}_{t,k}\right\rangle &= \eta_k \left\langle \nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right), \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{v}_{t,k}^i - \overline{\boldsymbol{x}}_{t,k} \right\rangle \\ &= \eta_k \left\langle \nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right), \frac{1}{n}\left(\sum_{i=1}^{n} \boldsymbol{v}_{t,k}^i - n\cdot\overline{\boldsymbol{x}}_{t,k}\right) \right\rangle \\ &= \frac{\eta_k}{n}\sum_{i=1}^{n} \left\langle \nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right), \boldsymbol{v}_{t,k}^i - \overline{\boldsymbol{x}}_{t,k}\right\rangle \end{aligned} \tag{4.2}$$

Let $\boldsymbol{x}_{t,k} \in \arg\min_{\boldsymbol{x}\in\mathcal{K}} \langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \boldsymbol{x}\rangle$. Hence,

$$\mathcal{G}_{t,k} = \max_{\boldsymbol{x}\in\mathcal{K}} \langle \nabla F(\overline{\boldsymbol{x}}_{t,k}), \overline{\boldsymbol{x}}_{t,k} - \boldsymbol{v}\rangle = \langle \nabla F(\overline{\boldsymbol{x}}_{t,k}), \overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}_{t,k}\rangle$$

We have :

$$\begin{aligned}&\left\langle \nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right), \boldsymbol{v}_{t,k}^i - \overline{\boldsymbol{x}}_{t,k}\right\rangle \\ &\quad= \langle \nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right) - \boldsymbol{d}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\rangle + \langle \boldsymbol{d}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\rangle + \langle \nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right), \boldsymbol{x}_{t,k} - \overline{\boldsymbol{x}}_{t,k}\rangle \\ &\quad\le \|\nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right) - \boldsymbol{d}_{t,k}^i\| \|\boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\| + \langle \boldsymbol{d}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\rangle + \langle \nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right), \boldsymbol{x}_{t,k} - \overline{\boldsymbol{x}}_{t,k}\rangle \\ &\quad\le \|\nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right) - \boldsymbol{d}_{t,k}^i\| D + \langle \boldsymbol{d}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\rangle + \langle \nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right), \boldsymbol{x}_{t,k} - \overline{\boldsymbol{x}}_{t,k}\rangle.\end{aligned}$$

where we use Cauchy-Schwarz in the first inequality. Using lemmas 2.4.1 and 2.4.2 and β-smoothness of F^t,

$$\begin{aligned}&\left\|\nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right) - \boldsymbol{d}_{t,k}^i\right\| \\ &\quad\le \left\|\nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right) - \frac{1}{n}\sum_{i=1}^{n} \nabla f_t^i(\boldsymbol{y}_{t,k}^i)\right\| + \left\|\frac{1}{n}\sum_{i=1}^{n} \nabla f_t^i(\boldsymbol{y}_{t,k}^i) - \boldsymbol{d}_{t,k}^i\right\| \\ &\quad\le \left\|\frac{1}{n}\sum_{i=1}^{n} \nabla f_i^t\left(\overline{\boldsymbol{x}}_{t,k}\right) - \frac{1}{n}\sum_{i=1}^{n} \nabla f_t^i(\boldsymbol{y}_{t,k}^i)\right\| + \left\|\frac{1}{n}\sum_{i=1}^{n} \nabla f_t^i(\boldsymbol{y}_{t,k}^i) - \boldsymbol{d}_{t,k}^i\right\| \\ &\quad\le \frac{1}{n}\sum_{i=1}^{n} \left\|\nabla f_i^t\left(\overline{\boldsymbol{x}}_{t,k}\right) - \nabla f_t^i(\boldsymbol{y}_{t,k}^i)\right\| + \left\|\frac{1}{n}\sum_{i=1}^{n} \nabla f_t^i(\boldsymbol{y}_{t,k}^i) - \boldsymbol{d}_{t,k}^i\right\| \\ &\quad\le \frac{\beta}{n}\sum_{i=1}^{n} \left\|\overline{\boldsymbol{x}}_{t,k} - \boldsymbol{y}_{t,k}^i\right\| + \left\|\frac{1}{n}\sum_{i=1}^{n} \nabla f_t^i(\boldsymbol{y}_{t,k}^i) - \boldsymbol{d}_{t,k}^i\right\| \qquad \text{(by } \beta \text{ smoothness)} \\ &\quad\le \frac{\beta C_d + C_g}{k^\alpha}\end{aligned}$$

Thus,

$$\left\langle \nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right), \boldsymbol{v}_{t,k}^i - \overline{\boldsymbol{x}}_{t,k}\right\rangle \le \left(\frac{\beta C_d + C_g}{k^\alpha}\right) D + \langle \boldsymbol{d}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\rangle - \mathcal{G}_{t,k}$$

Upper bound the right hand side of equation (4.2) by the above inequality, we have :

$$\left\langle \nabla F_t\left(\overline{\boldsymbol{x}}_{t,k}\right), \overline{\boldsymbol{x}}_{t,k+1} - \overline{\boldsymbol{x}}_{t,k}\right\rangle \le \eta_k \frac{\left(\beta C_d + C_g\right) D}{k^\alpha} + \frac{\eta_k}{n}\sum_{i=1}^{n} \langle \boldsymbol{d}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\rangle - \eta_k \mathcal{G}_{t,k} \tag{4.3}$$

Combining equation (4.1) with equation (4.3) and re-arrange the terms, as $\eta_k = \frac{A}{k^\alpha}$, we have :

$$\eta_k \mathcal{G}_{t,k} \le F_t\left(\overline{\boldsymbol{x}}_{t,k}\right) - F_t\left(\overline{\boldsymbol{x}}_{t,k+1}\right) + \eta_k \frac{(\beta C_d + C_g) D}{k^\alpha} + \frac{\eta_k}{n}\sum_{i=1}^{n} \langle \boldsymbol{d}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{a}_k^t\rangle + \eta_k^2 D^2 \frac{\beta}{2} \tag{4.4}$$

Dividing by η_k yields :

$$\mathcal{G}_{t,k} \le \frac{k^\alpha}{A}\left(F_t\left(\overline{\boldsymbol{x}}_{t,k}\right) - F_t\left(\overline{\boldsymbol{x}}_{t,k+1}\right)\right) + \frac{(\beta C_d + C_g) D}{k^\alpha} + \frac{1}{n}\sum_{i=1}^{n} \langle \boldsymbol{d}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\rangle + \eta_k D^2 \frac{\beta}{2} \tag{4.5}$$

Let $\mathcal{G}_t$ be a random variable such that $\mathcal{G}_t = \mathcal{G}_{t,k}$ with probability $\frac{1}{K}$. We are now bounding $\mathbb{E}_{\overline{\boldsymbol{x}}^t}[\mathcal{G}_t]$. By equation (4.5), using the definition of $\eta_k = \frac{A}{k^\alpha}$, G-Lipschitz of F_t and the fact that $k \in [K]$, we have

$$\begin{aligned}
\mathbb{E}_{\overline{\boldsymbol{x}}^t}[\mathcal{G}_t] &= \frac{1}{K}\sum_{k=1}^{K}\mathcal{G}_{t,k} \\
&\leq \frac{K^\alpha GDA^{-1}}{K} + \frac{(\beta C_d + C_g)\,D}{K}\sum_{k=1}^{K}\frac{1}{k^\alpha} + \frac{1}{nK}\sum_{k=1}^{K}\sum_{i=1}^{n}\langle \boldsymbol{d}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\rangle + \frac{AD^2\beta/2}{K}\sum_{k=1}^{K}\frac{1}{k^\alpha} \\
&\leq \frac{GDA^{-1}}{K^{1-\alpha}} + \frac{(\beta C_d + C_g)\,D + AD^2\beta/2}{K}\sum_{k=1}^{K}\frac{1}{k^\alpha} + \frac{1}{nK}\sum_{k=1}^{K}\sum_{i=1}^{n}\langle \boldsymbol{d}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\rangle \\
&\leq \frac{GDA^{-1}}{K^{1-\alpha}} + \frac{(\beta C_d + C_g)\,D + AD^2\beta/2}{K}\frac{K^{1-\alpha}}{1-\alpha} + \frac{1}{nK}\sum_{k=1}^{K}\sum_{i=1}^{n}\langle \boldsymbol{d}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\rangle \\
&\leq \frac{GDA^{-1}}{K^{1-\alpha}} + \frac{(\beta C_d + C_g)\,D + AD^2\beta/2}{K^\alpha(1-\alpha)} + \frac{1}{nK}\sum_{k=1}^{K}\sum_{i=1}^{n}\left\langle \boldsymbol{d}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\right\rangle
\end{aligned} \tag{4.6}$$

Summing the above inequality for $t \in [T]$ and note that $\frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{d}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\rangle$ is the regret of the oracle $\mathcal{O}_k^i$, we get

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\overline{\boldsymbol{x}}^t}[\mathcal{G}_t] \leq \frac{GDA^{-1}}{K^{1-\alpha}} + \frac{(\beta C_d + C_g)\,D + AD^2\beta/2}{K^\alpha(1-\alpha)} + \mathcal{R}_T \tag{4.7}$$

By uniformly random choice of $\boldsymbol{x}_t^i$ (over all $\boldsymbol{x}_{t,k}^i$ for $k \in [K]$) in the algorithm, we have

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}_t^i}\left[\max_{\boldsymbol{x}\in\mathcal{K}}\langle \nabla F_t(\boldsymbol{x}_t^i), \boldsymbol{x}_t^i - \boldsymbol{x}\rangle\right] = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{K}\sum_{k=1}^{K}\left[\max_{\boldsymbol{x}\in\mathcal{K}}\langle \nabla F_t(\boldsymbol{x}_{t,k}^i), \boldsymbol{x}_{t,k}^i - \boldsymbol{x}\rangle\right] \\
&\leq \frac{1}{T}\sum_{t=1}^{T}\frac{1}{K}\sum_{k=1}^{K}\left[\max_{\boldsymbol{x}\in\mathcal{K}}\langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}\rangle + (\beta D + G)\frac{2C_d}{k^\alpha}\right] \\
&= \frac{1}{T}\sum_{t=1}^{T}\left[\mathbb{E}_{\overline{\boldsymbol{x}}^t}[\mathcal{G}_t] + 2C_d(\beta D + G)\frac{1}{K}\sum_{k=1}^{K}\frac{1}{k^\alpha}\right] \\
&\leq \frac{1}{T}\sum_{t=1}^{T}\left[\mathbb{E}_{\overline{\boldsymbol{x}}^t}[\mathcal{G}_t] + (2C_d(\beta D + G))\frac{K^{1-\alpha}}{K(1-\alpha)}\right] \\
&\leq \frac{1}{T}\sum_{t=1}^{T}\left[\mathbb{E}_{\overline{\boldsymbol{x}}^t}[\mathcal{G}_t] + \frac{2C_d(\beta D + G)}{K^\alpha(1-\alpha)}\right] \\
&\leq \frac{GDA^{-1}}{K^{1-\alpha}} + \frac{(\beta C_d + C_g)\,D + AD^2\beta/2}{K^\alpha(1-\alpha)} + \frac{2C_d(\beta D + G)}{K^\alpha(1-\alpha)} + \mathcal{R}_T \\
&= \frac{GDA^{-1}}{K^{1-\alpha}} + \frac{M}{K^\alpha(1-\alpha)} + \mathcal{R}_T
\end{aligned} \tag{4.8}$$

where we have used the definition

$$\mathbb{E}_{\overline{\boldsymbol{x}}^t}[\mathcal{G}_t] = \mathbb{E}_{\overline{\boldsymbol{x}}^t}\left[\max_{\boldsymbol{x}\in\mathcal{K}}\langle \nabla F_t(\overline{\boldsymbol{x}}_{t,k}), \overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}\rangle\right]$$

and denote by $M = (\beta C_d + C_g)\,D + AD^2\beta/2 + 2C_d(\beta D + G)$. Using Jensen's inequality, we have :

$$\max_{\boldsymbol{x}\in\mathcal{K}}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}_t^i}\left[\langle \nabla F_t(\boldsymbol{x}_t^i), \boldsymbol{x}_t^i - \boldsymbol{x}\rangle\right] \leq \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}_t^i}\left[\max_{\boldsymbol{x}\in\mathcal{K}}\langle \nabla F_t(\boldsymbol{x}_t^i), \boldsymbol{x}_t^i - \boldsymbol{x}\rangle\right] \tag{4.9}$$

The theorem follows equation (4.9), equation (4.8) and equation (4.7) and setting $K = T$. □

4.4 Algorithm with Stochastic Gradients

Assumption 4.4.1. *The function f_t verifies Assumption 1.2.2 and its stochastic gradient $\widetilde{\nabla} f_t(\boldsymbol{x})$ is unbiased, uniformly upper-bounded and has a bounded variance, i.e.,* $\mathbb{E}\left[\widetilde{\nabla} f_t(\boldsymbol{x})\right] = \nabla f_t(\boldsymbol{x})$, $\left\|\widetilde{\nabla} f_t(\boldsymbol{x})\right\| \leq G_0$, *and* $\mathbb{E}\left[\left\|\widetilde{\nabla} f_t(\boldsymbol{x}) - \nabla f_t(\boldsymbol{x})\right\|^2\right] \leq \sigma_0^2$.

In this section, we generalize the previous algorithm to handle the stochastic case, where the agent has access only to a noisy stochastic function. The key difference between the two algorithms is the variance reduction technique used in 18 of Algorithm 10. We treat $\widetilde{\boldsymbol{a}}_{t,k}^i$ as an estimator of the stochastic local gradient $\widetilde{\boldsymbol{d}}_{t,k}^i$ and update it using a momentum-like approach [Mokhtari2017, Ruszczyński1980, Ruszczyński2008, Yang2016] with a parameter ρ_k , as shown below:

$$\widetilde{\boldsymbol{a}}_{t,k}^i = (1-\rho_k)\widetilde{\boldsymbol{a}}_{t,k-1}^i + \rho_k \widetilde{\boldsymbol{d}}_{t,k}^i$$

Since each function f_t^i is stochastic, the local gradient $\widetilde{\boldsymbol{d}}_{t,k}^i$ is a noisy estimate of $\boldsymbol{d}_{t,k}^i$ and, consequently, of the global gradient $\nabla F_t(\boldsymbol{x}_{t,k}^i)$. Using $\widetilde{\boldsymbol{d}}_{t,k}^i$ as feedback for the oracle may lead to divergence in the algorithm due to the presence of non-vanishing noise. To address this, we replace the noisy local gradient $\widetilde{\boldsymbol{d}}_{t,k}^i$ with the variance-reduced feedback $\widetilde{\boldsymbol{a}}_{t,k}^i$ and gradually decrease the momentum parameter ρ_k to 0. This iteratively reduces the noise of the feedback variable by utilizing past gradient estimates. Furthermore, we demonstrate that the noise on the feedback $\widetilde{\boldsymbol{a}}_{t,k}^i$ decrease to 0 at a rate of $O(k^{-2\alpha/3})$, where $\alpha \in (0,1)$, as shown in Lemma 4.4.2.

Algorithm 10 Stochastic online decentralized algorithm

Input: A convex set $\mathcal{K}$, a time horizon T, a parameter $\boldsymbol{x}$, online linear optimization oracles $\mathcal{O}_1^i, \ldots, \mathcal{O}_K^i$ for each player $i \in [n]$, step sizes $\eta_k \in (0,1)$ for all $k \in [K]$

Initialize linear optimizing oracle $\mathcal{O}_k^i$ for all $k \in [K]$
for $t = 1 \ldots T$ **do**
  for every agent $i \in [n]$ **do**
    Initialize arbitrarily $\boldsymbol{x}_{t,1}^i \in \mathcal{K}$ and set $\widetilde{\boldsymbol{a}}_{t,0}^i \leftarrow \boldsymbol{0}$
    for $k = 1 \ldots K$ **do**
      Query $\boldsymbol{v}_{t,k}^i$ from $\mathcal{O}_k^i$.
      Exchange $\boldsymbol{x}_{t,k}^i$ with neighbours $\mathcal{N}(i)$
      $\boldsymbol{y}_{t,k}^i \leftarrow \sum_j w_{ij} \boldsymbol{x}_{t,k}^j$.
      $\boldsymbol{x}_{t,k+1}^i \leftarrow (1-\eta_k)\boldsymbol{y}_{t,k}^i + \eta_k \boldsymbol{v}_{t,k}^i$.
    end for
    Choose $\boldsymbol{x}_t^i \leftarrow \boldsymbol{x}_{t,k}^i$ for $k \in [K]$ with probability $\frac{1}{K}$ and play $\boldsymbol{x}_t^i$
    Receive function f_t^i and an unbiased gradient estimate $\widetilde{\nabla} f_t^i$
    Set $\widetilde{\boldsymbol{g}}_{t,1}^i \leftarrow \widetilde{\nabla} f_t^i(\boldsymbol{x}_{i,1}^t)$
    for $k = 1 \ldots K$ **do**
      Exchange $\widetilde{\boldsymbol{g}}_{t,k}^i$ with neighbours $\mathcal{N}(i)$.
      $\widetilde{\boldsymbol{d}}_{t,k}^i \leftarrow \sum_{j \in \mathcal{N}(i)} w_{ij} \widetilde{\boldsymbol{g}}_{t,k}^j$
      $\widetilde{\boldsymbol{g}}_{i,k+1}^t \leftarrow \left(\widetilde{\nabla} f_t^i(\boldsymbol{x}_{t,k+1}^i) - \widetilde{\nabla} f_t^i(\boldsymbol{x}_{t,k}^i)\right) + \widetilde{\boldsymbol{d}}_{t,k}^i$.
      $\widetilde{\boldsymbol{a}}_{t,k}^i \leftarrow (1-\rho_k) \cdot \widetilde{\boldsymbol{a}}_{t,k-1}^i + \rho_k \cdot \widetilde{\boldsymbol{d}}_{t,k}^i$.
      Feedback function $\langle \widetilde{\boldsymbol{a}}_{t,k}^i, \cdot \rangle$ to oracles $\mathcal{O}_k^i$.
    end for
  end for
end for

Theorem 4.4.1. *Let $\mathcal{K}$ be a convex set with diameter D. Assume that the functions f_t^i's verify Assumption 1.2.2 with constant β, G and its stochastic estimate $\widetilde{f}_t^i$'s verify Assumption 4.4.1 with constant σ_0 and G_0. Then, let $A \in \mathbb{R}_+^*$ and the step-sizes $\eta_k = \min\{1, \frac{A}{k^{3/4}}\}$, we have for all*

$i \in [n]$, we have

$$\max_{\boldsymbol{x}\in\mathcal{K}} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}_t^i}\left[\left\langle \nabla F_t\left(\boldsymbol{x}_t^i\right), \boldsymbol{x}_t^i - \boldsymbol{x}\right\rangle\right]\right] \leq \frac{N}{K^{1/4}} + \frac{M}{K^{3/4}} + \mathcal{R}_T$$

where we note $M = 2D\left((\beta C_d + C_g) + AD\beta + 8C_d\left(\beta D + G\right)\right)$, $N = GDA^{-1} + 2Q^{1/2}D$ and Q, C_d, C_g are defined in Lemmas 2.4.1, 2.4.2 and 4.4.2. Choosing $K = T$ and oracles with regret $\mathcal{R}_T = O(\sqrt{T})$, we obtain the convergence gap of $O\left(T^{-1/4}\right)$.

4.4.1 Technical Analysis

In this section, we are going to provide the proof for Theorem 4.4.1. We will continue to use the same notation that was introduced in Section 4.3.1, but we will substitute the exact gradients with their stochastic equivalents. Since the primary distinction between the two algorithms lies in the variance reduction method, we will develop some additional results to manage the stochastic nature of the gradients.

The proof of Theorem 4.4.1 is structured as follows:

- We first derive a bound on the variance of the local stochastic gradient $\widetilde{d}_{t,k}^i$ in Lemma 4.4.1
- In Proposition 4.4.1, we quantify the distance between two consecutive local gradient $\boldsymbol{d}_{t,k+1}^i$ and $\boldsymbol{d}_{t,k}^i$.
- From the results in Lemma 4.4.1 and Proposition 4.4.1, we can establish the distance between the local gradient $\boldsymbol{d}_{t,k}^i$ and the variance reduction estimate $\widetilde{\boldsymbol{a}}_{t,k}^i$ the in Lemma 4.4.2.
- The main part proceeds by combining the results from Equation (4.6) from Section 4.3.1 and Lemma 4.4.2 to establish an upper bound on the expected duality gap $\mathbb{E}_{\boldsymbol{x}_t}\left[\mathcal{G}_t\right]$. Following the same steps as in Section 4.3.1, we use Lemma 4.3.1 to establish the connection between the expected duality gap at local level $\mathcal{G}_{t,k}^i$ and the global duality gap $\mathcal{G}_{t,k}$.
- We finalize the proof by avering over T rounds and use Jensen's inequality to establish the final result.

The detailed proofs of Theorem 4.4.1 are presented in the subsequent section and we postpone the proof of the lemmas and propositions to the end of the chapter.

Lemma 4.4.1. *Under Assumption 4.4.1 and let $\sigma_1^2 = 4n\left[\left(\frac{G+G_0}{\frac{1}{\lambda(\mathbf{W})}-1}\right)^2 + 2\sigma_0^2\right]$. For $i \in [n], k \in [K]$, the variance of the local stochastic gradient is uniformly bounded i.e*

$$\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{d}}_{t,k}^i\right\|^2\right] \leq \sigma_1^2$$

Proof. See Lemma 4.7.2. □

Proposition 4.4.1. *For $t \in [T]$, $i \in [n]$, it holds that,*

$$\|\boldsymbol{d}_{t,k+1}^i - \boldsymbol{d}_{t,k}^i\| \leq \frac{B}{(k+3)^\alpha}$$

where $B = 9C_g + 5\beta\left(4C_d + AD\right)$.

Proof. See Proposition 4.7.1. □

Lemma 4.4.2. *Let $Q = \max\left\{5^{2\alpha/3}\left\|\boldsymbol{d}_{t,1}^i - \widetilde{\boldsymbol{a}}_{t,1}^i\right\|^2, 4\sigma_1^2 + 2B^2\right\}$, where $\sigma_1^2 = 4n\left[\left(\frac{G+G_0}{\frac{1}{\lambda(\mathbf{W})}-1}\right)^2 + 2\sigma_0^2\right]$ and B are defined in Lemma 4.4.1 and Proposition 4.4.1. For $i \in [n]$ and $k \in [K]$, we have*

$$\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{a}}_{t,k}^i\right\|^2\right] \leq \frac{Q}{(k+4)^{2\alpha/3}}$$

Proof. See Lemma 4.7.1. □

4.4.2 Proof of Theorem 4.4.1

Proof. By equation (4.6) in the proof of theorem 4.3.1, we have:

$$\mathbb{E}_{\overline{x}_t}\left[\mathcal{G}_t\right] \leq \frac{GDA^{-1}}{K^{1/4}} + \frac{2D\left((\beta C_d + C_g) + AD\beta\right)}{K^{3/4}} + \frac{1}{nK}\sum_{k=1}^{K}\sum_{i=1}^{n}\langle \boldsymbol{d}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\rangle$$

$$\leq \frac{GDA^{-1}}{K^{1/4}} + \frac{2D\left((\beta C_d + C_g) + AD\beta\right)}{K^{3/4}} + \frac{1}{nK}\sum_{k=1}^{K}\sum_{i=1}^{n}\left\langle \boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{a}}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\right\rangle + \frac{1}{nK}\sum_{k=1}^{K}\sum_{i=1}^{n}\left\langle \widetilde{\boldsymbol{a}}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\right\rangle$$

$$\leq \frac{GDA^{-1}}{K^{1/4}} + \frac{2D\left((\beta C_d + C_g) + AD\beta\right)}{K^{3/4}} + \frac{1}{nK}\sum_{k=1}^{K}\sum_{i=1}^{n}\left\|\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{a}}_{t,k}^i\right\|\left\|\boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\right\| + \frac{1}{nK}\sum_{k=1}^{K}\sum_{i=1}^{n}\left\langle \widetilde{\boldsymbol{a}}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\right\rangle$$

$$\leq \frac{GDA^{-1}}{K^{1/4}} + \frac{2D\left((\beta C_d + C_g) + AD\beta\right)}{K^{3/4}} + \frac{D}{nK}\sum_{k=1}^{K}\sum_{i=1}^{n}\left\|\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{a}}_{t,k}^i\right\| + \frac{1}{nK}\sum_{k=1}^{K}\sum_{i=1}^{n}\left\langle \widetilde{\boldsymbol{a}}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\right\rangle$$

We take the average over T iterations and the expectation over the randomness of the stochastic gradient estimates. We obtain:

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\overline{x}_t}\left[\mathcal{G}_t\right]\right]$$

$$\leq \frac{GDA^{-1}}{K^{1/4}} + \frac{2D\left((\beta C_d + C_g) + AD\beta\right)}{K^{3/4}} + \frac{D}{nK}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{a}}_{t,k}^i\right\|\right] + \mathbb{E}\left[\frac{1}{nKT}\sum_{k=1}^{K}\sum_{i=1}^{n}\sum_{t=1}^{T}\langle \widetilde{\boldsymbol{a}}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\rangle\right]$$

$$\leq \frac{GDA^{-1}}{K^{1/4}} + \frac{2D\left((\beta C_d + C_g) + AD\beta\right)}{K^{3/4}} + \frac{Q^{1/2}D}{K}\sum_{k=1}^{K}\frac{1}{(k+4)^{1/4}} + \mathbb{E}\left[\frac{1}{nKT}\sum_{k=1}^{K}\sum_{i=1}^{n}\sum_{t=1}^{T}\left\langle \widetilde{\boldsymbol{a}}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\right\rangle\right]$$

$$\leq \frac{GDA^{-1}}{K^{1/4}} + \frac{2D\left((\beta C_d + C_g) + AD\beta\right)}{K^{3/4}} + \frac{2Q^{1/2}D}{K^{1/4}} + \mathbb{E}\left[\frac{1}{nLT}\sum_{k=1}^{K}\sum_{i=1}^{n}\sum_{t=1}^{T}\left\langle \widetilde{\boldsymbol{a}}_{t,k}^i, \boldsymbol{v}_{t,k}^i - \boldsymbol{x}_{t,k}\right\rangle\right]$$

$$\leq \frac{GDA^{-1}}{K^{1/4}} + \frac{2D\left((\beta C_d + C_g) + AD\beta\right)}{K^{3/4}} + \frac{2Q^{1/2}D}{K^{1/4}} + \mathcal{R}_T$$

where we have used the fact that $\sum_{k=1}^{K}\frac{1}{(k+4)^{1/4}} \leq 2K^{3/4}$ on the and the regret of the online optimization oracles $\mathcal{R}_T$. Recall that,

$$\mathbb{E}_{\overline{\boldsymbol{x}}_t}\left[\mathcal{G}_t\right] = \mathbb{E}_{\overline{\boldsymbol{x}}_t}\left[\max_{\boldsymbol{x}\in\mathcal{K}}\langle \nabla F_t\left(\overline{\boldsymbol{x}}_t\right), \overline{\boldsymbol{x}}_t - \boldsymbol{x}\rangle\right]$$

Using lemma 4.3.1, we have

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}_t^i}\left[\max_{\boldsymbol{x}\in\mathcal{K}}\left\langle \nabla F_t\left(\boldsymbol{x}_t^i\right), \boldsymbol{x}_t^i - \boldsymbol{x}\right\rangle\right]\right] \leq \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\overline{\boldsymbol{x}}_t}\left[\max_{\boldsymbol{x}\in\mathcal{K}}\left\langle \nabla F_t\left(\overline{\boldsymbol{x}}_t\right), \overline{\boldsymbol{x}}_t - \boldsymbol{x}\right\rangle\right]\right] + \frac{2C_d\left(\beta D + G\right)}{K^{\alpha}(1-\alpha)}$$

$$\leq \frac{GDA^{-1}}{K^{1/4}} + \frac{2D\left((\beta C_d + C_g) + AD\beta\right)}{K^{3/4}} + \frac{2Q^{1/2}D}{K^{1/4}} + \frac{8C_d\left(\beta D + G\right)}{K^{3/4}} + \mathcal{R}_T$$

$$= \frac{GDA^{-1} + 2Q^{1/2}D}{K^{1/4}} + \frac{2D\left((\beta C_d + C_g) + AD\beta + 8C_d\left(\beta D + G\right)\right)}{K^{3/4}} + \mathcal{R}_T$$

The result follows by setting $K = T$, $M = 2D\left((\beta C_d + C_g) + AD\beta + 8C_d\left(\beta D + G\right)\right)$ and Jensen's inequality of the max function on the left-hand side of the inequality. □

4.5 Experiments

The data-set used for experimentation comes from a 7 storey building with 24 sensor equipped zones [Pipattanasomporn2020].The zone-wise knowledge exchange happens through the edges of

an undirected graph of n nodes participating in the learning process. For every round t, each node i receives a batch $\mathcal{B}_i^t$ of 32 time-series sequences corresponding to a look-back period 13 timestep to predict the temperature of the next timestep. We extract the data from March 7^{th} to April 20^{th} for training, set L equal to 360, $\alpha = 0.95$ and $A = 1$. A min-max scaler is used to normalize the data and we apply a rolling window with stride 1 on the original time series. Each node is embedded with a model built from a two-layers long-short-time-memory (LSTM) network followed by a fully connected layer. Denote the output of the model i for a data sequence b at time t by $\hat{y}_{i,b}^t$ and its ground truth by $y_{i,b}^t$. Consider the ℓ_1 loss as the objective function :

$$\mathcal{L}(\hat{y}_{i,b}^t, y_{i,b}^t) = \begin{cases} \dfrac{(\hat{y}_{i,b}^t - y_{i,b}^t)^2}{2} & \text{if } |\hat{y}_{i,b}^t - y_{i,b}^t| \leq 1 \\ |\hat{y}_{i,b}^t - y_{i,b}^t| - \frac{1}{2} & \text{otherwise.} \end{cases}$$

Consider the constraint set $\mathcal{K} = \{\boldsymbol{x} \in \mathbb{R}^d, \|\boldsymbol{x}\|_1 \leq r\}$, where $\boldsymbol{x}$ is the model's weight, d its dimension and $r = 1$. The (normalized) loss incurred by the data of agent i is $\frac{1}{|\mathcal{B}_i^t|}\sum_{b\in\mathcal{B}_i^t}\mathcal{L}(\hat{y}_{i,b}^t, y_{i,b}^t)$. The global loss function incurred by the overall data is

$$F^t(\boldsymbol{x}) = \frac{1}{|\cup_{i=1}^n \mathcal{B}_i^t|}\sum_{b\in\cup_{i=1}^n\mathcal{B}_i^t}\mathcal{L}(\hat{y}_{i,b}^t, y_{i,b}^t),$$

that can be written as $F^t(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^n f_i^t(\boldsymbol{x})$ where $f_i^t(\boldsymbol{x}) = \frac{1}{|\mathcal{B}_i^t|}\sum_{b\in\mathcal{B}_i^t}\mathcal{L}(\hat{y}_{i,b}^t, y_{i,b}^t)$. Note that the non-convexity here is due to the non-convexity of $\hat{y}_{i,b}^t$ as a function of $\boldsymbol{x}_i^t$. In the following section, if not specify otherwise, we call *loss* the temporal average of the global loss function F^t defined as $\frac{1}{T}\sum_{t=1}^T F^t$.

4.5.1 Prediction Performance

Figures 4.1 shows the loss and gap values for different network sizes. The implementation justifies our theoretical results about the convergence of the gap. Besides, we also observe the convergence of loss value, an expected implication of the gap convergence. We set M the number of prediction points between the 21^{st} and 24^{th} of April and n the number of zones within one configuration. We use the mean absolute error (MAE $= \frac{1}{nM}\sum_{i=1}^n\sum_{m=1}^M|\hat{y}_{i,m} - y_{i,m}|$) and mean square error (MSE $= \frac{1}{nM}\sum_{i=1}^n\sum_{m=1}^M(\hat{y}_{i,m} - y_{i,m})^2$) as a measure between the prediction and the ground truth. We observe that increasing nodes in a network does not always lead to better online performance. In-fact, a 7 node configuration achieves the lowest MSE (0.65) and MAE (0.78) for floors 6 and 7. We see a 40 % drop in MSE and 20 % reduction in MAE for floor 6 zonal models when 3 extra peers from floor 7 joined the group. We observe 19 % and 25 % increase in MSE and MAE values by adding zonal nodes from floor 7 to a 10 node group. This can be best argued by the fact that the top floor of a building has a non identical thermal variation with the rest of the storeys.

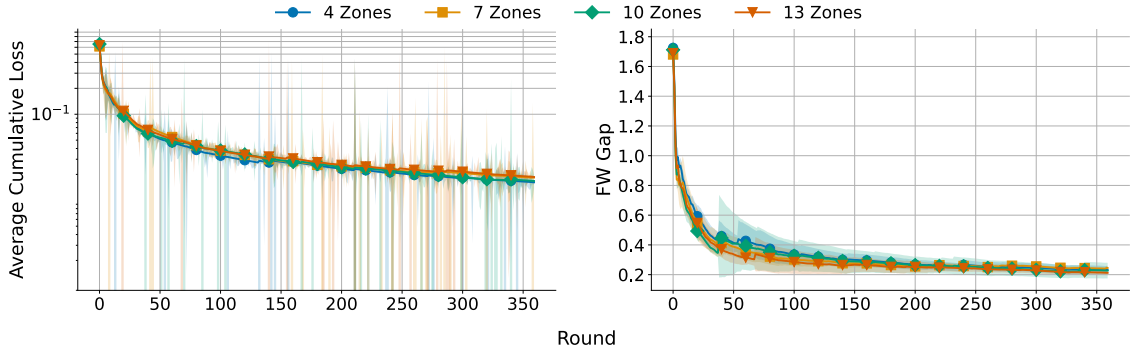


Figure 4.1: Loss and Gap values of different network size on complete topology (Plot on log-scale)

4.5.2 Effect of Network Topology

We study the effect of topology in learning for a 7 node configuration with a complete, cycle and line graph containing 28, 7 and 6 edges respectively and with 13 nodes having 78,13 and 12 edges

respectively. For both 7 (Table 4.1) and 13 (Table 4.2) node configurations, we observe that the complete graph yields the least amount of prediction error, mean absolute error $\in [0.66, 1.3]°C$. However we note the peculiarity that the line graph can perform better than a cycle graph and has roughly a 10 % error margin compared to the complete configuration.

Topology	Metric	Mean	Var	Max	Min
Cycle	MAE	1.09	0.48	1.80	0.56
	MSE	0.78	0.21	1.09	0.52
Complete	MAE	**0.77**	**0.38**	**1.47**	0.27
	MSE	**0.64**	**0.20**	**1.04**	0.39
Line	MAE	0.81	0.53	1.95	**0.24**
	MSE	0.66	0.28	1.26	**0.34**

Tableau 4.1: Impact of Topology on 7 learners configuration.

Topology	Metric	Mean	Var	Max	Min
Cycle	MAE	1.51	1.46	6.16	0.36
	MSE	0.94	0.38	1.90	0.48
Complete	MAE	**1.26**	**0.82**	3.64	**0.32**
	MSE	**0.85**	**0.27**	**1.50**	**0.42**
Line	MAE	1.38	0.91	3.17	0.50
	MSE	0.90	0.35	**1.66**	0.49

Tableau 4.2: Impact of Topology on 13 learners configuration.

4.5.3 Effect of Decentralization

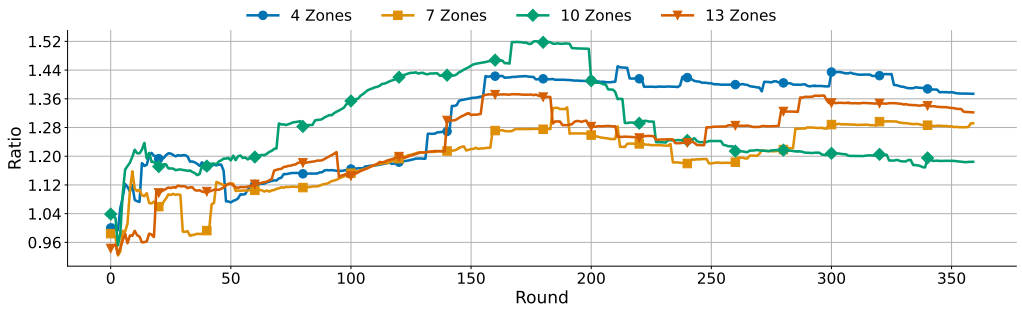


Figure 4.2: Loss ratio of decentralized and centralized Meta Frank-Wolfe on different network size.

We are interested in understanding the role of decentralization in terms of accuracy of zonal learners. Let $L_{MFW}(t)$ be the loss from Meta Frank Wolfe (MFW) at time t. The approximation ratio $A(t) = \frac{L_{DMFW}(t)}{L_{MFW}(t)}$ at time t represents how worse is our decentralized version compared to a centralized optimization. $A(t) \leq B_{max}$ will mean our algorithm performs no worse than B_{max} times of the MFW. On figure 4.2, we plot the ratio $A(t)$ for a 13 node network and show that $A(t) \leq 1.4$. The 7 node network has the closest approximation bounded by 1.35 which can be explained by earlier insights on performance accuracy. We notice that the 10 node network performs worse till

$t = 200$ and after $t \geq 250$ or 21 hours, the approximation ratio becomes close to centralised version with less than 20 % error.

4.6 Concluding remarks

In this chapter, we presented an online algorithm aimed at minimizing non-convex loss functions that are aggregated from local data distributed across a network. We introduced a measure called the convergence gap, which is a generalized version of the Frank-Wolfe gap in the online setting, and demonstrated a convergence rate of $O(T^{-1/2})$ and $O(T^{-1/4})$ for the exact and stochastic gradient settings, respectively. To validate our theoretical analysis, we performed experiments using a real-life smart building dataset. The results of these experiments highlight the value of our approach for learning in distributed settings. However, it is important to note that while the algorithms achieve good performance in terms of regret-like measures, the convergence gap does not directly guarantee finding a stationary point of the objective function. A potential avenue for future work is to investigate convergence to a stationary point of each local function F_t by analyzing the duality gap and examining the function variations. This line of research could lead to a more comprehensive understanding of the algorithm's performance in online nonconvex optimization problems.

4.7 Missing proofs of Chapter 4

Lemma 4.7.1 (Lemma 4.4.2)**.** *Let* $Q = \max\left\{5^{2\alpha/3}\left\|\boldsymbol{d}_{t,1}^i - \widetilde{\boldsymbol{a}}_{t,1}^i\right\|^2, 4\sigma_1^2 + 2B^2\right\}$*, where* $\sigma_1^2 = 4n\left[\left(\frac{G+G_0}{\frac{1}{\lambda(\mathbf{W})^{-1}}}\right)^2 + 2\sigma_0^2\right]$ *and* B *are defined in Lemma 4.4.1 and Proposition 4.4.1. For* $i \in [n]$ *and* $k \in [K]$*, we have*

$$\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{a}}_{t,k}^i\right\|^2\right] \leq \frac{Q}{(k+4)^{2\alpha/3}}$$

Proof. The proof follows similar idea to the one of Lemma 3.3.4 and Lemma 3 in [Zhang2020]. We state it here for completeness. By definition of variance reduction step, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{a}}_{t,k}^i\right\|^2\right] = \mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i - (1-\rho_k)\widetilde{\boldsymbol{a}}_{t,k-1}^i - \rho_k\widetilde{\boldsymbol{d}}_{t,k}^i\right\|^2\right] \\
=&\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i + \rho_k\boldsymbol{d}_{t,k}^i - \rho_k\boldsymbol{d}_{t,k}^i + (1-\rho_k)\widetilde{\boldsymbol{a}}_{t,k-1}^i - \rho_k\widetilde{\boldsymbol{d}}_{t,k}^i\right\|^2\right] \\
=&\mathbb{E}\left[\left\|(1-\rho_k)\boldsymbol{d}_{t,k}^i + \rho_k\left(\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{d}}_{t,k}^i\right) + (1-\rho_k)\widetilde{\boldsymbol{a}}_{t,k-1}^i\right\|^2\right] \\
=&\mathbb{E}\left[\left\|(1-\rho_k)\left(\boldsymbol{d}_{t,k}^i - \boldsymbol{d}_{t,k-1}^i\right) + (1-\rho_k)\left(\boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{t,k-1}^i\right) + \rho_k\left(\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{d}}_{t,k}^i\right)\right\|^2\right] \\
=&\rho_k^2\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{d}}_{t,k}^i)\right\|^2\right] + (1-\rho_k)^2\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i - \boldsymbol{d}_{t,k-1}^i\right\|^2\right] + (1-\rho_k)^2\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{t,k-1}^i\right\|^2\right] \\
&+ 2\rho_k(1-\rho_k)\mathbb{E}\left[\left\langle\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{d}}_{t,k}^i, \boldsymbol{d}_{t,k}^i - \boldsymbol{d}_{t,k-1}^i\right\rangle\right] + 2\rho_k(1-\rho_k)\mathbb{E}\left[\left\langle\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{d}}_{t,k}^i, \boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{t,k-1}^i\right\rangle\right] \\
&+ 2(1-\rho_k)^2\mathbb{E}\left[\left\langle\boldsymbol{d}_{t,k}^i - \boldsymbol{d}_{t,k-1}^i, \boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\rangle\right] \qquad (4.10)
\end{aligned}$$

Recall that $\mathbb{E}\left[\widetilde{\boldsymbol{d}}_{t,k}^i\right] = \boldsymbol{d}_{t,k}^i$, we have the following results

$$\mathbb{E}\left[\left\langle\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{d}}_{t,k}^i, \boldsymbol{d}_{t,k}^i - \boldsymbol{d}_{t,k-1}^i\right\rangle\right] = 0 \quad \text{and} \quad \mathbb{E}\left[\left\langle\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{d}}_{t,k}^i, \boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{t,k-1}^i\right\rangle\right] = 0$$

From Lemma 4.4.1 and Proposition 4.4.1, we have:

$$\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{d}}_{t,k}^i\right\|^2\right] \leq \sigma_1^2 \quad \text{and} \quad \mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i - \boldsymbol{d}_{t,k-1}^i\right\|^2\right] \leq \frac{B^2}{(k+3)^{2\alpha}}$$

From Young's inequality, we have

$$\begin{aligned}
\mathbb{E}\left[\left\langle\boldsymbol{d}_{t,k}^i - \boldsymbol{d}_{t,k-1}^i, \boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\rangle\right] &\leq \frac{1}{2\alpha_k}\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i - \boldsymbol{d}_{t,k-1}^i\right\|^2\right] + \frac{\alpha_k}{2}\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\|^2\right] \\
&\leq \frac{1}{2\alpha_k}\frac{B^2}{(k+3)^{2\alpha}} + \frac{\alpha_k}{2}\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\|^2\right]
\end{aligned}$$

Combining the above results with equation (4.10), we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{a}}_{t,k}^i\right\|^2\right] \leq \rho_k^2\sigma_1^2 + (1-\rho_k)^2\frac{B^2}{(k+3)^{2\alpha}} + (1-\rho_k)^2\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{t,k-1}^i\right\|^2\right] \\
&\quad + 2(1-\rho_k)^2\mathbb{E}\left[\left\langle\boldsymbol{d}_{t,k}^i - \boldsymbol{d}_{t,k-1}^i, \boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\rangle\right] \\
&\leq \rho_k^2\sigma_1^2 + (1-\rho_k)^2\frac{B^2}{(k+3)^{2\alpha}} + (1-\rho_k)^2\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{t,k-1}^i\right\|^2\right] \\
&\quad + \frac{(1-\rho_k)^2}{\alpha_k}\frac{B^2}{(k+3)^{2\alpha}} + (1-\rho_k)^2\alpha_k\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\|^2\right] \\
&= \rho_k^2\sigma_1^2 + (1-\rho_k)^2\left(1 + \frac{1}{\alpha_k}\right)\frac{B^2}{(k+3)^{2\alpha}} + (1-\rho_k)^2(1+\alpha_k)\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\|^2\right]
\end{aligned}$$

Let $\alpha_k = \frac{\rho_k}{2}$, since $\rho_k = \frac{2}{(k+3)^{2\alpha/3}}$, we have $\alpha_k = \frac{1}{(k+3)^{2\alpha/3}}$. Then, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{a}}_{t,k}^i\right\|^2\right] \leq \rho_k^2\sigma_1^2 + \left(1+\frac{2}{\rho_k}\right)\frac{B^2}{(k+3)^{2\alpha}} + (1-\rho_k)\,\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\|^2\right] \\
&\leq \frac{4}{(k+3)^{4\alpha/3}}\sigma_1^2 + \left(1+(k+3)^{2\alpha/3}\right)\frac{B^2}{(k+3)^{2\alpha}} + \left(1-\frac{2}{(k+3)^{2\alpha/3}}\right)\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\|^2\right] \\
&\leq \frac{4\sigma_1^2}{(k+3)^{4\alpha/3}} + \frac{B^2}{(k+3)^{2\alpha}} + \frac{B^2}{(k+3)^{4\alpha/3}} + \left(1-\frac{2}{(k+3)^{2\alpha/3}}\right)\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\|^2\right] \\
&\leq \frac{4\sigma_1^2+2B^2}{(k+3)^{4\alpha/3}} + \left(1-\frac{2}{(k+3)^{2\alpha/3}}\right)\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\|^2\right] \\
&\triangleq \frac{Q_0}{(k+3)^{4\alpha/3}} + \left(1-\frac{2}{(k+3)^{2\alpha/3}}\right)\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\|^2\right]
\end{aligned}$$

We will prove the result by induction. For $k=1$, we let $Q_1 = \mathbb{E}\left[\left\|\boldsymbol{d}_{t,1}^i - \widetilde{\boldsymbol{a}}_{t,1}^i\right\|^2\right]$ and we define $Q = \max\left\{5^{2\alpha/3}Q_1, 4\sigma_1^2+2B^2\right\}$.Suppose the result holds for $k-1$, we have

$$\begin{aligned}
\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{a}}_{t,k}^i\right\|^2\right] &\leq \frac{Q_0}{(k+3)^{4\alpha/3}} + \left(1-\frac{2}{(k+3)^{2\alpha/3}}\right)\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k-1}^i - \widetilde{\boldsymbol{a}}_{q,k-1}^i\right\|^2\right] \\
&\leq \frac{Q_0}{(k+3)^{4\alpha/3}} + \left(1-\frac{2}{(k+3)^{2\alpha/3}}\right)\frac{Q}{(k+3)^{2\alpha/3}} \\
&\leq \frac{Q_0}{(k+3)^{4\alpha/3}} + \frac{Q}{(k+3)^{2\alpha/3}}\cdot\frac{(k+3)^{2\alpha/3}-2}{(k+3)^{2\alpha/3}} \\
&\leq \frac{Q\left((k+3)^{2\alpha/3}-1\right)}{(k+3)^{4\alpha/3}} \leq \frac{Q}{(k+4)^{2\alpha/3}}
\end{aligned}$$

since $\frac{(k+3)^{2\alpha/3}-1}{(k+3)^{4\alpha/3}} \leq \frac{1}{(k+4)^{2\alpha/3}}$ for all $k \geq 1$. The proof is complete. □

Lemma 4.7.2 (Lemma 4.4.1)**.** *Under Assumption 4.4.1 and let* $\sigma_1^2 = 4n\left[\left(\frac{G+G_0}{\frac{1}{\lambda(\mathbf{W})}-1}\right)^2 + 2\sigma_0^2\right]$. *For* $i \in [n], k \in [K]$*, the variance of the local stochastic gradient is uniformly bounded i.e*

$$\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i - \widetilde{\boldsymbol{d}}_{t,k}^i\right\|^2\right] \leq \sigma_1^2$$

Proof. We denote $\widetilde{\boldsymbol{d}}^{cat}$ the stochastique version of $\boldsymbol{d}^{cat}$, following proposition 5, we have

$$\begin{aligned}
\widetilde{\boldsymbol{d}}_{t,k}^{cat} = &\sum_{\tau=1}^{k-1}\left[\left(\mathbf{W}^{k-\tau} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\left(\widetilde{\nabla} f_{t,\tau+1}^{cat} - \widetilde{\nabla} f_{t,\tau}^{cat}\right) \\
&+ \left[\left(\mathbf{W}^k - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\widetilde{\nabla} f_{t,1}^{cat} + \left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\otimes I_d\right)\widetilde{\nabla} f_{t,k}^{cat}
\end{aligned} \tag{4.11}$$

Then, we have

$$\begin{aligned}
\boldsymbol{d}_{t,k}^{cat} - \widetilde{\boldsymbol{d}}_{t,k}^{cat} = &\sum_{\tau=1}^{k-1}\left[\left(\mathbf{W}^{k-\tau} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\left(\nabla f_{t,\tau+1}^{cat} - \widetilde{\nabla} f_{t,\tau+1}^{cat} + \widetilde{\nabla} f_{t,k}^{cat} - \nabla f_{t,k}^{cat}\right) \\
&+ \left[\left(\mathbf{W}^k - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\otimes I_d\right]\left(\nabla f_{t,1}^{cat} - \widetilde{\nabla} f_{t,1}^{cat}\right) + \left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\otimes I_d\right)\left(\nabla f_{t,k}^{cat} - \widetilde{\nabla} f_{t,k}^{cat}\right)
\end{aligned} \tag{4.12}$$

By Assumption 4.4.1 and Jensen's inequality, we have

$$\begin{aligned}
\mathbb{E}\left[\left\|\nabla f_{t,k}^{cat} - \widetilde{\nabla} f_{t,k}^{cat}\right\|^2\right] &= \mathbb{E}\left[\sum_{i=1}^n\left\|\nabla f_{t,k}^i\left(\boldsymbol{x}_{t,k}^i\right) - \widetilde{\nabla} f_{t,k}^i\left(\boldsymbol{x}_{t,k}^i\right)\right\|^2\right] \\
&\leq \sqrt{\sum_{i=1}^n\mathbb{E}\left[\left\|\nabla f_{t,k}^i\left(\boldsymbol{x}_{t,k}^i\right) - \widetilde{\nabla} f_{t,k}^i\left(\boldsymbol{x}_{t,k}^i\right)\right\|^2\right]} \leq \sqrt{n}\sigma_0
\end{aligned} \tag{4.13}$$

Taking the second moment of equation (4.12), we have

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^{cat}-\widetilde{\boldsymbol{d}}_{t,k}^{cat}\right\|^2\right]\\
\leq&\mathbb{E}\left[\left(\sum_{\tau=1}^{k-1}\left\|\mathbf{W}^{k-\tau}-\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\|\left\|\nabla f_{t,\tau+1}^{cat}-\widetilde{\nabla} f_{t,\tau+1}^{cat}+\widetilde{\nabla} f_{t,k}^{cat}-\nabla f_{t,k}^{cat}\right\|\right)^2\right]\\
&+\mathbb{E}\left[\left\|\left(\mathbf{W}^k-\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\otimes I_d\right)\left(\nabla f_{t,1}^{cat}-\widetilde{\nabla} f_{t,1}^{cat}\right)+\left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\otimes I_d\right)\left(\nabla f_{t,k}^{cat}-\widetilde{\nabla} f_{t,k}^{cat}\right)\right\|^2\right]\\
\leq&\mathbb{E}\left[\left(\sum_{\tau=1}^{k-1}\left\|\mathbf{W}^{k-\tau}-\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\|\left\|\nabla f_{t,\tau+1}^{cat}-\widetilde{\nabla} f_{t,\tau+1}^{cat}+\widetilde{\nabla} f_{t,k}^{cat}-\nabla f_{t,k}^{cat}\right\|\right)^2\right]\\
&+4\left(\mathbb{E}\left[\left\|\mathbf{W}^k-\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\|^2\left\|\nabla f_{t,1}^{cat}-\widetilde{\nabla} f_{t,1}^{cat}\right\|^2\right]+\mathbb{E}\left[\left\|\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\|^2\left\|\nabla f_{t,k}^{cat}-\widetilde{\nabla} f_{t,k}^{cat}\right\|^2\right]\right)\\
\leq&4n\left(G+G_0\right)^2\left(\sum_{\tau=1}^{k-1}\lambda(\mathbf{W})^{k-\tau}\right)^2+4n\sigma_0^2\left(\lambda(\mathbf{W})^{2k}+1\right)\\
\leq&4n\left(G+G_0\right)^2\left(\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})}\right)^2+4n\sigma_0^2(\lambda(\mathbf{W})+1)\leq 4n\left[\left(\frac{G+G_0}{\frac{1}{\lambda(\mathbf{W})}-1}\right)^2+2\sigma_0^2\right]
\end{aligned}
\tag{4.14}
$$

where the first inequality holds since $\mathbb{E}\left[\nabla f_{t,k+1}^{cat}-\widetilde{\nabla} f_{t,k+1}^{cat}+\widetilde{\nabla} f_{t,k}^{cat}-\nabla f_{t,k}^{cat}\right]=0$. The second inequality follows the fact that $\|a+b\|^2\leq 4\left(\|a\|^2+\|b\|^2\right)$. The third inequality comes from Assumption 4.4.1 and the analysis in Lemma 3.3.2. Finally, one can obtain the desired result by noticing $\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i-\widetilde{\boldsymbol{d}}_{t,k}^i\right\|^2\right]\leq\sum_{i=1}^n\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^i-\widetilde{\boldsymbol{d}}_{t,k}^i\right\|^2\right]=\mathbb{E}\left[\left\|\boldsymbol{d}_{t,k}^{cat}-\widetilde{\boldsymbol{d}}_{t,k}^{cat}\right\|^2\right]$ □

Proposition 4.7.1 (Proposition 4.4.1)**.** *For $t\in[T]$, $i\in[n]$, it holds that,*

$$\|\boldsymbol{d}_{t,k+1}^i-\boldsymbol{d}_{t,k}^i\|\leq\frac{B}{(k+3)^\alpha}$$

where $B=9C_g+5\beta\left(4C_d+AD\right)$.

Proof. For $k\geq 2$, we have:

$$
\begin{aligned}
\left\|\boldsymbol{d}_{t,k}^i-\boldsymbol{d}_{t,k-1}^i\right\|&\leq\left\|\boldsymbol{d}_{t,k}^i-\nabla F_{t,k}\right\|+\left\|\nabla F_{t,k}-\nabla F_{t,k-1}\right\|+\left\|\nabla F_{t,k-1}-\boldsymbol{d}_{t,k-1}^i\right\|\\
&\leq\frac{C_g}{k^\alpha}+\frac{C_g}{(k-1)^\alpha}+\frac{1}{n}\sum_{i=1}^n\left\|\nabla f_t^i(\boldsymbol{x}_{t,k}^i)-\nabla f_t^i(\boldsymbol{x}_{t,k-1}^i)\right\|\\
&\leq\frac{C_g}{k^\alpha}+\frac{C_g}{(k-1)^\alpha}+\frac{\beta}{n}\sum_{i=1}^n\left\|\boldsymbol{x}_{t,k}^i-\boldsymbol{x}_{t,k-1}^i\right\|\\
&\leq\frac{C_g}{k^\alpha}+\frac{C_g}{(k-1)^\alpha}+\beta\frac{4C_d+AD}{(k-1)^\alpha}\\
&\leq\frac{4C_g}{(k+3)^\alpha}+\frac{5C_g}{(k+3)^\alpha}+\frac{5\beta\left(4C_d+AD\right)}{(k+3)^\alpha}\\
&\leq\frac{9C_g+5\beta\left(4C_d+AD\right)}{(k+3)^\alpha}
\end{aligned}
$$

where we triangle inequality, smoothness of f_t and Proposition 2.7.1, Lemmas 2.4.1 and 2.4.2 with an additional parameter α on the learning rate. □

Remark 1. *By lemma 4.4.2 and Jensen's inequality, we can deduce the following inequality*

$$\mathbb{E}\|\boldsymbol{d}_{t,k}^i-\widetilde{\boldsymbol{a}}_{t,k}^i\|\leq\sqrt{\mathbb{E}\|\boldsymbol{d}_{t,k}^i-\widetilde{\boldsymbol{a}}_{t,k}^i\|^2}\leq\frac{Q^{1/2}}{(k+4)^{1/4}}$$

5

Energy Consumption of Distributed Training

Contents

5.1 Introduction

Federated learning is a new machine learning paradigm that allows multiple clients to collaboratively train a model without the need to share data. This approach is particularly useful in applications where preserving privacy is required such as healthcare, financial services, or where the cost of transferring data is not affordable. In this setting, each client trains a machine-learning model on its private data and exchanges the parameters with a server at each communication round. The server orchestrates the process by selecting clients, setting configuration, and handling client aggregation. In contrast to traditional machine learning where only one machine learning model is trained in a centralized manner, we are facing a situation where a few to thousands of machines work collaboratively to train a model. This setting raises new challenges such as communication overhead, data/machine heterogeneity, and more importantly, in the context of climate change and sustainable AI, the energy consumption (EC) and environmental impact of the system.

The energy consumed by centralized learning (CL) has been well studied in the literature for many ML applications [Strubell2019, Henderson2020, Luccioni2023] and at the level of ML companies [Patterson2022, Wu2022]. The interest in the energy consumed by federated learning has been increasing, along with the rise of data coming from mobile device applications. For example [Qiu2023] studies the carbon emission of training FL model on embedded devices such as Jetson Xavier and Tegra while comparing with EC of centralized training for the same task. They show that the EC of FL is higher than centralized training due to the communication overhead, especially when training with only 1 local epoch. [Savazzi2022] show that tradeoffs can be made to make training less consuming at the edge, by adapting the number of rounds and the communication efficiency. The findings are based on energy models rather than on measurements. [Wu2022] compare the carbon emitted by centralized and federated learning for the Transformer model, including the additional energy consumed by communication. Their conclusion is not clear: it depends on the hardware used for CL and whether renewable energy fuels the data center. [Patterson2024] are more resolved. Studying the energy consumed by smartphones in the context of Google FL, they find that in one use case, smartphones require more than 12 times the energy centralized settings would have consumed on a similar task. They support this by estimating the PUE of smartphones to be around 3, based on charger efficiency and user charging behavior.

Existing literature focuses on comparing CL and FL when both settings are fundamentally different by the purposes and constraints. CL relies on hardware designed to be computationally efficient while edge devices need to limit their power and energy consumption. FL has to deal with distributed and non-IID data when CL operates on huge IID databases and its goal is training foundation models from scratch. Additionally, there is a lack of recommendations on how to reduce the energy consumed by FL. The particular settings of FL suggest that new opportunities can be found to reduce the energy consumed. The energy models proposed in the literature don't take into account the joint dependency of parameters on the training and the energy consumed. For example, fewer local epochs reduce the energy consumed by the client but required more rounds to reach desired target. Moreover, there is no existing study on the impact of dataset size and the optimizer on the total energy consumption.

This particular setting raises the following questions :

- *What is the impact of FL hyperparameters choices on the energy consumption of the system?*

The federated learning setting often involves an interdisciplinary approach where machine learning, distributed systems, algorithm design, and hardware are combined. This interdisciplinary approach creates a challenging environment when it comes to EC measurement as it depends on the hardware used for training, the communication network, and the choice of algorithm. For example, training a model GPU is more energy efficient than training on a CPU, or training on embedded devices GPU such as Jetson Xavier is even more efficient than traditional GPU [Lacoste2019]. In the context of FL, the communication overhead is a major factor that can impact the EC of the system, the communication at the server side increases with the number of clients might leads to higher EC on server side due to the amount of data to be processed and communicated with clients. On the otherhand, number of communication rounds between server and clients is an important factor to be considered as balancing between computation/communication on clients sides.

In this chapter, we aim to answer the above question in two steps : 1) measuring the energy consumption of each client in FL setting under highly heterogenous data using multiple devices connected through a network, and study the impact of FL parameters on the EC of each clients

and the total EC of the system; 2) studying the EC variation when increasing the number of active clients in the training process, which implicitly increase the variance to the global model and the communication overhead.

5.1.1 Federated Learning

We consider n clients $[1, \ldots, n]$ and a central parameter server (PS) acts as a coordinator of the training process. Each client $i \in [n]$ has a local dataset D_i following a distribution $\mathcal{D}_i$ such that $\mathcal{D}_i \neq \mathcal{D}_j$ for $i \neq j$ i.e heterogenous data distribution. The goal of the federated learning is to learn a global model by minimizing the global loss function $F(\boldsymbol{x})$ defined as :

$$\min_{\boldsymbol{x}} F(\boldsymbol{x}) := \sum_{i=1}^{n} w_i f_i(\boldsymbol{x})$$

where f_i is local loss function on each client, defined as $f_i(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z} \sim \mathcal{D}_i}[\ell(\boldsymbol{x}, \boldsymbol{z})]$ and w_i is the weight of client i that can be set based on the number of samples in the local dataset. At each communication round $t \in [T]$, a set of selected clients $\mathcal{S}_t$ is chosen to participate in the training process and receive the global parameter $\boldsymbol{x}_t$. Each client $i \in \mathcal{S}_t$ updates its parameters $\boldsymbol{x}_{t+1}^i$ by minimizing its local loss function f_i using an local optimizer Opt and send back the updated parameters to the PS. The process is repeated for T communication rounds. The PS updates the global parameter $\boldsymbol{x}_t$ using the local parameters received from clients as follows :

$$\boldsymbol{x}_{t+1} = \mathcal{A}\left(\{w_i, \boldsymbol{x}_{t+1}^i\}_{i=1}^{|\mathcal{S}_t|}\right)$$

where $\mathcal{A}$ is an aggregation methods that differs by the chosen strategy. The detailed pseudo algorithm is described in Algorithm 11.

Algorithm 11 Federated Learning

Initialize $\boldsymbol{x}_0$
for each communication round $t = 1, \ldots, T$ **do**
  Server select a set of clients $\mathcal{S}_t$
  Server broadcast $\boldsymbol{x}_t$ to all clients in $\mathcal{S}_t$
  for each client $i \in \mathcal{S}_t$ **do**
    $\boldsymbol{x}_{t,0}^i = \boldsymbol{x}_t$
    for each local epoch $e = 0, \ldots, E-1$ **do**
      $\boldsymbol{x}_{t,e+1}^i = Opt(f_i, \boldsymbol{x}_{t,e}^i)$
    end for
    $\boldsymbol{x}_{t+1}^i = \boldsymbol{x}_{t,E}^i$
  end for
  $\boldsymbol{x}_{t+1} = \mathcal{A}\left(\{w_i, \boldsymbol{x}_{t+1}^i\}_{i=1}^{|\mathcal{S}_t|}\right)$
end for

The setting of FL encompasses a few important choices such as the number of clients participate on each round of training $|\mathcal{S}_t|$, the local epochs E, the heterogeneity of clients which influence the choice of aggreation methods $\mathcal{A}$ and also the local optimizer Opt. The choice of these parameters will have an impact on the convergence of the model, the communication overhead and also the energy consumption of the system. In the next few sections, we will study in depth the impact of these settings on the training of FL and see how it can impact the overall EC of the system.

5.2 Experiment Setting

We executed all experiments on nodes from the Estats cluster of the large-scale test beds for experimental research called Grid'5000 [Balouek2013]. This cluster was selected because its nodes have similar computational capabilities as embedded devices specialized for AI training.

The nodes in this cluster have the following specifications:

- System model: Nvidia Jetson AGX Xavier
- CPU: 1 Nvidia Carmel (Carmel), aarch64, 8 cores
- GPU: NVIDIA GV10B, Volta architecture
- Memory: 32 GiB
- TDP: 30W

For all experiments, we used Ubuntu 20.04 as available on the Grid'5000 testbed. In order to increase consumption stability and the consistency of our results, we have set the CPU frequency to the maximum supported. We also installed an Nvidia GPU driver when relevant, with default power management configuration. Nvidia processors are equipped with power meters that monitor the instant power consumed by the GPU, the CPU, and the memory. At the beginning of each experiment, we launch the Jetson-stats application[1] to monitor the CPU and GPU power and usage of each host. The acquisition frequency is 1Hz. We wait 30 seconds between each experiment to make sure that the hosts have the time to cool down and that the temperature doesn't impact the power. We also record the time taken to complete the training process.

We utilize the Flower framework to manage client-server communication and other federated learning (FL) related configurations in our experiments. We measure the energy consumption of training various FL algorithms, including FedAvg [McMahan2023] and adaptive methods such as FedAdam, FedYogi, and FedAdaGrad [Reddi2021]. Stochastic gradient descent is employed as the client optimizer (*Opt*) for these methods. Additionally, we consider also FedAvg with other types of *Opt* including Stochastic Frank-Wolfe (SFW) [Hazan2016b] and Adam [Kingma2015].

We consider the CIFAR-10 dataset which is divided into client shards using a Dirichlet distribution with $\alpha = 0.5$. Each client has access to one shard, and we vary the number of splits to be 10, 20, 30, 50, or 100, depending on the configuration. The model architecture is ResNet-18 where BatchNorm layer is replaced with a GroupNorm layer of 32 groups to make it more suitable for the FL setting. Cross-entropy is used as the loss function and we maintain a batch size of 32. We divide the client data into an 80-20 train-validation split. The client learning rate is set to 0.0316 with momentum to 0.9 for all client optimizers except Adam. For adaptive strategies, we set the server learning rate to 0.01.

For FedAvg-SFW, an L_2 norm ball with a diameter of 300 is used as the constraint [Pokutta2020]. For FedAvg-Adam, we set β_1 and β_2 to 0.9 and 0.999, respectively. We repeat each experiment five times and report the average results.For all experiments, we set the target accuracy to 0.75. The table below summarizes the common settings for all strategies.

α	Batch Size	Client LR	Server LR	#Groups	Momentum
0.5	32	0.0316	0.01	32	0.9

Tableau 5.1: Common parameter setting for all strategies

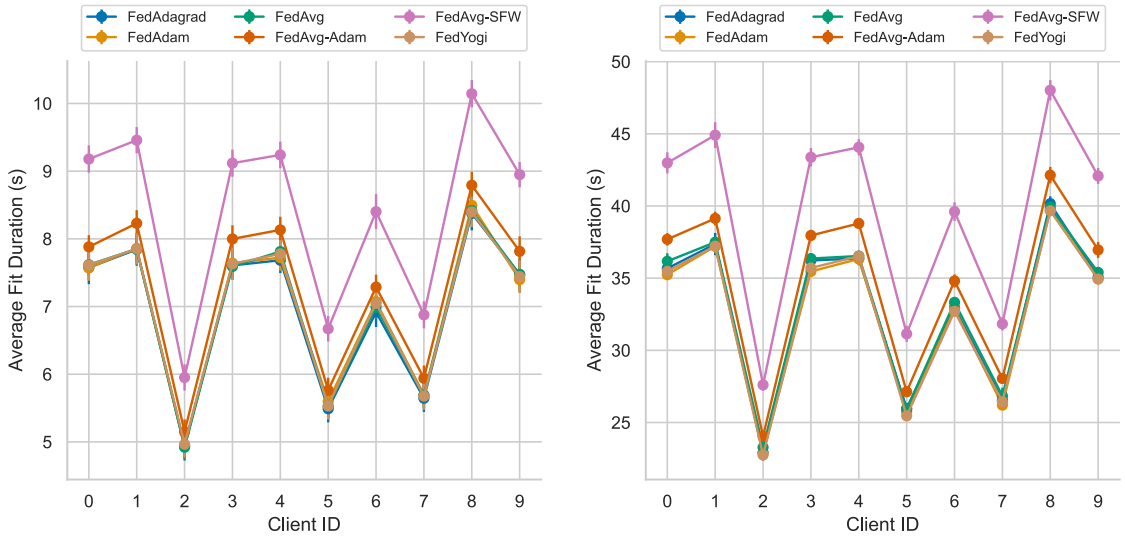
5.2.1 Full Client Participation

In this section, we present the results of training FL algorithms when the data is split into 10 shards following the Dirichlet distribution, as described in the previous section. We consider the scenario of full client participation, i.e., $|\mathcal{S}_t| = 10, \forall t \in [T]$. For each algorithm, experiments are conducted for 1 and 5 local epochs, measuring the time taken to complete the training process and the energy consumption of each client host. For each communication round, "fit time" refers to the time taken to complete client local training on its dataset, and "fit energy" refers to the energy consumed during this process. These metrics are measured individually for each client.

Figures 5.1a and 5.1b show the average fit time across clients for all algorithms for 1 and 5 local epochs, respectively. The average fit time is not uniform among clients, even though each client has the same model architecture and optimization settings. This is due to the heterogeneous

[1] `https://rnext.it/jetson_stats/reference/jtop.html#jtop.jtop.power`

nature of the data split among clients, not only in terms of label distribution but also in the number of samples, causing variations in the number of client optimization steps needed to complete one local epoch. From these graphs, it is evident that the server aggregation strategy has almost no impact on the client fit time since strategies using SGD as the local optimizer, such as FedAvg and adaptive strategies, show the same average fit time. However, replacing SGD with Adam or SFW in FedAvg results leads to an increase in the average fit time. This is because Adam and SFW involve more computational steps than SGD to complete one optimization step. Comparing the two graphs, it is clear that the average fit time for 5 local epochs is approximately 5 times higher than for 1 local epoch, which is expected since the number of optimization steps is 5 times greater.



(a) Fit duration of each client of 1 local epoch

(b) Fit duration of each client of 5 local epoch

Figure 5.1: Fit duration of each client for different local epochs

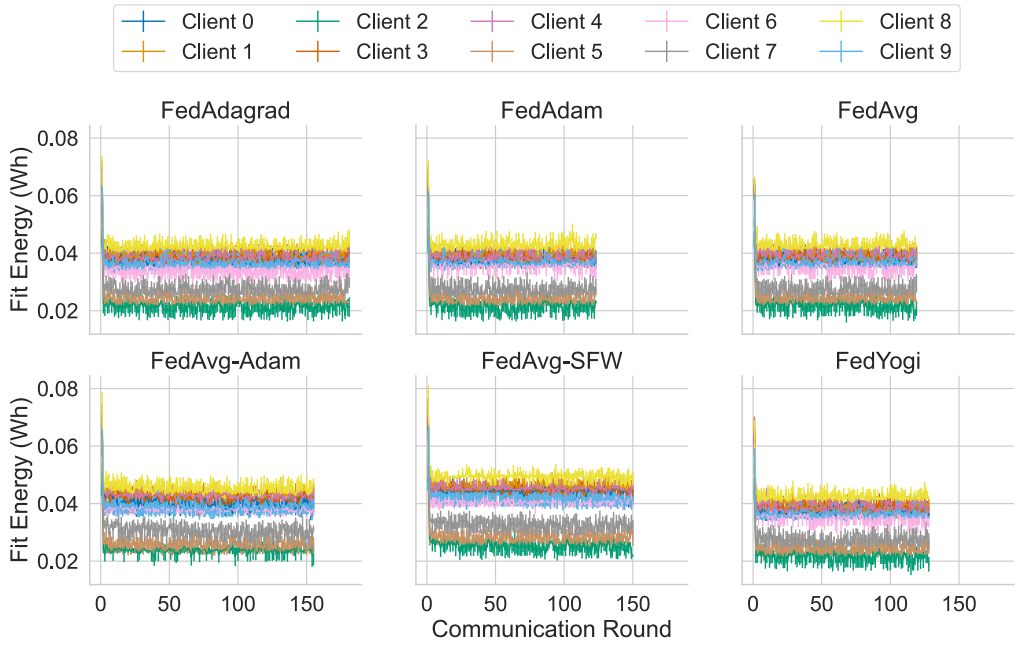


Figure 5.2: Evolution of the energy consumed during training for 1 local epoch

Figures 5.2 and 5.3 show the fit energy for 1 and 5 local epochs throughout the training process, respectively. Our first observation is that the variation in fit energy between clients correlates with

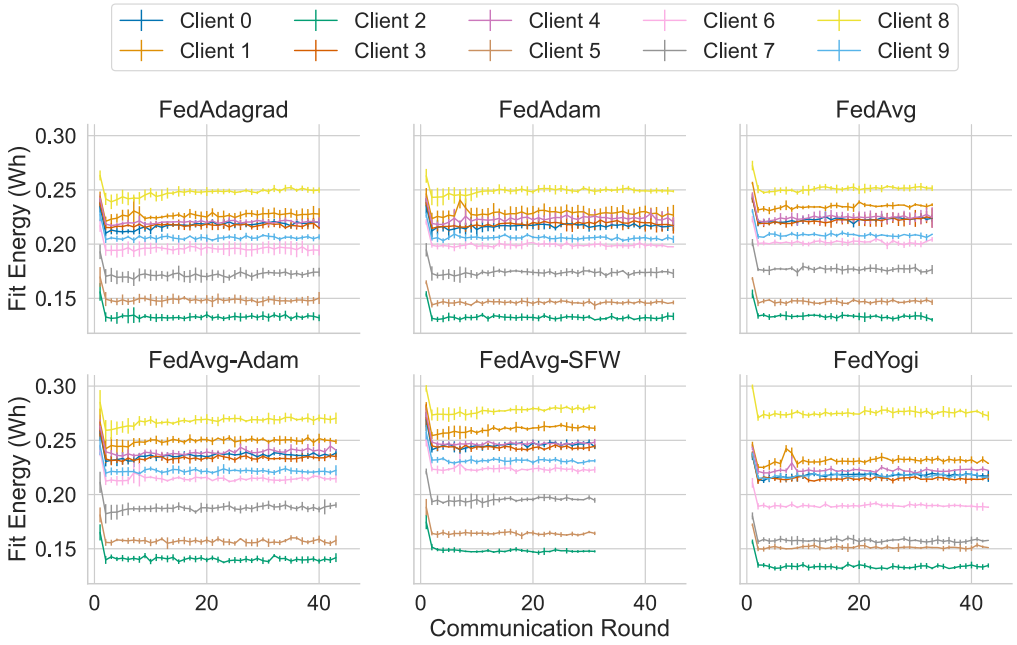


Figure 5.3: Evolution of the energy consumed during training for 5 local epochs

the time required to complete the local training in Figure 5.1, as the energy consumption is directly proportional to the time taken to complete the training process. We also observe that the energy consumption is uniform across communication rounds for each client. From these two figures, we can see a significant impact of the server aggregation method and the number of local epochs on the time to convergence. For 1 epoch, FedAdagrad takes the most rounds to reach the desired accuracy, whereas FedAvg is the fastest. In terms of energy consumption, FedAvg and other adaptive strategies have similar values as expected since they use the same client optimizer. When using other local optimizers with FedAvg, there is an increase in energy consumption, especially for SFW, which consumes the most energy per round of training. The results differ between 1 and 5 local epochs in terms of round to convergence. FedAdam and FedAvg-Adam are the slowest to reach the desired accuracy. Moreover, we observe an anormal increase in energy consumption with FedYogi at client 8 in compares to other strategy using SGD as local optimizer. This requires further investigation to understand the reason behind this.

Table 5.2 presents a summary of the training for each strategy for 1 and 5 local epochs, along with the corresponding measures. The columns "Server" and "Client" represent the total energy consumed by the server and clients, respectively whereas "Total" is the sum of server and client energy. We also report the total client fit energy and the average fit energy for each strategy, as well as the total time in minutes to reach the desired accuracy. It is observed that training with 5 local epochs is more energy efficient for FedAdagrad and FedAvg, showing a clear reduction in total energy consumption, whereas other strategies exhibit a slight increase in energy consumption when switching from 1 to 5 local epochs. For the FedAvg strategy, using SGD as the local optimizer is more energy efficient than using Adam or SFW. Interestingly, for 1 local epoch, FedAvg with Adam and SFW shows relatively high energy consumption compared to FedAvg with SGD, despite having the same server computation. We suggest this is due to the time taken by each strategy to achieve the desired accuracy, which is faster for FedAvg with SGD than with Adam or SFW. This observation requires further investigation to understand the underlying reasons.

In conclusion, both the number of local epochs and the choice of optimizer has a signficant impact on the energy consumption of FL training. Increasing the number of local epochs multiplies the energy consumed proportionnaly to the increase but it also reduces the number of training rounds needed to reach to targeted accuracy since each round is more effective in terms of learining. The optimizer is less impactful in regards of round energy consumption since the additional computation per round is not signficiant. But it has a strong relation to the number of training round thus impacting the total energy consumed by training.

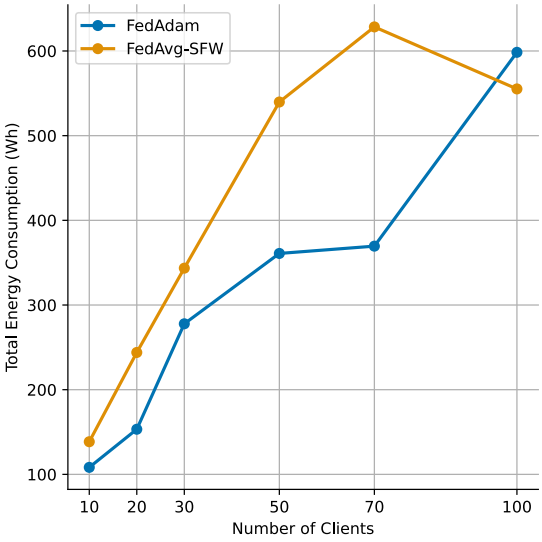
Strategy	Epoch	Round	Time (min)	Energy (Wh)				
				Server	Client	Total	Total Fit	Average Fit
FedAdagrad	1	181	76.87	13.15	141.56	154.71	61.69	0.0340
	5	40	38.26	5.34	102.94	108.27	79.51	0.1987
FedAdam	1	123	52.27	9.00	96.95	105.96	42.27	0.0343
	5	45	42.31	5.91	115.44	121.35	89.85	0.1996
FedAvg	1	119	50.23	6.90	93.65	100.55	41.05	0.0344
	5	33	30.91	4.35	85.20	89.55	66.90	0.2027
FedYogi	1	128	54.81	9.41	100.62	110.04	43.81	0.0342
	5	43	40.32	5.62	110.80	116.42	86.64	0.2014
FedAvg-Adam	1	155	66.53	11.42	126.52	137.94	57.62	0.0371
	5	43	42.62	5.93	119.15	125.08	92.52	0.2151
FedAvg-SFW	1	150	68.29	11.46	126.59	138.06	60.00	0.0400
	5	31	33.94	4.64	87.63	92.26	69.54	0.2243

Tableau 5.2: Training summary of each strategy for 1 and 5 local epochs. Red color indicates the lowest value of corresponding columns for 5 local epochs. Blue color indicates the lowest value of corresponding columns for 1 local epoch.

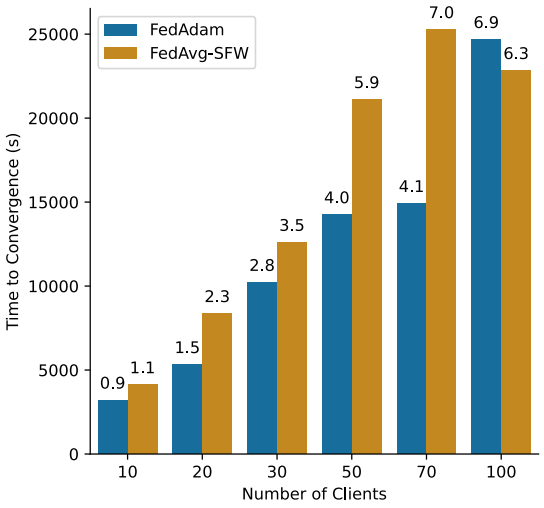
5.2.2 Increased Active Clients

In this section, we examine the energy evolution as the number of active clients increases during the training process. The data is split into multiple shards in a non-IID manner, with the number of shards being 10, 20, 30, 50, 70, and 100, corresponding to the number of active clients. It is important to note that this setup reduces the data size on each client, thus decreasing computation time but, conversely, increases the number of communication rounds required to achieve the desired accuracy. To ensure fair measurement for each participant, we maintain the number of clients participating in each communication round $|\mathcal{S}_t|$ at 10. This means that for large pools of active clients, we randomly assign a client to one of the hosts so that each host handles only one client per round. We keep the local epoch to 1 and consider two strategies: FedAvg-SFW and FedAdam, using the same settings as in the previous section. Figure 5.4 reports the total energy consumption and the time to convergence for different numbers of active clients. Recall that the total energy consumption is the sum of the energy consumed by both the clients and the server during the training process. We observe in fig. 5.4a that the total energy increases with the number of active clients for both strategies, except for FedAvg-SFW, where there is a drop in energy between 70 and 100 active clients. This drop in energy is due to the total training time for 70 active clients being higher than for 100 active clients in the case of FedAvg-SFW (Figure 5.4b). We also observe that the total energy consumption for FedAvg-SFW is higher than for FedAdam except for 100 active clients. This aligns with the results from the previous section when full client participation was considered. An explanation for the behavior of FedAvg-SFW between 70 and 100 active clients is the randomness of client selection in each round.

In fact, client selection is done randomly for both fitting and evaluation at each communication round. When running multiple experiments with the same settings, we observe that this random process significantly impacts the time to convergence of FL. The algorithm may fit on one subset of clients and evaluate on another subset with large gradient dissimilarity from the fitted set, leading to slow convergence as more sampling is required. This behavior is even more pronounced as the number of active clients increases, as the global model encounters higher variance due to noisy client stochastic updates. This observation is confirmed in Figure 5.5, where we report the accuracy curves of FedAdam (fig. 5.5a) and FedAvg-SFW (fig. 5.5b) for all numbers of active clients. We can see that with a higher number of active clients, the noisier the curve becomes.

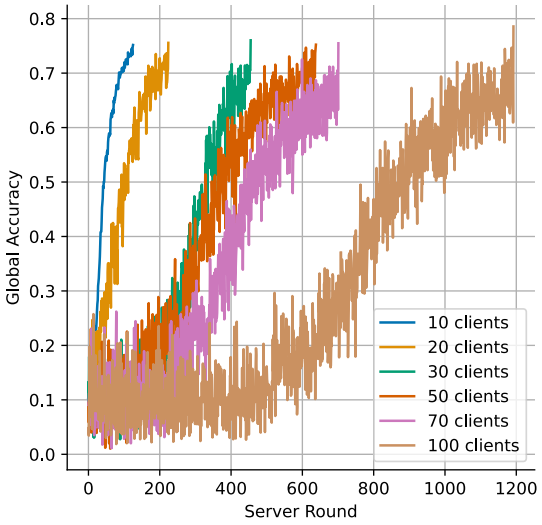


(a) Total training energy consumption in Watt-hour (y-axis) for each pool of active clients

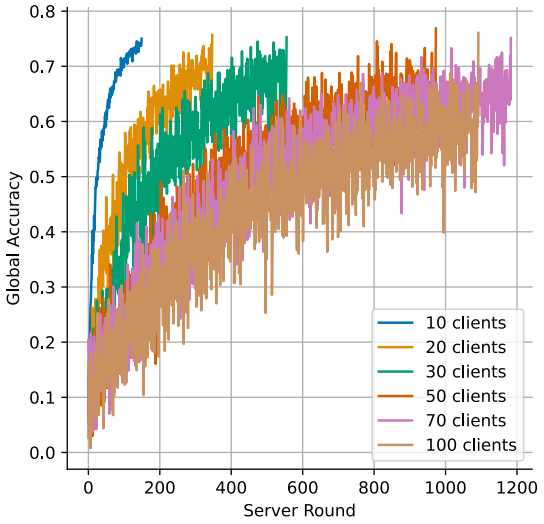


(b) Time to convergence in seconds (y-axis) for each pool of active clients. Value above the bar indicates time in hours.

Figure 5.4: Energy consumption (left) and time to convergence (right) for different number of active clients.



(a) Fit duration of each client of 1 local epoch



(b) Fit duration of each client of 5 local epoch

Figure 5.5: Fit duration of each client for different local epochs

5.3 Concluding Remarks

In this research, we investigate the energy usage of Federated Learning (FL) algorithms on edge devices. We showed that the energy consumption of FL algorithms is significantly influenced by the choice of hyperparameters, such as the number of local epochs, the local optimizer, and the aggregation strategy. Additionally, we discovered that client selection is crucial to the overall energy consumption of the system, as it can prolong the algorithm's convergence time, due to the unpredictability in the client selection process. We also observed that the server's energy consumption is negligible compared to that of the clients. For future work, we aim to explore the communication costs of FL algorithms on each client and the server. We also intend to broaden our study to include different datasets and various edge device configurations.

Conclusion

In this thesis, we have introduced a series of algorithms that address the challenges of distributed online optimization problems, specifically tailored for edge devices with limited computational resources. We have examined various settings of online optimization problems, including online convex optimization with adversarial delayed feedback, online distributed optimization for convex and monotone submodular functions, and online non-convex optimization in a distributed setting.

For adversarial delayed feedback, our algorithms achieve the optimal regret bound of $O(\sqrt{dT})$ in both centralized and distributed settings with bounded delay. Experimental results show that our algorithms outperform existing solutions in terms of regret. Potential future directions include adapting these algorithms to stochastic gradients with variance reduction to make them more practical for real-world applications and addressing communication delays in distributed settings.

To reduce the communication cost in online distributed setting of projection-free algorithm, we proposed an approach that only need one gradient query at each round, thus reducing the communication complexity to $O(T)$. The algorithm achieves regret and $(1-\frac{1}{e})$-regret bounds of $O(T^{4/5})$. We also extend the algorithm to bandit setting while ensuring a $(1-\frac{1}{e})$-regret bound of $O(T^{8/9})$ for DR-Submodular functions. We provided a detailed analysis for scenarios where the constraint set is either a general convex set or a downward-closed convex set, under full information and bandit settings, respectively. Experimental results on a real-life movie recommendation problem highlight the efficacy of the proposed algorithm for learning in decentralized settings.

Additionally, we presented an online algorithm aimed at minimizing non-convex loss functions aggregated from local data distributed across a network. We introduced the convergence gap, a generalized version of the Frank-Wolfe gap in the online setting, demonstrating convergence rates of $O(T^{-1/2})$ and $O(T^{-1/4})$ for exact and stochastic gradient settings, respectively. Our experiments using a real-life smart building dataset validate the theoretical analysis and underscore the value of our approach for learning in distributed settings. Future work could investigate convergence to a stationary point of each local function F_t by analyzing the duality gap and examining function variations, potentially leading to a more comprehensive understanding of the algorithm's performance in online non-convex optimization problems.

Lastly, we study the energy consumption of distributed learning algorithms on edge devices. We highlight the important role of hyperparameters, such as local epochs, local optimizers and client sampling methods in the energy consumption of federated learning algorithms. Overall, our research demonstrates significant advancements in the development of projection-free algorithms for various online optimization problems, providing valuable insights and setting the stage for future explorations in this field.

A
Useful Results

A.1 Inequalities

Proposition 1 (Jensen's Inequality). *Let f be a convex function defined over a convex set $\mathcal{K}$, and let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathcal{K}$. Then for any positive weights $w_1, w_2, \ldots, w_n$ (i.e., $w_i > 0$ for all i and $\sum_{i=1}^n w_i = 1$), we have*

$$f\left(\sum_{i=1}^n w_i \boldsymbol{x}_i\right) \leq \sum_{i=1}^n w_i f(\boldsymbol{x}_i).$$

Proposition 2 (Cauchy Schwartz Inequality). *For any vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$, we have*

$$|\langle \boldsymbol{a}, \boldsymbol{b} \rangle| \leq \|\boldsymbol{a}\| \, \|\boldsymbol{b}\| .$$

Proposition 3 (Young's Inequality). *For vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ and for all positive real numbers p, q such that $\frac{1}{p} + \frac{1}{q} = 1$, we have:*

$$|\langle \boldsymbol{a}, \boldsymbol{b} \rangle| \leq \frac{\|\boldsymbol{a}\|^p}{p} + \frac{\|\boldsymbol{b}\|^q}{q}$$

Proposition 4 ([Koloskova2019], Lemma 16). *Let $\mathbf{W}$ be a stochastic matrix and denote by $\lambda(\mathbf{W})$ its second largest eigenvalue. Then, we have*

$$\left\| \mathbf{W}^n - \frac{1}{n} \mathbf{1}\mathbf{1}^T \right\| \leq \lambda(\mathbf{W})^n.$$

Proof. As $\mathbf{W}$ is a stochastic matrix, the first eigenvector associated with eigenvalue 1 is written as $\mathbf{u}_1 = \frac{1}{\sqrt{n}}\mathbf{1}$. Using SVD decomposition $\mathbf{W} = \mathbf{U}\Sigma\mathbf{U}^T$, we have

$$\begin{aligned}
&\left\| \mathbf{W}^n - \frac{1}{n} \mathbf{1}\mathbf{1}^T \right\| = \left\| \mathbf{U}\Sigma^n\mathbf{U}^T - \frac{1}{n} \mathbf{1}\mathbf{1}^T \right\| = \left\| \mathbf{U}\Sigma^n\mathbf{U}^T - \mathbf{u}_1\mathbf{u}_1^T \right\| \\
&= \left\| \mathbf{U}\Sigma^n\mathbf{U}^T - \mathbf{U} \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \mathbf{U}^T \right\| = \left\| \Sigma^n - \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \right\| = \lambda(\mathbf{W})^n.
\end{aligned}$$

□

A.2 Lemmas

Proposition 5. *Let $\mathbf{W}$ be a stochastic matrix and denote by $\lambda(\mathbf{W})$ its second largest eigenvalue. We call $\boldsymbol{d}_{t,k}^{cat} = \left[\boldsymbol{d}_{t,k}^{1\top}, \ldots, \boldsymbol{d}_{t,k}^{n\top}\right]^\top \in \mathbb{R}^{dn}$ the concatenation of local average gradient updates at round t and sub-iteration k. Then, for all $t \in [T], k \in [K]$, we have*

$$\begin{aligned}
\boldsymbol{d}_{t,k}^{cat} = &\sum_{\tau=1}^{k-1} \left[\left[\left(\mathbf{W}^{k-\tau} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T \right) \otimes I_d \right] \left(\nabla f_{t,\tau+1}^{cat} - \nabla f_{t,\tau}^{cat} \right) \right] \\
&+ \left[\left(\mathbf{W}^k - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T \right) \otimes I_d \right] \nabla f_{t,1}^{cat} + \left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T \otimes I_d \right) \nabla f_{t,k}^{cat}
\end{aligned}$$

By taking the norm on both side of the equation, we have

$$\left\| \boldsymbol{d}_{t,k}^{cat} \right\| \leq 2\sqrt{n}G \left(\frac{\lambda(\mathbf{W})}{1 - \lambda(\mathbf{W})} + 1 \right)$$

Proof.

$$
\begin{aligned}
&\boldsymbol{d}_{t,k}^{cat} = (\mathbf{W} \otimes I_d)\left(\nabla f_{t,k}^{cat} - \nabla f_{t,k-1}^{cat} + \boldsymbol{d}_{t,k-1}^{cat}\right) \\
&= (\mathbf{W} \otimes I_d)\left(\nabla f_{t,k}^{cat} - \nabla f_{t,k-1}^{cat}\right) + (\mathbf{W} \otimes I_d)^2\left(\nabla f_{t,k-1}^{cat} - \nabla f_{t,k-2}^{cat} + \boldsymbol{d}_{t,k-2}^{cat}\right) \\
&= \sum_{\tau=1}^{k-1}\left[(\mathbf{W} \otimes I_d)^{k-\tau}\left(\nabla f_{t,\tau+1}^{cat} - \nabla f_{t,\tau}^{cat}\right)\right] + (\mathbf{W} \otimes I_d)^k \nabla f_{t,1}^{cat} \qquad \text{(A.1)} \\
&= \sum_{\tau=1}^{k-1}\left[(\mathbf{W} \otimes I_d)^{k-\tau}\left(\nabla f_{t,\tau+1}^{cat} - \nabla f_{t,\tau}^{cat}\right)\right] + (\mathbf{W} \otimes I_d)^k \nabla f_{t,1}^{cat} - \sum_{\tau=1}^{k-1}\left[\nabla F_{t,\tau+1}^{cat} - \nabla F_{t,\tau}^{cat}\right] - \nabla F_{t,1}^{cat} + \nabla F_{t,k}^{cat} \\
&= \sum_{\tau=1}^{k-1}\left[\left[\left(\mathbf{W}^{k-\tau} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right) \otimes I_d\right]\left(\nabla f_{t,\tau+1}^{cat} - \nabla f_{t,\tau}^{cat}\right)\right] + \left[\left(\mathbf{W}^k - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right) \otimes I_d\right]\nabla f_{t,1}^{cat} + \left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T \otimes I_d\right)\nabla f_{t,k}^{cat}
\end{aligned}
$$

where the fourth equality holds since $\nabla F_{\sigma_q(k)}^{cat} - \sum_{\tau=1}^{k-1}\left(\nabla F_{\sigma_q(\tau+1)}^{cat} - \nabla F_{\sigma_q(\tau)}^{cat}\right) - \nabla F_{\sigma_q(1)}^{cat} = 0$. The fifth equality can be deduced using $\nabla F_{\sigma_q(k)}^{cat} = \left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T \otimes I_d\right)\nabla f_{\sigma_q(k)}^{cat}$ and $(\mathbf{W} \otimes I_d)^k = \left(\mathbf{W}^k \otimes I_d\right)$. Recall that $\|\mathbf{W} \otimes I_d\| = \|\mathbf{W}\|$. Taking the norm on equation (A.1), we have

$$
\left\|\boldsymbol{d}_{q,k}^{cat}\right\| \leq 2\sqrt{n}G\sum_{\tau=1}^{k-1}\lambda(\mathbf{W})^{k-\tau} + \sqrt{n}G\left(\lambda(\mathbf{W})^k + 1\right) \leq 2\sqrt{n}G\left(\frac{\lambda(\mathbf{W})}{1-\lambda(\mathbf{W})} + 1\right)
$$

where we have used $\left\|\nabla f_{\sigma_q(\tau+1)}^{cat} - \nabla f_{\sigma_q(\tau)}^{cat}\right\| \leq 2\sqrt{n}G$, $\left\|\mathbf{W}^k - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\| \leq \lambda(\mathbf{W})^k$ and $\left\|\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right\| \leq 1$ in the first inequality. □

Proposition 6. *For all the algorithms presented in this thesis, the following bounds hold for all $t \in [T], k \in [K]$ and $i \in [n]$:*

$$
\left\|\overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}_{t,k}^i\right\| \leq \frac{2C_d}{k}
$$

$$
\left\|\boldsymbol{x}_{t,k+1}^i - \boldsymbol{x}_{t,k}^i\right\| \leq \frac{4C_d + AD}{k}
$$

Proof. For the first bound, recall the definition of FW-update in Algorithm 6 and using Lemma 2.7.2, we have

$$
\begin{aligned}
\left\|\overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}_{t,k}^i\right\| &= \left\|(1-\eta_{k-1})\left(\overline{\boldsymbol{x}}_{t,k-1} - \boldsymbol{y}_{t,k-1}^i\right) + \eta_{k-1}\left(\overline{\boldsymbol{v}}_{t,k-1} - \boldsymbol{v}_{t,k-1}^i\right)\right\| \\
&\leq \frac{C_d}{k-1} - \frac{AC_d}{(k-1)^2} + \frac{AD}{k-1} \leq \frac{C_d}{k-1} - \left[\frac{AC_d}{(k-1)^2} - \frac{AD}{k-1}\right] \\
&\leq \frac{C_d}{k-1} - \left[\frac{AC_d - AD}{(k-1)^2}\right] \leq \frac{C_d}{k-1} \leq \frac{2C_d}{k}
\end{aligned}
$$

Applying the first bound on the second one yields

$$
\begin{aligned}
\left\|\boldsymbol{x}_{t,k+1}^i - \boldsymbol{x}_{t,k}^i\right\| &\leq \left\|\boldsymbol{x}_{t,k+1}^i - \overline{\boldsymbol{x}}_{t,k+1}\right\| + \left\|\overline{\boldsymbol{x}}_{t,k+1} - \overline{\boldsymbol{x}}_{t,k}\right\| + \left\|\overline{\boldsymbol{x}}_{t,k} - \boldsymbol{x}_{t,k}^i\right\| \\
&\leq \frac{2C_d}{k+1} + \frac{AD}{k} + \frac{2C_d}{k} \\
&\leq \frac{4C_d + AD}{k}
\end{aligned}
$$

□

Proposition 7. *For every $t \in [T], k \in [K]$, it holds that*

$$
\overline{\boldsymbol{x}}_{t,k+1} - \overline{\boldsymbol{x}}_{t,k} = \eta_k\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{v}_{t,k}^i - \overline{\boldsymbol{x}}_{t,k}\right)
$$

Proof. Using definition of $\boldsymbol{y}_{t,k}^i$ and $\boldsymbol{x}_{t,k+1}^i$, we have

$$\overline{\boldsymbol{x}}_{t,k+1} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_{t,k+1}^i = \frac{1}{n}\sum_{i=1}^n \left((1-\eta_k)\,\boldsymbol{y}_{t,k}^i + \eta_k \boldsymbol{v}_{t,k}^i\right) = \frac{1}{n}\sum_{i=1}^n \left[(1-\eta_k)\left(\sum_{j=1}^n w_{ij}\boldsymbol{x}_{t,k}^j\right) + \eta_k \boldsymbol{v}_{t,k}^i\right]$$

$$= (1-\eta_k)\,\frac{1}{n}\sum_{i=1}^n \left[\sum_{j=1}^n w_{ij}\boldsymbol{x}_{t,k}^j\right] + \frac{1}{n}\eta_k \sum_{i=1}^n \boldsymbol{v}_{t,k}^i = (1-\eta_k)\,\frac{1}{n}\sum_{j=1}^n \left[\boldsymbol{x}_{t,k}^j \sum_{i=1}^n w_{ij}\right] + \frac{1}{n}\eta_k \sum_{i=1}^n \boldsymbol{v}_{t,k}^i$$

$$= (1-\eta_k)\,\frac{1}{n}\sum_{j=1}^n \boldsymbol{x}_{t,k}^j + \frac{1}{n}\eta_k \sum_{i=1}^n \boldsymbol{v}_{t,k}^i = (1-\eta_k)\,\overline{\boldsymbol{x}}_{t,k} + \frac{1}{n}\eta_k \sum_{i=1}^n \boldsymbol{v}_{t,k}^i$$

$$= \overline{\boldsymbol{x}}_{t,k} + \eta_k \left(\frac{1}{n}\sum_{i=1}^n \boldsymbol{v}_{t,k}^i - \overline{\boldsymbol{x}}_{t,k}\right)$$

where we use the doubly stochastic property of $\mathbf{W}$ i.e $\forall j \in [n], \sum_{i=1}^n w_{ij} = 1$. □

Bibliography

[Abdel-Aziz2019] Hamzah Abdel-Aziz et Xenofon Koutsoukos. *Data-driven online learning and reachability analysis of stochastic hybrid systems for smart buildings.* Cyber-Physical Systems, vol. 5, no. 1, pages 41–64, 2019.

[Abernethy2008] Jacob D. Abernethy, Elad Hazan et Alexander Rakhlin. *Competing in the Dark: An Efficient Algorithm for Bandit Linear Optimization.* In Proc. 21st Annual Conference on Learning Theory (COLT), pages 263–274, 2008.

[Balouek2013] Daniel Balouek, Alexandra Carpen Amarie, Ghislain Charrier, Frédéric Desprez, Emmanuel Jeannot, Emmanuel Jeanvoine, Adrien Lèbre, David Margery, Nicolas Niclausse, Lucas Nussbaum, Olivier Richard, Christian Perez, Flavien Quesnel, Cyril Rohr et Luc Sarzyniec. *Adding Virtualization Capabilities to the Grid'5000 Testbed.* In Ivan I. Ivanov, Marten van Sinderen, Frank Leymann et Tony Shan, editeurs, Cloud Computing and Services Science, pages 3–20, Cham, 2013. Springer International Publishing.

[Bian2017] Andrew An Bian, Baharan Mirzasoleiman, Joachim Buhmann et Andreas Krause. *Guaranteed Non-convex Optimization: Submodular Maximization over Continuous Domains.* In Artificial Intelligence and Statistics, pages 111–120, 2017.

[Cai2019] Mengmeng Cai, Manisa Pipattanasomporn et Saifur Rahman. *Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques.* Applied energy, vol. 236, pages 1078–1088, 2019.

[Calinescu2011] Gruia Calinescu, Chandra Chekuri, Martin Pál et Jan Vondrák. *Maximizing a monotone submodular function subject to a matroid constraint.* SIAM Journal on Computing, vol. 40, no. 6, pages 1740–1766, 2011.

[Cao2021] Xuanyu Cao, Junshan Zhang et H. Vincent Poor. *Constrained Online Convex Optimization With Feedback Delays.* IEEE Transactions on Automatic Control, vol. 66, no. 11, pages 5049–5064, 2021.

[Cao2022] Xuanyu Cao et Tamer Başar. *Decentralized Online Convex Optimization With Feedback Delays.* IEEE Transactions on Automatic Control, vol. 67, no. 6, pages 2889–2904, 2022.

[Chen2018a] Lin Chen, Christopher Harshaw, Hamed Hassani et Amin Karbasi. *Projection-Free Online Optimization with Stochastic Gradient: From Convexity to Submodularity.* In Proceedings of the 35th International Conference on Machine Learning, pages 814–823, 2018.

[Chen2018b] Lin Chen, Hamed Hassani et Amin Karbasi. *Online continuous submodular maximization.* In Proc. 21st International Conference on Artificial Intelligence and Statistics (AISTAT), 2018.

[Chen2020] Lin Chen, Mingrui Zhang, Hamed Hassani et Amin Karbasi. *Black Box Submodular Maximization: Discrete and Continuous Settings.* In The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy], Proceedings of Machine Learning Research, pages 1058–1070. PMLR, 2020.

[Chiliang2019] Zhang Chiliang, Hu Tao, Guan Yingda et Ye Zuochang. *Accelerating Convolutional Neural Networks with Dynamic Channel Pruning*. In 2019 Data Compression Conference (DCC), pages 563–563, 2019.

[Cover1967] Thomas M. Cover et Peter E. Hart. *Nearest neighbor pattern classification.* IEEE Trans. Inf. Theory, vol. 13, no. 1, pages 21–27, 1967.

[Deori2016] L. Deori, K. Margellos et M. Prandini. *On decentralized convex optimization in a multi-agent setting with separable constraints and its application to optimal charging of electric vehicles*. In IEEE Conference on Decision and Control (CDC), pages 6044–6049, 2016.

[Dhasade2024] Akash Dhasade, Anne-Marie Kermarrec, Tuan-Anh Nguyen, Rafael Pires et Martijn de Vos. *Harnessing Increased Client Participation with Cohort-Parallel Federated Learning*, 2024.

[Duchi2012] J. C. Duchi, A. Agarwal et M. J. Wainwright. *Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling*. IEEE Transactions on Automatic Control, vol. 57, no. 3, pages 592–606, 2012.

[Flaxman2005] Abraham D Flaxman, Adam Tauman Kalai et H Brendan McMahan. *Online convex optimization in the bandit setting: gradient descent without a gradient*. In Proc. 16th Symposium on Discrete Algorithms, pages 385–394, 2005.

[Frank1956] M. Frank et P. Wolfe. *An algorithm for quadratic programming*. Naval Research Logistics Quarterly, vol. 3, pages 95–110, 1956.

[Gupta2015] Santosh K Gupta, Koushik Kar, Sandipan Mishra et John T Wen. *Distributed consensus algorithms for collaborative temperature control in smart buildings*. In 2015 American Control Conference (ACC), pages 5758–5763. IEEE, 2015.

[Gupta2017] Chirag Gupta, Arun Sai Suggala, Ankit Goyal, Harsha Vardhan Simhadri, Bhargavi Paranjape, Ashish Kumar, Saurabh Goyal, Raghavendra Udupa, Manik Varma et Prateek Jain. *ProtoNN: Compressed and Accurate kNN for Resource-scarce Devices*. In Doina Precup et Yee Whye Teh, editeurs, Proceedings of the 34th International Conference on Machine Learning, volume 70 of *Proceedings of Machine Learning Research*, pages 1331–1340. PMLR, 06–11 Aug 2017.

[Hannan1958] James Hannan. Approximation to rayes risk in repeated play, pages 97–140. Princeton University Press, Princeton, 1958.

[Hassani2017] Hamed Hassani, Mahdi Soltanolkotabi et Amin Karbasi. *Gradient methods for submodular maximization*. In Advances in Neural Information Processing Systems, pages 5841–5851, 2017.

[Hazan2012] Elad Hazan et Satyen Kale. *Projection-free online learning*. In Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Proceedings of the 29th International Conference on Machine Learning, ICML 2012, pages 521–528, 2012. 29th International Conference on Machine Learning, ICML 2012 ; Conference date: 26-06-2012 Through 01-07-2012.

[Hazan2016a] Elad Hazan. *Introduction to online convex optimization*. Foundations and Trends® in Optimization, vol. 2, no. 3-4, pages 157–325, 2016.

[Hazan2016b] Elad Hazan et Haipeng Luo. *Variance-reduced and projection-free stochastic optimization*. In International Conference on Machine Learning, pages 1263–1271, 2016.

[He2018] Lie He, An Bian et Martin Jaggi. *Cola: Decentralized linear learning.* In Advances in Neural Information Processing Systems, pages 4536–4546, 2018.

[Henderson2020] Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky et Joelle Pineau. *Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning.* Journal of Machine Learning Research 21, vol. 1-43, 2020.

[Hosseini2013] S. Hosseini, A. Chapman et M. Mesbahi. *Online distributed optimization via dual averaging.* In 52nd IEEE Conference on Decision and Control, pages 1484–1489, 2013.

[Jaggi2013] Martin Jaggi. *Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization.* In Proceedings of the 30th International Conference on Machine Learning, 2013.

[Joulani2013] Pooria Joulani, Andras Gyorgy et Csaba Szepesvari. *Online Learning under Delayed Feedback.* In Proceedings of the 30th International Conference on Machine Learning, Proceedings of Machine Learning Research, pages 1453–1461, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[Kairouz2021] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet*et al. Advances and Open Problems in Federated Learning.* Foundations and Trends® in Machine Learning, vol. 14, no. 1, 2021.

[Kalai2005] Adam Kalai et Santosh Vempala. *Efficient algorithms for online decision problems.* Journal of Computer and System Sciences, vol. 71, no. 3, pages 291–307, 2005. Learning Theory 2003.

[Kempe2003] David Kempe, Jon Kleinberg et 'Eva Tardos. *Maximizing the spread of influence through a social network.* In KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 137–146. ACM, 2003.

[Kingma2015] Diederik Kingma et Jimmy Ba. *Adam: A Method for Stochastic Optimization.* In International Conference on Learning Representations (ICLR), San Diega, CA, USA, 2015.

[Koloskova2019] Anastasia Koloskova, Sebastian Stich et Martin Jaggi. *Decentralized Stochastic Optimization and Gossip Algorithms with Compressed Communication.* In Kamalika Chaudhuri et Ruslan Salakhutdinov, editeurs, Proceedings of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, pages 3478–3487. PMLR, 09–15 Jun 2019.

[Lacoste-Julien2016] Simon Lacoste-Julien. *Convergence Rate of Frank-Wolfe for Non-Convex Objectives*, 2016.

[Lacoste2019] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt et Thomas Dandres. *Quantifying the Carbon Emissions of Machine Learning*, 2019.

[Li2022] X Li, X Yi et L Xie. *Distributed online convex optimization with an aggregative variable.* IEEE Transactions on Control of Network Systems, vol. 9, no. 1, pages 438–449, 2022.

[Li2023] Xiuxian Li, Lihua Xie et Na Li. *A survey on distributed online optimization and online games.* Annual Reviews in Control, vol. 56, page 100904, 2023.

[Lian2017] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang et Ji Liu. *Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent.* In Advances in Neural Information Processing Systems, pages 5330–5340, 2017.

[Lin2017] Ji Lin, Yongming Rao, Jiwen Lu et Jie Zhou. *Runtime Neural Pruning*. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan et R. Garnett, editeurs, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

[Lorenzo2016] Paolo Di Lorenzo et Gesualdo Scutari. *NEXT: In-Network Nonconvex Optimization*. IEEE Trans. Signal Inf. Process. over Networks, vol. 2, no. 2, pages 120–136, 2016.

[Luccioni2023] Alexandra Sasha Luccioni, Sylvain Viguier et Anne-Laure Ligozat. *Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model*. Journal of Machine Learning Research, vol. 24, no. 253, pages 1–15, 2023.

[MacCarthy2018] Mark MacCarthy. *In Defense of Big Data Analytics*. The Cambridge Handbook of Consumer Privacy, pages 47–78, 2018.

[McMahan2017] Brendan McMahan et Daniel Ramage. *Collaborative machine learning without centralized training data*. Google Research Blog, vol. 3, 2017.

[McMahan2023] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson et Blaise Agüera y Arcas. *Communication-Efficient Learning of Deep Networks from Decentralized Data*, 2023.

[Mitra2023] Angan Mitra, Nguyen Kim Thang, Tuan-Anh Nguyen, Denis Trystram et Paul Youssef. *Online Decentralized Frank-Wolfe: From Theoretical Bound to Applications in Smart-Building*. In Internet of Things: 5th The Global IoT Summit, GIoTS 2022, Dublin, Ireland, June 20–23, 2022, Revised Selected Papers, page 43–54, Berlin, Heidelberg, 2023. Springer-Verlag.

[Mokhtari2017] A Mokhtari, A Koppel, G Scutari et A Ribeiro. *Large-scale nonconvex stochastic optimization by doubly stochastic successive convex approximation*. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4701–4705. IEEE, 2017.

[Mokhtari2018a] Aryan Mokhtari, Hamed Hassani et Amin Karbasi. *Conditional Gradient Method for Stochastic Submodular Maximization: Closing the Gap*. In Conference on Artificial Intelligence and Statistics, volume 84, pages 1886–1895, 2018.

[Mokhtari2018b] Aryan Mokhtari, Hamed Hassani et Amin Karbasi. *Decentralized Submodular Maximization: Bridging Discrete and Continuous Settings*. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, Proceedings of Machine Learning Research, pages 3613–3622. PMLR, 2018.

[Nan2016] Feng Nan, Joseph Wang et Venkatesh Saligrama. *Pruning Random Forests for Prediction on a Budget*. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon et R. Garnett, editeurs, Advances in Neural Information Processing Systems, volume 29. Curran Associates, Inc., 2016.

[Nedic2009] Angelia Nedic et Asuman Ozdaglar. *Distributed Subgradient Methods for Multi-Agent Optimization*. IEEE Transactions on Automatic Control, vol. 54, no. 1, pages 48–61, 2009.

[Nguyen2023] Tuan-Anh Nguyen, Nguyen Kim Thang et Denis Trystram. *One Gradient Frank-Wolfe for Decentralized Online Convex and Submodular Optimization*. In Proceedings of The 14th Asian Conference on Machine Learning, volume 189 of *Proceedings of Machine Learning Research*, pages 802–815. PMLR, 12–14 Dec 2023.

[Nguyen2024] Tuan-Anh Nguyen, Nguyen Kim Thang et Denis Trystram. *Handling Delayed Feedback in Distributed Online Optimization : A Projection-Free Approach*. In Machine Learning and Knowledge Discovery in Databases: Research Track. ECML PKDD, 2024.

[Patterson2022] David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier et Jeff Dean. *The Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink*. Rapport technique arXiv:2204.05149, arXiv, April 2022. arXiv:2204.05149 [cs] type: article.

[Patterson2024] David Patterson, Jeffrey M. Gilbert, Marco Gruteser, Efren Robles, Krishna Sekar, Yong Wei et Tenghui Zhu. *Energy and Emissions of Machine Learning on Smartphones vs. the Cloud*. Communications of the ACM, vol. 67, no. 2, pages 86–97, February 2024.

[Pipattanasomporn2020] Manisa Pipattanasomporn, Gopal Chitalia, Jitkomut Songsiri, Chaodit Aswakul, Wanchalerm Pora, Surapong Suwankawin, Kulyos Audomvongseree et Naebboon Hoonchareon. *CU-BEMS, smart building electricity consumption and indoor environmental sensor datasets*. Scientific Data, vol. 7, no. 1, pages 1–14, 2020.

[Pokutta2020] Sebastian Pokutta, Christoph Spiegel et Max Zimmer. *Deep Neural Network Training with Frank-Wolfe*, 2020.

[Pu2018] Shi Pu et Angelia Nedić. *A Distributed Stochastic Gradient Tracking Method*. In 2018 IEEE Conference on Decision and Control (CDC), pages 963–968, 2018.

[Qiu2023] Xinchi Qiu, Titouan Parcollet, Javier Fernandez-Marques, Pedro P. B. Gusmao, Yan Gao, Daniel J. Beutel, Taner Topal, Akhil Mathur et Nicholas D. Lane. *A First Look into the Carbon Footprint of Federated Learning*. Journal of Machine Learning Research, vol. 24, no. 129, pages 1–23, 2023.

[Quanrud2015] Kent Quanrud et Daniel Khashabi. *Online Learning with Adversarial Delays*. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama et R. Garnett, editeurs, Advances in Neural Information Processing Systems, volume 28. Curran Associates, Inc., 2015.

[Ram2010] Sai Sundar Ram, Angelia Nedić et Venugopal V Veeravalli. *Asynchronous gossip algorithm for stochastic optimization: Constant stepsize analysis*. In Recent Advances in Optimization and its Applications in Engineering, pages 51–60. Springer, 2010.

[Reddi2021] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar et Hugh Brendan McMahan. *Adaptive Federated Optimization*. In International Conference on Learning Representations, 2021.

[Reisizadeh2019] A. Reisizadeh, A. Mokhtari, H. Hassani et R. Pedarsani. *An Exact Quantized Decentralized Gradient Descent Algorithm*. IEEE Transactions on Signal Processing, vol. 67, no. 19, pages 4934–4947, 2019.

[Ruszczyński1980] A Ruszczyński. *Feasible direction methods for stochastic programming problems*. Mathematical Programming, vol. 19, no. 3, pages 220–229, 1980.

[Ruszczyński2008] A Ruszczyński. *A merit function approach to the subgradient method with averaging*. Optimization Methods and Software, vol. 23, no. 1, pages 161–172, 2008.

[Savazzi2022] Stefano Savazzi, Vittorio Rampa, Sanaz Kianoush et Mehdi Bennis. *On the Energy and Communication Efficiency Tradeoffs in Federated and Multi-Task Learning.* In 2022 IEEE 33rd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pages 1431–1437, September 2022. ISSN: 2166-9589.

[Shahrampour2018] S Shahrampour et A Jadbabaie. *Distributed online optimization in dynamic environments using mirror descent.* IEEE Transactions on Automatic Control, vol. 63, no. 3, pages 714–725, 2018.

[Shalev-Shwartz2007] Shai Shalev-Shwartz et Yoram Singer. *A primal-dual perspective of online learning algorithms.* Mach. Learn., vol. 69, no. 2–3, page 115–142, dec 2007.

[Shalev-Shwartz2012] Shai Shalev-Shwartz. *Online Learning and Online Convex Optimization.* Foundations and Trends® in Machine Learning, vol. 4, no. 2, pages 107–194, 2012.

[Shotton2013] Jamie Shotton, Toby Sharp, Pushmeet Kohli, Sebastian Nowozin, John Winn et Antonio Criminisi. *Decision Jungles: Compact and Rich Models for Classification.* In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani et K.Q. Weinberger, editeurs, Advances in Neural Information Processing Systems, volume 26. Curran Associates, Inc., 2013.

[Simons2019] Taylor Simons et Dah-Jye Lee. *A Review of Binarized Neural Networks.* Electronics, vol. 8, no. 6, 2019.

[Strubell2019] Emma Strubell, Ananya Ganesh et Andrew McCallum. *Energy and Policy Considerations for Deep Learning in NLP.* In 57th Annual Meeting of the Association for Computational Linguistics (ACL), Florence, Italy, June 2019. arXiv: 1906.02243.

[Tang2018] Hanlin Tang, Xiangru Lian, Ce Zhang, Tong Zhang, Wei Zhang et Ji Liu. *D^2: Decentralized training over decentralized data.* In International Conference on Machine Learning, pages 4857–4866. PMLR, 2018.

[Thang2022] Nguyen Kim Thang, Abhinav Srivastav, Denis Trystram et Paul Youssef. *A stochastic conditional gradient algorithm for decentralized online convex optimization.* Journal of Parallel and Distributed Computing, vol. 169, pages 334–351, 2022.

[Wai2017] H. Wai, J. Lafond, A. Scaglione et E. Moulines. *Decentralized Frank–Wolfe algorithm for convex and nonconvex problems.* IEEE Transactions on Automatic Control, vol. 62, no. 11, pages 5522–5537, 2017.

[Wan2022a] Yuanyu Wan, Wei-Wei Tu et Lijun Zhang. *Online Frank-Wolfe with Arbitrary Delays.* In Advances in Neural Information Processing Systems, volume 35, pages 19703–19715. Curran Associates, Inc., 2022.

[Wan2022b] Yuanyu Wan, Guanghui Wang, Wei-Wei Tu et Lijun Zhang. *Projection-free Distributed Online Learning with Sublinear Communication Complexity.* Journal of Machine Learning Research, vol. 23, no. 172, pages 1–53, 2022.

[Wang2018] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He et Kevin Chan. *When Edge Meets Learning: Adaptive Control for Resource-Constrained Distributed Machine Learning.* In IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, pages 63–71, 2018.

[Wei2015] Kai Wei, Rishabh Iyer et Jeff Bilmes. *Submodularity in Data Subset Selection and Active Learning.* In Francis Bach et David Blei, editeurs,

Proceedings of the 32nd International Conference on Machine Learning, volume 37 of *Proceedings of Machine Learning Research*, pages 1954–1963, Lille, France, 07–09 Jul 2015. PMLR.

[Wu2022] Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga Behram, James Huang, Charles Bai, Michael Gschwind, Anurag Gupta, Myle Ott, Anastasia Melnikov, Salvatore Candido, David Brooks, Geeta Chauhan, Benjamin Lee, Hsien-Hsin S Lee, Bugra Akyildiz, Max Balandat, Joe Spisak, Ravi Jain, Mike Rabbat et Kim Hazelwood. *Sustainable AI: Environmental Implications, Challenges and Opportunities*. In Proceedings of the 5th MLSys Conference, Santa Clara, CA, USA, 2022.

[Xie2019] Jiahao Xie, Chao Zhang, Zebang Shen, Chao Mi et Hui Qian. *Decentralized Gradient Tracking for Continuous DR-Submodular Maximization*. In Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, pages 2897–2906. PMLR, 2019.

[Yan2013] F. Yan, S. Sundaram, S. V. N. Vishwanathan et Y. Qi. *Distributed Autonomous Online Learning: Regrets and Intrinsic Privacy-Preserving Properties*. IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 11, pages 2483–2493, 2013.

[Yang2016] Y Yang, G Scutari, DP Palomar et M Pesavento. *A parallel decomposition method for nonconvex stochastic multi-agent optimization problems*. IEEE Transactions on Signal Processing, vol. 64, no. 11, pages 2949–2964, 2016.

[Yuan2016] K. Yuan, Q. Ling et W. Yin. *On the Convergence of Decentralized Gradient Descent*. SIAM Journal on Optimization, vol. 26, no. 3, pages 1835–1854, 2016.

[Zamora-Martinez2014] Fransisco Zamora-Martinez, Pablo Romeu, Pablo Botella-Rocamora et Juan Pardo. *On-line learning of indoor temperature forecasting models towards energy efficiency*. Energy and Buildings, vol. 83, pages 162–172, 2014.

[Zhang2017] W. Zhang, P. Zhao, W. Zhu, S.C.V. Hoi et T. Zhang. *Projection-Free Distributed Online Learning in Networks*. In Proceedings of the 34th International Conference on Machine Learning, pages 4054–4062, 2017.

[Zhang2019] Mingrui Zhang, Lin Chen, Hamed Hassani et Amin Karbasi. *Online Continuous Submodular Maximization: From Full-Information to Bandit Feedback*. In Advances in Neural Information Processing Systems, pages 9206–9217, 2019.

[Zhang2020] Mingrui Zhang, Lin Chen, Aryan Mokhtari, Hamed Hassani et Amin Karbasi. *Quantized Frank-Wolfe: Faster Optimization, Lower Communication, and Projection Free*. In Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, volume 108, pages 3696–3706, 2020.

[Zheng2018] W. Zheng, A. Bellet et P. Gallinari. *A Distributed Frank—Wolfe Framework for Learning Low-Rank Matrices with the Trace Norm*. Machine Learning, vol. 107, no. 810, pages 1457–1475, 2018.

[Zhu2021] Junlong Zhu, Qingtao Wu, Mingchuan Zhang, Ruijuan Zheng et Keqin Li. *Projection-free Decentralized Online Learning for Submodular Maximization over Time-Varying Networks*. Journal of Machine Learning Research, 2021.

[Zinkevich2003] Martin Zinkevich. *Online convex programming and generalized infinitesimal gradient ascent.* In Proceedings of the Twentieth International Conference on International Conference on Machine Learning, ICML'03, page 928–935. AAAI Press, 2003.

[Zinkevich2009] Martin Zinkevich, John Langford et Alex Smola. *Slow Learners are Fast.* In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams et A. Culotta, editeurs, Advances in Neural Information Processing Systems, volume 22. Curran Associates, Inc., 2009.

List of Figures

List of Tables