A Unified Framework for Understanding Distributed Optimization Algorithms: System Design and its Applications

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Xinwei Zhang

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Mingyi Hong, Sairaj Dhople

November, 2023

Acknowledgements

There are many people who have earned my gratitude for their contribution to my time in graduate school.

I would like to express my deepest appreciation to my advisors, Professor Mingyi Hong, and Professor Sairaj Dhople, for their patient guidance throughout my academic and research life over the past five years. I also could not have undertaken this journey without my defense committee members, Professor Nicola Elia and Professor Steven Wu, who generously provided knowledge and expertise. Additionally, this endeavor would not have been possible without the generous support from the University's Doctoral Dissertation Fellowship that supported my research.

I am also grateful to my office mates, for their editing help, late-night feedback, and project inspirations. Thanks should also go to the managers and mentors during my internships at Alibaba, IBM, and AWS for their kind help and guidance through my intern projects and valuable experiences. I would like to extend my sincere thanks to all my project collaborators, who helped me with paper writing, academic advising, and experiment support.

I'd like to acknowledge my Undergraduate advisor, Professor Qing Ling, who led me to my first step in this research area. Lastly, I would be remiss in not mentioning my parents. Their support has kept my spirits and motivation high during this process.

Abstract

More than ever before, technology advances across the spectrum have meant that we have individualized and decentralized access to data, resources, and human capital. The capability to utilize massively and distributedly generated data (e.g., personal shopping records) and distributed computation (e.g., fast smartphone processors) has simplified our lives, facilitated optimal resource allocation, and unlocked innovation across industries. Distributed algorithms play a central role in the optimal operation of distributed systems in many applications, such as machine learning, signal processing, and control. Significant research efforts have been devoted to developing and analyzing new algorithms for various applications. However, existing methods are still facing difficulties in using computational resources and distributed data safely and efficiently. The three major challenges in state-of-the-art distributed systems are 1) finding appropriate models to describe the resources and problems in the system, 2) developing a general approach to solving problems efficiently, and 3) ensuring participants' privacy. My thesis research focuses on building an algorithmic framework to resolve these fundamental and practical challenges. This thesis provides a fresh perspective to understand, analyze, and design distributed optimization algorithms. Through the lens of multi-rate feedback control, this thesis theoretically proves that a wide class of distributed algorithms, including popular decentralized and federated schemes, can be viewed as discretizing a certain continuous-time feedback control system, possibly with multiple sampling rates, while preserving the same convergence behavior. Further, the proposed system unifies the stochasticities in a wide range of distributed optimization algorithms as several types of noises injected into the control system, and provides a uniform convergence analysis to a class of distributed stochastic optimization algorithms. The control-based framework is applied to designing new algorithms in decentralized optimization and federated learning to meet different system requirements including achieving convergence, optimal performance, or meeting privacy concerns. In summary, this thesis establishes a control-based framework to understand, analyze, and design distributed optimization algorithms, with applications in decentralized optimization and federated learning algorithm design.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Distributed Optimization

The recent success of Machine Learning (ML) can be largely attributed to its exceptional ability to process data on a massive scale. On the one hand, the size of modern machine learning models have increased tremendously over the past few years. To achieve high performance, these models are becoming so big that they cannot fit into a single GPU or one computational node with multiple GPUs. The training procedure of large foundation models with billions of parameters requires us to solve challenging problems using massive computational resources, either under a centralized parameter server setting [1][2] or a fully decentralized system [3]. On the other hand, the growing network size, the increased amount of distributed data, and the requirements for real-time response often make traditional centralized processing unviable. For example, self-driving cars should be carefully coordinated when meeting at an intersection, but since every such vehicle can generate up to 40 Gbit of data (e.g., from LIDAR and cameras) per second – an amount that overwhelms the fastest cellular network – it is impossible to pool the entirety of data for real-time central coordination. This and other examples, from small and ordinary (e.g. coordinating smart appliances in homes) to large and vitally important (e.g., national power distribution), show how paramount fast distributed processing will be to our collective well-being, productivity, and prosperity.

Distributed optimization algorithms have played an increasingly important role in efficiently utilizing the network resources in modern machine learning applications. The

most widespread approach to distributed optimization is to learn a single global model in a distributed system with N agents connected by a graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, each optimizing a smooth and possibility non-convex local function $f_i(x)$. The global optimization problem is formulated as [4]

$$\min_{\mathbf{x}\in\mathbb{R}^{Nd_x}} \quad f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f_i(x_i), \quad \text{s.t.} \quad x_i = x_j,\ \forall\ (i,j) \in \mathbf{E}, \tag{1.1}$$

where $\mathbf{x} \in \mathbb{R}^{N\times d_x}$ stacks N local variables $\mathbf{x} := [x_1; \ldots; x_N]$; $x_i \in \mathbb{R}^{d_x},\ \forall\ i \in [N]$.

This problem has received much attention in recent years, see [5, 6] for a few recent surveys. Heterogeneous computational and communication resources in the distributed system create a number of different scenarios in distributed learning. In specific, based on the application scenarios, we can roughly classify distributed optimization algorithms into those that solve Decentralized Optimization (DO) problems, those that solve Federated Learning (FL) problems, and those that accelerate the consensus step in decentralized optimization (AC). Some of the related works are discussed below.

Decentralized Optimization

In the scenario of decentralized optimization (DO), the agents are usually connected through a connected but incomplete communication graph, that is, each agent in the communication graph only communicates with several neighboring agents, rather than all agents. When solving the DO problems, the agents are typically modeled as nodes on a communication graph, and the communication and computation resources are equally important. So the algorithms alternatingly perform communication and communication steps. For instance, the Decentralized Gradient Descent (DGD) algorithm [7, 8] extends gradient descent (GD) to the decentralized setting, where each agent performs one step of local gradient descent and local model average in each round. Other related algorithms such as the DLM [9], the Decentralized Gradient Tracking (DGT) [10] and the NEXT [11] all utilize this kind of alternating updates.

Federated Learning

Federated learning (FL)—a distributed machine learning approach proposed in [12]—has gained popularity for applications involving learning from distributed data. In

FL, a cloud server (the "server") can communicate with distributed data sources (the "agents"). The goal is to train a global model that works well for all the distributed data, but without requiring the agents to reveal too much local information. In such systems, communication is more time-consuming than the computation steps and considered as the bottleneck of the system. Since its inception, the broad consensus on FL's implementation appears to involve a generic "local update" strategy to save communication efforts. The basic communication pattern "computation then aggregation" (CTA) protocol involves the following steps: S1) the server sends the global model $\mathbf{x}$ to the agents; S2) the agents update their local models $\mathbf{x}_i$'s based on their local data for several iterations; S3) the server aggregates $\mathbf{x}_i$'s to obtain a new global model $\mathbf{x}$. The FL algorithms, such as the well-known FedAvg [13], perform multiple local updates before one communication step. However, when the data is *heterogeneous* among the agents, it is difficult for these algorithms to achieve convergence [14, 15]. Recent algorithms such as the FedProx [16], SCAFFOLD [17] and FedPD [18] have developed new techniques to improve upon FedAvg.

Accelerated Consensus Algorithm

There have been a number of recent algorithms that are designed to utilize the *minimum* computation and/or communication resources, while computing high-quality solutions, over a decentralized network. In order to accelerate the consensus of the local models over the network and achieve an optimal dependency on the network topology, they typically perform multiple communication steps before one local update. For examples, in [19] a multi-step gossip protocol is used to achieve the optimal convergence rate in decentralized convex optimization; the xFilter [20] is designed for decentralized non-convex problems, and it implements the Chebyshev filter on the communication graph, which requires multi-step communication, and achieves the optimal dependency on the graph spectrum.

1.1.1 Challenges in Distributed Optimization

As previously discussed, distributed optimization comes with several fundamental and practical challenges that differentiate it from the conventional field of centralized optimization. In particular, we list four core challenges that distinguish distributed optimization from traditional setups.

Unifying Framework

Despite the proliferation of distributed algorithms, there is no standard design procedure and methodology for distributed optimization algorithms. For some hot applications, there are simply *too many algorithms* available, so much so that it becomes difficult to track all the technical details. Much of the recent research on this topic appears to be *increasingly focused* on a specific setting. For examples, there has been remarkably high interest in distributed algorithms in recent years across applications. These algorithms are typically developed in an application-specific manner. They are designed, for example, to: improve communication efficiency by utilizing model compression schemes [21, 22]; perform occasional communication [23, 24]; improve computational efficiency by utilizing SGD based schemes [3, 25]; understand the best possible communication and computation complexity [19, 26]; incorporate differential privacy (DP) guarantee into the system [27]; or to deal with the practical situation where even the (stochastic) gradients may not be accessible [28, 29]. However, an algorithm developed for FL may have already been rigorously developed, analyzed, and tested for the DO setting; and vice versa. Since developing algorithms and performing analyses take significant time and effort. Since developing algorithms and performing analyses take significant time and effort, it is desirable to have some mechanisms in place to reduce the possibility of reinventing the wheel.

Note that many existing works analyze optimization algorithms using control theory, but they mainly focus on some very special class of algorithms. For examples, [30] studies continuous-time gradient flow for convex problems; [4, 31] study continuous-time first-order convex optimization algorithms; [32, 33, 34] investigate the acceleration approaches including Nesterov and Heavy-ball momentum methods for centralized problems in discrete time and interpret them as discrete-time controllers; [4, 34] focus on

the continuous-time system and ignore the impact of the discretization; [35, 36, 37] investigate the connection between continuous-time system and discretized gradient descent algorithm, but their approaches and analyses do not generalize to other federated/decentralized algorithms. Further, to our knowledge, none of the above-referred works provide insights about the relationship between different scenarios of distributed algorithms (e.g., between DO and FL).

Agents' System Heterogeneity

In distributed systems, the agents typically generate and store data locally, which leads to non-identical data distribution across the network. For example, in a federated learning next-word-prediction application, the typing history of each application user can be very different. Such data heterogeneity leads to update and model divergence and introduces extra challenges in designing computation- and communication-efficient distributed algorithms. A brute-force implementation of centralized optimization techniques to the distributed system leads to undesirable system behavior, such as bad model performance or heavy resource consumption. In the early stage of distributed algorithm design, either extra assumptions are used to overcome the heterogeneity issue: [12, 38] assume that the local functions are homogeneous and [15, 39] requires the gradients to be bounded; or the algorithm only converges to a neighborhood of the stationary solution, e.g., [40, 3, 14] or sub-optimal convergence rate [41, 42].

Communication Efficiency

It is well understood that communication cost can be the primary bottleneck in distributed optimization, which is mainly due to unreliable communication channels, agent synchronization, and communication bandwidth. In distributed algorithms, the agents need to repeatedly communicate the gradients of every parameter in the model with the central server or with its neighboring agents. This can be time-intensive for large-scale foundation models. In decentralized optimization and accelerated consensus algorithms, the agents perform at least one step of neighbor aggregation between consecutive update steps. Such frequent communication can be extremely time-consuming. On the other hand, Federated learning performs multiple local updates between consecutive communication steps, thus significantly reduces the communication overhead. In recent works,

it has been shown that for FL algorithms solving non-convex problems, such as [39, 14], to achieve ϵ-stationary solution, one can perform $\mathcal{O}(1/\epsilon^{1/2})$ local (stochastic) computation step between every two aggregation steps, so that a total of $\mathcal{O}(1/\epsilon^{3/2})$ aggregation steps are needed. However, it is not clear if this achieves the best communication complexity.

Privacy

Due to the unavoidable communication among the agents during the optimization, the concern of the agents' local data privacy leakage has become a major concern in the algorithm design procedure. While federated learning offers a solution for protecting local data by communicating model updates, e.g., accumulated gradient updates, and avoiding raw data exchange, it does not offer any theoretical guarantee ensuring that such updates do not leak sensitive agent information. Recent works have shown that they are vulnerable to inference attacks and can leak local information during training [43, 44, 45].

Recently, various FL algorithms [46, 47, 48, 49, 50] have been proposed to provide the formal guarantees of *differential privacy* (DP) [51]. In these algorithms, the clients perform multiple local updates between two communication steps, and then perturbation mechanisms are applied to aggregate updates across individual clients. In order for the perturbation mechanism to have formal privacy guarantees, each client's model update needs to have a bounded norm, which is ensured by applying a clipping operation that shrinks individual model updates when their norm exceeds a given threshold. While there has been prior work that studies the clipping effects on stochastic gradients [52, 53, 54] in the differentially private SGD [55], there has not been any work on providing understanding how clipping the model updates affect the optimization performance of FL subject to DP.

1.2 Chapter Organization and Outline of Contributions

This introductory chapter is followed by five chapters, each based on a single published paper. In each chapter, we first introduce the background and motivation of the chapter

and the related work. For readers' convenience, we then introduce the specific problem, notations, and assumptions used in the chapter. Next, we describe the proposed methods to address the specific challenges of the chapter, as well as the theoretical analysis. Finally, we present the numerical experiments to verify our theoretical analysis and the proposed approaches. Each chapter, together with the corresponding appendix containing detailed proofs and extra discussions, is mostly self-contained.

The first two chapters (Chapter 2 and 3) build a framework that addresses the first challenge in distributed optimization as presented in Section 1.1.1, which serves as a base theory for the rest three chapters. Chapter 4-6 identify and address a specific challenge presented in Section 1.1.1. These three chapters not only serve as specific applications to the framework proposed in the first two chapters, but further expand the border of the framework by studying some of the fundamental aspects that the framework has not covered. These provided results are meant to bring us closer to the overarching goal of providing a framework to help researchers and practitioners understand algorithm behavior, predict algorithm performance, and provide guidelines for designing application-specific efficient and secure algorithms to utilize the resources in the distributed system.

Let us now briefly discuss the outline and scope of each chapter and their connections.

1.2.1 A Control-based Framework for Understanding Distributed Optimization

The first two chapters propose a control-based framework for understanding distributed optimization algorithms. The first chapter studies deterministic algorithms belonging to the scenarios of decentralized optimization, federated learning, and accelerated consensus. Through the lens of control theory, we show that distributed algorithms in the three scenarios can be modeled as a double-loop continuous-time system, which differs only by the sampling rate of the two loops during discretization. The second chapter complements the first chapter by extending the framework from deterministic algorithms to stochastic ones. In this chapter, we model the stochasticity in different algorithms as a few classes of noises injected into the control system satisfying certain properties.

The proposed framework provides a pipeline for designing distributed algorithms

that meet specific system requirements with basic convergence guarantees.

Chapter 2: A Deterministic Framework

Distributed algorithms have been playing an increasingly important role in many applications such as machine learning, signal processing, and control. Significant research efforts have been devoted to developing and analyzing new algorithms for various applications. In this work, we provide a fresh perspective to understand, analyze, and design distributed optimization algorithms. Through the lens of multi-rate feedback control, we show that a wide class of distributed algorithms, including popular decentralized/federated schemes, can be viewed as discretizing a certain continuous-time feedback control system, possibly with multiple sampling rates, such as decentralized gradient descent, gradient tracking, and federated averaging. This key observation not only allows us to develop a generic framework to analyze the convergence of the entire algorithm class. More importantly, it also leads to an interesting way of designing new distributed algorithms. We develop the theory behind our framework and provide examples to highlight how the framework can be used in practice.

This chapter is based on: [56] Xinwei Zhang, Mingyi Hong, and Nicola Elia. Understanding a class of decentralized and federated optimization algorithms: A multirate feedback control perspective. SIAM Journal on Optimization, 33(2):652–683, 2023.

Chapter 3: Modeling Stochastic Algorithms with the Framework

In modern machine learning systems, distributed algorithms are deployed across applications to ensure data privacy and optimal utilization of computational resources. This work offers a fresh perspective to model, analyze, and design distributed optimization algorithms through the lens of stochastic multi-rate feedback control. We show that a substantial class of distributed algorithms—including popular Gradient Tracking for decentralized learning, and FedPD and Scaffold for federated learning—can be modeled as a certain discrete-time stochastic feedback-control system, possibly with multiple sampling rates. This key observation allows us to develop a generic framework to analyze the convergence of the entire algorithm class. It also enables one to add desirable features such as differential privacy guarantees easily, or to deal with practical settings such as

partial agent participation, communication compression, and imperfect communication in algorithm design and analysis.

This chapter is based on: [57] Xinwei Zhang, Mingyi Hong, Sairaj Dhople, and Nicola Elia. A stochastic multi-rate control framework for modeling distributed optimization algorithms. In International Conference on Machine Learning, pages 26206–26222. PMLR, 2022.

1.2.2 Non-convex Distributed Optimization Algorithm Design

These three chapters focus on distributed non-convex optimization algorithm design in decentralized optimization and federated learning scenarios with certain system requirements. Firstly, they serve as algorithm design examples of the framework proposed above. Further, each of the chapters addresses one of the fundamental challenges in Section 1.1.1. Chapter 4 adopts the gradient-tracking controller to resolve the data heterogeneity issue in decentralized optimization. Chapter 5 proposes using the primal-dual updates to address the data and agents' heterogeneity issue in the federated learning scenario, which also achieves the communication complexity lower bound for distributed optimization. Chapter 6 addresses the privacy issue in distributed optimization by theoretically and numerically studying the impact of data heterogeneity on the privacy-utility trade-off in federated learning.

Chapter 4: Gradient Tracking for Decentralized Optimization

In the era of big data, it is challenging to train a machine learning model on a single machine or over a distributed system with a central controller over a large-scale dataset. In this chapter, we propose a **g**radient-tracking based **n**onconvex **s**tochastic **d**ecentralized (GNSD) algorithm for solving non-convex optimization problems, where the data is partitioned into multiple parts and processed by the local computational resource. Through exchanging the parameters at each node over a network, GNSD is able to find the first-order stationary points (FOSP) efficiently. From the theoretical analysis, it is guaranteed that the convergence rate of GNSD to FOSPs matches the well-known convergence rate of stochastic gradient descent by shrinking the step-size. Finally, we perform extensive numerical experiments on computational clusters to demonstrate the advantage of GNSD compared with other state-of-the-art methods.

This chapter is based on: [25] Songtao Lu, Xinwei Zhang, Haoran Sun, and Mingyi Hong. GNSD: A gradient-tracking based non-convex stochastic algorithm for decentralized optimization. In 2019 IEEE Data Science Workshop (DSW), pages 315–321. IEEE, 2019.

Chapter 5: Optimal Convergence for Federated Learning

Federated Learning (FL) has become a popular paradigm for learning from distributed data. To effectively utilize data at different devices without moving them to the cloud, algorithms such as the Federated Averaging (FedAvg) have adopted a "computation then aggregation" (CTA) model, in which multiple local updates are performed using local data, before sending the local models to the cloud for aggregation.

However, these schemes typically require strong assumptions, such as the local data are identically independently distributed (i.i.d), or the size of the local gradients are bounded. In this chapter, we first explicitly characterize the behavior of the FedAvg algorithm, and show that without strong and unrealistic assumptions on the problem structure, the algorithm can behave erratically (e.g., diverge to infinity). Aiming at designing FL algorithms that are provably fast and require as few assumptions as possible, we propose a new algorithm design strategy from the primal-dual optimization perspective. Our strategy yields a family of algorithms that take the same CTA model as existing algorithms, but they can deal with the general non-convex objective, and achieve the best possible optimization and communication complexity while being able to deal with both the full batch and mini-batch local computation models. Most importantly, the proposed algorithms are *communication efficient*, in the sense that the communication pattern can be adaptive to the level of heterogeneity among the local data. To the best of our knowledge, this is the first algorithmic framework for FL that achieves all the above properties.

This chapter is based on: [18] Xinwei Zhang, Mingyi Hong, Sairaj Dhople, Wotao Yin, and Yang Liu. FedPD: A federated learning framework with adaptivity to non-iid data. IEEE Transactions on Signal Processing, 69:6055–6070, 2021.

Chapter 6: Privacy Preserving Algorithm for Federated Learning

Providing privacy protection has been one of the primary motivations of Federated Learning (FL). Recently, there has been a line of work on incorporating the formal privacy notion of differential privacy with FL. To guarantee the client-level differential privacy in FL algorithms, the clients' transmitted model updates have to be *clipped* before adding privacy noise. Such clipping operation is substantially different from its counterpart of gradient clipping in the centralized differentially private SGD and has not been well-understood. In this chapter, we first empirically demonstrate that the clipped FedAvg can perform surprisingly well even with substantial data heterogeneity when training neural networks. This is partially because the clients' updates become *similar* for several popular deep architectures. Based on this key observation, we provide the convergence analysis of a differential private (DP) FedAvg algorithm and highlight the relationship between clipping bias and the distribution of the clients' updates. To the best of our knowledge, this is the first work that rigorously investigates theoretical and empirical issues regarding the clipping operation in FL algorithms.

This chapter is based on: [58] Xinwei Zhang, Xiangyi Chen, Mingyi Hong, Steven Wu, and Jinfeng Yi. Understanding clipping for federated learning: Convergence and client-level differential privacy. In International Conference on Machine Learning, pages 26048–26067. PMLR, 2022.

1.2.3 Additional Works not Included in the Dissertation

During my PhD, I co-authored eight more papers that are not a part of this thesis. The list includes:

- Two papers related to optimization algorithms on embedding systems [59, 60].
- Four FL papers. One focuses on hybrid federated learning, one on vertical federated learning, one on over-parameterized networks, and one on federated model ensemble [61, 62, 63, 64].
- One survey paper related to decentralized optimization [5].
- One paper that I co-authored while in IBM focusing on federated graph neural networks [65].

Chapter 2

A Control-based Framework for Understanding Distributed Optimization Algorithms: Deterministic System

2.1 Motivation

Distributed computation has played an important role in popular applications such as machine learning, signal processing, and wireless communications, partly due to the dramatically increased size of the models and the datasets. In this chapter, we consider a distributed system with N agents connected by a graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, each optimizing a smooth and possibility non-convex local function $f_i(x)$. The global optimization problem is formulated as [4]

$$\min_{\mathbf{x}\in\mathbb{R}^{Nd_x}} \quad f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f_i(x_i), \quad \text{s.t.} \quad x_i = x_j,\ \forall\ (i,j) \in \mathbf{E}, \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^{N\times d_x}$ stacks N local variables $\mathbf{x} := [x_1; \ldots; x_N]$; $x_i \in \mathbb{R}^{d_x},\ \forall\ i \in [N]$. This problem has received much attention in recent years, see [5, 6] for a few recent surveys. Heterogeneous computational and communication resources in the distributed system create a number of different scenarios in distributed learning. In specific, based on

the application scenarios, we can roughly classify distributed optimization algorithms into those that solve Decentralized Optimization (DO) problems, that solve Federated Learning (FL) problems, and those that accelerate model consensus (AC). Some of the related works are discussed below.

a) When solving the DO problems, the agents are typically modeled as nodes on a communication graph, and the communication and computation resources are equally important. So the algorithms alternatingly perform communication and communication steps. For instance, the Decentralized Gradient Descent (DGD) algorithm [7, 8] extends gradient descent (GD) to the decentralized setting, where each agent performs one step of local gradient descent and local model average in each round. Other related algorithms such as the DLM [9], the Decentralized Gradient Tracking (DGT) [10] and the NEXT [11] all utilize this kind of alternating updates.

b) The FL problems typically consider the setting that the clients are directly connected to a parameter-server, and that the communication at the server is the bottleneck of the system. The FL algorithms, such as the well-known FedAvg [13], perform multiple local updates before one communication step. However, when the data is *heterogeneous* among the agents, it is difficult for these algorithms to achieve convergence [14, 15]. Recent algorithms such as the FedProx [16], SCAFFOLD [17] and FedPD [18] have developed new techniques to improve upon FedAvg.

c) There have been a number of recent algorithms that are designed to utilize the *minimum* computation and/or communication resources, while computing high-quality solutions. They typically perform multiple communication steps to accelerate model consensus before one local update. For examples, in [19] a multi-step gossip protocol is used to achieve the optimal convergence rate in decentralized convex optimization; the xFilter [20] is designed for decentralized non-convex problems, and it implements the Chebyshev filter on the communication graph, which requires multi-step communication, and achieves the optimal dependency on the graph spectrum.

Despite the proliferation of distributed algorithms, there are a few concerns and challenges. First, for some hot applications, there are simply *too many algorithms* available, so much so that it becomes difficult to track all the technical details. Is it possible to establish some general guidelines to understand the relations between, and the fundamental principles of, those algorithms that provide similar functionalities?

Second, much of the recent research on this topic appears to be *increasingly focused* on a specific setting (e.g., those mentioned in the previous paragraph). However, an algorithm developed for FL may have already been rigorously developed, analyzed, and tested for the DO setting, and vice versa. Since developing algorithms and performing analyses take significant time and effort, it is desirable to have some mechanisms in place to reduce the possibility of reinventing the wheel.

2.2 Preliminaries

We introduce some useful assumptions and notations.

First, let $\otimes$ denote the Kronecker product. the incidence matrix A of a graph $\mathcal{G}$ is defined as: if edge $e(i,j) \in \mathbf{E}$ connects vertex i and j with $i > j$, then $A_{ei} = 1$, $A_{ej} = -1$ and $A_{ek} = 0,\ \forall k \neq i, j$. Let us use $\mathcal{N}_i \subset [N]$ to denote the neighbors for agent i. For a symmetric matrix X, let us use $\lambda(X)$ to denote its eigenvalues. Then we can write the constraint of (2.1) in a more compact form:

$$\min_{\mathbf{x}\in\mathbb{R}^{Nd_x}} \quad f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f_i(x_i), \quad \text{s.t.} \quad (A \otimes I)\cdot \mathbf{x} = 0.$$

For simplicity of notation, the Kronecker products are ignored in the subsequent discussion, e.g., we use $A\mathbf{x}$ in place of $(A\otimes I)\cdot\mathbf{x}$. Define the averaging matrix $R := \frac{\mathbb{1}\mathbb{1}^T}{N}$ and the average of x_i's as $\bar{\mathbf{x}} := \frac{\mathbb{1}^T}{N}\mathbf{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$. Note, we have $R^2 = R$. The consensus error can be written as $[x_1 - \bar{\mathbf{x}}, \ldots, x_N - \bar{\mathbf{x}}] = (I - R)\mathbf{x}$, and we have $\nabla f(\bar{\mathbf{x}}) = \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(\bar{\mathbf{x}})$. The stationary solution of (2.1) is defined as follows:

Definition 1 (First-order Stationary Point) *We define the first-order stationary solution and the ϵ-stationary solution respectively, as:*

$$\sum_{i=1}^{N} \nabla f_i\left(\frac{1}{N}\sum_{i=1}^{N} x_i\right) = 0, \quad \mathbf{x} - \frac{\mathbb{1}\mathbb{1}^T}{N}\mathbf{x} = 0, \tag{2.2a}$$

$$\left\|\frac{1}{N}\sum_{i=1}^{N} \nabla f_i\left(\frac{1}{N}\sum_{i=1}^{N} x_i\right)\right\|^2 + \left\|\mathbf{x} - \frac{\mathbb{1}\mathbb{1}^T}{N}\mathbf{x}\right\|^2 \leq \epsilon. \tag{2.2b}$$

We refer to the left hand side (LHS) of (2.2b) *as the stationarity gap of* (2.1).

We will make the following assumptions on problem (2.1) throughout the chapter:

A 1 (Graph Connectivity) *The graph is fixed, and strongly connected at all time* $t \in [0, \infty)$*, i.e.* 0 *is a simple eigenvalue of* $A^T A$*, with corresponding eigenvector* $\frac{\mathbb{1}}{\sqrt{N}}$*.*

This assumption can be extended to time-varying graphs (denoted as $A(t)$'s), as they can be treated as sub-sampling on a strongly connected graph $A = \bigcup_t A(t)$. However, to stay focused on the main point of the chapter (e.g., build the connection of different algorithms from the control perspective) and to reduce notation, we choose to consider the simple static graph $A(t) = A,\ \forall\ t \in [0, \infty)$ in this work.

Since the agents are connected by a fixed communication graph, we can further define the averaging matrix of the communication graph as $W := I - A^T \mathrm{diag}(\mathbf{w})\mathrm{A}$, where $\mathbf{w}$ is a vector each of whose entries $\mathbf{w}[e(i,j)]$ is positive, and it corresponds to the weight of edge $e(i,j)$. It is easy to check that W has the following properties:

$$W = W^T,\ \mathbb{1}^T W = \mathbb{1}^T,\ W_{ij} \geq 0, \quad \forall e(i,j) \in \mathbf{E}. \tag{2.3}$$

A 2 (Lipschitz gradient) *The* f_i*'s have Lipschitz gradient with constant* L_f*:*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_f \|x - y\|, \quad \forall\ x, y \in \mathbb{R}^{d_x}, \forall\ i \in [N].$$

A 3 (Lower bounded functions) *Each* f_i *is lower bounded as:*

$$f_i(x) \geq \underline{f}_i > -\infty, \quad \forall x \in \mathbb{R}^{d_x}, \quad \forall i \in [N].$$

A 4 (Coercive functions) *Each* f_i *approaches infinity as* $\|x\|$ *approaches infinity:*

$$f_i(x) \to \infty, \text{ as } \|x\| \to \infty, \quad \forall i \in [N].$$

A3 and A4 imply that there exists at least one globally optimal solution $\mathbf{x}^\star$ for problem (2.1). Let us denote the corresponding optimal objective as $f^\star := f(\mathbf{x}^\star)$.

2.3 Continuous-time System

We present a continuous-time feedback control system. We will provide a number of key properties of the controllers and the entire system, to ensure that the system converges to the set of first-order stationary points with guaranteed speed. These properties will

be instrumental when we subsequently analyze discretized version of the system (hence, various distributed algorithms).

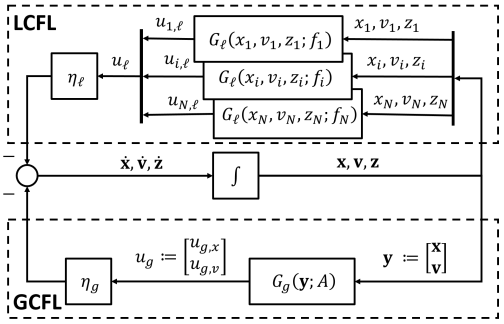


Figure 2.1: The proposed continuous-time double-feedback system for modeling the decentralized optimization problem (2.1). The system dynamics are given in (2.8).

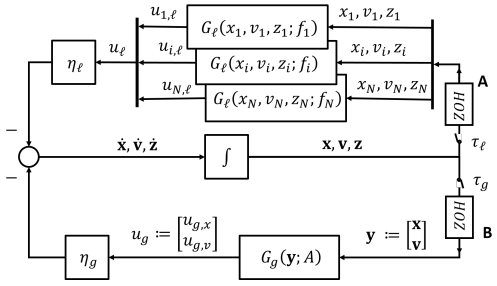


Figure 2.2: Discretized system using ZOH on both the GCFL and LCFL control loops with possibly different sampling times τ_g, τ_ℓ. The system dynamics are given in (2.14)-(2.17)

2.3.1 System Description

To optimize problem (2.1), our approach is to design a continuous-time feedback control system, such that the state variables belong to the set of stationary points of the system if and only if they correspond to a stationary solution of (2.1). Towards this end, define $\mathbf{x} \in \mathbb{R}^{Nd_x}$ as the main state variable of the system; introduce the *global consensus feedback loop* (GCFL) and *local computation feedback loop* (LCFL), where the former incorporates the dynamics from multi-agent interactions and pushes $\mathbf{x}$ to consensus, while the latter helps stabilize the system and finds the stationary solution. Specifically, these loops are defined as below:

• **(The GCFL).** Define an auxiliary state variable $\mathbf{v} := [v_1; \ldots; v_N] \in \mathbb{R}^{Nd_v}$, with $v_i \in \mathbb{R}^{d_v}$, $\forall\ i$; define $\mathbf{y} := [\mathbf{x}; \mathbf{v}] \in \mathbb{R}^{N(d_x+d_v)}$; define a feedback controller $G_g(\cdot; A) : \mathbb{R}^{N(d_x+d_v)} \to \mathbb{R}^{N(d_x+d_v)}$. Then the GCFL uses $G_g(\cdot; A)$ to operate on $\mathbf{y}$, to ensure the agents remain coordinated, and their local control variables remain close to consensus;

• **(The LCFL).** Define an auxiliary state variable $\mathbf{z} := [z_1; \ldots; z_N] \in \mathbb{R}^{Nd_z}$, with $z_i \in \mathbb{R}^{d_z}$, $\forall\ i$; define a set of feedback controller $G_\ell(\cdot; f_i) : \mathbb{R}^{d_x+d_v+d_z} \to \mathbb{R}^{d_x+d_v+d_z}$, one for each agent i. Then each agent will use LCFL to operate on its local state variables x_i, z_i and v_i, to ensure that its local system can be stabilized.

The overall system is described in Fig. 2.1. The detailed description of properties of different controllers, as well as the notations used, will be given in the next sections.

To have a rough idea of how these loops can be mapped to a distributed algorithm, let us consider the PI distributed optimization algorithm [66], whose updates are:

$$\begin{aligned}\dot{\mathbf{x}} &= -k_G \nabla f(\mathbf{x}) - k_P \cdot (I - W) \cdot \mathbf{x} - k_P k_I \mathbf{v},\\ \dot{\mathbf{v}} &= k_P k_I \cdot (I - W)\mathbf{x}.\end{aligned}$$

The corresponding controllers are given by:

$$G_g(\mathbf{x}, \mathbf{v}; A) := \begin{bmatrix} (I - W) \cdot \mathbf{x} + k_I \mathbf{v} \\ -k_I \cdot (I - W) \cdot \mathbf{x} \end{bmatrix}, \quad G_\ell(x_i, v_i, z_i; f_i) := \begin{bmatrix} \nabla f_i(x_i) \\ 0 \\ 0 \end{bmatrix},$$

with $\eta_\ell = k_G$ and $\eta_g = k_P$. Note that auxiliary state variable $\mathbf{z}$ has not been used in this algorithm.

Next, we describe in detail the properties of the two feedback loops.

2.3.2 Global Consensus Feedback Loop

The GCFL performs inter-agent communication based on the incidence matrix A, and it controls the consensus of the global variable $\mathbf{y} := [\mathbf{x}; \mathbf{v}]$. Specifically, at time t, define the output of the controller as $u_g(t) = G_g(\mathbf{y}(t); A)$, which can be further decomposed into two outputs $u_g(t) := [u_{g,x}(t); u_{g,v}(t)]$, one to control the consensus of $\mathbf{x}$ and the other for $\mathbf{v}$. After multiplied by the control gain $\eta_g(t) > 0$, the resulting signal will be combined with the output of the LCFL, and be fed back to local controllers.

We require that the global controller $G_g(\cdot; A)$ to have the following properties:

P 1 (Control Signal Direction) *The output of the controller G_g aligns with the direction that reduces the consensus error, that is:*

$$\langle (I - R) \cdot \mathbf{y}, G_g(\mathbf{y}; A) \rangle \geq C_g \cdot \| (I - R) \cdot \mathbf{y} \|^2, \quad \forall\, \mathbf{y},$$

for some constant $C_g > 0$. Further, the controller G_g satisfies:

$$\langle \mathbb{1}, G_g(\mathbf{y}; A) \rangle = 0, \quad \forall\, \mathbf{y}, \quad \text{which implies} \quad \langle \mathbb{1}, u_g(t) \rangle = 0, \quad \forall\, t.$$

P 2 (Linear Operator) *The controller G_g is a linear operator of $\mathbf{y}$, that is, we have $G_g(\mathbf{y}; A) = W_A\mathbf{y}$ for some matrix $W_A \in \mathbb{R}^{N(d_x+d_v)}$ parameterized by A, and its eigenvalues satisfy: $|\lambda(W_A)| \in [0, 1]$.*

Combining P1 and P2, we have $\langle \mathbb{1}, W_A \rangle = 0$, which indicates $R \cdot W_A = 0$ and the eigenvectors of W_A are orthogonal to the ones of R. Further we have

$$\begin{aligned}\|(I-R)\mathbf{y}\|^2 - \|G_g(\mathbf{y}; A)\|^2 &= \mathbf{y}^T((I-R)^2 - W_A^2)\mathbf{y} \\ &= \mathbf{y}^T\left(I - 2R + R - W_A^2\right)\mathbf{y} = \mathbf{y}^T(I - (R + W_A^2))\mathbf{y}.\end{aligned}$$

Notice the eigenvectors of R and W_A are orthogonal and all eigenvalues are in $[0, 1]$, so we have matrix $I - (R + W_A^2) \succeq 0$. Thus $\mathbf{y}^T(I - (R + W_A^2))\mathbf{y} \geq 0$ and $\|(I-R)\mathbf{y}\|^2 \geq \|G_g(\mathbf{y}; A)\|^2$. Therefore, we have:

$$C_g^2 \|(I-R)\cdot\mathbf{y}\|^2 \leq \|G_g(\mathbf{y}; A)\|^2 \leq \|(I-R)\cdot\mathbf{y}\|^2, \text{ and } R \cdot W_A = 0. \tag{2.4}$$

It is easy to check that both P1 and P2 hold in most of the existing consensus-based algorithms. For example, when the communication graph is strongly connected, we can choose $G_g(\mathbf{y}; A) = (I - W)\cdot\mathbf{y}$. It is easy to verify that, $C_g = 1 - \lambda_2(W)$ where $\lambda_2(\cdot)$ denotes the eigenvalue withe the second largest magnitude [8, 5]. As another example, consider the accelerated averaging algorithms [67], where we have

$$G_g(\mathbf{y}, A) = \begin{bmatrix} I - (c+1)\cdot W & c\cdot I \\ -I & I \end{bmatrix}\begin{bmatrix} \mathbf{x} \\ \mathbf{v} \end{bmatrix}, \text{ with } c := \frac{1-\sqrt{1-\lambda_2(W)}}{1+\sqrt{1-\lambda_2(W)^2}}.$$

In this case, one can verify that $C_g = 1 - \frac{\lambda_2(W)}{1+\sqrt{1-\lambda_2(W)^2}} \geq 1 - \lambda_2(W)$.

By using P1, we can follow the general analysis of averaging systems [68], and show that the GCFL will behave *as expected*, that is, if the system *only* performs GCFL and shuts off the LCFL, then the consensus can be achieved. More precisely, assuming that $\eta_\ell(t) = 0, \eta_g(t) = 1$, then under P1, the local state $\mathbf{y}$ converges to the average of the initial states linearly:

$$\|(I-R)\cdot\mathbf{y}(t)\|^2 \leq \mathrm{e}^{-2C_g t}\|(I-R)\cdot\mathbf{y}(0)\|^2. \tag{2.5}$$

For completeness, we include the derivation in Appendix A.4.1.

2.3.3 The Local Computation Feedback Loop

The LCFL optimizes the local function $f_i(\cdot)$'s for each agent. At time t, the ith local controller takes the local variables $x_i(t), v_i(t), z_i(t)$ as inputs and produces a local control signal. To describe the system, let us denote the output of the local controllers as $u_{i,\ell}(t) = G_\ell(x_i(t), v_i(t), z_i(t); f_i),\ \forall\ i \in [N]$; further decompose it into three parts:

$$u_{i,\ell}(t) := [u_{i,\ell,x}(t); u_{i,\ell,v}(t); u_{i,\ell,z}(t)].$$

Denote the concatenated local controller outputs as: $u_{\ell,x}(t) := [u_{1,\ell,x}(t); \ldots; u_{N,\ell,x}(t)]$, and define $u_{\ell,v}(t), u_{\ell,z}(t)$ similarly. Note that we have assumed that all the agents use the same local controller $G_\ell(\cdot;\cdot)$, but they are parameterized by different f_i's. After multiplied by the control gain $\eta_\ell(t) > 0$, the resulting signal will be combined with the output of GCFL, and be fed back to the local controllers.

The local controllers are designed to have the following properties:

P 3 (Lipschitz Smoothness) *The controller is Lipschitz continuous, that is:*

$$\begin{aligned}&\left\|G_\ell(x_i, v_i, z_i; f_i) - G_\ell(x_i', v_i', z_i'; f_i)\right\| \le L \left\|[x_i; v_i; z_i] - [x_i'; v_i'; z_i']\right\|, \\ &\forall\ i \in [N],\ x_i, x_i' \in \mathbb{R}^{d_x},\ v_i, v_i' \in \mathbb{R}^{d_v},\ z_i, z_i' \in \mathbb{R}^{d_z}.\end{aligned}$$

P 4 (Control Signal Direction and Size) *The local controllers are designed such that there exist initial values $x_i(t_0)$, $v_i(t_0)$ and $z_i(t_0)$ ensuring that the following holds:*

$$\langle \nabla f_i(x_i(t)), u_{i,\ell,x}(t) \rangle \ge \alpha(t) \cdot \|\nabla f_i(x_i(t))\|^2, \quad \forall\ t \ge t_0,$$

where $\alpha(t) > 0$ satisfies $\lim_{t\to\infty} \int_{t_0}^{t} \alpha(\tau)\mathrm{d}\tau \to \infty$.

Further, for any given x_i, v_i, z_i, the sizes of the control signals are upper bounded by those of the local gradients. That is, for some positive constants C_x, C_v and C_z:

$$\|u_{i,\ell,x}\| \le C_x \|\nabla f_i(x_i)\|, \ \ \|u_{i,\ell,v}\| \le C_v \|\nabla f_i(x_i)\|, \ \ \|u_{i,\ell,z}\| \le C_z \|\nabla f_i(x_i)\|.$$

Let us comment on these properties. P3 is easy to verify for a given realization of the local controllers; P4 abstracts the convergence property of the local optimizer. This property implies that the update direction $-u_{i,\ell,x}(t)$ points to a direction that decreases the local objective. Note that it is postulated that x_i, v_i and z_i are initialized properly,

because in some of the cases, improper initial values lead to non-convergence of the local controllers (or equivalently, the local algorithm). For example, for accelerated gradient descent method [69, 70], $z_i(t_0)$ should be initialized as $\nabla f_i(x_i(t_0))$.

By using P4, we can follow the general analysis of the gradient flow algorithms (e.g., [71]), and show that the LCFL will behave *as expected*, in the sense that the agents can properly optimize their local problems. More precisely, assume that $\eta_g(t) = 0, \eta_\ell(t) = 1$, that is, the system shuts off the GCFL. Assume that $G_\ell(\cdot;\cdot)$ satisfies P4, then each local system produces $x_i(t)$'s that satisfy:

$$\min_{\tau} \|\nabla f_i(x_i(t+\tau))\|^2 \leq \gamma(\tau) \cdot (f_i(x_i(t)) - \underline{f}_i), \tag{2.6}$$

where $\{\gamma(\tau)\}$ is a sequence of positive constants satisfying:

$$\gamma(\tau) = \frac{1}{\int_0^t \alpha(\tau)\mathrm{d}\tau} \to 0, \quad \text{as } \tau \to \infty. \tag{2.7}$$

We include the proof of the above result in the Appendix A.4.2.

To close this subsection, we note that the continuous-time system we have presented so far (cf. Figure 2.1) can be described using the following dynamics:

$$\begin{aligned} \dot{\mathbf{v}}(t) &= -\eta_g(t) \cdot u_{g,v}(t) - \eta_\ell(t) \cdot u_{\ell,v}(t) \\ \dot{\mathbf{x}}(t) &= -\eta_g(t) \cdot u_{g,x}(t) - \eta_\ell(t) \cdot u_{\ell,x}(t), \quad \dot{\mathbf{z}}(t) = -\eta_\ell(t) \cdot u_{\ell,z}(t). \end{aligned} \tag{2.8}$$

Additionally, throughout the chapter, we will use u_g and G_g, u_ℓ and G_ℓ interchangeably.

2.3.4 Convergence Properties

We proceed to analyze the convergence of the continuous-time system. Toward this end, we define an energy-like function:

$$\mathcal{E}(t) := f(\bar{\mathbf{x}}(t)) - f^\star + \frac{1}{2} \|(I - R) \cdot \mathbf{y}(t)\|^2 . \tag{2.9}$$

Note that $\mathcal{E}(t) \geq 0$ for all $t \geq 0$. It follows that its derivative can be expressed as:

$$\dot{\mathcal{E}}(t) = -\left\langle \nabla f(\bar{\mathbf{x}}(t), \eta_\ell(t) \cdot \frac{\mathbb{1}^T}{N} u_{\ell,x}(t) \right\rangle + \langle (I - R) \cdot \mathbf{y}(t), \eta_g(t) u_g(t) + \eta_\ell(t) u_{\ell,y}(t) \rangle . \tag{2.10}$$

In the following, we study the convergence of $\mathcal{E}(t)$ and characterize the set of stationary points that the states satisfy $\dot{\mathcal{E}}(t) = 0$. We do not attempt to analyze the stronger

property of *stability*, not only because such kind of analysis can be challenging due to the non-convexity of the local functions $f_i(\cdot)$'s, but more importantly, analyzing the convergence of $\mathcal{E}(t)$ is already sufficient for us to understand the convergence of the state variable $\mathbf{x}$ to the set of stationary solutions of problem (2.1), as we will show shortly.

To proceed, we require that the system satisfies the following property:

P 5 (Energy Function Reduction) *The derivative of the energy function, $\dot{\mathcal{E}}(\cdot)$ as expressed in* (2.10)*, satisfies the following:*

$$\begin{aligned} &-\int_0^t \left(\left\langle \nabla f(\bar{\mathbf{x}}(\tau), \eta_\ell(\tau) \cdot \frac{\mathbb{1}^T}{N} u_{\ell,x}(\tau) \right\rangle + \langle (I-R)\cdot \mathbf{y}(\tau), \eta_g(\tau) u_g(\tau) + \eta_\ell(\tau) u_{\ell,y}(\tau) \rangle \right) \mathrm{d}\tau \\ &\leq -\int_0^t \left(\gamma_1(\tau) \cdot \left\| \nabla f(\bar{\mathbf{x}}(\tau)) \right\|^2 + \gamma_2(\tau) \cdot \| (I-R)\cdot \mathbf{y}(\tau) \|^2 \right) d\tau, \end{aligned} \tag{2.11}$$

where $\gamma_1(\tau), \gamma_2(\tau) > 0$ are some time-dependent coefficients.

P5 is a property about the entire continuous-time system. Although one could show that by using P1 - P4, and by selecting $\eta_g(t)$ and $\eta_\ell(t)$ appropriately, this property can be satisfied with some *specific* $\gamma_1(\tau)$ and $\gamma_2(\tau)$ (cf. Corollary 1.), here we still list it as an independent property, because at this point we want to keep the choice of $\gamma_1(\tau)$, $\gamma_2(\tau)$ general; please see Sec. 2.3.5 for more detailed discussion.

Next, we will show that under P5, the continuous-time system will converge to the set of stationary points, and that $\mathbf{x}$ will converge to the set of stationary solutions of problem (2.1).

Theorem 1 *Suppose P5 holds true. Then we have the following results:*
1) *Further, suppose that P1, P2 and P4 hold, then $\dot{\mathcal{E}} = 0$ implies that the corresponding state variable $\mathbf{x}_s$ is bounded, and the following holds:*

$$\dot{\mathbf{x}}_s = 0, \quad \dot{\mathbf{v}}_s = 0, \quad \dot{\mathbf{z}}_s = 0, \quad u_g = 0, \quad u_\ell = 0. \tag{2.12}$$

Additionally, let us define the set $\mathbf{S}$ as below:

$$\mathbf{S} := \left\{ \mathbf{v}, \mathbf{z} \,\middle|\, \eta_\ell u_{\ell,v} + \eta_g u_{g,v} = 0,\ u_{\ell,z} = 0,\ \eta_\ell u_{\ell,x} + \eta_g u_{g,x} = 0 \right\}.$$

If we assume that $\mathbf{S}$ is compact for any state variable $\mathbf{x}$ that satisfies the stationarity condition (2.2a)*, then the auxiliary state variables $\{\mathbf{v}(t)\}$ and $\{\mathbf{z}(t)\}$ are also bounded.*

2) *The control system asymptotically converges to the set of stationary points, in that* $\mathbf{x}(t)$ *is bounded* $\forall t \in [0, \infty)$*, and* $\dot{\mathcal{E}} \to 0$*. Further, the stationary gap* (2.2b) *can be upper bounded by the following:*

$$\min_t \left\{ \|\nabla f(\bar{\mathbf{x}}(t))\|^2 + \|(I-R)\cdot \mathbf{y}(t)\|^2 \right\} = \mathcal{O}\left(\max\left\{ \frac{1}{\int_0^T \gamma_1(\tau)\mathrm{d}\tau}, \frac{1}{\int_0^T \gamma_2(\tau)\mathrm{d}\tau} \right\} \right). \tag{2.13}$$

Proof 1 *To show part (1), consider a set of states* $\mathbf{x}_s, \mathbf{v}_s, \mathbf{z}_s$ *in which* $\dot{\mathcal{E}}(\mathbf{x}_s, \mathbf{v}_s) = 0$*. P5 implies that* $\nabla f(\bar{\mathbf{x}}_s) = 0$*, and P4 implies* $\|u_\ell\| \leq (C_x + C_v + C_z)\|\nabla f(\bar{\mathbf{x}}_s)\| = 0$*. Similarly, with P1 and P2 we have that* $\langle u_g, (I-R)\mathbf{y}_s\rangle = 0$ *and* $\mathbb{1}^T u_g = 0$ *so* $u_g = 0$*. Therefore* $\dot{\mathbf{x}}_s = 0, \dot{\mathbf{v}}_s = 0, \dot{\mathbf{z}}_s = 0$*. Combining* $\nabla f(\bar{\mathbf{x}}_s) = 0$ *and A4 implies that* $\mathbf{x}_s$ *is bounded. Note that the value of* $\mathbf{v}(t), \mathbf{z}(t)$ *may not be bounded, even if the system converges to a stationary solution. Using the compactness assumption on the set* $\mathbf{S}$*, it is easy to show that* $\mathbf{v}(t), \mathbf{z}(t)$ *are also bounded.*

To show part (2), we can integrate $\dot{\mathcal{E}}(t)$ *from* $t = 0$ *to* T *to obtain:*

$$\int_0^T \gamma_2(t)\|(I-R)\cdot\mathbf{y}(t)\|^2 \mathrm{d}t + \int_0^T \gamma_1(t)\|\nabla f(\bar{\mathbf{x}}(t))\|^2 \mathrm{d}t \leq \mathcal{E}(0) - \mathcal{E}(T),$$

divide both sides by $\int_0^T \gamma_1(t)\mathrm{d}t$ *or* $\int_0^T \gamma_2(t)\mathrm{d}t$*, we obtain*(2.13)*. By P5 we know* $\int_0^t \dot{\mathcal{E}}(\tau)\mathrm{d}\tau \leq 0$*,* $\forall t$*, but since* $\mathcal{E}(t) \geq 0$*, it follows that* $\lim_{t\to\infty} \dot{\mathcal{E}}(t) = 0$*.*

Note that without the compactness assumption, $\mathbf{v}$ and $\mathbf{z}$ can be unbounded. As an example, FedYogi uses AdaGrad for LCFL [72] where $\mathbf{v}(t)$ accumulates the norm of the gradients and does not satisfy the compactness assumption, so $\lim_{t\to\infty} \mathbf{v}(t) \to \infty$. Although such unboundedness does not affect the convergence of the main state variable in part (2), from the control perspective it is still desirable to have a sufficient condition to guarantee the boundedness of all state variables.

Part (2) of the above result indicates that if P5 is satisfied, not only will the system asymptotically converge to the set of stationary points, but more importantly, we can use $\{\gamma_1(t), \gamma_2(t)\}$ to characterize the rate in which the stationary gap of problem (2.1) shrinks. This result, although rather simple, will serve as the basis for our subsequent system discretization analysis.

2.3.5 Summary

So far, we have completed the setup of the continuous-time feedback control system, by specifying the state variables, the feedback loops, and by introducing a few desired properties of the local controllers and the entire system. In particular, we show that property P5 is instrumental in ensuring that the system converges to the set of stationary points. However, there are two key questions remain to be answered:
(i) How to ensure property P5 for a given continuous-time feedback control system?
(ii) How to map the continuous-time system to a distributed optimization algorithm, and to transfer the convergence guarantees of the former to the latter?

There are two different ways to answer question **(i)**. First, for a *generic* system that satisfies properties P1 – P4, we can show that when the control gains $\eta_g(t), \eta_\ell(t)$ are selected appropriately, then P5 will be satisfied; see Corollary 1 below.

Corollary 1 *Suppose that P1, P3, P4 are satisfied. By choosing* $\eta_g(t) = 1, \eta_\ell(t) = \mathcal{O}(1/\sqrt{T})$*, P5 holds true with* $\gamma_1(t) = \mathcal{O}(\eta_\ell(t))$*,* $\gamma_2(t) = \mathcal{O}(1)$ *Further,*

$$\min_t \left\{ \|\nabla f(\bar{\mathbf{x}}(t))\|^2 + \|(I - R) \cdot \mathbf{y}(t)\|^2 \right\} = \mathcal{O}\left(\frac{1}{\int_0^T \eta_\ell(\tau) \mathrm{d}\tau} \right) = \mathcal{O}\left(\frac{1}{\sqrt{T}} \right).$$

The proof of the above result follows the steps used in analyzing distributed gradient flow algorithm [37]; see Appendix A.4.3.

The second answer to question **(i)** is that one can also verify P5 in a case-by-case manner for individual systems. In this way, it is possible that one can obtain larger gains $\eta_\ell(t), \eta_g(t)$, hence larger coefficients $\gamma_1(t)$ and $\gamma_2(t)$ to further improve the convergence rate estimate. In fact, verifying property P5, and computing the corresponding coefficients is a key step in our proposed analysis framework for distributed algorithms. Shortly in Sec. 2.5.1, we will provide an example to showcase how to verify that the continuous-time system which corresponds to the DGT algorithm satisfies P5 with $\gamma_1(t) = \mathcal{O}(1)$ and $\gamma_2(t) = \mathcal{O}(1)$, leading to a convergence rate of $\mathcal{O}(1/T)$.

On the other hand, the answer to question **(ii)** is more involved, so this question will be addressed in the main technical part of this work to be presented shortly. Generally speaking, one needs to discretize the continuous-time system properly to map the system to a particular distributed algorithm. Further, one needs to utilize all the properties

P1 – P5, and carefully select the discretization intervals, to ensure that the resulting discretized systems perform appropriately.

2.4 System Discretization

In this section, we discuss how to use system discretization to map the continuous-time system introduced in the previous section to distributed algorithms.

2.4.1 Modeling the Discretization

Typically, a continuous-time system is discretized by using a switch that samples the input with sample time τ, followed by a zeroth-order hold (ZOH) that keeps the signal constant between the consecutive sampling instances [73]; see Figure 2.3.

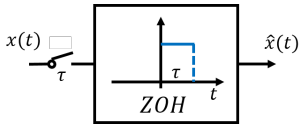


Figure 2.3: The discretization block that has a switch and a Zero-Order Hold.

Now, let us use ZOH to discretize the continuous-time system depicted in Fig. 2.1. We will place the ZOH before the variables enter the controllers, i.e., at points A and B in Fig. 2.2. Note that the original continuous-time system can be discretized in many different ways, by customizing the sampling rates for the discretization blocks. Each of these discretization schemes will correspond to a *multi-rate* control system, in which different parts of the system run on different sampling rates. To describe such kinds of multi-rate systems, let us define the *sampling intervals* for the GCFL and LCFL as τ_g and τ_ℓ, respectively. Then we can consider the following five cases:

- **Case I.** $\tau_g > 0, \tau_\ell = 0$, the GCFL is discretized while the LCFL is not;
- **Case II.** $\tau_g = 0, \tau_\ell > 0$, the GCFL remains continuous while the LCFL is not;
- **Case III.** $\tau_g = \tau_\ell > 0$, the GCFL and LCFL are discretized with the same rate;
- **Case IV.** $\tau_g > \tau_\ell > 0$, both the GCFL and LCFL are discretized, while the local computation loop is updated more frequently;

• **Case V.** $\tau_\ell > \tau_g > 0$, both GCFL and LCFL are discretized, while the global communication loop is updated more frequently.

We note that the systems in cases I and II are *sampled data* systems which has both continuous-time part and discretized part, while systems in cases IV, V are *multi-rate discrete-time* systems. Further, the entire system in case III operates on the same sampling rate. For simplicity, we refer both sampled data systems and fully discretized systems as *discretized system* in the rest of the chapter.

2.4.2 Distributed Algorithms as Multi-Rate Discretized Systems

In this section, we make the connection between *sub-classes* of distributed algorithms and different discretization patterns. For convenience, let t_k denote the times at which the inputs of the ZOHs get sampled by *both* the global and local controllers.

Case I ($\tau_g > 0, \tau_\ell = 0$): The system can be described as:

$$\begin{aligned} \dot{\mathbf{v}}(t) &= -\eta_g(t) \cdot u_{g,v}(t_k) - \eta_\ell(t) \cdot u_{\ell,v}(t) \\ \dot{\mathbf{x}}(t) &= -\eta_g(t) \cdot u_{g,x}(t_k) - \eta_\ell(t) \cdot u_{\ell,x}(t), \quad \dot{\mathbf{z}}(t) = -\eta_\ell(t) \cdot u_{\ell,z}(t). \end{aligned} \tag{2.14}$$

Due to the use of ZOH, during an interval $[t_k, t_k + \tau_g)$, the control signals $u_{g,v}$ and $u_{g,x}$ are fixed. By P4, it follows that the dynamic system finds a stationary point of the local problem satisfying $\dot{x}_i = 0,\ \forall\ i$, that is $\eta_\ell(t) \cdot u_{\ell,x}(t) + \eta_g(t) \cdot u_{g,x}(t_k) = 0$. This is the stationary solution of the following perturbed problem for each agent:

$$\min_{x_i}\ \widetilde{f}_i(x_i) := f_i(x_i) + \frac{\eta_g(t)}{\eta_l(t)} \left\langle u_{i,g,x}(t_k), x_i \right\rangle . \tag{2.15}$$

Using (2.6), it follows that the above problem is optimized to satisfy:

$$\min_{t \in [t_k, t_k + \tau_g]} \left\| \nabla \widetilde{f}_i(x_i(t)) \right\|^2 \leq \gamma(\tau_g) \cdot \left(\widetilde{f}_i(x_i(t_k)) - \widetilde{f}_i(x_i(t_k + \tau_g)) \right),$$

with $\gamma(\tau_g) = \frac{1}{\int_0^{\tau_g} \alpha(t) \mathrm{d}t}$. That is, we obtain a $\gamma(\tau_g)$-stationary solution for the local problem (2.15). This system has the same form as the distributed algorithms that require to solve some local problems to a given accuracy, before any local communication steps take place; see for examples FedProx [16], FedPD [18] and NEXT [11].

Case II ($\tau_g = 0, \tau_\ell > 0$): The system can be described as:

$$\begin{aligned} \dot{\mathbf{v}}(t) &= -\eta_g(t) \cdot u_{g,v}(t) - \eta_\ell(t) \cdot u_{\ell,v}(t_k) \\ \dot{\mathbf{x}}(t) &= -\eta_g(t) \cdot u_{g,x}(t) - \eta_\ell(t) \cdot u_{\ell,x}(t_k), \quad \dot{\mathbf{z}}(t) = -\eta_\ell(t) \cdot u_{\ell,z}(t_k). \end{aligned} \tag{2.16}$$

During $[t_k, t_k+\tau_\ell)$ the control signals $u_{\ell,x}(t), u_{\ell,v}(t), u_{\ell,z}(t)$ are fixed. By P1, the system finds a solution $\dot{\mathbf{y}} = 0$, which implies that $-\eta_g(t) \cdot u_{g,x}(t) - \eta_\ell(t) \cdot u_{\ell,x}(t_k) = 0$. By (2.5), in $[t_k, t_k + \tau_\ell)$, the system optimizes the following network problem:

$$\min_{\mathbf{y}} g(\mathbf{y}) := \|(I - R) \cdot \mathbf{y} + (\eta_\ell(t)/\eta_g(t)) \cdot u_{\ell,y}(t_k)\|^2 ,$$

and obtain a solution that satisfies: $\|\nabla g(\mathbf{y}(t_k + \tau_\ell))\|^2 \leq \mathrm{e}^{-2C_g\tau_\ell} g(\mathbf{y}(t_k))$. This system is related to those algorithms that achieve the optimal communication complexity [19, 20]. In these algorithms, it is often the case that some networked problems are solved (to sufficient accuracies) between two local optimization steps.

Case III ($\tau_g = \tau_\ell > 0$): The system is discretized with a single sampling interval. Once sampled at t_k, the controllers' inputs remain to be $\mathbf{x}(t_k), \mathbf{v}(t_k), \mathbf{z}(t_k)$ during the sampling interval, the output of the controllers are also kept constant $u_g(t) = u_g(t_k), u_\ell(t) = u_\ell(t_k), \forall t \in [t_k, t_k + \tau_g)$. So the system update can be written as:

$$\begin{aligned}
\mathbf{x}(t_{k+1}) &= \mathbf{x}(t_k) - \eta'_\ell(t_k) \cdot u_{\ell,x}(t_k) - \eta'_g(t_k) \cdot u_{g,x}(t_k), \\
\mathbf{v}(t_{k+1}) &= \mathbf{v}(t_k) - \eta'_\ell(t_k) \cdot u_{\ell,v}(t_k) - \eta'_g(t_k) \cdot u_{g,v}(t_k), \\
\mathbf{z}(t_{k+1}) &= \mathbf{z}(t_k) - \eta'_g(t_k) \cdot u_{\ell,z}(t_k),
\end{aligned} \tag{2.17}$$

where $\eta'_\ell(t_k) = \int_{t_k}^{t_k+\tau_g} \eta_\ell(t)\mathrm{d}t$, $\eta'_g(t_k) = \int_{t_k}^{t_k+\tau_g} \eta_g(t)\mathrm{d}t$. The above updates are equivalent to many existing decentralized optimization algorithms, such as DGD, DLM, which perform one step local update, followed by one step of communication.

Case IV ($\tau_g > \tau_\ell > 0$): We assume that $\tau_g = Q \cdot \tau_\ell$, which means that each agent performs Q steps of local computation between every two communication steps. This update strategy is related to the class of (horizontal) federated learning algorithms [13].

Case V ($\tau_\ell > \tau_g > 0$): We assume that $\tau_\ell = K \cdot \tau_g$, that the agents perform K steps of communication between two local computation steps. Although K can be arbitrary, in practice, it is typically chosen large enough so that certain network problem is solved approximately; therefore in practice, this case is closely related to Case II.

We summarize the above discussion in Table 2.1, and provide some example algorithms for each case. In Sec. 2.5.1, we will specify the controllers for these algorithms so that we can precisely map them to a discretization setting. It is important to note that the connection identified here is useful in helping predict algorithm performance, as well as facilitates new algorithm design.However, these benefits can be realized only if

there is a systematic way of transferring the theoretical results from the continuous-time system to different discretization settings. This will be discussed in detail in the next subsection.

Case	$\tau_\ell, \tau_{\mathbf{g}}$	**Comm.**	**Comp.**	**Related Algorithm**
I	$\tau_g > 0, \tau_\ell = 0$	Slow	Continuous	NEXT [11], FedProx [16], NIDS [74]
II	$\tau_g = 0, \tau_\ell > 0$	Continuous	Slow	MSDA [19], xFilter [20],AGD [70]
III	$\tau_g = \tau_\ell > 0$	Same rate		DGD [8], DGT [10]
IV	$\tau_g > \tau_\ell > 0$	Slow	Fast	Local GD [14], Scaffold [17]
V	$\tau_\ell > \tau_g > 0$	Fast	Slow	Same as Case II

Table 2.1: Summary of discretization settings and the corresponding distributed algorithms.

2.4.3 Convergence of Discretized Systems

Next, we leverage the convergence results of the continuous-time system to analyze distributed algorithms. The key challenge is to properly deal with the potential instability introduced by discretization. The proof of this subsection is relegated to Appendix A.1.1 – A.2.

Discretized Communication ($\tau_g > 0, \tau_\ell = 0$, Case I). Recall that the system dynamics are given in (2.14). Let us first show how the sampling error affects $\dot{\mathcal{E}}$.

Lemma 1 ($\dot{\mathcal{E}}$ in Case I) *Suppose the GCFL and LCFL satisfy P1-P5, and consider the discretized system with $\tau_\ell = 0, \tau_g > 0$. Then we have the following:*

$$\begin{aligned}\int_0^t \dot{\mathcal{E}}(\tau)\mathrm{d}\tau \leq \int_0^t -\underbrace{(\gamma_1(\tau) - C_{11})}_{:=\hat{\gamma}_1(\tau)} \cdot \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2 \,\mathrm{d}\tau \\ + \int_0^t -\underbrace{\left(\frac{\gamma_2(\tau)}{2} - C_{11}\right)}_{:=\hat{\gamma}_2(\tau)} \cdot \|(I-R)\cdot \mathbf{y}(\tau)\|^2 \,\mathrm{d}\tau,\end{aligned} \tag{2.18}$$

where $C_{11} := \frac{q_{\max}^2}{2\gamma_2(\tau)}$ *and* $q_{\max} := \exp\left\{\sqrt{2}\tau_g \cdot \left(\sqrt{C_x^2 + C_v^2}\eta_\ell(t) \cdot \left(1 + \frac{L_f}{N}\right)^2\right)\right\} - 1.$

The lemma shows that discretizing the communication with sufficiently small τ_g leads to a small $q_{\max}$, which preserves the desired descent property.

Discretized Computation ($\tau_\ell > 0, \tau_g = 0$, Case II). Recall that the system dynamics can be expressed in (2.16). We have the following result:

Lemma 2 ($\dot{\mathcal{E}}$ in Case II) *Suppose the GCFL and LCFL satisfy P1-P5, and consider the discretized system with $\tau_g = 0, \tau_\ell > 0$. Then we have the following:*

$$\begin{aligned}\int_0^t \dot{\mathcal{E}}(\tau)\mathrm{d}\tau \leq \int_0^t -\underbrace{\left(\frac{\gamma_1(\tau)}{2} - C_{21}\right)}_{:=\hat{\gamma}_1(\tau)} \cdot \left\|\nabla f(\bar{\mathbf{x}}(\tau))\right\|^2 \mathrm{d}\tau \\ + \int_0^t -\underbrace{\left(\frac{\gamma_2(\tau)}{2} - C_{22}\right)}_{:=\hat{\gamma}_2(\tau)} \cdot \left\|(I-R)\cdot \mathbf{y}(\tau)\right\|^2 \mathrm{d}\tau,\end{aligned} \tag{2.19}$$

where we have defined:

$$\begin{aligned}&C_{21} := \frac{4L^2 C_f C_\ell^2 \eta_\ell^2(\tau)}{2(1-2L^2C_\ell^2)\cdot \min\{N\gamma_1(\tau), \gamma_2(\tau)\}},\ C_{22} := \frac{L^2\eta_\ell^2(\tau)\cdot\left(\left(\frac{1-C_y}{C_y^2}\right) + 4L_f^2 C_f C_\ell^2\right)}{2(1-2L^2C_\ell^2)\cdot \min\{N\gamma_1(\tau), \gamma_2(\tau)\}}, \\ &C_f := C_x^2 + C_v^2 + C_z^2,\ C_y = \mathrm{e}^{-C_g\tau_\ell\eta_g(\tau)}, \quad C_\ell := \frac{\tau_\ell\eta_\ell(\tau)}{\min\{2C_g\eta_g(\tau), 1\}}.\end{aligned}$$

Note that the requirements on $\hat{\gamma}_1(\tau) > 0, \hat{\gamma}_2(\tau) > 0$ result in the constraint on τ_ℓ, which will be discussed at the end of this section.

Two-sided Discretization ($\tau_\ell > 0, \tau_g > 0$, Case III-V). We then analyze the more challenging cases where *both* the communication and the computation are discretized. Note that Case III with $\tau_\ell = \tau_g > 0$ can be merged into Case IV, with $Q = 1$.

Lemma 3 ($\dot{\mathcal{E}}$ in Case III-IV) *Suppose the GCFL and LCFL satisfy properties P1-P5, and consider the discretized system with $\tau_g = Q\cdot\tau_\ell$. Then we have:*

$$\begin{aligned}\int_0^t \dot{\mathcal{E}}(\tau)\mathrm{d}\tau \leq \int_0^t -\underbrace{\left(\frac{\gamma_1(\tau)}{2} - C_{41}(\tau)\right)}_{:=\hat{\gamma}_1(\tau)} \cdot \left\|\nabla f(\bar{\mathbf{x}}(\tau))\right\|^2 \mathrm{d}\tau \\ + \int_0^t -\underbrace{\left(\frac{\gamma_2(\tau)}{2} - C_{42}(\tau)\right)}_{:=\hat{\gamma}_2(\tau)} \cdot \left\|(I-R)\cdot \mathbf{y}(\tau)\right\|^2 \mathrm{d}\tau,\end{aligned} \tag{2.20}$$

where the constants $C_{41}(\tau)$ and $C_{42}(\tau)$ are defined as:

$$
\begin{aligned}
C_{41} &:= \frac{L^2\eta_\ell^2(\tau)\cdot\left(C_{45}\cdot(1+L_f^2C_{47}+C_{45})+C_{46}L_f^2\right)}{2\min\{N\gamma_1(\tau),\gamma_2(\tau)\}}+\frac{C_g\eta_g^2(\tau)\cdot(C_{43}+L_f^2C_{47})}{2\gamma_2(\tau)},\\
C_{42} &:= \frac{L^2\eta_\ell^2(\tau)\cdot(C_{46}+C_{45}C_{47})}{2\min\{N\gamma_1(\tau),\gamma_2(\tau)\}}+\frac{C_g\eta_g^2(\tau)C_{47}}{2\gamma_2(\tau)},\quad C_{47}:=Q^2C_{44}^2\cdot(C_x^2+C_v^2),\\
C_{43} &:= \frac{4\tau_g^2\eta_g^2(t)}{1-4\tau_g^2\eta_g^2(\tau)},\ C_{44}:=\frac{2\tau_\ell^2\tau_\ell^2(\tau)}{1-4\tau_g^2\eta_g^2(\tau)},\\
C_{45} &:= \frac{4\tau_\ell^2\eta_g^2(\tau)}{1-4L^2\tau_\ell^2\eta_\ell^2(\tau)},\ C_{46}:=\frac{8L^2C_f\tau_\ell^2\eta_\ell^2(\tau)}{1-4L^2\tau_\ell^2\eta_\ell^2(\tau)}.
\end{aligned}
$$

Furthermore, we can check that when $\tau_g=0$ and $\tau_\ell=0$, then $C_{41}(\tau)$, $C_{42}(\tau)$ are both zero. Additionally, $\hat{\gamma}_1(\tau)>0,\hat{\gamma}_2(\tau)>0$ determine the upper bounds for τ_g,τ_ℓ, as well as the choice of the stepsizes of the discretized algorithms.

Finally, we note that for Case V, a similar result with different $\hat{\gamma}_1(\tau),\hat{\gamma}_2(\tau)$ can be proved using the same technique as Lemma 2 and Lemma 3. Since the utility of Case V can be covered mostly by that of Case II (cf. Table 2.1), and due to the space limitation, we will not discuss this case in detail here.

By using the above results, it is easy to obtain the following convergence characterization. The proof is straightforward and follows that of Theorem 1.

Theorem 2 (Convergence of the discretized systems) *Suppose the GCFL and LCFL satisfy properties P1-P5, and consider the discretized system with $\tau_\ell\geq 0,\tau_g\geq 0$. Then the convergence of the discretized system can be characterized as:*

$$
\min_t\left\{\|\nabla f(\bar{\mathbf{x}}(t))\|^2+\|(I-R)\cdot\mathbf{y}(t)\|^2\right\}=\mathcal{O}\left(\max\left\{\frac{1}{\int_0^T\hat{\gamma}_1(\tau)\mathrm{d}\tau},\frac{1}{\int_0^T\hat{\gamma}_2(\tau)\mathrm{d}\tau}\right\}\right),
$$

where $\hat{\gamma}_1(\tau)>0$ and $\hat{\gamma}_2(\tau)>0$ depend on $\gamma_1(\tau),\gamma_2(\tau),N,C_g,L$ and $\eta_\ell,\eta_g,\tau_\ell,\tau_g,K,Q$, and their choices are specified in Lemmas 1 – 3.

This result indicates that as long as $\hat{\gamma}_1(\tau)>0$ and $\hat{\gamma}_2(\tau)>0$, the discretized system preserves the convergence rate of the continuous-time system, but it slows down by a factor $\max\left\{\gamma_1(\tau)/\hat{\gamma}_1(\tau),\gamma_2(\tau)/\hat{\gamma}_2(\tau)\right\}$. Further, the condition that $\hat{\gamma}_1(\tau)>0,\hat{\gamma}_2(\tau)>0$ give a way to decide the maximum sampling intervals and the choice of the hyperparameters (e.g., stepsize, the number of communication steps and local update steps K,Q) for different algorithms, as we explain below.

Let us consider Case I first. By Lemma 1,

$$\min\{\gamma_2, 2\gamma_1\} \geq \frac{q_{\max}^2}{\gamma_2}, \text{ with } q_{\max} = \mathrm{e}^{\sqrt{2}\tau_g \cdot \left(\sqrt{C_x^2+C_v^2}\eta_\ell(t)\cdot\left(1+\frac{L_f}{N}\right)^2\right)} - 1.$$

It follows that $\tau_g \leq \frac{\ln(\min\{\gamma_2(t), \sqrt{2\gamma_1(t)\cdot\gamma_2(t)}\}+1)}{\sqrt{2}\sqrt{C_x+C_v}\eta_\ell(t)\cdot\left(\frac{L_f}{N}+1\right)^2}$. Note that all the variables on the right hand side (RHS) can be determined from the continuous-time system. This indicates that by having a convergent continuous-time system, the maximum sampling interval of the GCFL can be determined. Similarly, for Case II, by Lemma 2, $\gamma_1(t) \geq 2C_{21},\ \gamma_2(t) \geq 2C_{22}$, which implies:

$$\tau_\ell \leq \min\left\{\frac{\tilde{\gamma}_1(t)}{\sqrt{2(\tilde{\gamma}_1^2(t)+4C_f)}L\eta_\ell^2(t)}, \frac{\log\left(\frac{\tilde{\gamma}_2(t)+2L\eta_\ell(t)}{2L\eta_\ell(t)}\right)}{C_g\eta_g(t)}\right\},$$

where $\tilde{\gamma}_1^2(t) := \min\{N\gamma_1^2(t), \gamma_1(t)\cdot\gamma_2(t)\},\ \tilde{\gamma}_2^2(t) := \min\{\gamma_2^2(t), N\gamma_1(t)\cdot\gamma_2(t)\}$. All the variables on the RHS can be determined from the continuous-time system, so the maximum sampling interval of the LCFL can be determined.

For Case III-IV, it requires $2C_{41} \leq \gamma_1(t), 2C_{42} \leq \gamma_2(t)$ and $\{C_{4i}\}_{i=3}^6$ to be positive. It may be difficult to obtain the exact bound for τ_g, τ_ℓ and Q, but we can derive an approximate bound on these parameters. For $\{C_{4i}\}_{i=3}^6$ to be positive, it requires $\tau_\ell \leq \frac{1}{2L\eta_\ell(t)}$, $\tau_g \leq \frac{1}{2\eta_g(t)}$. Set $\tau_\ell = \frac{c}{2L\eta_\ell(t)}, \tau_g = \frac{c}{2\eta_g(t)}$ for some $c < 1$. By choosing

$$c^2 < \min\left\{\frac{1}{4}, \min\{\tilde{\gamma}_1^2(t), \tilde{\gamma}_2^2(t)\}\right\} \cdot \min\left\{\frac{1}{L^2\eta_\ell(t)^2\cdot(1+L_f^2)}, \frac{1}{C_g\eta_g^2(t)}\right\}, \tag{2.21}$$

we have $C_{41} = \mathcal{O}(\gamma_1(t)), C_{42} = \mathcal{O}(\gamma_2(t))$. In addition, $Q = \tau_g/\tau_\ell \approx \frac{2L\eta_\ell(t)}{\eta_g(t)}$.

2.5 Application of the Framework

In this section, we discuss some applications of the proposed framework. We first show that by properly choosing the controllers and the discretization scheme, the multi-rate feedback control system can be specialized to a number of popular distributed algorithms. Due to space limitations, we relegate the discussion some additional algorithms to appendix Appendix A.3. Second, we show how the proposed framework can help identify the relationship between different algorithms. Finally, we use DGT as an example to show how the framework can be used to streamline the convergence analysis of a series of algorithms, as well as to facilitate the development of new ones.

2.5.1 A New Interpretation of Distributed Algorithms

In this part, we map some popular distributed algorithms to the discretized multi-rate systems, with specific GCFL and LCFL, and specific discretization setting. These mappings together provides a new perspective for understanding distributed algorithms.

Let us begin with mapping the decentralized optimization algorithms.

DGT [10]: The updates are given by:

$$\mathbf{x}(k+1) = W\mathbf{x}(k) - c\mathbf{v}(k),\ \mathbf{v}(k+1) = W\mathbf{v}(k) + \nabla f(\mathbf{x}(k+1)) - \nabla f(\mathbf{x}(k)), \quad (2.22)$$

where $c > 0$ is the stepsize. It corresponds to the discretization Case III with the following continuous-time controllers:

$$\begin{aligned} &u_{g,x} = (I - W)\cdot \mathbf{x}, \quad u_{g,v} = (I - W)\cdot \mathbf{v}, \\ &u_{\ell,x} = c\mathbf{v}, \quad u_{\ell,v} = -\nabla f(\mathbf{x}) + \nabla f(\mathbf{z}), \quad u_{\ell,z} = \mathbf{z} - \mathbf{x}. \end{aligned} \quad (2.23)$$

NEXT [11]: The updates of NEXT in discrete time are:

$$\begin{aligned} \mathbf{x}(k+1/2) &= \arg\min_{\mathbf{x}} \tilde{f}(\mathbf{x}; \mathbf{x}(k)) + \langle N\mathbf{v}(k) - \nabla f(\mathbf{x}(k)), \mathbf{x} - \mathbf{x}(k)\rangle\,, \\ \mathbf{x}(k+1) &= W\left(\mathbf{x}(k) + \alpha\cdot(\mathbf{x}(k+1/2) - \mathbf{x}(k))\right), \\ \mathbf{v}(k+1) &= W\mathbf{v}(k) + \nabla f(\mathbf{x}(k+1)) - \mathbf{z}(k), \quad \mathbf{z}(k+1) = \nabla f(\mathbf{x}(k+1)), \end{aligned}$$

where $\tilde{f}$ is some surrogate function; k indicates the iteration index; $\alpha > 0$ and $c > 0$ are some stepsize parameters. By using the common choice that $\tilde{f}(\mathbf{x}; \mathbf{x}(k)) = \langle \nabla f(\mathbf{x}(k)), \mathbf{x} - \mathbf{x}(k)\rangle + \frac{\eta}{2}\left\|\mathbf{x} - \mathbf{x}(k)\right\|^2$, (where $\eta > 0$ are some constant) the algorithm can be simplified as:

$$\begin{aligned} &\mathbf{x}(k+1) = W\mathbf{x}(k) - N\alpha/\eta\cdot\mathbf{v}(k), \quad \mathbf{z}(k+1) = \mathbf{x}(k+1), \\ &\mathbf{v}(k+1) = W\mathbf{v}(k) + \nabla f(\mathbf{x}(k+1)) - \nabla f(\mathbf{z}(k)). \end{aligned} \quad (2.24)$$

Here, $\mathbf{x}$ is the optimization variable, $\mathbf{v}$ tracks the average of the gradients, $\mathbf{z}$ records the one-step-behind state of $\mathbf{x}$. It corresponds to Case III, with the continuous-time controllers given by:

$$G_g(\mathbf{x}, \mathbf{v}; A) := \begin{bmatrix} (I - W)\cdot\mathbf{x} \\ (I - W)\cdot\mathbf{v} \end{bmatrix}, \quad G_\ell(x_i, v_i, z_i; f_i) := \begin{bmatrix} v_i \\ \nabla f_i(z_i) - \nabla f_i(x_i) \\ z_i - x_i \end{bmatrix}. \quad (2.25)$$

Next, we discuss two popular federated learning algorithms. In this class of algorithms, the agents are connected with a central server which performs averaging. So the communication graph is a fully connected graph, with the weight matrix being the averaging matrix, i.e., $W = R$, $W_A = I - R$.

FedAvg [13]: The updates are given by (where GD is used for the local steps):

$$\mathbf{x}(k+1) = \begin{cases} R\mathbf{x}(k) - \eta \nabla f(\mathbf{x}(k)), & k \bmod Q = 0, \\ \mathbf{x}(k) - \eta \nabla f(\mathbf{x}(k)), & k \bmod Q \neq 0. \end{cases}$$

This algorithm has the following continuous-time controller:

$$u_{g,x} = \sum_{k=0}^{\infty} \delta(t - k\tau_g) \cdot (I - R) \cdot \mathbf{x}(t) \tag{2.26}$$

where $\delta(t)$ denotes the Dirac delta function. It is interesting to note that FedAvg *cannot* be mapped to a continuous-time double-feedback system, as it does not have a *persistent* GCFL (it is only activated when $t = k\tau_g$; see (2.26)). This partially explains why FedAvg algorithm requires additional assumptions for convergence.

Scaffold [17]: The updates are given by (where $k_0 := k - (k \bmod K)$):

$$\begin{aligned}
\mathbf{x}(k+1) &= \begin{cases} \mathbf{x}(k) - \eta_1 \cdot (\nabla f(\mathbf{x}(k)) - \mathbf{z}(k) + \mathbf{v}(k_0)) - \eta_2 \cdot (\mathbf{x}(k) - \mathbf{w}(k)), (k \bmod Q) = 0, \\ \mathbf{x}(k) - \eta_1 \cdot (\nabla f(\mathbf{x}(k)) - \mathbf{z}(k) + \mathbf{v}(k_0)), \ (k \bmod Q) \neq 0. \end{cases} \\
\mathbf{v}(k+1) &= \begin{cases} \mathbf{v}(k) - R \cdot (\mathbf{v}(k) + \frac{1}{Q\eta_1} \cdot (\mathbf{w}(k) - \mathbf{x}(k))), & k \bmod Q = 0 \\ \mathbf{v}(k), & k \bmod Q \neq 0, \end{cases} \\
\mathbf{w}(k+1) &= \begin{cases} R\mathbf{x}(k) & k \bmod Q = 0 \\ \mathbf{w}(k), & k \bmod Q \neq 0, \end{cases} \\
\mathbf{z}(k+1) &= \mathbf{z}(k) - \frac{1}{Q}\mathbf{v}(k) - \frac{1}{Q\eta_1} \cdot (\mathbf{x}(k+1) - \mathbf{x}(k)).
\end{aligned}$$

So it uses the discretization Case IV. Observe that $\mathbf{w}$ tracks $R\mathbf{x}$, so in continuous-time we have: $\mathbf{x} - \mathbf{w} = (I - R) \cdot \mathbf{x} + (R\mathbf{x} - \mathbf{w}) = (I - R) \cdot \mathbf{x} + R\dot{\mathbf{x}}$. Then we can replace $\mathbf{w}$ by $R \cdot (\mathbf{x} - \dot{\mathbf{x}})$, and obtain the continuous-time controller as:

$$\begin{aligned}
&u_{g,x} = \eta_2 \cdot (I - R) \cdot \mathbf{x} + \eta_1 \mathbf{v} + \eta_2 R\dot{\mathbf{x}}, \quad u_{g,v} = -(I - R) \cdot (\mathbf{v} + \dot{\mathbf{x}}/\eta_1), \\
&u_{\ell,x} = \nabla f(\mathbf{x}) - \mathbf{z}, \quad u_{\ell,v} = \mathbf{v} + \dot{\mathbf{x}}/\eta_1, \quad u_{\ell,z} = \mathbf{v} + \dot{\mathbf{x}}/\eta_1.
\end{aligned} \tag{2.27}$$

Finally, we discuss one accelerated consensus algorithm:

xFilter [20]: The updates are given by (where $k_0 := k - (k \bmod K)$):

$$\begin{aligned}
\mathbf{x}(k+1) &= \eta_1 \cdot ((1-\eta_2)I - \eta_2 \cdot (I-W)) \cdot \mathbf{x}(k) + (1-\eta_1) \cdot \mathbf{x}(k-1) + \eta_1\eta_2 \mathbf{v}(k_0) \\
&= \mathbf{x}(k) - \eta_1\eta_2 \cdot (2I - W)\mathbf{x}(k) - (1-\eta_1) \cdot (\mathbf{x}(k) - \mathbf{x}(k-1)) + \eta_1\eta_2 \mathbf{v}(k_0), \\
\mathbf{v}(k+1) &= \begin{cases} \mathbf{v}(k) + (\mathbf{w}_1(k) - \mathbf{w}_2(k)) - (I-W) \cdot \mathbf{x}(k), & k \bmod K = 0 \\ \mathbf{v}(k), & k \bmod K \neq 0, \end{cases} \\
\mathbf{w}_1(k+1) &= \begin{cases} \mathbf{x}(k) - \eta_3 \nabla f(\mathbf{x}(k)), & k \bmod K = 0 \\ \mathbf{w}_1(k), & k \mod K \neq 0, \end{cases} \\
\mathbf{w}_2(k+1) &= \begin{cases} \mathbf{w}_1(k), & k \bmod K = 0 \\ \mathbf{w}_2(k), & k \bmod K \neq 0, \end{cases}
\end{aligned}$$

This algorithm uses the discretization Case V. We can see $\mathbf{w}_2$ tracks $\mathbf{w}_1$, and $\mathbf{w}_1$ tracks $\mathbf{x} - \eta_3 \nabla f(\mathbf{x})$, therefore in continuous-time we have $\mathbf{w}_1 - \mathbf{w}_2 = \dot{\mathbf{x}} - \eta_3 \cdot \dot{\nabla} f(\mathbf{x})$, with the following continuous-time system:

$$\begin{aligned}
\dot{\mathbf{x}} &= -\eta_1\eta_2 \cdot (2I - W) \cdot \mathbf{x} + \eta_1\eta_2 \mathbf{v} - (1-\eta_1) \cdot \dot{\mathbf{x}}, \\
\dot{\mathbf{v}} &= \dot{\mathbf{x}} - \eta_3 \dot{\nabla} f(\mathbf{x}) - (I-W) \cdot \mathbf{x}.
\end{aligned} \tag{2.28}$$

Integrating over time, and use the initialization that $\mathbf{v}(0) = \mathbf{x}(0) - \eta_3 \nabla f(\mathbf{x}(0))$, we have the following expression for $\mathbf{v}(t)$:

$$\mathbf{v}(t) = \int_0^t (\dot{\mathbf{x}}(\tau) - \eta_3 \dot{\nabla} f(\mathbf{x}(\tau)) - (I-W) \cdot \mathbf{x}(\tau)) \mathrm{d}\tau = \mathbf{x}(t) - \eta_3 \nabla f(\mathbf{x}(t)) - \int_0^t (I-W) \cdot \mathbf{x}(\tau) \mathrm{d}\tau.$$

Define $\mathbf{v}_1 = \frac{1}{2-\eta_1} \cdot (\mathbf{x} - \mathbf{v})$, $\mathbf{z} = \frac{\eta_3}{2-\eta_1} \nabla f(\mathbf{x})$, then (2.28) can be equivalently written as:

$$\begin{aligned}
\dot{\mathbf{x}} &= -\eta_1\eta_2 \cdot (I-W) \cdot \mathbf{x} - \eta_1\eta_2 \cdot (2-\eta_1) \cdot \mathbf{v}_1 - (1-\eta_1) \cdot \dot{\mathbf{x}}, \\
&= -\eta_1\eta_2 \cdot (I-W) \cdot \mathbf{x} - \eta_1\eta_2 \cdot (2-\eta_1) \cdot (\mathbf{v}_1 - \mathbf{z}) - (1-\eta_1) \cdot \dot{\mathbf{x}} - \eta_1\eta_2\eta_3 \nabla f(\mathbf{x}) \\
\dot{\mathbf{v}}_1 &= \frac{1}{2-\eta_1} \cdot (I-W) \cdot \mathbf{x} + \frac{\eta_3}{2-\eta_1} \dot{\nabla} f(\mathbf{x}), \quad \dot{\mathbf{z}} = \frac{\eta_3}{2-\eta_1} \dot{\nabla} f(\mathbf{x}).
\end{aligned}$$

The dynamic of $\dot{\mathbf{x}}$ implies $\frac{1}{2-\eta_1}(I-R) \cdot (I-W) \cdot \mathbf{x} = -(I-R) \cdot \left(\mathbf{v}_1 + \frac{1}{\eta_1\eta_2} \dot{\mathbf{x}} \right)$, where $(I-R) \cdot (I-W) = (I-W)$ by P1. Substituting this into $\dot{\mathbf{v}}_1$, defining $\eta_4 := \eta_1\eta_2, \eta_5 := (2-\eta_1), \eta_6 := \eta_1\eta_2\eta_3$, and rearranging the terms, we obtain the following equivalent

controller:

$$u_{g,x} = \eta_4 \cdot (I - W) \cdot \mathbf{x} + \eta_4\eta_5\mathbf{v}_1 + (\eta_5 - 1) \cdot \dot{\mathbf{x}}, \quad u_{g,v} = -(I - R) \cdot (\mathbf{v}_1 + \dot{\mathbf{x}}/\eta_4),$$
$$u_{\ell,x} = \eta_6\nabla f(\mathbf{x}) - \eta_4\eta_5\mathbf{z}, \quad u_{\ell,v} = \frac{\eta_3}{\eta_5}\dot{\nabla} f(\mathbf{x}), \quad u_{\ell,z} = \frac{\eta_3}{\eta_5}\dot{\nabla} f(\mathbf{x}).$$

Interestingly, the above dynamics is close to those of Scaffold in (2.27), except that Scaffold uses R instead of W, a different stepsize, and use $R\dot{\mathbf{x}}$ in $u_{g,x}$ instead of $\dot{\mathbf{x}}$.

2.5.2 Algorithms Connections

We summarize the discussion in the previous subsection in Table 2.2. It is interesting to observe that, some seemingly unrelated algorithms, in fact are very closely related in continuous-time. For example, somewhat surprisingly, Scaffold and xFilter share very similar continuous-time dynamics, although they are designed for very different purposes: the former is designed to improve FedAvg algorithm to better deal with data heterogeneity, while the latter is a primal-dual algorithm designed to achieve the optimal graph dependency. Similarly, each pair of algorithms FedPD and DLM, FedProx and DGD shares the same continuous-time dynamics (these algorithms are discussed in detail in Appendix A.3). The latter two relations are relatively easier to identify. For example, FedPD and DLM are in fact designed from the same primal-dual perspective.

GCFL	LCFL	FL	AC	DO
$(I - W) \cdot \mathbf{x}$	$\nabla f(\mathbf{x})$	FedProx	–	DGD
$(I - W) \cdot \mathbf{y}$	$-\nabla f(\mathbf{x}) + \nabla f(\mathbf{z})$	–	–	DGT, NEXT
$c \cdot (I - W) \cdot \mathbf{x} + \mathbf{v}$	$\nabla f(\mathbf{x})$	FedPD	–	DLM
$(I - W) \cdot \mathbf{x} + \eta\mathbf{v} + R\dot{\mathbf{x}}$	$\nabla f(\mathbf{x}) - \mathbf{z}$	Scaffold	–	–
$(I - W) \cdot \mathbf{x} + \eta\mathbf{v} + \dot{\mathbf{x}}$	$\nabla f(\mathbf{x}) - \mathbf{z}$	–	xFilter	–

Table 2.2: A summary of the controllers used in different algorithms. In GCFL and LCFL we abstract the most important steps of the controller.

Additionally, from the table we can see that there are a few missing entries. Each of these entries represents a new algorithm. Also, we can combine different GCFLs and LCFLs, or design new controllers, to create new control systems (hence algorithms) that are not included in this table.

2.5.3 Convergence Analysis and Algorithm Design: A Case Study

In this subsection, we use the DGT algorithm as an example to illustrate how our proposed framework can be used in practice to analyze algorithm behavior, and to facilitate the development of new algorithms.

The iteration of the DGT is given in (2.22). Under A1 – A3, this algorithm converges to the stationary point of the problem at a rate of $\mathcal{O}(1/T)$ [25, 75]. To use our framework to analyze it, we will first construct a continuous-time double-feedback system, apply the discretization scheme III, and finally leverage Lemma 3 and Theorem 2 to obtain the convergence rate.

Continuous-time Analysis

We begin by analyzing the continuous-time counterpart of the DGT, whose dynamics, according to (2.23), is given by:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= -\eta_g(t)\cdot(I-W)\cdot\mathbf{x}(t) - \eta_\ell(t)\cdot(c\mathbf{v}(t)), \quad \dot{\mathbf{z}}(t) = -\eta_\ell(t)\cdot(\mathbf{z}(t)-\mathbf{x}(t))\\ \dot{\mathbf{v}}(t) &= -\eta_g(t)\cdot(I-W)\cdot\mathbf{v}(t) + \eta_\ell(t)\cdot(\nabla f(\mathbf{x}(t)) - \nabla f(\mathbf{z}(t)))\end{aligned} \tag{2.29}$$

where $\eta_g(t) = 1, \eta_\ell(t) = 1, \forall\ t$.

Let us verify properties P1-P5. First, it is easy to prove P2 with the definition of u_g given in (2.23). To show P1, recall that we have defined $W := I - A^T\mathrm{diag}(\mathbf{w})\mathrm{A}$, so it is easy to verify that $\mathbb{1}^T\cdot(I-W) = \mathbb{1}^T\cdot A^T\mathrm{diag}(\mathbf{w})\mathrm{A} = 0$ and $C_g = 1-\lambda_2(W)$.

To show P3, we have the following bounds for different parts of the local controller:

$$\begin{aligned}\|G_{\ell,x}(x_i,v_i,z_i;f_i) - G_{\ell,x}(x_i',v_i',z_i';f_i)\| &= \|c(v_i-v_i')\| = c\,\|v_i-v_i'\|\\ \|G_{\ell,v}(x_i,v_i,z_i;f_i) - G_{\ell,v}(x_i',v_i',z_i';f_i)\| &= \|\nabla f_i(x_i) - \nabla f_i(z_i) - \nabla f_i(x_i') + \nabla f_i(z_i')\|\\ &\le \|\nabla f_i(x_i) - \nabla f_i(x_i')\| + \|\nabla f_i(z_i) - \nabla f_i(z_i')\|\\ &\le L_f(\|x_i-x_i'\| + \|z_i-z_i'\|)\\ \|G_{\ell,z}(x_i,v_i,z_i;f_i) - G_{\ell,z}(x_i',v_i',z_i';f_i)\| &= \|x_i-z_i-x_i'+z_i'\| \le \|x_i-x_i'\| + \|z_i-z_i'\|,\end{aligned}$$

where L_f is the constant of the Lipschitz gradient in A2. So the smoothness constant of the local controller g_ℓ can be expressed as $L = \max\{L_f, c, 1\}$.

To verify P4, let us initialize $\mathbf{v}(t) = \nabla f(\mathbf{x}(t)), \mathbf{z}(t) = \mathbf{x}(t)$, and assume that $\eta_g(t) = 0$

in (2.29), that is, the GCFL is inactive. Then we have:

$$\begin{aligned}\mathbf{z}(t+\tau) &= \mathbf{x}(t+\tau),\ \mathbf{v}(t+\tau) = \nabla f(\mathbf{x}(t+\tau)),\\ \dot{\mathbf{x}}(t+\tau) &= -c\mathbf{v}(t+\tau) = -c\nabla f(\mathbf{x}(t+\tau)).\end{aligned} \tag{2.30}$$

Further, we can verify that the output of the LCFL can be bounded by

$$\begin{aligned}\|u_{i,\ell,x}(t)\| &= \|c\cdot v_i(t)\| = c\,\|\nabla f_i(x_i(t))\|\\ \|u_{i,\ell,v}(t)\| &= \|\nabla f_i(x_i(t)) - \nabla f_i(z_i(t))\| \le 2\,\|\nabla f_i(x_i(t))\|\\ \|u_{i,\ell,z}(t)\| &= \|z_i(t) - x_i(t)\| = \|c\cdot v_i(t)\| = c\,\|\nabla f_i(x_i(t))\|\,.\end{aligned}$$

The algorithm becomes the gradient flow algorithm that satisfies P4 with $\alpha(t) = c$, $C_x = c, C_v \le 2, C_z = c$. Finally, we verify P5. We can compute $\dot{\mathcal{E}}(t)$ as follows:

$$\begin{aligned}\dot{\mathcal{E}}(t) &= -\left\langle \nabla f(\bar{\mathbf{x}}(t)), \frac{1}{N}\sum_{i=1}^N u_{\ell,x}(t)\right\rangle - \langle (I-R)\cdot\mathbf{y}(t), u_{g,y}(t) + u_{\ell,y}(t)\rangle\\ &\overset{(2.23)}{=} -\langle \nabla f(\bar{\mathbf{x}}(t)), c\bar{\mathbf{v}}(t)\rangle - \langle (I-R)\cdot\mathbf{y}(t), (I-W)\cdot\mathbf{y}(t)\rangle\\ &\quad - \langle (I-R)\cdot\mathbf{x}(t), c\mathbf{v}(t)\rangle + \langle (I-R)\cdot\mathbf{v}(t), \nabla f(\mathbf{x}(t)) - \nabla f(\mathbf{z}(t))\rangle\,.\end{aligned} \tag{2.31}$$

Then we bound each term on the RHS above separately, and finally integrate it. The detailed derivation is relegated to Appendix A.5 The final bound we can obtain is:

$$\begin{aligned}\int_0^t \dot{\mathcal{E}} \le &-\frac{c}{2}\int_0^t \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2\,\mathrm{d}\tau - \frac{c - 8L_f c^2/\beta}{2}\int_0^t \|\bar{\mathbf{v}}(\tau)\|^2\,\mathrm{d}\tau\\ &- (C_g - \frac{c + 2cL_f + \beta + 16cL_f/\beta}{2})\cdot\int_0^t \|(I-R)\cdot\mathbf{y}(\tau)\|^2\,\mathrm{d}\tau.\end{aligned}$$

By choosing $\beta < C_g/2$, $\frac{C_g^2}{64L_f} \le c \le \frac{C_g^2}{32L_f}$, we can verify that the dynamics of the continuous-time system (2.29) satisfy (2.11), with $\gamma_1(t) \ge \frac{C_g^2}{128L_f}$ and $\gamma_2(t) \ge \frac{C_g}{4}$. Applying Theorem 1, we know that continuous-time gradient tracking algorithm converges in $\mathcal{O}(1/T)$.

New Algorithm Design

Now that we have verified properties P1-P5 for the continuous-time system (2.29), we can derive a number of related algorithms by adjusting the discretization schemes, or by changing the GCFL.

Let us first consider changing the discretization scheme from Case III to Case IV, where $\tau_g = Q\tau_\ell > 0$. In this case, there will be Q local computation steps between every two communication steps. This kind of update scheme is closely related to algorithms in FL, and we refer to the

resulting algorithm the Decentralized Federated Gradient Tracking (D-FedGT) algorithm. Its steps are listed below (where $k_0 = k - (k \bmod Q)$):

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{x}(k) - \tau_\ell \mathbf{v}(k) - \tau_g(I-W)\mathbf{x}(k_0),\\ \mathbf{v}(k+1) &= \mathbf{v}(k) + \nabla f(\mathbf{x}(k+1)) - \nabla f(\mathbf{x}_k) - \tau_g(I-W)\mathbf{v}(k_0).\end{aligned} \tag{2.32}$$

By applying Lemma 3 and Theorem 2, we can directly obtain that this new algorithm also converges with rate $\mathcal{O}(\frac{1}{T})$ with properly chosen constant τ_ℓ, τ_g and Q following Lemma 3 and (2.21).

Second, we can replace the GCFL of the DGT with an *accelerated* consensus controller [67]. This leads to a new Accelerated Gradient Tracking (AGT) algorithm:

$$\begin{aligned}&\mathbf{x}(k+1) = \mathbf{x}(k) - \eta'_\ell \mathbf{v}(k) - \eta'_g(1+c)\mathbf{x}(k) + c\mathbf{v}_x(k),\\ &\mathbf{v}(k+1) = \mathbf{v}(k) + \nabla f(\mathbf{x}(k+1)) - \nabla f(\mathbf{x}(k)) - \eta_g(1+c)\mathbf{v}(k) + c\mathbf{v}_v(k),\\ &\mathbf{v}_x(k+1) = \mathbf{x}(k), \quad \mathbf{v}_v(k+1) = \mathbf{v}(k), \ \text{ where } c := \frac{1-\sqrt{1-\lambda_2(W)}}{1+\sqrt{1-\lambda_2(W)^2}}.\end{aligned} \tag{2.33}$$

Then by examining P1, we know that the network dependency of the new algorithm improved from C_g to $\hat{C}_g = C_g \cdot \frac{\sqrt{C_g}+\sqrt{2-C_g}}{\sqrt{C_g}+C_g\sqrt{2-C_g}} > C_g$. And when C_g is small, $\hat{C}_g$ scales with $\sqrt{C_g}$. Then, according to the derivation in the last subsection, we have $\gamma_2(t) \geq \frac{\hat{C}_g}{4}$. Finally, we can apply Theorem 2, and assert that the new algorithm improves the convergence speed from $\mathcal{O}(\frac{1}{C_g T})$ to $\mathcal{O}(\frac{1}{\hat{C}_g T})$.

Numerical Results

We provide numerical results for implementations of Continuous-time (CT) DGT, the D-FedGT, and D-AGT algorithms discussed in the previous subsection. We first verify an observation from Theorem 2, that discretization slows down the convergence speed of the system. Towards this end, we conduct numerical experiments with different discretization patterns and compare the convergence speed in terms of the stationarity gap. Then we compare the convergence speed of CT-DGT and CT-AGT, to demonstrate the benefit of changing the controller in the GCFL from the standard consensus controller to the accelerated one.

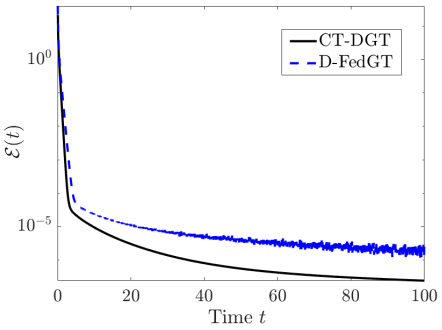
In the experiments, we consider the non-convex regularized logistic regression problem:

$$f_i(\mathbf{x}; (\mathbf{a}_i, b_i)) = \log(1 + \exp(-b_i \mathbf{x}^T \mathbf{a}_i)) + \sum_{d=1}^{d_x} \frac{\beta\alpha(\mathbf{x}[d])^2}{1+\alpha(\mathbf{x}[d])^2},$$

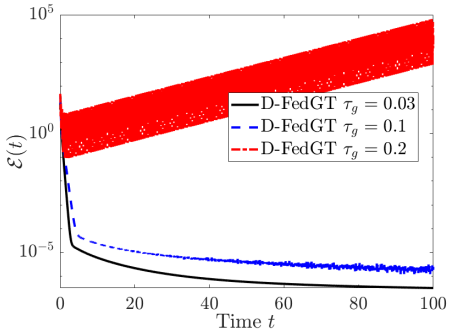
where $\mathbf{a}_i$ denotes the features and b_i denotes the labels of the dataset on the i^{th} agent. We set the number of agents $N = 20$, and each agent has a local dataset of size 500. We use an

Erdős–Rényi random graph with density 0.5 for the network and optimize the weight matrix W to achieve the optimal C_g. We set $c = 1$ for the gradient tracking algorithm.

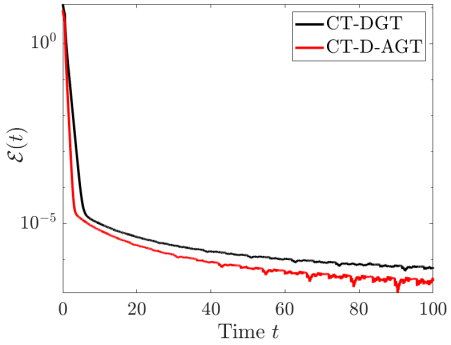
We first compare CT-DGT ($\tau_g = \tau_\ell = 0$) and D-FedGT ($\tau_g = 0.1, \tau_\ell = 0.005, Q = 20$), the result of CT-DGT and D-FedGT is showed in Figure 2.4a. We can see that by discretizing each loop, the system converges slower than the continuous time system. Figure 2.4b shows the convergence behavior of the D-FedGT algorithm with different τ_g. We observe that by increasing the sampling interval for GCFL, the convergence of the system slows down, and it eventually diverges. Figure 2.4c and Figure 2.4d show the convergence results of D-AGT compared with DGT in both continuous-time and in Case III. We observe that by changing the GCFL, D-AGT converges faster than DGT.



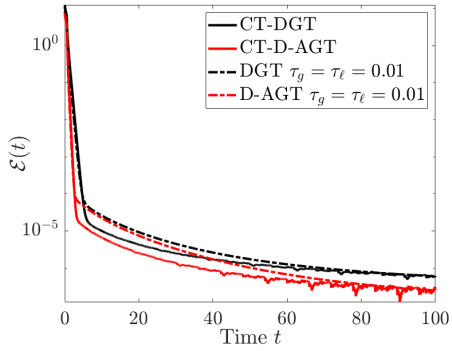
(a) The evoluation of the Energy function $\mathcal{E}(t)$ of CT-CGT, D-FedGT.



(b) Energy function $\mathcal{E}(t)$ of D-FedGT with different intervals τ_g.



(c) The evolution of the Energy function $\mathcal{E}(t)$ of CT-DGT and CT-D-AGT.



(d) The evolution of the Energy function $\mathcal{E}(t)$ of DGT and D-AGT.

Figure 2.4: The performance of Continuous-GT, D-FedGT, D-MGT and AGT.

Chapter 3

A Control-based Framework for Understanding Distributed Optimization Algorithms: Modeling Stochastic Algorithm

3.1 Motivation

Distributed optimization has played an important role in several traditional system-theoretic domains such as control and signal processing, and more recently, in machine learning (ML). Some contemporary applications where distributed optimization finds useful include large-scale decentralized neural network training, federated learning (FL), and multi-agent reinforcement learning. In a typical distributed optimization setting, the agents in the network jointly solve a system-level optimization problem, with the constraint that they only utilize local data, local computation, and local communication resources.

3.1.1 Design Considerations and Challenges

A few key design considerations for contemporary distributed algorithms are listed below:
Efficient Computation. Since local agents may contend with computational-resource and power limitations, it is desirable that they perform computation in a cost-effective manner. In practice, state-of-the-art distributed algorithms in ML applications typically utilize stochastic

gradient descent (SGD) based algorithm as their local computation engine [5]. So a key design consideration is to reduce the total number of data sample access, or equivalently, to improve sample efficiency.

Efficient Communication. Frequent inter-agent message exchanges can present several bottlenecks to system performance in addition to consuming power. In applications such as decentralized training (DT) and federated learning (FL), communication links may not have high enough bandwidth [13, 6]. Therefore, it is desirable that the local communication between the agents happen only when necessary, and when it happens, as little information is exchanged as possible.

Flexibility based on Practical System Considerations. Since distributed algorithms are often implemented in different environments, and they are used in applications across different domains, it is desirable that they are flexible and can take into consideration practical requirements (e.g., preserving user privacy), accommodate desired communication patterns, and allow for the possibility of agents participating occasionally [76, 77, 27].

Guaranteed Performance. The performance of distributed algorithms can be very different compared with their centralized counterpart, and if not designed carefully, distributed algorithms can diverge easily [3, 18]. So, it is important that algorithms offer convergence guarantees at a minimum. Further, it is desirable if such guarantees can characterize the efficiency in computation and communication.

There has been remarkably high interest in distributed algorithms in recent years across applications. These algorithms are typically developed in an application-specific manner. They are designed, for example, to: improve communication efficiency by utilizing model compression schemes [21, 22]; perform occasional communication [23, 24]; improve computational efficiency by utilizing SGD based schemes [3, 25]; understand the best possible communication and computation complexity [19, 26]; incorporate differential privacy (DP) guarantee into the system [27]; or to deal with the practical situation where even the (stochastic) gradients may not be accessible [28, 29].

Despite extensive research in distributed algorithms, several challenges persist in their synthesis and application. First, the proliferation of the algorithms indeed gives practitioners many alternatives to choose from. However, the *downside* is that there are simply too many algorithms available, so it becomes difficult to appreciate all underlying technical details and common themes linking them. Second, the current practice is that we need to design a new algorithm and develop the corresponding analysis for each particular application scenario (e.g., FL) with a specific set of requirements (e.g., communication efficiency + privacy). Given the combinatorial number of different applications and requirements, this general process readily becomes very tedious.

Therefore, we ask: Is it possible to have a generic "model" of distributed algorithms, which can abstract their important features (e.g., DP preserving mechanism, compressed communication, occasional communication) into tractable modules? If the answer is affirmative, can we design a framework that utilizes these abstract modules, unifies the analysis of (possibly a large subclass of) distributed algorithms, and subsequently facilitates the design of new ones?

A limited number of existing works have attempted to address these two questions, but the scope is still very restricted. Reference [78] focuses only on the DT algorithms with linear operators on the gradients and fails to cover the FL or stochastic settings. [77] only considers stochastic gradient descent in FL setting, which cannot generalize to any other algorithms. Other works related to continuous-time analysis of distributed algorithms, as well as using control theory to facilitate the design and analysis, are provided in Appendix B.1

3.2 Preliminaries

In this section, we introduce assumptions and notations leveraged in the remainder. First, we formally define the distributed optimization problem as minimizing a sum of smooth and possibly non-convex local loss functions on N agents [4]:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^{Nd_x}} \quad & f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f_i(x_i), \\ \text{s.t.} \quad & x_i = x_j,\ \forall\ (i,j) \in \mathbf{E}, \end{aligned} \tag{3.1}$$

where $\mathbf{x} \in \mathbb{R}^{Nd_x}$ stacks N local variables $\mathbf{x} := [x_1; \ldots; x_N]$, $x_i \in \mathbb{R}^{d_x}$, $\forall\, i \in [N]$, where we denote the set $[N] := \{1, \ldots, N\}$, and the agents are connected by a communication graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, which consists of a set $\mathbf{V}$ of agents indexed by $i \in [N]$, and a undirected edge set $\mathbf{E} \subset \mathbf{V} \times \mathbf{V}$. The incidence matrix $A \in \{-1, 0, 1\}^{|\mathbf{E}|\times|\mathbf{V}|}$ of graph $\mathcal{G}$ is defined as follows: if edge $(i, j) \in \mathbf{E}$ connects agent i, j with $i > j$, then $A_{(i,j),i} = 1$, $A_{(i,j),j} = -1$ and $A_{(i,j),k} = 0$, $\forall k \neq i, j$. The Laplacian matrix of the graph can be expressed as $\mathcal{L} = -A^T A$. We denote the length-n all-one vector by $\mathbb{1}_n$, averaging matrix $R := \frac{\mathbb{1}_N \mathbb{1}_N^T}{N}$, and identity matrix of dimension $N \times N$ by I. For simplicity of notation, we ignore the possible Kronecker products and vectorization when dealing with stacked vectors and matrices; for instance, we write the average of $\mathbf{x}$ as $\bar{\mathbf{x}} := \frac{\mathbb{1}_N^T}{N}\mathbf{x}$, the stacked local gradient as $\nabla f(\mathbf{x}) = [\nabla f_1(x_1); \ldots; \nabla f_N(x_N)]$, and the averaged gradient as $\nabla f(\bar{\mathbf{x}}) = \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(\bar{\mathbf{x}})$.

We make the following blanket assumptions on (3.1):

A 5 (Graph connectivity) *The union of the communication graphs over time $t \in [0, \infty)$ is connected, i.e., 0 is a simple eigenvalue its Laplacian matrix, with corresponding eigenvector $\frac{\mathbb{1}_N}{\sqrt{N}}$.*

A 6 (Lipschitz gradient) *The f_i's have Lipschitz gradient with constant L_f:*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_f \|x - y\|, \ \forall \ x, y \in \mathbb{R}^{d_x}, \forall \ i \in [N].$$

A 7 (Lower bounded functions) *The loss functions are lower bounded:*

$$f_i(x) \geq \underline{f}_i > -\infty, \quad \forall x \in \mathbb{R}^{d_x}, \quad \forall i \in [N],$$
$$f(\mathbf{x}) \geq f^\star \geq \sum_{i=1}^{N} \underline{f}_i, \quad \forall \mathbf{x} \in \mathbb{R}^{Nd_x},$$

where $f^\star$ is the infimum of $f(\mathbf{x})$.

Let us briefly comment on these assumptions. First, A5 is necessary for the problem (3.1) to be solved with distributed iterative methods, while allowing directed and/or not strongly connected time-varying communication graphs $\mathcal{G}(t)$. Subsequently, in Section 3.3, we will show that time-varying graphs can be related to many practical algorithm implementations. Second, A6 is a commonly used assumption for analyzing non-convex optimizations. We are interested in finding the (ϵ-accurate) first-order stationary points (FOSP) of the problem, which is defined as follows:

Definition 2 (FOSP, ϵ-stationary point) *The FOSP and ϵ-stationary point are defined respectively as:*

$$\nabla f(\bar{\mathbf{x}}) = 0, \quad (I - R) \cdot \mathbf{x} = 0, \tag{3.2a}$$
$$\|\nabla f(\bar{\mathbf{x}})\|^2 + \|(I - R) \cdot \mathbf{x}\|^2 \leq \epsilon. \tag{3.2b}$$

In addition, we refer to the left-hand-side (LHS) of (3.2b) as the stationarity gap of (3.1), $\|\nabla f(\bar{\mathbf{x}})\|^2$ as the convergence error, and $\|(I - R) \cdot \mathbf{x}\|^2$ as the consensus error.

To analyze stochastic systems, we define the expectation conditioning on all the information until time t as $\mathbb{E}_t[(\cdot)] := \mathbb{E}[(\cdot)|\text{information until } t]$, the variance as $\text{Var}_t(\cdot)$, and covariance as $\text{Cov}_t(\cdot, \cdot)$. Further, $\tilde{(\cdot)}$ denotes the stochastic version of the variables and functions.

3.3 System Description

In this section, we present the stochastic multi-rate feedback-control system that we propose to "model" distributed algorithms. We first develop a deterministic version of the system, discuss its properties, as well as how the system can model certain classes of (deterministic) algorithms under different sampling strategies. Then, we establish the link between different kinds of system stochasticity to desirable features of distributed algorithms.

3.3.1 Deterministic System

To find the FOSP of problem (3.1), we first develop a deterministic control system, in such a way that the system enters its stationary points if and only if one set of the state variables of the system correspond to a stationary solution of (3.1). First, let us define $\mathbf{x}$ as the main state variable of the system; introduce the *global consensus feedback loop* (GCFL) and *local computation feedback loop* (LCFL), where the former incorporates the dynamics from multi-agent interactions and pushes $\mathbf{x}$ to consensus, while the latter steers the system to find the stationary solution. See Figure 3.1 as an illustration of the system. In what follows, we introduce the different subsystems involved; note that $\eta_g(t)$ and $\eta_l(t)$ are the controller gains for the global and local controllers.

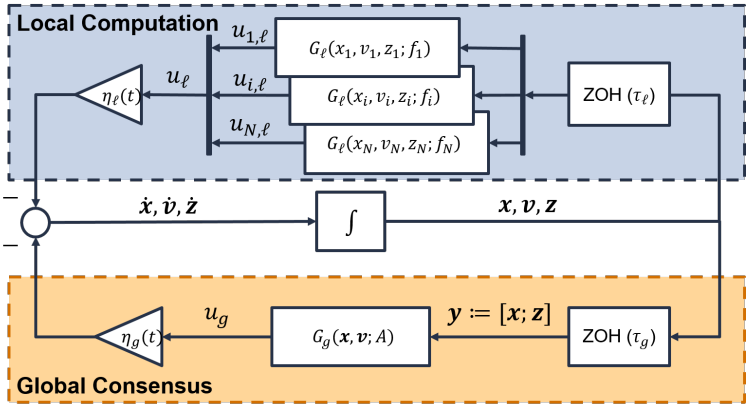


Figure 3.1: The multi-agent multi-rate double-loop feedback control system for solving (3.1).

• **(GCFL).** Define a set of auxiliary state variables $\mathbf{v} := [v_1; \ldots; v_N] \in \mathbb{R}^{Nd_v}$, with $v_i \in \mathbb{R}^{d_v}$, $\forall\, i$; further define $\mathbf{y} := [\mathbf{x}; \mathbf{v}] \in \mathbb{R}^{N(d_x+d_v)}$; the time-invariant feedback controller $G_g(\cdot; A) : \mathbb{R}^{N(d_x+d_v)} \to \mathbb{R}^{N(d_x+d_v)}$ operates on $\mathbf{y}$ to ensure the agents remain coordinated, and the states $\mathbf{y}$ remain close to consensus. Finally, we denote the output at time t as $u_g(t) := G_g(\mathbf{y}(t); A)$, which can be split as $u_g(t) = [u_{g,x}(t); u_{g,v}(t)]$;

• **(LCFL).** Define another set of auxiliary state variables $\mathbf{z} := [z_1; \ldots; z_N] \in \mathbb{R}^{Nd_z}$, with $z_i \in \mathbb{R}^{d_z}$, $\forall\, i$; define a set of time-invariant feedback controllers $G_\ell(\cdot; f_i) : \mathbb{R}^{d_x+d_v+d_z} \to \mathbb{R}^{d_x+d_v+d_z}$, one for each agent i. Further define $\mathbf{s} := [\mathbf{x}; \mathbf{v}; \mathbf{z}] \in \mathbb{R}^{N(d_x+d_v+d_z)}$. Then each agent will use LCFL to operate on its local state variables $s_i := [x_i; v_i; z_i]$, to ensure that its local system converges to a stationary solution. Finally, we denote the output at time t as $u_{i,\ell}(t) := G_\ell(s_i(t); f_i)$, which can further be split as $u_{i,\ell}(t) = [u_{i,\ell,x}(t), u_{i,\ell,v}(t), u_{i,\ell,z}(t)]$.

Throughout the chapter, we use $u_{i,\ell}(t), u_g(t)$ and $G_\ell(s_i(t); f_i), G_g(\mathbf{y}(t); A)$ interchangeably.

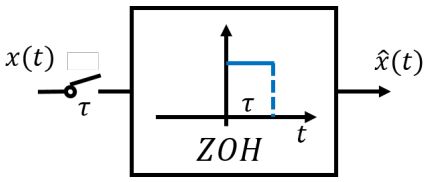


Figure 3.2: The zeroth-order hold (ZOH) for discretizing a continuous-time system.

System Discretization: The double-loop continuous-time system can be discretized by using a switch that samples the input with sample time τ, followed by a zeroth-order hold (ZOH) that keeps the signal constant between the consecutive sampling instances [73]; see Figure 3.2. More specifically, we place two ZOH units before the signal enters the two loops. This architecture offers the flexibility of choosing different sampling time for different loops resulting in three kinds of discretized systems:

• **Case I.** $\tau_g = \tau_\ell > 0$, the GCFL and LCFL are discretized with the same rate. In this case, the algorithm performs one local update followed by one step of global communication. Such an update pattern belongs to the scheme of decentralized training (DT) algorithms;

• **Case II.** $\tau_g > \tau_\ell > 0$, the local computation loop is updated more frequently. Let $\tau_g = Q \cdot \tau_\ell$, i.e., each agent performs Q steps of local computation between every two communication steps. This update strategy is related to the class of (horizontal) FL algorithms [13]. Further note that in the FL setting, the communication graph takes the fully connected graph as a special case;

• **Case III.** $\tau_\ell > \tau_g > 0$, the global communication loop is updated more frequently. We assume that $\tau_\ell = K \cdot \tau_g$, i.e., the agents perform K steps of communication between two local computation steps. This system is related to algorithms that aims to achieve the optimal communication complexity [19, 20, 79].

Let us define $\tau := \min\{\tau_g, \tau_\ell\}$ as the minimum sampling time interval, and assume $t \bmod \tau = 0$ for the rest of the chapter. We summarize the above discretization cases in Table 3.1 and provide some example algorithms that fit in the three cases.

Case	$\tau_\ell, \tau_\mathbf{g}$	**Comm.**	**Comp.**	**Related Algorithm**
I	$\tau_g = \tau_\ell > 0$	Same rate		DGD [8], DGT [10]
II	$\tau_g = Q\tau_\ell > 0$	Slow	Fast	FedPD [18], Scaffold [17]
III	$\tau_\ell = K\tau_g > 0$	Fast	Slow	xFilter [20], DSAGD [79]

Table 3.1: Summary of discretization settings, and the corresponding distributed algorithms.

We use the distributed gradient tracking (DGT) algorithm [11, 10] as an example to illustrate how to place it within the structure of the proposed system. The steps of DGT are:

$$\begin{aligned} \mathbf{x}^+ &= W\mathbf{x} - \alpha\mathbf{v}, \quad \mathbf{z}^+ = \mathbf{x}, \\ \mathbf{v}^+ &= W\mathbf{v} + (\nabla f(\mathbf{x}) - \nabla f(\mathbf{z})), \end{aligned} \tag{3.3}$$

where the states are initialized as $\mathbf{v}^0 = \nabla f(\mathbf{x}^0), \mathbf{z}^0 = \mathbf{x}^0$, α is the stepsize, and W is some mixing matrix. The continuous-time system corresponding to the DGT is:

$$\begin{aligned} \dot{\mathbf{x}} &= -(I - W)\mathbf{x} - \alpha\mathbf{v}, \quad \dot{\mathbf{z}} = \mathbf{x} - \mathbf{z}, \\ \dot{\mathbf{v}} &= -(I - W)\mathbf{v} + (\nabla f(\mathbf{x}) - \nabla f(\mathbf{z})), \end{aligned} \tag{3.4}$$

with $\tau_\ell = \tau_g = 1$. Such a discretization pattern places the above transcription in Case I. We can also extract the local and consensus controllers of the system as:

$$\begin{aligned} u_g(t) &= \begin{bmatrix} (I - W) & 0 \\ 0 & (I - W) \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{v}(t) \end{bmatrix}, \\ u_{i,\ell}(t) &= \begin{bmatrix} \alpha v_i(t) \\ (\nabla f_i(x_i(t)) - \nabla f_i(z_i(t))) \\ x_i(t) - z_i(t) \end{bmatrix}, \end{aligned}$$

with $\eta_g(t) = \eta_\ell(t) = 1$. Note that using the discretization patterns in Case II and Case III, instead of Case I, leads to new variants of the DGT algorithm.

Next, let us specify a few abstract properties that the controllers need to have. These properties will later help us analyze the behavior of the entire system, and therefore, all the algorithms that it can be used to model.

PD 1 (Linear Averaging GCFL) *The controller G_g is a linear averaging operator of $\mathbf{y}$, i.e., $G_g(\mathbf{y}; A) = W_A\mathbf{y}$ for some matrix $W_A \in \mathbb{R}^{N(d_x+d_v)}$ parameterized by A, and satisfies the following properties:*

$$\begin{aligned} C_g \left\|(I - R)\cdot\mathbf{y}\right\|^2 \leq \left\|W_A\mathbf{y}\right\|^2 \leq \left\|(I - R)\cdot\mathbf{y}\right\|^2, \\ W_A = W_A^T,\ \langle \mathbb{1}_N, W_A\rangle = 0. \end{aligned} \tag{3.5}$$

PD 2 (Lipschitz Smoothness) *The local controller is Lipschitz continuous, that is:*

$$\begin{aligned} &\left\|G_\ell(s_i; f_i) - G_\ell(s_i'; f_i)\right\| \leq L\left\|s_i - s_i'\right\|, \\ &\forall\ i \in [N],\ s_i, s_i' \in \mathbb{R}^{d_x+d_v+d_z}. \end{aligned}$$

PD 3 (Size of Control Signals) *For given s_i, the sizes of the control signals are upper bounded by that of the local gradients, i.e., for some positive constants C_x, C_v, C_z and $C_f = C_x^2 + C_v^2 + C_z^2$:*

$$\begin{aligned} &\left\|u_{i,\ell,x}\right\| \leq C_x\left\|\nabla f_i(x_i)\right\|,\ \left\|u_{i,\ell,v}\right\| \leq C_v\left\|\nabla f_i(x_i)\right\|, \\ &\left\|u_{i,\ell,z}\right\| \leq C_z\left\|\nabla f_i(x_i)\right\|,\ \left\|u_{i,\ell}\right\|^2 \leq C_f\left\|\nabla f_i(x_i)\right\|^2. \end{aligned}$$

These properties are easy to verify: PD1 follows A5, PD2 and PD3 can be derived from A6. Further, assume that within the sampling intervals the stepsizes are kept as constants, i.e., $\eta_g(t_1) = \eta_g(t),\ \forall t_1 \in [t, t+\tau_g)$, and $\eta_\ell(t_1) = \eta_\ell(t),\ \forall t_i \in [t, t+\tau_\ell)$,

3.3.2 System Stochasticity

As mentioned in the introduction, in practical ML applications, it is often preferred to use stochastic algorithms rather than deterministic ones. Therefore, we consider replacing the deterministic controllers introduced previously (Fig. 3.1) with stochastic ones, denoted by $\tilde{G}_\ell(\cdot), \tilde{G}_g(\cdot)$. We start by providing generic discussions on how these stochastic controllers are modeled. Specific correspondence of these controllers to concrete applications will be presented in Section 3.5.
Additive Noise: The first form of stochastic controller has additive noise at its output. That is:

$$\tilde{u} = u + w,$$

where w is the additive noise, and in most cases we consider white noise (i.e., $\mathbb{E}[w(t)] = 0$ and $\mathrm{Cov}(w(t), w(t+h)) = 0, \forall\ h \neq 0$). Additive white noises arise in many situations, for example, in algorithms involving stochastic gradients or differential privacy.
Multiplicative Noise: The second form of stochastic controller has multiplicative noise. That is:

$$\tilde{u} = (I + \mathcal{W}) \cdot u,$$

where $\mathcal{W}$ is a random matrix. This type of stochasticity can be used to model random communication graphs, partial participation, and communication sparsification.
Mixture of Noise: The third form of stochastic controller is the combination of the previous two, involving a *mixture* of additive and multiplicative noises. This setting can be used to model complex algorithms, e.g., FL algorithms that involve both differentially private noise and agent sampling; cf. [76].

From the above-mentioned scenarios, we can abstract the following assumptions on the stochastic controllers:

PS 1 (Expected Control Signal) *The stochastic GCFL is an unbiased estimator of its deterministic counterpart:*

$$\mathbb{E}[\tilde{G}_g(\mathbf{x}, \mathbf{v}; A)] = G_g(\mathbf{x}, \mathbf{v}; A), \forall \mathbf{x} \in \mathbb{R}^{Nd_x}, \mathbf{v} \in^{Nd_v},$$

*and (**A**) the stochastic LCFL is also unbiased, satisfying:*

$$\mathbb{E}_t[\tilde{G}_\ell(s_i; f_i)] = G_\ell(s_i; f_i),\ \forall\ i \in [N], s_i \in \mathbb{R}^{d_x + d_v + d_z},$$

*or (***B***) the stochastic LCFL is biased: there exist positive constants* C_1, C_2, σ_G *satisfying the following:*

$$\mathbb{E}\left[\left\langle \tilde{G}_\ell(s_i; f_i), G_\ell(s_i; f_i)\right\rangle\right] \geq C_2 \left\|G_\ell(s_i; f_i)\right\|^2 - \sigma_G^2,$$
$$\left\|\mathbb{E}[\tilde{G}_\ell(s_i; f_i)]\right\|^2 \leq C_1, \ \forall\ i \in [N], s_i \in \mathbb{R}^{d_x+d_v+d_z}.$$

Note that the controller $G_g(\mathbf{y}(t); A)$ is linear in $\mathbf{y}(t)$, thus we can guarantee it is unbiased. However, the LCFL may be nonlinear or non-convex; consequently, PS1(A) can be difficult to satisfy. Therefore, we make a relaxed assumption PS1(B), which allows certain degrees of bias and misalignment between the deterministic controller and its stochastic counterpart. It is easy to see that PS1(A) is a special case of (B) with $C_1 = \infty, C_2 = 1, \sigma_G = 0$.

PS 2 (Bounded Variance) *There exist positive constants* $B_g, B_\ell, \sigma_g, \sigma_\ell$*, such that the following hold:*

$$\mathbb{E}\left[\left\|\tilde{G}_\ell(s_i; f_i) - \mathbb{E}[\tilde{G}_\ell(s_i; f_i)]\right\|^2\right]$$
$$\leq B_\ell \left\|\mathbb{E}[\tilde{G}_\ell(s_i; f_i)]\right\|^2 + \sigma_\ell^2, \ \forall\ i \in [N], s_i \in \mathbb{R}^{d_x+d_v+d_z},$$
$$\mathbb{E}\left[\left\|\tilde{G}_g(\mathbf{x}, \mathbf{v}; A) - G_g(\mathbf{x}, \mathbf{v}; A)\right\|^2\right]$$
$$\leq B_g \left\|G_g(\mathbf{x}, \mathbf{v}; A)\right\|^2 + \sigma_g^2, \ \forall \mathbf{x} \in \mathbb{R}^{Nd_x}, \mathbf{v} \in R^{Nd_v}.$$

Note that if the stochasticity in the controller is an additive white noise, then it is easy to see that $B_\ell = 0, B_g = 0$ and PS1(A) is satisfied.

PS 3 (Independence) *The stochastic noise terms in the controllers are independent, satisfying the following:*

$$\mathrm{Cov}_t\left(\tilde{G}_g(\mathbf{x}(t), \mathbf{v}(t); A), \tilde{G}_\ell(s_i(t); f_i)\right) = 0.$$

Note that we only assume independence between the consensus and local control signals at time t, while the control signals at different times can be correlated.

3.4 Convergence Analysis

In this section, we analyze the theoretical behavior of the stochastic system described in Section 3.3.2. First, we introduce an energy-like function for the system:

$$\mathcal{E}(t) := f(\bar{\mathbf{x}}(t)) - f^\star + \left\|(I - R) \cdot \mathbf{y}(t)\right\|^2. \tag{3.6}$$

Note that $\mathcal{E}(t) \geq 0$ for all $\mathbf{s}(t) = [\mathbf{x}(t); \mathbf{v}(t); \mathbf{z}(t)]$.

Let us begin by assuming that the deterministic system satisfies the following property.

PD 4 (Descent of Deterministic System) *The difference of the energy function of the deterministic system satisfies:*

$$\begin{aligned}\mathcal{E}(t) - \mathcal{E}(0) \leq & -\sum_{r=0}^{t/\tau-1} \gamma_1(r\tau) \cdot \|\nabla f(\bar{\mathbf{x}}(r\tau)\|^2 \\ & -\sum_{r=0}^{t/\tau-1} \gamma_2(r\tau) \cdot \|(I-R)\cdot \mathbf{y}(r\tau)\|^2,\end{aligned} \tag{3.7}$$

where $\gamma_1(r\tau), \gamma_2(r\tau) > 0$ are coefficients depending on the choice of $\eta_\ell, \eta_g, \tau_\ell, \tau_g$.

This property immediately implies that the algorithm converges to the FOSP of the problem, in the sense that the following holds: the convergence error and consensus error are both decreasing to zero as the LHS is lower bounded by $-\mathcal{E}(0)$. Property PD4 appears to be strong compared with Properties PD1 – PD3, since it is about the entire sequence generated by the control system. We require that the deterministic system satisfies this property because: 1) This is in fact a standard property that a wide range of deterministic algorithms can satisfy; 2) Having this property can help us focus on investigating the effect of various kinds of stochasticity on the system performance. To see point 1) above, we note that this property has been explicitly shown in algorithms such as DGD [80][Theorem 2], DGT [11][Theorem 3], xFilter [20][Theorem 5.1], and FedDP [18][Theorem 1 Case I]. Of course, when designing a *new* (stochastic) algorithm, this property has to be verified for its deterministic counterpart, before we move to analyze the entire stochastic system.

Next, we move on to characterize the impact of the stochasticity in the controllers satisfying PS1 - PS3. The key challenge is to characterize the deviations of $\mathcal{E}(t)$ caused by the system stochasticity in different discretization cases.

Case I: For Case I, $\tau_g = \tau_\ell > 0$. Let us denote the states at the r^{th} sampling time instance as $(\cdot)^r := (\cdot)(r\tau_\ell)$, then the discretized system can be written as:

$$\begin{aligned}\tilde{\mathbf{x}}^{r+1} &= \tilde{\mathbf{x}}^r - \eta_\ell'^r \cdot \tilde{u}_{\ell,x}^r - \eta_g'^r \cdot \tilde{u}_{g,x}^r \\ \tilde{\mathbf{v}}^{r+1} &= \tilde{\mathbf{v}}^r - \eta_\ell'^r \cdot \tilde{u}_{\ell,v}^r - \eta_g'^r \cdot \tilde{u}_{g,v}^r \\ \tilde{\mathbf{z}}^{r+1} &= \tilde{\mathbf{z}}^r - \eta_\ell'^r \cdot \tilde{u}_{\ell,z}^r,\end{aligned} \tag{3.8}$$

where $\eta_\ell'^r = \tau_\ell \cdot \eta_\ell(r\tau_\ell)$, $\eta_g'^r = \tau_\ell \cdot \eta_g(r\tau_\ell)$.

Then, we have the following results:

Lemma 4 *Suppose the deterministic system satisfies PD1 - PD4, and the stochastic controllers satisfy PS2 and PS3. Consider the discretization Case I with $\tau_g = \tau_\ell > 0$.*
(A) *If PS1(A) is satisfied, then we have the following:*

$$\begin{aligned}\mathbb{E}[\tilde{\mathcal{E}}^t] - \mathcal{E}^0 \leq & -\sum_{r=0}^{t-1} \underbrace{(\gamma_1^r - C_{11}^r)}_{=:\gamma_1'(r)} \cdot \mathbb{E}[\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2] \\ & - \sum_{r=0}^{t-1} \underbrace{(\gamma_2^r - C_{12}^r)}_{=:\gamma_2'(r)} \cdot \mathbb{E}[\|(I-R)\cdot\tilde{\mathbf{y}}^r\|^2] \\ & + C_{13}(t)\sigma_g^2 + C_{14}(t)\sigma_\ell^2,\end{aligned} \tag{3.9}$$

where $C_{11}^r := B_\ell \cdot (C_x^2 + C_v^2) \cdot (1 + \frac{L_f}{2N}) \cdot (\eta_\ell'^r)^2$, $C_{12}^r := C_{11}^r L_f^2 + B_g \cdot (\eta_g'^r)^2 \cdot (1 + \frac{L_f}{2N})$, $C_{13}(t) := \sum_{r=0}^{t-1} (\eta_g'^r)^2 \cdot (1 + \frac{L_f}{2N}), C_{14}(t) := \sum_{r=0}^{t-1} (\eta_\ell'^r)^2 \cdot (1 + \frac{L_f}{2N})$.

(B) *If PS1(B) is satisfied, then we have the following:*

$$\begin{aligned}\mathbb{E}[\tilde{\mathcal{E}}^t] - \mathcal{E}^0 \leq & -\sum_{r=0}^{t-1} \underbrace{(\gamma_1^r - C_{11}'^r)}_{:=\gamma_1'(r)} \cdot \mathbb{E}[\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2] \\ & - \sum_{r=0}^{t-1} \underbrace{(\gamma_2^r - C_{12}'^r)}_{:=\gamma_2'(r)} \cdot \mathbb{E}[\|(I-R)\cdot\tilde{\mathbf{y}}^r\|^2] \\ & + C_{13}(t)\sigma_g^2 + C_{14}(t)\sigma_\ell^2 + C_{15}(t)C_1 + C_{16}(t)\sigma_G^2.\end{aligned}$$

where $C_{11}'^r, C_{12}'^r, C_{15}(t), C_{16}(t)$ are positive coefficients depending on $L, L_f, C_2, C_x, C_v, B_\ell, B_g, \eta_\ell'^r, \eta_g'^r,$.

The proofs and choices of the parameters for Lemma 4(A) and (B) are provided in Appendix B.3.2 and Appendix B.3.3 due to space limits. This lemma indicates that by using stochastic controllers, the system introduces extra perturbations. Compared with (A), the result in (B) has two extra error terms which are caused by the biased stochastic local controllers. The key point is to choose $\eta_\ell'^r, \eta_g'^r$ such that $\gamma_1'(r) > 0, \gamma_2'(r) > 0$ and minimize $\{C_{1i}(t)\}_{i=3}^6$, so that the error terms accumulate slower than the rate at which the first two terms decrease. This choice depends on the specification of the deterministic algorithm. Further, we have:

Theorem 3 *Suppose the deterministic system in Case I satisfies PD1 - PD4, with stochastic controllers satisfying PS1, PS2 and PS3. The algorithm converges with:*

$$\mathbb{E}\left[\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^{r_1})\right\|^2 + \|(I-R)\cdot\tilde{\mathbf{y}}^{r_1}\|^2\right] \leq \frac{\mathcal{E}^0 + C_3(t)}{\sum_{r=0}^{t-1}\gamma'(r)},$$

where $\gamma'(r) := \min\{\gamma_1'(r), \gamma_2'(r)\}$, $C_3(t) = C_{13}(t)\sigma_g^2 + C_{14}(t)\sigma_\ell^2$ for PS1(A) and $C_3(t) = C_{13}(t)\sigma_g^2 + C_{14}(t)\sigma_\ell^2 + C_{15}(t)C_1 + C_{16}(t)\sigma_G^2$ for PS1(B).

For Case II and Case III, similar results can be derived. Detailed derivations are provided in Appendix B.3.4.

In summary, starting with a convergent deterministic system, we can replace the controllers with their stochastic versions that satisfy properties PS1-PS3. The resulting stochastic systems not only slow down by a certain factor depending on C_{i1}, C_{i2}, but also suffers form additional error terms in C_3. Let us comment on these terms:

1) Suppose that PS1(A) is satisfied and $\sigma_g = \sigma_\ell = 0$ in PS2, i.e., the variance of the controller can be fully bounded by the size of the deterministic control signal, then $C_3 = 0$. Therefore, it only requires $C_{i1} < c\gamma_1, C_{i2} < c\gamma_2$ with constant $0 \leq c < 1$ for the stochastic algorithm to converge. In this case, the convergence rate of the stochastic algorithm will have the same order as the baseline deterministic algorithm.

2) If $\sigma_g, \sigma_\ell > 0$, i.e., the variance of the controller stays constant, then we need to balance between the error term and the descent terms. In this case, the convergence rate of the stochastic algorithm may slow down in order, or lose it convergence. In Section 3.5.3, we use the DGT algorithm to demonstrate how the parameters are specified to balance the error and the convergence rate.

3.5 Application of the Framework

In this section, we demonstrate the modeling capability of the proposed control system. We first show that a few important algorithmic features can be mapped to specific types of stochastic controllers. We then combine these controllers in different ways to construct a number of popular distributed algorithms. Finally, we use the DGT algorithm as an example to illustrate how the proposed framework facilitates new algorithm design.

3.5.1 Mapping Features to the Stochastic Controllers

We first discuss how a number of features that are desirable to distributed algorithms can be mapped to specific stochastic controllers, which satisfy PS1-PS3.

First, we discuss a few realizations of $\tilde{G}_g(\mathbf{y}(t); A)$:

• Randomized Communication Graph (RG): Suppose the communication graph $\mathcal{G}(t)$ is randomly time-varying. This can be caused by limited bandwidth or unreliable connection, so that at time t, the agents randomly choose a subset of their neighbors to broadcast local information, and gather the information from a possibly different random subset of neighbours [77, 27]. In this case, $\tilde{G}_g(\mathbf{y}(t); A) := \tilde{W}_A(t)\mathbf{y}(t)$, where $\tilde{W}_A(t)$ is a random matrix satisfying $\mathbb{E}[\tilde{W}_A(t)] = W_A$ and if $(i, j) \notin \mathbf{E}$, $\tilde{W}_{A,ij}(t) = 0$. An extreme is that $\tilde{W}_A$ is diagonal and no communication happens. This case satisfies PS1 and PS2.

• Partial Agent Participation (PP): Partial agent participation often arises in FL, where at each communication round, only a subset of P agents send their updates to the server [13, 81]. PP is a more practical approach than full agent aggregation and can be viewed as a special case of randomized communication graph $\tilde{G}_g(\mathbf{y}(t); A) := \tilde{W}_A(t)\mathbf{y}(t)$, where the averaging matrix takes the following form:

$$\tilde{W}_A(t) = \frac{\mathbb{1}_N \mathbf{B}^T(t)}{\mathbb{1}_N^T \mathbf{B}(t)},\ \mathbf{B}(t) \in \{0,1\}^N,\ \mathbb{E}[\mathbf{B}(t)] = \frac{P}{N}\mathbb{1}_N,$$

where $\mathbf{B}(t)$ is a length-N random vector. In this case, it satisfies PS1 that $\mathbb{E}[\tilde{W}_A(t)] = R$ and PS2 with $\sigma_g = 0$.

• Compressed Communication (CC): A different way of resolving the communication bandwidth issue is to reduce the data transmitted as each communication round by using compression methods such as (randomized) quantization and sparsification [82, 83]. The controller can be written as:

$$\tilde{G}_g(\mathbf{y}; A) := G_g(\mathcal{W}\mathbf{y}; A),\ \mathbb{E}[\mathcal{W}] = I,$$

where $\mathcal{W}$ is a diagonal multiplicative noise matrix for compression and satisfies PS1, PS2. For example, we can set $\mathcal{W}$ as the sparsification matrix with:

$$\mathcal{W}_{jj} = \begin{cases} \frac{1}{p}, & \text{w.p. } p, \\ 0, & \text{w.p. } 1-p, \end{cases}$$

where $p < 1$ denotes the compression rate [21]; or set $\mathcal{W}$ as the quantization matrix with:

$$\mathcal{W}_{jj} = \begin{cases} \frac{\lceil \mathbf{y}_j \rceil}{\mathbf{y}_j}, & \text{w.p. } \frac{\mathbf{y}_j - \lfloor \mathbf{y}_j \rfloor}{\lceil \mathbf{y}_j \rceil - \lfloor \mathbf{y}_j \rfloor}, \\ \frac{\lfloor \mathbf{y}_j \rfloor}{\mathbf{y}_j}, & \text{w.p. } \frac{\lceil \mathbf{y}_j \rceil - \mathbf{y}_j}{\lceil \mathbf{y}_j \rceil - \lfloor \mathbf{y}_j \rfloor}, \end{cases}$$

where $\lceil \cdot \rceil, \lfloor \cdot \rfloor$ denote the upper and lower quantization levels [22] which satisfies PS1 and PS2. These methods can efficiently save the communication on structured data.

• Differential Privacy Noise: One important motivation to implement distributed systems is to guarantee user data privacy. DP is a widely used notion for measuring privacy, because it provides strong guarantees, while being easily implementable [55]. The most popular mechanism to ensure DP is called the *Gaussian mechanism*, which adds noise to the algorithm outputs [55]. In a distributed setting, this mechanism can be viewed as adding noise to the local messages before they get transmitted. To model the DP noise, the stochastic controller can be written as

$$\tilde{G}_g(\mathbf{y}; A) := W_A \cdot (\mathbf{y} + \mathbf{w}_g),$$

where $\mathbf{w}_g \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma^2 = \Omega(\frac{pt \log(\delta^{-1})}{N\epsilon^2})$ capturing the privacy noise [76].

Next, we discuss a few realizations of $\tilde{G}_\ell(s_i(t); f_i)$:

• Clipping: Note that when implementing differentially private algorithms, local clipping operation is usually needed to bound the algorithm sensitivity, which can be written as:

$$\text{clip}(G_{\ell,i}(s_i; f_i); c) := G_{\ell,i}(s_i; f_i) \cdot \max\left\{1, \frac{c}{\|G_{\ell,i}(s_i; f_i)\|}\right\},$$

where c denotes the clipping threshold. In this case, even if $G_{\ell,i}(s_i; f_i)$ is unbiased, the non-linear clipping operation will introduce extra biased noise [53] satisfying PS1(B) with $C_1 = c$, and C_2, σ_G depending on data distribution.

• Stochastic Gradient (SG): As mentioned before, state-of-the-art ML applications often use SGD based local updates. This can be easily translated to a stochastic local controller where the stochastic gradient is estimated on sampled data:

$$\tilde{u}_{i,\ell}(t) = \nabla f_i(x_i(t)) + \underbrace{\nabla f_i(x_i(t); \xi_i(t)) - \nabla f_i(x_i(t))}_{w_i(t)},$$

where $w_i(t)$ is the additive noise; $\xi_i(t)$ is drawn uniformly from the local dataset. So $\mathbb{E}[\tilde{\nabla} f_i(x_i(t))] = \nabla f_i(x_i(t))$ which satisfies PS1, and it is common to assume that $\text{Var}(w_i(t))$ satisfies PS2 [3, 25, 17].

• Zeroth-order Optimization (ZO): the zeroth-order optimization method have been developed in recent years in the setting that only the loss values $f_i(x_i)$ can be accessed [28, 84, 29]. One can use zeroth-order method to approximate the gradient:

$$\tilde{\nabla} f_i(x_i) := \frac{f_i(x_i + \delta h) - f_i(x_i - \delta h)}{2h}\delta,$$

where δ uniformly samples from the unit sphere and h is a sufficiently small scalar. Similar to the previous case, we have:

$$\tilde{u}_{i,\ell}(t) = \nabla f_i(x_i(t)) + \underbrace{\tilde{\nabla} f_i(x_i(t)) - \nabla f_i(x_i(t))}_{w_i(t)},$$

where $w_i(t)$ is a *biased* additive noise [28].

Note that different forms of noises can be combined together for more complex applications, e.g., in DP, we may combine DP with Clipping and SG for better performance.

3.5.2 Algorithm Classification

In this subsection, we discuss some popular distributed algorithms and how they fall into the proposed framework.

• We first start with DT algorithms, which belongs to Case I: DSGD [3] uses stochastic gradient

as LCFL with deterministic GCFL. Its variations include [77] which studies random communication graph, [85, 22] with communication compression, and D-(DP)2SGD [27] with differential privacy. GNSD [25] uses gradient tracking on stochastic gradient, and ZONE [29] uses zeroth-order optimization for gradient estimation.

• FL is another popular class of distributed algorithms, which uses discretization Case II. Popular algorithms include FedPD [18] that implements the ADMM algorithm with stochastic gradient as the local solver, and uses a random aggregation scheme to save communication while FedDyn [81] considers partial client participation. Scaffold [17] tracks local stochastic gradients to correct the update direction; DP-FedAvg [76, 58] apply differential privacy to FedAvg; Qsparse-Local-SGD uses communication sparsification on FedAvg [21].

• Finally, we give an example algorithm trying to optimize the convergence rate dependencies via multi-step communication in Case III: DSAGD [79] uses stochastic gradient and multi-step averaging on random communication graphs to accelerate consensus.

We summarize the above discussions in Table 3.2, where we specify the discretization cases and the stochasticities in each algorithm. More detailed algorithmic correspondence are included in Appendix B.2

3.5.3 Algorithm Design: A Case Study

In this subsection, we take the decentralized gradient tracking (DGT) algorithm as an example to illustrate how the framework can be applied to design new algorithms for different applications. In specific, we modify the DGT algorithm to include features such as SG, RG and DP, and name the resulting algorithms as Distributed Stochastic Gradient Tracking (DSGT) (which is the same as GNSD [25]), Distributed Dynamic-graph Gradient Tracking (D^2GT) and Differentially Private DSGT (DP-DSGT). By verifying PD1-PD4 and PS1-PS3 for each case, we have the following informal theoretical result:

Corollary 2 (Informal) *With properly chosen stepsize, the expected stationarity gaps of DSGT, D^2GT, and DP-DSGT converge with rates $\mathcal{O}(\frac{\log(t)}{\sqrt{t}})$, $\mathcal{O}(\frac{1}{t})$, and $\mathcal{O}(\frac{\sqrt{d_x+d_v}\log(\delta^{-1})}{N\epsilon})$ respectively, where the expectation is taken over the iterations, and DP-DSGT satisfies the (ϵ, δ)-differential privacy.*

We can see that with multiplicative noise, D^2GT has the fastest convergence rate, which is essentially the same order as DGT; DSGT converges slower due to the additive noise in SG, and recovers the rate obtained in [25]; DP-DSGT has a constant error independent of t due to the additive noises caused by DP.

Algorithm	**Discretization**	**Stochasticity**
DSGD	Case I	SG, CC, RG
GNSD	Case I	SG
D-(DP)2SGD	Case I	SG, DP, RG
ZONE	Case I	ZO
FedPD/FedDyn	Case II	SG, RG/PP
Scaffold	Case II	SG, PP
Qsparse-Local-SGD	Case II	SG, CC
DP-FedAvg	Case II	SG, DP, PP
DSAGD	Case III	SG, RG

Table 3.2: Summary of the distributed stochastic algorithms, with discretization cases and stochasticity in the controller.

Numerical results for the algorithms on the non-convex regularized logistic regression problem [86] are shown in Figure 3.3. In the experiment, we choose the stepsizes based on the theoretical result, i.e., η_g', η_ℓ' as constants for DGT, D^{2}GT; and $\eta_\ell'^r = \mathcal{O}(1/\sqrt{r})$ for GNSD and DP-GNSD.

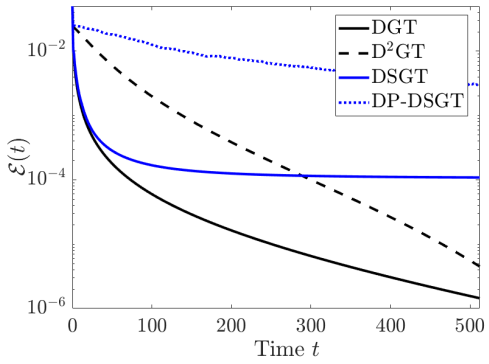


Figure 3.3: The convergence of the stationarity gap of DGT, D^{2}GT, GSGT and DP-DSGT.

It can be observed that D^{2}GT has the same convergence rate as DGT with a constant slow down, while GNSD and DP-GNSD have slower convergence rates. Due to page limitation, we refer to Appendix B.4 for detailed discussions on the algorithm modifications, theoretical analyses and experiment settings and additional results.

Chapter 4

Gradient Tracking for Decentralized Optimization

4.1 Motivation

Recent advances in deep learning have dramatically improved the performance of many classical machine learning tasks, such as image processing and natural language processing [87]. However, as the sizes of model parameters and training datasets keep increasing, the model training time also increases dramatically. For example, training the visual geometry group (VGG) neural network on a single machine usually takes 2 to 3 weeks [88]. This motivates us to solve challenging problems using massive computational resources, either using a centralized parameter server setting [1][2] or a fully decentralized system [3]. However, the centralized distributed system has its own bottleneck due to its fragile structure, bandwidth limit, latency requirement and large communication overhead. Therefore, an efficient decentralized algorithm is highly needed such that the learning tasks can be performed efficiently by using multiple computational nodes.

In this work, we consider a network of n agents defined by a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each agent indexed by i only has access to its own local function $f_i(\mathbf{x}_i)$ and can only communicate with its immediate neighbors. The goal of the optimization problem is to minimize the total loss value of the network, which can be formulated as the following non-convex finite sum problem with the consensus constraint:

$$\min_{\{\mathbf{x}_i \in \mathbb{R}^d\}} \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}_i), \quad \text{s. t.} \quad \mathbf{x}_i = \mathbf{x}_j, j \in \mathcal{N}_i, \forall i \tag{4.1}$$

where $\mathcal{N}_i$ denotes the set of the ith node's neighbors.

When the size of the dataset is large, the calculation of the full gradient requires accessing the entire dataset, which is computationally expensive. One of the most efficient ways in practice is to use the stochastic gradient to approximate the true gradient at each iteration. To be specific, the algorithm samples a subset of data ξ_i randomly at each iteration and calculates the stochastic gradient $g_i(\mathbf{x}_i, \xi_i)$, where the data follows some distribution $\mathcal{D}_i$. Obviously, if those samples are collected randomly and independently, we have the following unbiasedness property of estimating the gradient: $\mathbb{E}_{\xi_i \sim \mathcal{D}_i}[g_i(\mathbf{x}_i, \xi_i)] = \nabla f_i(\mathbf{x}_i), \forall i$.

4.1.1 Related work

Motivated by applications such as distributed machine learning [2] or statistical learning [89], distributed optimization has attracted significant attention nowadays. Extensive work has been conducted by focusing on convex optimization problems with applications in signal processing and communications, for example, the primal variable based methods such as distributed subgradient (DSG) method [90, 91], the EXTRA method [92], distributed adaptive filters [93], and primal-dual based methods such as [94, 89, 95]. However, in practice, most of the problems, e.g., training neural networks, are highly non-convex, which are more difficult to be solved compared with the convex cases, especially in decentralized settings. For example, the classic DSG algorithm cannot find the first-order stationary points (FOSP) by using a constant step-size. Recently, there are several works on developing non-convex decentralized methods. For instance, primal-dual based methods proposed in [96, 97, 20], gradient-tracking based methods shown in [11, 98], non-convex extensions of decentralized gradient descent (DGD) methods described in [80]. These methods are all deterministic, which may not be suitable when the dataset is large.

Stochastic distributed non-convex methods can be traced back to [41] [42], where distributed stochastic algorithms consisting of one local stochastic gradient descent (SGD) step and one gossip step has been proposed. Related works on stochastic distributed non-convex methods include the consensus-based distributed SGD [40] and decentralized PSGD [3]. Although these algorithms can obtain some reasonably high-quality solutions, the convergence analysis as well as the numerical experiments show that they either do not have global convergence rates (see, e.g., [41, 42]), or they can only converge to a neighborhood of FOSP [40, 3]. More recently, a variant of PSGD [99], named D^2, has been proposed, and it has been shown that this algorithm can converge to FOSPs with a quantifiable rate of $\mathcal{O}(1/\sqrt{T})$. However, D^2 requires that the mixing matrix satisfy certain restrictive assumptions (which will be discussed in details shortly).

4.2 Preliminaries

4.2.1 Assumptions

We consider the following standard assumptions:

A1: The objective function has Lipschitz gradient with constant L:

$$\|\nabla_{\mathbf{x}_i} f_i(\mathbf{x}_i) - \nabla_{\mathbf{x}_i'} f_i(\mathbf{x}_i')\| \leq L\|\mathbf{x}_i - \mathbf{x}_i'\|, \forall i \tag{4.2}$$

A2: The samples are collected independently at each iteration and the stochastic gradient is unbiased, i.e.,

$$\mathbb{E}_{\xi_i}[\nabla_{\mathbf{x}_i} g_i(\mathbf{x}_i, \xi_i)] = \nabla f_i(\mathbf{x}_i), \forall i \tag{4.3}$$

A3: The estimates of gradient have bounded variance, i.e.,

$$\mathbb{E}_{\xi_i}\|\nabla_{\mathbf{x}_i} g_i(\mathbf{x}_i, \xi_i) - \nabla f_i(\mathbf{x}_i)\|^2 \leq \sigma^2, \forall i \tag{4.4}$$

A4: The mixing matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ is symmetric (to be defined more formally soon), satisfying the following

$$|\underline{\lambda}_{\max}(\mathbf{W})| \triangleq \eta < 1, \quad \mathbf{W}\mathbb{1} = \mathbb{1}. \tag{4.5}$$

where $\underline{\lambda}_{\max}(\mathbf{W})$ denotes the second largest eigenvalue of $\mathbf{W}$, and (4.5) also implies $\|\mathbf{W} - \frac{1}{n}\mathbb{1}\mathbb{1}^T\| < 1$, where $\mathbb{1} \in \mathbb{R}^{n\times 1}$ is an all one vector.

Remark 1. Note the requirement of the spectral norm of $\mathbf{W}$ is relaxed to $|\underline{\lambda}_{\max}(\mathbf{W})| < 1$ compared with the existing results $-1/3 < \underline{\lambda}_{\max}(\mathbf{W}) < 1$ shown in [99].

Remark 2. Note that many choices of $\mathbf{W}$ satisfy the above condition. For example, $\mathbf{W} = \mathbf{I} - \alpha\mathcal{L}$, where $\mathcal{L}$ denotes the normalized graph Laplacian matrix, and $\alpha \in (0, 1)$ is some sufficiently small weighting parameter.

4.3 Algorithm Design

Let r denote the index of the iteration. The GNSD algorithm is shown in Algorithm 1, where the iterates of GNSD $\mathbf{x}_i^r, \mathbf{y}_i^r, \forall i$ are updated locally as follows, for all $r \geq 1$

$$\mathbf{x}_i^{r+1} = \sum_{j\in\mathcal{N}_i} \mathbf{W}_{ij}\mathbf{x}_j^r - \alpha\mathbf{y}_i^r, \tag{4.6a}$$

$$\mathbf{y}_i^{r+1} = \sum_{j\in\mathcal{N}_i} \mathbf{W}_{ij}\mathbf{y}_j^r + \nabla_{\mathbf{x}_i} g_i(\mathbf{x}_i^{r+1}, \xi_i^{r+1}) - \nabla_{\mathbf{x}_i} g_i(\mathbf{x}_i^r, \xi_i^r), \tag{4.6b}$$

and $\mathbf{x}_i^1 \triangleq \sum_{j\in\mathcal{N}_i} \mathbf{W}_{ij}\mathbf{x}_j^0, \forall i$, $\mathbf{y}_i^1 \triangleq \nabla_{\mathbf{x}_i} g_i(\mathbf{x}_i^1, \xi_i^1), \forall i$. We further define some global optimization variables by a concatenation of local variables as follows, $\mathbf{x} \triangleq [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T$, $\mathbf{y} \triangleq [\mathbf{y}_1, \ldots, \mathbf{y}_n]^T$,

$F(\mathbf{x}) \triangleq [f_1(\mathbf{x}_1), \ldots, f_n(\mathbf{x}_n)]^T$, $G(\mathbf{x}, \xi) \triangleq [g_1(\mathbf{x}_1, \xi_1), \ldots, g_n(\mathbf{x}_n, \xi_n)]^T$, $\xi \triangleq [\xi_1, \ldots, \xi_n]^T$. Then we have the following updates

$$\mathbf{x}^{r+1} = \mathbf{W}\mathbf{x}^r - \alpha \mathbf{y}^r, \tag{4.7a}$$

$$\mathbf{y}^{r+1} = \mathbf{W}\mathbf{y}^r + \nabla_{\mathbf{x}} G(\mathbf{x}^{r+1}, \xi^{r+1}) - \nabla_{\mathbf{x}} G(\mathbf{x}^r, \xi^r). \tag{4.7b}$$

Algorithm 1 Gradient-Tracking based Nonconvex Stochastic Decentralized (GNSD) Algorithm

Input: $\mathbf{x}^{(0)}$

for $r = 0, 1, \ldots$ **do**

Random sample ξ_i^r at each node

Calculate the stochastic gradient $\nabla g_i(\mathbf{x}_i^r, \xi_i^r)$ at each node

Update $\mathbf{x}_i^{r+1}$ by (4.6a)

Update $\mathbf{y}_i^{r+1}$ by (4.6b)

end for

Remark 3. It can be easily shown that when $\mathbf{W} = \mathbf{I}$, GNSD is the D^2 algorithm [99], see below.

$$\begin{aligned}
\mathbf{x}^{r+2} &= \mathbf{W}\mathbf{x}^{r+1} - \alpha \mathbf{W}\mathbf{y}^r - \alpha \left(\nabla G(\mathbf{x}^{r+1}, \xi^{r+1}) - \nabla G(\mathbf{x}^r, \xi^r) \right) \\
&= 2\mathbf{W}\mathbf{x}^{r+1} - \mathbf{W}^2\mathbf{x}^r - \alpha \left(\nabla G(\mathbf{x}^{r+1}, \xi^{r+1}) - \nabla G(\mathbf{x}^r, \xi^r) \right) \\
&= 2\mathbf{x}^{r+1} - \mathbf{x}^r - \alpha \left(\nabla G(\mathbf{x}^{r+1}, \xi^{r+1}) - \nabla G(\mathbf{x}^r, \xi^r) \right).
\end{aligned}$$

4.4 Convergence Analysis

To facilitate our analysis, we first define a "virtual sequence" $\{\underline{\mathbf{y}}^r\}$, which characterizes the updated by using the true gradients. That is:

$$\underline{\mathbf{y}}^{r+1} \triangleq \mathbf{W}\underline{\mathbf{y}}^r + \nabla_{\mathbf{x}} F(\mathbf{x}^{r+1}) - \nabla_{\mathbf{x}} F(\mathbf{x}^r), \forall r \geq 1 \tag{4.8}$$

as the counterpart of (4.7b), to help us quantify the difference between the estimated gradient and the true one, where $\underline{\mathbf{y}}^1 \triangleq \nabla F(\mathbf{x}^1)$.

Also, let $\bar{\mathbf{x}}^r \triangleq \frac{1}{n}\mathbb{1}^T\mathbf{x}^r$, $\bar{\mathbf{y}}^r \triangleq \frac{1}{n}\mathbb{1}^T\mathbf{y}^r$, $\underline{\bar{\mathbf{y}}}^r \triangleq \frac{1}{n}\mathbb{1}^T\underline{\mathbf{y}}^r$. Substituting (4.8) into the definition of $\underline{\bar{\mathbf{y}}}^r$ and applying the telescope sum, we can get $\underline{\bar{\mathbf{y}}}^r = \frac{1}{n}\sum_{i=1}^n \nabla_{\mathbf{x}} f_i(\mathbf{x}^r)$. Then, the average of

iterates can be expressed as:

$$\begin{aligned}\bar{\mathbf{x}}^{r+1} &= \bar{\mathbf{x}}^r - \frac{\alpha}{n}\mathbb{1}^T\mathbf{y}^r = \bar{\mathbf{x}}^r - \frac{\alpha}{n}\mathbb{1}^T(\mathbf{y}^r - \mathbb{1}\bar{\underline{\mathbf{y}}}^r + \mathbb{1}\bar{\underline{\mathbf{y}}}^r) \\ &= \bar{\mathbf{x}}^r - \alpha\bar{\underline{\mathbf{y}}}^r - \frac{\alpha}{n}\mathbb{1}^T(\mathbf{y}^r - \mathbb{1}\bar{\underline{\mathbf{y}}}^r). \end{aligned} \tag{4.9}$$

Further, the tracked full gradient can be expressed as:

$$\bar{\mathbf{y}}^{r+1} = \bar{\mathbf{y}}^r + \bar{g}(\mathbf{x}^{r+1}) - \bar{g}(\mathbf{x}^r), \tag{4.10}$$

where we denote the average of the randomly sampled gradient as $\bar{g}(\mathbf{x}^r) \triangleq \frac{1}{n}\mathbb{1}^T\nabla G(\mathbf{x}^r, \xi^r)$. Note that for notational simplicity we dropped ξ^r in $\bar{g}(\mathbf{x}^r)$.

Further, we define $\mathcal{F}^r \triangleq \{\xi^r, \ldots, \xi^0, \mathbf{x}^r, \ldots, \mathbf{x}^0\}$ as the history of the random samples.

Next, we are ready to provide a convergence rate analysis of the proposed method, by following the steps below.

Step 1. We bound the variance of the tracked full gradient compared to its deterministic counterpart by using Lemma 5.

Step 2. We analyze the dynamics of $f(\bar{\mathbf{x}}^r)$ by Lemma 6.

Step 3. We establish the contraction property of the $\mathbf{x}$ and $\mathbf{y}$ update by Lemma 7.

Step 4. We construct a potential function $P(\mathbf{x}^r)$ and show that it is monotonic decreasing as the algorithm proceeds, by using Lemma 8.

Step 5. We show in Theorem 4 that the expected optimality gap decreases in the order of $O(\frac{1}{\sqrt{T}})$, where T denotes the number of iterations.

Lemma 5 *(Bounded Variance) The iterates $\{\mathbf{y}^r\}$ are generated by GNSD. Under assumption A, we have*

$$\mathbb{E}\|\mathbf{y}^r - \underline{\mathbf{y}}^r\|^2 \leq \kappa\sigma^2, \tag{4.11}$$

where $\kappa \triangleq (1+\widetilde{\eta}/(1-\eta))^2 n^2$ and $\|\mathbf{W}-\mathbf{I}\| \triangleq \widetilde{\eta}$.

Proof 2 *From* (4.7b) *and* (4.8) *and using the triangle inequality, we have*

$$\begin{aligned}&\|\mathbf{y}^{r+1} - \underline{\mathbf{y}}^{r+1}\| \overset{(a)}{\leq} \|\nabla G(\mathbf{x}^{r+1}, \xi^{r+1}) - \nabla F(\mathbf{x}^{r+1})\| \\ &+ \|\mathbf{W}(\mathbf{y}^r - \underline{\mathbf{y}}^r) - (\nabla G(\mathbf{x}^r, \xi^r) - \nabla F(\mathbf{x}^r))\| \\ &\overset{(b)}{\leq} \|\nabla G(\mathbf{x}^{r+1}, \xi^{r+1}) - \nabla F(\mathbf{x}^{r+1})\| \qquad (4.12) \\ &+ \|(\mathbf{W}-\mathbf{I})(\nabla G(x^r, \xi^r) - \nabla F(\mathbf{x}^r))\| \\ &+ \|\mathbf{W}(\mathbf{W}-\mathbf{I})(\nabla G(\mathbf{x}^{r-1}, \xi^{r-1}) - \nabla F(\mathbf{x}^{r-1}))\| + \ldots \qquad (4.13)\end{aligned}$$

where in (b) we use inequality (a) recursively. Then take expectation over $\mathcal{F}^{r+1}$ conditioned on $\mathcal{F}^r$ on both sides, we have

$$\begin{aligned}\mathbb{E}\|\mathbf{y}^{r+1} - \underline{\mathbf{y}}^{r+1}\| &\leq n\sigma + \|(\mathbf{W}-\mathbf{I})(\nabla G(x^r, \xi^r) - \nabla F(\mathbf{x}^r))\| \qquad (4.14) \\ &+ \ldots + \|\mathbf{W}^{r-1}(\mathbf{W}-\mathbf{I})(\nabla G(x^0, \xi^0) - \nabla F(\mathbf{x}^0))\| \qquad (4.15)\end{aligned}$$

where we use the variance of the estimated gradient is upper bounded, i.e., $\mathbb{E}\|\nabla G(\mathbf{x}^{r+1},\xi^{r+1}) - \nabla F(\mathbf{x}^{r+1})\| \leq n\sigma$. *By leveraging the above fact, we take expectation on* $\mathcal{F}^r$ *conditioned on* $\mathcal{F}^{r-1}$ *on both sides of* (4.15) *recursively until* $r = 1$. *Then, due to A3 and the property of conditional expectation, we can get*

$$\begin{aligned}
&\mathbb{E}\|\mathbf{y}^{r+1} - \underline{\mathbf{y}}^{r+1}\| \\
&\leq \left(1 + \|\mathbf{W} - \mathbf{I}\| + \|\mathbf{W}(\mathbf{W} - \mathbf{I})\| + \ldots + \|\mathbf{W}^{r-1}(\mathbf{W} - \mathbf{I})\|\right) n\sigma \\
&\overset{(a)}{\leq} \left(1 + \frac{\widetilde{\eta}}{1-\eta}\right) n\sigma,
\end{aligned}$$

where in (a) *we use* $\|\mathbf{W}(\mathbf{W} - \mathbf{I})\| \leq |\underline{\lambda}_{\max}(\mathbf{W})|\|\mathbf{W} - \mathbf{I}\|$ *and* $|\underline{\lambda}_{\max}(\mathbf{W})| < 1$ *since we know* $(\mathbf{W} - \mathbf{I})\mathbb{1} = 0$ *(i.e.,* $\mathbf{W} - \mathbf{I}$ *lies in the null space of* $\mathbb{1}$*) and* $\mathbf{W}\mathbb{1} = \mathbb{1}$ *by condition A4.*

Lemma 6 *(Descent Lemma) Assume the sequence* $(\mathbf{x}^r, \mathbf{y}^r)$ *is generated by Algorithm 1. We have*

$$\begin{aligned}
\mathbb{E}\left[f(\bar{\mathbf{x}}^{r+1})\right] \leq \mathbb{E}\left[f(\bar{\mathbf{x}}^r)\right] - \left(\alpha - \left(\frac{\alpha\beta}{2} + \alpha^2 L\right)\right)\mathbb{E}\|\bar{\underline{\mathbf{y}}}^r\|^2 \\
+ \frac{\alpha}{2\beta}\frac{L^2}{n}\mathbb{E}\|\mathbf{x}^r - \mathbb{1}\bar{\mathbf{x}}^r\|^2 + \frac{\alpha^2 L\sigma^2}{n},
\end{aligned}$$

where β *is some constant.*

Proof 3 *According the Lipschitz continuity, we have*

$$\begin{aligned}
f(\bar{\mathbf{x}}^{r+1}) &\leq f(\bar{\mathbf{x}}^r) + \langle \nabla f(\bar{\mathbf{x}}^r), \bar{\mathbf{x}}^{r+1} - \bar{\mathbf{x}}^r\rangle + \frac{L}{2}\|\bar{\mathbf{x}}^{r+1} - \bar{\mathbf{x}}^r\|^2 \\
&\overset{(4.9)}{=} f(\bar{\mathbf{x}}^r) - \alpha\langle \nabla f(\bar{\mathbf{x}}^r), \bar{\underline{\mathbf{y}}}^r\rangle - \alpha\langle \nabla f(\bar{\mathbf{x}}^r), \frac{1}{n}\mathbb{1}^T(\mathbf{y}^r - \mathbb{1}\bar{\underline{\mathbf{y}}}^r)\rangle \\
&\quad + \frac{\alpha^2 L}{2}\|\bar{\underline{\mathbf{y}}}^r - \frac{1}{n}\mathbb{1}^T(\mathbf{y}^r - \mathbb{1}\bar{\underline{\mathbf{y}}}^r)\|^2 \\
&\leq f(\bar{\mathbf{x}}^r) + \frac{\alpha}{2\beta}\|\nabla f(\bar{\mathbf{x}}^r) - \bar{\underline{\mathbf{y}}}^r\|^2 + \frac{\alpha\beta}{2}\|\bar{\underline{\mathbf{y}}}^r\|^2 - \alpha\|\bar{\underline{\mathbf{y}}}^r\|^2 \\
&\quad - \alpha\langle \nabla f(\bar{\mathbf{x}}^r), \frac{1}{n}\mathbb{1}^T(\mathbf{y}^r - \underline{\mathbf{y}}^r)\rangle - \alpha\langle \nabla f(\bar{\mathbf{x}}^r), \frac{1}{n}\mathbb{1}^T(\underline{\mathbf{y}}^r - \mathbb{1}\bar{\underline{\mathbf{y}}}^r)\rangle \\
&\quad + \alpha^2 L\|\bar{\underline{\mathbf{y}}}^r\|^2 + \alpha^2 L\|\bar{\mathbf{y}}^r - \mathbb{1}\bar{\underline{\mathbf{y}}}^r\|^2 \\
&\leq f(\bar{\mathbf{x}}^r) + \frac{\alpha}{2\beta}\|\nabla f(\bar{\mathbf{x}}^r) - \bar{\underline{\mathbf{y}}}^r\|^2 + \frac{\alpha\beta}{2}\|\bar{\underline{\mathbf{y}}}^r\|^2 - \alpha\|\bar{\underline{\mathbf{y}}}^r\|^2 \\
&\quad - \alpha\langle \nabla f(\bar{\mathbf{x}}^r), \frac{1}{n}\mathbb{1}^T(\mathbf{y}^r - \underline{\mathbf{y}}^r)\rangle + \alpha^2 L\|\bar{\underline{\mathbf{y}}}^r\|^2 + \alpha^2 L\|\bar{\mathbf{y}}^r - \mathbb{1}\bar{\underline{\mathbf{y}}}^r\|^2,
\end{aligned}$$

where the first inequality use the variants of the Cauchy-Schwarz inequality $\langle a, b\rangle \leq \frac{1}{2\beta}\|a\|^2 + \frac{\beta}{2}\|b\|^2$ *in which* β *is some parameter that can be tuned later, and the last inequality we use the fact that* $\mathbb{1}^T(\underline{\mathbf{y}}^r - \mathbb{1}\bar{\underline{\mathbf{y}}}^r) = 0$.

Taking expectation on both sides, according to the unbiasedness assumption (4.3), *we have*

$$\mathbb{E}_{\mathcal{F}^{r+1}}[\langle \nabla f(\bar{\mathbf{x}}^r), \frac{1}{n}\mathbb{1}^T(\mathbf{y}^r - \underline{\mathbf{y}}^r)\rangle | \mathcal{F}^r] = 0.$$

Then we have

$$\begin{aligned}
&\mathbb{E}\left[f(\bar{\mathbf{x}}^{r+1})\right] \overset{(a)}{\leq} \mathbb{E}\left[f(\bar{\mathbf{x}}^{r})\right] + \frac{\alpha}{2\beta}\mathbb{E}\|\nabla f(\bar{\mathbf{x}}^{r}) - \bar{\underline{\mathbf{y}}}^{r}\|^2 + \frac{\alpha\beta}{2}\mathbb{E}\|\bar{\underline{\mathbf{y}}}^{r}\|^2 \\
&- \alpha\mathbb{E}\|\bar{\underline{\mathbf{y}}}^{r}\|^2 + \alpha^2 L\mathbb{E}\|\bar{\underline{\mathbf{y}}}^{r}\|^2 + \frac{\alpha^2 L\sigma^2}{n} \\
&\leq \mathbb{E}\left[f(\bar{\mathbf{x}}^{r})\right] + \left(-\alpha + \frac{\alpha\beta}{2} + \alpha^2 L\right)\mathbb{E}\|\bar{\underline{\mathbf{y}}}^{r}\|^2 + \frac{\alpha}{2\beta}\frac{L^2}{n}\mathbb{E}\|\mathbf{x}^{r} - \mathbb{1}\bar{\mathbf{x}}^{r}\|^2 \\
&+ \frac{\alpha^2 L\sigma^2}{n},
\end{aligned}$$

where (a) *is true because* $\mathbb{E}\|\bar{\mathbf{y}}^{r} - \bar{\underline{\mathbf{y}}}^{r}\|^2 \leq \sigma^2/n$*, and the last inequality we use* $\mathbb{E}\|\nabla f(\bar{\mathbf{x}}^{r}) - \bar{\underline{\mathbf{y}}}^{r}\|^2 \leq \frac{1}{n}\mathbb{E}\|\mathbf{x}^{r} - \mathbb{1}\bar{\mathbf{x}}^{r}\|^2$ *by applying assumption A1 and Lemma 5.*

Lemma 7 *(Iterates Contraction) Using the assumption of* $\mathbf{W}$*, we have following contraction property of iterates generated by GNSD:*

$$\begin{aligned}
&\mathbb{E}\|\mathbf{x}^{r+1} - \mathbb{1}\bar{\mathbf{x}}^{r+1}\|^2 \leq (1+\beta)\eta^2\mathbb{E}\|\mathbf{x}^{r} - \mathbb{1}\bar{\mathbf{x}}^{r}\|^2 \\
&+ 3(1+\frac{1}{\beta})\alpha^2\mathbb{E}\|\underline{\mathbf{y}}^{r} - \mathbb{1}\bar{\underline{\mathbf{y}}}^{r}\|^2 + 6(1+\frac{1}{\beta})\alpha^2\kappa\sigma^2 \\
&\mathbb{E}\|\mathbf{y}^{r+1} - \mathbb{1}\bar{\mathbf{y}}^{r+1}\|^2 \leq 4nL^2\alpha^2(1+\frac{1}{\beta})^2\|\bar{\underline{\mathbf{y}}}^{r}\|^2 \\
&+ \left(L^2\eta^2(1+\beta)(1+\frac{1}{\beta}) + 4L^2(1+\frac{1}{\beta})^2\right)\mathbb{E}\|\mathbf{x}^{r} - \mathbb{1}\bar{\mathbf{x}}^{r}\|^2 \\
&+ \left((1+\beta)\eta^2 + 4L^2\alpha^2(1+\frac{1}{\beta})^2\right)\mathbb{E}\|\underline{\mathbf{y}}^{r} - \mathbb{1}\bar{\underline{\mathbf{y}}}^{r}\|^2 \\
&+ 4L^2\alpha^2(1+\frac{1}{\beta})^2\kappa\sigma^2
\end{aligned}$$

where β *is some constant such that* $(1+\beta)\eta^2 < 1$ *and* $\|\mathbf{I} - \frac{1}{n}\mathbb{1}\mathbb{1}^T\| \leq 1$*.*

Proof 4 *First, using the assumption of* $\mathbf{W}$*, we can obtain the contraction property of the iterates, i.e.,*

$$\|\mathbf{W}\mathbf{x}^{r} - \mathbb{1}\bar{\mathbf{x}}^{r}\| = \|\mathbf{W}(\mathbf{x}^{r} - \mathbb{1}\bar{\mathbf{x}}^{r})\| \leq \eta\|\mathbf{x}^{r} - \mathbb{1}\bar{\mathbf{x}}^{r}\| \tag{4.16}$$

where the inequality comes from $\mathbb{1}^T(\mathbf{x}^{r} - \mathbb{1}\bar{\mathbf{x}}^{r}) = 0$*, i.e.,* $\mathbf{x}^{r} - \mathbb{1}\bar{\mathbf{x}}^{r} \in col(\mathbf{W})$ *and* $|\lambda_{\max}(\mathbf{W})| = \eta < 1$*.*

Then applying the definition of (4.7a) *and the Cauchy-Schwartz inequality, we have*

$$\begin{aligned}
&\|\mathbf{x}^{r+1} - \mathbb{1}\bar{\mathbf{x}}^{r+1}\|^2 = \|\mathbf{W}\mathbf{x}^{r} - \alpha\mathbf{y}^{r} - \mathbb{1}(\bar{\mathbf{x}}^{r} - \alpha\bar{\mathbf{y}}^{r})\|^2 \\
&\leq (1+\beta)\|\mathbf{W}\mathbf{x}^{r} - \mathbb{1}\bar{\mathbf{x}}^{r}\|^2 + (1+\frac{1}{\beta})\alpha^2\|\mathbf{y}^{r} - \mathbb{1}\bar{\mathbf{y}}^{r}\|^2 \\
&\leq (1+\beta)\eta^2\|\mathbf{x}^{r} - \mathbb{1}\bar{\mathbf{x}}^{r}\|^2 + 3(1+\frac{1}{\beta})\alpha^2\|\mathbf{y}^{r} - \underline{\mathbf{y}}^{r}\|^2 \\
&+ 3(1+\frac{1}{\beta})\alpha^2\|\underline{\mathbf{y}}^{r} - \mathbb{1}\bar{\underline{\mathbf{y}}}^{r}\|^2 + 3(1+\frac{1}{\beta})\alpha^2\|\mathbb{1}\bar{\mathbf{y}}^{r} - \mathbb{1}\bar{\underline{\mathbf{y}}}^{r}\|^2
\end{aligned}$$

in which β *is some constant parameter can be tuned later. Take exceptions on both sides we have the desired results.*

Similarly, applying the definition of $\underline{\mathbf{y}}^r$ *shown in* (4.8)*, we have*

$$\begin{aligned}
&\|\underline{\mathbf{y}}^{r+1} - \mathbb{1}\bar{\underline{\mathbf{y}}}^{r+1}\|^2 = \|\mathbf{W}\underline{\mathbf{y}}^r + \nabla_{\mathbf{x}} F(\mathbf{x}^{r+1}) - \nabla_{\mathbf{x}} F(\mathbf{x}^r) \\
&- \frac{1}{n}\mathbb{1}\mathbb{1}^T(\mathbf{W}\underline{\mathbf{y}}^r + \nabla_{\mathbf{x}} F(\mathbf{x}^{r+1}) - \nabla_{\mathbf{x}} F(\mathbf{x}^r))\|^2 \\
&\leq (1+\beta)\eta^2\|\underline{\mathbf{y}}^r - \mathbb{1}\bar{\underline{\mathbf{y}}}^r\|^2 + (1+\frac{1}{\beta})\|\nabla_{\mathbf{x}} F(\mathbf{x}^{r+1}) - \nabla_{\mathbf{x}} F(\mathbf{x}^r)\|^2.
\end{aligned}$$

Therefore, combining the following

$$\begin{aligned}
&\|\nabla_{\mathbf{x}} F(\mathbf{x}^{r+1}) - \nabla_{\mathbf{x}} F(\mathbf{x}^r)\|^2 \leq L^2\|\mathbf{x}^{r+1} - \mathbf{x}^r\|^2 \\
&= L^2\|\mathbf{W}\mathbf{x}^r - \mathbf{x}^r - \alpha\mathbf{y}^r\|^2 \\
&= L^2\|\mathbf{W}(\mathbf{x}^r - \mathbb{1}\bar{\mathbf{x}}^r) + \mathbb{1}\bar{\mathbf{x}}^r - \mathbf{x}^r - \alpha\mathbf{y}^r\|^2 \\
&\leq L^2\eta^2(1+\beta)\|\mathbf{x}^r - \mathbb{1}\bar{\mathbf{x}}^r\|^2 + L^2(1+\frac{1}{\beta})\|\mathbb{1}\bar{\mathbf{x}}^r - \mathbf{x}^r - \alpha\mathbf{y}^r\|^2 \\
&\leq L^2\eta^2(1+\beta)\|\mathbf{x}^r - \mathbb{1}\bar{\mathbf{x}}^r\|^2 + 4L^2(1+\frac{1}{\beta})\|\mathbf{x}^r - \mathbb{1}\bar{\mathbf{x}}^r\|^2 \\
&+ \frac{4L^2\alpha^2}{n^2}(1+\frac{1}{\beta})\|\mathbf{y}^r - \underline{\mathbf{y}}^r\|^2 + 4L^2\alpha^2(1+\frac{1}{\beta})\|\underline{\mathbf{y}}^r - \mathbb{1}\bar{\underline{\mathbf{y}}}^r\|^2 \\
&+ 4L^2\alpha^2(1+\frac{1}{\beta})\|\mathbb{1}\bar{\underline{\mathbf{y}}}^r\|^2,
\end{aligned}$$

we have

$$\begin{aligned}
&\|\underline{\mathbf{y}}^{r+1} - \mathbb{1}\bar{\underline{\mathbf{y}}}^{r+1}\|^2 \leq \left((1+\beta)\eta^2 + 4L^2\alpha^2(1+\frac{1}{\beta})^2\right)\|\underline{\mathbf{y}}^r - \mathbb{1}\bar{\underline{\mathbf{y}}}^r\|^2 \\
&+ \left(L^2\eta^2(1+\beta)(1+\frac{1}{\beta}) + 4L^2(1+\frac{1}{\beta})^2\right)\|\mathbf{x}^r - \mathbb{1}\bar{\mathbf{x}}^r\|^2 \\
&+ 4nL^2\alpha^2(1+\frac{1}{\beta})^2\|\bar{\underline{\mathbf{y}}}^r\|^2 + 4L^2\alpha^2(1+\frac{1}{\beta})^2\|\mathbf{y}^r - \underline{\mathbf{y}}^r\|^2
\end{aligned}$$

After taking exceptions on both sides and applying Lemma 5 the proof is complete.

Lemma 8 *(Potential Function) Constructing the potential function*

$$P(\mathbf{x}^r) \triangleq \mathbb{E}\left[f(\bar{\mathbf{x}}^r)\right] + \frac{L^2\alpha}{2\beta^2\eta^2}\mathbb{E}\|\mathbf{x}^r - \mathbb{1}\bar{\mathbf{x}}^r\|^2 + \alpha^2\mathbb{E}\|\underline{\mathbf{y}}^r - \mathbb{1}\bar{\underline{\mathbf{y}}}^r\|^2,$$

then we have

$$\begin{aligned}
P(\mathbf{x}^{r+1}) - P(\mathbf{x}^r) &\leq -C_1\alpha\mathbb{E}\|\bar{\underline{\mathbf{y}}}^r\|^2 - \frac{L^2\alpha}{2\beta^2\eta^2}C_2\mathbb{E}\|\mathbf{x}^r - \mathbb{1}\bar{\mathbf{x}}^r\|^2 \\
&- \alpha^2 C_3\mathbb{E}\|\underline{\mathbf{y}}^r - \mathbb{1}\bar{\underline{\mathbf{y}}}^r\|^2 + \frac{\alpha^2 L\sigma^2}{n} + C_4 L^2\alpha^3\kappa n^2\sigma^2,
\end{aligned} \tag{4.17}$$

where constants are defined as follows:

$$
\begin{aligned}
C_1 &\triangleq 1-\frac{\beta}{2}-\alpha L-4nL^2\alpha^3(1+\frac{1}{\beta})^2,\\
C_2 &\triangleq 1-(1+\beta)\eta^2-\frac{\beta\eta^2}{n}\\
&\quad -2\beta\alpha(\eta^2(1+\beta)(1+\frac{1}{\beta})-4(1+\frac{1}{\beta})^2),\\
C_3 &\triangleq 1-(1+\beta)\eta^2-4L^2\alpha^2(1+\frac{1}{\beta})^2-\frac{3(1+\frac{1}{\beta})\alpha L^2}{2\beta^2\eta^2},\\
C_4 &\triangleq 3\frac{1+\frac{1}{\beta}}{\beta^2\eta^2}+4\alpha(1+\frac{1}{\beta})^2L^2.
\end{aligned}
$$

Proof 5 *Combining Lemma 6 and Lemma 7, and by definition of* $P(\mathbf{x}^r)$*, we can get* (4.17) *after some simple manipulations.*

Theorem 4 *If we pick* $\alpha\sim\mathcal{O}(\frac{1}{\sqrt{T/n}})$*, then we have*

$$
\frac{1}{T}\sum_r \mathbb{E}\|\bar{\underline{\mathbf{y}}}^r\|^2+\mathbb{E}\|\mathbf{x}^r-\mathbb{1}\bar{\mathbf{x}}^r\|^2\le\mathcal{O}\left(\frac{\sigma^2}{\sqrt{nT}}\right)
$$

where T *is large.*

Proof 6 *Given* $\eta<1$*, first pick up* $\beta\le 1$*. Then to have* $C_1\ge 0$ *we must have* $0\le\alpha\le K_1$*, where*

$$
K_1\triangleq\min\left\{\frac{1}{\sqrt{nL}(1+\frac{1}{\beta})},\frac{1}{2L}\right\}.
$$

Similarly, choose $\beta\in(0,1)$ *such that* $(1+(1+1/n)\beta)\eta^2<1$*. in order to have* $C_2\ge 0$ *and* $C_3\ge 0$*, we require that step-size* α *needs to satisfy* $0\le\alpha\le K_2$ *and* $0\le\alpha\le K_3$*, respectively, where*

$$
\begin{aligned}
K_2 &\triangleq \frac{1-(1+(1+1/n)\beta)\eta^2}{2\beta((1+\beta)(1+\frac{1}{\beta})\eta^2+4(1+\frac{1}{\beta})^2}\\
K_3 &\triangleq \frac{-1.5L+\sqrt{9L^2/4+16\beta^4\eta^4(1-(1+\beta)\eta^2)}}{8L(1+\frac{1}{\beta})\beta^2\eta^2}
\end{aligned}
$$

Therefore, if we choose $C_0\triangleq\min\{K_1,K_2,K_3\}$ *and* $\alpha=\frac{C_0}{\sqrt{T/n}}$*, we have* $C_1\ge 0, C_2\ge 0, C_3\ge 0$*. Next, we divide* α *on both sides of* (4.17) *in Lemma 8 and apply the telescope sum. Finally, we can obtain*

$$
\begin{aligned}
&\frac{1}{T}\sum_r\left(C_1\mathbb{E}\|\bar{\underline{\mathbf{y}}}^r\|^2+\frac{L^2C_2}{2\beta^2\eta^2}\mathbb{E}\|\mathbf{x}^r-\mathbb{1}\bar{\mathbf{x}}^r\|^2\right)\\
&\qquad\le\frac{\sqrt{n}(P^0-\underline{P})}{C_0T^{3/2}}+\frac{L\sigma^2C_0}{\sqrt{nT}}+\frac{C_4C_0^2L^2\kappa n^3\sigma^2}{T},\quad(4.18)
\end{aligned}
$$

where $\underline{P}$ *denotes the lower bound of* $P(\mathbf{x}^r)$*. Obviously, when* T *is large, term* $\frac{L\sigma^2C_0}{\sqrt{nT}}$ *dominates the convergence rate of GNSD.*

Remark 4. Alternatively, we can also choose step-size as $\mathcal{O}(1/\sqrt{r})$, which will result that the convergence rate of the algorithm is $\mathcal{O}(\log(T)/\sqrt{T})$.

4.5 Numerical Results

In this section, we present the numerical performance of GNSD compared with the existing works, i.e., the DSG [3] and D^2 [99]. We evaluate the performance of the algorithms with an *optimality gap* defined as the sum of local error and consensus error as the following,

$$\mathcal{G}(\mathbf{x}_1, \ldots, \mathbf{x}_n) \triangleq \|\frac{1}{n}\sum_{i=0}^{n}\sum_{j=0}^{B}\nabla f(\mathbf{x}_i, \xi_j)\|^2 + \sum_{i=0}^{n}\|\bar{\mathbf{x}} - \mathbf{x}_i\|^2 \tag{4.19}$$

where B denotes the batchsize. In all simulations, tested algorithms use diminishing step-sizes $D_0/(10+\sqrt{r})$ where D_0 denotes a constant. The (initial) step-sizes of each algorithm are chosen with binary search to get the best performance. Also if not otherwise specified, the batchsize is chosen as 1.

First, we compare the convergence performance of the algorithm on the binary classification problem using the metropolis mixing matrix (shown in Figure 4.1) and a shifted mixing matrix (shown in Figure 4.2). The optimality gaps is averaged over different types of graphs. Next, we evaluate the convergence performance of the algorithms on training the convolutional neural network (CNN) model when the distributions of data one each agent are different in Figure 4.3 and Figure 4.4. It can be observed that GNSD converges faster and better than the other algorithms. In Figure 4.6, we show the impact of different batch sizes on the convergence performance of the algorithm in the CNN model training problem when different agents have different data distributions.

In our simulation, we test the algorithms using different kinds of undirected graphs with 10 and 20 agents, including a fully connected graph, a circle graph, a star graph and Erdős–Rényi random graphs with density 0.5 for the network including 10 agents and densities 0.4 and 0.2 for the network having 20 agents. The mixing matrices used include metropolis weight $\mathbf{W}^{\mathsf{M}}$ and a shifted version $\mathbf{W}^{\mathsf{S}}$.

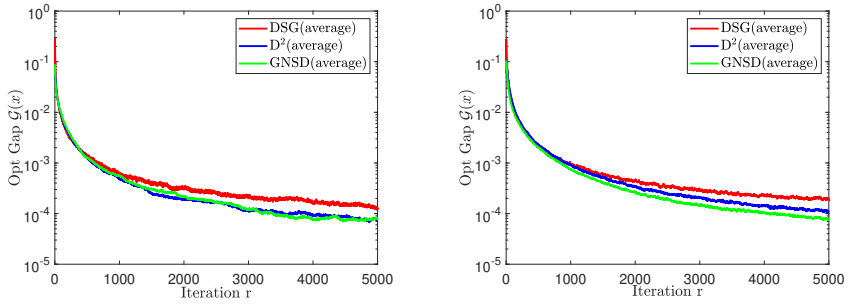
$$w_{ij}^{\mathsf{M}} \triangleq \left\{ \begin{array}{ll} \frac{1}{\max\{d_i, d_j\}+1}, & \text{for } (i,j) \in \mathcal{E}, \\ 0, & \text{for } (i,j) \notin \mathcal{E} \text{ and } i \neq j, \\ 1 - \sum_{j \neq i} w_{ij}^{\mathsf{M}}, & \text{for } i = j. \end{array} \right\}, \tag{4.20}$$

$$\mathbf{W}^{\mathsf{S}} \triangleq \frac{I + 2\mathbf{W}^{\mathsf{M}}}{3}. \tag{4.21}$$

If not specified, the algorithms use the shifted version. All algorithms are implemented in

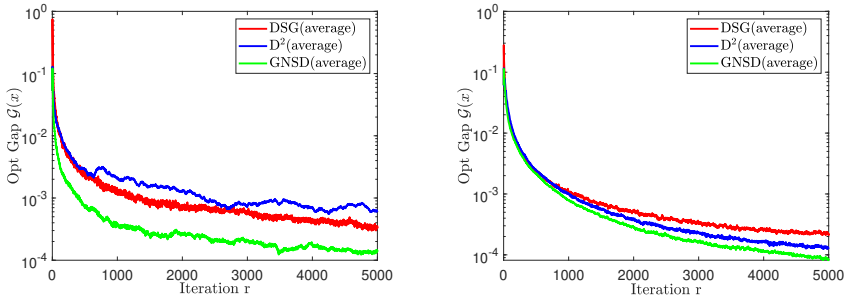
Python, run on multiple Intel Haswell E5-2680v3 processors in the Minnesota Supercomputing Institute (MSI), each agent in the graph is assigned with a processor core and allowed to communicate with each other using MPI [100].

We consider two learning models in the numerical experiments: penalized likelihood regression [86] for binary classification by synthetic dataset, and CNN for multi-classification by the MNIST dataset. In the non-convex binary classification problem [86], the feature vectors are randomly generated, following standard normal distribution $\mathcal{N}(0, \mathbf{I})$ except the first entries are fixed to be 1s, and the labels are also generated randomly, following uniform distribution in $\{-1, 1\}$. In the multi-classification problem, we consider the decentralized training problem by CNN on MNIST [101] dataset. The CNN model is built with TensorFlow, constructed with three 3×3 convolutional layers using sigmoid function defined by $f(x) = 1/(1 + e^x)$ as the rectifier, one average pooling layer and one fully connected layer. The classification loss is evaluated by cross-entropy, so it is a smooth non-convex problem. In the balanced case, each agent randomly takes 300 different samples in the MNIST dataset as its training dataset. In the unbalanced case, which is used as default, each agent takes total 300 samples from 2 classes in the MNIST dataset as its training dataset, different agents take samples from different classes, so the variation of the data is very large.



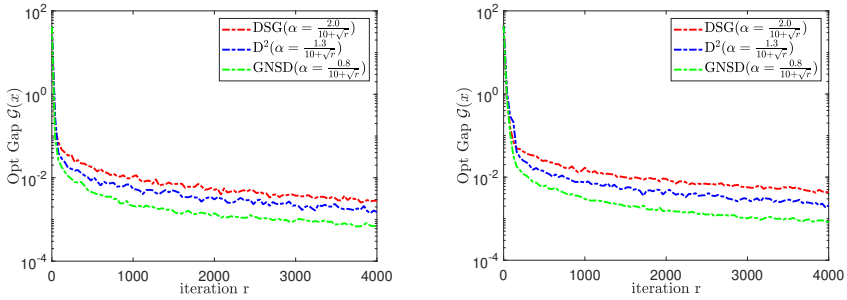
(a) 10 agents (b) 20 agents

Figure 4.1: Optimality gap (averaged over different types of graphs) of DSG, D^2 and GNSD algorithms in solving binary classification problem using metropolis weight with a) 10 agents; b) 20 agents.



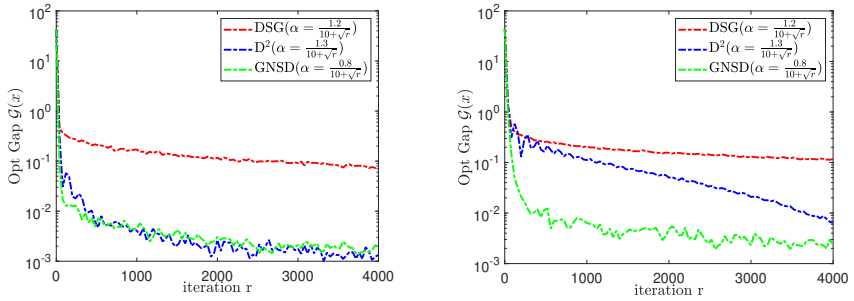
(a) 10 agents (b) 20 agents

Figure 4.2: Optimality gap (averaged over different types of graphs) of DSG, D^2 and GNSD algorithms in solving binary classification problem using shifted metropolis weight with a) 10 agents; b) 20 agents.



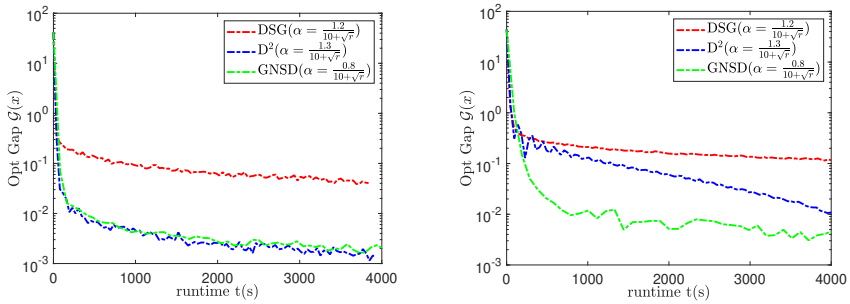
(a) 10 agents (b) 20 agents

Figure 4.3: The average optimality gap (averaged over different types of graphs) of DSG, D^2 and GNSD algorithms in training the CNN model with balanced data a) 10 agents; b) 20 agents.



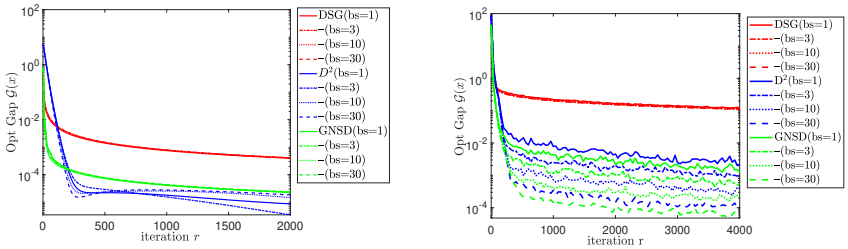
(a) 10 agents (b) 20 agents

Figure 4.4: The average optimality gap (averaged over different types of graphs) of DSG, D^2 and GNSD algorithms in training the CNN model with unbalanced data a) 10 agents; b) 20 agents.



(a) 10 agents (b) 20 agents

Figure 4.5: Optimality gap of DSG, D^2 and GNSD algorithms in training the CNN model under random graphs with respect to runtime t with a) 10 agents; b) 20 agents.



(a) Binary classification (b) CNN model

Figure 4.6: Optimality gap of DSG, D^2 and GNSD algorithms in random graph with different batchsizes, a) binary classification problem; b) training a CNN model.

Chapter 5

Primal-dual Based Federated Learning Algorithm

5.1 Motivation

Federated learning (FL)—a distributed machine learning approach proposed in [12]—has gained popularity for applications involving learning from distributed data. In FL, a cloud server (the "server") can communicate with distributed data sources (the "agents"). The goal is to train a global model that works well for all the distributed data, but without requiring the agents to reveal too much local information. Since its inception, the broad consensus on FL's implementation appears to involve a generic "local update" strategy to save communication efforts. The basic communication pattern "computation then aggregation" (CTA) protocol involves the following steps: S1) the server sends the global model $\mathbf{x}$ to the agents; S2) the agents update their local models $\mathbf{x}_i$'s based on their local data for several iterations; S3) the server aggregates $\mathbf{x}_i$'s to obtain a new global model $\mathbf{x}$. The CTA protocol is popular, partly because transmitting local gradients and other statistics to the server is undesirable. For instance, it has been shown that local gradient information can leak private data [43, 102, 45] and increase the cost when applying privacy-preserving methods.

Even though the FL paradigm has attracted significant attention from both academia and industry, and many algorithms such as Federated Averaging (FedAvg) have been proposed [103, 104, 105, 39], several attributes are not clearly established. In particular, the commonly adopted local update strategy poses significant theoretical and practical challenges to designing effective FL algorithms. This work attempts to provide a deeper understanding of FL by raising and resolving key theoretical questions, as well as by developing an effective algorithmic framework

with several desirable features.

5.2 Preliminaries

Problem Formulation. Consider the following problem:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) &\triangleq \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x}), \\ f_i(\mathbf{x}) &\triangleq w_i \sum_{\xi_i\in\mathcal{D}_i} F(\mathbf{x};\xi_i), \end{aligned} \tag{5.1}$$

where ξ_i denotes one sample in data set $\mathcal{D}_i$ stored on the i-th agent; $F : \mathbb{R}^d \to \mathbb{R}$ is the "loss function" for data point ξ_i; and $w_i > 0$ is a "weight coefficient" (a common choice is $w_i = 1/|\mathcal{D}_i|$ [6]). We assume that the loss function takes the same form across different agents, and furthermore, we denote $M := \sum_{i=1}^{N} |\mathcal{D}_i|$ to be the total number of samples. One can also consider a related setting, where each $f_i(\mathbf{x})$ represents the expected loss [39]

$$f_i(\mathbf{x}) \triangleq \mathbb{E}_{\xi_i\in\mathcal{P}_i} F(\mathbf{x};\xi_i), \tag{5.2}$$

where $\mathcal{P}_i$ denotes the data distribution on the i-th agent. Throughout the chapter, we will make the following blanket assumptions for problem (5.1):

A 8 *Each $f_i(\cdot)$, as well as $f(\cdot)$ in (5.1) is L–smooth:*

$$\begin{aligned} &\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \le L\|\mathbf{x}-\mathbf{y}\|, \\ &\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \le L\|\mathbf{x}-\mathbf{y}\|, \ \forall\, \mathbf{x},\mathbf{y}\in\mathbb{R}^d, i = 1,\dots,N. \end{aligned}$$

A 9 *The objective of problem (5.1) is lower bounded: $f(\mathbf{x}) \ge c > -\infty$, $\forall\ \mathbf{x}\in\mathbb{R}^d$.*

In addition to these standard assumptions, state-of-the-art efforts on analysis of FL algorithms oftentimes invoke a number of more *restrictive* assumptions.

A 10 (Bounded Gradient Dissimilarity (BGD)) *[17] The gradients ∇f_i's are upper bounded (by a constant $G > 0$ and $D \ge 0$)*

$$\frac{1}{N}\sum_{i=1}^{N} \|\nabla f_i(\mathbf{x})\|^2 \le G^2 + D^2\|\nabla f(\mathbf{x})\|^2, \quad \forall\ \mathbf{x}\in\mathbb{R}^d. \tag{5.3}$$

Table 5.1: Convergence rates of FL algorithms, measured by total rounds of communication (RC), number of local updates (LC) and number of samples (SC), before reaching ϵ-stationary solution. CVX refers to convexity, NC is non-convex, μSC means μ-Strongly Convex, BGD refers to bounded gradient dissimilarity, CTA refers to CTA protocol and LP refers to solving the local problem to a certain accuracy. p is the function of $\mathcal{O}(\frac{\epsilon}{G^2})$ illustrated in Fig. 5.1.

Algorithm	CVX	BGD	CTA	LP	RC (T)	LC (QT)	SC
FedAvg [15]	μSC	(G,0)	✓	×	$\mathcal{O}\left(1/\epsilon\right)$	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon)$
FedAvg [17]	μSC	(G,D)	✓	×	$\mathcal{O}\left(1/\epsilon^{1/2}\right)$	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon)$
FedSplit [106]	μSC	-	✓	✓	$\mathcal{O}(\log(1/\epsilon))$	$\mathcal{O}(Q\log(1/\epsilon))$	$\mathcal{O}(QB\log(1/\epsilon))$
Local-GD [14]	C	-	✓	×	$\mathcal{O}(1/\epsilon^{3/2})$	$\mathcal{O}(1/\epsilon^{3/2})$	$\mathcal{O}(Q/\epsilon^{3/2})$
FedAc [107]	C	(G,0)	✓	×	$\mathcal{O}(\log(1/\epsilon))$	$\mathcal{O}(\log(1/\epsilon)/\epsilon)$	$\mathcal{O}(\log(1/\epsilon)/\epsilon)$
FedAvg [103]	NC	(G,0)	✓	×	$\mathcal{O}(1/\epsilon^{3/2})$	$\mathcal{O}(1/\epsilon^{2})$	$\mathcal{O}(1/\epsilon^{2})$
FedAvg [17]	NC	(G,D)	✓	×	$\mathcal{O}(1/\epsilon^{3/2})$	$\mathcal{O}(1/\epsilon^{2})$	$\mathcal{O}(1/\epsilon^{2})$
VRL-SGD [108]	NC	-	✓	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^{2})$	$\mathcal{O}(1/\epsilon^{2})$
F-SVRG [109]	NC	-	×	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(Q/\epsilon)$	$\mathcal{O}((M+Q)/\epsilon)$
SCAFFOLD [17]	NC	-	×	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^{2})$	$\mathcal{O}(1/\epsilon^{2})$
FedProx [38]	NC	(0,D)	✓	✓	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(Q/\epsilon)$	$\mathcal{O}(QB/\epsilon)$
Fed-PD	NC	-	✓	✓	$\mathcal{O}\left(1/\epsilon\right)$	$\mathcal{O}(Q/\epsilon)$	$\mathcal{O}(QB/\epsilon)$
Fed-PD	NC	(G,1)	✓	✓	$\mathcal{O}((1-p)/\epsilon)$	$\mathcal{O}(Q(1-p)/\epsilon)$	$\mathcal{O}(QB(1-p)/\epsilon)$
Fed-PD (VR)	NC	-	✓	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(Q/\epsilon)$	$\mathcal{O}(M+\sqrt{M}/\epsilon)$

Let us comment on the two special cases of this assumption. 1) When $D = 0$: this assumption is the so-called bounded gradient (BG) assumption, and it indicates the local gradients are upper bounded by some constant. In early works, the BG assumption is used to bound the deviation between the agents after multiple local updates, which is critical for analyzing FL algorithms; 2) When $G = 0$ and $D = 1$: this assumption indicates that the local functions have the same gradient for all $\mathbf{x}$, or equivalently the distribution of local data is homogeneous. We provided a few commonly used functions that satisfy this assumption in Appendix C.5. Overall, the above assumption can be used to characterize the non-i.i.d.-ness of the local data set – the larger the values of D and G, the higher the level of non-i.i.d.-ness (or data heterogeneity) there is among the local data.

Finally, we mention that our objective is to understand the FL algorithm from an optimization perspective. So we say that a solution $\mathbf{x}$ is an ϵ-stationary solution if the following holds:

$$\|\nabla f(\mathbf{x})\|^2 \leq \epsilon. \tag{5.4}$$

We are interested in finding the *minimum* system resources required, such as the number of

local updates, the number of times local data are transmitted to the server, and the number of times local samples $F(\mathbf{x};\xi_i)$'s are accessed, before computing an ϵ-solution (5.4). These quantities are referred to as *local computation*, *communication complexity*, and *sample complexities*, respectively. Below, we list four questions to be addressed in this work.

Q1 (local updates). What are the best local update directions for the agents to take to achieve the best overall system performance (stability, sample complexity, etc.)?

Q2 (global aggregation). Can we use more sophisticated processing in the aggregation step to improve system performance (sample or communication complexity)?

Q3 (communication efficiency). What is the minimum communication (at each round and in total) to achieve a desired solution accuracy? Can the communication efficiency of FL algorithms be adapted to the local data non-i.i.d.-ness (as defined in Assumption 10)?

Q4 (assumptions). What is the best performance that a CTA type algorithm can achieve while relying on a minimum set of assumptions (e.g. only relying on A8-A9)?

Although these questions are not directly related to data privacy—another important aspect of FL—we argue that answering these fundamental questions can provide a much-needed understanding of the FL approach. A few recent works have touched upon those questions. Still, to our knowledge, none of them have provided a thorough investigation of the questions listed above.

Related Works. We discuss existing algorithms in FL by roughly classifying them based on two considerations: 1) *Communication protocol*: whether the algorithm follows the CTA protocol, i.e., only transfer the models during the communication, or transfer more information; 2) *Local update strategy*: whether the local agents solve a local problem to a certain accuracy, or just perform certain fixed steps of local update. The results are summarized in Table 5.1. It is pertinent to consider how these algorithms address questions Q1–Q4.

Table 5.2: Summary of notation used in the chapter

N, i	total number, and index of clients
M, B, b	total number, batch size and index of samples
T, r	total number and index of communication rounds
Q, q	total number and index of local updates
$\mathbf{x}_0^r$	global model at communication round r
$\mathbf{x}_{0,i}^r$	i^{th} client's estimated global model at round r
$\mathbf{x}_i^{r,q}$	i^{th} client's model at round r and step q
$\mathbf{x}_{0,i}^{r,+}$	the model i^{th} client send to server after round r

To answer Q1, let us review the local steps used for state-of-the-art algorithms. The well-known FedAvg algorithm performs multiple local (stochastic) GD steps to minimize the local loss function between two aggregation steps; see Algorithm 2 below.

Algorithm 2 FedAvg Algorithm

Initialize: $\mathbf{x}_i^0 \triangleq \mathbf{x}^0, i = 1, \ldots, N$
for $r = 0, \ldots, T-1$ *(stage)* **do**
  for $q = 0, \ldots, Q-1$ *(iteration)* **do**
    for $i = 1, \ldots, N$ in parallel **do**
      Local update: $\mathbf{x}_i^{r,q+1} = \mathbf{x}_i^{r,q} - \eta \nabla F(\mathbf{x}_i^{r,q}; \xi_i^{r,q}) \; \forall i$
    end for
  end for
  Global averaging: $\mathbf{x}^{r+1} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i^{r,Q}$
  Update agents' $\mathbf{x}_i^{r+1,0} = \mathbf{x}^{r+1}, \; i = 1, \ldots, N$
end for

However, in most cases, successive local GD steps lead to sub-optimal communication complexity [14, 110]. By using correction terms, FedSplit [106] greatly reduces the communication complexity in the convex setting; VRL-SGD [108] and SCAFFOLD [17] also reduce the communication complexity in certain non-convex settings, but VRL-SGD requires some bounded variance assumption, which essentially implies that the (stochastic) gradients are bounded. Additionally, SCAFFOLD needs to communicate both the local models and the local gradients, which doubles the communication overhead.

For Q2, although most algorithms use simple averaging, F-SVRG [109] and SCAFFOLD break the CTA protocol. F-SVRG shows an improvement in sample complexity, and SCAFFOLD improves the dependence on agent number N compared with VRL-SGD. However, there is little discussion on whether other types of linear processing are helpful or the CTA protocol is enough for FL algorithms.

For Q3, a number of recent works show that a total of $O(1/\epsilon)$ aggregation steps are needed for non-convex problems to achieve ϵ-solution (5.4). However, bounded variance assumption and local statistics are needed. It is not clear if this achieves the best communication complexity.

As for Q4, the algorithms typically require either bounded variance assumption, or some BGD assumption, or both to achieve a good performance. FedSplit shows a possible optimal performance under the strongly convex setting, but the best performance under the non-convex setting remains unknown.

5.3 Properties of CTA Protocols

In this section, we formally address questions raised in the previous section about the CTA protocol.

5.3.1 Communication Lower Complexity Bounds

We first address Q2–Q3 under the CTA protocol. Specifically, for problems satisfying A8–A9, does performing multiple local updates or using different ways to combine local models reduce communication complexity? We show below that under the CTA protocol, such a saving is impossible.

Consider the following generic CTA protocol. Let r denote the index for communication rounds. Between two rounds $r-1$ and r, each agent performs Q local updates. Denote $x_i^{r-1,q}$ to be the q-th local update. Then, $x_i^{r-1,Q}$'s are sent to the server, combined through a (possibly time-varying) function $V^r(\cdot) : \mathbb{R}^{Nd} \to \mathbb{R}^d$, and sent back. The agents then generate a new iterate by combining the received message with past gradients using a (possibly time-varying) function $W_i^r(\cdot)$:

$$x^r = V^r(\{x_i^{r-1,Q}\}_{i=1}^N),\ x_i^{r,0} = x^r,\ \ \forall\ i \in [N], \tag{5.5a}$$

$$x_i^{r,q} \in W_i^r\left(\{x_i^{r,k}, \{\nabla F(x_i^{r,q};\xi_i)\}_{\xi_i \in D_i}\}_{k\in[q-1], r\in[r]}\right),$$
$$\forall\ q \in [Q],\ \forall\ i \in [N]. \tag{5.5b}$$

We focus on the case where the $V^r(\cdot)$'s and $W_i^r(\cdot)$'s are *linear* operators, which implies that $x_i^{r,q}$ can use all past iterates and (sample) gradients for its update. Clearly, (5.5) covers both the local-GD and local-SGD versions of FedAvg as special cases.

In the following, we provide an informal statement of the result. The formal statement and the full proof are given in Theorem 11, which is relegated to Appendix C.6.

Claim 1 (Informal) *Consider any algorithm A that belongs to the class described in* (5.5)*, with $V^r(\cdot)$ and $W_i^r(\cdot)$'s being linear and possibly time-varying operators. Then, there exists a non-convex problem instance satisfying Assumptions 8–9 such that for any $Q > 0$, algorithm A takes at least $\mathcal{O}(1/\epsilon)$ communication rounds to reach an ϵ-stationary solution satisfying* (5.4)*.*

Remark 5. The proof technique is related to those developed from both classical and recent works that characterize lower bounds for first-order methods, in both centralized [111, 112] and decentralized [19, 20] settings. The main technical difference is that our processing model (5.5) additionally allows local processing iterations, and there is a central aggregator. In the proof, we construct problem instances in which f_i's are non-i.i.d. (i.e., G in assumption A10 grows with

the total number of iterations T, and $D = 1$). Then we show that it is necessary to aggregate (thus communicate) to make any progress. On the other hand, it is obvious that in another extreme case where the data are homogeneous (i.e., $G = 0$, $D = 1$), only $\mathcal{O}(1)$ communication rounds are needed. ■

5.3.2 Local Update Strategy and Bounded Gradient

We now address Q1 and Q4. We consider the FedAvg Algorithm and show that when using (stochastic) gradient as the local update direction, the bounded gradient assumption A10 is critical to ensure performance.

Claim 2 *Fix any constant $\eta > 0$, $Q > 1$ for FedAvg. There exists a problem that satisfies A8 and A9 but fails to satisfy A10, on which FedAvg diverges to infinity.*

Due to space limitation, the proof of the above result is relegated to Appendix C.1.

Remark 6. A recent work [14] has shown that FedAvg with *constant* stepsize $\eta > 0$ can only converge to a neighborhood of the global minimizer for *convex* problems. Beyond that, our result indicates that when f_i's are *non-convex*, FedAvg can perform much worse without the BGD assumption. Even if $Q = 2$ and there exists a solution such that $\sum_{i=1}^{N} \|f_i(\hat{\mathbf{x}})\|^2 = 0$, FedAvg (with constant stepsize η) diverges and the iteration can go to ∞. ■

The above result suggests that, despite its popularity, the pure local (stochastic) gradient direction is not compatible with the CTA protocol. This motivates the design of local update strategies that allow the agents to work together properly.

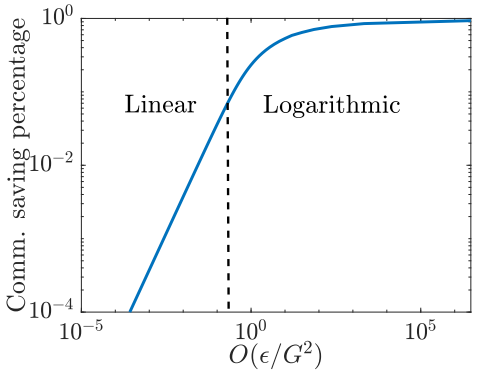


Figure 5.1: Relation of the percentage of comm. savings, accuracy ϵ, heterogeneity G. Details in Section 5.5.

5.4 Algorithm Design

In this section, we propose a meta-algorithm called Federated Primal-Dual (FedPD), which is an efficient algorithm following the CTA protocol. Among many of its features, the FedPD achieves the communication lower bound mentioned in the previous section without requiring additional assumptions such as BGD (5.3). Further, we show that for problems satisfying the BGD assumption (5.3), the proposed algorithm can effectively reduce communication overhead.

Our algorithm is based on the following *global consensus* reformulation of the original problem (5.1):

$$\min_{\mathbf{x}_0, \mathbf{x}_i} \frac{1}{N} \sum_{i=1}^{N} f_i(\mathbf{x}_i), \quad \text{s.t.} \;\; \mathbf{x}_i = \mathbf{x}_0, \; \forall i \in [N]. \tag{5.6}$$

To present our algorithm, let us define the augmented Lagrangian (AL) function of (5.6) as

$$\begin{aligned} \mathcal{L}(\mathbf{x}_{0:N}, \boldsymbol{\lambda}) &\triangleq \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_i(\mathbf{x}_0, \mathbf{x}_i, \lambda_i), \\ \mathcal{L}_i(\mathbf{x}_i, \mathbf{x}_0, \lambda_i) &\triangleq f_i(\mathbf{x}_i) + \langle \lambda_i, \mathbf{x}_i - \mathbf{x}_0 \rangle + \frac{1}{2\eta} \left\| \mathbf{x}_i - \mathbf{x}_0 \right\|^2. \end{aligned}$$

Fixing $\mathbf{x}_0$, the AL is separable over all local pairs $\{(\mathbf{x}_i, \lambda_i)\}$. The key technique in the design is to specify *how* each local AL $\mathcal{L}_i(\cdot)$ should be optimized, and *when* to perform model aggregation.

Federated primal-dual algorithm (FedPD) can be easily implemented in the FL setting, while capturing the main idea of the classical primal-dual based algorithm; see Algorithm 3. In particular, its update rules share a similar pattern as the Alternating Direction Method of Multipliers (ADMM), but it does not specify how the local models are updated. Instead, an *oracle* $\text{Oracle}_i(\cdot)$ is used as a placeholder for local processing, and we will see that careful instantiations of these oracles lead to algorithms with different properties. Importantly, we introduce a critical constant $p \in [0, 1)$, which determines the frequency at which the aggregation and communication steps are skipped. By using FedPD, we can see that at each communication round, only the local models are exchanged. In Algorithm 4 and Algorithm 5 we provide different oracles for FedPD. It is worth noting that Oracle II is based on the idea of variance reduction, and it can achieve a lower sample complexity compared with those in Oracle I. Below, we provide more discussion about these proposed local oracles.

Algorithm 3 Federated Primal-Dual Algorithm

Input: $\mathbf{x}^0, \eta, p, T, Q_1, \ldots, Q_N$

Initialize: $\mathbf{x}_0^0 = \mathbf{x}^0$,

for $r = 0, \ldots, T-1$ **do**

for $i = 1, \ldots, N$ in parallel do local updates **do**

$\mathbf{x}_i^{r+1} = \text{Oracle}_i(\mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_{0,i}^r, \lambda_i^r), Q_i)$

$\lambda_i^{r+1} = \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r)$ #Dual updates

$\mathbf{x}_{0,i}^{r+} = \mathbf{x}_i^{r+1} + \eta\lambda_i^{r+1}$

end for

With probability $1-p$ do global communication:

Global Communicate:

$\mathbf{x}_0^{r+1} = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_{0,i}^{r+}$

$\mathbf{x}_{0,i}^{r+1} = \mathbf{x}_0^{r+1},\ i = 1, \ldots, N$

With probability p skip global communication:

Local Update: $\mathbf{x}_{0,i}^{r+1} \triangleq \mathbf{x}_{0,i}^{r+}$

end for

Algorithm 4 Oracle Choice I

Input: $\mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_{0,i}^r, \lambda_i^r), Q_i$

Initialize: $\mathbf{x}_{i,0}^r = \mathbf{x}_i^r$,

Option I (GD)

for $q = 0, \ldots, Q_i - 1$ **do**

$\mathbf{x}_i^{r,q+1} = \mathbf{x}_i^{r,q} - \eta_1 \nabla_{\mathbf{x}_i} \mathcal{L}(\mathbf{x}_i^{r,q}, \mathbf{x}_{0,i}^r, \lambda_i^r)$

end for

Option II (SGD)

for $q = 0, \ldots, Q_i - 1$ **do**

$\mathbf{x}_i^{r,q+1} = \mathbf{x}_i^{r,q} - \eta_1(h_i(\mathbf{x}_i^{r,q}; \xi_i^{r,q}) + \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r,q} - \mathbf{x}_{0,i}^r))$

end for

Output: $\mathbf{x}_i^{r+1} \triangleq \mathbf{x}_i^{r,Q_i}$

In Algorithm 4, the stochastic gradient is defined as

$$h_i(\mathbf{x}_i^{r,q}; \xi_i^{r,q}) \triangleq \nabla F(\mathbf{x}_i^{r,q}; \xi_i^{r,q}), \text{ with } \xi_i^{r,q} \sim \mathcal{D}_i, \tag{5.7}$$

where $\sim$ denotes uniform sampling. Further, for both options, Q_i's are chosen so that the local problems are solved accurately enough. Specifically, for GD (Option I) we need to ensure that

we run the inner iterations long enough such that the following holds:

$$\left\|\nabla_{\mathbf{x}_i}\mathcal{L}(\mathbf{x}_i^{r+1},\mathbf{x}_{0,i}^r,\lambda_i^r)\right\|^2 \leq \epsilon_1. \tag{5.8}$$

Similarly, for SGD (Option II), we need to assume that the following holds:

$$\mathbb{E}\left\|\nabla_{\mathbf{x}_i}\mathcal{L}(\mathbf{x}_i^{r+1},\mathbf{x}_{0,i}^r,\lambda_i^r)\right\|^2 \leq \epsilon_1. \tag{5.9}$$

Note that in Algorithm 2 we provide two ways for solving this subproblem by using GD and SGD, but any other solver that achieves (5.8) can be used. Despite the simplicity of the local updates, we will show that using Oracle I makes FedPD adaptive to the non-i.i.d. parameter G.

Algorithm 5 Oracle Choice II

Input: $\mathcal{L}_i(\mathbf{x}_i^r,\mathbf{x}_{0,i}^r,\lambda_i^r), Q, I, B$

Initialize: $\mathbf{x}_i^{r,0}=\mathbf{x}_i^r$,

if $r \mod I = 0$ **then** $g_i^{r,0}=\nabla f_i(\mathbf{x}_i^{r,0})$

else $g_i^{r,0}=g_i^{r-1,Q}$

end if

for $q=0,\ldots,Q-1$ **do**

$\mathbf{x}_i^{r,q+1}=\arg\min_{\mathbf{x}_i}\tilde{\mathcal{L}}_i(\mathbf{x}_i,\mathbf{x}_{0,i}^r,\lambda_i^r;\mathbf{x}_i^{r,q},g_i^{r,q})$

$g_i^{r,q+1}=g_i^{r,q}+\frac{1}{B}\sum_{b=1}^{B}(h_i(\mathbf{x}_i^{r,q+1};\xi_{i,b}^{r,q})-h_i(\mathbf{x}_i^{r,q};\xi_{i,b}^{r,q}))$

end for

Output: $\mathbf{x}_i^{r+1}\triangleq\mathbf{x}_i^{r,Q}$, $g_i^{r,Q}$

Alternatively, when instantiating the local oracle using Algorithm 5, the original local problems are not required to solve to ϵ_1 accuracy. Instead, we successively optimize a linearized AL function:

$$\tilde{\mathcal{L}}_i^r(\mathbf{x}_i)\triangleq\tilde{f}_i(\mathbf{x}_i;\mathbf{x}_i^{r,q})+\left\langle\lambda_r^i,\mathbf{x}_i-\mathbf{x}_{0,i}^r\right\rangle+\frac{1}{2\eta}\left\|\mathbf{x}_i-\mathbf{x}_{0,i}^r\right\|^2.$$

In the above expression, we linearize $f_i(\mathbf{x}_i)$ at inner iteration $\mathbf{x}_i^{r,q}$ as

$$\tilde{f}_i^r(\mathbf{x}_i;\mathbf{x}_i^{r,q})\triangleq f(\mathbf{x}_i^{r,q})+\langle g_i^{r,q},\mathbf{x}_i-\mathbf{x}_i^r\rangle+\frac{1}{2\gamma}\left\|\mathbf{x}_i-\mathbf{x}_i^{r,q}\right\|^2,$$

where γ is a constant and $g_i^{r,q}$ is an approximation of $\nabla f_i(\mathbf{x}_i^{r,q})$. The optimizer has a closed-form expression:

$$\mathbf{x}_i^{r,q+1}=\frac{\eta}{\eta+\gamma}\mathbf{x}_i^{r,q}+\frac{\gamma}{\eta+\gamma}\mathbf{x}_{0,i}^r-\frac{\eta\gamma}{\eta+\gamma}(g_i^{r,q}+\lambda_i^r).$$

In Oracle II, an agent i first decides whether to compute the full gradient $\nabla f_i(\mathbf{x}_i^{r,0})$, or to keep using the previous estimate $g_i^{r-1,Q}$. Then Q local steps are performed, each requiring B local data samples. In this scheme, Q can be chosen as *any* positive integer.

It is important to note that this oracle does not simply apply the variance reduction (VR) technique (such as F-SVRG) to solve the subproblem of optimizing $\mathcal{L}_i(\mathbf{x}_i, \mathbf{x}_{0,i}^r, \lambda_i^r)$. That is, it is *not* a variation of Oracle I. Instead, the VR technique is applied to the entire primal-dual iteration, and the full gradient evaluation $\nabla f_i(\mathbf{x}_i^{r,0})$ is only needed every I iteration r. Later we will see that if I is large enough, then there is an $\mathcal{O}(\sqrt{M})$ reduction of sample complexity.

5.5 Convergence Analysis

In this section, we first provide a basic convergence analysis of FedPD without assuming A10 (or effectively, with G in (10) being infinity). Then, we show that with Assumption A10, FedPD allows some communication rounds to be *skipped* when the local functions become similar (that is, when G becomes smaller). We refer the readers to Appendix C.2 for detailed proof of Theorem 5 and 7, Appendix C.4.1 for proof of Theorem 6.

5.5.1 Analysis Without the BGD Assumption

We first characterize the convergence of FedPD with different oracles, without assuming the BGD assumption A10.

Theorem 5 *Suppose A8 –A9 hold. Define $D_0 := f(\mathbf{x}_0^0) - f(\mathbf{x}^\star)$. Consider FedPD with Oracle I, where Q_i's are selected such that* (5.8) *holds if Option I is used, and* (5.9) *holds if Option II is used. Set $0 < \eta < \frac{\sqrt{5}-1}{4L}$, $p = 0$. Then we have:*

$$\frac{1}{T}\sum_{r=0}^{T} \|\nabla f(\mathbf{x}_0^r)\|^2 \le \frac{C_2}{T} D_0 + C_4 \epsilon_1,$$

where C_2, C_4 are constants only depending on L, η, and are independent of T, G, p.

Theorem 6 *Suppose A8–A9 hold. Consider FedPD with Oracle II. Choose $p = 0$, $\gamma > \frac{5\eta}{B\sqrt{L}}$, and $\eta \in \big(0, \frac{1}{3(Q+\sqrt{QI/B})L}\big)$. Then, the following holds (where $C_9 > 0$ is a constant that depends on L, η, B, Q):*

$$\frac{1}{T}\sum_{r=0}^{T} \mathbb{E}\,\|\nabla f(\mathbf{x}_0^r)\|^2 \le \frac{C_9}{T}(f(\mathbf{x}_0^0) - f(\mathbf{x}^\star)). \tag{5.10}$$

Remark 7. For Oracle I to achieve ϵ accuracy, we need to set the communication round $T = C_2 D_0/\epsilon$ and local accuracy $\epsilon_1 = \epsilon/C_4$. As the local AL is strongly convex with respect to $\mathbf{x}_i$, optimizing it to ϵ accuracy requires $Q_i = \mathcal{O}(\log(\epsilon))$ iterations for GD and $Q_i = \mathcal{O}(1/\epsilon)$ for SGD [113]. ■

Remark 8. Suppose Oracle II runs for T communication rounds, the total number of full gradient evaluation is $T/I+1$, each uses M samples. Meanwhile, the total number of mini-batch stochastic gradient evaluation is TQ, each uses $2B$ samples per node. So the total sample complexity is $\mathcal{O}(M+MT/I+2TQBN)$. Therefore, we choose $I=\sqrt{M}, B=I/QN=\sqrt{M}/QN$, then the sample complexity of Algorithm 5 is $\mathcal{O}(M+\frac{\sqrt{M}}{\epsilon})$. ■

5.5.2 Analysis with the BGD Assumption

In this subsection, we analyze how the additional assumption A10 can affect the proposed algorithm. Towards this end, let us consider the following $(G,1)$-BGD assumption (which is equivalent to A10 with $D=1$).

A 11 (G,1)-BGD *The local functions are called $(G,1)$-BGD if either one of the equivalent conditions below holds:*

$$\begin{aligned} &\|\nabla f_i(\mathbf{x})-\nabla f_j(\mathbf{x})\|\le G,\ \forall\ \mathbf{x}\in\mathbb{R}^d,\ \forall\ i\neq j,\\ &\text{or } \|\nabla f_i(\mathbf{x})-\nabla f(\mathbf{x})\|\le G\ \forall\ \mathbf{x}\in\mathbb{R}^d,\ \forall\ i. \end{aligned} \tag{5.11}$$

Theorem 7 *Suppose A8 –A9 and A11 holds. Consider FedPD with Oracle I, where Q_i's are selected such that* (5.8) *holds if Option I is used, and* (5.9) *holds if Option II is used. Set $0<\eta<\frac{\sqrt{5}-1}{4L}$, $0\le p<1$. Then we have:*

$$\begin{aligned} \frac{1}{T}\sum_{r=0}^{T}\mathbb{E}\,\|\nabla f(\mathbf{x}_0^r)\|^2 &\le \frac{C_2}{T}D_0+C_4\epsilon_1+\eta^2(1-p)(N-1)C_5\\ &\times(1-C_3^{\frac{1}{(1-p)}})^2p^2\frac{(1+L\eta)^2(2L\eta+p(3+L\eta))^2}{N(1-2L\eta-p(1+L\eta))^2}(G^2+\epsilon_1). \end{aligned} \tag{5.12}$$

Here $C_2,C_4,C_5>0$ are constants independent of T,G,p; $C_3:=\frac{p(1+L\eta)+L\eta}{1-L\eta}\ge 0$.

Table 5.3: The relation between p and $\frac{\epsilon}{G^2}$ with fixed $\eta=\frac{\sqrt{5}-1}{8L}$.

Range of p	C_3	$C(p)$	p as function of $\frac{\epsilon}{G^2}$	Relation
$[0,\frac{1-2L\eta}{1+L\eta})$	<1	$\approx 12\eta^2p^2$	$\sqrt{\frac{1}{36\eta^2}\frac{\epsilon}{G^2}}$	**Linear**
$[\frac{1-2L\eta}{1+L\eta},1)$	≥ 1	$\approx 14\eta^2C_3^{2/(1-p)}$	$1-2/\log(\frac{1}{42\eta^2}\frac{\epsilon}{G^2})$	**Log**

Remark 9. **(Communication reduction).** Note that since $0\le p<1$, the total communication rounds is given by $T(1-p)$. To achieve the ϵ accuracy, we need to chose $T=C_2D_0/\epsilon$, $\epsilon_1=\epsilon/C_4$, and need to chose p appropriately so that the last term in (5.12) is also smaller than

ϵ. This implies that $T = C_2 D_0/\epsilon$ and the following shall hold

$$C(p) \triangleq \eta^2(1-p)(N-1)C_5(1-C_3^{1/(1-p)})^2 p^2 \times \frac{(1+L\eta)^2(2L\eta+p(3+L\eta))^2}{N(1-2L\eta-p(1+L\eta))^2} \leq \frac{\epsilon}{3G^2}.$$

The above relation implies that p and $\frac{\epsilon}{G^2}$ should be related by $(1-p)T = \mathcal{O}(\frac{G-\sqrt{\epsilon}}{G\epsilon})$ when $G^2 \in (\mathcal{O}(\epsilon), \infty)$; further, $p \to 1$ at a log-rate when $G^2 \to 0$, that is, when $G = \mathcal{O}\left(\frac{\sqrt{\epsilon}}{\exp(\frac{2}{1-\epsilon})}\right)$, $1-p = \mathcal{O}(\frac{1}{T})$ and it requires $\mathcal{O}(1)$ total communication; see Table 5.3 for details. These results characterize the relation between communication saving and the homogeneity of the local problems. ■

5.5.3 Connection with Other Algorithms

Before we close this section, we discuss the relation of FedPD with a few existing algorithms. In FedProx [38] the agents optimize the following local objective: $f_i(\mathbf{x}_i) + \frac{\rho}{2}\|\mathbf{x}_i - \mathbf{x}_0^r\|^2$. FedProx algorithm fails to converge to the global stationary solution. In contrast, FedPD introduces extra local dual variables $\{\lambda_i\}$ that record the gap between the local model $\mathbf{x}_i$ and the global model $\mathbf{x}_0$ which help the global convergence. FedDANE [114] also proposes a way of designing the subproblem by using the global gradient, but this violates the CTA protocol. Compared with these two algorithms, the proposed FedPD has weaker assumptions, and it achieves better sample and/or communication complexity. In SCAFFOLD [17], the clients perform the following update:

$$\mathbf{x}_i^{r,q+1} = \mathbf{x}_i^{r,q} - \eta(g_i^{r,q} - c_i^r + c^r),$$
$$c_i^{r+1} = c_i^r - c + \frac{1}{K\eta}(\mathbf{x}_0^r - \mathbf{x}_i^{r,Q}),$$

and the server performs the following step:

$$\mathbf{x}_0^{r+1} = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i^{r,Q}, \quad c^{r+1} = \frac{1}{N}\sum_{i=1}^N c_i^{r+1}.$$

Compared to the update with FedPD, we can observe that $c - c_i$'s play the same role as the dual variables λ_i's in FedPD. But SCAFFOLD requires the clients to send the c_i's to the server which breaks the CTA protocol and doubles the communicated information. However, our results showed that even without the extra communication, FedPD can achieve the same convergence rate by adopting a more sophisticated local update direction. FedSplit [106] also keeps a local variable for local update and the algorithm is based on the Peaceman-Rachford splitting method, while FedPD is based on ADMM which can be related to the Douglas–Rachford

splitting method. However, FedSplit only deals with convex problems while FedPD works in the non-convex case. Finally, FedDyn [81] is a recently developed algorithm that mainly deals with partial user participation. The algorithm turns out to be closely related to our proposed FedPD. For detailed discussion about the connection of these two algorithms, please see Appendix C.8.

5.6 Numerical Results

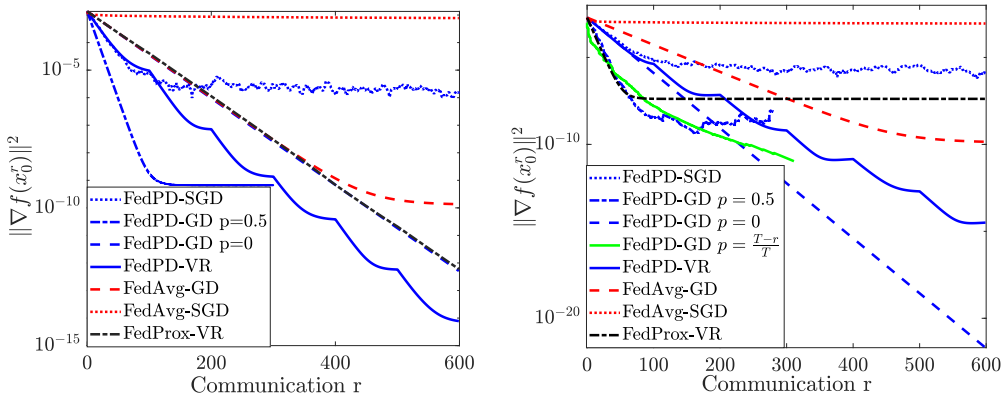
In the first experiment, we show the convergence of the proposed algorithms on synthetic data with FedAvg and FedProx as baselines. We use the non-convex penalized logistic regression [86] as the loss function. The loss function evaluated on a single sample $(\mathbf{a}, b) = \xi$ is given by:

$$F(\mathbf{x}; (\mathbf{a}, b)) = \log(1 + \exp(-b\mathbf{x}^T\mathbf{a})) + \sum_{d=1}^{D} \frac{\beta\alpha(\mathbf{x}[d])^2}{1 + \alpha(\mathbf{x}[d])^2}. \tag{5.13}$$

Here $\mathbf{x}[d]$ denotes the d^{th} component of $\mathbf{x}$. The feature vector and model parameter $\mathbf{a}, \mathbf{x} \in \mathbb{R}^D$ have dimension D and $b \in \{-1, 1\}$ is the label corresponding to the feature. During the simulation, we set the constants to be $\alpha = 1$ and $\beta = 0.1$.

In the experiment, we use two ways to generate the data. In the first case (referred to as the "weakly non-i.i.d" case), the features and the labels on the agents are randomly generated, so the local data sets are not very non-i.i.d. In the second case (referred to as the "strong non-i.i.d." case), we first generate the feature vector $\mathbf{a}$'s following the standard Normal distribution, then we generate the local model $\mathbf{x}_i$ on the i^{th} agent by using uniform distribution in the range of $[-10, 10]$ for each component. Then we compute the label b's according to the local models and the features, and then add noise following the standard normal distribution. In this case, the agents' data distribution is more non-i.i.d compared to the first case. In both cases, there are 400 samples on each agent with total 100 agents.

We run FedPD with Oracle I (FedPD-SGD and FedPD-GD) and Oracle II (FedPD-VR). For FedPD-SGD, we set $Q = 600$, and for FedPD-GD and FedPD-VR we set $Q = 8$. For FedPD-GD we set $p = 0$ and $p = 0.5$, where in the latter case, the agents skip half of the communication rounds. For FedPD-VR, we set mini-batch size $B = 1$ and gradient computation frequency $I = 20$. For comparison, we also run FedAvg with local GD/SGD and FedProx. For FedAvg with GD, $Q = 8$, and for FedAvg with SGD, $Q = 600$. For FedProx, we solve the local problem using variance reduction for $Q = 8$ iterations. The total number of iterations T is set as 600 for all algorithms.



(a) Stationary gap of FedAvg, FedProx and FedPD; weakly non-i.i.d. data.

(b) Stationary gap of FedAvg, FedProx and FedPD; strongly non-i.i.d. data.

Figure 5.2: The convergence result of the algorithms on penalized logistic regression with weakly and strongly non-i.i.d. data with respect to the number of communication rounds.

Figure 5.2 shows the results with respect to the number of communication rounds. In Fig. 5.2(a), we compare the convergence of the tested algorithms on weakly non-i.i.d. data set. It is clear that FedProx and FedPD with $p = 0$ (i.e., no communication skipping) are comparable. Meanwhile, FedAvg with local GD will not converge to the stationary point with a constant stepsize when local update step $Q > 1$. By skipping half of the communication, FedPD with local GD can still achieve a similar error as FedAvg, but using fewer communication rounds. In Fig. 5.2(b), we compare the convergence results of different algorithms with the strongly non-i.i.d. data set. We can see that the algorithms using stochastic solvers become less stable compared with the case when the data sets are weakly non-i.i.d. Further, FedPD-VR and FedPD-GD with $p = 0$ are still able to converge to the global stationary point while FedProx will achieve a similar error as the FedAvg with local GD.

We have included more details on the experimental results and additional experiments in Appendix C.7.

Chapter 6

Understanding Clipping in Privatized Federated Learning

6.1 Motivation

First proposed by [12], *Federated Learning* (FL) is a distributed learning framework that aims to reduce communication complexity and to provide privacy protection during training. The popular FedAvg algorithm [12] has been proposed to reduce the communication cost by using periodic averaging and client sampling. There has been many extensions of this algorithm, mostly by modifying the local update directions [17, 18, 108]. Even though FL algorithms have the goal of privacy protection, recent works have shown that they are vulnerable to inference attacks and leak local information during training [43, 44, 45]. As a result, striking a balance between *formal* privacy guarantees and desirable optimization performance remains one of the fundamental challenges in FL.

Recently, various FL algorithms [46, 47, 48, 49, 50] have been proposed to provide the formal guarantees of *differential privacy* (DP) [51]. In these algorithms, the clients perform multiple local updates between two communication steps, and then perturbation mechanisms are added to aggregate updates across individual clients. In order for the perturbation mechanism to have formal privacy guarantees, each client's model update needs to have a bounded norm, which is ensured by applying a clipping operation that shrinks individual model updates when their norm exceeds a given threshold. While there has been prior work that studies the clipping effects on stochastic gradients [52, 53, 54] in the differentially private SGD [55], there has not been any work on providing understanding how clipping the model updates affect the optimization performance of FL subject to DP. Our work provides the first in-depth study on such clipping effects.

6.2 Preliminaries

Federated learning typically considers the following optimization problem:

$$\min_{\mathbf{x}} \left[f(\mathbf{x}) \triangleq \sum_{i=1}^{N} f_i(\mathbf{x}) \right], \text{ where } f_i(\mathbf{x}) = \mathbb{E}_{\xi \sim \mathcal{D}_i} F(\mathbf{x}; \xi), \tag{6.1}$$

where N is the number of participating clients; the i^{th} client optimizes a local model f_i, which is the expectation of a loss function $F(\mathbf{x}; \xi)$, where the expectation is taken over local data distribution $\mathcal{D}_i$. At each communication round t, the server samples a subset of clients $\mathcal{P}_t$ and broadcasts the global model parameters $\mathbf{x}^t$. The sampled clients perform Q steps of SGD updates and compute the total update differences $\Delta \mathbf{x}_i^t$'s, and then the server aggregates the update differences to update the global model. In Algorithm 6, we present a slightly generalized FedAvg algorithm [17, 115], in which the server uses a stepsize η_g to perform its update. When $\eta_g = 1$, the algorithm becomes the same as the original FedAvg.

Algorithm 6 FedAvg Algorithm

Initialize: $\mathbf{x}_i^0 \triangleq \mathbf{x}^0, i = 1, \ldots, N$
for $t = 0, \ldots, T-1$ *(stage)* **do**
  for $i \in \mathcal{P}_t \subseteq [N]$ in parallel **do**
    Update agents' $\mathbf{x}_i^{t,0} = \mathbf{x}^t$
    for $q = 0, \ldots, Q-1$ *(iteration)* **do**
      Compute stochastic gradient $g_i^{t,q}$ with $\mathbb{E}[g_i^{t,q}] = \nabla f_i(x_i^{t,q})$
      Local update: $\mathbf{x}_i^{t,q+1} = \mathbf{x}_i^{t,q} - \eta_l g_i^{t,q}$
    end for
  end for
  Global averaging: $\Delta \mathbf{x}_i^t = \mathbf{x}_i^{t,Q} - \mathbf{x}^t, \quad \mathbf{x}^{t+1} = \mathbf{x}^t + \eta_g \frac{1}{|\mathcal{P}_t|} \sum_{i \in \mathcal{P}_t} \Delta \mathbf{x}_i^t$
end for

In this work, we study FL subject to the rigorous privacy guarantees of *Differential Privacy* (DP) [51], whose formal definition is given below.

Definition 3 *[51] An algorithm $\mathcal{M}$ is (ϵ, δ)-differentially private if*

$$P(\mathcal{M}(\mathcal{D}) \in \mathcal{S}) \leq e^{\epsilon} P(\mathcal{M}(\mathcal{D}') \in \mathcal{S}) + \delta, \tag{6.2}$$

where $\mathcal{D}$ and $\mathcal{D}'$ are neighboring datasets, $\mathcal{S}$ is an arbitrary subset of outputs of $\mathcal{M}$.

The common mechanism used to protect DP in centralized training is straightforward: 1) clip the stochastic gradient with the so-called clipping operation (6.3); 2) add a random perturbation

$\mathbf{z} \sim \mathcal{N}(0, \sigma^2 I)$ to the clipped quantity [55]. The clipping operation is the key step to guarantee DP as the noise level σ^2 is determined by the clipping threshold c [116]:

$$\text{clip}(g^t, c) = g^t \cdot \min\left\{1, \frac{c}{\|g^t\|}\right\}. \tag{6.3}$$

However, DP is more complex in FL than that in centralized training. Two key factors distinguish FL from existing DP machine learning framework are:

- *Data distribution*: unlike centralized training, in FL the data are naturally distributed on the clients, and the clients can potentially have very different data distributions. In the centralized setting, the recent work [53] has shown that the distribution of the samples affects the performance of the DP-SGD, but how heterogeneous data distribution affects the design and analysis of FL algorithm that protects DP is unclear.
- *Local updates*: as described in Algorithm 6, the clients will perform multiple local update steps before sending the model to the server, and it is well-known that when $Q > 1$, the data heterogeneity will cause performance degradation in FedAvg even without clipping and perturbation [14]. Although there are multiple alternatives of how the DP mechanism can be applied to FL algorithms, none of those mechanisms has a rigorous theoretical guarantee, and it is not clear how to properly balance the optimization performance and privacy guarantees.

These two factors result in different *definitions* and *clipping operations* in FL.

DP definitions in FL: Based on the distribution pattern of the client and local datasets, two DP definitions correspond to the neighboring datasets in Definition 3, are commonly considered in FL algorithm design:

- Sample-level differential privacy (SL-DP): SL-DP directly follows the centralized DP and protects each local sample so that the server could not identify one sample from the union of all local datasets, i.e., $\mathcal{D} = \bigcup_{i=1}^{N} \mathcal{D}_i$, and $\mathcal{D}, \mathcal{D}'$ differ by one sample ξ. SL-DP fits in the cross-silo FL scenario that has a relatively small number of clients, each with a large dataset. E.g., SL-DP is used in medical image classification application to protect patients' personal information [117]. However, in the Google Keyboard application [118] where each client is an application user, SL-DP that only protects one sample (i.e., an input record) will not be sufficient to protect the user's personal information.
- Client-level differential privacy (CL-DP): CL-DP has a stricter privacy guarantee compared with SL-DP. It requires that the server cannot identify the participation of one client by observing the output of the local updates, i.e., $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^{N}$, and $\mathcal{D}, \mathcal{D}'$ differ by one dataset $\mathcal{D}_i$. CL-DP is suitable for the cross-device FL scenario such as the Google Keyboard application, which has a large number of distributed clients.

Clipping operation in FL: Based on different DP requirements and the algorithm structures, a number of FL algorithms have been proposed which protect DP to some extent.

To protect SL-DP, [48] proposes to clip and inject noise to every local update. That is, some Gaussian noise is added to the stochastic gradients $g_i^{t,q}$ given in Algorithm 6. However, as intermediate updates are kept local and private, the clipping and perturbation to the local steps appear to be unnecessary, and such operations result in significant performance degradation. Moreover, it is not clear how such kind of operation impact other aspects of the algorithm performance (such as algorithm convergence, quality of solutions, etc.)

To protect CL-DP, [119] proposes to clip the local models to be transmitted directly. Similarly, [47] assumes that the model parameters are upper and lower bounded by some constant and directly apply perturbations to the local models. However, this scheme also significantly reduces the training and test accuracy empirically and has no theoretical convergence guarantee. Recently, [46] proposes to clip the difference between the input model and the output models of the FedAvg algorithm. In particular, one can replace the update directions $\Delta \mathbf{x}_i^t$'s of line 8 in Algorithm 1 by their clipped versions as expressed below:

$$\begin{aligned} \text{clip}(\Delta \mathbf{x}_i^t, c) &= \Delta \mathbf{x}_i^t \cdot \min\left\{1, \frac{c}{\|\Delta \mathbf{x}_i^t\|}\right\}, \\ \mathbf{x}^{t+1} &= \mathbf{x}^t + \eta_g \frac{1}{|\mathcal{P}_t|} \sum_{i \in \mathcal{P}_t} \text{clip}(\Delta \mathbf{x}_i^t, c). \end{aligned} \tag{6.4}$$

It is shown that such a scheme has better numerical performance than model clipping, but no convergence proof for the algorithm is given. Reference [50] also clips the update difference and proposed Bayesian DP to measure the privacy loss and only demonstrates the numerical performance of the proposed algorithm. D2P-Fed [49] follows the same clipping strategy and further apply compression and quantization during communication to improve communication efficiency while having DP guarantee, but its convergence guarantee only applies to the non-clipping version.

In summary, despite extensive recent research about DP-enabled FL, there are still a number of technical challenges and open research questions in this area. First, it is not clear how various kinds of clipping operations can affect the performance of FL algorithms. Second, it is not clear how to add noise to balance the convergence of FL algorithms and its CL-DP guarantee.

6.3 Clipping Issues in FL

As discussed above, clipping is a key operation in providing DP guarantee for FL algorithms. Therefore, to design algorithms that protect DP in FL, the first step is to understand how clipping affects the convergence performance of a FL algorithm. Towards this end, we start

with analyzing two common clipping strategies, and identify their theoretical properties. Then we provide a series of empirical studies to demonstrate how system parameters such as training models, datasets and data distributions can affect the performance of clipping-enabled FedAvg algorithm. These empirical studies will be combined with our theoretical analysis in the next section to provide a comprehensive understanding about the optimization performance and CL-DP guarantees in FL.

6.3.1 Model clipping versus Difference Clipping

The two major clipping strategies used in protecting CL-DP for FL algorithms are *local model* clipping and *local update difference* clipping, as we describe below.

1. **Model clipping** [119]: The clients directly clip the models sent to the server. For FedAvg algorithm, this means performing $\text{clip}(x_i^{t,Q}, c)$. This method appears to be straightforward, but clipping the model directly results in relatively large clipping threshold, so it requires to add larger perturbation.

2. **Difference clipping** [46]: The clients clip the local update difference between the initial model and the output model according to (6.4). This method needs to record the initial model and to perform extra computation before clipping, but the update difference typically has smaller magnitudes than the model itself, so the clipping threshold and the perturbation can be smaller than using model clipping. Note that when $Q = 1$, the difference clipping is equivalent to the standard mini-batch gradient clipping (i.e., the DP-SGD), but in the general case where $Q > 1$, their behaviors are very different.

Below we analyze how they perform on simple quadratic problems. Our results indicate that the difference clipping strategy is more preferable, because it is less likely to have strong impact on the optimization performance. The full proofs of the claims are given in Appendix D.3.

Claim 3 *Given any constant clipping threshold c, there exists a convex quadratic problem, for which FedAvg with model clipping does not converge to the global optimal solution with any fixed $Q \geq 1$ and $\eta_l > 0$.*

Claim 4 *For all linear regression problem with fixed clipping threshold c, there exist η_l and local update step $Q \geq 1$ such that FedAvg with difference clipping converges to the global optimal solution. Furthermore, there exist a linear regression problem such that under the same c, η_l and Q, FedAvg with difference clipping converges to a better solution with smaller loss than the original FedAvg.*

Remark 10. To prove Claim 3, we construct a problem whose magnitude of the optimal solution is larger than the clipping threshold. Then FedAvg with model clipping will converge to a stationary point with magnitude bounded by the clipping threshold, therefore the algorithm will not converge to global optimal solution.

The technique to prove the first part of Claim 4 is related to the analysis for centralized gradient clipping algorithms [120]. The main difference is that our algorithm consider Q steps of local update before clipping. We show that by allowing multiple local updates, FedAvg algorithm with difference clipping optimizes the sum of the Huberzied re-weighted local loss functions. By properly choosing the learning rate η_l for each local loss function, we can balance the re-weighting factors so that the optimal solution to the new loss function matches the solution to the original problem. ■

The above claims indicate that the difference clipping should outperform the model clipping in terms of convergence guarantees. Therefore, in the subsequent analysis, we will focus on understanding the difference clipping enabled FL algorithms. In particular, we consider the Clipping-Enabled FedAvg (CE-FedAvg) algorithm described in Algorithm 7, which combines the difference clipping with the slightly generalized FedAvg algorithm described in Algorithm 6 (which uses two stepsizes η_l, η_g, one for local and one for global updates, respectively). The reason to consider such a *bi-level-stepsize* version of FedAvg is that, it has been proved to have superior performance, especially when not all clients participate in each round of communication [17, 115].

Algorithm 7 Clipping-enabled FedAvg (CE-FedAvg)

Initialize: $\mathbf{x}_i^0 \triangleq \mathbf{x}^0, i = 1, \ldots, N$
for $t = 0, \ldots, T-1$ *(stage)* **do**
  for $i \in \mathcal{P}_t \subseteq [N]$ in parallel **do**
    Update agents' $\mathbf{x}_i^{t,0} = \mathbf{x}^t$
    for $q = 0, \ldots, Q-1$ *(iteration)* **do**
      Compute stochastic gradient $g_i^{t,q}$ with $\mathbb{E}[g_i^{t,q}] = \nabla f_i(x_i^{t,q})$
      Local update: $\mathbf{x}_i^{t,q+1} = \mathbf{x}_i^{t,q} - \eta_l g_i^{t,q}$
    end for
    Compute update difference: $\Delta \mathbf{x}_i^t = \mathbf{x}_i^{t,Q} - \mathbf{x}_i^{t,0}$
    Clip: $\hat{\Delta}\mathbf{x}_i^t = \text{clip}(\Delta \mathbf{x}_i^t, c)$, where $\text{clip}(\cdot)$ is defined in (6.3)
  end for
  Global averaging: $\mathbf{x}^{t+1} = \mathbf{x}^t + \eta_g \frac{1}{|\mathcal{P}_t|} \sum_{i \in \mathcal{P}_t} \hat{\Delta}\mathbf{x}_i^t$
end for

6.3.2 Empirical Results

Experiment Setting. To have a thorough understanding about how the difference clipping can impact the FedAvg, we conduct numerical experiments with different models, datasets and local data distributions. We compare the test accuracies between CE-FedAvg and the original FedAvg. Note that in this set of experiments we do not consider the privacy issues yet, so we do not add perturbation.

To have a fair comparison, we set Q, T, N, $|\mathcal{P}_t|$, η_l and η_g to be identical for both FedAvg and CE-FedAvg. We first run the original FedAvg, compute $\|\Delta \mathbf{x}_i^t\|$ and average over all clients i and iterations t to obtain $\bar{\Delta}$ and choose the clipping threshold $c = 0.5\bar{\Delta}$.

We run the algorithm using AlexNet [121] and ResNet-18 [122] with EMNIST dataset [123] and Cifar-10 dataset [124] for comparison. We split the dataset in two different ways: 1) *IID Data* setting, where the samples are uniformly distributed to each client; 2) *Non-IID Data* setting, where the clients have unbalanced samples. Details are described below. For EMNIST digit classification dataset, each client has 500 samples without overlapping. In the IID case, each client has around 50 samples of each class and in the Non-IID case, there are 8 classes each has around 5 samples and 2 classes each has 230 samples on each client. For the Cifar-10 dataset, in the IID case (resp. Non-IID case), each client also has 500 samples (resp. 50 samples); these samples can overlap with those on the other clients and the samples on each client are uniformly distributed in 10 classes, i.e., each client has 50 samples (resp. 5 samples) from each class.

Performance Degradation. In Table 6.1, we compare the classification results produced by using AlexNet and ResNet-18 on the two datasets.

Model	dataset	IID(%)	- Clipping (% drop)	Non-IID (%)	- Clipping (% drop)
AlexNet	EMNIST	98.20	0.19	95.60	3.60
	Cifar-10	66.01	4.83	57.14	7.30
ResNet-18	EMNIST	99.61	0.02	95.43	0.10
	Cifar-10	76.36	0.53	59.46	1.55

Table 6.1: The accuracy drop between a) FedAvg and clipping-enabled FedAvg, used for training AlexNet and ResNet-18, on IID and Non-IID data.

There are three interesting observations: 1) The data distribution will greatly affect the clipping performance in FL. When data are IID across the clients, clipping has far less impact on the final accuracy, otherwise the clipping will introduce some accuracy drop to the trained models; 2) Clipping has quite different impact on different models – the best accuracy of the models drops 0.10% and 3.60% for ResNet-18 and AlexNet on EMNIST, respetively. The drop is 1.55% for ResNet-18 and 7.30% for AlexNet on Cifar-10, comparing CE-FedAvg with non-clipped

version on the Non-IID data; 3) Data complexity also affects the behavior of the CE-FedAvg – the accuracy drop on Cifar-10 dataset is much larger than that on EMNIST dataset.

The empirical experiments show that heterogeneous data distribution among the clients is one of the main causes of the different behavior between the clipped and non-clipped algorithms. The data heterogeneity issue is unique in FL cause by periodical communication. It does not happen in centralized optimization where the data are shared among all workers.

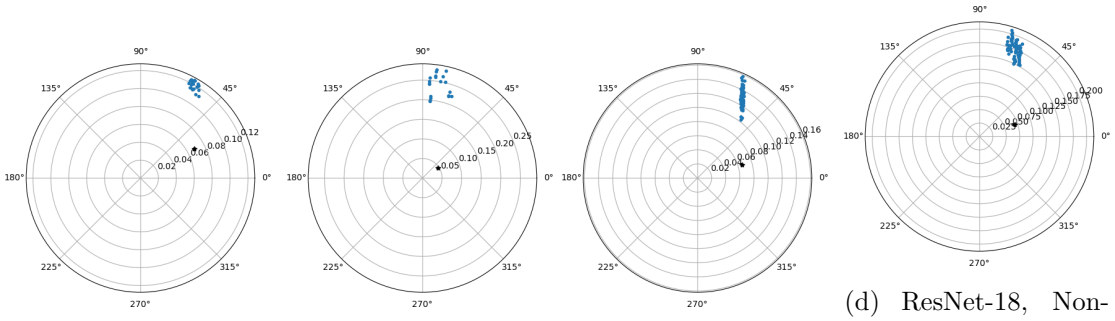
Update Difference Distribution. To further understand the clipping procedure, we plot in Fig. 6.1 and Fig. 6.2 the magnitudes of local updates $\|\Delta \mathbf{x}_i^t\|$ and the cosine angles between the last iteration's global update and $\Delta \mathbf{x}_i^t$:

$$\cos^{-1}\left(\frac{\left\langle \Delta \mathbf{x}_i^t, \frac{1}{|\mathcal{P}_t|}\sum_{i\in\mathcal{P}_{t-1}} \Delta \mathbf{x}_i^{t-1}\right\rangle}{\|\Delta \mathbf{x}_i^t\|\left\|\frac{1}{|\mathcal{P}_t|}\sum_{i\in\mathcal{P}_{t-1}} \Delta \mathbf{x}_i^{t-1}\right\|}\right).$$

Due to page limitation, we only put the distribution of communication round $T = 16$. More detailed results are given in Appendix D.2. In the plots, we mainly focus on the variance of the magnitudes of the clients' update difference (i.e., the blue dots). Larger variance indicates that the updates made by different clients are more different from each other.

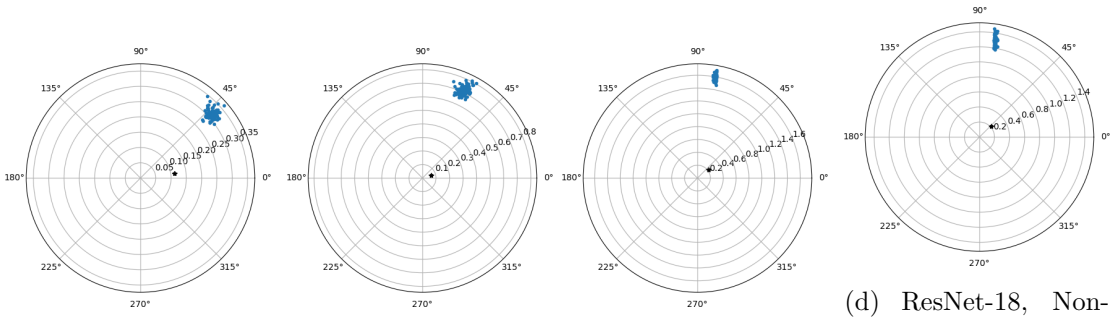
Comparing Fig. 6.1 with Fig. 6.2 we can see that the update magnitudes on EMNIST dataset are more concentrated than that on Cifar-10 dataset by having smaller mean and variance. Similarly, by comparing Fig. 6.1a with Fig. 6.1b or Fig. 6.1c with Fig. 6.1d, it is clear that the local update magnitudes are more concentrated on IID data than on Non-IID data. Moreover, ResNet-18 has a more concentrated distribution of update magnitudes than AlexNet. Importantly, comparing Table 6.1 with Fig. 6.1 and Fig. 6.2, one can observe that the drop in final accuracy of a model caused by clipping is correlated with *the degree of concentration* of update magnitudes, as AlexNet with less concentrated update magnitudes suffers more from clipping, while ResNet-18 exhibits the opposite behavior.

The above results about the update difference distributions match the accuracy results in Table 6.1, in the sense that clipping performs worse when update differences distribution has a larger divergence and vise versa. Inspired by this observation, in the next subsection, we will characterize the impact of clipping based on the degree of concentration in local updates and develop the convergence analysis of CE-FedAvg.



(a) AlexNet, IID (b) AlexNet, Non-IID (c) ResNet-18, IID (d) ResNet-18, Non-IID

Figure 6.1: The distribution of local updates for AlexNet and ResNet-18 on IID and Non-IID data at communication round 16 for EMNIST dataset. Each blue dot corresponds to the local update from one client. The black dot shows the magnitude and the cosine angle of averaged local update at iteration t.



(a) AlexNet, IID (b) AlexNet, Non-IID (c) ResNet-18, IID (d) ResNet-18, Non-IID

Figure 6.2: The distribution of local updates for AlexNet and ResNet-18 on IID and Non-IID data at communication round 16 for Cifar-10 dataset. Each blue dot corresponds to the local update from one client. The black dot shows the magnitude and the cosine angle of averaged local update at iteration t.

6.4 Convergence Analysis

In this section, we analyze the theoretical performance of CE-FedAvg as well as its randomly perturbed version, in order to gain a better understanding of our previous empirical observations and the trade-off between the convergence performance of FedAvg and its DP guarantees.

Towards this end, we will provide the convergence analysis and privacy guarantees for the DP-FedAvg algorithm, described in Algorithm 8. Compared to CE-FedAvg, this algorithm further adds a random perturbation $\mathbf{z}_i^t$ to the locally clipped model differences. During the communication, we assume that the attacker can only observe the aggregated update $\sum_{i\in\mathcal{P}_t} \tilde{\Delta}\mathbf{x}_i^t$,

and this can be guaranteed by using secure aggregation [125] or assuming the uplink of the clients to the server is secure.

Despite the similar mechanism used in DPSGD and DP-FedAvg, let us point their major differences: in DPSGD, the goal is to protect SL-DP, while DP-FedAvg is to protect CL-DP. The key difference in DP-FedAvg is that the local dataset size is large enough so that after performing multiple local update steps, the resulting model has relatively good performance. By doing so, we can largely reduce the number of communications and the corresponding privacy noise added per communication. Note that DP-FedAvg becomes DPSGD with the following choices of hyperparameters: 1) enlarge the client number to be the same as the size of the dataset, 2) decrease the local dataset size to 1; 3) decrease the number of local updates to 1; 4) decrease the privacy noise accordingly.

6.4.1 Convergence Analysis

Theorem 8 (Convergence of DP-FedAvg) *For Algorithm 8, assume*

$$\begin{aligned}
&\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x-y\|,\ \forall\, i, x, y,\ \min_x f(x) \geq f^*;\\
&\mathbb{E}[\|g_i^{t,q} - \nabla f_i(x_i^{t,q})\|^2] \leq \sigma_l^2, \quad \|g_i^{t,q}\| \leq G,\ \forall\, t, q, i,\\
&\|\nabla f_i(x) - \nabla f(x)\|^2 \leq \sigma_g^2,\ \forall i,
\end{aligned}$$

where L is the Lipschitz constant of gradient, σ_l^2 and σ_g^2 are intra-client and inter-client gradient variance, G is the bound on stochastic gradient.

By letting $\eta_g\eta_l \leq \min\{\frac{P}{48Q}, \frac{P}{6QL(P-1)}\}$ and $\eta_l \leq \frac{1}{\sqrt{60QL}}$, we have

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\overline{\alpha}^t\|\nabla f(x^t)\|^2]\\
&\leq \underbrace{\frac{4(f(x^0)-f^*)}{\eta_g\eta_l QT} + \frac{25}{2}\eta_l^2 LQ(\sigma_l^2 + 6Q\sigma_g^2)\gamma_1(T) + \frac{6\eta_g\eta_l L\sigma_l^2}{P}\gamma_2(T)}_{\text{standard terms for FedAvg}}\\
&+ \underbrace{\frac{2\eta_g L d\sigma^2}{\eta_l PQ}}_{\text{caused by privacy noise}} + \underbrace{G^2\frac{4}{T}\sum_{t=1}^{T}\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}(|\alpha_i^t - \tilde{\alpha}_i^t| + |\tilde{\alpha}_i^t - \overline{\alpha}^t|)\right]}_{\text{caused by clipping}}\\
&+ \underbrace{\eta_g\eta_l LQG^2\frac{6}{T}\sum_{t=1}^{T}\mathbb{E}\left[\frac{1}{P}\sum_{i=1}^{N}(|\alpha_i^t - \tilde{\alpha}_i^t|^2 + |\tilde{\alpha}_i^t - \overline{\alpha}^t|^2)\right]}_{\text{caused by clipping}}
\end{aligned}$$

where $P := |\mathcal{P}_t|$, $\alpha_i^t := \frac{c}{\max(c, \eta_l\|\sum_{q=0}^{Q-1} g_i^{t,q}\|)}$, $\tilde{\alpha}_i^t := \frac{c}{\max(c, \eta_l\|\mathbb{E}[\sum_{q=0}^{Q-1} g_i^{t,q}]\|)}$, $\overline{\alpha}^t := \frac{1}{N}\sum_{i=1}^{N}\tilde{\alpha}_i^t$; d is the dimension of x, $\gamma_1(T) = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\overline{\alpha}^t] \leq 1$, $\gamma_2(T) = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[(\overline{\alpha}^t)^2] \leq 1$.

Algorithm 8 DP-FedAvg Algorithm

Initialize: $\mathbf{x}_i^0 \triangleq \mathbf{x}^0, i = 1, \ldots, N$
for $t = 0, \ldots, T-1$ *(stage)* **do**
  for $i \in \mathcal{P}_t \subseteq [N]$ in parallel **do**
    Update agents' $\mathbf{x}_i^{t,0} = \mathbf{x}^t$
    for $q = 0, \ldots, Q-1$ *(iteration)* **do**
      Compute stochastic gradient $g_i^{t,q}$ with $\mathbb{E}[g_i^{t,q}] = \nabla f_i(x_i^{t,q})$
      Local update: $\mathbf{x}_i^{t,q+1} = \mathbf{x}_i^{t,q} - \eta_l g_i^{t,q}$
    end for
    Compute update difference: $\Delta \mathbf{x}_i^t = \mathbf{x}_i^{t,Q} - \mathbf{x}_i^{t,0}$
    Clip and perturb: $\tilde{\Delta}\mathbf{x}_i^t = \text{clip}(\Delta \mathbf{x}_i^t, c) + \mathbf{z}_i^t$, where $\text{clip}(\cdot)$ is defined in (6.3)
  end for
  Global averaging: $\mathbf{x}^{t+1} = \mathbf{x}^t + \eta_g \frac{1}{|\mathcal{P}_t|} \sum_{i \in \mathcal{P}_t} \tilde{\Delta}\mathbf{x}_i^t$
end for

In the bound of Theorem 8, the standard terms are inherited from standard FedAvg with two-sided learning rates which can yield a convergence rate of $O(\frac{1}{\sqrt{PQT}} + \frac{1}{T})$ when setting $\eta_g = \sqrt{QP}$ and $\eta_l = \frac{1}{\sqrt{TQL}}$. When there is no clipping bias and privacy noise, Theorem 8 exactly recovers the standard convergence bounds for FedAvg up to a constant, see Theorem 1 in [115]. In addition to the standard terms, we have extra terms caused by the privacy noise z_i^t and the clipping operation. We highlight the terms caused by clipping which characterize the estimation bias caused by clipping. The bias can be decomposed into terms caused by $|\alpha_i^t - \tilde{\alpha}_i^t|$ and terms caused by $|\tilde{\alpha}_i^t - \overline{\alpha}^t|$. Notice that $|\alpha_i^t - \tilde{\alpha}_i^t| \leq \eta_l \| \| \sum_{q=0}^{Q-1} g_i^{t,q} \| - \| \mathbb{E}[\sum_{q=0}^{Q-1} g_i^{t,q}] \| \|$, it is clear $\mathbb{E}[|\alpha_i^t - \tilde{\alpha}_i^t|]$ will be small if the stochastic local updates have similar variance or magnitudes in norm, and $\mathbb{E}[|\alpha_i^t - \tilde{\alpha}_i^t|] = 0$ if $\sigma_l = 0$. This term characterizes the bias caused by local update variance. In addition, $\mathbb{E}[|\tilde{\alpha}_i^t - \overline{\alpha}^t|]$ will be small if the expected local model updates have similar magnitudes in norm across clients and $\mathbb{E}[|\tilde{\alpha}_i^t - \overline{\alpha}^t|] = 0$ if $\|\mathbb{E}[\Delta x_i^t]\| = \|\mathbb{E}[\Delta x_j^t]\|, \forall i, j$. This term shows the bias caused by cross-client update variance.

In FL, sometimes each client will have limited amount of data, and the local model updates can be performed with small σ_l or even $\sigma_l = 0$ (full batch update). Thus, the bias caused by $|\alpha_i^t - \tilde{\alpha}_i^t|$ can be small and is avoidable. However, the bias caused by $|\tilde{\alpha}_i^t - \overline{\alpha}^t|$ is *unavoidable* since this term will not diminish even each client updates its local model with full batch gradient. In addition, this term might be large with heterogeneous data distribution since the heterogeneity may induce quite disparate gradient distributions across clients. Thus, it is crucial to investigate the bias caused by $|\tilde{\alpha}_i^t - \overline{\alpha}^t|$ in practice. Note that $|\tilde{\alpha}_i^t - \overline{\alpha}^t|$ is fully controlled by differences

in magnitudes of local model updates when $\sigma_l = 0$ for fixed c. Going back to Fig. 6.1, we do see that how such differences in update magnitudes can be affected by both the neural network models and data heterogeneity.

6.4.2 Differential Privacy Guarantee

The privacy guarantee of DP-FedAvg can be characterized by standard privacy theorems on Gaussian mechanism. We rephrase [55, Theorem 1] for client privacy in Theorem 9.

Theorem 9 (Privacy of DP-FedAvg) *There exist constants u and v so that given the number of iterations T, for any $\epsilon \leq uq^2T$ with $q = \frac{P}{N}$ and $|\mathcal{P}_t| = P$, $\forall t$, Algorithm 6 is (ϵ, δ)-differentially private for any $\delta > 0$ if $\sigma^2 \geq v\frac{c^2PT\ln(\frac{1}{\delta})}{N^2\epsilon^2}$.*

The privacy-utility trade-off of DP-FedAvg can be analyzed by substituting σ^2 from Theorem 9 into Theorem 8. To get more insights on how parameters like T, η_g, η_l and ϵ affect DP-FedAvg, let us consider simplified Theorem 8 in Corollary 3 with $c \geq \eta_l QG$ and σ^2 substituted . If $c' < G$ in Corollary 3, then there will be extra bias terms inherited from the bound in Theorem 8. it can be affected by c' and the distribution of update magnitude of different clients.

Corollary 3 (Convergence with privacy guarantee) *Assume all assumptions in Theorem 8, for any clipping threshold $c = \eta_l Qc'$ with $c' \geq G$, and set σ^2 as in Theorem 9, for any (ϵ, δ) satisfying the constraints in Theorem 9, we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(x^t)\|^2] \leq \underbrace{O\left(\frac{1}{\eta_g\eta_l QT} + \eta_l^2Q^2 + \frac{\eta_g\eta_l}{P}\right)}_{\text{standard terms for FedAvg}} + \underbrace{O\left(\frac{\eta_g\eta_l QTd\ln(\frac{1}{\delta})}{N^2\epsilon^2}\right)}_{\text{caused by privacy noise}} \tag{6.5}$$

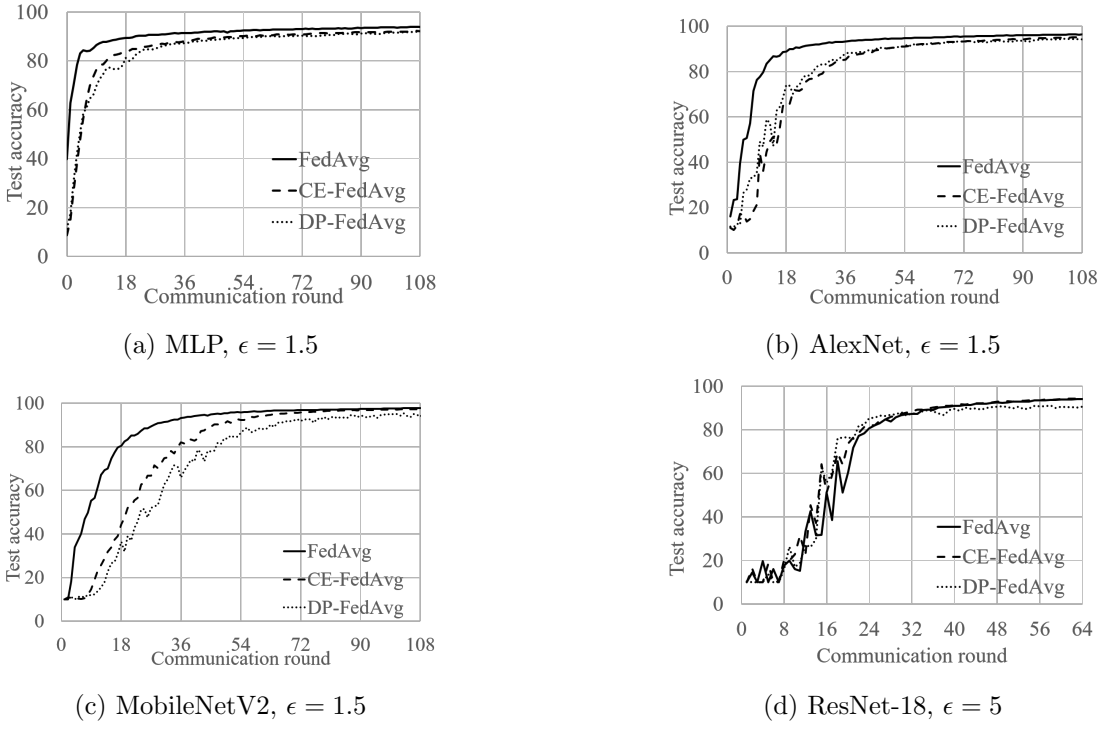
and the best rate one can get from the above bound is $\tilde{O}(\frac{\sqrt{d}}{N\epsilon})$ by optimizing η_g, η_l, Q, T.

A direct implication of Corollary 3 is that the big-O convergence rate of DP-FedAvg is the same as differentially private SGD (DP-SGD) in terms of d, ϵ, and N (note that N which will be the number of training samples in DP-SGD).

6.5 Numerical Results

Model	# Parameters	# Layers	Acc. (%)	Clipping (% drop)	DP (% drop)
MLP	159K	2	94.0	1.84	0.29
AlexNet	3.3M	7	96.4	1.47	0.16
MobileNetV2	2.3M	24	97.8	0.35	1.62
ResNet-18	11.1M	18	95.2	-0.15	3.76*

Table 6.2: The accuracy drop between a) FedAvg and clip-enabled FedAvg and b) clip-enabled FedAvg and DP-FedAvg. The clipping threshold is 0.5 of the average magnitude and privacy budget $\epsilon = 1.5$ for MLP, AlexNet and MobileNetV2 and $\epsilon = 5$ for ResNet-18.



(a) MLP, $\epsilon = 1.5$

(b) AlexNet, $\epsilon = 1.5$

(c) MobileNetV2, $\epsilon = 1.5$

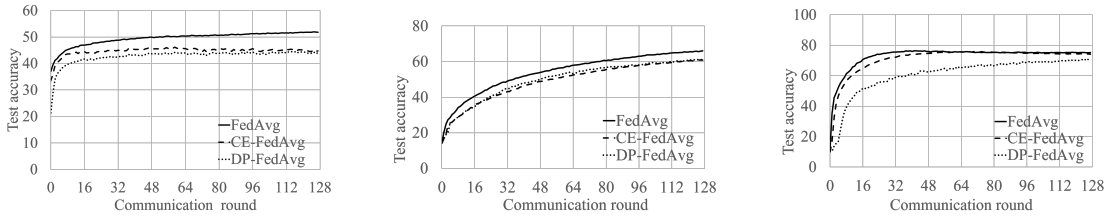
(d) ResNet-18, $\epsilon = 5$

Figure 6.3: The test accuracy of FedAvg, CE-FedAvg and DP-FedAvg on different models on EMNIST. The privacy budgets for MLP, AlexNet and MobileNet are $\epsilon = 1.5$ while for ResNet, we set $\epsilon = 5$.

Cifar-10 dataset. The dataset we use is the Cifar-10 dataset, which has 50K training samples and 10K testing samples. We distribute the data in the IID way described in Section II and each client has 500 samples. We conduct experiments on a 2-layer MLP with one hidden layer, AlexNet and ResNet-18. The results are listed in Table 6.3 and Figure 6.4.

Model	# Parameters	# Layers	Accuracy (%)	Clipping (% drop)	DP (% drop)
MLP	616K	2	51.90	7.39	0.90
AlexNet	3.3M	7	66.01	4.83	-0.18
ResNet-18	11.1M	18	76.36	0.53	5.15

Table 6.3: The accuracy drop between a) FedAvg and CE-FedAvg and b) CE-FedAvg and DP-FedAvg. The clipping threshold is 0.5 of the average magnitude and privacy budget $\epsilon = 1.5$ for MLP, AlexNet and ResNet-18.



(a) MLP, $\epsilon = 1.5$ (b) AlexNet, $\epsilon = 1.5$ (c) ResNet-18, $\epsilon = 1.5$

Figure 6.4: The test accuracy of FedAvg, CE-FedAvg and DP-FedAvg on different models on Cifar-10. The privacy budgets for MLP, AlexNet and ResNet are $\epsilon = 1.5$.

In the experiment, we compare the performance of FedAvg, CE-FedAvg and DP-FedAvg on two datasets. In both experiments, we set client number $N = 1920$, the number of client participates in each round $|\mathcal{P}_t| = 80$, $\forall\ t$, the number of local iterations $Q = 32$ and the mini-batch size 64. The clipping threshold is set to 50% of the average (over clients and iterations) of local update magnitudes recorded in FedAvg. For DP-FedAvg we set the clipping threshold the same as in CE-FedAvg, we fix the number of communication rounds and privacy budget for the algorithms to obtain the noise variance that needs to be added. Among all the experiments, we fix privacy budget $\delta = 10^{-5}$.

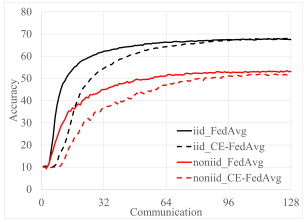
EMNIST dataset. We use the digit part of the EMNIST dataset, which has 240K training samples and 40K testing samples. We distribute the data in the Non-IID way described in Section II and each client has 125 samples. We conduct experiments on a 2-layer MLP with one hidden layer, AlexNet, ModelNetV2 [126] and ResNet-18. The results are listed in Table 6.2

and Figure 6.3.

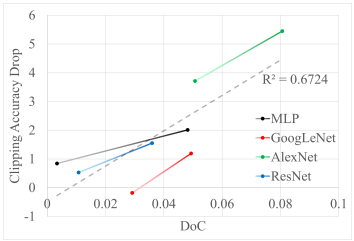
Discussion. Let us discuss the relation between our empirical observations and the theoretical results.

1) One of the main claims we made in Section 6.4 is that clipping performs worse when the update differences are less concentrated and vice versa. To support such a claim, let us first clarify the relationship between the "degree of concentration" (DoC) of the update differences, and the clipping error in Theorem 8. Define the DoC as the averaged *normalized variance* of the magnitude of the clients' update differences, i.e., DoC:= $\mathrm{Var}(\|\Delta\mathbf{x}_i^t\|)/\overline{\|\Delta\mathbf{x}_i^t\|}^4$, where $\overline{\|\Delta\mathbf{x}_i^t\|} := \frac{1}{M}\sum_{i=1}^{M}\|\Delta\mathbf{x}_i^t\|$. Then the clipping error in Theorem 3.1 can be approximated by the DoC as follows: when clipping is activated, the clipping factors are $\alpha_i^t = c/\|\Delta\mathbf{x}_i^t\|$, and the clipping error equals to $c^2 \cdot \mathrm{Var}(1/\|\Delta\mathbf{x}_i^t\|) + c \cdot \mathrm{MAD}(1/\|\Delta\mathbf{x}_i^t\|)$ where MAD denotes Mean Absolute Deviation. With Taylor expansion, the above terms can be approximated by: $c^2 \cdot \mathrm{Var}(\|\Delta\mathbf{x}_i^t\|)/\overline{\|\Delta\mathbf{x}_i^t\|}^4 + c \cdot \sqrt{\mathrm{Var}(\|\Delta\mathbf{x}_i^t\|)/\overline{\|\Delta\mathbf{x}_i^t\|}^4} = c^2 \cdot \mathrm{DoC} + c \cdot \sqrt{\mathrm{DoC}}$. Therefore, DoC can be used to estimate the clipping error.

Based on the above discussion, we conduct experiments on MLP, AlexNet, ResNet-18, and GoogLeNet on both IID and Non-IID Cifar-10 datasets and plot the accuracy drop caused by clipping versus DoC averaged over all iterations. The results are shown in Figure 6.5b. The lighter side of the line denotes the result of IID data, and the darker side denotes the Non-IID data. We can see that when DoC is small (i.e., the update differences' magnitudes are more concentrated), then the accuracy drop caused by the clipping is also small, and vice versa.



(a) GoogLeNet Test Accuracy on Cifar-10 dataset with IID and Non-IID data distribution.



(b) The relationship between the averaged degree of concentration of the update differences and the clipping accuracy drop for the IID (light end) and Non-IID (dark end) Cifar-10 dataset.

2) It appears that when the underlying machine learning model is *structured* (e.g., many layers, has convolution layers, skip connections, etc), the update difference of FedAvg becomes

concentrated, yielding a better clipping performance (as suggested by the terms related to clipping in Theorem 8);

3) When the model has too many parameters and/or layers, they are sensitive to privacy noise. This is reasonable since the error term caused by privacy noise in Theorem 8 is linearly dependent on the size of the model d and the square of the Lipschitz constant L (note, that $\eta_\ell \propto 1/L$). From [127, Corollary 3.3], we know that L increases exponentially with the number of layers. Therefore, larger and deeper models are potentially more sensitive to privacy noise.

4) We conjecture that, to ensure good performance of DP-FedAvg, we need to pick a neural network that is structured enough, while not having too many variables and too many number of layers.

Chapter 7

Conclusion and Discussion

In this chapter, we first summarize the contribution of each chapter and then comment on the potential future work enabled by the results of this thesis.

7.1 Summary

In Chapter 2, we have designed a framework to understand distributed optimization algorithms from a control perspective. We have shown that a multi-rate double-feedback control system can represent a wide range of deterministic distributed optimization algorithms. We use a few examples to demonstrate how the proposed framework can help understand the connection between algorithms, as well as facilitate new algorithm design. In the future, we plan to extend the framework to model distributed stochastic algorithms.

In Chapter 3, we have proposed a feedback-control system to model distributed optimization algorithms from the multi-rate stochastic control perspective. We have shown that the multi-rate stochastic control system can represent a variety of distributed stochastic algorithms. Illustrative examples demonstrate how the system can help understand existing algorithms and design new algorithms.

In Chapter 4, a gradient-based non-convex stochastic decentralized algorithm was proposed for solving non-convex optimization problems over a network. The GNSD is able to process the data locally at each node and minimize the objective function by gradient tracking over the network so that the interest of the parameters can be learned faster without loss of accuracy. The algorithm can be applied to solve multiple learning tasks, when the size of data is large, such as training deep neural networks.

In Chapter 5, we study federated learning under the CTA protocol. We explore a number of theoretical properties of this protocol and design a meta-algorithm called FedPD, which contains various algorithms with desirable properties, such as achieving the best communication/computation complexity and adapting its communication pattern with data heterogeneity.

In Chapter 6, we provide an empirical and theoretical understanding of the clipping operation in FL. We show how to properly combine the clipping operation with existing FL algorithms to achieve the desirable trade-off between convergence and differential privacy guarantees. Extensive numerical results also corroborate our theory and suggest that the distribution of the clients' updates is a key factor that affects the performance of the clipping-enabled FL algorithm.

7.2 Future Research Work

In this section, we outline a few directions for future work.

- Firstly, throughout the thesis, we focus on distributed optimization with model consensus constraint. However, many applications (e.g., distributed power generation, distributed vehicle control) have more complicated network constraints on the optimization variables. Therefore, we are interested in developing a generic framework for solving distributed optimization with complex network constraints.
- Secondly, we would like to extend the framework to complex global communication controllers, e.g., with a directed graph where the communication matrix is no longer symmetric, or when the consensus controllers are no longer linear. In those cases, the analysis of the current framework no longer applies and requires a different analysis technique.
- Finally, it is worth investigating whether the accuracy drop caused by the clipping operation in Chapter 6 can be reduced or fully eliminated with more advanced algorithms.

References

[1] Mu Li, David G Andersen, Alexander J Smola, and Kai Yu. Communication efficient distributed machine learning with the parameter server. In *Advances in Neural Information Processing Systems (NIPS)*, pages 19–27, 2014.

[2] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1223–1231, 2012.

[3] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5336–5346, 2017.

[4] Jing Wang and Nicola Elia. A control perspective for centralized and distributed convex optimization. In *2011 50th IEEE conference on decision and control and European control conference*, pages 3800–3805. IEEE, 2011.

[5] Tsung-Hui Chang, Mingyi Hong, Hoi-To Wai, Xinwei Zhang, and Songtao Lu. Distributed learning in the nonconvex world: From batch data to streaming and beyond. *IEEE Signal Processing Magazine*, 37(3):26–38, 2020.

[6] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[7] Angelia Nedić, Alex Olshevsky, Asuman Ozdaglar, and John N Tsitsiklis. On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11):2506–2517, 2009.

[8] Kun Yuan, Qing Ling, and Wotao Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

[9] Qing Ling, Wei Shi, Gang Wu, and Alejandro Ribeiro. DLM: Decentralized linearized alternating direction method of multipliers. *IEEE Transactions on Signal Processing*, 63(15):4051–4064, 2015.

[10] Kun Yuan, Wei Xu, and Qing Ling. Can primal methods outperform primal-dual methods in decentralized dynamic optimization? *IEEE Transactions on Signal Processing*, 68:4466–4480, 2020.

[11] Paolo Di Lorenzo and Gesualdo Scutari. Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016.

[12] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[13] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale: System design. *Proceedings of machine learning and systems*, 1:374–388, 2019.

[14] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. First analysis of local gd on heterogeneous data. *arXiv preprint arXiv:1909.04715*, 2019.

[15] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2019.

[16] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. volume 2, pages 429–450, 2020.

[17] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

[18] Xinwei Zhang, Mingyi Hong, Sairaj Dhople, Wotao Yin, and Yang Liu. FedPD: A federated learning framework with adaptivity to non-iid data. *IEEE Transactions on Signal Processing*, 69:6055–6070, 2021.

[19] Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. pages 3027–3036, 2017.

[20] Haoran Sun and Mingyi Hong. Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms. *IEEE Transactions on Signal processing*, 67(22):5912–5928, 2019.

[21] Debraj Basu, Deepesh Data, Can Karakus, and Suhas Diggavi. Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations. *Advances in Neural Information Processing Systems*, 32, 2019.

[22] Anastasia Koloskova, Tao Lin, Sebastian U Stich, and Martin Jaggi. Decentralized deep learning with arbitrary communication compression. In *International Conference on Learning Representations*, 2019.

[23] Tianyi Chen, Georgios B Giannakis, Tao Sun, and Wotao Yin. Lag: lazily aggregated gradient for communication-efficient distributed learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 5055–5065, 2018.

[24] Jun Sun, Tianyi Chen, Georgios B Giannakis, Qinmin Yang, and Zaiyue Yang. Lazily aggregated quantized gradient innovation for communication-efficient federated learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[25] Songtao Lu, Xinwei Zhang, Haoran Sun, and Mingyi Hong. GNSD: A gradient-tracking based nonconvex stochastic algorithm for decentralized optimization. In *2019 IEEE Data Science Workshop (DSW)*, pages 315–321. IEEE, 2019.

[26] Yucheng Lu and Christopher De Sa. Optimal complexity in decentralized training. In *International Conference on Machine Learning*, pages 7111–7123. PMLR, 2021.

[27] Yuan Yuan, Zongrui Zou, Dong Li, Li Yan, Dongxiao Yu, and Zhuojun Duan. D-(dp)2sgd: Decentralized parallel sgd with differential privacy in dynamic networks. *Wirel. Commun. Mob. Comput.*, 2021, jan 2021.

[28] Deming Yuan, Daniel WC Ho, and Shengyuan Xu. Zeroth-order method for distributed optimization with approximate projections. *IEEE transactions on neural networks and learning systems*, 27(2):284–294, 2015.

[29] Davood Hajinezhad, Mingyi Hong, and Alfredo Garcia. Zone: Zeroth-order nonconvex multiagent optimization over networks. *IEEE Transactions on Automatic Control*, 64(10):3995–4010, 2019.

[30] Riccarda Rossi and Giuseppe Savaré. Gradient flows of non convex functionals in hilbert spaces and applications. *ESAIM: Control, Optimisation and Calculus of Variations*, 12(3):564–614, 2006.

[31] Akhil Sundararajan. *Analysis and Design of Distributed Optimization Algorithms*. The University of Wisconsin-Madison, 2021.

[32] Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.

[33] Bin Hu and Laurent Lessard. Control interpretations for first-order optimization methods. In *2017 American Control Conference (ACC)*, pages 3114–3119. IEEE, 2017.

[34] Michael Muehlebach and Michael Jordan. A dynamical systems perspective on nesterov acceleration. In *International Conference on Machine Learning*, pages 4656–4662, 2019.

[35] Brian Swenson, Ryan Murray, H Vincent Poor, and Soummya Kar. Distributed gradient flow: Nonsmoothness, nonconvexity, and saddle point evasion. *IEEE Transactions on Automatic Control*, 2021.

[36] Guilherme França, Daniel P Robinson, and Rene Vidal. A dynamical systems perspective on nonsmooth constrained optimization. *arXiv preprint arXiv:1808.04048*, 2018.

[37] Brian Swenson, Ryan Murray, H Vincent Poor, and Soummya Kar. Distributed gradient descent: Nonconvergence to saddle points and the stable-manifold theorem. *arXiv preprint arXiv:1908.02747*, 2019.

[38] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[39] Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7184–7193, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[40] Zhanhong Jiang, Aditya Balu, Chinmay Hegde, and Soumik Sarkar. Collaborative deep learning in fixed topology networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5904–5914, 2017.

[41] Pascal Bianchi and Jérémie Jakubowicz. Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization. *IEEE Transactions on Automatic Control*, 58(2):391–405, 2013.

[42] Pascal Bianchi, Gersende Fort, and Walid Hachem. Performance of a distributed stochastic approximation algorithm. *IEEE Transactions on Information Theory*, 59(11):7405–7418, 2013.

[43] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.

[44] Ligeng Zhu and Song Han. Deep leakage from gradients. In *Federated Learning*, pages 17–31. Springer, 2020.

[45] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. A framework for evaluating gradient leakage attacks in federated learning. *arXiv preprint arXiv:2004.10397*, 2020.

[46] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[47] Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. LDP-Fed: Federated learning with local differential privacy. In *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, pages 61–66, 2020.

[48] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 1–11, 2019.

[49] Lun Wang, Ruoxi Jia, and Dawn Song. D2p-fed: Differentially private federated learning with efficient communication. *arXiv preprint arXiv:2006.13039*, 2020.

[50] Aleksei Triastcyn and Boi Faltings. Federated learning with bayesian differential privacy. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2587–2596. IEEE, 2019.

[51] Cynthia Dwork, F. McSherry, K. Nissim, and A. Smith. *Calibrating Noise to Sensitivity in Private Data Analysis*, pages 265–284. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[52] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

[53] Xiangyi Chen, Steven Z Wu, and Mingyi Hong. Understanding gradient clipping in private sgd: A geometric perspective. *Advances in Neural Information Processing Systems*, 33, 2020.

[54] Shuang Song, Thomas Steinke, Om Thakkar, and Abhradeep Thakurta. Evading the curse of dimensionality in unconstrained private glms. In *International Conference on Artificial Intelligence and Statistics*, pages 2638–2646. PMLR, 2021.

[55] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[56] Xinwei Zhang, Mingyi Hong, and Nicola Elia. Understanding a class of decentralized and federated optimization algorithms: A multirate feedback control perspective. *SIAM Journal on Optimization*, 33(2):652–683, 2023.

[57] Xinwei Zhang, Mingyi Hong, Sairaj Dhople, and Nicola Elia. A stochastic multi-rate control framework for modeling distributed optimization algorithms. In *International Conference on Machine Learning*, pages 26206–26222. PMLR, 2022.

[58] Xinwei Zhang, Xiangyi Chen, Mingyi Hong, Steven Wu, and Jinfeng Yi. Understanding clipping for federated learning: Convergence and client-level differential privacy. In *International Conference on Machine Learning*, pages 26048–26067. PMLR, 2022.

[59] Xinwei Zhang, John Sartori, Mingyi Hong, and Sairaj Dhople. Implementing first-order optimization methods: Algorithmic considerations and bespoke microcontrollers. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 296–300. IEEE, 2019.

[60] Xinwei Zhang, Victor Purba, Mingyi Hong, and Sairaj Dhople. A sum-of-squares optimization method for learning and controlling photovoltaic systems. In *2020 American Control Conference (ACC)*, pages 2376–2381. IEEE, 2020.

[61] Xinwei Zhang, Wotao Yin, Mingyi Hong, and Tianyi Chen. Hybrid federated learning: Algorithms and implementation. *arXiv preprint arXiv:2012.12420*, 2020.

[62] Yang Liu, Xinwei Zhang, Yan Kang, Liping Li, Tianjian Chen, Mingyi Hong, and Qiang Yang. Fedbcd: A communication-efficient collaborative learning framework for distributed features. *IEEE Transactions on Signal Processing*, 70:4277–4290, 2022.

[63] Bingqing Song, Prashant Khanduri, Xinwei Zhang, Jinfeng Yi, and Mingyi Hong. Fedavg converges to zero training loss linearly for overparameterized multi-layer neural networks. 2023.

[64] Xinwei Zhang, Bingqing Song, Mehrdad Honarkhah, Jie Ding, and Mingyi Hong. Building large machine learning models from small distributed models: A layer matching approach.

In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

[65] Xinwei Zhang, Mingyi Hong, and Jie Chen. Glasu: A communication-efficient algorithm for federated learning with vertically distributed graph data. *arXiv preprint arXiv:2303.09531*, 2023.

[66] Greg Droge, Hiroaki Kawashima, and Magnus B Egerstedt. Continuous-time proportional-integral distributed optimisation for networked systems. *Journal of Control and Decision*, 1(3):191–213, 2014.

[67] Euhanna Ghadimi, Mikael Johansson, and Iman Shames. Accelerated gradient methods for networked optimization. In *Proceedings of the 2011 American Control Conference*, pages 1668–1673. IEEE, 2011.

[68] Alex Olshevsky and John N Tsitsiklis. Convergence speed in distributed consensus and averaging. *SIAM journal on control and optimization*, 48(1):33–55, 2009.

[69] Sébastien Bubeck, Yin Tat Lee, and Mohit Singh. A geometric alternative to nesterov's accelerated gradient descent. *arXiv preprint arXiv:1506.08187*, 2015.

[70] Haishan Ye, Luo Luo, Ziang Zhou, and Tong Zhang. Multi-consensus decentralized accelerated gradient descent. *arXiv preprint arXiv:2005.00797*, 2020.

[71] Antonio Orvieto and Aurelien Lucchi. Continuous-time models for stochastic optimization algorithms. volume 32, 2019.

[72] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *International Conference on Learning Representations*, 2021.

[73] Benjamin C. Kuo. *Digital Control Systems*. Oxford series in electrical and computer engineering. Oxford University Press, 1992.

[74] Zhi Li, Wei Shi, and Ming Yan. A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Transactions on Signal Processing*, 67(17):4494–4506, 2019.

[75] Haoran Sun, Songtao Lu, and Mingyi Hong. Improving the sample and communication complexity for decentralized non-convex optimization: Joint gradient estimation and tracking. In *International Conference on Machine Learning*, pages 9217–9228. PMLR, 2020.

[76] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

[77] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, pages 5381–5393. PMLR, 2020.

[78] Akhil Sundararajan, Bryan Van Scoy, and Laurent Lessard. A canonical form for first-order distributed optimization algorithms. In *2019 American Control Conference (ACC)*, pages 4075–4080. IEEE, 2019.

[79] Alexander Rogozin, Mikhail Bochko, Pavel Dvurechensky, Alexander Gasnikov, and Vladislav Lukoshkin. An accelerated method for decentralized distributed stochastic optimization over time-varying graphs. pages 3367–3373, 2021.

[80] Jinshan Zeng and Wotao Yin. On nonconvex decentralized gradient descent. *IEEE Transactions on signal processing*, 66(11):2834–2848, 2018.

[81] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021.

[82] Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7663–7673, 2018.

[83] Zhenheng Tang, Shaohuai Shi, Xiaowen Chu, Wei Wang, and Bo Li. Communication-efficient distributed deep learning: A comprehensive survey. *arXiv preprint arXiv:2003.06307*, 2020.

[84] Anit Kumar Sahu, Dusan Jakovetic, Dragana Bajovic, and Soummya Kar. Distributed zeroth order optimization over random networks: A kiefer-wolfowitz stochastic approximation approach. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4951–4958. IEEE, 2018.

[85] Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *International Conference on Machine Learning*, pages 3478–3487. PMLR, 2019.

[86] Anestis Antoniadis, Irène Gijbels, and Mila Nikolova. Penalized likelihood regression for generalized linear models with non-quadratic penalties. *Annals of the Institute of Statistical Mathematics*, 63(3):585–615, 2011.

[87] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[88] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

[89] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[90] A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[91] Songtao Lu, Desheng Liu, and Jinping Sun. A distributed adaptive gsc beamformer over coordinated antenna arrays network for interference mitigation. In *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 237–242. IEEE, 2012.

[92] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

[93] Songtao Lu, VH Nascimento, Jinping Sun, and Zhuangji Wang. Sparsity-aware adaptive link combination approach over distributed networks. *Electronics Letters*, 50(18):1285–1287, 2014.

[94] Dušan Jakovetić, José MF Moura, and Joao Xavier. Linear convergence rate of a class of distributed augmented lagrangian algorithms. *IEEE Transactions on Automatic Control*, 60(4):922–936, 2015.

[95] I. Schizas, G. Mateos, and G. Giannakis. Distributed LMS for consensus-based in-network adaptive processing,. *IEEE Transactions on Signal Processing*, 57(6):2365 – 2382, 2009.

[96] Mingyi Hong, Zhi-Quan Luo, and Meisam Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

[97] Mingyi Hong, Davood Hajinezhad, and Ming-Min Zhao. Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks. In *International Conference on Machine Learning*, pages 1529–1538. PMLR, 2017.

[98] Amir Daneshmand, Gesualdo Scutari, and Vyacheslav Kungurtsev. Second-order guarantees of distributed gradient algorithms. *SIAM Journal on Optimization*, 30(4):3029–3068, 2020.

[99] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D^2: Decentralized training over decentralized data. In *Proceedings of International Conference on Machine Learning (ICML)*, volume 80, pages 4848–4856, July 2018.

[100] David W Walker and Jack J Dongarra. MPI: a standard message passing interface. Technical report, 1996.

[101] Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *AT&T Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2:18, 2010.

[102] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR, 2020.

[103] Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5693–5700, 2019.

[104] Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of local-update sgd algorithms. *The Journal of Machine Learning Research*, 22(1):9709–9758, 2021.

[105] Sebastian Urban Stich. Local sgd converges fast and communicates little. *ICLR 2019 - International Conference on Learning Representations*, page 17, 2019.

[106] Reese Pathak and Martin J Wainwright. Fedsplit: An algorithmic framework for fast federated optimization. volume 33, pages 7057–7066, 2020.

[107] Honglin Yuan and Tengyu Ma. Federated accelerated stochastic gradient descent. volume 33, 2020.

[108] Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng. Variance reduced local sgd with lower communication complexity. *arXiv preprint arXiv:1912.12844*, 2019.

[109] Shicong Cen, Huishuai Zhang, Yuejie Chi, Wei Chen, and Tie-Yan Liu. Convergence of distributed stochastic variance reduced methods without sampling extra data. *IEEE Transactions on Signal Processing*, 68:3976–3989, 2020.

[110] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020.

[111] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

[112] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, 184(1-2):71–120, 2020.

[113] Kun Yuan, Bicheng Ying, Stefan Vlaski, and Ali H Sayed. Stochastic gradient descent with finite samples sizes. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2016.

[114] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smithy. Feddane: A federated newton-type method. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 1227–1231. IEEE, 2019.

[115] Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in Non-IID federated learning. *International Conference on Learning Representations*, 2021.

[116] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[117] Olivia Choudhury, Aris Gkoulalas-Divanis, Theodoros Salonidis, Issa Sylla, Yoonyoung Park, Grace Hsu, and Amar Das. Differential privacy-enabled federated learning for sensitive health data. *arXiv preprint arXiv:1910.02578*, 2019.

[118] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[119] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[120] Shuang Song, Om Thakkar, and Abhradeep Thakurta. Characterizing private clipped gradient descent on convex generalized linear problems. *arXiv preprint arXiv:2006.06783*, 2020.

[121] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[122] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[123] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. EMNIST: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[124] Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009.

[125] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[126] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[127] Calypso Herrera, Florian Krach, and Josef Teichmann. Estimating full lipschitz constants of deep neural networks. *arXiv preprint arXiv:2004.13135*, 2020.

[128] René A Poliquin and R Tyrrell Rockafellar. Generalized hessian properties of regularized nonsmooth functions. *SIAM Journal on Optimization*, 6(4):1121–1137, 1996.

[129] Tengfei Liu, Zhong-Ping Jiang, and David J Hill. *Nonlinear control of dynamic networks*. CRC Press, 2018.

[130] Dušan Jakovetić. A unification and generalization of exact distributed first-order methods. *IEEE Transactions on Signal and Information Processing over Networks*, 5(1):31–46, 2018.

[131] M Reyasudin Basir Khan, Razali Jidin, and Jagadeesh Pasupuleti. Multi-agent based distributed control architecture for microgrid energy management and optimization. *Energy Conversion and Management*, 112:288–307, 2016.

[132] Khaoula El Mekkaoui, Diego Mesquita, Paul Blomstedt, and Samuel Kaski. Federated stochastic gradient langevin dynamics. In *Uncertainty in Artificial Intelligence*, pages 1703–1712. PMLR, 2021.

[133] Ngoc Huy Chau, Éric Moulines, Miklos Rásonyi, Sotirios Sabanis, and Ying Zhang. On stochastic gradient langevin dynamics with dependent data streams: The fully nonconvex case. *SIAM Journal on Mathematics of Data Science*, 3(3):959–986, 2021.

[134] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[135] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

Appendix A

Additional Results and Proofs of Chapter 2

A.1 Proofs of Section 2.4

Let t_ℓ (resp. t_g) denote the time at which the local (resp. global) controller samples, that is: $t_\ell := t - t \bmod \tau_\ell$ and $t_g := t - t \bmod \tau_g$. To simplify the analysis, we treat the stepsizes $\eta_\ell(t), \eta_g(t)$ as constants in each sampling intervals. Also recall that $\mathbf{y}(t) = [\mathbf{x}(t); \mathbf{v}(t)]$. The following relations will be useful:

$$\langle a, b\rangle = \frac{1}{2\alpha}\|a\|^2 + \frac{\alpha}{2}\|b\|^2 - \frac{1}{2}\left\|\frac{1}{\sqrt{\alpha}}a + \sqrt{\alpha}b\right\|^2 \leq \frac{1}{2\alpha}\|a\|^2 + \frac{\alpha}{2}\|b\|^2 \tag{A.1}$$

$$(I-R)^2 = I - 2R + R^2 = I - R, \quad \|R\| \leq 1, \quad \|I-R\| \leq 1. \tag{A.2}$$

The proofs of Lemma 1 - Lemma 3 adopt a similar concept in robust control theory. The time derivative of the energy function of the discretized system is given by:

$$\dot{\mathcal{E}}(t) = \underbrace{-\left\langle \nabla f(\bar{\mathbf{x}}(t)), \frac{1}{N}\mathbb{1}^T \eta_\ell(t) u_{\ell,x}(t)\right\rangle - \langle (I-R)\cdot \mathbf{y}(t), \eta_\ell(t)\cdot u_{\ell,y}(t) + \eta_g(t)\cdot u_g(t)\rangle}_{\text{term I}}$$
$$+ \hat{\mathcal{E}}(t), \tag{A.3}$$

where "term I" is the derivative of the continuous-time energy function given in (2.10); $\hat{\mathcal{E}}(t)$ is the error caused by discretization. Integrate (A.3) and apply P5, we have:

$$\int_0^t \dot{\mathcal{E}}(t) \leq -\int_0^t \gamma_1(\tau)\|\nabla f(\bar{\mathbf{x}}(\tau))\|^2 + \gamma_2(\tau)\|(I-R)\cdot\mathbf{y}(\tau)\|^2 \,\mathrm{d}\tau + \int_0^t \hat{\mathcal{E}}(\tau)\mathrm{d}\tau. \tag{A.4}$$

The key idea of proofs is to bound $\int_0^t \hat{\mathcal{E}}(\tau)d\tau$ by the first two terms.

A.1.1 Proof of Lemma 1

In this case $\hat{u}_g(t) = G_g(\mathbf{x}(t_g), \mathbf{v}(t_g); A)$. By taking derivative of $\mathcal{E}(t)$, and by comparing with (A.3), we can obtain

$$\hat{\mathcal{E}}(t) = \eta_g(t) \langle (I - R) \cdot \mathbf{y}(t), u_g(t) - \hat{u}_g(t) \rangle. \tag{A.5}$$

Next, we bound $\int_0^t \hat{\mathcal{E}}(\tau) d\tau$. Towards this end, we first observe that:

$$\begin{aligned}
\langle (I - R) \cdot \mathbf{y}(t), u_g(t) - \hat{u}_g(t) \rangle &\overset{(i)}{=} \langle (I - R) \cdot \mathbf{y}(t), G_g(\mathbf{y}(t) - \mathbf{y}(t_g); A) \rangle \\
&= \left\langle (I - R) \cdot \mathbf{y}(t), G_g\Big(\int_{t_g}^t \dot{\mathbf{y}}(s) ds; A \Big) \right\rangle \\
&\overset{\text{(A.1)}}{\leq} \frac{\gamma_2(t)}{2} \left\| (I - R) \cdot \mathbf{y}(t) \right\|^2 + \frac{1}{2\gamma_2(t)} \left\| G_g\Big(\int_{t_g}^t \dot{\mathbf{y}}(s) ds; A \Big) \right\|^2,
\end{aligned}$$

where (i) is due to the linearity property P2. Next, we bound the last term above by $\|\nabla f(\bar{\mathbf{x}}(t))\|^2$ and $\|(I - R) \cdot \mathbf{y}(t)\|^2$. To proceed, let us define

$$\begin{aligned}
&\tilde{\mathbf{y}}(t) := G_g\Big(\int_{t_g}^t \dot{\mathbf{y}}(s) ds; A \Big) = u_g(t) - \hat{u}_g(t), \quad \mathbf{w}(t) := [(I - R) \cdot \mathbf{y}(t); \nabla f(\bar{\mathbf{x}}(t))], \\
&q(t) := \left\| G_g\Big(\int_{t_g}^t \dot{\mathbf{y}}(s) ds; A \Big) \right\| / \left\| [(I - R) \cdot \mathbf{y}(t); \nabla f(\bar{\mathbf{x}}(t))] \right\| = \|\tilde{\mathbf{y}}(t)\| / \|\mathbf{w}(t)\|.
\end{aligned} \tag{A.6}$$

Using the above definition, we have:

$$\left\| G_g\big(\int_{t_g}^t \dot{\mathbf{y}}(s) ds; A \big) \right\|^2 = \|\tilde{\mathbf{y}}(t)\|^2 = q^2(t) \|\mathbf{w}(t)\|^2. \tag{A.7}$$

It then suffices to bound $q(t)$. Towards this end, let us first bound $\|\dot{\mathbf{w}}(t)\|$ by:

$$\begin{aligned}
\|\dot{\mathbf{w}}(t)\| &\overset{(i)}{=} \left\| \left[(I - R) \cdot (\eta_g(t) \hat{u}_g(t) + \eta_\ell(t) u_{\ell,y}(t)); \left\langle \partial^2 f(\bar{\mathbf{x}}(t)), \eta_\ell(t) \frac{\mathbb{1}^T}{N} u_{\ell,x}(t) \right\rangle \right] \right\| \\
&\leq \eta_g(t) \|(I - R) \cdot \hat{u}_g(t)\| + \min\left\{ \eta_\ell(t), \frac{\eta_\ell(t) \left\| \partial^2 f(\bar{\mathbf{x}}(t)) \right\|}{N} \right\} \|u_{\ell,y}(t)\| \\
&\overset{(ii)}{\leq} \eta_g(t) \left(\|(I - R) \cdot (u_g(t) - \hat{u}_g(t))\| + \|(I - R) \cdot u_g(t)\| \right) \\
&\quad + \sqrt{C_x^2 + C_v^2} \cdot \eta_\ell(t) \cdot (1 + \frac{L_f}{N}) \cdot \|\nabla f(\mathbf{x}(t))\| \\
&\overset{(iii)}{\leq} \eta_g(t) \left(\|\tilde{\mathbf{y}}(t)\| + \|(I - R) \cdot \mathbf{y}(t)\| \right) + \sqrt{C_x^2 + C_v^2} \cdot \eta_\ell(t) \cdot (1 + \frac{L_f}{N}) \cdot \|\nabla f(\mathbf{x}(t))\| \\
&\overset{(iv)}{\leq} \eta_g(t) \cdot q(t) \cdot \|\mathbf{w}(t)\| + \eta_g(t) \cdot \|(I - R) \cdot \mathbf{y}(t)\| \\
&\quad + \sqrt{C_x^2 + C_v^2} \cdot \eta_\ell(t) \cdot (1 + \frac{L_f}{N}) \cdot \left(\|\nabla f(\bar{\mathbf{x}}(t))\| + \frac{L_f}{N} \|(I - R) \cdot \mathbf{x}(t)\| \right)
\end{aligned}$$

$$\overset{(v)}{\leq} \sqrt{2}\left(\eta_g(t)q(t) + \eta_g(t) + \sqrt{C_x^2 + C_v^2}\cdot\eta_\ell(t)\cdot\left(1+\frac{L_f}{N}\right)^2\right)\cdot\|\mathbf{w}(t)\|, \tag{A.8}$$

where (i) can be derived similarly as in (2.10); in (ii) we add and subtract $u_g(t)$ to the first term, apply P4 to the last term, used the following definition of sub-Hessian:

$$\lim_{\delta\to 0}\frac{\left\|f(x+\delta) - f(x) - \langle\nabla f(x),\delta\rangle - \frac{1}{2}\delta^T\partial^2 f(x)\delta\right\|}{\|\delta\|^2} = 0,$$

and the fact that that under the smoothness A2, it holds that $\left\|\partial^2 f(\mathbf{x})\right\| \leq L$ [128, Theorem 3.1]; in (iii) we combine $\|I-R\|\leq 1$ and (2.4) to the second term, use the definition of $\tilde{\mathbf{y}}(t)$ in (A.6); in (iv) we use the definition of $q(t)$ in (A.6), add and subtract $\nabla f(\bar{\mathbf{x}}(t))$ to the last term and apply A2; in (v) we use the fact that $\|a\|+\|b\|\leq\sqrt{2(\|a\|^2+\|b\|^2)}$, and $\mathbf{x}$ is a subvector of $\mathbf{y}$. Then we can bound $\dot{q}(t)$ by:

$$\begin{aligned}
\dot{q}(t) &= \frac{\dot{\tilde{\mathbf{y}}}(t)^T\tilde{\mathbf{y}}(t)}{\|\mathbf{w}(t)\|\,\|\tilde{\mathbf{y}}(t)\|} - \frac{\|\tilde{\mathbf{y}}(t)\|\,\mathbf{w}(t)^T\dot{\mathbf{w}}(t)}{\|\mathbf{w}(t)\|^3}\\
&\overset{(i)}{\leq} \frac{\left\|\dot{\tilde{\mathbf{y}}}(t)\right\|\|\tilde{\mathbf{y}}(t)\|}{\|\mathbf{w}(t)\|\,\|\tilde{\mathbf{y}}(t)\|} + \frac{\|\tilde{\mathbf{y}}(t)\|\,\|\mathbf{w}(t)\|\,\|\dot{\mathbf{w}}(t)\|}{\|\mathbf{w}(t)\|^3} \overset{(ii)}{\leq} (1+q(t))\frac{\|\dot{\mathbf{w}}(t)\|}{\|\mathbf{w}(t)\|}\\
&\overset{(A.8)}{\leq} (1+q(t))\cdot\sqrt{2}\left(q(t)\eta_g(t) + \eta_g(t) + \sqrt{C_x^2+C_v^2}\eta_\ell(t)\cdot\left(1+\frac{L_f}{N}\right)^2\right),
\end{aligned}$$

where in (i) we apply the Cauchy–Schwarz inequality; (ii) is due to the definition of $q(t)$ in (A.6), and the relations below (where equality comes from the linearity property P2):

$$\left\|\dot{\tilde{\mathbf{y}}}(t)\right\| = \|G_g(\dot{\mathbf{y}}(t);A)\| \overset{(2.4)}{\leq} \|(I-R)\cdot\dot{\mathbf{y}}(t)\| \leq \|\dot{\mathbf{w}}(t)\|.$$

Note that $q(t_g)=0$, solve the above inequality of $\dot{q}(t)$ by using Grownwall's inequality, we obtain $q(t)\leq q_{\max} := \exp\left\{\sqrt{2}\tau_g\cdot\left(\sqrt{C_x^2+C_v^2}\cdot\eta_\ell(t)\cdot\left(1+L_f/N\right)^2\right)\right\} - 1$. Plug in this estimate to (A.7), and further to (A.5) and (A.4), we obtain:

$$\begin{aligned}
\int_0^t\dot{\mathcal{E}}(\tau)\mathrm{d}\tau &\leq \int_0^t\left(-\gamma_1(\tau)\|\nabla f(\bar{\mathbf{x}}(\tau))\|^2 - \gamma_2(\tau)\|(I-R)\cdot\mathbf{y}(\tau)\|^2\right)\mathrm{d}\tau\\
&\quad + \int_0^t\left(\frac{\gamma_2(\tau)}{2}\|(I-R)\cdot\mathbf{y}(\tau)\|^2 + \frac{1}{2\gamma_2(\tau)}q_{\max}^2\|\mathbf{w}(\tau)\|^2\right)\mathrm{d}\tau\\
&= \int_0^t -\left(\gamma_1(\tau) - \frac{q_{\max}^2}{2\gamma_2(\tau)}\right)\cdot\|\nabla f(\bar{\mathbf{x}}(\tau))\|^2 - \left(\frac{\gamma_2(\tau)}{2} - \frac{q_{\max}^2}{2\gamma_2(\tau)}\right)\cdot\|(I-R)\cdot\mathbf{y}(\tau)\|^2\,\mathrm{d}\tau.
\end{aligned}$$

A.1.2 Proof of Lemma 2

For notation simplicity, let us define the discrete time controller output as $\hat{u}_{i,\ell}(t) = G_{i,\ell}(x_i(t_\ell), v_i(t_\ell), z_i(t_\ell); f_i)$. Then we can write $\dot{\mathcal{E}}(t)$ similarly as in (A.3), and the error term $\hat{\mathcal{E}}(t)$ in this case can be expressed,

and bounded as below:

$$\begin{aligned}
\hat{\mathcal{E}}(t) &= \left\langle \nabla f(\bar{\mathbf{x}}(t)), \frac{\eta_\ell(t)}{N} \mathbb{1}^T (u_{\ell,x}(t) - \hat{u}_{\ell,x}(t)) \right\rangle + \langle (I-R)\mathbf{y}(t), \eta_\ell(t)(I-R)\cdot(u_{\ell,y}(t) - \hat{u}_{\ell,y}(t))\rangle \\
&\overset{(A.1)}{\leq} \frac{\gamma_1(t)}{2} \left\|\nabla f(\bar{\mathbf{x}}(t))\right\|^2 + \frac{\gamma_2(t)}{2} \left\|(I-R)\cdot\mathbf{y}(t)\right\|^2 \\
&\quad + \frac{\eta_\ell^2(t)}{2N\gamma_1(t)} \left\|R\cdot(u_{\ell,y}(t) - \hat{u}_{\ell,y}(t))\right\|^2 + \frac{\eta_\ell^2(t)}{2\gamma_2(t)} \left\|(I-R)\cdot(u_{\ell,y}(t) - \hat{u}_{\ell,y}(t))\right\|^2 \\
&\leq \frac{\gamma_1(t)}{2} \left\|\nabla f(\bar{\mathbf{x}}(t))\right\|^2 + \frac{\gamma_2(t)}{2} \left\|(I-R)\cdot\mathbf{y}(t)\right\|^2 \\
&\quad + \frac{\eta_\ell^2(t) L^2}{2\min\{N\gamma_1(t), \gamma_2(t)\}} \left(\left\|\mathbf{y}(t) - \mathbf{y}(t_\ell)\right\|^2 + \left\|\mathbf{z}(t) - \mathbf{z}(t_\ell)\right\|^2 \right). \qquad \text{(A.9)}
\end{aligned}$$

where the last inequality combines (A.2) and the Lipschitz gradient property P3, which gives:

$$\begin{aligned}
\left\|u_{\ell,y}(t) - \hat{u}_{\ell,y}(t)\right\|^2 &= \sum_{i=1}^N \left\|G_\ell(\mathbf{x}_i(t), \mathbf{v}_i(t), \mathbf{z}_i(t)) - G_\ell(\mathbf{x}_i(t_\ell), \mathbf{v}_i(t_\ell), \mathbf{z}_i(t_\ell))\right\|^2 \\
&\leq L^2 (\left\|\mathbf{y}(t) - \mathbf{y}(t_\ell)\right\|^2 + \left\|\mathbf{z}(t) - \mathbf{z}(t_\ell)\right\|^2).
\end{aligned}$$

The key step is to bound the last term in (A.9). Towards this end, first note that we have the following relations from (2.16) and P2:

$$\begin{aligned}
(I-R)\cdot\dot{\mathbf{y}}(t) &= -\eta_g(t)\cdot(I-R)\cdot u_{g,y}(t) - \eta_\ell(t)\cdot(I-R)\cdot\hat{u}_{\ell,y}(t). \\
&= -\eta_g(t)\cdot(I-R)\cdot W_A\mathbf{y}(t) - \eta_\ell(t)\cdot(I-R)\cdot\hat{u}_{\ell,y}(t).
\end{aligned}$$

Solving this differential equation with initial condition $\mathbf{y}(t_\ell)$, we obtain:

$$(I-R)\cdot\mathbf{y}(t) = \mathrm{e}^{-(I-R)\cdot W_A \int_{t_\ell}^t \eta_g(s)\mathrm{d}s} \left(\mathbf{y}(t_\ell) - \int_{t_\ell}^t \eta_\ell(s) \mathrm{e}^{(I-R)\cdot W_A \int_{t_\ell}^s \eta_g(s_1)\mathrm{d}s_1} \mathrm{d}s \cdot \hat{u}_{\ell,y}(t) \right). \qquad \text{(A.10)}$$

This expression for $\mathbf{y}(t_\ell)$ can be used to further bound the following term:

$$
\begin{aligned}
&\|(I-R)\cdot(\mathbf{y}(t)-\mathbf{y}(t_\ell))\|^2 \\
&\overset{(A.10)}{=} \Bigg\|(I-R)\cdot\Bigg(\mathbf{y}(t)-\left(\mathrm{e}^{-(I-R)\cdot W_A\int_{t_\ell}^t \eta_g(s)\mathrm{d}s}\right)^{-1}(I-R)\cdot\mathbf{y}(t) \qquad (A.11)\\
&\quad -\int_{t_\ell}^t \eta_\ell(s)\mathrm{e}^{(I-R)\cdot W_A\int_{t_\ell}^s \eta_g(s_1)\mathrm{d}s_1}\mathrm{d}s\cdot\hat{u}_{\ell,y}(t)\Bigg)\Bigg\|^2 \\
&\overset{(i)}{\le}(1+\beta)\left\|I-(I-R)\cdot\left(\mathrm{e}^{-(I-R)\cdot W_A\int_{t_\ell}^t \eta_g(s)\mathrm{d}s}\right)^{-1}\right\|^2\|(I-R)\cdot\mathbf{y}(t)\|^2 \\
&\qquad +(1+\frac{1}{\beta})\left\|\int_{t_\ell}^t \eta_\ell(s)\mathrm{e}^{(I-R)\cdot W_A\int_{t_\ell}^s \eta_g(s_1)\mathrm{d}s_1}\mathrm{d}s\cdot(I-R)\cdot\hat{u}_{\ell,y}(t))\right\|^2 \\
&\overset{(ii)}{\le}(1+\beta)\cdot\left(\frac{1-C_y}{C_y}\right)^2\cdot\|(I-R)\cdot\mathbf{y}(t)\|^2+(1+\frac{1}{\beta})\cdot\left(\frac{\tau_\ell\eta_\ell(t)}{C_y}\right)^2\cdot\|(I-R)\cdot\hat{u}_{\ell,y}(t))\|^2 \\
&\overset{(iii)}{=}\left(\frac{1-C_y}{C_y^2}\right)\cdot\|(I-R)\cdot\mathbf{y}(t)\|^2+\left(\frac{\tau_\ell^2\eta_\ell^2(t)}{C_y}\right)\cdot\|(I-R)\cdot\hat{u}_{\ell,y}(t))\|^2, \qquad (A.12)
\end{aligned}
$$

where in (i) we use Cauchy–Schwarz inequality (with $\beta>0$ being an arbitrary constant); in (ii) we bound the first norm with P1 so that $\|(I-R)W_A\|=\|W_A\|\ge C_g$, which implies the following:

$$
\left\|I-(I-R)\cdot\left(\mathrm{e}^{-(I-R)\cdot W_A\int_{t_\ell}^t \eta_g(s)\mathrm{d}s}\right)^{-1}\right\|^2\le\left(1-(\mathrm{e}^{-C_g\int_{t_\ell}^t \eta_g(s)\mathrm{d}s})^{-1}\right)^2;
$$

then by using the fact that $t-t_\ell\le\tau_\ell$, $\eta_g(s)$ can be treat as constant in the integration, and define $C_y:=\mathrm{e}^{-C_g\tau_\ell\eta_g(t)}$, the bound can be further simplified as $\left(1-(\mathrm{e}^{-C_g\int_{t_\ell}^t \eta_g(s)\mathrm{d}s})^{-1}\right)^2\le\left(1-\frac{1}{C_y}\right)^2$; in (iii) we choose $\beta=\frac{C_y}{1-C_y}$.

Using the system dynamics (2.16), we have

$$
R\cdot\mathbf{y}(t)=R\cdot\mathbf{y}(t_\ell)-\left(\int_{t_\ell}^t \eta_\ell(s)\mathrm{d}s\right)R\hat{u}_{\ell,y}(t). \qquad (A.13)
$$

Then we can bound the last term of (A.9) by:

$$
\begin{aligned}
&\|\mathbf{y}(t)-\mathbf{y}(t_\ell)\|^2+\|\mathbf{z}(t)-\mathbf{z}(t_\ell)\|^2 \\
&\overset{(i)}{=}\|(I-R)\cdot(\mathbf{y}(t)-\mathbf{y}(t_\ell))\|^2+\|R\cdot(\mathbf{y}(t)-\mathbf{y}(t_\ell))\|^2+\left\|\int_{t_\ell}^t \eta_\ell(s)\mathrm{d}s\right\|^2\cdot\|\hat{u}_{\ell,z}(t)\|^2 \\
&\overset{(A.12),(A.13)}{\le}\left(\frac{1-C_y}{C_y^2}\right)\cdot\|(I-R)\cdot\mathbf{y}(t)\|^2+\left(\frac{\tau_\ell^2\eta_\ell^2(t)}{C_y}\right)\cdot\|(I-R)\cdot\hat{u}_{\ell,y}(t)\|^2 \\
&\qquad +\left\|\int_{t_\ell}^t \eta_\ell(s)\mathrm{d}s\right\|^2\|R\hat{u}_{\ell,y}(t)\|^2+\left\|\int_{t_\ell}^t \eta_\ell(s)\mathrm{d}s\right\|^2\cdot\|\hat{u}_{\ell,z}(t)\|^2
\end{aligned}
$$

$$
\begin{aligned}
&\overset{(ii)}{\leq} \left(\frac{1-C_y}{C_y^2}\right)\cdot\|(I-R)\cdot\mathbf{y}(t)\|^2+\frac{(\tau_\ell\eta_\ell(t))^2}{\min\{C_y,1\}}\left(\|\hat{u}_{\ell,y}(t)\|^2+\|\hat{u}_{\ell,z}(t)\|^2\right)\\
&\overset{(iii)}{\leq} \left(\frac{1-C_y}{C_y^2}\right)\cdot\|(I-R)\cdot\mathbf{y}(t)\|^2+2C_\ell^2\left(\|u_\ell(t)-\hat{u}_\ell(t)\|^2+\|u_\ell(t)\|^2\right)\\
&\overset{(iv)}{\leq} \left(\frac{1-C_y}{C_y^2}\right)\cdot\|(I-R)\cdot\mathbf{y}(t)\|^2+2L^2C_\ell^2\left(\|\mathbf{y}(t)-\mathbf{y}(t_\ell)\|^2+\|\mathbf{z}(t)-\mathbf{z}(t_\ell)\|^2\right)\\
&\quad+4C_\ell^2\cdot(C_x^2+C_v^2+C_z^2)\cdot(\|\nabla f(\bar{\mathbf{x}}(t))\|^2+\|\nabla f(\mathbf{x}(t))-\nabla f(\bar{\mathbf{x}}(t))\|^2)\\
&\overset{(v)}{\leq} \frac{\left(\frac{1-C_y}{C_y^2}\right)+4L_f^2C_\ell^2C_f}{1-2L^2C_\ell^2}\|(I-R)\cdot\mathbf{y}(t)\|^2+\frac{4C_\ell^2C_f}{1-2L^2C_\ell^2}\|\nabla f(\bar{\mathbf{x}}(t))\|^2,
\end{aligned}
\tag{A.14}
$$

where in (i) we separate $\mathbf{y}(t)-\mathbf{y}(t_\ell)$ into $R\cdot(\mathbf{y}(t)-\mathbf{y}(t_\ell))+(I-R)\cdot(\mathbf{y}(t)-\mathbf{y}(t_\ell))$, expand the square, and use the fact that $R\cdot(I-R)=0$; in (ii) we bound the integration interval in the last two terms with $t-t_\ell\leq\tau_\ell$, using the fact that $\eta_\ell(s)$ is treated as constant in the integration, and combine the last three terms; in (iii) we add and subtract $u_\ell(t)$ to the last term and apply the Cauchy–Schwarz inequality and further define $C_\ell:=\frac{\tau_\ell\eta_\ell(t)}{\min\{C_y,1\}}$; in (iv) we apply P3 and P4 to the last two terms and define

$$
C_f:=C_x^2+C_v^2+C_z^2; \tag{A.15}
$$

in (v) we apply A2 to the last term and move $\|\mathbf{y}(t)-\mathbf{y}(t_\ell)\|^2+\|\mathbf{z}(t)-\mathbf{z}(t_\ell)\|^2$ to the left and divide both sides by $1-2L^2C_\ell^2$ (note that this operation is legitimate since we have chosen $\tau_\ell\leq\frac{1+2C_g\eta_g(t)}{2L\eta_\ell(t)}$ such that $2L^2C_\ell^2<1$).

Substitute to $\hat{\mathcal{E}}$ in (A.4), we have:

$$
\int_0^t\dot{\mathcal{E}}(\tau)\mathrm{d}\tau\leq\int_0^t\left(-\left(\frac{\gamma_1(\tau)}{2}-C_{21}\right)\|\nabla f(\bar{\mathbf{x}}(\tau))\|^2-\left(\frac{\gamma_2(\tau)}{2}-C_{22}\right)\|(I-R)\cdot\mathbf{y}(\tau)\|^2\right)\mathrm{d}\tau,
$$

where $C_{21}:=\frac{4L^2C_\ell^2\eta_\ell^2(\tau)\cdot C_f}{2(1-2L^2C_\ell^2)\cdot\min\{N\gamma_1(\tau),\gamma_2(\tau)\}}$ and $C_{22}:=\frac{L^2\eta_\ell^2(\tau)\cdot\left(\left(\frac{1-C_y}{C_y^2}\right)+4L_f^2C_\ell^2C_f\right)}{2(1-2L^2C_\ell^2)\cdot\min\{N\gamma_1(\tau),\gamma_2(\tau)\}}$.

A.2 Proof for Lemma 3

In Case III-IV, we have $\tau_g=Q\tau_\ell$. Also note that t_g, t_ℓ were defined at the beginning of Appendix A.1. The update of the states can be written as:

$$
\begin{aligned}
\mathbf{y}(t_g+(q+1)\tau_\ell)&=\mathbf{y}(t_g+q\tau_\ell)-\int_{t_g+q\tau_\ell}^{t_g+(q+1)\tau_\ell}\eta_g(s)\hat{u}_g(s)+\eta_\ell(s)\hat{u}_{\ell,y}(s)\mathrm{d}s,\\
\mathbf{z}(t_g+(q+1)\tau_\ell)&=\mathbf{z}(t_g+q\tau_\ell)-\int_{t_g+q\tau_\ell}^{t_g+(q+1)\tau_\ell}\eta_\ell(s)\hat{u}_{\ell,z}(s)\mathrm{d}s.
\end{aligned}
\tag{A.16}
$$

Using the decomposition $\mathcal{E}(t) = \text{term I} + \hat{\mathcal{E}}(t)$, one can express, and subsequently bound the sampling error as:

$$\begin{aligned}
\hat{\mathcal{E}}(t) &= \left\langle \nabla f(\bar{\mathbf{x}}(t)), \frac{\eta_\ell(t)}{N} \mathbb{1}^T \cdot (u_{\ell,x}(t) - \hat{u}_{\ell,x}(t)) \right\rangle + \langle (I-R)\cdot \mathbf{y}(t), \eta_g(t)\cdot(u_g(t) - \hat{u}_g(t))\rangle \\
&\quad + \langle (I-R)\cdot \mathbf{y}(t), \eta_\ell(t)\cdot(u_{\ell,y}(t) - \hat{u}_{\ell,y}(t))\rangle \\
&\overset{(A.1)}{\leq} \frac{\gamma_1(t)}{2}\left\|\nabla f(\bar{\mathbf{x}}(t))\right\|^2 + \frac{\gamma_2(t)}{2}\left\|(I-R)\cdot\mathbf{y}(t)\right\|^2 \\
&\quad + \frac{\eta_g^2(t)}{2\gamma_2(t)}\left\|(I-R)\cdot(u_g(t)-\hat{u}_g(t))\right\|^2 + \frac{\eta_\ell^2(t)}{2\min\{N\gamma_1(t),\gamma_2(t)\}}\left\|u_{\ell,y}(t)-\hat{u}_{\ell,y}(t)\right\|^2 \\
&\overset{(i)}{\leq} \frac{\gamma_1(t)}{2}\left\|\nabla f(\bar{\mathbf{x}}(t))\right\|^2 + \frac{\gamma_2(t)}{2}\left\|(I-R)\cdot\mathbf{y}(t)\right\|^2 + \frac{\eta_g^2(t)}{2\gamma_2(t)}\left\|(I-R)\cdot(\mathbf{y}(t)-\mathbf{y}(t_g))\right\|^2 \\
&\quad + \frac{L^2\eta_\ell^2(t)}{2\min\{N\gamma_1(t),\gamma_2(t)\}}\left(\left\|\mathbf{y}(t)-\mathbf{y}(t_\ell)\right\|^2 + \left\|\mathbf{z}(t)-\mathbf{z}(t_\ell)\right\|^2\right), \qquad \text{(A.17)}
\end{aligned}$$

where in (i) we apply P2 and (2.4) to the third term, such that $\left\|(I-R)\cdot(u_g(t)-\hat{u}_g(t))\right\|^2 = \left\|(I-R)\cdot W_A(\mathbf{y}(t)-\mathbf{y}(t_g))\right\|^2 \leq \left\|(I-R)\cdot(\mathbf{y}(t)-\mathbf{y}(t_g))\right\|^2$, and we have used P3 to the last term. The key is to bound the last three terms of (A.17). We divide it into three steps.

Step 1) We bound the third term involving $\left\|(I-R)\cdot(\mathbf{y}(t)-\mathbf{y}(t_g))\right\|^2$. With (A.2), we have $\left\|(I-R)\cdot(\mathbf{y}(t)-\mathbf{y}(t_g))\right\|^2 \leq \left\|\mathbf{y}(t)-\mathbf{y}(t_g)\right\|^2$, then we bound the RHS by:

$$\begin{aligned}
\left\|\mathbf{y}(t)-\mathbf{y}(t_g)\right\|^2 &\overset{(i)}{=} \left\|(I-R)\cdot\int_{\tau_g}^{t}\eta_g(s)\hat{u}_g(s)\mathrm{d}s + \int_{t_g}^{t}\eta_\ell(s)\hat{u}_{\ell,y}(s)\mathrm{d}s\right\|^2 \\
&\overset{(ii)}{\leq} 2\tau_g^2\eta_g^2(t)\left\|\hat{u}_g(t)\right\|^2 + 2\left\|\int_{t_g}^{t}\eta_\ell(s)\hat{u}_{\ell,y}(s)\mathrm{d}s\right\|^2 \\
&\overset{(iii)}{\leq} 4\tau_g^2\eta_g^2(t)\left(\left\|\hat{u}_g(t)-u_g(t)\right\|^2 + \left\|u_g(t)\right\|^2\right) + 2\tau_\ell^2\sum_{\tau=t_g}^{t_\ell}\eta_\ell^2(\tau)\left\|\hat{u}_{\ell,y}(\tau)\right\|^2 \\
&\overset{(iv)}{\leq} 4\tau_g^2\eta_g^2(t)\left(\left\|\mathbf{y}(t)-\mathbf{y}(t_g)\right\|^2 + \left\|(I-R)\cdot\mathbf{y}(t)\right\|^2\right) + 2\tau_\ell^2\sum_{\tau=t_g}^{t_\ell}\eta_\ell^2(\tau)\left\|\hat{u}_{\ell,y}(\tau)\right\|^2 \\
&\overset{(v)}{\leq} \frac{4\tau_g^2\eta_g^2(t)}{1-4\tau_g^2\eta_g^2(t)}\left\|(I-R)\cdot\mathbf{y}(t)\right\|^2 + \frac{2\tau_\ell^2}{1-4\tau_g^2\eta_g^2(t)}\sum_{\tau=t_g}^{t_\ell}\eta_\ell^2(\tau)\left\|\hat{u}_{\ell,y}(\tau)\right\|^2, \qquad \text{(A.18)}
\end{aligned}$$

where (i) uses the first relation in (A.16), and $R\cdot\hat{u}_g(t) = 0$ (see P1); in (ii) we apply Cauchy-Schwarz inequality and use the fact that $t - t_\ell \leq \tau_g$ and $\hat{u}_g(s), \eta_g(s)$ remain constants in the integration; in (iii) we add and subtract $u_g(t)$ in the first term and applied Cauchy-Schwarz inequality, and (A.2); in (iv) we apply P2 to the first term and get $\hat{u}_g(t) - u_g(t) = G_g(\mathbf{y}(t) - \mathbf{y}(t_g); A)$, and apply the second inequality in (2.4), and the last inequality in (A.2); (v) holds because we moved $\left\|\mathbf{y}(t)-\mathbf{y}(t_g)\right\|^2$ to the left and divide both sides by $1-4\tau_g^2\eta_g^2(t)$, and choose

$\tau_g < \frac{1}{2\eta_g(t)}$ such that $4\tau_g^2\eta_g^2(t) < 1$. To bound the last term of (A.18), we note that following series of relations:

$$
\begin{aligned}
&\|\hat{u}_{\ell,y}(\tau)\|^2 \le \|\hat{u}_\ell(\tau)\|^2 \le 2\|\hat{u}_\ell(\tau) - u_\ell(\tau)\|^2 + 2\|u_\ell(\tau)\|^2 \qquad \text{(A.19)}\\
&\overset{(P3)}{\le} 2L^2 \cdot \left(\|\mathbf{y}(\tau) - \mathbf{y}(t_\ell)\|^2 + \|\mathbf{z}(\tau) - \mathbf{z}(t_\ell)\|^2\right) + 2\|u_\ell(\tau)\|^2\\
&\overset{(P4)}{\le} 2L^2 \cdot \left(\|\mathbf{y}(\tau) - \mathbf{y}(t_\ell)\|^2 + \|\mathbf{z}(\tau) - \mathbf{z}(t_\ell)\|^2\right) + 2C_f\|\nabla f(\mathbf{x}(\tau))\|^2\\
&\le 2L^2 \cdot \left(\|\mathbf{y}(\tau) - \mathbf{y}(t_\ell)\|^2 + \|\mathbf{z}(\tau) - \mathbf{z}(t_\ell)\|^2\right)\\
&\quad + 4C_f\left(\|\nabla f(\mathbf{x}(\tau)) - \nabla f(\bar{\mathbf{x}}(\tau))\|^2 + \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2\right)\\
&\overset{(A2)}{\le} 2L^2 \cdot \left(\|\mathbf{y}(\tau) - \mathbf{y}(t_\ell)\|^2 + \|\mathbf{z}(\tau) - \mathbf{z}(t_\ell)\|^2\right)\\
&\quad + 4C_f\left(L_f^2\|(I-R)\cdot\mathbf{x}(\tau))\|^2 + \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2\right),
\end{aligned}
$$

where C_f is defined in (A.15). Note that we need to further bound $\|\mathbf{y}(\tau) - \mathbf{y}(t_\ell)\|^2 + \|\mathbf{z}(\tau) - \mathbf{z}(t_\ell)\|^2$, which is the same to the last two terms in (A.16).

Step 2. We then bound $\|\mathbf{y}(t) - \mathbf{y}(t_\ell)\|^2 + \|\mathbf{z}(t) - \mathbf{z}(t_\ell)\|^2$. By (A.16), we have:

$$
\begin{aligned}
&\|\mathbf{y}(t) - \mathbf{y}(t_\ell)\|^2 + \|\mathbf{z}(t) - \mathbf{z}(t_\ell)\|^2 \overset{\text{(A.16)}}{=} \left\|\int_{t_\ell}^{t} \eta_g(s)\hat{u}_g(s) + \eta_\ell(s)\cdot\hat{u}_\ell(s)\mathrm{d}s\right\|^2 \qquad \text{(A.20)}\\
&\overset{(i)}{\le} 2\tau_\ell^2\eta_g^2(t)\|\hat{u}_g(t)\|^2 + 2\tau_\ell^2\eta_\ell^2(t)\cdot\|\hat{u}_\ell(t)\|^2\\
&\overset{\text{(A.19)}}{\le} 2\tau_\ell^2\eta_g^2(t)\|\hat{u}_g(t)\|^2 + 4L^2\tau_\ell^2\eta_\ell^2(t)\cdot\left(\|\mathbf{y}(t) - \mathbf{y}(t_\ell)\|^2 + \|\mathbf{z}(t) - \mathbf{z}(t_\ell)\|^2\right)\\
&\quad + 8L^2C_f\tau_\ell^2\eta_\ell^2(t)\cdot\left(\|\nabla f(\bar{\mathbf{x}}(t))\|^2 + L_f^2\|(I-R)\cdot\mathbf{x}(t)\|^2\right)\\
&\overset{(ii)}{\le} \frac{4\tau_\ell^2\eta_g^2(t)}{1-4L^2\tau_\ell^2\eta_\ell^2(t)}\left(\|u_g(t) - \hat{u}_g(t)\|^2 + \|u_g(t)\|^2\right)\\
&\quad + \frac{8L^2C_f\tau_\ell^2\eta_\ell^2(t)}{1-4L^2\tau_\ell^2\eta_\ell^2(t)}\cdot\left(\|\nabla f(\bar{\mathbf{x}}(t))\|^2 + L_f^2\|(I-R)\cdot\mathbf{x}(t)\|^2\right)\\
&\overset{(iii)}{\le} \frac{4\tau_\ell^2\eta_g^2(t)}{1-4L^2\tau_\ell^2\eta_\ell^2(t)}\left(\|\mathbf{y}(t) - \mathbf{y}(t_g)\|^2 + \|(I-R)\cdot\mathbf{y}(t)\|^2\right)\\
&\quad + \frac{8L^2C_f\tau_\ell^2\eta_\ell^2(t)}{1-4L^2\tau_\ell^2\eta_\ell^2(t)}\cdot\left(\|\nabla f(\bar{\mathbf{x}}(t))\|^2 + L_f^2\|(I-R)\cdot\mathbf{x}(t)\|^2\right),
\end{aligned}
$$

where in (i) we apply Cauchy-Schwarz inequality; in (ii) add and subtract $u_g(t)$ to the first term and move $\|\mathbf{y}(t) - \mathbf{y}(t_\ell)\|^2 + \|\mathbf{z}(t) - \mathbf{z}(t_\ell)\|^2$ to the left and divide both sides by $1 - 4L^2\tau_\ell^2\eta_\ell^2(t)$, and choose $\tau_\ell < \frac{1}{2L\eta_\ell(t)}$ such that $4L^2\tau_\ell^2\eta_\ell^2(t) < 1$; in (iii) we apply the second inequality in (2.4), as well as the fact that $\|I - R\| \le 1$.

To proceed, let us define $C_{43} := \frac{4\tau_g^2\eta_g^2(t)}{1-4\tau_g^2\eta_g^2(t)}$, $C_{44} := \frac{2\tau_\ell^2\eta_\ell^2(t)}{1-4\tau_g^2\eta_g^2(t)}$, $C_{45} := \frac{4\tau_\ell^2\eta_g^2(t)}{1-4L^2\tau_\ell^2\eta_\ell^2(t)}$, $C_{46} :=$

$\frac{8L^2C_f\tau_\ell^2\eta_\ell^2(t)}{1-4L^2\tau_\ell^2\eta_\ell^2(t)}$. Then by plug (A.18) into (A.20), we have:

$$\begin{aligned}
\|\mathbf{y}(t)-\mathbf{y}(t_\ell)\|^2+\|\mathbf{z}(t)-\mathbf{z}(t_\ell)\|^2 &\overset{(i)}{\leq} \left(C_{45}+C_{43}C_{45}+C_{46}L_f^2\right)\cdot\|(I-R)\cdot\mathbf{y}(t)\|^2 \\
&+C_{46}\|\nabla f(\bar{\mathbf{x}}(t))\|^2+QC_{44}C_{45}\cdot\sum_{\tau=t_g}^{t_\ell}\|\hat{u}_{\ell,y}(\tau)\|^2 \\
&\overset{(ii)}{\leq} \left(C_{45}+C_{43}C_{45}+C_{46}L_f^2\right)\cdot\|(I-R)\cdot\mathbf{y}(t)\|^2+C_{46}\|\nabla f(\bar{\mathbf{x}}(t))\|^2 \\
&+QC_{44}C_{45}\cdot\sum_{\tau=t_g}^{t_\ell}(C_x^2+C_v^2)\cdot\left(\|\nabla f(\mathbf{x}(\tau))-\nabla f(\bar{\mathbf{x}}(\tau))\|^2+\|\nabla f(\bar{\mathbf{x}}(\tau))\|^2\right) \\
&\overset{(A2)}{\leq} \left(C_{45}+C_{43}C_{45}+C_{46}L_f^2\right)\cdot\|(I-R)\cdot\mathbf{y}(t)\|^2+C_{46}\|\nabla f(\bar{\mathbf{x}}(t))\|^2 \\
&+QC_{44}C_{45}\cdot\sum_{\tau=t_g}^{t_\ell}(C_x^2+C_v^2)\cdot\left(L_f^2\|(I-R)\cdot\mathbf{x}(\tau)\|^2+\|\nabla f(\bar{\mathbf{x}}(\tau))\|^2\right),
\end{aligned} \tag{A.21}$$

where in (i) we use the fact that $t-t_g\leq Q\tau_\ell$; in (ii) we first apply P4 to the last term, then subtract $\nabla f(\bar{\mathbf{x}}(\tau))$, and finally used Cauchy-Schwartz inequality. This completes Part II of the proof.

Step 3. Finally, we substitute (A.21) into Part I (A.19) then to (A.18), we obtain:

$$\begin{aligned}
\|\mathbf{y}(t)-\mathbf{y}(t_g)\|^2 &\overset{\text{(A.19)}}{\leq} 4C_fC_{44}\sum_{\tau=t_g}^{t}\left(L_f^2\|(I-R)\cdot\mathbf{x}(\tau))\|^2+\|\nabla f(\bar{\mathbf{x}}(\tau))\|^2\right) \\
&+C_{43}\|(I-R)\cdot\mathbf{y}(t)\|^2+2L^2C_{44}\sum_{\tau=t_g}^{t}\left(\|\mathbf{y}(\tau)-\mathbf{y}(t_\ell)\|^2+\|\mathbf{z}(\tau)-\mathbf{z}(t_\ell)\|^2\right) \\
&\overset{\text{(A.21)}}{\leq} C_{43}\|(I-R)\cdot\mathbf{y}(t)\|^2+4C_fC_{44}\sum_{\tau=t_g}^{t}\left(L_f^2\|(I-R)\cdot\mathbf{x}(\tau))\|^2+\|\nabla f(\bar{\mathbf{x}}(\tau))\|^2\right) \\
&+2L^2C_{44}\sum_{\tau=t_g}^{t}C_{46}\|\nabla f(\bar{\mathbf{x}}(\tau))\|^2 \\
&+2L^2C_{44}^2C_{45}\cdot\sum_{\tau=t_g}^{t}\sum_{\tau_1=t_g}^{\tau}(C_x^2+C_v^2)\cdot\left(L_f^2\|(I-R)\cdot\mathbf{x}(\tau_1)\|^2+\|\nabla f(\bar{\mathbf{x}}(\tau_1))\|^2\right).
\end{aligned}$$

Then we substitute (A.18) and (A.21) to (A.17) then to (A.4), we obtain:

$$\begin{aligned}
\int_0^t\dot{\mathcal{E}}(\tau)\mathrm{d}\tau \leq &-\int_0^t(\frac{\gamma_1(\tau)}{2}-C_{41}(\tau))\cdot\|\nabla f(\bar{\mathbf{x}}(\tau))\|^2\,\mathrm{d}\tau \\
&-\int_0^t(\frac{\gamma_2(\tau)}{2}-C_{42}(\tau))\cdot\|(I-R)\cdot\mathbf{y}(\tau)\|^2\,\mathrm{d}\tau,
\end{aligned}$$

where we have defined

$$C_{41} := \frac{L^2\eta_\ell^2(\tau)\cdot\left(C_{45}\cdot(1+L_f^2C_{47}+C_{45})+C_{46}L_f^2\right)}{2\min\{N\gamma_1(\tau),\gamma_2(\tau)\}}+\frac{C_g\eta_g^2(\tau)\cdot(C_{43}+L_f^2C_{47})}{2\gamma_2(\tau)},$$
$$C_{42} := \frac{L^2\eta_\ell^2(\tau)\cdot(C_{46}+C_{45}C_{47})}{2\min\{N\gamma_1(\tau),\gamma_2(\tau)\}}+\frac{C_g\eta_g^2(\tau)C_{47}}{2\gamma_2(\tau)},\ \text{ and }\ C_{47} := Q^2C_{44}^2\cdot(C_x^2+C_v^2).$$

A.3 Distributed Algorithms as Discretized Multi-Rate Systems

In this section, we provide additional discussions on how to map the distributed algorithms to the discretized multi-rate systems. First, let us discuss decentralized algorithms.

DGD [7]: The updates are given by (where $c>0$ is the stepsize):

$$\mathbf{x}(k+1) = W\mathbf{x}(k) - c\nabla f(\mathbf{x}(k)) = \mathbf{x}(k) - ((I-W)\mathbf{x}(k)+c)\cdot\nabla f(\mathbf{x}(k)).$$

It uses the discretization Case III, with the following continuous-time controllers:

$$u_{g,x} = (I-W)\cdot\mathbf{x},\quad u_{\ell,x} = \nabla f(\mathbf{x}).$$

DLM [9]: The updates are given by:

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{x}(k) - \eta\cdot\left(\nabla f(\mathbf{x}(k)) + c\cdot(I-W)\cdot\mathbf{x}(k)+\mathbf{v}(k)\right),\\ \mathbf{v}(k+1) &= \mathbf{v}(k) + c\cdot(I-W)\cdot\mathbf{x}(k+1).\end{aligned}$$

It corresponds to Case III, with the following continuous-time controllers:

$$u_{g,x} = c\cdot(I-W)\cdot\mathbf{x}+\mathbf{v},\quad u_{g,v} = (I-W)\cdot\mathbf{x},\quad u_{\ell,x} = \nabla f(\mathbf{x}),\quad u_{\ell,v} = 0.$$

Next, we discuss some popular federated learning algorithms. For this class of algorithms, the agents are connected with a central server which performs averaging. The corresponding communication graph is a fully connected graph, with the weight matrix being the averaging matrix, i.e., $W = R, W_A = I-R$.

FedProx [16]: The updates are given by (where GD is used to solve local problems):

$$\mathbf{x}(k+1) = \begin{cases}\mathbf{x}(k)-\eta_1\nabla f(\mathbf{x}(k))-\eta_2(\mathbf{x}(k)-\mathbf{x}(k_0)),\ k \bmod Q \neq 0,\ k_0 = k-(k \bmod Q),\\ R\mathbf{x}(k)-\eta_1\nabla f(\mathbf{x}(k))-\eta_2\cdot(\mathbf{x}(k)-\mathbf{x}(k_0)), k \bmod Q = 0.\end{cases}$$

It uses the discretization Case I, with the following continuous-time controllers:

$$u_{g,x} = (I-R)\cdot\mathbf{x},\quad u_{\ell,x} = \nabla f(\mathbf{x}).$$

FedPD [18]: The updates are given by (where GD is used to solve local problems):

$$\mathbf{x}(k+1) = \mathbf{x}(k) - \eta_1 \cdot (\nabla f(\mathbf{x}(k)) + \mathbf{v}(k) + \eta_2 \cdot (\mathbf{x}(k_0) - R\mathbf{x}(k_0))),\ k_0 = k - (k \bmod Q),$$

$$\mathbf{w}(k+1) = \begin{cases} R\mathbf{x}(k), k \bmod Q = 0 \\ \mathbf{w}(k), k \bmod Q \neq 0, \end{cases}$$

$$\mathbf{v}(k+1) = \begin{cases} \mathbf{v}(k) + \frac{1}{\eta_2} \cdot (\mathbf{x}(k) - \mathbf{w}(k)), k \bmod Q = 0 \\ \mathbf{v}(k), k \bmod Q \neq 0. \end{cases}$$

It uses the discretization Case I or IV. Observe that $\mathbf{w}$ tracks $R\mathbf{x}$. Replace $\mathbf{w}$ with $R\mathbf{x}$, we can obtain the following controller:

$$u_{g,x} = (I - R) \cdot \mathbf{x} + \mathbf{v}, \quad u_{g,v} = -(I - R) \cdot \mathbf{x}, \quad u_{\ell,x} = \nabla f(\mathbf{x}), \quad u_{\ell,v} = 0.$$

Finally, we discuss one more rate optimal algorithm:

D-GPDA [20]: The update step of Distributed Gradient Primal-Dual Algorithm (D-GPDA) is given by:

$$\begin{aligned} \mathbf{x}(k+1) = \arg\min_{\mathbf{x}} &\left\langle \nabla f(\mathbf{x}(k)) + A^T \mathbf{v}(k), \mathbf{x} - \mathbf{x}(k) \right\rangle \\ &+ \frac{1}{2} \left\| \eta_1 A \mathbf{x} \right\|^2 + \left\| \eta_1 |A| \cdot (\mathbf{x} - \mathbf{x}(k)) \right\|^2 + \left\| \eta_2 \cdot (\mathbf{x} - \mathbf{x}(k)) \right\|^2 \\ \mathbf{v}(k+1) = \mathbf{v}(k) &+ \eta_1^2 A \mathbf{x}(k+1), \end{aligned}$$

where $\mathbf{v}$ is the dual variable for the linear consensus constraint. By assuming the minimization is solved with gradient flow or K-step gradient descent, this algorithm is using the discretization Case II, with the following continuous-time controllers:

$$\begin{aligned} u_{g,x} &= \eta_1 W \mathbf{x} + \eta_2 \cdot (\mathbf{x} - \mathbf{v}_2) - \eta_1 \left| A^T A \right| \mathbf{v}_2 + A^T \mathbf{v}_1, \quad u_{g,v} = [-\eta_1^2 A \mathbf{x};\ 0], \\ u_{\ell,x} &= \nabla f(\mathbf{x}), \quad u_{\ell,v} = [0;\ -(\mathbf{x} - \mathbf{v}_2)]. \end{aligned}$$

A.4 Proofs for Section 2.3

In this section, we provide the proofs for (2.5), (2.6) and Corollary 1 in Section 2.3.

A.4.1 Proof of (2.5)

From P1, we show that the time derivative of the consensus error is strictly negative:

$$\begin{aligned} \frac{\partial}{\partial t} \left\| (I - R) \cdot \mathbf{y}(t) \right\|^2 = 2 \left\langle (I - R) \cdot \mathbf{y}(t), \dot{\mathbf{y}}(t) \right\rangle &\overset{(i)}{=} -2 \left\langle (I - R) \cdot \mathbf{y}(t), u_g(t) \right\rangle \\ &\overset{(ii)}{\leq} -2C_g \left\| (I - R) \cdot \mathbf{y}(t) \right\|^2, \end{aligned}$$

where in (i) we apply (2.8) and substitute $\eta_g(t) = 1, \eta_\ell(t) = 0$ and in (ii) we apply P1.

By applying Gronwall's inequality, we have

$$\begin{aligned}\|(I-R)\cdot\mathbf{y}(t+\tau)\|^2 &\leq \exp\left\{\int_t^{t+\tau} -2C_g \mathrm{d}\tau_1\right\}\|(I-R)\cdot\mathbf{y}(t)\|^2 \\ &= \exp\left\{-2C_g\tau\right\}\|(I-R)\cdot\mathbf{y}(t)\|^2,\end{aligned}$$

which completes the proof of (2.5).

A.4.2 Proof of (2.6)

From P4, we show that the time derivatives of the local functions are strictly negative:

$$\begin{aligned}\frac{\partial}{\partial t}f_i(x_i(t)) = \langle\nabla f_i(x_i(t)), \dot{x}_i(t)\rangle &\overset{(i)}{=} -\langle\nabla f_i(x_i(t)), u_{i,\ell,x}(t)\rangle \\ &\overset{(ii)}{\leq} -\alpha(t)\cdot\|\nabla f_i(x_i(t))\|^2.\end{aligned}$$

where in (i) we apply (2.8) and substitute $\eta_g(t) = 0, \eta_\ell(t) = 1$; in (ii) we apply P4. Integrate it over time we have:

$$\int_0^t \beta(\tau,t)\cdot\|\nabla f_i(x_i(\tau))\|^2\,\mathrm{d}\tau \leq \frac{1}{\int_0^t\alpha(\tau)\mathrm{d}\tau}\left(f_i(x_i(0)) - f_i(x_i(t))\right), \tag{A.22}$$

$$\min_{\tau\in[0,t]}\|\nabla f_i(x_i(\tau))\|^2\,\mathrm{d}\tau \leq \frac{1}{\int_0^t\alpha(\tau)\mathrm{d}\tau}\left(f_i(x_i(0)) - f_i(x_i(t))\right), \tag{A.23}$$

where in (A.22), $\beta(\tau,t) = \frac{\alpha(\tau)}{\int_0^t\alpha(\tau)\mathrm{d}\tau}$ defines a distribution over time $[0,t]$ and the LHS is the expected value of $\|\nabla f_i(x_i(\tau))\|^2$; in (A.23) we use the fact that $\mathbb{E}_t[X(t)] \geq \min_t\{X(t)\}$ for an arbitrary random variable $X(t)$. This completes the proof of (2.6).

A.4.3 Proof of Corollary 1

In this part, we prove the convergence of the system under P1, P3, P4. First, we compute the derivative of $\mathcal{E}$, then we break it down into three terms. By bounding each term, we obtain P5. From Theorem 1, we perform integration over time, then we have the final convergence result.

The time derivative of $\mathcal{E}$ can be bounded by

$$\begin{aligned}\dot{\mathcal{E}}(t) &= \left\langle\nabla f(\bar{\mathbf{x}}(t)), \frac{1}{N}\sum_{i=1}^N\dot{x}_i(t)\right\rangle + \langle(I-R)\cdot\mathbf{y}(t), \dot{\mathbf{y}}(t)\rangle \\ &\overset{(i)}{=} -\left\langle\nabla f(\bar{\mathbf{x}}(t)), \frac{1}{N}\sum_{i=1}^N\eta_\ell(t)\cdot u_{i,\ell,x}(t) + \eta_g(t)\cdot\frac{\mathbb{1}^T}{N}u_{g,x}(t)\right\rangle \\ &\quad -\langle(I-R)\cdot\mathbf{y}(t), \eta_g(t)\cdot u_g(t) + \eta_\ell(t)\cdot u_{\ell,y}(t)\rangle\end{aligned}$$

$$
\begin{aligned}
&\overset{(P1)}{\leq} -C_g\eta_g(t)\left\|(I-R)\cdot\mathbf{y}(t)\right\|^2-\eta_\ell(t)\left\langle\nabla f(\bar{\mathbf{x}}(t)),\frac{1}{N}\sum_{i=1}^{N}u_{i,\ell,x}(t)\right\rangle\\
&\quad-\eta_\ell(t)\left\langle(I-R)\cdot\mathbf{y}(t),u_{\ell,y}(t)\right\rangle\\
&\overset{(A.2)}{=} -C_g\eta_g(t)\left\|(I-R)\cdot\mathbf{y}(t)\right\|^2-\eta_\ell(t)\left\langle(I-R)\cdot\mathbf{y}(t),(I-R)\cdot u_{\ell,y}(t)\right\rangle\\
&\quad-\eta_\ell(t)\left\langle\nabla f(\bar{\mathbf{x}}(t)),\frac{1}{N}\sum_{i=1}^{N}u_{i,\ell,x}(t)+c\nabla f(\bar{\mathbf{x}}(t))-c\nabla f(\bar{\mathbf{x}}(t))\right\rangle\\
&\overset{(ii)}{\leq} -C_g\eta_g(t)\left\|(I-R)\cdot\mathbf{y}(t)\right\|^2-\eta_\ell(t)\cdot c\left\|\nabla f(\bar{\mathbf{x}}(t))\right\|^2+\frac{\beta_1(t)}{2}\left\|(I-R)\cdot\mathbf{y}(t)\right\|^2\\
&\quad+\frac{\eta_\ell^2(t)}{2\beta_1(t)}\left\|(I-R)\cdot u_{\ell,y}(t)\right\|^2+\frac{\beta_2(t)}{2}\left\|\nabla f(\bar{\mathbf{x}}(t))\right\|^2+\frac{\eta_\ell^2(t)}{2\beta_2(t)}\left\|\frac{1}{N}\sum_{i=1}^{N}u_{i,\ell,x}(t)-c\nabla f(\bar{\mathbf{x}}(t))\right\|^2\\
&=-\left(C_g\eta_g(t)-\frac{\beta_1(t)}{2}\right)\cdot\left\|(I-R)\cdot\mathbf{y}(t)\right\|^2-(c\eta_\ell(t)-\beta_2(t)/2)\cdot\left\|\nabla f(\bar{\mathbf{x}}(t))\right\|^2\\
&\quad+\frac{\eta_\ell^2(t)}{2\beta_1(t)}\left\|(I-R)\cdot u_{\ell,y}(t)\right\|^2+\frac{\eta_\ell^2(t)}{2\beta_2(t)}\left\|\frac{1}{N}\sum_{i=1}^{N}(u_{i,\ell,x}(t)-c\nabla f_i(\bar{\mathbf{x}}(t)))\right\|^2,
\end{aligned}
\tag{A.24}
$$

where in (i) we substitute the system dynamics (2.8), and $u_g(t):=[u_{g,x}(t);u_{g,v}(t)]$; in (ii) we apply (A.1). Then, we bound the last two terms of (A.24) separately. We have:

$$
\begin{aligned}
\left\|(I-R)\cdot u_{\ell,y}(t)\right\|^2&=\sum_{i=1}^{N}\left\|u_{i,\ell,y}(t)-\frac{1}{N}\sum_{j=1}^{N}u_{j,\ell,y}(t)\right\|^2\leq\frac{N-1}{N}\sum_{i\neq j}^{N}\left\|u_{i,\ell,y}(t)-u_{j,\ell,y}(t)\right\|^2\\
&\leq\frac{4(N-1)}{N}\sum_{i=1}^{N}\left\|u_{i,\ell,y}(t)\right\|^2\overset{(P4)}{\leq}\frac{4(N-1)\cdot(C_x^2+C_v^2)}{N}\left\|\nabla f(\mathbf{x}(t))\right\|^2.
\end{aligned}
$$

Also we have:

$$
\begin{aligned}
&\left\|\frac{1}{N}\sum_{i=1}^{N}(u_{i,\ell,x}(t)-c\nabla f_i(\bar{\mathbf{x}}(t)))\right\|^2\\
&=\left\|\frac{1}{N}\sum_{i=1}^{N}(u_{i,\ell,x}(t)-c\nabla f_i(x_i(t))+c\nabla f_i(x_i(t))-c\nabla f_i(\bar{\mathbf{x}}(t)))\right\|^2\\
&\leq\frac{2}{N}\sum_{i=1}^{N}\left(\left\|u_{i,\ell,x}(t)-c\nabla f_i(x_i(t))\right\|^2+c^2\left\|\nabla f_i(x_i(t))-\nabla f_i(\bar{\mathbf{x}}(t))\right\|^2\right)\\
&\overset{(i)}{\leq}\frac{2}{N}\sum_{i=1}^{N}(\left\|u_{i,\ell,x}(t)\right\|^2+c^2\left\|\nabla f_i(x_i(t))\right\|^2-2c\left\langle u_{i,\ell,x}(t),\nabla f_i(x_i(t))\right\rangle+c^2L_f^2\left\|x_i(t)-\bar{\mathbf{x}}(t)\right\|^2)\\
&\overset{(ii)}{\leq}\frac{2(C_x^2+c^2-2c\alpha(t))}{N}\left\|\nabla f(\mathbf{x}(t))\right\|^2+\frac{2c^2L_f^2}{N}\left\|(I-R)\cdot\mathbf{x}(t)\right\|^2,
\end{aligned}
$$

where (i) we expand the first term and apply A2 to the second term; in (ii) we use P4 for the

first three terms and plug the definition of $I - R$ into the last term. Further, we have:

$$\begin{aligned}\|\nabla f(\mathbf{x}(t))\|^2 &= \sum_{i=1}^{N} \|\nabla f_i(x_i(t)) - \nabla f_i(\bar{\mathbf{x}}(t)) + \nabla f_i(\bar{\mathbf{x}}(t))\|^2 \\ &\overset{(A.1)}{\leq} 2\sum_{i=1}^{N} \left(\|\nabla f_i(x_i(t)) - \nabla f_i(\bar{\mathbf{x}}(t))\|^2 + \|\nabla f_i(\bar{\mathbf{x}}(t))\|^2\right) \\ &\overset{(P2)}{\leq} 2L_f \|(I-R)\cdot \mathbf{x}(t)\|^2 + 2\sum_{i=1}^{N} \|\nabla f_i(\bar{\mathbf{x}}(t))\|^2 .\end{aligned}$$

Substitute back to (A.24), we have

$$\begin{aligned}\dot{\mathcal{E}}(t) \leq &- \left(C_g\eta_g(t) - \frac{\beta_1(t)}{2} - \eta_\ell^2(t)\cdot\left(\frac{c^2L_f^2}{N\beta_2(t)} + C_{df}L_f\right)\right)\cdot\|(I-R)\cdot\mathbf{y}(t)\|^2 \\ &- \frac{2c\eta_\ell(t) - \beta_2(t))}{2}\|\nabla f(\bar{\mathbf{x}}(t))\|^2 + \eta_\ell^2(t)\cdot C_{df}\sum_{i=1}^{N}\|\nabla f_i(\bar{\mathbf{x}}(t))\|^2 ,\end{aligned} \tag{A.25}$$

where $C_{df} := \left(\frac{4(N-1)\cdot(C_x^2+C_v^2)}{N\beta_1(t)} + \frac{2(C_x^2+c^2-2c\alpha(t))}{N\beta_2(t)}\right)$. We analyze the convergence rate in two cases: i) $C_{df} \leq 0$, and ii) $C_{df} > 0$.

Case i: If $C_{df} \leq 0$, which implies $\alpha(t) > C_x$. Then, by choosing $\beta_1(t) \leq \frac{C_g\eta_g(t)}{4}$, $\beta_2(t) \leq c\eta_\ell(t)$, $\eta_\ell(t) \leq \frac{NC_g\eta_g(t)}{4cL_f^2}$, we have:

$$\dot{\mathcal{E}}(t) \leq -\frac{C_g\eta_g(t)}{2}\cdot\|(I-R)\cdot\mathbf{y}(t)\|^2 - \frac{c\eta_\ell(t)}{2}\cdot\|\nabla f(\bar{\mathbf{x}}(t))\|^2 .$$

In this case, by choosing $\eta_g(t) = 1, \eta_\ell(t) = \frac{NC_g\eta_g(t)}{4cL_f^2}, c = \alpha(t) > C_x$, then P5 satisfies with $\gamma_1(t) = \frac{NC_g}{4L_f^2}, \gamma_2(t) = \frac{C_g}{2}$, and

$$\min_\tau\{\|(I-R)\cdot\mathbf{y}(\tau)\|^2 + \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2\} = \mathcal{O}(1/t).$$

Case ii: If $C_{df} > 0$, we show that by choosing $\eta_\ell(t) = \Theta(\|(I-R)\cdot\mathbf{y}(\tau)\|^2 + \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2)$, $\eta_g(t) = \mathcal{O}(1)$, $\min_\tau\{\|(I-R)\cdot\mathbf{y}(\tau)\|^2 + \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2]\} = \mathcal{O}(1/\sqrt{t})$ is satisfied. We proceed by bounding $\sum_{i=1}^N \|\nabla f_i(\bar{\mathbf{x}}(t))\|^2$ in (A.25). First, we define the level set $\mathbf{S}(t) := \{x \mid f(x) \leq \mathcal{E}(t) + f^\star\}$. By A4, we can define the upper bound of $\sum_{i=1}^N \|\nabla f_i(\bar{\mathbf{x}}(t))\|^2$ as

$$D(t) := \sup_{x\in\mathbf{S}(t)}\left\{\sum_{i=1}^{N}\|\nabla f_i(x)\|^2\right\}.$$

Then, to guarantee that

$$D(\tau) \leq \frac{C_g\eta_g(\tau)}{4C_{df}\eta_\ell^2(\tau)}\cdot\|(I-R)\cdot\mathbf{y}(\tau)\|^2 + \frac{c}{4C_{df}\eta_\ell(\tau)}\|\nabla f(\bar{\mathbf{x}}(\tau))\|^2 , \ \forall\tau\in[0,t],$$

we can solve for $\beta_1(\tau)$, $\beta_2(\tau)$ and $\eta_\ell(\tau)$, which result in the following three relations:

$$\beta_1(\tau) \le \frac{C_g \eta_g(\tau)}{2}, \quad \beta_2(\tau) \le c \cdot \eta_\ell(\tau),$$
$$\eta_\ell(\tau) \le \max\left\{ \frac{\sqrt{C_g \eta_g(\tau) C_{df} L_f}\, \|(I-R)\cdot \mathbf{y}(\tau)\|^2}{4C_{df} D(\tau) + 2C_{df} L_f \|(I-R)\cdot \mathbf{y}(\tau)\|^2}, \frac{c\,\|\nabla f(\bar{\mathbf{x}}(\tau))\|^2}{4C_{df} D(\tau)} \right\}.$$

These choices of parameters guarantee that

$$\eta_\ell^2(\tau) \cdot C_{df} \sum_{i=1}^{N} \|\nabla f_i(\bar{\mathbf{x}}(\tau))\|^2 \le \eta_\ell^2(\tau) \cdot C_{df} D(\tau)$$
$$\le \frac{C_g \eta_g(\tau)}{4} \cdot \|(I-R)\cdot \mathbf{y}(\tau)\|^2 + \frac{c\eta_\ell(\tau)}{4} \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2, \ \forall \tau \in [0,t]. \tag{A.26}$$

Substituting (A.26) to (A.25), we have:

$$\dot{\mathcal{E}}(\tau) \le -\frac{C_g \eta_g(\tau)}{4} \cdot \|(I-R)\cdot \mathbf{y}(\tau)\|^2 - \frac{c\eta_\ell(\tau)}{4} \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2 < 0, \ \forall \tau \in [0,t].$$

Integrating over time, it gives $\mathcal{E}(t) = \mathcal{E}(0) + \int_0^t \dot{\mathcal{E}}(s)\mathrm{d}s \le \mathcal{E}(0)$. Therefore, $\mathbf{S}(\tau) := \{\mathbf{x} \mid f(\mathbf{x}) \le \mathcal{E}(\tau) + f^\star\} \subseteq \mathbf{S}(0)$, $D(\tau) \le D(0)$, $\forall \tau \in [0,t]$. So we can choose the parameters as:

$$\eta_g(\tau) = 1, \ c = \frac{1}{2}, \ \beta_1(\tau) = \frac{C_g}{4}, \ \beta_2(\tau) = \frac{\eta_\ell(\tau)}{2}$$
$$\eta_\ell(\tau) = \max\left\{ \frac{\sqrt{C_g C_{df} L_f}\, \|(I-R)\cdot \mathbf{y}(\tau)\|^2}{4C_{df} D(0) + 2C_{df} L_f \|(I-R)\cdot \mathbf{y}(\tau)\|^2}, \frac{\|\nabla f(\bar{\mathbf{x}}(\tau))\|^2}{8C_{df} D(0)} \right\}$$
$$= \Theta\left(\|(I-R)\cdot \mathbf{y}(\tau)\|^2 + \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2 \right), \ \forall \tau \in [0,t].$$

Based on the above choices of parameters, we will show below that the convergence rate of the system is $\mathcal{O}(1/\sqrt{t})$. If $\min_{\tau\in[0,t]} \|(I-R)\cdot \mathbf{y}(\tau)\|^2 + \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2 = \mathcal{O}(\frac{1}{\sqrt{t}})$, then the result is achieved. Otherwise we have:

$$\|(I-R)\cdot \mathbf{y}(\tau)\|^2 + \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2 = \Omega\left(\frac{1}{\sqrt{t}}\right), \ \forall \tau \in [0,t]. \tag{A.27}$$

This will guarantee that $\eta_\ell(\tau) = \Theta(\frac{1}{\sqrt{t}})$, $\forall \tau \in [0,t]$ and $\gamma_1(\tau) = \frac{\eta_\ell(\tau)}{4} = \Theta(\frac{1}{\sqrt{t}})$, $\gamma_2(\tau) = \frac{C_g}{4} = \mathcal{O}(1)$, $\forall \tau \in [0,t]$ for P5. Then we apply Theorem 1 and obtain that

$$\min_\tau \{\|(I-R)\cdot \mathbf{y}(\tau)\|^2 + \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2\} = \max\{\mathcal{O}(1/t), \mathcal{O}(1/\sqrt{t})\} = \mathcal{O}(1/\sqrt{t}).$$

Summarizing the above two cases, we have the worst convergence rate for the algorithm as: $\max\{\mathcal{O}(1/t), \mathcal{O}(1/\sqrt{t})\} = \mathcal{O}(1/\sqrt{t})$. This completes the proof for Corollary 1.

A.5 Verify Property P5 for DGT Algorithm

Recall that the derivative of the energy function is given by:

$$\begin{aligned}\dot{\mathcal{E}}(t) &= -\left\langle \nabla f(\bar{\mathbf{x}}(t)), \frac{1}{N}\sum_{i=1}^{N} u_{\ell,x}(t) \right\rangle - \langle (I-R)\cdot \mathbf{y}(t), u_{g,y}(t) + u_{\ell,y}(t)\rangle \\ &\overset{(2.23)}{=} -\langle \nabla f(\bar{\mathbf{x}}(t)), c\bar{\mathbf{v}}(t)\rangle - \langle (I-R)\cdot \mathbf{y}(t), (I-W)\cdot \mathbf{y}(t)\rangle \\ &\quad - \langle (I-R)\cdot \mathbf{x}(t), c\mathbf{v}(t)\rangle + \langle (I-R)\cdot \mathbf{v}(t), \nabla f(\mathbf{x}(t)) - \nabla f(\mathbf{z}(t))\rangle\,.\end{aligned} \tag{A.28}$$

Then we bound each term on the RHS above separtately.

To bound the first term, note that:

$$\begin{aligned}\frac{c}{2}\left\|\nabla f(\bar{\mathbf{x}}(t)) - \bar{\mathbf{v}}(t)\right\|^2 &= \frac{c}{2}\left\|\nabla f(\bar{\mathbf{x}}(t)) - \frac{1}{N}\sum_{i=1}^{N}\nabla f(x_i(t)) + \frac{1}{N}\sum_{i=1}^{N}\nabla f(x_i(t)) - \bar{\mathbf{v}}(t)\right\|^2 \\ &\overset{(i)}{\le} c\left(\frac{1}{N}\sum_{i=1}^{N}\left\|\nabla f(\bar{\mathbf{x}}(t)) - \nabla f(x_i(t))\right\|^2 + \left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f(x_i(t)) - \bar{\mathbf{v}}(t)\right\|^2\right) \\ &\overset{(ii)}{\le} c\left(\frac{L_f}{N}\sum_{i=1}^{N}\left\|\bar{\mathbf{x}}(t) - x_i(t)\right\|^2 + \left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f(x_i(t)) - \bar{\mathbf{v}}(t)\right\|^2\right) \\ &\overset{(iii)}{\le} c\left(\frac{L_f}{N}\left\|(I-R)\cdot\mathbf{x}(t)\right\|^2 + \left\|\frac{\mathbb{1}^T}{N}\nabla f(\mathbf{x}(t)) - \bar{\mathbf{v}}(t)\right\|^2\right),\end{aligned}$$

where in (i) we apply (A.1) and Jensen's inequality; in (ii) we apply A2; in (iii) we substitute the definition of R. From (2.30), $\bar{\mathbf{v}}(t) = \frac{\mathbb{1}^T}{N}\nabla f(\mathbf{x}(t))$, and we have

$$\frac{c}{2}\left\|\nabla f(\bar{\mathbf{x}}(t)) - \bar{\mathbf{v}}(t)\right\|^2 \le c\frac{L_f}{N}\left\|(I-R)\cdot\mathbf{x}(t)\right\|^2.$$

So the first term in (2.31) can be bounded as

$$\begin{aligned}-\langle \nabla f(\bar{\mathbf{x}}(t)), c\bar{\mathbf{v}}(t)\rangle &= -\frac{c}{2}\left(\left\|\nabla f(\bar{\mathbf{x}}(t))\right\|^2 + \left\|\bar{\mathbf{v}}(t)\right\|^2 - \left\|\nabla f(\bar{\mathbf{x}}(t)) - \bar{\mathbf{v}}(t)\right\|^2\right) \\ &\le -\frac{c}{2}\left(\left\|\nabla f(\bar{\mathbf{x}}(t))\right\|^2 + \left\|\bar{\mathbf{v}}(t)\right\|^2 - \frac{2L_f}{N}\left\|(I-R)\cdot\mathbf{x}(t)\right\|^2\right).\end{aligned} \tag{A.29}$$

The second term in (2.31) can be bounded by directly applying P1. That is, we have:

$$-\langle (I-R)\cdot\mathbf{y}(t), (I-W)\cdot\mathbf{y}(t)\rangle \le -C_g\left\|(I-R)\cdot\mathbf{y}(t)\right\|^2.$$

Next, the third term in (2.31) can be bounded as:

$$\begin{aligned}&-c\,\langle (I-R)\cdot\mathbf{x}(t), \mathbf{v}(t)\rangle \overset{(A.2)}{=} -c\,\langle (I-R)\cdot\mathbf{x}(t), (I-R)\cdot\mathbf{v}(t)\rangle \\ &\overset{(A.1)}{\le} \frac{c}{2}\cdot\left(\left\|(I-R)\cdot\mathbf{x}(t)\right\|^2 + \left\|(I-R)\cdot\mathbf{v}(t)\right\|^2\right) = \frac{c}{2}\left\|(I-R)\cdot\mathbf{y}(t)\right\|^2.\end{aligned}$$

Finally, we bound the last term in (2.31) by:

$$\begin{aligned}
&\langle (I-R)\cdot \mathbf{v}(t), \nabla f(\mathbf{x}(t)) - \nabla f(\mathbf{z}(t))\rangle \overset{(A.2)}{=} \langle (I-R)\cdot \mathbf{v}(t), (I-R)\cdot(\nabla f(\mathbf{x}(t)) - \nabla f(\mathbf{z}(t)))\rangle \\
&\overset{(A.1)}{\leq} \frac{\beta}{2}\left\|(I-R)\cdot \mathbf{v}(t)\right\|^2 + \frac{1}{2\beta}\left\|(I-R)\cdot(\nabla f(\mathbf{x}(t)) - \nabla f(\mathbf{z}(t)))\right\|^2 \\
&\overset{(i)}{=} \frac{\beta}{2}\left\|(I-R)\cdot \mathbf{v}(t)\right\|^2 + \frac{1}{2\beta N}\sum_{i=1}^N \left\| \frac{\mathbb{1}^T}{N}\nabla f(\mathbf{x}(t)) - \nabla f_i(x_i(t)) - \frac{\mathbb{1}^T}{N}\nabla f(\mathbf{z}(t)) + \nabla f_i(z_i(t))\right\|^2,
\end{aligned}$$

where (i) is due to $R := \frac{1}{N}\mathbb{1}\mathbb{1}^T$. The last term above can be further bounded by:

$$\begin{aligned}
&\frac{1}{2\beta N}\sum_{i=1}^N \left\| \frac{\mathbb{1}^T}{N}\nabla f(\mathbf{x}(t)) - \nabla f_i(x_i(t)) - \frac{\mathbb{1}^T}{N}\nabla f(\mathbf{z}(t)) + \nabla f_i(z_i(t))\right\|^2 \\
&\overset{(i)}{=} \frac{1}{2\beta N}\sum_{i=1}^N \left\| \left(\frac{\mathbb{1}^T}{N}\nabla f(\mathbf{x}(t)) - \nabla f(\bar{\mathbf{x}}(t))\right) + \left(\nabla f(\bar{\mathbf{x}}(t)) - \frac{\mathbb{1}^T}{N}\nabla f(\mathbf{z}(t))\right) - (\nabla f_i(x_i(t)) - \nabla f_i(z_i(t)))\right\|^2 \\
&\leq \frac{2}{\beta N}\sum_{i=1}^N \left(\left\| \frac{\mathbb{1}^T}{N}\nabla f(\mathbf{x}(t)) - \nabla f(\bar{\mathbf{x}}(t))\right\|^2 + \left\|\nabla f(\bar{\mathbf{x}}(t)) - \frac{\mathbb{1}^T}{N}\nabla f(\mathbf{z}(t))\right\|^2 + \|\nabla f_i(x_i(t)) - \nabla f_i(z_i(t))\|^2\right) \\
&\overset{(ii)}{\leq} \frac{2L_f}{\beta}(\|(I-R)\cdot \mathbf{x}(t)\|^2 + \|\bar{\mathbf{x}}(t) - \mathbf{z}(t)\|^2 + \|\mathbf{x}(t) - \mathbf{z}(t)\|^2) \\
&= \frac{2L_f}{\beta}(\|(I-R)\cdot \mathbf{x}(t)\|^2 + \|\bar{\mathbf{x}}(t) - \bar{\mathbf{z}}(t) + \bar{\mathbf{z}}(t) - \mathbf{z}(t)\|^2 + \|\mathbf{x}(t) - \bar{\mathbf{x}}(t) + \bar{\mathbf{x}}(t) - \bar{\mathbf{z}}(t) + \bar{\mathbf{z}}(t) - \mathbf{z}(t)\|^2) \\
&\leq \frac{8L_f}{\beta}(\|(I-R)\cdot \mathbf{x}(t)\|^2 + \|\bar{\mathbf{x}}(t) - \bar{\mathbf{z}}(t)\|^2 + \|(I-R)\cdot \mathbf{z}(t)\|^2),
\end{aligned}$$

where in (i) we add and substracts $\nabla f(\bar{\mathbf{x}}(t))$; in (ii) we apply A2.

Finally, we analyze $\|\bar{\mathbf{x}}(t) - \bar{\mathbf{z}}(t)\|^2$:

$$\begin{aligned}
\|\bar{\mathbf{x}}(t) - \bar{\mathbf{z}}(t)\|^2 &\overset{(2.29)}{=} \left\| \frac{\mathbb{1}^T}{N}\int_0^t ((I-W)\cdot \mathbf{x}(\tau) - c\mathbf{v}(\tau))\,\mathrm{e}^{-(t-\tau)}\mathrm{d}\tau\right\|^2 \\
&\overset{(i)}{=} c^2\left\|\int_0^t \bar{\mathbf{v}}(\tau)\mathrm{e}^{-(t-\tau)}\mathrm{d}\tau\right\|^2 \overset{(ii)}{\leq} c^2\int_0^t e^{-(t-\tau)}\mathrm{d}\tau\cdot\int_0^t \|\bar{\mathbf{v}}(\tau)\|^2\,\mathrm{e}^{-(t-\tau)}\mathrm{d}\tau \\
&\leq c^2\int_0^t \|\bar{\mathbf{v}}(\tau)\|^2\,\mathrm{e}^{-(t-\tau)}\mathrm{d}\tau,
\end{aligned}$$

where in (i) we apply (2.4), that $\mathbb{1}^T W_A = 0$, and in this case $W_A = (I-W)$; in (ii) we use Cauchy–Schwarz inequality to break the integration.

Plugging in the above into (2.23), the final bound we have is:

$$\begin{aligned}
\dot{\mathcal{E}}(t) \leq &-\frac{c}{2}\|\nabla f(\bar{\mathbf{x}}(t))\|^2 - \frac{c}{2}\|\bar{\mathbf{v}}(t)\|^2 + \frac{8L_f c^2}{\beta}\int_0^t \|\bar{\mathbf{v}}(\tau)\|^2\,\mathrm{e}^{-(t-\tau)}\mathrm{d}\tau \\
&- \left(C_g - \frac{c + 2cL_f/N + \beta + 16cL_f/\beta}{2}\right)\cdot\|(I-R)\cdot \mathbf{y}(t)\|^2.
\end{aligned} \tag{A.30}$$

Integrating the above relation over time, we have:

$$\int_0^t \dot{\mathcal{E}}(\tau)\mathrm{d}\tau \leq -\frac{c}{2}\int_0^t \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2\,\mathrm{d}\tau + \frac{8L_f c^2}{\beta}\int_0^t\int_0^\tau \|\bar{\mathbf{v}}(\tau_1)\|^2\,\mathrm{e}^{-(\tau-\tau_1)}\mathrm{d}\tau_1\mathrm{d}\tau$$

$$
\begin{aligned}
&- \frac{c}{2}\int_0^t \|\bar{\mathbf{v}}(\tau)\|^2 \,\mathrm{d}\tau - \left(C_g - \frac{c + 2cL_f + \beta + 16cL_f/\beta}{2}\right) \cdot \int_0^t \|(I-R)\mathbf{y}(\tau)\|^2 \,\mathrm{d}\tau \\
\overset{(i)}{=} &- \frac{c}{2}\int_0^t \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2 \,\mathrm{d}\tau + \frac{8L_f c^2}{\beta}\int_0^t \left(\|\bar{\mathbf{v}}(\tau_1)\|^2 \int_{\tau_1}^t \mathrm{e}^{-(\tau-\tau_1)}\mathrm{d}\tau\right)\mathrm{d}\tau_1 \\
&- \frac{c}{2}\int_0^t \|\bar{\mathbf{v}}(\tau)\|^2 \,\mathrm{d}\tau - \left(C_g - \frac{c + 2cL_f + \beta + 16cL_f/\beta}{2}\right) \cdot \int_0^t \|(I-R)\cdot\mathbf{y}(\tau)\|^2 \,\mathrm{d}\tau \\
\overset{(ii)}{\leq} &- \frac{c}{2}\int_0^t \|\nabla f(\bar{\mathbf{x}}(\tau))\|^2 \,\mathrm{d}\tau - \frac{c - 8L_f c^2/\beta}{2}\int_0^t \|\bar{\mathbf{v}}(\tau)\|^2 \,\mathrm{d}\tau \\
&- (C_g - \frac{c + 2cL_f + \beta + 16cL_f/\beta}{2}) \cdot \int_0^t \|(I-R)\cdot\mathbf{y}(\tau)\|^2 \,\mathrm{d}\tau,
\end{aligned}
$$

where in (i) we switch the order of integration; in (ii) we apply that $\int_{\tau_1}^t \mathrm{e}^{-(t-\tau)}\mathrm{d}\tau \leq 1$.

Appendix B

Additional Results and Proofs of Chapter 3

In this section, we provide additional discussions missing in the main body of Chapter 3.

B.1 Related Works in Dynamic Systems

In this subsection, we provide additional discussion on existing works, which are related to using control theory, and dynamic system to analyze distributed algorithms.

Controlling the stochastic system using robust control has been a standard approach in the control theory [129]. More recent works such as [130] generalizes the small gain theorem to nonlinear control systems to analyze the system stability with stochasticity. Distributed control system has been studied for optimizing global performance in distributed energy resources applications [131]. Research has shown that centralized and decentralized deterministic optimization algorithms [4, 36, 34] can be analyzed as dynamic systems. However, these works are restricted to convex optimization with deterministic controllers in continuous time, and fail to capture the impact of the "multi-rate" discretization, thus cannot cover the FL and algorithms that performs multiple consensus steps [19, 26, 79].

From the continuous-time perspective, there are a series of related researches focus on both gradient and stochastic gradient flow algorithms. The convergence rate of the non-convex stochastic gradient flow algorithm has been studied in [71] as the continuous-time counterpart of stochastic gradient descent algorithm in centralized setting. Some recent works focus on analyzing the stochastic gradient Langevin dynamics [132, 133] which are closely related to the stochastic gradient descent algorithms in both centralized and distributed settings. However,

they are hard to be generalized to other stochastic algorithms.

B.2 Algorithm Discussion

In this part, we provide some concrete examples on how the existing algorithms are covered by the proposed model.

First we start with the ZONE algorithm [29] in decentralized training setting:

The update steps of ZONE are:

$$\begin{aligned}\mathbf{x}^{r+1} &= \mathbf{x}^r - \rho \cdot (\mathbf{v}^r + W\mathbf{x}^r) - \eta'_\ell \tilde{\nabla} f(\mathbf{x}^r)\\ \mathbf{v}^{r+1} &= \mathbf{v}^r + W\mathbf{x}^{r+1},\end{aligned}$$

where $W = A^T A$, $\tilde{\nabla} f(\mathbf{x}^r)$ is the stochastic zeroth-order estimation of $\nabla f(\mathbf{x}^r)$. It is easy to see that the corresponding continuous-time deterministic controllers are:

$$\begin{aligned}u_g(t) &= \begin{bmatrix} \rho W & \rho I \\ -W & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{v}(t) \end{bmatrix},\\ u_{i,\ell}(t) &= \begin{bmatrix} \nabla f(\mathbf{x}(t)) \\ 0 \end{bmatrix}.\end{aligned}$$

ZONE corresponds to discretization Case I with $\tau_g = \tau_\ell = 1$, and has zeroth-order gradient as stochastic LCFL.

Second, we provide the mapping for FedPD [18] and FedDyn [81] in the federated learning setting where the communication graph can be viewed as a complete graph, and $W_A = I - R$:

The update of FedPD is given by [18]:

$$\begin{aligned}\mathbf{x}^{r,q+1} &= \mathbf{x}^{r,q} - \eta'_\ell \tilde{\nabla} f(\mathbf{x}^{r,q}; \xi^{r,q})\\ &\quad + \eta'_g \cdot (\rho \cdot (\mathbf{x}^{r,q} - \mathbf{w}^{r,q}) + \mathbf{v}^{r,0})\\ \mathbf{v}^{r,q+1} &= \begin{cases} \mathbf{v}^{r,q} + \eta'_g \cdot (\mathbf{x}^{r,q+1} - \mathbf{w}^{r,q}), & (q+1) = Q \\ \mathbf{v}^{r,q}, & (q+1) \neq Q, \end{cases}\\ \mathbf{w}^{r,q+1} &= \begin{cases} \frac{\eta'_g}{p} R \cdot (2\mathbf{x}^{r,q+1} - \mathbf{w}^{r,q}), & (q+1) = Q, \text{ w.p. } p \\ 2\mathbf{x}^{r,q+1} - \mathbf{w}^{r,q}, & (q+1) = Q, \text{ w.p. } 1-p \\ \mathbf{w}^{r,q}, & (q+1) \neq Q, \end{cases}\end{aligned}$$

where $\tilde{\nabla} f(\mathbf{x}^{r,q}; \xi^{r,q})$ denotes the stochastic gradient estimated on samples $\xi^{r,q}$. Observe that $\mathbf{w}$ tracks $R\mathbf{x}$ and update with probability p, so in continuous time, we can replace $\mathbf{w}$ with $R\mathbf{x}$, and

obtain the following continuous-time controllers:

$$u_g(t) = \begin{bmatrix} \rho \cdot (I - R) & \rho I \\ -(I - R) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{v}(t) \end{bmatrix},$$
$$u_{i,\ell}(t) = \begin{bmatrix} \nabla f(\mathbf{x}(t)) \\ 0 \end{bmatrix}.$$

FedPD corresponds to discretization Case II with $\tau_g = Q\tau_\ell = 1$, and has both stochastic gradient as stochastic LCFL and random communication graph

$$\tilde{W}_A = \begin{cases} \begin{bmatrix} \rho \cdot (I - R/p) & \rho I \\ -(I - R/p) & 0 \end{bmatrix} & \text{w.p. } p, \\ \begin{bmatrix} \rho I & \rho I \\ -I & 0 \end{bmatrix} & \text{w.p. } 1 - p, \end{cases}$$

in the stochastic GCFL.

The update of FedDyn is given by [81]:

$$\begin{aligned} \mathbf{x}^{r,q+1} &= \mathbf{x}^{r,q} - \eta'_\ell \tilde{\nabla} f(\mathbf{x}^{r,q}; \xi^{r,q}) \\ &\quad + \eta'_g \cdot (\rho \cdot (\mathbf{x}^{r,q} - \mathbf{w}^{r,q}) + \mathbf{v}^{r,0}) \\ \mathbf{v}^{r,q+1} &= \begin{cases} \mathbf{v}^{r,q} + \eta'_g \cdot (\mathbf{x}^{r,q+1} - \mathbf{w}^{r,q}), & (q+1) = Q \\ \mathbf{v}^{r,q}, & (q+1) \neq Q, \end{cases} \\ \mathbf{w}^{r,q+1} &= \begin{cases} \tilde{R} \cdot (2\mathbf{x}^{r,q+1} - \mathbf{w}^{r,q}), & (q+1) = Q, \\ \mathbf{w}^{r,q}, & (q+1) \neq Q, \end{cases} \end{aligned}$$

where $\tilde{\nabla} f(\mathbf{x}^{r,q}; \xi^{r,q})$ denotes the stochastic gradient estimated on samples $\xi^{r,q}$, and $\tilde{R} := \frac{\mathbb{1}_N \mathbf{B}^T}{\mathbb{1}_N^T \mathbf{B}}$, $\mathbf{B} \in \{0,1\}^N$ is a random vector denotes the partial participation pattern with $\mathbb{E}[\tilde{R}] = R$. Observe that $\mathbf{w}$ tracks $R\mathbf{x}$ in expectation, so in continuous time we can replace $\mathbf{w}$ with $R\mathbf{x}$, and obtain the following deterministic controllers:

$$u_g(t) = \begin{bmatrix} \rho \cdot (I - R) & \rho I \\ -(I - R) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{v}(t) \end{bmatrix},$$
$$u_{i,\ell}(t) = \begin{bmatrix} \nabla f(\mathbf{x}(t)) \\ 0 \end{bmatrix}.$$

FedDyn corresponds to discretization Case II with $\tau_g = Q\tau_\ell = 1$, and has both stochastic gradient as stochastic LCFL and random communication graph

$$\tilde{W}_A = \begin{bmatrix} \rho \cdot (I - \tilde{R}) & \rho I \\ -(I - \tilde{R}) & 0 \end{bmatrix},$$

in the stochastic GCFL.

Lastly, we map the DSAGD algorithm [79] to our system:

The update of DSAGD is given by [79]:

$$\begin{aligned}
\mathbf{x}^{r,k+1} &= \begin{cases} \mathbf{x}^{r,k} - \eta'_\ell \cdot (\mathbf{x}^{r,k} - \mathbf{v}^{r,k+1}), & k+1 = K \\ \mathbf{x}^{r,k}, & k+1 \neq K \end{cases}, \\
\mathbf{v}^{r,k+1} &= \tilde{W}^{r,k} \cdot (\alpha^k \mathbf{x}^{r,0} + (1-\alpha^k) \cdot \mathbf{v}^{r,0}) \\
&\quad - \alpha^k \beta^r \tilde{\nabla} f(\mathbf{z}^{r,0}; \xi^r) \\
\mathbf{z}^{r,k+1} &= \begin{cases} \mathbf{z}^{r,k} - \eta'_\ell \cdot (\mathbf{x}^{r,k+1} - \mathbf{v}^{r,k+1}), & k+1 = K \\ \mathbf{z}^{r,k}, & k+1 \neq K \end{cases},
\end{aligned}$$

where $\tilde{\nabla} f(\mathbf{z}^{r,0}; \xi^r)$ denotes the stochastic gradient estimated on samples ξ^r, and $\tilde{W}^{r,k}$ are random mixing matrices. We can obtain the following deterministic controller:

$$\begin{aligned}
u_g(t) &= \begin{bmatrix} 0 & 0 \\ -\alpha(t) \cdot W & I - (1-\alpha(t)) \cdot W \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{v}(t) \end{bmatrix}, \\
u_{i,\ell}(t) &= \begin{bmatrix} \mathbf{x}(t) - \mathbf{v}(t) \\ \alpha(t) \cdot \beta(t) \cdot \nabla f(\mathbf{z}(t)) \\ \mathbf{x}(t) - \mathbf{v}(t) \end{bmatrix}.
\end{aligned}$$

DSAGD corresponds to discretization Case III with $\tau_\ell = K\tau_g > 0$, and has both stochastic gradient as stochastic LCFL and random communication graph

$$\tilde{W}_A = \begin{bmatrix} 0 & 0 \\ -\alpha^k \tilde{W}^{r,k} & (I - (1-\alpha^k) \cdot \tilde{W}^{r,k}) \end{bmatrix},$$

in the stochastic GCFL.

Algorithm connections: Interestingly, we can observe that ZONE, FedPD and FedDyn has almost the same deterministic continuous-time controllers, where the only difference is the the mixing matrix $W = R$ in FL. These three algorithms distinguish from each other by having different sampling rates and introducing different forms of stochasticities.

B.3 Detailed Discussions for Section 3.4

In this section, we provide the proof for the lemmas in Section 3.4. Before we start, let us introduce some useful relations:

$$\begin{aligned}\langle a,b\rangle &= \frac{1}{2\alpha}\left\|a\right\|^2 + \frac{\alpha}{2}\left\|b\right\|^2 - \frac{1}{2}\left\|\frac{1}{\sqrt{\alpha}}a + \sqrt{\alpha}b\right\|^2 \\ &\leq \frac{1}{2\alpha}\left\|a\right\|^2 + \frac{\alpha}{2}\left\|b\right\|^2 \end{aligned} \tag{B.1}$$

$$(I-R)^2 = I - 2R + R^2 = I - R. \tag{B.2}$$

B.3.1 Proofs for Case I

We first present the proof for Lemma 4 in Case I.

B.3.2 Case I: Lemma 4(A)

The proof for Lemma 4(A) is straightforward. We first write the difference between the consecutive energy functions as:

$$\mathbb{E}_r\left[\tilde{\mathcal{E}}^{r+1} - \tilde{\mathcal{E}}^{r}\right] = \underbrace{\mathbb{E}_r\left[\tilde{\mathcal{E}}^{r+1}\right] - \mathcal{E}^{r+1}}_{\Delta^{r+1}} + \underbrace{\mathcal{E}^{r+1} - \tilde{\mathcal{E}}^{r}}_{\text{term I}}, \tag{B.3}$$

where we can apply PD4 to bound the sum of term I. The main challenge is to bound Δ^{r+1}. This can be proceed by the following:

$$\begin{aligned}
\mathbb{E}_r[\tilde{\mathcal{E}}^{r+1} - \mathcal{E}^{r+1}] &= \mathbb{E}_r\left[f(\tilde{\bar{\mathbf{x}}}^{r+1}) - f(\bar{\mathbf{x}}^{r+1})\right] \\
&\quad + \mathbb{E}_r\left[\left\|(I-R)\cdot\tilde{\mathbf{y}}^{r+1}\right\|^2 - \left\|(I-R)\cdot\mathbf{y}^{r+1}\right\|^2\right] \\
&\overset{(i)}{\leq} \mathrm{Var}_r((I-R)\cdot\tilde{\mathbf{y}}^{r+1}) + \frac{L_f}{2}\,\mathbb{E}_r(\left\|\tilde{\bar{\mathbf{x}}}^{r+1} - \bar{\mathbf{x}}^{r+1}\right\|^2) \\
&\quad + \mathbb{E}_r\left[\left\langle\nabla f(\bar{\mathbf{x}}^{r+1}), \tilde{\bar{\mathbf{x}}}^{r+1} - \bar{\mathbf{x}}^{r+1}\right\rangle\right] \\
&\overset{(ii)}{\leq} \left(1 + \frac{L_f}{2N}\right)\cdot\mathrm{Var}_r(\tilde{\mathbf{y}}^{r+1}) \\
&\overset{(iii)}{=} \left(1 + \frac{L_f}{2N}\right)\cdot\mathrm{Var}_r(\eta_g'^r\tilde{u}_g^r + \eta_\ell'^r\tilde{u}_{\ell,y}^r) \\
&\overset{(PS3)}{=} \left(1 + \frac{L_f}{2N}\right) \\
&\quad \times \left((\eta_g'^r)^2\cdot\mathrm{Var}_r(\tilde{u}_g^r) + (\eta_\ell'^r)^2\cdot\mathrm{Var}_r(\tilde{u}_{\ell,y}^r)\right) \\
&\overset{(PS2)}{\leq} \left(1 + \frac{L_f}{2N}\right)\cdot(\eta_g'^r)^2\cdot\left(B_g\left\|u_g^r\right\|^2 + \sigma_g^2\right)
\end{aligned}$$

$$
\begin{aligned}
&+\left(1+\frac{L_f}{2N}\right)\cdot(\eta_\ell'^r)^2\cdot\left(B_\ell\left\|u_{\ell,y}^r\right\|^2+\sigma_\ell^2\right)\\
&\overset{(iv)}{\leq}\left(1+\frac{L_f}{2N}\right)\cdot(\eta_g'^r)^2\cdot(B_g\left\|(I-R)\cdot\tilde{\mathbf{y}}^r\right\|^2+\sigma_g^2)\\
&+\left(1+\frac{L_f}{2N}\right)\cdot(\eta_\ell'^r)^2\cdot\Bigg(B_\ell(C_x^2+C_v^2)\\
&\cdot\left(\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2+L_f^2\left\|(I-R)\cdot\tilde{\mathbf{x}}^r\right\|^2\right)+\sigma_\ell^2\Bigg),
\end{aligned}
$$

where in (i) we apply A6 to the first two terms; in (ii) we apply PS1(A) to the last term as $\mathbb{E}_r\,\tilde{\bar{\mathbf{x}}}^{r+1}=\bar{\mathbf{x}}^{r+1}$ and merge the other two terms by using the fact that $\left\|\tilde{\bar{\mathbf{x}}}^r-\bar{\mathbf{x}}^r\right\|^2\leq\frac{1}{N}\left\|\tilde{\mathbf{x}}^r-\mathbf{x}^r\right\|^2$ and $\mathbf{x}$ is a sub-vector of $\mathbf{y}$; in (iii) we apply the update steps of $\tilde{\mathbf{y}}^r$ in (3.8); in (iv) we apply PD1 to the first term and bound the third term by

$$
\begin{aligned}
&\left\|u_{\ell,y}^r\right\|^2\overset{(PD3)}{\leq}(C_x^2+C_v^2)\left\|\nabla f(\tilde{\mathbf{x}}^r)\right\|^2\\
&\leq 2(C_x^2+C_v^2)\cdot(\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2+\left\|\nabla f(\tilde{\mathbf{x}}^r)-\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2)\\
&\overset{(A6)}{\leq}2(C_x^2+C_v^2)\cdot(\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2+L_f^2\left\|\tilde{\mathbf{x}}^r-\tilde{\bar{\mathbf{x}}}^r\right\|^2\\
&=2(C_x^2+C_v^2)\cdot(\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2+L_f^2\left\|(I-R)\cdot\tilde{\mathbf{x}}^r\right\|^2)
\end{aligned}
\tag{B.4}
$$

Finally, substitute the above result into Δ^{r+1} in (B.3) and apply PD4, then Lemma 4(A) is proved.

B.3.3 Case I: Lemma 4(B)

In Lemma 4(B), the key step is to bound:

$$
\begin{aligned}
\Delta^{r+1}&=\mathbb{E}_r\left[\tilde{\mathcal{E}}^{r+1}\right]-\mathcal{E}^{r+1}\\
&=\underbrace{\mathbb{E}_r\left[\tilde{\mathcal{E}}^{r+1}\right]-f(\mathbb{E}_r\,\tilde{\bar{\mathbf{x}}}^{r+1})-\left\|(I-R)\cdot\mathbb{E}_r\,\tilde{\mathbf{y}}^r\right\|^2}_{\Delta_A}\\
&\quad+\underbrace{f(\mathbb{E}_r\,\tilde{\bar{\mathbf{x}}}^{r+1})+\left\|(I-R)\cdot\mathbb{E}_r\,\tilde{\mathbf{y}}^r\right\|^2-\mathcal{E}^{r+1}}_{\Delta_B}.
\end{aligned}
\tag{B.5}
$$

First, we bound Δ_A, which is the same as Δ^{r+1} in the previous case:

$$
\begin{aligned}
\Delta_A&\leq(1+\frac{L_f}{2N})\cdot\mathrm{Var}_r(\eta_\ell'^r\tilde{u}_\ell^r+\eta_g'^r\tilde{u}_g^r)\\
&\overset{(PS2)}{\leq}(1+\frac{L_f}{2N})\cdot(\eta_\ell'^r)^2(B_\ell\left\|\mathbb{E}_r\,\tilde{u}_\ell^r\right\|^2+\sigma_\ell^2)\\
&\quad+(1+\frac{L_f}{2N})\cdot(\eta_g'^r)^2(B_g\left\|\mathbb{E}_r\,\tilde{u}_g^r\right\|^2+\sigma_g^2)
\end{aligned}
$$

$$\overset{(i)}{\leq} (1+\frac{L_f}{2N})\cdot(\eta_\ell'^r)^2(B_\ell C_1+\sigma_\ell^2) + (1+\frac{L_f}{2N})\cdot(\eta_g'^r)^2(B_g\left\|(I-R)\cdot\tilde{\mathbf{y}}^r\right\|^2+\sigma_g^2),$$

where in (i) we apply PS1(B) to bound the first term and PD1 to bound $\left\|\mathbb{E}_r\,\tilde{u}_g^r\right\|^2$.

We then bound Δ_B by:

$$\begin{aligned}
\Delta_B &\overset{(A6)}{\leq} \left\langle \nabla f(\bar{\mathbf{x}}^{r+1}), \mathbb{E}_r\,\tilde{\bar{\mathbf{x}}}^{r+1}-\bar{\mathbf{x}}^{r+1}\right\rangle \\
&\quad+\frac{L_f}{2}\left\|\mathbb{E}_r\,\tilde{\bar{\mathbf{x}}}^{r+1}-\bar{\mathbf{x}}^{r+1}\right\|^2 \\
&\quad+2\left\langle (I-R)\cdot\mathbf{y}^{r+1},(I-R)\cdot(\mathbb{E}_r\,\tilde{\mathbf{y}}^{r+1}-\mathbf{y}^{r+1})\right\rangle \\
&\quad+\left\|(I-R)\cdot(\mathbb{E}_r\,\tilde{\mathbf{y}}^{r+1}-\mathbf{y}^{r+1})\right\|^2 \\
&\overset{(i)}{\leq} \left\langle \nabla f(\bar{\mathbf{x}}^{r+1})-\nabla f(\tilde{\bar{\mathbf{x}}}^r), \mathbb{E}_r\,\tilde{\bar{\mathbf{x}}}^{r+1}-\bar{\mathbf{x}}^{r+1}\right\rangle \\
&\quad+\left\langle \nabla f(\tilde{\bar{\mathbf{x}}}^r), \mathbb{E}_r\,\tilde{\bar{\mathbf{x}}}^{r+1}-\bar{\mathbf{x}}^{r+1}\right\rangle \\
&\quad+\left(1+\frac{L_f}{2N}\right)\left\|\mathbb{E}_r\,\tilde{\mathbf{y}}^{r+1}-\mathbf{y}^{r+1}\right\|^2 \\
&\quad+2\left\langle (I-R)\cdot(\mathbf{y}^{r+1}-\tilde{\mathbf{y}}^r),(I-R)\cdot(\mathbb{E}_r\,\tilde{\mathbf{y}}^{r+1}-\mathbf{y}^{r+1})\right\rangle \\
&\quad+2\left\langle (I-R)\cdot\tilde{\mathbf{y}}^r,(I-R)\cdot(\mathbb{E}_r\,\tilde{\mathbf{y}}^{r+1}-\mathbf{y}^{r+1})\right\rangle \\
&\overset{(B.1)}{\leq} \frac{\beta_2}{2}\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2+\frac{\beta_3}{2}\left\|(I-R)\cdot\tilde{\mathbf{y}}^r\right\|^2 \\
&\quad+\frac{\beta_1(2+L_f^2)}{2}\left\|\mathbf{y}^{r+1}-\tilde{\mathbf{y}}^r\right\|^2 \\
&\quad+\left(1+\frac{L_f}{2N}+\frac{1}{\beta_1}+\frac{\beta_2+\beta_3}{2\beta_2\beta_3}\right)\left\|\mathbb{E}_r\,\tilde{\mathbf{y}}^{r+1}-\mathbf{y}^{r+1}\right\|^2,
\end{aligned}$$

where in (i) we use the fact that $\mathbf{x}$ is a sub-vector of $\mathbf{y}$ and combine the two norms; add and subtract $\nabla f(\tilde{\bar{\mathbf{x}}}^r)$ to the first term, and add and subtract $(I-R)\cdot\tilde{\mathbf{y}}^r$ to the third term.

To bound the last two terms in the above relation, we have:

$$\begin{aligned}
&\left\|\mathbf{y}^{r+1}-\tilde{\mathbf{y}}^r\right\|^2=\left\|\eta_\ell'^r u_{\ell,y}^r+\eta_g'^r u_g^r\right\|^2 \\
&\leq 2(\eta_\ell'^r)^2\cdot(C_x^2+C_v^2)\cdot(\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2+L_f^2\left\|(I-R)\cdot\tilde{\mathbf{x}}^r\right\|^2) \\
&\quad+2(\eta_g'^r)^2\left\|(I-R)\cdot\tilde{\mathbf{y}}^r\right\|^2,
\end{aligned}$$

where we apply (B.4) to bound $u_{\ell,y}^r$ and PD1 to bound u_g^r. And we have

$$\begin{aligned}
&\left\|\mathbb{E}_r\,\tilde{\mathbf{y}}^{r+1}-\mathbf{y}^{r+1}\right\|^2 \\
&=\left\|\eta_\ell'^r(\mathbb{E}_r\,\tilde{u}_{\ell,y}^r-u_{\ell,y}^r)+\eta_g'^r(\mathbb{E}_r\,\tilde{u}_g^r-u_g^r)\right\|^2 \\
&\overset{(PS1)}{=}(\eta_\ell'^r)^2\cdot\left(\left\|\mathbb{E}_r\,\tilde{u}_{\ell,y}^r\right\|^2+\left\|u_{\ell,y}^r\right\|^2-2\left\langle\mathbb{E}_r\,\tilde{u}_{\ell,y}^r,u_{\ell,y}^r\right\rangle^2\right)
\end{aligned}$$

$$
\begin{aligned}
&\overset{(PS1)}{\leq} (\eta_\ell'^r)^2 \cdot \left(C_1 + (1-2C_2)\cdot \left\|u_{\ell,y}^r\right\|^2 + 2\sigma_G^2\right) \\
&\overset{(B.4)}{\leq} 2(\eta_\ell'^r)^2 \cdot (C_x^2 + C_v^2)\cdot(1-2C_2)\cdot \left\|\nabla f(\tilde{\mathbf{x}}^r)\right\|^2 \\
&\quad + 2L_f^2 \cdot (\eta_\ell'^r)^2 \cdot (C_x^2 + C_v^2)\cdot(1-2C_2)\cdot \|(I-R)\cdot\tilde{\mathbf{x}}^r\|^2 \\
&\quad + (\eta_\ell'^r)^2 \cdot (C_1 + 2\sigma_G^2).
\end{aligned}
$$

substitute the above results to (B.5), we have:

$$
\begin{aligned}
&\mathbb{E}_r\left[\tilde{\mathcal{E}}^{r+1}\right] - \mathcal{E}^{r+1} \\
&\leq C_{11}' \left\|\nabla f(\tilde{\mathbf{x}}^r)\right\|^2 + C_{12}' \|(I-R)\cdot\tilde{\mathbf{y}}^r\|^2 \\
&\quad + (1+\frac{L_f}{2N})\cdot(\eta_g'^r)^2\cdot\sigma_g^2 + (1+\frac{L_f}{2N})\cdot(\eta_\ell'^r)^2\cdot\sigma_\ell^2 \\
&\quad + (B_\ell(1+\frac{L_f}{2N}) + C_{17})\cdot(\eta_\ell'^r)^2\cdot C_1 + (\eta_\ell'^r)^2\cdot C_{17}\sigma_G^2,
\end{aligned}
$$

where we define the following constants

$$
\begin{aligned}
C_{11}' &:= \frac{\beta_2}{2} + \beta_1\cdot(2+L_f)\cdot C_{18} + 2C_{17}C_{18}\cdot(1-2C_2), \\
C_{12}' &:= (1+\frac{L_f}{2N})\cdot(\eta_g'^r)^2 B_g + 2L_f^2 C_{17}C_{18}\cdot(1-2C_2) \\
&\qquad + \frac{\beta_2}{2} + \beta_1\cdot(2+L_f)\cdot(C_{18}\cdot L_f^2 + (\eta_g'^r)^2), \\
C_{15} &:= C_{14}B_\ell + \sum_{r=0}^{t} C_{17}\cdot(\eta_\ell'^r)^2, \\
C_{16} &:= \sum_{r=0}^{t} C_{17}\cdot(\eta_\ell'^r)^2, \\
C_{17} &:= 1 + \frac{L_f}{2N} + \frac{1}{\beta_1} + \frac{\beta_2+\beta_3}{2\beta_2\beta_3}, \\
C_{18} &:= (\eta_\ell'^r)^2\cdot(C_x^2 + C_v^2).
\end{aligned}
$$

Plug it into (B.3), and apply PD4, then Lemma 4(B) is proved.

B.3.4 Case II and III

Case II: For Case II, $\tau_g = Q\tau_\ell > 0$. Let us denote the states at r^{th} global sampling time instance as $(\cdot)^r := (\cdot)(r\tau_g)$, where the q^{th} local sampling time between two consecutive global sampling instance as $(\cdot)^{r,q} := (\cdot)(r\tau_g + q\tau_\ell)$, then the system can be written as:

$$
\begin{aligned}
\mathbf{x}^{r,q+1} &= \mathbf{x}^{r,q} - \eta_\ell'^{r,q}\cdot\tilde{u}_{\ell,x}^{r,q} - \eta_g'^{r,q}\cdot\tilde{u}_{g,x}^r \\
\mathbf{v}^{r,q+1} &= \mathbf{v}^{r,q} - \eta_\ell'^{r,q}\cdot\tilde{u}_{\ell,v}^{r,q} - \eta_g'^{r,q}\cdot\tilde{u}_{g,v}^r
\end{aligned} \tag{B.6}
$$

$$\mathbf{z}^{r,q+1} = \mathbf{z}^{r,q} - \eta_\ell'^{r,q} \cdot \tilde{u}_{\ell,z}^{r,q},$$

where $\eta_\ell'^{r,q} = \tau_\ell \eta_\ell^{r,q}$, $\eta_g'^{r,q} = \tau_\ell \eta_g^{r,q}$. Note that we have $(\cdot)^{r,Q} = (\cdot)^{r+1,0}$. In this case, we have the following result for the stochastic system:

Lemma 9 *Suppose the deterministic system satisfies PD1 - PD4, with stochastic controllers satisfy PS1(A) - PS3, and consider the discretization Case II with $\tau_g = Q\tau_\ell > 0$. Then we have the following:*

$$\begin{aligned}\mathbb{E}[\tilde{\mathcal{E}}^t] - \mathcal{E}^0 \leq & -\sum_{r=0}^{t-1}(\gamma_1^r - C_{21}^r) \cdot \mathbb{E}[\left\|\nabla f(\tilde{\mathbf{x}}^r)\right\|^2] \\ & - \sum_{r=0}^{t-1}(\gamma_2^r - C_{22}^r) \cdot \mathbb{E}[\|(I-R)\cdot \tilde{\mathbf{y}}^r\|^2] \\ & + C_{23}(t)\sigma_g^2 + C_{24}(t)\sigma_\ell^2,\end{aligned} \tag{B.7}$$

where $\{C_{2i}\}_{i=1}^4$ are some coefficients related to L, L_f, C_x, C_v, $B_\ell, B_g, \eta_\ell'^{r,q}, \eta_\ell'^{r,q}$, and Q.

This result is similar to Lemma 4(A). The proof of this lemma is given in Sec. B.3.5.

Also, a similar result with the LCFL satisfies PS1(B) can be proved following similar steps as Lemma 4(B) and Lemma 9. We omitted these derivations to avoid repetition.

Case III: For Case III, $\tau_\ell = K\tau_g > 0$. Let us denote the states at r^{th} local sampling time instance as $(\cdot)^r := (\cdot)(r\tau_\ell)$, where the k^{th} global sampling time between two consecutive local sampling time instances as $(\cdot)^{r,k} := (\cdot)(r\tau_\ell + k\tau_g)$, then the system can be written as:

$$\begin{aligned}\tilde{\mathbf{x}}^{r,k+1} &= \tilde{\mathbf{x}}^{r,k} - \eta_\ell'^{r,k} \cdot \tilde{u}_{\ell,x}^r - \eta_g'^{r,k} \cdot \tilde{u}_{g,x}^{r,k} \\ \tilde{\mathbf{v}}^{r,k+1} &= \tilde{\mathbf{v}}^{r,k} - \eta_\ell'^{r,k} \cdot \tilde{u}_{\ell,v}^r - \eta_g'^{r,k} \cdot \tilde{u}_{g,v}^{r,k} \\ \tilde{\mathbf{z}}^{r,k+1} &= \tilde{\mathbf{z}}^{r,k} - \eta_\ell'^{r,k} \cdot \tilde{u}_{\ell,z}^r,\end{aligned} \tag{B.8}$$

where $\eta_\ell'^{r,k} = \tau_g \eta_\ell^{r,k}$, $\eta_g'^r = \tau_g \eta_g^{r,k}$. Note that $(\cdot)^{r,K} = (\cdot)^{r+1,0}$.

A similar result to Case I can be shown for Case III:

Lemma 10 *Suppose the deterministic system satisfies PD1 - PD4, with stochastic controllers satisfy PS1(A) - PS3, and consider the discretization Case III with $\eta_\ell = K\eta_k > 0$. Then we have the following:*

$$\begin{aligned}\mathbb{E}[\tilde{\mathcal{E}}^t] - \mathcal{E}^0 \leq & -\sum_{r=0}^{t-1}(\gamma_1^r - C_{31}^r) \cdot \mathbb{E}[\left\|\nabla f(\tilde{\mathbf{x}}^r\right\|^2] \\ & - \sum_{r=0}^{t-1}(\gamma_2^r - C_{32}^r) \cdot \mathbb{E}[\|(I-R)\cdot \tilde{\mathbf{y}}^r\|^2] \\ & + C_{33}(t)\sigma_g^2 + C_{34}(t)\sigma_\ell^2,\end{aligned} \tag{B.9}$$

where $\{C_{3i}\}_{i=1}^4$ are some coefficients related to L, L_f, C_x, C_v, $B_\ell, B_g, \eta_\ell'^{r,q}, \eta_\ell'^{r,q}$, and K.

The proof follows the similar steps as in Case I and Case II so we omit it due to page limitation.

B.3.5 Proof of Lemma 9

The proof follows the similar steps as in Case I. We first break down the difference between the energy functions of the consecutive communications as:

$$\begin{aligned}\mathbb{E}_{r,0}\left[\tilde{\mathcal{E}}^{r+1,0}-\tilde{\mathcal{E}}^{r,0}\right] &= \underbrace{\mathbb{E}_{r,0}\left[\tilde{\mathcal{E}}^{r+1,0}\right]-\mathcal{E}^{r+1,0}}_{\Delta^{r+1}} \\ &+\underbrace{\mathcal{E}^{r+1,0}-\tilde{\mathcal{E}}^{r,0}}_{\text{term I}},\end{aligned} \tag{B.10}$$

Then the key is to bound Δ^{r+1}. We proceed by the following:

$$\begin{aligned}\Delta^{r+1} &= \mathbb{E}_{r,0}\left[f(\tilde{\bar{\mathbf{x}}}^{r+1,0})-f(\bar{\mathbf{x}}^{r+1,0})\right] \\ &\quad+\mathbb{E}_{r,0}\left[\left\|(I-R)\cdot\tilde{\mathbf{y}}^{r+1,0}\right\|^2-\left\|(I-R)\cdot\mathbf{y}^{r+1,0}\right\|^2\right] \\ &\overset{(A6)}{\leq} (1+\frac{L_f}{2N})\cdot\mathbb{E}_{r,0}\left\|\mathbf{y}^{r+1,0}-\tilde{\mathbf{y}}^{r+1,0}\right\|^2 \\ &\quad+\mathbb{E}_{r,0}\left[\left\langle\nabla f(\bar{\mathbf{x}}^{r+1,0}),\tilde{\bar{\mathbf{x}}}^{r+1,0}-\bar{\mathbf{x}}^{r+1,0}\right\rangle\right] \\ &= (1+\frac{L_f}{2N})\cdot\mathbb{E}_{r,0}\left\|\mathbf{y}^{r+1,0}-\tilde{\mathbf{y}}^{r+1,0}\right\|^2 \\ &\quad+\mathbb{E}_{r,0}\left[\left\langle\nabla f(\bar{\mathbf{x}}^{r+1,0})-\nabla f(\tilde{\bar{\mathbf{x}}}^{r,0}),\tilde{\bar{\mathbf{x}}}^{r+1,0}-\bar{\mathbf{x}}^{r+1,0}\right\rangle\right] \\ &\quad+\mathbb{E}_{r,0}\left[\left\langle\nabla f(\tilde{\bar{\mathbf{x}}}^{r,0}),\tilde{\bar{\mathbf{x}}}^{r+1,0}-\bar{\mathbf{x}}^{r+1,0}\right\rangle\right] \\ &\overset{(B.1)}{\leq} (1+\frac{L_f}{2N})\cdot\mathbb{E}_{r,0}\left\|\mathbf{y}^{r+1,0}-\tilde{\mathbf{y}}^{r+1,0}\right\|^2 \\ &\quad+\frac{\beta_1}{2}\left\|\nabla f(\bar{\mathbf{x}}^{r+1,0})-\nabla f(\tilde{\bar{\mathbf{x}}}^{r,0})\right\|^2+\frac{\beta_2}{2}\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^{r,0})\right\|^2 \\ &\quad+\frac{\beta_1+\beta_2}{2\beta_1\beta_2}\mathbb{E}_{r,0}\left\|\tilde{\bar{\mathbf{x}}}^{r+1,0}-\bar{\mathbf{x}}^{r+1,0}\right\|^2.\end{aligned}$$

We need to bound each term separately. Notice that we have

$$\left\|\tilde{\bar{\mathbf{x}}}^{r,q}-\bar{\mathbf{x}}^{r,q}\right\|^2\leq\frac{1}{N}\left\|\tilde{\mathbf{x}}^{r,q}-\mathbf{x}^{r,q}\right\|^2,$$

$$\left\|\tilde{\mathbf{x}}^{r,q}-\mathbf{x}^{r,q}\right\|^2\leq\left\|\tilde{\mathbf{y}}^{r,q}-\mathbf{y}^{r,q}\right\|^2\leq\left\|\tilde{\mathbf{s}}^{r,q}-\mathbf{s}^{r,q}\right\|^2.$$

Therefore, we first bound the first term and the last term in the above equation by:

$$\begin{aligned}&\mathbb{E}_{r,0}\left\|\tilde{\mathbf{s}}^{r,q}-\mathbf{s}^{r,q}\right\|^2 \\ &=\mathbb{E}_{r,0}\left\|\sum_{q_1=0}^{q}\eta_\ell^{\prime r,q_1}(\tilde{u}_\ell^{r,q_1}-u_\ell^{r,q_1})+q\eta_g^{\prime r}(\tilde{u}_g^{r,0}-u_g^{r,0})\right\|^2 \\ &\overset{(i)}{\leq} 2q\cdot\sum_{q_1=0}^{q}(\eta_\ell^{\prime r,q_1})^2\cdot\mathbb{E}_{r,0}\left\|\tilde{u}_\ell^{r,q_1}-u_\ell^{r,q_1}\right\|^2\end{aligned} \tag{B.11}$$

$$
\begin{aligned}
&+ 2q^2 \cdot (\eta_g'^r)^2 \cdot \mathbb{E}_{r,0} \left\| \tilde{u}_g^{r,0} - u_g^{r,0} \right\|^2 \\
&\overset{(ii)}{\leq} 4q \cdot \sum_{q_1=0}^{q} (\eta_\ell'^{r,q_1})^2 \cdot \mathbb{E}_{r,0} \left\| \tilde{u}_\ell^{r,q_1} - \mathbb{E}_{r,q_1} \tilde{u}_\ell^{r,q_1} \right\|^2 \\
&\quad + 4q \cdot \sum_{q_1=0}^{q} (\eta_\ell'^{r,q_1})^2 \cdot \mathbb{E}_{r,0} \left\| \mathbb{E}_{r,q_1} \tilde{u}_\ell^{r,q_1} - u_\ell^{r,q_1} \right\|^2 \\
&\quad + 2q^2 \cdot (\eta_g'^r)^2 \cdot \mathrm{Var}_{r,0}(\tilde{u}_g^{r,0}) \\
&\overset{(iii)}{\leq} 4q \cdot \sum_{q_1=0}^{q} (\eta_\ell'^{r,q_1})^2 \cdot \left(B_\ell \left\| \mathbb{E}_{r,q_1} \tilde{u}_\ell^{r,q_1} \right\|^2 + \sigma_\ell^2 \right) \\
&\quad + 4qL^2 \cdot \sum_{q_1=0}^{q} (\eta_\ell'^{r,q_1})^2 \cdot \mathbb{E}_{r,0} \left\| \tilde{\mathbf{s}}^{r,q_1} - \mathbf{s}^{r,q_1} \right\|^2 \\
&\quad + 2q^2 \cdot (\eta_g'^r)^2 \cdot \left(B_g \left\| \mathbb{E}_{r,0} \tilde{u}_g^{r,0} \right\|^2 + \sigma_g^2 \right),
\end{aligned}
$$

where in (i) we plug in the update (B.6) and apply Cauchy–Schwarz inequality; in (ii) we add and subtract $\mathbb{E}_{r,q_1} \tilde{u}_\ell^{r,q_1}$ to the first term and apply PS1 to the second term; in (iii) we apply PS2 to the first and third terms and apply PD2 to the second term. Note that same as (B.4), we have

$$
\begin{aligned}
&\left\| \mathbb{E}_{r,q} \tilde{u}_\ell^{r,q} \right\|^2 \leq C_f \left\| \nabla f(\tilde{\mathbf{x}}^{r,q}) \right\|^2 \\
&\leq 2C_f \left(\left\| \nabla f(\tilde{\tilde{\mathbf{x}}}^{r,q}) \right\|^2 + \left\| \nabla f(\tilde{\mathbf{x}}^{r,q}) - \nabla f(\tilde{\tilde{\mathbf{x}}}^{r,q}) \right\|^2 \right) \\
&\leq 2C_f \left(\left\| \nabla f(\tilde{\tilde{\mathbf{x}}}^{r,q}) \right\|^2 + L_f^2 \left\| \tilde{\mathbf{x}}^{r,q} - \tilde{\tilde{\mathbf{x}}}^{r,q} \right\|^2 \right) \\
&= 2C_f \left(\left\| \nabla f(\tilde{\tilde{\mathbf{x}}}^{r,q}) \right\|^2 + L_f^2 \left\| (I - R) \cdot \tilde{\mathbf{x}}^{r,q} \right\|^2 \right)
\end{aligned}
\tag{B.12}
$$

Applying (B.12) to the first term and PD1 to the last term, recursively apply (B.11) to the second term in (B.11), we obtain:

$$
\begin{aligned}
&\mathbb{E}_{r,0} \left\| \tilde{\mathbf{s}}^{r,q} - \mathbf{s}^{r,q} \right\|^2 \\
&\leq \sum_{q_1=0}^{q} C_{45}^{r,q_1} \left(B_\ell C_f \left\| \nabla f(\tilde{\tilde{\mathbf{x}}}^{r,q_1}) \right\|^2 + \sigma_\ell^2 \right) \\
&\quad + \sum_{q_1=0}^{q} C_{45}^{r,q_1} B_\ell C_f L_f^2 \left\| (I - R) \cdot \tilde{\mathbf{x}}^{r,q_1} \right\|^2 \\
&\quad + \frac{2q^3 \cdot (\eta_g'^r)^2}{1 - 4qL^2 \cdot (\eta_\ell'^{r,0})^2} \cdot \left(B_g \left\| (I - R) \cdot \mathbf{y}^{r,0} \right\|^2 + \sigma_g^2 \right),
\end{aligned}
\tag{B.13}
$$

where we define $C_{45}^{r,q} := \frac{4q \cdot (\eta_\ell'^{r,q})^2}{1 - 4qL^2 \cdot (\eta_\ell'^{r,q})^2}$.

Next, we bound the second term by:

$$
\left\| \nabla f(\bar{\mathbf{x}}^{r+1,0}) - \nabla f(\tilde{\tilde{\mathbf{x}}}^{r,0}) \right\|^2 \overset{(A6)}{\leq} L_f^2 \left\| \bar{\mathbf{x}}^{r+1,0} - \tilde{\tilde{\mathbf{x}}}^{r,0} \right\|^2
$$

$$
\begin{aligned}
&\overset{(B.6)}{=} L_f^2 \left\| \frac{\mathbb{1}_N^T}{N} \sum_{q=0}^{Q-1} \eta_\ell'^{r,q} u_\ell^{r,q} \right\|^2 \\
&\overset{(i)}{\leq} \frac{QL_f^2}{N} \sum_{q=0}^{Q-1} (\eta_\ell'^{r,q})^2 \left\| u_\ell^{r,q} - \mathbb{E}_{r,q}\, \tilde{u}_\ell^{r,q} + \mathbb{E}_{r,q}\, \tilde{u}_\ell^{r,q} \right\|^2 \\
&\overset{(ii)}{\leq} \frac{2QL_f^2}{N} \sum_{q=0}^{Q-1} (\eta_\ell'^{r,q})^2 \left\| \mathbb{E}_{r,q}\, \tilde{u}_\ell^{r,q} \right\|^2 \\
&\quad + \frac{2QL_f^2}{N} \sum_{q=0}^{Q-1} (\eta_\ell'^{r,q})^2 L^2 \left\| \mathbf{s}^{r,q} - \tilde{\mathbf{s}}^{r,q} \right\|^2, \qquad \text{(B.14)}
\end{aligned}
$$

where in (i) we apply Cauchy–Schwarz inequality; in (ii) we first apply Cauchy–Schwarz inequality and then apply PD2. Further plug (B.12) and (B.13) into (B.14), we have:

$$
\begin{aligned}
&\left\| \nabla f(\bar{\mathbf{x}}^{r+1,0}) - \nabla f(\tilde{\bar{\mathbf{x}}}^{r,0}) \right\|^2 \\
&\leq \frac{4QC_f L_f^2}{N} \sum_{q=0}^{Q-1} (\eta_\ell'^{r,q})^2 (\left\| \nabla f(\tilde{\mathbf{x}}^{r,q}) \right\|^2 + L_f^2 \left\| (I-R)\cdot \tilde{\mathbf{x}}^{r,q} \right\|^2) \\
&\quad + \frac{2QL_f^2}{N} \sum_{q=0}^{Q-1} (\eta_\ell'^{r,q})^2 \\
&\quad \times \left(L^2 \sum_{q_1=0}^{q} C_{45}^{r,q_1} \left(B_\ell C_f \left\| \nabla f(\tilde{\bar{\mathbf{x}}}^{r,q_1}) \right\|^2 + \sigma_\ell^2 \right) \right. \\
&\quad + \sum_{q_1=0}^{q} C_{45}^{r,q_1} B_\ell C_f L_f^2 \left\| (I-R)\cdot \tilde{\mathbf{x}}^{r,q_1} \right\|^2 \\
&\quad \left. + \frac{2q^3 \cdot (\eta_g'^r)^2}{1 - 4qL^2 \cdot (\eta_\ell'^{r,0})^2} \cdot \left(B_g \left\| (I-R)\cdot \mathbf{y}^{r,0} \right\|^2 + \sigma_g^2 \right) \right), \qquad \text{(B.15)}
\end{aligned}
$$

Substitute (B.13) and (B.15) into Δ^{r+1} in (B.10), then apply PD4, Lemma 9 is proved.

B.4 Algorithm Design: a Case Study

In this part, we take the gradient tracking algorithm as an example to illustrate how the framework can be applied to design new algorithms for different applications. In specific, we modify the stochastic local and consensus controllers for different applications. Then we verify PS1 - PS3 for the stochastic controllers and PD1-PD4 for the deterministic system, so that we can apply Theorem 3 to obtain the final convergence result and optimize the hyper-parameters. Finally we conduct additional numerical experiments to verify these convergence results.

B.4.1 Gradient-tracking Based Stochastic Algorithm

We start with the deterministic gradient tracking algorithm described in (3.4) as baseline. First, we consider adopting the stochastic gradient, which results in the Distributed Stochastic Gradient Tracking (DSGT) algorithm [25], with the following updates:

$$\begin{aligned}\mathbf{x}^+ &= \mathbf{x} - W\mathbf{x} - \alpha\mathbf{v},\\ \mathbf{v}^+ &= \mathbf{v} - W\mathbf{v} + (\tilde{\nabla} f(\mathbf{x}) - \tilde{\nabla} f(\mathbf{z})),\\ \mathbf{z}^+ &= \mathbf{x},\end{aligned} \tag{B.16}$$

where the LCFL for auxiliary state $\mathbf{v}$ are replaced by the difference of stochastic gradients $\tilde{\nabla} f(\cdot)$ estimated with a subset of samples.

Then, we consider the randomized communication scheme, where each communication connection between the agents has a p failure rate at each round of communication. Which result in gradient tracking on dynamic directed communication graph (D^2GT):

$$\begin{aligned}\mathbf{x}^+ &= \mathbf{x} - \eta_g' \tilde{W}\mathbf{x} - \alpha\mathbf{v},\\ \mathbf{v}^+ &= \mathbf{v} - \eta_g' \tilde{W}\mathbf{v} + (\nabla f(\mathbf{x}) - \nabla f(\mathbf{z})),\\ \mathbf{z}^+ &= \mathbf{x},\end{aligned} \tag{B.17}$$

where $\tilde{W}$ is a stochastic weight matrix satisfies

$$\tilde{W}_{ij} = \tilde{W}_{ji} := \begin{cases} W_{ij}/(1-p), & \text{w.p. } 1-p\\ 0, & \text{w.p. } p\end{cases}.$$

For the third application, we consider adopting the Gaussian mechanism [55] to provide DP guarantee for the local data. The resulting DP-DSGT algorithm is:

$$\begin{aligned}\mathbf{x}^+ &= \mathbf{x} - \eta_g' \tilde{W}\cdot(\mathbf{x} + \mathbf{w}_x) - \alpha\cdot \mathrm{clip}(\mathbf{v}, \beta_x),\\ \mathbf{v}^+ &= \mathbf{v} - \eta_g' \tilde{W}\cdot(\mathbf{v} + \mathbf{w}_v) + \mathrm{clip}(\tilde{\nabla} f(\mathbf{x}) - \tilde{\nabla} f(\mathbf{z}), \beta_v),\\ \mathbf{z}^+ &= \mathbf{x},\end{aligned} \tag{B.18}$$

where $\tilde{W}$ is the same as the one in (B.17), $\mathbf{w}_x \sim \mathcal{N}(0, \sigma_x^2 I), \mathbf{w}_v \sim \mathcal{N}(0, \sigma_v^2 I)$ are the privacy noises, and β_x, β_v are the clipping thresholds.

B.4.2 Theoretical Analysis

In this part, we show how the proposed framework helps analyze the stochastic algorithms. It is easy to verify PD1-PD3. We can also verify PD4 for the deterministic algorithm with $\gamma_1^r = \mathcal{O}(\alpha^r), \gamma_2^r = \mathcal{O}(\alpha^r)$, cf. [25]:

Lemma 11 ([25] Lemma 4) *With the energy function $\mathcal{E}(t)$ defined in (3.6), we have*

$$\mathcal{E}^{r+1} - \mathcal{E}^r \leq -c_1\alpha \left\|\nabla f(\bar{\mathbf{x}}^r)\right\|^2 - c_2\alpha \left\|(I-R)\cdot \mathbf{y}^r\right\|^2,$$

where c_1 and c_2 are some constants depending on C_g, L_f, N.

For DSGT, only the LCFL has stochasticity. By assuming the stochastic gradients are unbiased and has bounded variance, i.e.,

$$\mathbb{E}\tilde{\nabla} f(\mathbf{x}) = \nabla f(\mathbf{x}),\ \mathbb{E}\left\|\tilde{\nabla} f(\mathbf{x}) - \nabla f(\mathbf{x})\right\| \leq \sigma^2.$$

then PS1(A) is satisfied; PS2 is satisfied with $B_\ell = 0, \sigma_\ell = 2\sigma$; and PS3 is also satisfied. Therefore, apply Lemma 4(A), we obtain the following convergence result:

$$\begin{aligned}\mathbb{E}[\tilde{\mathcal{E}}^t] - \mathcal{E}^0 \leq &-\sum_{r=0}^{t-1}\mathcal{O}(\alpha^r)\cdot\mathbb{E}[\left\|\nabla f(\tilde{\mathbf{x}}^r)\right\|^2] \\ &-\sum_{r=0}^{t-1}\mathcal{O}(\alpha^r)\cdot\mathbb{E}[\|(I-R)\cdot\tilde{\mathbf{y}}^r\|^2] + C_{14}(t)\sigma_\ell^2,\end{aligned}$$

where $C_{14} = \sum_{r=0}^{t-1}(\alpha^r)^2\cdot(1+\frac{L_f}{2N})$. Therefore, we can choose $\alpha^r = \mathcal{O}(1/\sqrt{r})$, then the algorithm converges with

$$\begin{aligned}&\mathbb{E}\left[\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^{r_1})\right\|^2 + \|(I-R)\cdot\tilde{\mathbf{y}}^{r_1}\|^2\right] \\ &= \mathcal{O}\left(\frac{1}{\sum_{r=0}^{t-1}\alpha^r}\right)\mathcal{E}^0 + \mathcal{O}\left(\frac{\sum_{r=0}^{t-1}(\alpha^r)^2}{\sum_{r=0}^{t-1}\alpha^r}\right)\sigma_\ell^2.\end{aligned}$$

with rate $\mathcal{O}(\log(t)/\sqrt{t})$. This recovers the convergence result in [25].

For D^{2}GT, only the GCFL has stochasticity. We can verify that PS1(A) is satisfied, PS2 is satisfied with $B_g = p/(1-p), \sigma_g = 0$, and PS3 is also satisfied. Therefore, apply Lemma 4(A), it requires $C_{12}^r = B_g\cdot(\eta_g')^2\cdot(1+\frac{L_f}{2N}) < c_2\alpha^r$. So we can choose $\alpha = \mathcal{O}(1), \eta_g' = \mathcal{O}\left(\sqrt{B_g c_2\alpha^r\cdot(1+\frac{L_f}{2N})}\right)$, and we obtain the following convergence result:

$$\mathbb{E}\left[\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^{r_1})\right\|^2 + \|(I-R)\cdot\tilde{\mathbf{y}}^{r_1}\|^2\right] = \mathcal{O}\left(\frac{1}{\sum_{r=0}^{t-1}\alpha^r}\right)\mathcal{E}^0.$$

with rate $\mathcal{O}(1/t)$.

For DP-DSGT, both controllers have stochasticities. We can verify that PS1(B) is satisfied, with $C_1 = 2(\beta_x + \beta_v)$. For C_2, σ_G can be derived with similar technique in [53]. For PS2, we can verify that $B_\ell = 0, \sigma_\ell = 2\sigma$ and $B_g = p/(1-p), \sigma_g = \sigma_x + \sigma_v$. If we assume $\beta_x \geq \|\mathbf{v}\|^2$ and

$\beta_v \geq \left\|\tilde{\nabla} f(\mathbf{x}) - \tilde{\nabla} f(\mathbf{z})\right\|^2$ for all $t \in [0, \infty)$, then PS1(A) is satisfied. Applying Lemma 4(B), we obtain:

$$\begin{aligned}\mathbb{E}[\tilde{\mathcal{E}}^t] - \mathcal{E}^0 \leq & -\sum_{r=0}^{t-1}(\gamma_1^r - C_{11}'^r)\cdot \mathbb{E}[\left\|\nabla f(\tilde{\mathbf{x}}^r)\right\|^2] \\ & -\sum_{r=0}^{t-1}(\gamma_2^r - C_{12}'^r)\cdot \mathbb{E}[\|(I-R)\cdot \tilde{\mathbf{y}}^r\|^2] \\ & + C_{13}(t)\sigma_g^2 + C_{14}(t)\sigma_\ell^2 + C_{15}(t)C_1 + C_{16}(t)\sigma_G^2.\end{aligned}$$

where

$$\begin{aligned}C_{11}'^r = \mathcal{O}((\alpha^r)^2), \quad C_{12}'^r = \mathcal{O}((\alpha^r)^2 + (\eta_g'^r)^2), \\ \{C_{1i}\}_{i=3}^6 = \mathcal{O}(\sum_{r=0}^{t-1}(\alpha^r)^2),\ \sigma_x^2 = \Omega\left(\frac{C_1 d_x^2 t\cdot(1-p)}{N\epsilon^2}\right), \\ \sigma_v^2 = \Omega\left(\frac{C_1 d_v^2 t\cdot(1-p)}{N\epsilon^2}\right),\end{aligned}$$

σ_x, σ_v are chosen for the algorithm to provide (ϵ, δ)-DP guarantee, cf. [55][Definition 1, Theorem 1]:

Definition 4 ((ϵ, δ)-DP) *An algorithm $\mathcal{M}$ is (ϵ, δ)-DP if*

$$P(\mathcal{M}(\mathcal{D}) \in \mathcal{S}) \leq e^\epsilon P(\mathcal{M}(\mathcal{D}') \in \mathcal{S}) + \delta, \tag{B.19}$$

where $\mathcal{D}$ and $\mathcal{D}'$ are neighboring datasets, $\mathcal{S}$ is an arbitrary subset of outputs of $\mathcal{M}$.

Theorem 10 (Privacy of DP-DSGT) *There exist constants u and v so that given the number of iterations t, for any $\epsilon \leq u(1-p)^2 t$ with p as communication dropout rate, Algorithm DP-DSGT is (ϵ, δ)-differentially private for any $\delta > 0$ if $\sigma^2 \geq v\frac{C_1^2(1-p)T\ln(\frac{1}{\delta})}{N\epsilon^2}$.*

Optimizing p, α, t, β_x, β_v, σ_x, σ_v, we obtain the final convergence rate $\mathcal{O}(\frac{\sqrt{d_x+d_v}}{N\epsilon})$.

B.4.3 Numerical Results

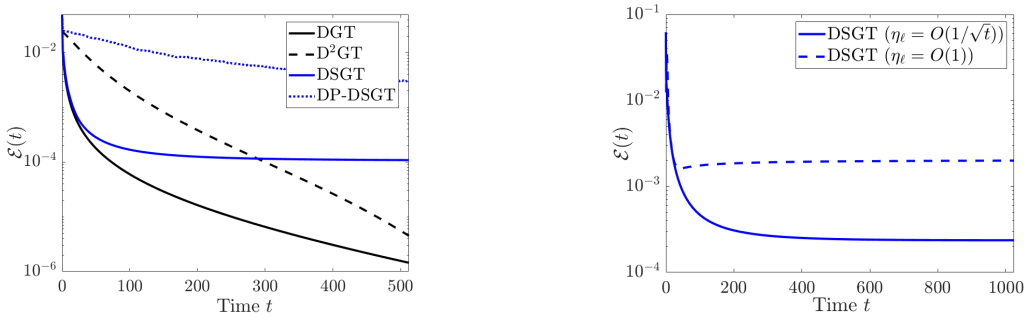
In this subsection, we provide numerical results for implementations of the three algorithms discussed in the previous subsection. We verify the convergence speed derived from the previous subsection for each algorithm.

In the experiments, we consider optimizing the non-convex regularized logistic regression problem:

$$f_i(\mathbf{x}; (\mathbf{a}_i, b_i)) = \log(1 + \exp(-b_i \mathbf{x}^T \mathbf{a}_i)) + \sum_{d=1}^{d_x} \frac{\beta\alpha(\mathbf{x}[d])^2}{1 + \alpha(\mathbf{x}[d])^2},$$

where $\mathbf{a}_i$ denotes the features and b_i denotes the labels of the dataset on the i^{th} agent. We set the number of agents $N = 200$, and each agent has a local dataset of size 1000. We use an Erdős–Rényi random graph with density 0.5 for the network and the weight matrix is selected as $W := 0.9A^TA/\max\{A^TA\}$

For DSGT and DP-DSGT, we use batch size 10 to estimate the stochastic gradients; for D^{2}GT, and DP-DSGT, we choose the communication dropout rate $p = 0.9$. The clipping threshold β_x, β_v are set as the average of local controller's magnitude of the DSGT algorithm and σ_x, σ_v are chosen following [76] with $(\epsilon, \delta) = (4, 10^{-5})$ at $t = 128$.



(a) The convergence of the Energy function $\mathcal{E}(t)$ of DGT, D^{2}GT, DSGT and DP-DSGT.

(b) Energy function $\mathcal{E}(t)$ of DSGT with different decreasing and constant stepsizes $\eta'_\ell(t)$.

Figure B.1: The performance of DGT, D^{2}GT, DSGT and DP-DSGT.

The result is shown in Figure B.1a. It can be observed that D^{2}GT has the same convergence rate as DGT with a constant slowdown, while DSGT and DP-DSGT have slower convergence rates. These results match with the theoretical results in the previous subsection.

In addition, we provided another example demonstrating the necessity of the $\mathcal{O}(1/\sqrt{t})$ rate for DSGT. We run the DSGT algorithm with batch size 2 to estimate the stochastic gradients. In one setting we choose $\alpha = \mathcal{O}(1)$ and $\alpha = \mathcal{O}(1/\sqrt{t})$ in the other setting. The result is shown in Figure B.1b. We can see that with improperly chosen constant stepsize, DSGT will not converge.

Appendix C

Additional Results and Proofs of Chapter 5

C.1 Proof of Claim 2

Proof. We consider the following problem with $N = 2$, which satisfies both Assumptions 8 and 9, with $f(\mathbf{x}) = 0$, $\forall\ \mathbf{x}$. It is easy to show that A10 is not satisfied.

$$f_1(\mathbf{x}) = \mathbf{x}^2, \quad f_2(\mathbf{x}) = -\mathbf{x}^2. \tag{C.1}$$

Each local iteration of the FedAvg is given by

$$\mathbf{x}_1^{r+1} = (1 - \eta^r)\mathbf{x}_1^r, \quad \mathbf{x}_2^{r+1} = (1 + \eta^r)\mathbf{x}_2^r. \tag{C.2}$$

For simplicity, let us define $\mathbf{y} = [\mathbf{x}_1, \mathbf{x}_2]^T$, and define the matrix $\mathbf{D}_r = [1 - \eta^r, 0; 0, 1 + \eta^r]$. Then running Q rounds of the FedAvg algorithm starting with $r = kQ$ for some non-negative integer $k \geq 0$, can be expressed as

$$\mathbf{y}^{(k+1)Q} = \prod_{r=kQ+1}^{(k+1)Q-1} \mathbf{D}_r \mathbf{y}^{kQ+1},\ \mathbf{y}^{kQ+1} = \frac{1}{2}\mathbf{1}\mathbf{1}^T \mathbf{D}_{kQ} \mathbf{y}^{kQ}.$$

Therefore, overall we have

$$\mathbf{y}^{(k+1)Q} = \frac{1}{2} \prod_{r=kQ+1}^{(k+1)Q-1} \mathbf{D}_r \mathbf{1}\mathbf{1}^T \mathbf{D}_{kQ} \mathbf{y}^{kQ}. \tag{C.3}$$

Then for $Q > 1$, we can show that the matrix $\frac{1}{2}\prod_{r=kQ+1}^{(k+1)Q-1} \mathbf{D}_r \mathbf{1}\mathbf{1}^T \mathbf{D}_{kQ}$ has an eigenvalue

given below:

$$\begin{aligned}\lambda &= \frac{1}{2}(\Pi_{r=kQ}^{(k+1)Q}(1-\eta^r) + \Pi_{r=kQ}^{(k+1)Q}(1+\eta^r)) \\ &\overset{(a)}{=} \frac{1}{2}(1+\sum_{q=1}^{Q}(-1)^q P_q(\eta^r) + 1 + \sum_{q=1}^{Q} P_q(\eta^r)) \\ &= 1 + \sum q = 2k, q \leq Q P_q(\eta^r) \\ &> 1,\end{aligned} \tag{C.4}$$

in (a) we break the product into the summation of polynomials of η^r's where $P_q(\eta^r)$ denotes the polynomial of η^r with degree q. This indicates that the algorithm will diverge. ■

C.2 Proofs for Results in Section 5.4

C.2.1 Proof of Theorem 5 and Theorem 7

First, let us assume that when the GD option in Oracle I is used, Q_i is large enough such that the following holds:

$$\left\|\nabla_{\mathbf{x}_i}\mathcal{L}(\mathbf{x}_i^{r,Q_i}, \mathbf{x}_0^r, \lambda_i^r)\right\|^2 = \left\|\nabla_{\mathbf{x}_i}\mathcal{L}(\mathbf{x}_i^{r+1}, \mathbf{x}_0^r, \lambda_i^r)\right\|^2 \leq \epsilon_1. \tag{C.5}$$

Similarly, when the SGD option is used, then Q_i is chosen such that the following holds true:

$$\mathbb{E}[\|\nabla_{\mathbf{x}_i}\mathcal{L}(\mathbf{x}_i^{r,Q_i}, \mathbf{x}_0^r, \lambda_i^r)\|^2] = \mathbb{E}[\|\nabla_{\mathbf{x}_i}\mathcal{L}(\mathbf{x}_i^{r+1}, \mathbf{x}_0^r, \lambda_i^r)\|^2] \leq \epsilon_1. \tag{C.6}$$

The difference does not significantly change the proofs and the results. So throughout the proof of this theorem, we use (C.5) as the condition.

Throughout the proof, we denote the expectation taken on the communication r^{th} iteration to the $r+1^{\text{th}}$ iteration conditioning on all the previous knowledge as $\mathbb{E}_{r+1}$. Using these notations, define the error between different nodes as

$$\triangle^r \triangleq [\triangle\mathbf{x}_0^r; \triangle\mathbf{x}^r], \text{ with} \tag{C.7}$$

$$\triangle\mathbf{x}_0^r \triangleq \max_{i,j} \left\|\mathbf{x}_{0,i}^r - \mathbf{x}_{0,j}^r\right\|, \ \triangle\mathbf{x}^r \triangleq \max_{i,j} \left\|\mathbf{x}_i^r - \mathbf{x}_j^r\right\|. \tag{C.8}$$

Here, $\triangle\mathbf{x}_0^r$ denotes the maximum difference of estimated center model among all the nodes and $\triangle\mathbf{x}^r$ denotes the maximum difference of local models among all nodes.

From the termination condition that generates $\mathbf{x}_i^{r+1}$ (given in (C.5)), we have

$$\begin{aligned}\nabla f_i(\mathbf{x}_i^{r+1}) + \lambda_i^{r+1} &= \nabla f_i(\mathbf{x}_i^{r+1}) + \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r) \\ &= \mathbf{e}_i^{r+1},\end{aligned} \tag{C.9}$$

where $\left\|\mathbf{e}_i^{r+1}\right\|^2 \leq \epsilon_1$, and the first equality holds because of the update rule of λ_i. Furthermore, from the update step of λ_i^{r+1}, we can explicitly write down the following expression

$$\lambda_i^{r+1} = \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r) = -\nabla f_i(\mathbf{x}_i^{r+1}) + \mathbf{e}_i^{r+1}.$$

The main lemmas that we need are outlined below. Their proofs can be found in Appendix Sec. C.3.

Lemma 12 *Suppose A8 holds true. Consider FedPD with Algorithm 5 (Oracle I) as the update rule. When the local problem is solved such that* (C.5) *is satisfied, we have*

$$\begin{aligned}
&\mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^{r+}, \lambda_i^{r+1}) - \mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_{0,i}^r, \lambda_i^r) \\
&\leq -\frac{1-2L\eta}{2\eta}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 - \frac{1}{2\eta}\left\|\mathbf{x}_{0,i}^{r+} - \mathbf{x}_{0,i}^r\right\|^2 \\
&+ \eta\left\|\lambda_i^{r+1} - \lambda_i^r\right\|^2 + \frac{\epsilon_1}{2L}.
\end{aligned} \tag{C.10}$$

Then we derive a key lemma about how the error propagates if the communication step is skipped.

Lemma 13 *Suppose A8 and A11 hold. Consider FedPD with Algorithm 5 (Oracle I) as the update rule. When the local problem is solved such that* (C.5) *is satisfied, the difference between the local models $\mathbf{x}_i^r$'s and the local copies of the global models $\mathbf{x}_{0,i}^r$'s is bounded by*

$$\mathbb{E}_{r+1}\triangle^{r+1} \leq \frac{1}{1-L\eta}(A\triangle^r + \eta B(G + 2\sqrt{\epsilon_1}), \tag{C.11}$$

where we have defined

$$A \triangleq \begin{bmatrix} p(1+L\eta) & pL\eta(1-L\eta) \\ 1 & L\eta \end{bmatrix},$$

which is a rank one matrix with eigenvalues $(0, L\eta + p(1+L\eta))$ and $B = [p(3+L\eta), 2]^T$.

Next, we define a virtual sequence $\{\overline{\mathbf{x}}_0^r\}$ where $\overline{\mathbf{x}}_0^r \triangleq \frac{1}{N}\sum_{i=1}^N \mathbf{x}_{0,i}^r$ which is the average of the local $\mathbf{x}_{0,i}^r$. We know that $\mathbf{x}_{0,i}^r = \mathbf{x}_0^r$ when the communication and aggregation step is performed). Next, we bound the error between the local AL and the global AL evaluated at the virtual sequence.

Lemma 14 *Suppose A8 holds. Consider FedPD with Algorithm 5 (Oracle I) as the update rule. When the local problem is solved such that* (C.5) *is satisfied, the difference between local AL and the global AL is bounded as below:*

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^N [\mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^{r+}, \lambda_i^{r+1}) - \mathcal{L}_i(\mathbf{x}_i^{r+1}, \overline{\mathbf{x}}_0^{r+1}, \lambda_i^{r+1})] \\
&\geq -\frac{(N-1)}{2N\eta}(\triangle \mathbf{x}_0^{r+1})^2.
\end{aligned} \tag{C.12}$$

Lastly we bound the objective function using the global AL.

Lemma 15 *Under A8 and A9, consider FedPD with Algorithm 5 (Oracle I) as the update rule. When the local problem is solved to ϵ_1 accuracy satisfying* (C.5)*, the difference between the original loss and the augmented Lagrangian is bounded.*

$$f(\mathbf{x}_0^r) \leq \mathcal{L}(\mathbf{x}_{0:N}^r, \boldsymbol{\lambda}^r) - \frac{1-2L\eta}{N\eta}\sum_{i=1}^N \|\mathbf{x}_i^r - \mathbf{x}_0^r\|^2 + \frac{\epsilon_1}{2L}. \tag{C.13}$$

Now we are ready to prove Theorem 5 and Theorem 7.

C.2.2 Proof of Theorem 5 and Theorem 7

First, for notational simplicity, let us define the following:

$$\begin{aligned} &\mathcal{L}_i^r \triangleq \mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_{0,i}^r, \lambda_i^r),\ \mathcal{L}_i^{r+1} \triangleq \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^{r+1}, \lambda_i^{r+1}) \\ &\mathcal{L}_i^{r+} \triangleq \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^{r+}, \lambda_i^{r+1}),\ \bar{\mathcal{L}}_i^{r+1} \triangleq \mathcal{L}_i(\mathbf{x}_i^{r+1}, \overline{\mathbf{x}}_0^{r+1}, \lambda_i^{r+1}). \end{aligned} \tag{C.14}$$

Notice that from the optimality condition (C.9), the following holds:

$$\left\|\lambda_i^r - \lambda_i^{r-1}\right\|^2 \leq 2L^2 \left\|\mathbf{x}_i^r - \mathbf{x}_i^{r-1}\right\|^2 + 4\epsilon_1. \tag{C.15}$$

Then we bound the gradients of $\mathcal{L}(\mathbf{x}_i^r, \mathbf{x}_{0,i}^r, \lambda_i^r)$.

$$\begin{aligned} \|\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\| &= \left\|\nabla f_i(\mathbf{x}_i^r) + \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^r - \mathbf{x}_{0,i}^r)\right\| \\ &\overset{(C.9)}{=} \left\|\nabla f_i(\mathbf{x}_i^r) + \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^r - \mathbf{x}_{0,i}^r) - \nabla f_i(\mathbf{x}_i^{r+1}) - \lambda_i^r \right. \\ &\left. -\frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r) + \mathbf{e}_i^{r+1}\right\| \leq \frac{1+L\eta}{\eta}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\| + \sqrt{\epsilon_1}. \end{aligned} \tag{C.16}$$

Note that when no aggregation has been performed at iteration r, then $\mathbf{x}_{0,i}^r = \mathbf{x}_i^r + \eta\lambda_i^r$, so the following holds

$$\|\nabla_{\mathbf{x}_0}\mathcal{L}_i^r\| = \left\|\lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^r - \mathbf{x}_{0,i}^r)\right\| = 0. \tag{C.17}$$

When aggregation has been performed at iteration r, then $\mathbf{x}_{0,i}^r = \frac{1}{N}\sum_{j=1}^N(\mathbf{x}_j^r + \eta\lambda_j^r),\ \forall i$, so we have

$$\|\nabla_{\mathbf{x}_0}\mathcal{L}(\mathbf{x}_{0:N}^r, \boldsymbol{\lambda}^r)\| = \left\|\frac{1}{N}\sum_{i=1}^N(\lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^r - \mathbf{x}_{0,i}^r))\right\| = 0. \tag{C.18}$$

Further by using the definition of $\mathcal{L}_i^r$ and the dual update step, we have:

$$\begin{aligned} \|\nabla_{\lambda_i}\mathcal{L}_i^r\| =& \left\|\mathbf{x}_i^r - \mathbf{x}_{0,i}^r\right\| \\ \leq& \left\|\mathbf{x}_i^r - \mathbf{x}_{0,i}^{r-1}\right\| + \left\|\mathbf{x}_{0,i}^{r-1} - \mathbf{x}_{0,i}^r\right\| \\ \leq& \eta\left\|\lambda_i^r - \lambda_i^{r-1}\right\| + \left\|\mathbf{x}_{0,i}^{r-1} - \mathbf{x}_{0,i}^r\right\| \\ \leq& \eta(L\left\|\mathbf{x}_i^r - \mathbf{x}_i^{r-1}\right\| + 2\sqrt{\epsilon_1}) + \left\|\mathbf{x}_{0,i}^{r-1} - \mathbf{x}_{0,i}^r\right\|. \end{aligned} \tag{C.19}$$

From (C.17), we know that $\|\nabla_{\mathbf{x}_0}\mathcal{L}_i^r\| = 0$. So we see that the size of the full gradient $\nabla\mathcal{L}_i^r$ can be expressed by:

$$\left\|\nabla\mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_{0,i}^r, \lambda_i^r)\right\|^2 = \|\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\|^2 + \|\nabla_{\lambda_i}\mathcal{L}_i^r\|^2 \tag{C.20}$$

$$\leq (\|\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\| + \|\nabla_{\lambda_i}\mathcal{L}_i^r\|)^2. \tag{C.21}$$

Then we have

$$\begin{aligned}
&\|\nabla\mathcal{L}_i^r\|^2 \leq \Bigg(\frac{1+L\eta}{\eta}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\| + \sqrt{\epsilon_1} \\
&+ \eta(L\left\|\mathbf{x}_i^r - \mathbf{x}_i^{r-1}\right\| + 2\sqrt{\epsilon_1}) + \left\|\mathbf{x}_{0,i}^{r-1} - \mathbf{x}_{0,i}^r\right\|\Big)^2 \\
&\leq C_6\left(\left\|\mathbf{x}_{0,i}^{r-1} - \mathbf{x}_{0,i}^r\right\|^2\right. \\
&\left.+ \left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 + \left\|\mathbf{x}_i^r - \mathbf{x}_i^{r-1}\right\|^2 + \epsilon_1\right),
\end{aligned} \tag{C.22}$$

where $C_6 \geq 3\max\{(\frac{1+L\eta}{\eta})^2, (1+2\eta)^2, L^2\eta^2\}$. Apply (C.15) to Lemma 12 we obtain

$$\begin{aligned}
&\frac{1-2L\eta-4L^2\eta^2}{2\eta}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 + \frac{1}{2\eta}\left\|\mathbf{x}_{0,i}^{r+} - \mathbf{x}_{0,i}^r\right\|^2 \\
&-\frac{1+8L\eta}{2L}\epsilon_1 \leq \mathcal{L}_i^r - \mathcal{L}_i^{r+}.
\end{aligned} \tag{C.23}$$

Notice that when communication is not performed, we have $\left\|\mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+1}\right\|^2 \leq \left\|\mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+}\right\|^2$. When communication is performed, the following holds:

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^N\left\|\mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+1}\right\|^2 \\
&= \frac{2}{N}\sum_{i=1}^N\left\|\mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+}\right\|^2 + \frac{2}{N}\sum_{i=1}^N\left\|\mathbf{x}_{0,i}^{r+} - \mathbf{x}_{0,i}^{r+1}\right\|^2 \\
&\leq \frac{2}{N}\sum_{i=1}^N\left\|\mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+}\right\|^2 + \frac{N-1}{\eta N}(\triangle\mathbf{x}_0^{r+1})^2,
\end{aligned} \tag{C.24}$$

where the last inequality holds due to the use of Jensen's inequality, and the definition of $\triangle\mathbf{x}_0^{r+1}$ in (C.7). It follows that summing both sides of (C.23) over i, we have

$$\begin{aligned}
&\frac{1-2L\eta-4L^2\eta^2}{2\eta}\sum_{i=1}^N\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 + \frac{N(1+8L\eta)}{2L}\epsilon_1 \\
&+\sum_{i=1}^N(\frac{1}{4\eta}\left\|\mathbf{x}_{0,i}^{r+1} - \mathbf{x}_{0,i}^r\right\|^2 - \frac{N-1}{4\eta}(\triangle\mathbf{x}_0^{r+1})^2) \\
&\leq \sum_{i=1}^N\left(\mathcal{L}_i^r - \mathcal{L}_i^{r+}\right) + \frac{N(1+8L\eta)}{L}\epsilon_1.
\end{aligned} \tag{C.25}$$

Taking the expectation over the randomness on the communication step, we obtain the following:

$$\begin{aligned}
&\mathbb{E}_{r+1}\frac{1}{N}\sum_{i=1}^{N}[\mathcal{L}_i^r-\mathcal{L}_i^{r+}]\\
&=\frac{1}{N}\sum_{i=1}^{N}[\mathcal{L}_i^r-\mathcal{L}_i^{r+1}]+\mathbb{E}_{r+1}\frac{1}{N}\sum_{i=1}^{N}[\mathcal{L}_i^{r+1}-\mathcal{L}_i^{r+}]\\
&\overset{(a)}{=}\frac{1}{N}\sum_{i=1}^{N}[\mathcal{L}_i^r-\mathcal{L}_i^{r+1}]+\frac{1}{N}\sum_{i=1}^{N}[(1-p)\bar{\mathcal{L}}_i^{r+1}+p\mathcal{L}_i^{r+}-\mathcal{L}_i^{+}]\\
&=\frac{1}{N}\sum_{i=1}^{N}[\mathcal{L}_i^r-\mathcal{L}_i^{r+1}]+(1-p)\frac{1}{N}\sum_{i=1}^{N}[\bar{\mathcal{L}}_i^{r+1}-\mathcal{L}_i^{r+}]\\
&\overset{(b)}{\leq}\frac{1}{N}\sum_{i=1}^{N}[\mathcal{L}_i^r-\mathcal{L}_i^{r+1}]+(1-p)\frac{N-1}{2\eta N}(\triangle\mathbf{x}_0^{r+1})^2
\end{aligned}\tag{C.26}$$

where (a) expands the expectation on p, and use the fact that with probability p, $\mathbf{x}_{0,i}^{r+1}=\mathbf{x}_{0,i}^{r+}$, and with probability $(1-p)$ $\mathbf{x}_0^{r+1}$ will be updated; in (b) we apply Lemma 14 to the last term.

Combining (C.25) and (C.26), we have

$$\begin{aligned}
&\min\left\{\frac{1-2L\eta-4L^2\eta^2}{2\eta},\frac{1}{2\eta},\frac{1+8L\eta}{2L}\right\}\\
&\times\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{r+1}\left[\left\|\mathbf{x}_i^{r+1}-\mathbf{x}_i^r\right\|^2+\left\|\mathbf{x}_{0,i}^{r+1}-\mathbf{x}_{0,i}^r\right\|^2+\epsilon_1\right]\\
&\leq\frac{1}{N}\sum_{i=1}^{N}\left[\mathcal{L}_i^r-\mathcal{L}_i^{r+1}\right]\\
&+\frac{1+8L\eta}{L}\epsilon_1+(1-p)\frac{(N-1)}{\eta N}(\triangle\mathbf{x}_0^{r+1})^2.
\end{aligned}\tag{C.27}$$

Combining (C.22), (C.25) and (C.27), define $C_7=2C_6/\min\{\frac{1-2L\eta-4L^2\eta^2}{2\eta},\frac{1}{2\eta},\frac{1+8L\eta}{2L}\}$ and sum

up the iterations, we have

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\sum_{r=0}^{T}\mathbb{E}\left\|\nabla\mathcal{L}_i^r\right\|^2 \\
&\overset{(C.22)(C.25)}{\leq} \frac{2C_6}{N}\sum_{i=1}^{N}\sum_{r=0}^{T}\mathbb{E}\left[\left\|\mathbf{x}_{0,i}^r-\mathbf{x}_{0,i}^{r+}\right\|^2+\left\|\mathbf{x}_i^r-\mathbf{x}_i^{r+1}\right\|^2\right. \\
&\left.+(1-p)\frac{(N-1)}{\eta N}(\triangle\mathbf{x}_0^{r+1})^2+\epsilon_1\right] \\
&\overset{(C.27)}{\leq} \frac{C_7}{N}\sum_{r=0}^{T}\sum_{i=1}^{N}(\mathcal{L}_i^r-\mathcal{L}_i^{r+1}) \\
&\quad+\frac{C_7(1+8L\eta)}{L}\epsilon_1+(1-p)C_7\sum_{r=0}^{T}\frac{N-1}{N\eta}\,\mathbb{E}(\triangle\mathbf{x}_0^{r+1})^2.
\end{aligned} \tag{C.28}$$

Next we bound the last term in the above inequality. By iteratively applying Lemma 13 from $\tau=0$ to r and use the fact that $G^0=0$, we have

$$\begin{aligned}
&\mathbb{E}\triangle\mathbf{x}_0^{r+1}\overset{(C.7)}{=}[1,0]\times\mathbb{E}\triangle^{r+1} \\
&\leq[1,0]\times\sum_{\tau=0}^{r}(\frac{A}{1-L\eta})^\tau\eta\frac{[p(3+L\eta),2]^T}{1-L\eta}(G+2\sqrt{\epsilon_1}).
\end{aligned} \tag{C.29}$$

From Lemma 13 we have:

$$\lambda\left(\frac{1}{1-L\eta}A\right)=\frac{p(1+L\eta)+L\eta}{1-L\eta}\triangleq C_8.$$

So by squaring both side of (C.29), we have

$$\begin{aligned}
&\mathbb{E}(\triangle\mathbf{x}_0^{r+1})^2 \\
&\leq\left\|[1,0]\sum_{\tau=1}^{r}(\frac{A}{1-L\eta})^\tau\eta\frac{[p(3+L\eta),2]^T}{1-L\eta}(G+2\sqrt{\epsilon_1})\right\|^2 \\
&\leq\mathbb{E}\left(\frac{1-C_8^{r+1}}{1-C_8}\right)^2(2p\eta(1+L\eta)(2L\eta+p(3+L\eta)))^2 \\
&\times(G^2+\epsilon_1) \\
&=4p^2\eta^2(1+L\eta)^2(2L\eta+p(3+L\eta))^2 \\
&\times\left(\frac{1-C_8^{1/(1-p)}}{1-C_8}\right)^2(G^2+\epsilon_1).
\end{aligned} \tag{C.30}$$

Substitute (C.30) into (C.28) and divide both sides by T we have

$$
\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\frac{1}{T}\sum_{r=0}^{T}\mathbb{E}\left\|\nabla\mathcal{L}_i^r\right\|^2\\
&\leq \frac{C_7}{T}\left(\mathcal{L}(\mathbf{x}_0^0,\mathbf{x}_i^0,\lambda_i^0)-\mathcal{L}(\mathbf{x}_i^T,\mathbf{x}_{0,i}^T,\lambda_i^T)\right)+\frac{C_7(1+8L\eta)}{L}\epsilon_1\\
&\quad+\eta^2(1-p)(N-1)C_7(1-C_8^{1/(1-p)})^2p^2\\
&\quad\times\frac{(1+L\eta)^2(2L\eta+p(3+L\eta))^2}{N(1-C_8)^2}(G^2+\epsilon_1).
\end{aligned}
\tag{C.31}
$$

From the initial conditions we have $\mathcal{L}(\mathbf{x}_0^0,\mathbf{x}_i^0,\lambda_i^0)=f(\mathbf{x}_0^0)$ and apply Lemma 15 we obtain

$$
\begin{aligned}
&\frac{1}{NT}\sum_{i=1}^{N}\sum_{r=0}^{T}\mathbb{E}\left\|\nabla\mathcal{L}_i^r\right\|^2\\
&\leq \frac{C_7(f(\mathbf{x}_0^0)-f(\mathbf{x}_0^T))}{T}+\frac{C_7(1+8L\eta)}{L}\epsilon_1\\
&\quad+\eta^2(1-p)(N-1)C_7(1-C_8^{1/(1-p)})^2p^2\\
&\quad\times\frac{(1+L\eta)^2(2L\eta+p(3+L\eta))^2}{N(1-C_8)^2}(G^2+\epsilon_1).
\end{aligned}
\tag{C.32}
$$

Finally we bound $\left\|\nabla f(\mathbf{x}_0^r)\right\|^2$ by

$$
\begin{aligned}
&\left\|\nabla f(\mathbf{x}_0^r)\right\|^2\\
&\leq\ 2\left\|\nabla f(\mathbf{x}_0^r)-\frac{1}{N}\sum_{i=1}^{N}\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\right\|^2+\frac{2}{N}\sum_{i=1}^{N}\left\|\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\right\|^2\\
&\leq \frac{4}{N}\sum_{i=1}^{N}\left\|\nabla f_i(\mathbf{x}_0^r)-\nabla f_i(\mathbf{x}_i^r)\right\|^2\\
&\quad+4\left\|\frac{1}{N\eta}\sum_{i=1}^{N}(\eta\lambda_i^r+\mathbf{x}_i^r-\mathbf{x}_{0,i}^r)\right\|^2+\frac{2}{N}\sum_{i=1}^{N}\left\|\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\right\|^2\\
&\overset{(a)}{\leq}\frac{4L^2}{N}\sum_{i=1}^{N}\left\|\mathbf{x}_0^r-\mathbf{x}_i^r\right\|^2+\frac{2}{N}\sum_{i=1}^{N}\left\|\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\right\|^2\\
&=\frac{4L^2}{N}\sum_{i=1}^{N}\left\|\nabla_{\lambda_i}\mathcal{L}_i^r\right\|^2+\frac{2}{N}\sum_{i=1}^{N}\left\|\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\right\|^2\leq\ \frac{4L^2}{N}\sum_{i=1}^{N}\left\|\nabla\mathcal{L}_i^r\right\|^2,
\end{aligned}
\tag{C.33}
$$

where in (a) we use the same argument in (C.17) and (C.18).

Therefore, set $p=0$ Theorem 5 is proved, and when $p\neq 0$, Theorem 7 is proved. During the proof, we need all $C_2,\ldots,C_7,C_8>0$, therefore, $0<\eta<\frac{\sqrt{5}-1}{4L}$.

Finally, let us note that if the local problems are solved with SGD, then the local problem needs to be solved such that the condition (C.6) holds true. As no other information of the local

solvers except error term $\mathbf{e}_i^r$ is used in the proof, the proofs and results of FedPD with SGD as local solver will not change much, except that all the results hold in expectation. Therefore we skip the proof for the SGD version.

C.2.3 Constants used in the proofs

In this subsection we list all the constants $C_2, \ldots, C_8$ used in the proof of Theorem 5 and Theorem 7.

$$
\begin{aligned}
&C_2 \geq 4L^2 C_7, \qquad C_3 = C_8, \qquad C_4 \geq \frac{C_2(1+8L\eta)}{L},\\
&C_5 = 8C_2, \qquad C_6 \geq 3\max\{(\frac{1+L\eta}{\eta})^2, (1+2\eta)^2, L^2\eta^2\},\\
&C_7 = 2C_6/\min\{\frac{1-2L\eta-4L^2\eta^2}{2\eta}, \frac{1}{2\eta}, \frac{1+8L\eta}{2L}\},\\
&C_8 = \frac{p(1+L\eta)+L\eta}{1-L\eta},
\end{aligned}
$$

we can see that when $0 < \eta < \frac{\sqrt{5}-1}{4L}$, all the terms are positive.

C.3 Proof for Lemma 12– Lemma 14

C.3.1 Proof of Lemma 12

We divide the left hand side (LHS) of (C.10), i.e., $\mathcal{L}_i^{r+} - \mathcal{L}_i^r$, into the sum of three parts (where $\mathcal{L}_i^{r+}, \mathcal{L}_i^r$ are defined in (C.14)):

$$
\begin{aligned}
\mathcal{L}_i^{r+} - \mathcal{L}_i^r = &\mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^r) - \mathcal{L}_i^r\\
&+ \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^{r+1}) - \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^r)\\
&+ \mathcal{L}_i^{r+} - \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^{r+1}).
\end{aligned} \tag{C.34}
$$

We bound the first difference by first applying A8 to $-f(\cdot)$:

$$
-f_i(\mathbf{x}_i^r) \leq -f_i(\mathbf{x}_i^{r+1}) + \left\langle -\nabla f_i(\mathbf{x}_i^{r+1}), \mathbf{x}_i^r - \mathbf{x}_i^{r+1} \right\rangle + \frac{L}{2}\left\|\mathbf{x}_i^r - \mathbf{x}_i^{r+1}\right\|^2,
$$

and obtain the following series of inequalities:

$$
\begin{aligned}
&\mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^r) - \mathcal{L}_i^r \\
&\leq \left\langle \nabla f_i(\mathbf{x}_i^{r+1}), \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\rangle + \frac{L}{2} \left\| \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\|^2 \\
&+ \left\langle \lambda_i^r, \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\rangle + \frac{1}{2\eta} \left\| \mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r \right\|^2 - \frac{1}{2\eta} \left\| \mathbf{x}_i^r - \mathbf{x}_{0,i}^r \right\|^2 \\
\overset{(a)}{=}\ &\left\langle \nabla f_i(\mathbf{x}_i^{r+1}) + \lambda_i^r, \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\rangle + \frac{L}{2} \left\| \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\|^2 \\
&+ \frac{1}{2\eta} \left\langle \mathbf{x}_i^{r+1} + \mathbf{x}_i^r - 2\mathbf{x}_{0,i}^r, \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\rangle \\
\overset{(b)}{=}\ &\left\langle \nabla f_i(\mathbf{x}_i^{r+1}) + \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r), \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\rangle \\
&+ \frac{L}{2} \left\| \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\|^2 - \frac{1}{2\eta} \left\| \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\|^2 \\
\overset{(c)}{\leq}\ &\frac{1}{2L} \left\| \nabla f_i(\mathbf{x}_i^{r+1}) + \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r) \right\|^2 \\
&+ \frac{L}{2} \left\| \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\|^2 - \frac{1 - L\eta}{2\eta} \left\| \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\|^2 \\
\overset{(d)}{\leq}\ &- \frac{1 - 2L\eta}{2\eta} \left\| \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\|^2 + \frac{\epsilon_1}{2L}.
\end{aligned}
\tag{C.35}
$$

In the above derivation, in (a) we use the fact that $\|a\|^2 - \|b\|^2 = \langle a + b, a - b \rangle$ when vector a, b has the same length to the last two terms; in (b) we split the last term into $2\mathbf{x}_i^{r+1} - 2\mathbf{x}_{0,i}^r$ and $-\mathbf{x}_i^{r+1} + \mathbf{x}_i^r$; in (c) we use the fact that $\langle a, b \rangle \leq \frac{L}{2} \|a\|^2 + \frac{1}{2L} \|b\|^2$); in (d) we apply the fact that $\mathbf{x}_i^{r+1}$ is the inexact solution; see (C.9).

Then we bound the second difference in (C.34) by the following:

$$
\begin{aligned}
&\mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^{r+1}) - \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^r) \\
&= \left\langle \lambda_i^{r+1} - \lambda_i^r, \mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r \right\rangle \\
&\overset{(a)}{=} \left\langle \lambda_i^{r+1} - \lambda_i^r, \eta(\lambda_i^{r+1} - \lambda_i^r) \right\rangle = \eta \left\| \lambda_i^{r+1} - \lambda_i^r \right\|^2,
\end{aligned}
\tag{C.36}
$$

where (a) directly comes from the update rule of λ_i^{r+1}.

Further we bound the third difference in (C.34) by the following:

$$
\begin{aligned}
&\mathcal{L}_i^{r+} - \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^{r+1}) \\
&= \left\langle \lambda_i^{r+1}, \mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^{r+} \right\rangle - \left\langle \lambda_i^{r+1}, \mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r \right\rangle \\
&\quad + \frac{1}{2\eta} \left\| \mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^{r+} \right\|^2 - \frac{1}{2\eta} \left\| \mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r \right\|^2 \\
&\overset{(a)}{=} \left\langle \lambda_i^{r+1}, \mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+} \right\rangle + \frac{1}{2\eta} \left\langle 2\mathbf{x}_i^{r+1} - 2\mathbf{x}_{0,i}^{r+} + \mathbf{x}_{0,i}^{r+} - \mathbf{x}_{0,i}^r, \mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+} \right\rangle \\
&= \left\langle \frac{1}{\eta}(\eta\lambda_i^{r+1} + \mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^{r+}), \mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+} \right\rangle - \frac{1}{2\eta} \left\| \mathbf{x}_{0,i}^{r+} - \mathbf{x}_{0,i}^r \right\|^2 \\
&\overset{(b)}{=} -\frac{1}{2\eta} \left\| \mathbf{x}_{0,i}^{r+} - \mathbf{x}_{0,i}^r \right\|^2,
\end{aligned}
\tag{C.37}
$$

where, in (a), we use the same reasoning as in (C.35) (a) and (b); in (b) we apply the update rule of $\mathbf{x}_{0,i}^{r+}$ in the FedPD algorithm, which implies that the first term becomes zero.

Finally we sum up (C.35), (C.36), (C.37) and Lemma 12 is proved.

C.3.2 Proof of Lemma 13

First we derive the relation between $\left\| \mathbf{x}_i^{r+1} - \mathbf{x}_j^{r+1} \right\|$ for arbitrary $i \neq j$ and $\triangle^r$ by using the definition of ϵ_1 (C.9):

$$
\begin{aligned}
&\left\| \mathbf{x}_i^{r+1} - \mathbf{x}_j^{r+1} \right\| \\
&\overset{\text{(C.9)}}{=} \left\| \mathbf{x}_{0,i}^r - \mathbf{x}_{0,j}^r - \eta(\nabla f_i(\mathbf{x}_i^{r+1}) + \lambda_i^r - \mathbf{e}_i^{r+1} - \nabla f_j(\mathbf{x}_j^{r+1}) - \lambda_j^r + \mathbf{e}_j^{r+1}) \right\| \\
&\leq \left\| \mathbf{x}_{0,i}^r - \mathbf{x}_{0,j}^r \right\| + \eta \left\| \nabla f_i(\mathbf{x}_i^{r+1}) - \nabla f_j(\mathbf{x}_j^{r+1}) \right\| \\
&\quad + \eta \left\| \lambda_i^r - \lambda_j^r \right\| + \eta(\left\| \mathbf{e}_i^{r+1} \right\| + \left\| \mathbf{e}_j^{r+1} \right\|) \\
&\overset{(a)}{\leq} \triangle\mathbf{x}_0^r + \eta \left\| \nabla f_i(\mathbf{x}_i^{r+1}) - \nabla f_i(\mathbf{x}_j^{r+1}) + \nabla f_i(\mathbf{x}_j^{r+1}) - \nabla f_j(\mathbf{x}_j^{r+1}) \right\| \\
&\quad + \eta \left\| \lambda_i^r - \lambda_j^r \right\| + 2\eta\sqrt{\epsilon_1} \\
&\overset{(b)}{\leq} \triangle\mathbf{x}_0^r + L\eta \left\| \mathbf{x}_i^{r+1} - \mathbf{x}_j^{r+1} \right\| + \eta \left\| \nabla f_i(\mathbf{x}_j^{r+1}) - \nabla f_j(\mathbf{x}_j^{r+1}) \right\| \\
&\quad + \eta \left\| \lambda_i^r - \lambda_j^r \right\| + 2\eta\sqrt{\epsilon_1} \\
&\overset{(c)}{\leq} \triangle\mathbf{x}_0^r + L\eta \left\| \mathbf{x}_i^{r+1} - \mathbf{x}_j^{r+1} \right\| + \eta G + \eta \left\| \lambda_i^r - \lambda_j^r \right\| + 2\eta\sqrt{\epsilon_1} \\
&\overset{(d)}{=} \frac{1}{1 - L\eta} \triangle\mathbf{x}_0^r + \frac{\eta}{1 - L\eta} G + \frac{\eta}{1 - L\eta} \left\| \lambda_i^r - \lambda_j^r \right\| + \frac{2\eta}{1 - L\eta} \sqrt{\epsilon_1},
\end{aligned}
\tag{C.38}
$$

where in (a) we plug the definition of $\triangle\mathbf{x}_0^r$ and $\mathbf{e}_i^{r+1}$; in (b) we use A8; (c) comes form A11; in (d) we move the second term to the left and divide both side by $1 - L\eta$.

Then we bound the difference $\left\| \lambda_i^r - \lambda_j^r \right\|$ by plugging in the expression of λ_i^r in (C.9), and

note that $\lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r) = \lambda_i^{r+1}$, we have:

$$\begin{aligned}
&\left\|\lambda_i^r - \lambda_j^r\right\| \\
=\ &\left\|-\nabla f_i(\mathbf{x}_i^r) + \mathbf{e}_i^r + \nabla f_j(\mathbf{x}_j^r) - \mathbf{e}_j^r\right\| \\
\overset{(a)}{\leq}\ &\left\|\nabla f_i(\mathbf{x}_i^r) - \nabla f_i(\mathbf{x}_j^r)\right\| + \left\|\nabla f_i(\mathbf{x}_j^r) - \nabla f_j(\mathbf{x}_j^r)\right\| + 2\sqrt{\epsilon_1} \\
\overset{(b)}{\leq}\ &L\left\|\mathbf{x}_i^r - \mathbf{x}_j^r\right\| + G + 2\sqrt{\epsilon_1} \\
\overset{(c)}{\leq}\ &L\triangle\mathbf{x}^r + G + 2\sqrt{\epsilon_1},
\end{aligned} \tag{C.39}$$

where (a) and (b) follow the same argument in (a), (b) and (c) of (C.38) ; in (c) we plug in the definition of $\triangle\mathbf{x}^r$.

Next we bound the difference $\left\|\mathbf{x}_{0,i}^{r+1} - \mathbf{x}_{0,j}^{r+1}\right\|$. With probability $1-p$ the aggregation step has just been done at iteration r, $\mathbf{x}_{0,i}^{r+1} = \mathbf{x}_{0,j}^{r+1}$.With probability p, they are not equal, then we take expectation with communication probability p, and obtain

$$\begin{aligned}
&\mathbb{E}_{r+1}\left\|\mathbf{x}_{0,i}^{r+1} - \mathbf{x}_{0,j}^{r+1}\right\| \\
=\ &p\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_j^{r+1} + \eta(\lambda_i^{r+1} - \lambda_j^{r+1})\right\| \\
\leq\ &p\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_j^{r+1}\right\| + p\eta\left\|\lambda_i^{r+1} - \lambda_j^{r+1}\right\| \\
\overset{(a)}{\leq}\ &p(1+L\eta)\triangle\mathbf{x}^{r+1} + p\eta(G + 2\sqrt{\epsilon_1}),
\end{aligned} \tag{C.40}$$

where in (a) we plug in the definition of $\triangle\mathbf{x}^{r+1}$ and (C.39). As these relations hold true for arbitrary (i,j) pairs, they are also true for the maximum of $\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_j^{r+1}\right\|$ and $\left\|\mathbf{x}_{0,i}^{r+1} - \mathbf{x}_{0,j}^{r+1}\right\|$.

Therefore stacking (C.38) and (C.40) and plug in (C.39), we have

$$\begin{aligned}
\triangle\mathbf{x}^{r+1} \leq&\frac{1}{1-L\eta}(L\eta\triangle\mathbf{x}^r + \triangle\mathbf{x}_0^r) + \frac{2\eta}{1-L\eta}(G + 2\sqrt{\epsilon_1}), \\
\mathbb{E}_{r+1}\triangle\mathbf{x}_0^{r+1} \leq&p\frac{1+L\eta}{1-L\eta}(L\eta\triangle\mathbf{x}^r + \triangle\mathbf{x}_0^r) + p\frac{\eta(3+L\eta)}{1-L\eta}(G + 2\sqrt{\epsilon_1}).
\end{aligned} \tag{C.41}$$

Rewrite it into matrix form then we complete the proof of Lemma 13.

C.3.3 Proof of Lemma 14

Let us first recall that the definition of local AL is given below:

$$\mathcal{L}_i(\mathbf{x}_i, \mathbf{x}_0, \lambda_i) \triangleq\ f_i(\mathbf{x}_i) + \langle\lambda_i, \mathbf{x}_i - \mathbf{x}_0\rangle + \frac{1}{2\eta}\left\|\mathbf{x}_i - \mathbf{x}_0\right\|^2.$$

Similar to (C.37), we have

$$
\begin{aligned}
\mathcal{L}_i^{r+} - \bar{\mathcal{L}}_i^{r+1} &= \left\langle \lambda_i^{r+1}, \mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^{r+} \right\rangle - \left\langle \lambda_i^{r+1}, \mathbf{x}_i^{r+1} - \bar{\mathbf{x}}_0^{r+1} \right\rangle \\
&\quad + \frac{1}{2\eta} \left\| \mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^{r+} \right\|^2 - \frac{1}{2\eta} \left\| \mathbf{x}_i^{r+1} - \overline{\mathbf{x}}_0^{r+1} \right\|^2 \\
&\overset{(a)}{=} -\frac{1}{2\eta} \left\| \mathbf{x}_{0,i}^{r+} - \overline{\mathbf{x}}_0^{r+1} \right\|^2 \\
&\overset{(b)}{=} -\frac{1}{2\eta} \left\| \mathbf{x}_{0,i}^{r+} - \frac{1}{N} \sum_{j=1}^{N} \mathbf{x}_{0,j}^{r+} \right\|^2 \\
&= -\frac{1}{2\eta} \left\| \frac{1}{N} \sum_{j=1}^{N} (\mathbf{x}_{0,i}^{r+} - \mathbf{x}_{0,j}^{r+}) \right\|^2 \\
&\overset{(c)}{\geq} -\frac{1}{2\eta N} \sum_{j \neq i} \left\| \mathbf{x}_{0,i}^{r+} - \mathbf{x}_{0,j}^{r+} \right\|^2 \\
&\overset{(d)}{\geq} -\frac{N-1}{2\eta N} (\triangle \mathbf{x}_0^{r+1})^2,
\end{aligned}
\tag{C.42}
$$

where (a) follows the same argument in (C.37); in (b),we plug in the definition of $\overline{\mathbf{x}}_0^{r+1}$; in (c) we use Jensen's inequality and we bound the term with $\triangle \mathbf{x}_0^{r+1}$. Then the lemma is proved.

C.3.4 Proof of Lemma 15

Applying A8, we have:

$$
\begin{aligned}
f_i(\mathbf{x}_0^r) \leq\ & f_i(\mathbf{x}_i^r) + \langle \nabla f_i(\mathbf{x}_i^r), \mathbf{x}_0^r - \mathbf{x}_i^r \rangle + \frac{L}{2} \left\| \mathbf{x}_0^r - \mathbf{x}_i^r \right\|^2 \\
\overset{(C.9)}{=}\ & \mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_0^r, \lambda_i^r) - \langle \mathbf{e}_i^r, \mathbf{x}_0^r - \mathbf{x}_i^r \rangle - \frac{1 - L\eta}{2\eta} \left\| \mathbf{x}_0^r - \mathbf{x}_i^r \right\|^2 \\
\leq\ & \mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_0^r, \lambda_i^r) + \frac{\epsilon_1}{2L} - \frac{1 - 2L\eta}{2\eta} \left\| \mathbf{x}_0^r - \mathbf{x}_i^r \right\|^2 .
\end{aligned}
\tag{C.43}
$$

Taking an average over N agents we are able to prove Lemma 15.

C.4 Proofs for Results in Section 5.4

C.4.1 Proof of Theorem 6

Following the similar proof of Theorem 5, we first analyze the descent between each outer iteration. Notice throughout the proof, we assume that $p = 0$, that is, there is no delayed communication. It follows that the following holds:

$$
\mathbf{x}_{0,i}^{r+1} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{x}_{0,j}^{r+}, \ \ \forall i = 1, \ldots, N.
$$

We also recall that r is the (outer) stage index, and q is the local update index. First we provide a series of lemmas.

Lemma 16 *Under Assumption 8, consider FedPD with Algorithm 5 (Oracle II) as the update rule. The difference of the local AL is bounded by* (C.44).

$$\begin{aligned}
&\mathcal{L}_i^{r+1} - \mathcal{L}_i^r \le -\frac{1}{2\eta}\left\|\mathbf{x}_{0,i}^{r+1} - \mathbf{x}_{0,i}^r\right\|^2 - \left(\frac{1}{2\eta} + \frac{1}{\gamma} - L - \frac{3\eta}{\gamma^2}\right)\left\|\mathbf{x}_i^{r,Q} - \mathbf{x}_i^{r,Q-1}\right\|^2 \\
&- (\frac{1}{2\eta} + \frac{1}{\gamma} - L - 9Q^2L^2\eta)\sum_{q=1}^{Q-1}\left\|\mathbf{x}_i^{r,q} - \mathbf{x}_i^{r,q-1}\right\|^2 \\
&+ \left(9Q^2L^2\eta + \frac{3\eta}{\gamma^2}\right)\left\|\mathbf{x}_i^{r-1,Q} - \mathbf{x}_i^{r-1,Q-1}\right\|^2 + \frac{1}{2L}\sum_{q=0}^{Q-2}\|\nabla f_i(\mathbf{x}_i^{r,q}) - g_i^{r,q}\|^2 \\
&+ \left\langle \lambda_i^{r+1} + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^{r+1}), \mathbf{x}_{0,i}^{r+1} - \mathbf{x}_{0,i}^r \right\rangle \\
&+ (\frac{1}{2L} + 9\eta)\left\|g_i^{r,Q-1} - \nabla f_i(\mathbf{x}_i^{r,Q-1})\right\|^2 + 9\eta\left\|g_i^{r-1,Q-1} - \nabla f_i(\mathbf{x}_i^{r-1,Q-1})\right\|^2
\end{aligned} \tag{C.44}$$

Then we deal with the variance of the stochastic gradients.

Lemma 17 *Suppose A8 holds and the samples are randomly sampled according to* (5.7)*, consider FedPD with Algorithm 5 (Oracle II) as the update rule. The expected norm square of the difference between $g_i^{r,q+1}$ and $\nabla f_i(\mathbf{x}_i^{r,q+1})$ is bounded by*

$$\mathbb{E}\left\|g_i^{r,q+1} - \nabla f_i(\mathbf{x}_i^{r,q+1})\right\|^2 \le \frac{L^2}{B}\sum_{\tau=\{r_0,1\}}^{\{r,q+1\}}\mathbb{E}\left\|\mathbf{x}_i^\tau - \mathbf{x}_i^{\tau-1}\right\|^2. \tag{C.45}$$

Lastly we upper bound the original loss function.

Lemma 18 *Under A8 and A9, the difference between the original loss and the AL is bounded as below:*

$$\begin{aligned}
&\mathbb{E}\, f(\mathbf{x}_0^r) \\
&\le \ \mathbb{E}\,\mathcal{L}(\mathbf{x}_0^r, \mathbf{x}_1^r, \ldots, \mathbf{x}_N^r, \lambda_1^r, \ldots, \lambda_N^r) - \frac{1 - 3L\eta}{2N\eta}\sum_{i=1}^N \mathbb{E}\left\|\mathbf{x}_i^r - \mathbf{x}_0^r\right\|^2 \\
&+ \frac{(1+L\gamma)^2 + L^2\gamma^2}{4L\gamma^2}\big[\frac{1}{B}\sum_{\tau=\{r_0,1\}}^{\{r-1,Q-1\}}\mathbb{E}\left\|\mathbf{x}_i^\tau - \mathbf{x}_i^{\tau-1}\right\|^2 \\
&+ \mathbb{E}\left\|\mathbf{x}_i^{r-1,Q} - \mathbf{x}_i^{r-1,Q-1}\right\|^2\big].
\end{aligned} \tag{C.46}$$

C.4.2 Proof of Lemma 16

Let us first express the difference of the local AL as:

$$\begin{aligned}\mathcal{L}_i^{r+1} - \mathcal{L}_i^r = \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^r) - \mathcal{L}_i^r + \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^{r+1}) \\ - \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^r) \ + \mathcal{L}_i^{r+1} - \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^{r+1}),\end{aligned} \tag{C.47}$$

where the above three differences respectively correspond to the three steps in the algorithm's update steps.

Let us bound the above three differences one by one. First, note that we have the following decomposition (by using the fact that $\mathbf{x}_i^{r,Q+1} = \mathbf{x}_i^{r+1}$ and $\mathbf{x}_i^{r,1} = \mathbf{x}_i^r$):

$$\begin{aligned}&\mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^r) - \mathcal{L}_i^r \\ &= \sum_{q=1}^{Q} \left(\mathcal{L}_i(\mathbf{x}_i^{r,q+1}, \mathbf{x}_{0,i}^r, \lambda_i^r) - \mathcal{L}_i(\mathbf{x}_i^{r,q}, \mathbf{x}_{0,i}^r, \lambda_i^r) \right).\end{aligned} \tag{C.48}$$

Each term on the right hand side (RHS) of the above equality can be bounded by (see a similar arguments in (C.35)):

$$\begin{aligned}&\mathcal{L}_i(\mathbf{x}_i^{r,q+1}, \mathbf{x}_{0,i}^r, \lambda_i^r) - \mathcal{L}_i(\mathbf{x}_i^{r,q}, \mathbf{x}_{0,i}^r, \lambda_i^r) \\ &\leq \left\langle \nabla f_i(\mathbf{x}_i^{r,q}) + \lambda_i^r + \frac{1}{\eta}(\mathbf{x}^{r,q+1} - \mathbf{x}_{0,i}^r), \mathbf{x}_i^{r,q+1} - \mathbf{x}_i^{r,q} \right\rangle \\ &- \frac{1 - L\eta}{2\eta} \left\| \mathbf{x}_i^{r,q+1} - \mathbf{x}_i^{r,q} \right\|^2 \\ &\overset{(a)}{=} \left\langle \nabla f_i(\mathbf{x}_i^{r,q}) - g_i^{r,q} - \frac{1}{\gamma}(\mathbf{x}^{r,q+1} - \mathbf{x}_i^{r,q}), \mathbf{x}_i^{r,q+1} - \mathbf{x}_i^{r,q} \right\rangle \\ &- (\frac{1}{2\eta} - \frac{L}{2}) \left\| \mathbf{x}_i^{r,q+1} - \mathbf{x}_i^{r,q} \right\|^2 \\ &= \left\langle \nabla f_i(\mathbf{x}_i^{r,q}) - g_i^{r,q}, \mathbf{x}_i^{r,q+1} - \mathbf{x}_i^{r,q} \right\rangle - \\ &(\frac{1}{2\eta} + \frac{1}{\gamma} - \frac{L}{2}) \left\| \mathbf{x}_i^{r,q+1} - \mathbf{x}_i^{r,q} \right\|^2 \\ &\overset{(b)}{\leq} \frac{1}{2L} \left\| \nabla f_i(\mathbf{x}_i^{r,q}) - g_i^{r,q} \right\|^2 - (\frac{1}{2\eta} + \frac{1}{\gamma} - L) \left\| \mathbf{x}_i^{r,q+1} - \mathbf{x}_i^{r,q} \right\|^2,\end{aligned}$$

where in (a) we use the optimal condition that $\nabla_{\mathbf{x}_i} \tilde{\mathcal{L}}_i(\mathbf{x}_i^{r,q+1}, \mathbf{x}_{0,i}^r, \lambda_i^r; \mathbf{x}_i^{r,q}, g_i^{r,q}) = 0$ which gives us the following relation

$$\lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r,q+1} - \mathbf{x}_{0,i}^r) + g_i^{r,q} + \frac{1}{\gamma}(\mathbf{x}_i^{r,q+1} - \mathbf{x}_i^{r,q}) = 0; \tag{C.49}$$

in (b) we use the fact that $2\langle a, b \rangle \leq L \|a\|^2 + \frac{1}{L} \|b\|^2$. Therefore, the first difference in the RHS

of (C.47) is given by

$$\mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^r) - \mathcal{L}_i^r \leq \frac{1}{2L} \sum_{q=1}^{Q} \|\nabla f_i(\mathbf{x}_i^{r,q}) - g_i^{r,q}\|^2 - (\frac{1}{2\eta} + \frac{1}{\gamma} - L) \sum_{q=1}^{Q} \left\|\mathbf{x}_i^{r,q+1} - \mathbf{x}_i^{r,q}\right\|^2. \tag{C.50}$$

The other two differences in (C.47) can be expressed as:

$$\mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^{r+1}) - \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^r) = \eta \left\|\lambda_i^{r+1} - \lambda_i^r\right\|^2, \tag{C.51}$$

$$\mathcal{L}_i^{r+1} - \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^{r+1}) = -\frac{1}{2\eta} \left\|\mathbf{x}_{0,i}^{r+1} - \mathbf{x}_{0,i}^r\right\|^2 + \left\langle \lambda_i^{r+1} + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^{r+1}), \mathbf{x}_{0,i}^{r+1} - \mathbf{x}_{0,i}^r \right\rangle. \tag{C.52}$$

Next we bound $\left\|\lambda_i^{r+1} - \lambda_i^r\right\|^2$. Notice that the from the update rule the following holds:

$$\lambda_i^{r+1} = \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r,Q} - \mathbf{x}_{0,i}^r) \overset{\text{(C.49)}}{=} -\frac{1}{\gamma}(\mathbf{x}_i^{r,Q} - \mathbf{x}_i^{r,Q-1}) - g^{r,Q-1}. \tag{C.53}$$

Using the above property, we have

$$\begin{aligned} \left\|\lambda_i^{r+1} - \lambda_i^r\right\|^2 &= \left\| \frac{1}{\gamma}(\mathbf{x}_i^{r,Q} - \mathbf{x}_i^{r,Q-1}) + g_i^{r,Q-1} \right. \\ &\quad \left. -\frac{1}{\gamma}(\mathbf{x}_i^{r-1,Q} - \mathbf{x}_i^{r-1,Q-1}) - g_i^{r-1,Q-1} \right\|^2 \\ &\overset{(a)}{\leq} 3 \left\|g_i^{r,Q-1} - g_i^{r-1,Q-1}\right\|^2 + \frac{3}{\gamma^2} \left\|\mathbf{x}_i^{r,Q} - \mathbf{x}_i^{r,Q-1}\right\|^2 \\ &\quad + \frac{3}{\gamma^2} \left\|\mathbf{x}_i^{r-1,Q} - \mathbf{x}_i^{r-1,Q-1}\right\|^2. \end{aligned} \tag{C.54}$$

where in (a) we apply Cauchy-Schwarz inequality. Next we bound $\left\|g_i^{r,Q-1} - g_i^{r-1,Q-1}\right\|^2$ by (C.55),

$$\begin{aligned} &\left\|g_i^{r,Q-1} - g_i^{r-1,Q-1}\right\|^2 \\ &= \left\|g_i^{r,Q-1} - \nabla f_i(\mathbf{x}_i^{r,Q-1}) + \nabla f_i(\mathbf{x}_i^{r,Q-1}) - \nabla f_i(\mathbf{x}_i^{r-1,Q-1}) + \nabla f_i(\mathbf{x}_i^{r-1,Q-1}) - g_i^{r-1,Q-1}\right\|^2 \end{aligned} \tag{C.55}$$

$$\begin{aligned} &\overset{(a)}{\leq} 3 \left\|g_i^{r,Q-1} - \nabla f_i(\mathbf{x}_i^{r,Q-1})\right\|^2 + 3 \left\|g_i^{r-1,Q-1} - \nabla f_i(\mathbf{x}_i^{r-1,Q-1})\right\|^2 + 3L^2 \left\|\mathbf{x}_i^{r,Q-1} - \mathbf{x}_i^{r-1,Q-1}\right\|^2 \\ &\overset{(b)}{\leq} 3 \left\|g_i^{r,Q-1} - \nabla f_i(\mathbf{x}_i^{r,Q-1})\right\|^2 + 3 \left\|g_i^{r-1,Q-1} - \nabla f_i(\mathbf{x}_i^{r-1,Q-1})\right\|^2 + 3Q^2L^2 \sum_{q=1}^{Q-1} \left\|\mathbf{x}_i^{r,q} - \mathbf{x}_i^{r,q-1}\right\|^2 \end{aligned}$$

$$+ 3Q^2L^2 \left\|\mathbf{x}_i^{r-1,Q} - \mathbf{x}_i^{r-1,Q-1}\right\|^2,$$

where in (a) and (b) we both apply Cauchy-Schwarz inequality, in (a) we use A8 to the last term and in (b) we notice $\mathbf{x}_i^{r-1,Q} = \mathbf{x}_i^{r,0}$.

Substitute (C.55) to (C.54) and sum the three parts, we have (C.56), which complete the proof of Lemma 16.

$$\begin{aligned}
\mathcal{L}_i^{r+1} - \mathcal{L}_i^r \leq & -\frac{1}{2\eta}\left\|\mathbf{x}_{0,i}^{r+1} - \mathbf{x}_{0,i}^r\right\|^2 - (\frac{1}{2\eta} + \frac{1}{\gamma} - L - \frac{3\eta}{\gamma^2})\left\|\mathbf{x}_i^{r,Q} - \mathbf{x}_i^{r,Q-1}\right\|^2 \\
& - (\frac{1}{2\eta} + \frac{1}{\gamma} - L - 9Q^2L^2\eta)\sum_{q=1}^{Q-1}\left\|\mathbf{x}_i^{r,q} - \mathbf{x}_i^{r,q-1}\right\|^2 \\
& + (9Q^2L^2\eta + \frac{3\eta}{\gamma^2})\left\|\mathbf{x}_i^{r-1,Q} - \mathbf{x}_i^{r-1,Q-1}\right\|^2 + \frac{1}{2L}\sum_{q=0}^{Q-2}\|\nabla f_i(\mathbf{x}_i^{r,q}) - g_i^{r,q}\|^2 \\
& + \left\langle \lambda_i^{r+1} + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^{r+1}), \mathbf{x}_{0,i}^{r+1} - \mathbf{x}_{0,i}^r \right\rangle \\
& + (\frac{1}{2L} + 9\eta)\left\|g_i^{r,Q-1} - \nabla f_i(\mathbf{x}_i^{r,Q-1})\right\|^2 + 9\eta\left\|g_i^{r-1,Q-1} - \nabla f_i(\mathbf{x}_i^{r-1,Q-1})\right\|^2
\end{aligned} \tag{C.56}$$

C.4.3 Proof of Lemma 17

To study $\mathbb{E}\left\|g_i^{r,q} - \nabla f_i(\mathbf{x}_i^{r,q})\right\|^2$, we denote the latest iteration before r that computes full gradients as r_0. That is, in r_0 we have $g_i^{r_0,0} = \nabla f_i(\mathbf{x}_i^{r_0,0})$. By the description of the algorithm we know

$$r_0 = kI, \quad k \in \mathbb{N}, \quad rQ + q - r_0Q \leq IQ.$$

That is, r_0 is a multiple of I and there is no more than IQ local update steps between step $\{r_0, 0\}$ and step $\{r, q\}$. By the update rule of $g_i^{r,q}$, we have

$$\begin{aligned}
& g_i^{r,q+1} - \nabla f_i(\mathbf{x}_i^{r,q+1}) \\
& = g_i^{r,q} - \nabla f_i(\mathbf{x}_i^{r,q+1}) + \frac{1}{B}\sum_{b=1}^{B}(h_i(\mathbf{x}_i^{r,q+1}; \xi_{i,b}^{r,q}) - h_i(\mathbf{x}_i^{r,q}; \xi_{i,b}^{r,q})).
\end{aligned} \tag{C.57}$$

Take expectation on both sides, we have

$$\begin{aligned}
& \mathbb{E}_{\{\xi_{i,b}^{r,q}\}_{b=1}^B}[g_i^{r,q+1} - \nabla f_i(\mathbf{x}_i^{r,q+1})] \\
& = g_i^{r,q} - \nabla f_i(\mathbf{x}_i^{r,q+1}) \\
& \quad + \mathbb{E}_{\{\xi_{i,b}^{r,q}\}_{b=1}^B}[\frac{1}{B}\sum_{b=1}^{B}(h_i(\mathbf{x}_i^{r,q+1}; \xi_{i,b}^{r,q}) - h_i(\mathbf{x}_i^{r,q}; \xi_{i,b}^{r,q}))] \\
& = g_i^{r,q} - \nabla f_i(\mathbf{x}_i^{r,q+1}) + \nabla f_i(\mathbf{x}_i^{r,q+1}) - \nabla f_i(\mathbf{x}_i^{r,q}) \\
& = g_i^{r,q} - \nabla f_i(\mathbf{x}_i^{r,q}).
\end{aligned} \tag{C.58}$$

By using the fact that $\mathbb{E}[X^2] = [\mathbb{E}\,X]^2 + \mathbb{E}[[X - \mathbb{E}\,X]^2]$ and substitute (C.58) we obtain (C.59),

$$
\begin{aligned}
&\mathbb{E}_{\{\xi_{i,b}^{r,q}\}_{b=1}^{B}} \left\| g_i^{r,q+1} - \nabla f_i(\mathbf{x}_i^{r,q+1}) \right\|^2 \\
&= \left\| \mathbb{E}_{\{\xi_{i,b}^{r,q}\}_{b=1}^{B}} [g_i^{r,q+1} - \nabla f_i(\mathbf{x}_i^{r,q+1})] \right\|^2 \\
&\quad + \mathbb{E}_{\{\xi_{i,b}^{r,q}\}_{b=1}^{B}} \left\| g_i^{r,q+1} - \nabla f_i(\mathbf{x}_i^{r,q+1}) - \mathbb{E}_{\{\xi_{i,b}^{r,q}\}_{b=1}^{B}} [g_i^{r,q+1} - \nabla f_i(\mathbf{x}_i^{r,q+1})] \right\|^2 \\
&\overset{(C.58)}{=} \left\| g_i^{r,q} - \nabla f_i(\mathbf{x}_i^{r,q}) \right\|^2 \\
&\quad + \mathbb{E}_{\{\xi_{i,b}^{r,q}\}_{b=1}^{B}} \left\| \frac{1}{B} \sum_{b=1}^{B} (h_i(\mathbf{x}_i^{r,q+1}; \xi_{i,b}^{r,q} - h_i(\mathbf{x}_i^{r,q}; \xi_{i,b}^{r,q})) - \nabla f_i(\mathbf{x}_i^{r,q+1}) + \nabla f_i(\mathbf{x}_i^{r,q}) \right\|^2 \\
&\overset{(a)}{\leq} \left\| g_i^{r,q} - \nabla f_i(\mathbf{x}_i^{r,q}) \right\|^2 + \frac{1}{B^2} \sum_{b=1}^{B} \mathbb{E}_{\{\xi_{i,b}^{r,q}\}_{b=1}^{B}} \left\| h_i(\mathbf{x}_i^{r,q+1}; \xi_{i,b}^{r,q}) - h_i(\mathbf{x}_i^{r,q}; \xi_{i,b}^{r,q})) \right\|^2 \\
&\overset{(b)}{\leq} \left\| g_i^{r,q} - \nabla f_i(\mathbf{x}_i^{r,q}) \right\|^2 + \frac{L^2}{B} \left\| \mathbf{x}_i^{r,q+1} - \mathbf{x}_i^{r,q} \right\|^2 .
\end{aligned}
\tag{C.59}
$$

where (a) comes form the fact that we view $h_i(\mathbf{x}_i^{r,q+1}; \xi_{i,b}^{r,q}) - h_i(\mathbf{x}_i^{r,q}; \xi_{i,b}^{r,q})$ as X and by identically random sampling strategy we have $\mathbb{E}\,X = \nabla f_i(\mathbf{x}_i^{r,q+1}) - \nabla f_i(\mathbf{x}_i^{r,q})$ and $\mathbb{E}[[X - \mathbb{E}\,X]^2 \leq \mathbb{E}[X]^2$, in (b) we use A8.

Iteratively taking expectation until $\{r, q\} = \{r_0, 0\}$, we have

$$
\mathbb{E} \left\| g_i^{r,q+1} - \nabla f_i(\mathbf{x}_i^{r,q+1}) \right\|^2 \leq \frac{L^2}{B} \sum_{\tau=\{r_0,1\}}^{\{r,q+1\}} \mathbb{E} \left\| \mathbf{x}_i^{\tau} - \mathbf{x}_i^{\tau-1} \right\|^2 , \tag{C.60}
$$

which completes the proof.

C.4.4 Proof of Lemma 18

Applying A8, we have

$$
\begin{aligned}
f_i(\mathbf{x}_0^r) &\leq f_i(\mathbf{x}_i^r) + \langle \nabla f_i(\mathbf{x}_i^r), \mathbf{x}_0^r - \mathbf{x}_i^r \rangle + \frac{L}{2} \left\| \mathbf{x}_0^r - \mathbf{x}_i^r \right\|^2 \\
&= \mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_0^r, \lambda_i^r) - \langle \nabla f_i(\mathbf{x}_i^r) + \lambda_i^r, \mathbf{x}_0^r - \mathbf{x}_i^r \rangle \\
&\quad - \frac{1 - L\eta}{2\eta} \left\| \mathbf{x}_0^r - \mathbf{x}_i^r \right\|^2 \\
&\leq \mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_0^r, \lambda_i^r) + \frac{1}{4L} \left\| \nabla f_i(\mathbf{x}_i^r) + \lambda_i^r \right\|^2 \\
&\quad - \frac{1 - 3L\eta}{2\eta} \left\| \mathbf{x}_0^r - \mathbf{x}_i^r \right\|^2 .
\end{aligned}
\tag{C.61}
$$

Then notice $\mathbf{x}_i^r = \mathbf{x}_i^{r-1,Q}$ and apply (C.53), we can bound $\mathbb{E}\left\|\nabla f_i(\mathbf{x}_i^r) + \lambda_i^r\right\|^2$ by the following:

$$
\begin{aligned}
&\mathbb{E}\left\|\nabla f_i(\mathbf{x}_i^r) + \lambda_i^r\right\|^2 \\
&\overset{(C.53)}{=} \mathbb{E}\left\|\nabla f_i(\mathbf{x}_i^{r-1,Q}) - g_i^{r-1,Q-1} - \frac{1}{\gamma}(\mathbf{x}_i^{r-1,Q} - \mathbf{x}_i^{r-1,Q-1})\right\|^2 \\
&\overset{(a)}{\leq} (1 + \frac{(1+L\gamma)^2}{L^2\gamma^2})\,\mathbb{E}\left\|\nabla f_i(\mathbf{x}_i^{r-1,Q-1}) - g_i^{r-1,Q-1}\right\|^2 \\
&+ (1 + \frac{L^2\gamma^2}{(1+L\gamma)^2})(1 + \frac{1}{L\gamma})\,\mathbb{E}\left\|\nabla f_i(\mathbf{x}_i^{r-1,Q}) - \nabla f_i(\mathbf{x}_i^{r-1,Q-1})\right\|^2 \\
&+ \frac{(1 + \frac{L^2\gamma^2}{(1+L\gamma)^2})(1+L\gamma)}{\gamma^2}\,\mathbb{E}\left\|\mathbf{x}_i^{r-1,Q} - \mathbf{x}_i^{r-1,Q-1}\right\|^2 \\
&\overset{(b)}{\leq} \frac{(1+L\gamma)^2 + L^2\gamma^2}{B\gamma^2} \sum_{\tau=\{r_0,1\}}^{\{r-1,Q-1\}} \mathbb{E}\left\|\mathbf{x}_i^{\tau} - \mathbf{x}_i^{\tau-1}\right\|^2 \\
&+ (1 + \frac{L^2\gamma^2}{(1+L\gamma)^2})\left((1+\frac{1}{L\gamma})L^2 + \frac{1+L\gamma}{\gamma^2}\right) \\
&\times \mathbb{E}\left\|\mathbf{x}_i^{r-1,Q} - \mathbf{x}_i^{r-1,Q-1}\right\|^2 \\
&= \frac{(1+L\gamma)^2 + L^2\gamma^2}{B\gamma^2} \sum_{\tau=\{r_0,1\}}^{\{r-1,Q-1\}} \mathbb{E}\left\|\mathbf{x}_i^{\tau} - \mathbf{x}_i^{\tau-1}\right\|^2 \\
&+ \frac{(1+L\gamma)^2 + L^2\gamma^2}{\gamma^2}\,\mathbb{E}\left\|\mathbf{x}_i^{r-1,Q} - \mathbf{x}_i^{r-1,Q-1}\right\|^2,
\end{aligned}
\tag{C.62}
$$

where in (a) we apply Cauchy-Schwarz inequality twice:

$$
\begin{aligned}
\|x+y+z\|^2 &\leq (1+\frac{1}{a})\,\|x\|^2 + (1+a)\,\|y+z\|^2 \\
&\leq (1+\frac{1}{a})\,\|x\|^2 + (1+a)(1+b)\,\|y\|^2 + (1+a)(1+\frac{1}{b})\,\|z\|^2;
\end{aligned}
$$

in (b) we apply Lemma 17 to the first term and apply A8 to the second term.

Substitute (C.62) to (C.61) and average over the agents, Lemma 18 is proved.

C.4.5 Proof of Theorem 6

By the update step of $\mathbf{x}_0^r$, following (C.17) we have

$$
\begin{aligned}
&\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla_{\mathbf{x}_{0,i}}\mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_{0,i}^r, \lambda_i^r)\right\| \\
&= \left\|\frac{1}{N}\sum_{i=1}^{N}(\frac{1}{\eta}(\mathbf{x}_i^r - \mathbf{x}_{0,i}^r) + \lambda_i^r)\right\| = 0.
\end{aligned}
$$

We also have (C.63)

$$\begin{aligned}
\left\|\nabla\mathcal{L}_i(\mathbf{x}_i^r,\mathbf{x}_{0,i}^r,\lambda_i^r)\right\|^2 =&\ \left\|\nabla_{\mathbf{x}_i}\mathcal{L}_i(\mathbf{x}_i^r,\mathbf{x}_{0,i}^r,\lambda_i^r)\right\|^2+\left\|\nabla_{\lambda_i}\mathcal{L}_i(\mathbf{x}_i^r,\mathbf{x}_{0,i}^r,\lambda_i^r)\right\|^2\\
=&\ \left\|\nabla f_i(\mathbf{x}_i^r)+\lambda_i^r+\frac{1}{\eta}(\mathbf{x}_i^r-\mathbf{x}_{0,i}^r)\right\|^2+\left\|\mathbf{x}_i^r-\mathbf{x}_{0,i}^r\right\|^2\\
\overset{(a)}{=}&\ \left\|\nabla f_i(\mathbf{x}_i^r)-g_i^{r,0}-\frac{\eta+\gamma}{\eta\gamma}(\mathbf{x}_i^{r,1}-\mathbf{x}_i^r)\right\|^2+\left\|\mathbf{x}_i^r-\mathbf{x}_{0,i}^r+\mathbf{x}_{0,i}^{r-1}-\mathbf{x}_{0,i}^{r-1}\right\|^2\\
\leq&\ \left\|\nabla f_i(\mathbf{x}_i^r)-g_i^{r,0}-\frac{\eta+\gamma}{\eta\gamma}(\mathbf{x}_i^{r,1}-\mathbf{x}_i^r)\right\|^2+2\left\|\mathbf{x}_i^r-\mathbf{x}_{0,i}^{r-1}\right\|^2+2\left\|\mathbf{x}_{0,i}^r-\mathbf{x}_{0,i}^{r-1}\right\|^2\\
\leq&\ 2\left\|\nabla f_i(\mathbf{x}_i^r)-g_i^{r,0}\right\|^2+2(\frac{\eta+\gamma}{\eta\gamma})^2\left\|\mathbf{x}_i^{r,1}-\mathbf{x}_i^r\right\|^2+2\eta^2\left\|\lambda_i^r-\lambda_i^{r-1}\right\|^2+2\left\|\mathbf{x}_{0,i}^r-\mathbf{x}_{0,i}^{r-1}\right\|^2.
\end{aligned}\tag{C.63}$$

where in (a), the first term is obtained by plugging in (C.53) given below

$$\lambda_i^r=-g_i^{r,0}-\frac{1}{\gamma}(\mathbf{x}_i^{r,1}-\mathbf{x}_i^r)-\frac{1}{\eta}(\mathbf{x}_i^{r,1}-\mathbf{x}_{0,i}^r).$$

Next we take expectation and substitute (C.54), (C.55) to obtain (C.64),

$$\begin{aligned}
&\mathbb{E}\left\|\nabla\mathcal{L}_i(\mathbf{x}_i^r,\mathbf{x}_{0,i}^r,\lambda_i^r)\right\|^2\leq 2\,\mathbb{E}\left\|\nabla f_i(\mathbf{x}_i^r)-g_i^{r,0}\right\|^2+2(\frac{\eta+\gamma}{\eta\gamma})^2\,\mathbb{E}\left\|\mathbf{x}_i^{r,1}-\mathbf{x}_i^r\right\|^2+2\,\mathbb{E}\left\|\mathbf{x}_{0,i}^r-\mathbf{x}_{0,i}^{r-1}\right\|^2\\
&+\frac{6\eta^2}{\gamma^2}(\gamma^2\,\mathbb{E}\left\|g_i^{r-1,Q-1}-g_i^{r-2,Q-1}\right\|^2+\mathbb{E}\left\|\mathbf{x}_i^{r-1,Q}-\mathbf{x}_i^{r-1,Q-1}\right\|^2+E\left\|\mathbf{x}_i^{r-2,Q}-\mathbf{x}_i^{r-2,Q-1}\right\|^2)\\
&\overset{(a)}{\leq}\frac{2L^2}{B}\sum_{\tau=\{r_0,1\}}^{\{r,0\}}\mathbb{E}\left\|\mathbf{x}_i^{\tau}-\mathbf{x}_i^{\tau-1}\right\|^2+2(\frac{\eta+\gamma}{\eta\gamma})^2\,\mathbb{E}\left\|\mathbf{x}_i^{r,1}-\mathbf{x}_i^r\right\|^2+2\,\mathbb{E}\left\|\mathbf{x}_{0,i}^r-\mathbf{x}_{0,i}^{r-1}\right\|^2\\
&+\frac{6\eta^2}{\gamma^2}(\mathbb{E}\left\|\mathbf{x}_i^{r-1,Q}-\mathbf{x}_i^{r-1,Q-1}\right\|^2+\mathbb{E}\left\|\mathbf{x}_i^{r-2,Q}-\mathbf{x}_i^{r-2,Q-1}\right\|^2)\\
&+18\eta^2\left(\mathbb{E}\left\|g_i^{r-1,Q-1}-\nabla f_i(\mathbf{x}_i^{r-1,Q-1})\right\|^2+\mathbb{E}\left\|g_i^{r-2,Q-1}-\nabla f_i(\mathbf{x}_i^{r-2,Q-1})\right\|^2\right)\\
&+18\eta^2Q^2L^2\left(\sum_{q=1}^{Q-1}\mathbb{E}\left\|\mathbf{x}_i^{r-1,q}-\mathbf{x}_i^{r-1,q-1}\right\|^2+\mathbb{E}\left\|\mathbf{x}_i^{r-2,Q}-\mathbf{x}_i^{r-2,Q-1}\right\|^2\right).
\end{aligned}\tag{C.64}$$

where we substitute Lemma 17 and (C.55) in (a).

Taking expectation of (C.44), summing over $r=0$ to $r=T-1$ and average over the agents, we obtain (C.65)

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^N\mathbb{E}[\mathcal{L}_i(\mathbf{x}_i^T,\mathbf{x}_{0,i}^T,\lambda_i^T)-\mathcal{L}_i(\mathbf{x}_i^0,\mathbf{x}_{0,i}^0,\lambda_i^0)]\\
&\leq-\frac{1}{2\eta}\sum_{r=0}^{T-1}\mathbb{E}\left\|\mathbf{x}_0^{r+1}-\mathbf{x}_0^r\right\|^2-(\frac{1}{2\eta}+\frac{1}{\gamma}-L-\frac{6\eta}{\gamma^2}-9Q^2L^2\eta)\frac{1}{N}\sum_{i=1}^N\sum_{q=0}^{Q-1}\sum_{r=0}^{T-1}\mathbb{E}\left\|\mathbf{x}_i^{r,q+1}-\mathbf{x}_i^{r,q-1}\right\|^2
\end{aligned}$$

$$
\begin{aligned}
&+(\frac{1}{2L}+18\eta)\frac{1}{N}\sum_{i=1}^{N}\sum_{r=0}^{T-1}\sum_{q=0}^{Q-1}\mathbb{E}\left\|\nabla f_i(\mathbf{x}_i^{r,q})-g_i^{r,q}\right\|^2\\
&\quad+\sum_{r=0}^{T-1}\frac{1}{N}\mathbb{E}\left\langle\sum_{i=1}^{N}(\lambda_i^{r+1}+\frac{1}{\eta}(\mathbf{x}_i^{r+1}-\mathbf{x}_{0,i}^{r+1})),\mathbf{x}_{0,i}^{r+1}-\mathbf{x}_{0,i}^{r}\right\rangle\\
&\overset{(a)}{\leq}-(\frac{1}{2\eta}+\frac{1}{\gamma}-L-\frac{6\eta}{\gamma^2}-9Q^2L^2\eta)\frac{1}{N}\sum_{i=1}^{N}\sum_{q=0}^{Q-1}\sum_{r=0}^{T-1}\mathbb{E}\left\|\mathbf{x}_i^{r,q+1}-\mathbf{x}_i^{r,q-1}\right\|^2-\frac{1}{2\eta}\sum_{r=0}^{T-1}\mathbb{E}\left\|\mathbf{x}_0^{r+1}-\mathbf{x}_0^{r}\right\|^2\\
&\quad+\frac{(1+18L\eta)LIQ}{2B}\frac{1}{N}\sum_{i=1}^{N}\sum_{r=0}^{T-1}\sum_{q=0}^{Q-1}\mathbb{E}\left\|\mathbf{x}_i^{r,q+1}-\mathbf{x}_i^{r,q-1}\right\|^2\\
&=-\frac{C_{10}}{N}\sum_{i=1}^{N}\sum_{q=0}^{Q-1}\sum_{r=0}^{T-1}\mathbb{E}\left\|\mathbf{x}_i^{r,q+1}-\mathbf{x}_i^{r,q-1}\right\|^2-\frac{1}{2\eta}\sum_{r=0}^{T-1}\mathbb{E}\left\|\mathbf{x}_0^{r+1}-\mathbf{x}_0^{r}\right\|^2. \qquad \text{(C.65)}
\end{aligned}
$$

where in (a) we apply Lemma 17 and (C.17).

Finally, in the last equation of (C.65), we have defined the constant C_{10} as

$$C_{10}:=\frac{1}{2\eta}+\frac{1}{\gamma}-L-\frac{6\eta}{\gamma^2}-9Q^2L^2\eta-\frac{(1+18L\eta)LIQ}{2B}.$$

Then by taking expectation and applying Lemma 18, we obtain

$$
\begin{aligned}
&\mathbb{E}[f(\mathbf{x}_0^T)-f(\mathbf{x}_0^0)]\\
&\leq\ -\frac{C_{10}-\frac{(1+L\gamma)^2+L^2\gamma^2}{4BL\gamma^2}}{N}\sum_{i=1}^{N}\sum_{q=0}^{Q-1}\sum_{r=0}^{T-1}\mathbb{E}\left\|\mathbf{x}_i^{r,q+1}-\mathbf{x}_i^{r,q-1}\right\|^2\\
&-\frac{1}{2\eta}\sum_{r=0}^{T-1}\mathbb{E}\left\|\mathbf{x}_0^{r+1}-\mathbf{x}_0^{r}\right\|^2,
\end{aligned}
$$

where by the initialization that $\mathbf{x}_i^0=\mathbf{x}_0^0$ we have $f_(\mathbf{x}_0^0)=\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i(\mathbf{x}_i^0,\mathbf{x}_{0,i}^0,\lambda_i^0)$.

Combine (C.64) and (C.66), we can find a positive constant C_{11} satisfying

$$C_{11}\leq\min\left\{C_{12}/C_{13},1/(4\eta)\right\}, \qquad \text{(C.66)}$$

where we have defined

$$
\begin{aligned}
C_{12}&\triangleq C_{10}-\frac{(1+L\gamma)^2+L^2\gamma^2}{4BL\gamma^2},\\
C_{13}&\triangleq Q\left(2(\frac{\eta+\gamma}{\eta\gamma})^2+\frac{2I(1+18\eta^2)L^2}{B}\right)\\
&\quad+Q\left(\frac{3L(1+9L\eta)\eta^2}{2B\gamma^2}+18Q^2L^2\eta^2\right) \qquad \text{(C.67)}
\end{aligned}
$$

so that the following holds

$$\begin{aligned}
&\frac{C_{11}}{NT}\sum_{r=0}^{T}\sum_{i=1}^{N}\mathbb{E}\left\|\nabla\mathcal{L}_i(\mathbf{x}_i^r,\mathbf{x}_{0,i}^r,\lambda_i^r)\right\|^2\\
&\leq\frac{C_{10}-\frac{(1+L\gamma)^2+L^2\gamma^2}{4BL\gamma^2}}{NT}\sum_{i=1}^{N}\sum_{q=0}^{Q-1}\sum_{r=0}^{T-1}\mathbb{E}\left\|\mathbf{x}_i^{r,q+1}-\mathbf{x}_i^{r,q-1}\right\|^2\\
&\quad+\frac{1}{2\eta T}\sum_{r=0}^{T-1}\mathbb{E}\left\|\mathbf{x}_0^{r+1}-\mathbf{x}_0^r\right\|^2\\
&\leq\frac{1}{T}(f(\mathbf{x}_0^0)-\mathbb{E}\,f(\mathbf{x}_0^T))\leq\frac{1}{T}(f(\mathbf{x}_0^0)-f(\mathbf{x}^\star)).
\end{aligned}\tag{C.68}$$

Similar to the proof of Theorem 5, we can bound $\|\nabla f(\mathbf{x}_0^r)\|^2$ by $\frac{1}{N}\sum_{i=1}^{N}\|\nabla\mathcal{L}_i(\mathbf{x}_i^r,\mathbf{x}_0^r,\lambda_i^r)\|^2$, therefore Theorem 6 is proved.

Note that during the proof we need the following constants C_9, C_{10}, C_{11} in (C.69) to be positive

$$\begin{aligned}
&C_9=4L^2/C_{11},\quad C_{10}=\frac{1}{2\eta}+\frac{1}{\gamma}-L-\frac{6\eta}{\gamma^2}-9Q^2L^2\eta-\frac{(1+18L\eta)LIQ}{2B},\\
&C_{11}\leq\min\left\{\frac{\left(C_{10}-\frac{(1+L\gamma)^2+L^2\gamma^2}{4BL\gamma^2}\right)}{Q\left(2(\frac{\eta+\gamma}{\eta\gamma})^2+\frac{2I(1+18\eta^2)L^2}{B}+\frac{3L(1+9L\eta)\eta^2}{2B\gamma^2}+18Q^2L^2\eta^2\right)},\frac{1}{4\eta}\right\}.
\end{aligned}\tag{C.69}$$

By selecting $\gamma>\frac{5}{B\sqrt{L}}\eta$, and $0<\eta<\frac{1}{3(Q+\sqrt{QI/B})L}$, this is guaranteed.

C.5 Examples of Cost Functions Satisfying A11

In this part, we provide a commonly used function that satisfies A11.

Logistic Regression. Consider the case where the k^{th} sample $\xi_{i,k}$ in data set $\mathcal{D}_i$ consist of a feature vector $\mathbf{a}_k$ and a scalar label b_k. The feature vector $\mathbf{a}_k$ has the same length as $\mathbf{x}$ and b_k is a scalar in $\mathbb{R}$. Then the loss function of a logistic regression problem is expressed as

$$f_i(\mathbf{x})=\frac{1}{|\mathcal{D}_i|}\sum_{(\mathbf{a}_k,b_k)\in\mathcal{D}_i}\frac{1}{1+\exp(b_k-\mathbf{a}_k^T\mathbf{x})}.\tag{C.70}$$

The gradient of this loss function is

$$\nabla f_i(\mathbf{x})=\frac{1}{|\mathcal{D}_i|}\sum_{(\mathbf{a}_k,b_k)\in\mathcal{D}_i}\frac{\mathbf{a}_k\exp(b_k-\mathbf{a}_k^T\mathbf{x})}{(1+\exp(b_k-\mathbf{a}_k^T\mathbf{x}))^2}.\tag{C.71}$$

Define the scalar $\frac{\exp(b_k-\mathbf{a}_k^T\mathbf{x})}{(1+\exp(b_k-\mathbf{a}_k^T\mathbf{x}))^2}$ as $v(\mathbf{a}_k,b_k,\mathbf{x})$, we have $v(\mathbf{a}_k,b_k,\mathbf{x})\in(0,1),\ \forall x,\mathbf{a}_k,b_k$. Further stack $v(\mathbf{a}_k,b_k,\mathbf{x})$ as $\mathbf{v}(\mathcal{D}_i,\mathbf{x})$, that is $\mathbf{v}(\mathcal{D}_i,\mathbf{x})=[v(\mathbf{a}_1,b_1,\mathbf{x});\ldots,;v(\mathbf{a}_{|\mathcal{D}_i|},b_{|\mathcal{D}_i|},\mathbf{x})]$. Further we define A_i as the stacked matrix of all $\mathbf{a}_k\in\mathcal{D}_i$ (i.e., $A_i=[\mathbf{a}_1,\ldots,\mathbf{a}_{|\mathcal{D}_i|}]$), then we can

express $\nabla f_i(\mathbf{x})$ as

$$\nabla f_i(\mathbf{x}) = \frac{1}{|\mathcal{D}_i|} A_i \mathbf{v}(\mathcal{D}_i, \mathbf{x}). \tag{C.72}$$

The difference between the gradients of f_i and f_j is

$$\begin{aligned} \|\nabla f_i(\mathbf{x}) - \nabla f_j(\mathbf{x})\| &= \left\| \frac{1}{|\mathcal{D}_i|} A_i \mathbf{v}(\mathcal{D}_i, \mathbf{x}) - \frac{1}{|\mathcal{D}_j|} A_j \mathbf{v}(\mathcal{D}_j, \mathbf{x}) \right\| \\ &\leq \frac{1}{|\mathcal{D}_i|} \|A_i \mathbf{v}(\mathcal{D}_i, \mathbf{x})\| + \frac{1}{|\mathcal{D}_j|} \|A_j \mathbf{v}(\mathcal{D}_j, \mathbf{x})\|. \end{aligned} \tag{C.73}$$

As $v(\mathbf{a}, b, \mathbf{x}) \in (0, 1)$, we know $\|\mathbf{v}(\mathcal{D}_i, \mathbf{x})\| \leq \|[1, \ldots, 1]\| = \sqrt{|\mathcal{D}_i|}$, which implies:

$$\|A_i\| \geq \frac{\|A_i \mathbf{v}(\mathcal{D}_i, \mathbf{x})\|}{\|\mathbf{v}(\mathcal{D}_i, \mathbf{x})\|} \geq \frac{\|A_i \mathbf{v}(\mathcal{D}_i, \mathbf{x})\|}{\sqrt{|\mathcal{D}_i|}}.$$

Utilizing the above inequality in (C.73), we obtain:

$$\begin{aligned} \|\nabla f_i(\mathbf{x}) - \nabla f_j(\mathbf{x})\| &\leq \frac{1}{|\mathcal{D}_i|} \|A_i \mathbf{v}(\mathcal{D}_i, \mathbf{x})\| + \frac{1}{|\mathcal{D}_j|} \|A_j \mathbf{v}(\mathcal{D}_j, \mathbf{x})\| \\ &\leq \frac{1}{\sqrt{|\mathcal{D}_i|}} \|A_i\| + \frac{1}{\sqrt{|\mathcal{D}_j|}} \|A_j\|. \end{aligned} \tag{C.74}$$

So we can define $G = \max_{i,j} \left\{ \frac{1}{\sqrt{|\mathcal{D}_i|}} \|A_i\| + \frac{1}{\sqrt{|\mathcal{D}_j|}} \|A_j\| \right\}$ which is a finite constant. Note that the above analysis holds true for any D_i and $\mathbf{x}$. Note that with finer analysis we can obtain better bounds for G.

Hyperbolic Tangent. Similar to logistic regression, we can also show that A11 holds for hyperbolic tangent function which is commonly used in neural network models. First, notice that the hyperbolic tangent is a rescaled version of logistic regression:

$$\begin{aligned} \tanh(b_k - \mathbf{a}_k^T \mathbf{x}) &= \frac{\exp(b_k - \mathbf{a}_k^T \mathbf{x}) - \exp(\mathbf{a}_k^T \mathbf{x} - b_k)}{\exp(b_k - \mathbf{a}_k^T \mathbf{x}) + \exp(\mathbf{a}_k^T \mathbf{x} - b_k)} \\ &= \frac{2}{1 + \exp(2(b_k - \mathbf{a}_k^T \mathbf{x}))} - 1. \end{aligned}$$

Therefore we have

$$\nabla_{\mathbf{x}} \tanh(b_k - \mathbf{a}_k^T \mathbf{x}) = 4 \nabla_{\mathbf{x}} \frac{1}{1 + \exp(2(b_k - \mathbf{a}_k^T \mathbf{x}))}.$$

So, G for tanh is 4 times that applicable to the logistic regression problem. Note that this analysis can further cover a wide range of neural network training problems that uses cross entropy loss and sigmoidal activation functions (e.g. MLP, CNN and RNN).

Special Case in Linear Regression. Consider the linear regression problem

$$f_i(x) = \frac{1}{2} \|A_i \mathbf{x} + \mathbf{b}_i\|^2, i = 1, \ldots, N.$$

We have

$$\nabla f_i(\mathbf{x}) = A_i^T A_i \mathbf{x} + A_i^T \mathbf{b}_i.$$

Then if the feature A_i's satisfy $A_i^T A_i = A_j^T A_j, \forall\ i \neq j$, we have

$$G = \max_{i,j} \left| A_i^T \mathbf{b}_i - A_j^T \mathbf{b}_j \right|.$$

C.6 Proof of Claim 1

The proof is related to techniques developed in classical and recent works that characterize lower bounds for first-order methods in centralized [111, 112] and decentralized [19, 20] settings. Technically, our computational/communication model is *different* compared to the aforementioned works, since we allow arbitrary number of local processing iterations, and we have a central aggregator. The difference here is that our goal is *not* to show the lower bounds on the number of total (centralized) gradient access, nor to show the optimal graph dependency. The main point we would like to make is that there exist constructions of *local* functions f_i's such that *no matter* how many times that local first-order processing is performed, without *communication* and *aggregation*, no significant progress can be made in reducing the stationarity gap of the original problem.

For notational simplicity, we will assume that the full local gradients $\{\nabla f_i(x_i^r)\}$ can be evaluated. Later we will comment on how to extend this result to enable access to the sample gradients $\nabla F(x_i^r; \xi_i)$. In particular, we consider the following slightly simplified model for now:

$$x^r = V^r(\{x_i^{r-1,Q}\}_{i=1}^N),\ x_i^{r,0} = x^r,\ \ \forall\ i \in [N], \tag{C.75a}$$

$$x_i^{r,q} \in W_i^r \left(\{x_i^{r,k}, \left\{ \nabla f_i(x_i^{r,k}) \right\}_{r=0:t}^{k=0:q-1} \right), q \in [Q],\ \forall\ i. \tag{C.75b}$$

C.6.1 Notation.

In this section, we will call each r a "stage," and call each local iteration q an "iteration." We use x to denote the variable located at the server. We use x_i (and sometimes x_q) to denote the local variable at node i, and use $x_i[j]$ and $x_i[k]$ to denote its jth and kth elements, respectively. We use $g_i(\cdot)$ and $f_i(\cdot)$ to denote some functions related to node i, and $g(\cdot)$ and $f(\cdot)$ to denote the average functions of g_i's and f_i's, respectively. We use N to denote the total number of nodes.

C.6.2 Main Constructions.

Suppose there are N distributed nodes in the system, and they can all communicate with the server. To begin, we construct the following two non-convex functions

$$g(x) := \frac{1}{N}\sum_{i=1}^{N} g_i(x), \quad f(x) := \frac{1}{N}\sum_{i=1}^{N} f_i(x). \tag{C.76}$$

Here we have $x \in \mathbb{R}^{T+1}$. We assume N is constant, and T is the total number of stages (a large number and one that can potentially increase). For notational simplicity, and without loss of generality, we assume that $T \geq N$, and it is divisible by N. Let us define the component functions g_i's in (C.76) as follows.

$$g_i(x) = \Theta(x, 1) + \sum_{j=1}^{T/N} \Theta(x, (j-1)N + i + 1), \tag{C.77}$$

where we have defined the following functions

$$\begin{aligned} \Theta(x, j) &:= \Psi(-x[j-1])\Phi(-x[j]) - \Psi(x[j-1])\Phi(x[j]), \\ &\qquad \forall\ j = 2, \cdots, T+1, \\ \Theta(x, 1) &:= -\Psi(1)\Phi(x[1]). \end{aligned} \tag{C.78a}$$

Clearly, each $\Theta(x, j)$ is only related to two components in x, i.e., $x[j-1]$ and $x[j]$.

The component functions $\Psi, \Phi : \mathbb{R} \to \mathbb{R}$ are given as below

$$\begin{aligned} \Psi(w) &:= \begin{cases} 0 & w \leq 0 \\ 1 - e^{-w^2} & w > 0, \end{cases} \\ \Phi(w) &:= 4 \arctan w + 2\pi. \end{aligned}$$

By the above definition, the average function becomes:

$$\begin{aligned} g(x) &:= \frac{1}{M}\sum_{j=1}^{M} g_i(x) = \Theta(x, 1) + \sum_{j=2}^{T+1} \Theta(x, j) \\ &= -\Psi(1)\Phi\left(x[1]\right) \\ &\quad + \sum_{j=2}^{T+1} \left[\Psi\left(-x[j-1]\right)\Phi\left(-x[j]\right) - \Psi\left(x[j-1]\right)\Phi\left(x[j]\right)\right]. \end{aligned} \tag{C.79}$$

See Fig. C.1 for an illustration of the construction discussed above.

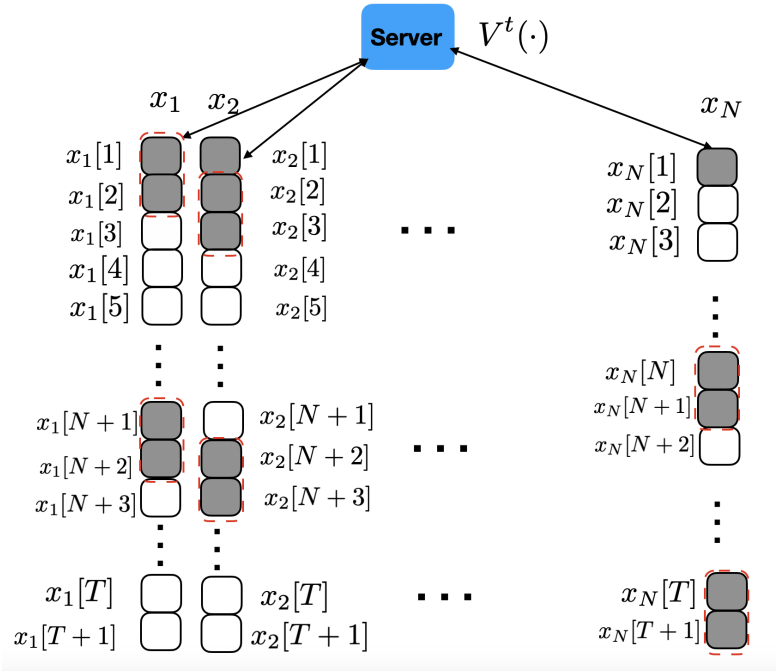


Figure C.1: The example constructed for proving Claim 2.1. Here each agent has a local length $T+1$ vector x_i; each block in the figure represents one dimension of the local vector. If for agent i, its jth block is white it means that f_i is not a function of $x_i[j]$, while if jth block is shaded means f_i is a function of $x_i[j]$. Each dashed red box contains two variables that are coupled together by a function $\Theta(\cdot)$.

Further, for a given error constant $\epsilon > 0$ and a given the Lipschitz constant L, let us define

$$f_i(x) := \frac{2\pi\epsilon}{L} g_i\left(\frac{xL}{\pi\sqrt{2\epsilon}}\right). \tag{C.80}$$

Therefore, we also have

$$f(x) := \frac{1}{N}\sum_{i=1}^{N} f_i(x) = \frac{2\pi\epsilon}{L} g\left(\frac{xL}{\pi\sqrt{2\epsilon}}\right). \tag{C.81}$$

C.6.3 Properties.

First we present some properties of the component functions h_i's.

Lemma 19 *The functions Ψ and Φ satisfy the following:*

1. *For all $w \leq 0$, $\Psi(w) = 0$, $\Psi'(w) = 0$.*

2. The following bounds hold for the functions and their first- and second-order derivatives:

$$0 \le \Psi(w) < 1, \quad 0 \le \Psi'(w) \le \sqrt{\frac{2}{e}},$$
$$-\frac{4}{e^{\frac{3}{2}}} \le \Psi''(w) \le 2, \quad \forall w > 0.$$

$$0 < \Phi(w) < 4\pi, \quad 0 < \Phi'(w) \le 4,$$
$$-\frac{3\sqrt{3}}{2} \le \Phi''(w) \le \frac{3\sqrt{3}}{2}, \quad \forall w \in \mathbb{R}.$$

3. The following key property holds:

$$\Psi(w)\Phi'(v) > 1, \quad \forall\ w \ge 1,\ |v| < 1. \tag{C.82}$$

4. The function h is lower bounded as follows:

$$g_i(0) - \inf_x g_i(x) \le 5\pi T/N,$$
$$g(0) - \inf_x g(x) \le 5\pi T/N.$$

5. The first-order derivative of g (respectively, g_i) is Lipschitz continuous with constant $\ell = 27\pi$ (respectively, $\ell_i = 27\pi$, $\forall\ i$).

Proof. Property 1) is easy to check.

To prove Property 2), note that following holds for $w > 0$:

$$\Psi(w) = 1 - e^{-w^2}, \quad \Psi'(w) = 2e^{-w^2}w,$$
$$\Psi''(w) = 2e^{-w^2} - 4e^{-w^2}w^2,\ \forall\ w > 0.$$

Obviously, $\Psi(w)$ is an increasing function over $w > 0$, therefore the lower and upper bounds are $\Psi(0) = 0, \Psi(\infty) = 1$; $\Psi'(w)$ is increasing on $[0, \frac{1}{\sqrt{2}}]$ and decreasing on $[\frac{1}{\sqrt{2}}, \infty]$, where $\Psi''(\frac{1}{\sqrt{2}}) = 0$, therefore the lower and upper bounds are $\Psi'(0) = \Psi'(\infty) = 0, \Psi'(\frac{1}{\sqrt{2}}) = \sqrt{\frac{2}{e}}$; $\Psi''(w)$ is decreasing on $(0, \sqrt{\frac{3}{2}}]$ and increasing on $[\sqrt{\frac{3}{2}}, \infty)$ (this can be verified by checking the signs of $\Psi'''(w) = 4e^{-w^2}w(2w^2 - 3)$ in these intervals). Therefore the lower and upper bounds are $\Psi''(\sqrt{\frac{3}{2}}) = -\frac{4}{e^{\frac{3}{2}}}, \Psi''(0^+) = 2$, i.e.,

$$0 \le \Psi(w) < 1, \quad 0 \le \Psi'(w) \le \sqrt{\frac{2}{e}},$$
$$-\frac{4}{e^{\frac{3}{2}}} \le \Psi''(w) \le 2, \quad \forall w > 0.$$

Further, for all $w \in \mathbb{R}$, the following holds:

$$\Phi(w) = 4\arctan w + 2\pi, \quad \Phi'(w) = \frac{4}{w^2+1},$$
$$\Phi''(w) = -\frac{8w}{(w^2+1)^2}. \tag{C.83}$$

Similarly, as above, we can obtain the following bounds:

$$0 < \Phi(w) < 4\pi, \quad 0 < \Phi'(w) \leq 4,$$
$$-\frac{3\sqrt{3}}{2} \leq \Phi''(w) \leq \frac{3\sqrt{3}}{2}, \quad \forall w \in \mathbb{R}.$$

To show Property 3), note that for all $w \geq 1$ and $|v| < 1$,

$$\Psi(w)\Phi'(v) > \Psi(1)\Phi'(1) = 2(1 - e^{-1}) > 1$$

where the first inequality is true because $\Psi(w)$ is strictly increasing and $\Phi'(v)$ is strictly decreasing for all $w > 0$ and $v > 0$, and that $\Phi'(v) = \Phi'(|v|)$.

Next we show Property 4). Note that $0 \leq \Psi(w) < 1$ and $0 < \Phi(w) < 4\pi$. Therefore we have $g(0) = -\Psi(1)\Phi(0) < 0$ and using the construction in (C.77)

$$\inf_x g_i(x) \geq -\Psi(1)\Phi(x[1]) - \sum_{j=1}^{T/N} \sup_{w,v} \Psi(w)\Phi(v) \tag{C.84}$$
$$\geq -4\pi - 4(T/N)\pi \geq -5\pi T/N, \tag{C.85}$$

where the first inequality follows from $\Psi(w)\Phi(v) > 0$, the second follows from $\Psi(w)\Phi(v) < 4\pi$, and the last is true because $T/N \geq 1$.

Finally, we show Property 5), using the fact that a function is Lipschitz if it is piecewise smooth with bounded derivative. Before proceeding, let us note a few properties of the construction in (C.79) (also see Fig. C.1). First, for a given node q, its local function h_q is only related to the following $x[j]$'s

$$j = 1 + q + \ell \times N \geq 1, \ \ell = 0, \cdots, (N-1),$$
$$j = q + \ell \times N \geq 1, \ \ell = 0, \cdots, (N-1),$$

or equivalently

$$q = j - 1 - \ell \times N \geq 1, \ \ell = 0, \cdots, (N-1),$$
$$q = j - \ell \times N \geq 1, \ \ell = 0, \cdots, (N-1).$$

Then the first-order partial derivative of $g_q(y)$ can be expressed below.

Case I) If $j \neq 1$ we have

$$\frac{\partial g_q}{\partial x[j]} = \begin{cases} \left(-\Psi\left(-x[j-1]\right)\Phi'\left(-x[j]\right) - \Psi\left(x[j-1]\right)\Phi'\left(x[j]\right)\right), \\ \qquad\qquad q = j-1-N(\ell) \geq 1,\ \ell = 0, \cdots, \frac{T}{N}-1, j = 2, 3, \cdots, T+1 \\ \left(-\Psi'\left(-x[j]\right)\Phi\left(-x[j+1]\right) - \Psi'\left(x[j]\right)\Phi\left(x[j+1]\right)\right), \\ \qquad\qquad q = j-N(\ell) \geq 1,\ \ell = 0, \cdots, \frac{T}{N}-1, j = 3, 4, \cdots T \\ 0 \qquad\qquad \text{otherwise.} \end{cases} . \tag{C.86}$$

Case II) If $j = 1$, we have

$$\frac{\partial g_q}{\partial x[1]} = \begin{cases} -\Psi(1)\Phi'(x[1]) + \left(-\Psi'\left(-x[1]\right)\Phi\left(-x[2]\right) - \Psi'\left(x[1]\right)\Phi\left(x[2]\right)\right), & q = 1 \\ -\Psi(1)\Phi'(x[1]), & q \neq 1 \end{cases} . \tag{C.87}$$

From the above derivation, it is clear that for any j, q, $\frac{\partial g_q}{\partial x[j]}$ is either zero or is a piecewise smooth function separated at the non-differentiable point $x[j] = 0$, because the function $\Psi'(\cdot)$ is not differentiable at 0.

Further, fix a point $x \in \mathbb{R}^{T+1}$ and a unit vector $v \in \mathbb{R}^{T+1}$ where $\sum_{j=1}^{T+1} v[j]^2 = 1$. Define

$$\ell_q(\theta; x, v) := g_q(x + \theta v)$$

to be the directional projection of g_q on to the direction v at point x. We will show that there exists $C > 0$ such that $|\ell_q{}''(0; x, v)| \leq C$ for all $x \neq 0$ (where the second-order derivative is taken with respect to θ).

First, by noting the fact that each if $x[j]$ appears in $g_q(x)$, then it must also be *coupled with* either $x[j+1]$ or $x[j-1]$, but not other $x[k]$'s for $k \neq j-1, j+1$. This means that $\frac{\partial^2 g_q(x)}{\partial x[j_1]\partial x[j_2]} = 0$, $\forall\ j_2 \neq \{j_1, j_1+1, j_1-1\}$. Using this fact, we can compute $\ell_q{}''(0; x, v)$ as follows:

$$\begin{aligned} \ell_q^{''}(0; x, v) &= \sum_{j_1, j_2 = 1}^{T} \frac{\partial^2 g_q(x)}{\partial x[j_1]\partial x[j_2]} v[j_1] v[i_2] \\ &= \sum_{G \in \{0,1,-1\}} \sum_{j=1}^{T} \frac{\partial^2 g_q(x)}{\partial x[j]\partial x[j+G]} v[j] v[j+G], \end{aligned}$$

where we take $v[0] := 0$ and $v[T+1] := 0$.

By using (C.86) and the above facts, the second-order partial derivative of $g_q(x)$ ($\forall x \neq 0$) is given as follows when $j \neq 1$:

$$\frac{\partial^2 g_q}{\partial x[j]\partial x[j]} = \begin{cases} \left(\Psi\left(-x[j-1]\right)\Phi''\left(-x[j]\right) - \Psi\left(x[j-1]\right)\Phi''\left(x[j]\right)\right), \\ \qquad\qquad q = j-1-N(\ell) \geq 1,\ \ell = 0, \cdots, \frac{T}{N}-1, j = 2, 3, \cdots, T+1 \\ \left(\Psi''\left(-x[j]\right)\Phi\left(-x[j+1]\right) - \Psi''\left(x[j]\right)\Phi\left(x[j+1]\right)\right), \\ \qquad\qquad q = j-N(\ell) \geq 1,\ \ell = 0, \cdots, \frac{T}{N}-1, j = 3, 4, \cdots, T \\ 0, \qquad\qquad \text{otherwise} \end{cases} \tag{C.88}$$

$$\frac{\partial^2 g_q}{\partial x[j]\partial x[j+1]} = \begin{cases} \left(\Psi'\left(-x[j]\right)\Phi'\left(-x[j+1]\right) - \Psi'\left(x[j]\right)\Phi'\left(x[j+1]\right)\right), \\ \qquad\qquad q = j - N(\ell) \geq 1,\ \ell = 0, \cdots, \frac{T}{N} - 1, j = 3, 4, \cdots, T \\ 0, \qquad\qquad \text{otherwise} \end{cases} \tag{C.89}$$

$$\frac{\partial^2 g_q}{\partial x[j]\partial x[j-1]} = \begin{cases} \left(\Psi'\left(-x[j-1]\right)\Phi'\left(-x[j]\right) - \Psi'\left(x[j-1]\right)\Phi'\left(x[j]\right)\right), \\ \qquad\qquad q = j - N(\ell) \geq 1,\ \ell = 0, \cdots, \frac{T}{N} - 1, j = 2, 3, \cdots, T+1 \\ 0, \qquad\qquad \text{otherwise} \end{cases} . \tag{C.90}$$

By applying Lemma 19 – i) [i.e., $\Psi(w) = \Psi'(w) = \Psi''(w) = 0$ for $\forall\ w \leq 0$], we can obtain that at least one of the terms $\Psi\left(-x[j-1]\right)\Phi''\left(-x[j]\right)$ or $-\Psi\left(x[j-1]\right)\Phi''\left(x[j]\right)$ is zero. It follows that

$$\Psi\left(-x[j-1]\right)\Phi''\left(-x[j]\right) - \Psi\left(x[j-1]\right)\Phi''\left(x[j]\right) \leq \sup_w |\Psi(w)| \sup_v |\Phi''(v)|.$$

Taking the maximum over equations (C.88) to (C.90) and plug in the above inequalities, we obtain

$$\begin{aligned} \left|\frac{\partial^2 g_q}{\partial x[j_1]\partial x[j_2]}\right| &\leq \max\{\sup_w |\Psi''(w)| \sup_v |\Phi(v)|, \sup_w |\Psi(w)| \sup_v |\Phi''(v)|, \sup_w |\Psi'(w)| \sup_v |\Phi'(v)|\} \\ &= \max\left\{8\pi, \frac{3\sqrt{3}}{2}, 4\sqrt{\frac{2}{e}}\right\} < 8\pi, \quad \forall\ j_1 \neq 1, \end{aligned}$$

where the equality comes from Lemma 19 – ii).

When $j = 1$, by using (C.87), we have the following:

$$\frac{\partial^2 g_q(x)}{\partial x[1]\partial x[1]} = \begin{cases} -\Psi(1)\Phi''(x[1]) + \left(-\Psi''\left(-x[1]\right)\Phi\left(-x[2]\right) - \Psi''\left(x[1]\right)\Phi\left(x[2]\right)\right), & q = 1 \\ -\Psi(1)\Phi''(x[1]), & \text{otherwise} \end{cases},$$

$$\frac{\partial^2 g_q(x)}{\partial x[1]\partial x[2]} = \begin{cases} \left(-\Psi'\left(-x[1]\right)\Phi'\left(-x[2]\right) - \Psi'\left(x[1]\right)\Phi'\left(x[2]\right)\right), & q = 1 \\ 0, & \text{otherwise} \end{cases}.$$

Again by applying Lemma 19 – i) and ii),

$$\begin{aligned} \left|\frac{\partial^2 g_q(x)}{\partial x[1]\partial x[j_2]}\right| &\leq \max\{\sup_w |\Psi(1)\Phi''(w)| + \sup_w |\Psi''(w)| \sup_v |\Phi(v)|, \sup_w |\Psi'(w)| \sup_v |\Phi'(v)|\} \\ &= \max\left\{\frac{3\sqrt{3}}{2}(1 - e^{-1}) + 8\pi, 4\sqrt{\frac{2}{e}}\right\} < 9\pi,\ \forall\ j_2. \end{aligned}$$

Summarizing the above results, we obtain:

$$\begin{aligned}
|\ell_q''(0;x,v)| &= |\sum_{G\in\{0,1,-1\}}\sum_{j=1}^{T}\frac{\partial^2 g_q(y)}{\partial x[j]\partial x[j+G]}v[j]v[j+G]| \\
&\leq 9\pi\sum_{G\in\{0,1,-1\}}|\sum_{j=1}^{T}v[j]v[j+G]| \\
&\leq 9\pi\left(|\sum_{j=1}^{T}v[j]^2|+2|\sum_{j=1}^{T}v[j]v[j+1]|\right) \\
&\leq 27\pi\sum_{j=1}^{T}|v[j]^2| = 27\pi.
\end{aligned}$$

Overall, the first-order derivatives of h_q are Lipsschitz continuous for any q with constant at most $\ell = 27\pi$. ■

The following lemma is a simple extension of the previous result.

Lemma 20 *We have the following properties for the functions f defined in* (C.81) *and* (C.80)*:*

1. *We have $\forall\ x\in\mathbb{R}^{T+1}$*
$$f(0)-\inf_x f(x)\leq\frac{10\pi^2\epsilon}{LN}T.$$
2. *We have*
$$\|\nabla f(x)\| = \sqrt{2\epsilon}\left\|\nabla g\left(\frac{xL}{\pi\sqrt{2\epsilon}}\right)\right\|,\ \forall\ x\in\mathbb{R}^{T+1}. \tag{C.91}$$
3. *The first-order derivatives of f and that for each $f_i, i\in[N]$ are Lipschitz continuous, with the same constant $U>0$.*

Proof. To show that property 1) is true, note that we have the following:

$$f(0)-\inf_x f(x) = \frac{2\pi\epsilon}{L}\left(g(0)-\inf_x g(x)\right).$$

Then by applying Lemma 19 we have that for any $T\geq 1$, the following holds

$$f(0)-\inf_x f(x)\leq\frac{2\pi\epsilon}{L}\times\frac{5\pi T}{N}.$$

Property 2) is true is due to the definition of f_i, so that we have:

$$\nabla f_i(x) = \sqrt{2\epsilon}\times\nabla g_i\left(\frac{xL}{\pi\sqrt{2\epsilon}}\right).$$

Property 3) is true because the following:

$$\|\nabla f(z)-\nabla f(y)\| = \sqrt{2\epsilon}\left\|\nabla g\left(\frac{zU}{\pi\sqrt{2\epsilon}}\right)-\nabla g\left(\frac{yU}{\pi\sqrt{2\epsilon}}\right)\right\|\leq U\|z-y\|$$

where the last inequality comes from Lemma 19 – (5). This completes the proof. ■

Next let us analyze the size of ∇g. We have the following result.

Lemma 21 *If there exists $k \in [T]$ such that $|x[k]| < 1$, then*

$$\|\nabla g(x)\| = \left\| \frac{1}{N} \sum_{i=1}^{N} \nabla g_i(x) \right\| \geq \left| \frac{1}{N} \sum_{i=1}^{N} \frac{\partial g_i(x)}{\partial x[k]} \right| > 1/N.$$

Proof. The first inequality holds for all $k \in [T]$, since $\frac{1}{N}\sum_{i=1}^{N} \frac{\partial}{\partial y[k]} g_i(x)$ is one element of $\frac{1}{N}\sum_{i=1}^{N} \nabla g_i(x)$. We divide the proof for the second inequality into two cases.

Case 1. Suppose $|x[j-1]| < 1$ for all $2 \leq j \leq k$. Therefore, we have $|x[1]| < 1$. Using (C.87), we have the following inequalities:

$$\frac{\partial g_i(x)}{\partial x[1]} \overset{\text{(i)}}{\leq} -\Psi(1)\Phi'(x[1]) \overset{\text{(ii)}}{<} -1, \forall i \tag{C.92}$$

where (i) is true because $\Psi'(w), \Phi(w)$ are all non-negative from Lemma 19 -(2); (ii) is true due to Lemma 19 – (3). Therefore, we have the following

$$\left\| \frac{1}{N} \sum_{i=1}^{N} \nabla g_i(x) \right\| \geq \left| \frac{1}{N} \sum_{i=1}^{N} \frac{\partial}{\partial x[1]} g_i(x) \right| > 1.$$

Case 2) Suppose there exists $2 \leq j \leq k$ such that $|x[j-1]| \geq 1$.

We choose j so that $|x[j-1]| \geq 1$ and $|x[j]| < 1$. Therefore, depending on the choices of (i, j) we have three cases:

$$\frac{\partial g_i(x)}{\partial x[j]} = \begin{cases} \left(-\Psi\left(-x[j-1]\right)\Phi'\left(-x[j]\right) - \Psi\left(x[j-1]\right)\Phi'\left(x[j]\right)\right), \\ \qquad\qquad i = j - 1 - N(\ell) \geq 1,\ \ell = 0, \cdots, \frac{T}{N} - 1, j = 2, 3, \cdots, T+1 \\ \left(-\Psi'\left(-x[j]\right)\Phi\left(-x[j+1]\right) - \Psi'\left(x[j]\right)\Phi\left(x[j+1]\right)\right), \\ \qquad\qquad i = j - 1 - N(\ell) \geq 1,\ \ell = 0, \cdots, \frac{T}{N} - 1, j = 3, 4, \cdots, T \\ 0 \\ \qquad\qquad \text{otherwise} \end{cases}. \tag{C.93}$$

First, note that $\frac{\partial g_i(x)}{\partial x[j]} \leq 0$, for all i, j, by checking the definitions of $\Psi(\cdot), \Phi'(\cdot), \Psi'(\cdot), \Phi(\cdot)$.

Then for (i, j) satisfying the first condition, because $|x[j-1]| \geq 1$ and $|x[j]| < 1$, using Lemma 19 – (3), and the fact that the negative part is zero for Ψ, and Φ' is even function, the expression further simplifies to:

$$-\Psi(|x[j-1]|)\Phi'\left(|x[j]|\right)] \overset{\text{(C.82)}}{<} -1. \tag{C.94}$$

If the second condition holds true, the expression is obviously non-positive because both Ψ' and Φ are non-negative. Overall, we have

$$\left| \frac{1}{N} \sum_{i=1}^{N} \frac{\partial g_i(x)}{\partial x[j]} \right| > \frac{1}{N}.$$

This completes the proof. ■

Lemma 22 *Consider using an algorithm of the form* (C.75) *to solve the following problem:*

$$\min_{x\in\mathbb{R}^{T+1}} \; g(x) = \frac{1}{N}\sum_{i=1}^{N} g_i(x). \tag{C.95}$$

Assume the initial solution: $x_i = 0,\ \forall\ i \in [N]$. *Let* $\bar{x} = \frac{1}{N}\sum_{i=1}^{N}\alpha_i x_i$ *denote some linear combination of local variables, where* $\{\alpha_i > 0\}$ *are the coefficients (possibly time-varying and dependent on* t*). Then no matter how many local computation steps* (C.75b) *are performed, at least* T *communication steps* (C.75a) *are needed to ensure* $\bar{x}[T] \neq 0$.

Proof. For a given $j \geq 2$, suppose that $x_i[j], x_i[j+1], ..., x_i[T] = 0, \forall i$, that is, support$\{x_i\} \subseteq \{1, 2, 3, ..., j-1\}$ for all i. Then $\Psi'(x_i[j]) = \Psi'(-x_i[j]) = 0$ for all i, and g_i has the following partial derivative (see (C.86))

$$\frac{\partial g_i(x_i)}{\partial x_i[j]} = -\left(\Psi\left(-x_i[j-1]\right)\Phi'\left(-x_i[j]\right)\right) + \left(\Psi\left(x_i[j-1]\right)\Phi'\left(x_i[j]\right)\right), \tag{C.96}$$

$$i = j - 1 - N(\ell) \geq 1,\ \ell = 0, \cdots, \frac{T}{N} - 1, j = 2, 3, \cdots, T+1. \tag{C.97}$$

Clearly, if $x_i[j-1] = 0$, then by the definition of $\Psi(\cdot)$, the above partial gradient is also zero. In other words, the above partial gradient is only non-zero if $x_i[j-1] \neq 0$.

Recall that we have assumed that the server aggregation is performed using a linear combination $\bar{x} = \frac{1}{N}\sum_{i=1}^{N}\alpha_i x_i$, with the coefficients α_i's possibly depending on the stage t (but such a dependency will be irrelevant for our purpose, as will be see shortly). Therefore, at a given stage t, for a given node i, when $j \geq 3$, its jth element will become *nonzero* only if one of the following two cases hold true:

- If before the aggregation step (i.e., at stage $t-1$), some other node q has $x_q[j]$ being nonzero.
- If $\frac{\partial g_i(x_i)}{\partial x_i[j]}$ is nonzero at stage t.

Now suppose that the initial solution is $x_i[j] = 0$ for all (i, j). Then at the first iteration only $\frac{\partial g_i(x_i)}{\partial x_i[1]}$ is non-zero for all i, due to the fact that $\frac{\partial g_i(x_i)}{\partial x_i[1]} = \Psi(1)\Phi'(0) = 4(1 - e^{-1})$ for all i from (C.87). It is also important to observe that, if all nodes $i \neq 1$ were to perform subsequent local updates (C.75b), the local variable x_j will have the same support (i.e., only the first element is non-zero). To see this, suppose $k = 2$, then for $i = 2$, we have

$$\frac{\partial g_i(x_i)}{\partial x_i[2]} = \left(-\Psi'\left(-x[2]\right)\Phi\left(-x[3]\right) - \Psi'\left(x[2]\right)\Phi\left(x[3]\right)\right) = 0, \tag{C.98}$$

since $x[2] = 0$ implies $\Psi'(-x[2]) = 0$. Similarly reasoning applies when $i = 2$, $k \geq 3$.

If $i \geq 3$, then these local functions are not related to $x_i[2]$, so the partial derivative is also zero.

Now let us look at node $i = 1$. For this node, according to (C.96), we have

$$\frac{\partial g_1(x_1)}{\partial x_1[2]} = -\left(\Psi\left(-x_1[1]\right)\Phi'\left(-x_1[2]\right)\right) + \left(\Psi\left(x_1[1]\right)\Phi'\left(x_1[2]\right)\right). \tag{C.99}$$

Since $x_1[1]$ can be non-zero, then this partial gradient can also be non-zero. Further, with a similar argument as above, we can also confirm that no matter how many local computation steps that node 1 performs, only the first two elements of x_1 can be non-zero.

So for the first stage $t = 1$, we conclude that, no matter how many local computation that the nodes perform (in the form of the computation step given in (C.75b)), only x_1 can have two non-zero entries, while the rest of the local variables only have one non-zero entries.

Then suppose that the communication and aggregation step is performed once. It follows that after broadcasting $\bar{x} = \frac{1}{N}\sum_{i=1}^{N}\alpha_i x_i$ to all the nodes, everyone can have two non-zero entries. Then the nodes proceed with local computation, and by the same argument as above, one can show that this time only x_2 can have three non-zero entries. Following the above procedure, it is clear that each aggregation step can advance the non-zero entry of $\bar{x}$ by one, while performing multiple local updates does not advance the non-zero entry. Then we conclude that we need at least T communication steps, and local gradient computation steps, to make $x_i[T]$ possibly non-zero. ■

C.6.4 Main Result for Claim 2.1.

Below we state and prove a formal version of Claim 2.1 given in the main text.

Theorem 11 *Let ϵ be a positive number. Let $x_i^0[j] = 0$ for all $i \in [N]$, and all $j = 1, \cdots, T+1$. Consider any algorithm obeying the rules given in* (5.5)*, where the $V^r(\cdot)$ and $W_i^r(\cdot)$'s are linear operators. Then regardless of the number of local updates there exists a problem satisfying Assumption 8 – 9, such that it requires at least the following number of stages t (and equivalently, aggregation and communications rounds in* (C.75a)*)*

$$r \geq \frac{\left(f(0) - \inf_x f(x)\right) LN}{10\pi^2}\epsilon^{-1} \tag{C.100}$$

to achieve the following error

$$h_r^* = \left\| \frac{1}{N}\sum_{i=1}^{N}\nabla f_i(x^r) \right\|^2 < \epsilon. \tag{C.101}$$

Proof of Claim 2.1. First, let us show that the algorithm obeying the rules given in (C.75) has the desired property. Note that the difference between two rules is whether the *sampled* local gradients are used for the update, or the full local gradients are used.

By Lemma 22 we have $\bar{x}[T] = 0$ for all $r < T$. Then by applying Lemma 20 – (2) and Lemma 21, we can conclude that the following holds

$$\|\nabla f(\bar{x}[T])\| = \sqrt{2\epsilon} \left\| \nabla h \left(\frac{\bar{x}[T]U}{\pi\sqrt{2\epsilon}} \right) \right\| > \sqrt{2\epsilon}/N, \tag{C.102}$$

where the second inequality follows that there exists $k \in [T]$ such that $|\frac{\bar{x}[k]U}{\pi\sqrt{2\epsilon}}| = 0 < 1$, then we can directly apply Lemma 21.

The third part of Lemma 20 ensures that f_i's are L-Lipschitz continuous gradient, and the first part shows

$$f(0) - \inf_x f(x) \leq \frac{10\pi^2\epsilon}{LN} T,$$

Therefore we obtain

$$T \geq \frac{(f(0) - \inf_x f(x))\, LN}{10\pi^2} \epsilon^{-1}. \tag{C.103}$$

This completes the proof.

Second, consider the algorithm obeying the rules give in (5.5), in which local *sampled* gradients are used. By careful inspection, the result for this case can be trivially extended from the previous case. We only need to consider the following local functions

$$\hat{f}_i(x) = \sum_{\xi_i \in D_i} F(x; \xi_i) \tag{C.104}$$

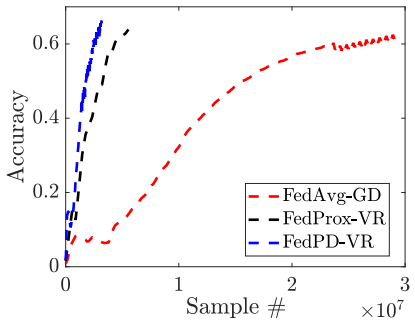
where each sampled loss function $F(x; \xi_i)$ is defined as

$$F(\mathbf{x}; \xi_i) = G(\xi_i) f_i(x), \quad \text{where } f_i(x) \text{ is defined in (C.80).} \tag{C.105}$$

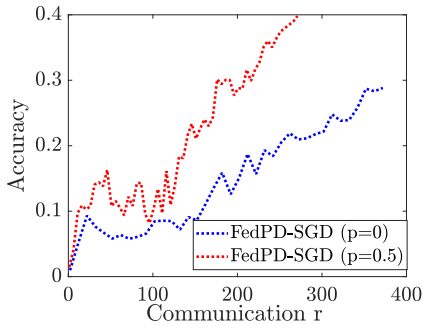
where $G(\xi_i)$'s satisfy $G(\xi_i) > 0$ and $\sum_{\xi_i \in D_i} G(\xi_i) = 1$. It is easy to see that, the local sampled gradients have the same dependency on x as their averaged version (by dependency we meant the structure that is depicted in Fig. C.1). Therefore, the progression of the non-zero pattern of the average $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ is exactly the same as the batch gradient version. Additionally, since the local function $\hat{f}(x)$ is exactly the same as the previous local function $f(x)$, so other estimates, such as the one that bounds $f(0) - \inf f(x)$, also remain the same.

C.7 Additional Numerical Results

C.7.1 Handwritten Character Classification

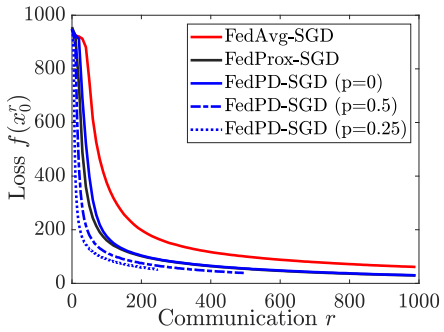


(a) The testing accuracy of FedAvg-GD, FedProx-VR and FedPD-VR with respect to the number of samples.

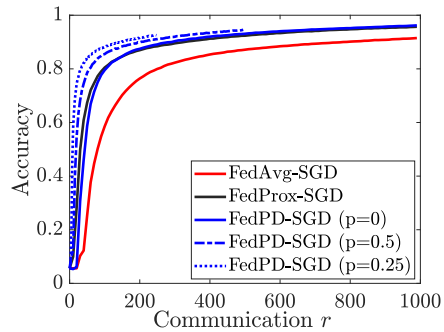


(b) The testing accuracy of FedPD-SGD with $p = 0$ and $p = 0.5$ with respect to the number of communications.

Figure C.2: The convergence result of the algorithms on training neural network for handwriting character classification.

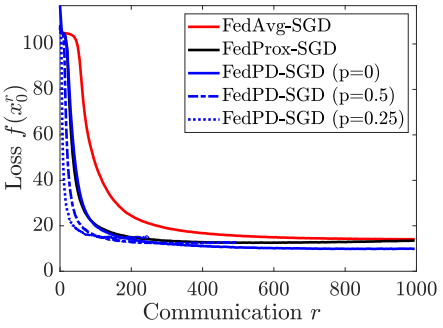


(a) The loss value of FedAvg-SGD, FedProx-SGD and FedPD-SGD with respect to the number of communication rounds.

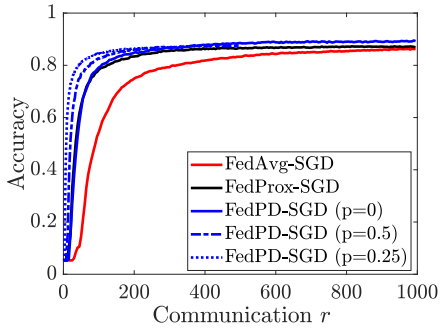


(b) The training accuracy of of FedAvg-SGD, FedProx-SGD and FedPD-SGD with respect to the number of communication rounds.

Figure C.3: The convergence results of the algorithms on training neural networks on the federated handwritten characters classification problem.



(a) The testing loss value of FedAvg-SGD, FedProx-SGD and FedPD-SGD with respect to the number of communication rounds.



(b) The testing accuracy of of FedAvg-SGD, FedProx-SGD and FedPD-SGD with respect to the number of communication rounds.

Figure C.4: The convergence results of the algorithms on training neural networks on the federated handwritten characters classification problem with test data set.

In the second experiment, we compare FedPD with FedAvg and FedProx on the FEMNIST data set [134]. The FEMNIST data set collects the handwritten characters, including numbers 1–10 and the upper- and lower-case letters A–Z and a–z, from different writers and is separated by the writers, therefore the data set naturally preserves non-i.i.d-ness.

The entire data set contains 805,000 samples collected from 3,550 writers. In our experiments, we use the data collected from 100 writers with an average of 300 samples per writer and the size of the whole data set is 29,214. We set the number of agent $N = 90$, the first ten agents are assigned with data from two writers, and the rest of the agents are assigned with data form one writer. Therefore, the data distribution is neither i.i.d. nor balanced. We use the neural network given in [134] as the training model, which consists of 2 convolutional layers and two fully connected layers. The output layer has 62 neurons that matches the number of classes in the FEMNIST data set.

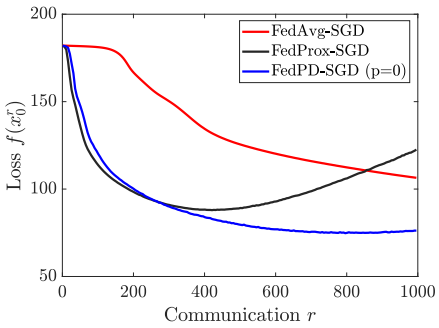
The numerical results shown in Fig. C.2 in the main text were generated by running MATLAB codes on Amazon Web Services (AWS), with Intel Xeon E5-2686 v4 CPUs. In the training phase, we train the CNN model with FedAvg, FedProx and FedPD. In Fig. C.2(a), for FedAvg, we use gradient descent for $Q = 8$ local update steps between each communication rounds; to solve the local problem for FedProx, we use SARAH with $Q = 20$ local steps; we use FedPD with Oracle II, computing full gradient every $I = 20$ communication rounds and perform $Q = 2$ local steps between two communication rounds. The hyper-parameters we use for FedAvg is $\eta = 0.005$; for FedProx we use $\rho = 1$ and stepsize $\eta = 0.01$; for FedPD we use $\eta = 100$ and

$\gamma = 400$. In Fig. C.2(b), we use FedPD with Oracle I, with $Q = 20$, $\eta = 100$ and $\gamma = 400$ and the mini-batch size 2. We set the communication saving to $p = 0$ and $p = 0.5$.

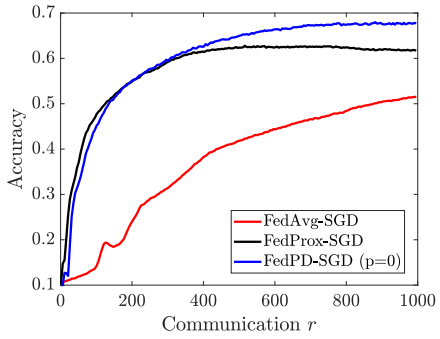
The results shown in Fig. C.3 were generated by running Python codes (using the the PyTorch package[1]) with AMD EPYC 7702 CPUs and an NVIDIA V100 GPU.

In the training phase, we train with FedProx, FedAvg and FedPD with a total $T = 1000$ outer iterations. The local problems are solved with SGD for $Q = 300$ local iterations and the mini-batch size in evaluating the stochastic gradient is 2. The stepsize choice for FedAvg, FedProx and FedPD are 0.001, 0.01 and 0.01 by grid-search from $\{1, 0.3, 0.1, 0.03, 0.01, 0.003, 0.001\}$, the hyper-parameter of FedProx is $\rho = 1$ and for FedPD $\eta = 1$. In the experiment, we set the communication saving for FedPD to be $p = 0$, $p = 0.5$ and $p = 0.25$. Note that we also tested FedAvg with larger stepsize 0.01, but the algorithm becomes unstable, and its performance degrages significantly. As shown in Fig. C.3 and C.4, FedAvg is slower than FedPD and FedProx, while FedProx has similar performance as FedPD when $p = 0$. Further, we can see that as the frequency of communication of FedPD decreases, the final accuracy decreases and the final loss increases. However, the drop of accuracy is not significant, so FedPD is able to achieve a better performance with the same number of communication rounds.

C.7.2 Cifar-10 Dataset Classification



(a) The testing loss value of FedAvg-SGD, FedProx-SGD and FedPD-SGD with respect to the number of communication rounds.



(b) The testing accuracy of of FedAvg-SGD, FedProx-SGD and FedPD-SGD with respect to the number of communication rounds.

Figure C.5: The convergence results of the algorithms on training neural networks on the Cifar-10 classification problem with test data set.

[1] PyTorch: An Imperative Style, High-Performance Deep Learning Library, `https://pytorch.org/`

In the third experiment, we compare FedPD with FedAvg and FedProx on the Cifar-10 dataset [124]. In the experiment, we set the number of agent $N = 90$. The data partitioning method follows the one used in [135] and the details of the method is described as follows: first we sort the samples by the labels and divide the dataset into 200 non-overlapping subsets and each subset of samples only has one class of samples; then for 10 agents, we randomly sample 4 subsets from the 200 subsets without replacement and assign to each agent; for the other 80 agents, we sample 2 subsets without replacement and assign to each client. Each agent only has at most 4 classes of samples and have different number of samples, therefore, the data distribution is non-i.i.d. and unbalanced.

The results are shown in Fig. C.5. In the training phase, we train with FedProx, FedAvg and FedPD with a total $T = 1000$ outer iterations. The local problems are solved with SGD for $Q = 64$ local iterations and the mini-batch size in evaluating the stochastic gradient is 16. The stepsize choice for FedAvg, FedProx and FedPD are 0.001, 0.01 and 0.01, the hyper-parameter of FedProx is $\rho = 1$ and FedPD is $\eta = 1$. We use the neural network which consists of 2 convolutional layers and three fully connected layers as the training model. As shown in the result, FedAvg is slower than FedPD and FedProx, while FedProx has a lower final accuracy than FedPD when $p = 0$.

C.8 The Connection Between FedDyn and FedPD

In this section, we provide a short discussion about the connections of FedPD and FedDyn [81]. The bottom line is that, without communication reduction, these two algorithms are identical. In particular, the so-called "dynamic regularization" step in FedDyn is precisely the dual update step in FedPD.

The Federated Dynamic Regularization Algorithm (FedDyn) proposes "a dynamic regularizer for each device at each round, so that in the limit the global and device solutions are aligned". Further, in each iteration r, only a subset of users $\mathcal{P}_r \subseteq \mathcal{U}$ is selected, out of a total of N users. The FedDyn algorithm is given below.

Algorithm 9 Federated Dynamic Regularizer

Input: $\mathbf{x}^0, \eta, T$,

Initialize: $\mathbf{x}_0^0 = \mathbf{x}^0$, $\mathbf{h}^0$

for $r = 0, \ldots, T-1$ **do**

for $i \in \mathcal{P}_r$ in parallel do local updates **do**

$$\mathbf{x}_i^{r+1} = \arg\min_{\mathbf{x}} f_i(\mathbf{x}) + \langle \nabla f_i(\mathbf{x}_i^r), \mathbf{x} - \mathbf{x}_0^r \rangle + \frac{1}{2\eta} \|\mathbf{x} - \mathbf{x}_0^r\|^2 \tag{C.106}$$

$$\nabla f_i(\mathbf{x}_i^{r+1}) = \nabla f_i(\mathbf{x}_i^r) + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_0^r) \tag{C.107}$$

end for

for $i \notin \mathcal{P}_r$ in parallel do local updates **do**

$$\mathbf{x}_i^{r+1} = \mathbf{x}_i^r, \quad \nabla f_i(\mathbf{x}_i^{r+1}) = \nabla f_i(\mathbf{x}_i^r) \tag{C.108}$$

end for

Global Communicate:

$$\mathbf{h}^{r+1} = \mathbf{h}^r + \frac{1}{\eta N} \sum_{i \in \mathcal{P}_r} (\mathbf{x}_i^{r+1} - \mathbf{x}_0^r) \tag{C.109}$$

$$\mathbf{x}_0^{r+1} = \frac{1}{|\mathcal{P}_r|} \sum_{i \in \mathcal{P}_r} \mathbf{x}_i^{r+1} + \eta \mathbf{h}^{r+1} \tag{C.110}$$

end for

To see the relation between these two algorithms, let us assume the following:

- Let $\mathcal{P}_r = \mathcal{N}$ for FedDyn, that is, all clients will participate in communication in all the iterations;
- Consider $p = 0$ for FedPD, that is, communication will take place in all the iterations;
- Consider a simplified version of FedPD where the local problem is solved *exactly*.
- FedPD and FedDyn are initialized such that their initial $\mathbf{x}^0$ are the same, and that the following holds:

$$\lambda_i^0 = \lambda_j^0 = \mathbf{h}^0 = \nabla f_i(\mathbf{x}^0), \ \forall \ i, j. \tag{C.111}$$

Based on the above conditions, we will show that the $\{\mathbf{x}_i^r\}$ and $\{\mathbf{x}_0^r\}$ iterates generated by the two algorithms are the same.

Below, we will show that the following two relations hold:

$$\nabla f_i(\mathbf{x}_i^r) = \lambda_i^r, \quad \mathbf{h}^r = \frac{1}{N}\sum_{i=1}^N \lambda_i^r \quad r = 0, 1, \cdots, T.$$

First, at $r = 0$, the two relations hold trivially because of the initialization.

Let us consider the update in $r = 0$. Recall that the local AL at each client is defined as

$$\mathcal{L}_i(\mathbf{x}_i, \mathbf{x}_0, \lambda_i) \triangleq f_i(\mathbf{x}_i) + \langle \lambda_i, \mathbf{x}_i - \mathbf{x}_0 \rangle + \frac{1}{2\eta} \|\mathbf{x}_i - \mathbf{x}_0\|^2,$$

and the local primal update step for FedPD is

$$\mathbf{x}_i^{r+1} = \arg\min_{\mathbf{x}} \mathcal{L}_i(\mathbf{x}_i, \mathbf{x}_0^r, \lambda_i^r). \tag{C.112}$$

Clearly, the $\mathbf{x}_i^1$ updates in (C.106) and (C.112) are exactly the same, since they are both minimizing the local augmented Lagrangian function, and that $\nabla f_i(\mathbf{x}_i^0) = \lambda_i^0, \ \forall \ i$. Therefore, the two algorithms generate the same $\mathbf{x}_i^1$. It follows that for the FedDyn, the following hold:

$$\begin{aligned} \nabla f_i(\mathbf{x}_i^1) &= \nabla f_i(\mathbf{x}_i^0) + \frac{1}{\eta}(\mathbf{x}_i^1 - \mathbf{x}_0^0) \ {}^{(i)} \\ &= \lambda_i^0 + \frac{1}{\eta}(\mathbf{x}_i^1 - \mathbf{x}_0^0) \overset{(ii)}{=} \lambda_i^1, \ \forall \ i, \end{aligned} \tag{C.113}$$

where in (i) we used the initialization (C.111), and in (ii) we used the dual update of FedPD

$$\lambda_i^{r+1} = \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r),$$

and the fact that the $\mathbf{x}_i^1$ generated by the two algorithms are exactly the same.

Next, we note that the following relations hold for FedDyn:

$$\begin{aligned} \mathbf{h}^{r+1} &= \mathbf{h}^r + \frac{1}{\eta N}\sum_{i=1}^N (\mathbf{x}_i^{r+1} - \mathbf{x}_0^r) \\ &\overset{\text{(C.107)}}{=} \mathbf{h}^r + \frac{1}{N}\sum_{i=1}^N \left(\nabla f_i(\mathbf{x}_i^{r+1}) - \nabla f_i(\mathbf{x}_i^r)\right). \end{aligned}$$

And in particular

$$\begin{aligned} \mathbf{h}^1 &= \mathbf{h}^0 + \frac{1}{N}\sum_{i=1}^N \left(\nabla f_i(\mathbf{x}_i^1) - \nabla f_i(\mathbf{x}_i^0)\right) \\ &= \frac{1}{N}\sum_{i=1}^N \nabla f_i(\mathbf{x}_i^1) = \frac{1}{N}\sum_{i=1}^N \lambda_i^1, \ \forall \ i, \end{aligned}$$

where the last equality comes from (C.113).

Utilizing the fact that $\mathbf{h}^1 = \frac{1}{N}\sum_{i=1}^N \lambda_i^1$, and the two algorithms have the same $\mathbf{x}_i^1$, by a direct comparison of (C.110) and the aggregation step of FedPD

$$\mathbf{x}_0^{r+1} = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_{0,i}^{r+},$$

we obtain that $\mathbf{x}_0^1$ generated by the two algorithms are the same.

The case for all $r \geq 1$ can be similarly derived.

In conclusion, under the initialization (C.111), and assuming that $p = 0$ for FedPD and $\mathcal{P}_r = \mathcal{N}$ for all r, and the FedPD solves local problem exactly, then the FedPD and FedDyn are identical. The key observation from the above analysis is that, the so-called "dynamic regularization" updates in FedDyn are the dual variable updates in FedPD.

Appendix D

Additional Results and Proofs of Chapter 6

D.1 Proof of Theorem 8

By Lipschitz smoothness, we have

$$f(x_{t+1}) \leq f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t \rangle + \frac{L}{2}\|x_{t+1} - x_t\|^2. \tag{D.1}$$

Before we proceed, we define the following quantities to simplify notation:

$$\begin{aligned}
&\alpha_i^t := \frac{c}{\max(c, \eta_l \| \sum_{q=0}^{Q-1} g_i^{t,q} \|)}, \quad \tilde{\alpha}_i^t := \frac{c}{\max(c, \eta_l \| \mathbb{E}[\sum_{q=0}^{Q-1} g_i^{t,q}] \|)}, \quad \overline{\alpha}^t := \frac{1}{N}\sum_{i=1}^{N} \tilde{\alpha}_i^t, \\
&\Delta_i^t := -\eta_l \sum_{q=0}^{Q-1} g_i^{t,q} \cdot \alpha_i^t, \quad \tilde{\Delta}_i^t := -\eta_l \sum_{q=0}^{Q-1} g_i^{t,q} \cdot \tilde{\alpha}_i^t, \\
&\overline{\Delta}_i^t := -\eta_l \sum_{q=0}^{Q-1} g_i^{t,q} \cdot \overline{\alpha}^t, \quad \breve{\Delta}_i^t := -\eta_l \sum_{q=0}^{Q-1} \nabla f_i(x_i^{t,q}) \cdot \overline{\alpha}^t \quad P := |\mathcal{P}_t|,
\end{aligned} \tag{D.2}$$

where the expectation in $\tilde{\alpha}_i^t$ is taken over all possible randomness.

By using the above definitions, the model difference between two consecutive iterations can be expressed as:

$$x_{t+1} - x_t = \eta_g \frac{1}{P} \sum_{i \in \mathcal{P}_t} (\Delta_i^t + z_i^t),$$

with $z_i^t \sim \mathcal{N}(0, \sigma^2 I)$. Using the above expressions, and take an conditional expectation of (D.1)

(conditioned on x_t), we obtain:

$$\begin{aligned}\mathbb{E}[f(x_{t+1})] &\leq f(x_t) + \eta_g \left\langle \nabla f(x_t), \mathbb{E}\left[\frac{1}{P}\sum_{i\in\mathcal{P}_t}\Delta_i^t + z_i^t\right]\right\rangle + \frac{L}{2}\eta_g^2\mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\Delta_i^t + z_i^t\right\|^2\right] \\ &= f(x_t) + \eta_g \left\langle \nabla f(x_t), \mathbb{E}\left[\frac{1}{P}\sum_{i\in\mathcal{P}_t}\Delta_i^t\right]\right\rangle + \frac{L}{2}\eta_g^2\mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\Delta_i^t\right\|^2\right] + \frac{L}{2}\eta_g^2\frac{1}{P}\sigma^2 d,\end{aligned} \tag{D.3}$$

where d in the last expression represents dimension of x_t; in the last equation we use the fact that z_i^t is zero mean.

Next, we will analyze the bias caused by clipping, through analyzing the first order term in (D.3). Towards this end, we have the following series of relations:

$$\begin{aligned}&\left\langle \nabla f(x_t), \mathbb{E}\left[\frac{1}{P}\sum_{i\in\mathcal{P}_t}\Delta_i^t\right]\right\rangle \\ &\overset{(i)}{=} \left\langle \nabla f(x_t), \mathbb{E}\left[\frac{1}{P}\mathbb{E}_i[\sum_{i\in\mathcal{P}_t}\Delta_i^t]\right]\right\rangle = \left\langle \nabla f(x_t), \frac{1}{P}P\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N\Delta_i^t\right]\right\rangle \\ &= \left\langle \nabla f(x_t), \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N\Delta_i^t - \tilde{\Delta}_i^t\right]\right\rangle + \left\langle \nabla f(x_t), \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N\tilde{\Delta}_i^t - \overline{\Delta}_i^t\right]\right\rangle \\ &+ \left\langle \nabla f(x_t), \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N\overline{\Delta}_i^t\right]\right\rangle\end{aligned} \tag{D.4}$$

where (i) we takes expectation on the randomness of the client sampling, i.e., $\mathbb{E}_i\,\Delta_i^t = \frac{1}{N}\sum_{i=1}^N\Delta_i^t$. The first two terms of RHS of the above equality can be viewed as bias caused by clipping. The first order predicted descent can be analyzed from the last term by completing the square:

$$\begin{aligned}&\left\langle \nabla f(x_t), \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N\overline{\Delta}_i^t\right]\right\rangle \\ &\overset{(i)}{=}\mathbb{E}\left[\left\langle \nabla f(x_t), \frac{1}{N}\sum_{i=1}^N\breve{\Delta}_i^t\right]\right\rangle \\ &\overset{(ii)}{=}\frac{-\eta_l\overline{\alpha}^t Q}{2}\|\nabla f(x_t)\|^2 - \frac{\eta_l\overline{\alpha}^t}{2Q}\mathbb{E}\left[\left\|\frac{1}{\eta_l N\overline{\alpha}^t}\sum_{i=1}^N\breve{\Delta}_i^t\right\|^2\right] \\ &\quad + \frac{\eta_l\overline{\alpha}^t}{2}\underbrace{\mathbb{E}\left[\left\|\sqrt{Q}\nabla f(x_t) - \frac{1}{\sqrt{Q}}\frac{1}{\eta_l N\overline{\alpha}^t}\sum_{i=1}^N\breve{\Delta}_i^t\right\|^2\right]}_{A_1},\end{aligned} \tag{D.5}$$

where (i) comes from $\mathbb{E}\,\overline{\Delta}_i^t = \breve{\Delta}_i^t$, (ii) is because $\langle a,b\rangle = -\frac{1}{2}\|a\|^2 - \frac{1}{2}\|b\|^2 + \frac{1}{2}\|a-b\|^2$ holds true for any vector a, b.

We further upper bound A_1 as

$$\begin{aligned}
A_1 =& Q\mathbb{E}\left[\left\|\nabla f(x_t) - \frac{1}{QN}\sum_{i=1}^{N}\sum_{q=0}^{Q-1}\nabla f_i(x_i^{t,q})\right\|^2\right] \\
=& Q\mathbb{E}\left[\left\|\frac{1}{QN}\sum_{i=1}^{N}\sum_{q=0}^{Q-1}\nabla f_i(x^t) - \nabla f_i(x_i^{t,q})\right\|^2\right] \\
\leq& \frac{1}{N}\sum_{i=1}^{N}\sum_{q=0}^{Q-1}\mathbb{E}[\|\nabla f_i(x^t) - \nabla f_i(x_i^{t,q})\|^2] \\
\leq& \frac{1}{N}\sum_{i=1}^{N}\sum_{q=0}^{Q-1}L^2\mathbb{E}[\|x^t - x_i^{t,q}\|^2] \\
\leq& L^2 5Q^2\eta_l^2(\sigma_l^2 + 6Q\sigma_g^2) + L^2 30Q^3\eta_l^2\|\nabla f(x_t)\|^2
\end{aligned} \tag{D.6}$$

where the first inequality comes from Jensen's inequality, the second inequality comes from L-smoothness and the last inequality is due to [72, Lemma 3].

Now we turn to upper bounding the second order term in (D.3), as follows

$$\begin{aligned}
&\mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\Delta_i^t\right\|^2\right] \\
\leq& 3\mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\Delta_i^t - \tilde{\Delta}_i^t\right\|^2\right] + 3\mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\tilde{\Delta}_i^t - \overline{\Delta}_i^t\right\|^2\right] + 3\mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\overline{\Delta}_i^t\right\|^2\right].
\end{aligned} \tag{D.7}$$

We can bound the expectation in the last term of (D.7) as follows:

$$\begin{aligned}
&\mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\overline{\Delta}_i^t\right\|^2\right] \\
=& \mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\left(\eta_l\sum_{q=0}^{Q-1}g_i^{t,q}\cdot\overline{\alpha}^t\right)\right\|^2\right] \\
\leq& \eta_l^2\mathbb{E}\left[2\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\sum_{q=0}^{Q-1}\nabla f(x_i^{t,q})\cdot\overline{\alpha}^t\right\|^2 + 2\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\sum_{q=0}^{Q-1}(\nabla f(x_i^{t,q}) - g_i^{t,q})\cdot\overline{\alpha}^t\right\|^2\right] \\
\leq& 2\mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\breve{\Delta}_i^t\right\|^2\right] + \frac{2}{P}\eta_l^2\overline{\alpha}^2 Q\sigma_l^2
\end{aligned} \tag{D.8}$$

where the last inequality is because the assumption that $\mathbb{E}[\|g_i^{t,q} - \nabla f_i(x_i^{t,q})\|^2] \leq \sigma_l^2$. Let us further bound the expectation in the first term of (D.8) as:

$$\begin{aligned}
\mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\breve{\Delta}_i^t\right\|^2\right] &= \frac{1}{P^2}\,\mathbb{E}\left[\left\|\sum_{i\in\mathcal{P}_t}\breve{\Delta}_i^t\right\|^2\right] \\
&\overset{(i)}{=} \frac{1}{P^2}\,\mathbb{E}\left[\mathbb{E}_i\sum_{i\in\mathcal{P}_t}\left\|\breve{\Delta}_i^t\right\|^2 + \mathbb{E}_{i,j}\sum_{i\neq j\in\mathcal{P}_t}\left\langle\breve{\Delta}_i^t,\breve{\Delta}_j^t\right\rangle\right] \\
&\overset{(ii)}{=} \frac{1}{P^2}\,\mathbb{E}\left[\frac{P}{N}\sum_{i=1}^N\left\|\breve{\Delta}_i^t\right\|^2 + P(P-1)\left\langle\mathbb{E}_i\,\breve{\Delta}_i^t,\mathbb{E}_j\,\breve{\Delta}_j^t\right\rangle\right] \\
&= \frac{1}{P^2}\,\mathbb{E}\left[\frac{P}{N}\sum_{i=1}^N\left\|\breve{\Delta}_i^t\right\|^2 + P(P-1)\left\|\frac{1}{N}\sum_{i=1}^N\breve{\Delta}_i^t\right\|^2\right],
\end{aligned} \tag{D.9}$$

where in (i) we expand the square and take expectation on the randomness of client sampling, and (ii) is due to independent sampling the clients *with* replacement so that $\mathbb{E}_{i,j}\left\langle\Delta_i^t,\Delta_j^t\right\rangle = \left\langle\mathbb{E}_i\,\Delta_i^t,\mathbb{E}_j\,\Delta_j^t\right\rangle$.

Additionally, note we have:

$$\begin{aligned}
\mathbb{E}\sum_{i=1}^N\left\|\breve{\Delta}_i^t\right\|^2 &\overset{(i)}{=} \mathbb{E}\sum_{i=1}^N\eta_l^2(\overline{\alpha}^t)^2\left\|\sum_{q=0}^{Q-1}\nabla f_i(x^t) + \nabla f_i(x_i^{t,q}) - \nabla f_i(x^t)\right\|^2 \\
&\overset{(ii)}{\leq} 2\eta_l^2\overline{\alpha}^t\sum_{i=1}^N\left(Q^2\sum_{q=0}^{Q-1}\mathbb{E}\left\|\nabla f_i(x^t) + \nabla f_i(x_i^{t,q})\right\|^2 + Q^2\sum_{q=0}^{Q-1}\left\|\nabla f_i(x^t)\right\|^2\right) \\
&\overset{(iii)}{\leq} 2\eta_l^2\overline{\alpha}^t N\Big(L^2 5Q^2\eta_l^2(\sigma_l^2 + 6Q\sigma_g^2) + L^2 30Q^3\eta_l^2\|\nabla f(x_t)\|^2 + 2Q^3\|\nabla f(x_t)\|^2 + 2Q^3\sigma_g^2\Big) \\
&= 10N\eta_l^4\overline{\alpha}^t L^2Q^2\sigma_l^2 + 4N\eta_l^2\overline{\alpha}^t Q^3(15L^2\eta_l^2+1)(\|\nabla f(x_t)\|^2 + \sigma_g^2).
\end{aligned} \tag{D.10}$$

where (i) comes from the definition of $\breve{\Delta}_i^t$; (ii) comes from the fact that $\|a+b\|^2 \leq 2(\|a\|^2 + \|b\|^2)$; in (iii) we apply (D.6) to the first term and bound the second term by the assumption that $\|\nabla f_i(x) - \nabla f(x)\|^2 \leq \sigma_g^2$.

Combining (D.3)-(D.10), we have

$$\begin{aligned}
\mathbb{E}[f(x_{t+1})] \leq & f(x_t) - \frac{\eta_g\eta_l\overline{\alpha}^t Q}{2}\|\nabla f(x_t)\|^2 - \frac{\eta_g\eta_l\overline{\alpha}^t}{2Q}\mathbb{E}\left[\left\|\frac{1}{\eta_l N\overline{\alpha}^t}\sum_{i=1}^N\breve{\Delta}_i^t\right\|^2\right] \\
&+ \frac{\eta_g\eta_l\overline{\alpha}^t}{2}(5L^2Q^2\eta_l^2(\sigma_l^2 + 6Q\sigma_g^2) + 30L^2Q^3\eta_l^2\|\nabla f(x_t)\|^2) \\
&+ \eta_g\left\langle\nabla f(x_t),\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N\Delta_i^t - \tilde{\Delta}_i^t\right]\right\rangle + \eta_g\left\langle\nabla f(x_t),\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N\tilde{\Delta}_i^t - \overline{\Delta}_i^t\right]\right\rangle
\end{aligned}$$

$$
\begin{aligned}
&+\frac{3L\eta_g^2(P-1)}{P}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\breve{\Delta}_i^t\right\|^2\right]+\frac{3L}{P}\eta_g^2\eta_l^2(\overline{\alpha}^t)^2Q\sigma_l^2+\frac{L}{2}\eta_g^2\frac{1}{P}\sigma^2 d\\
&+\frac{30}{P}\eta_l^4\eta_g^2\overline{\alpha}^tL^2Q^2\sigma_l^2+\frac{12}{P}\eta_l^2\eta_g^2\overline{\alpha}^tQ^3(15L^2\eta_l^2+1)(\|\nabla f(x_t)\|^2+\sigma_g^2)\\
&+\frac{3L}{2}\eta_g^2\mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\Delta_i^t-\tilde{\Delta}_i^t\right\|^2\right]+\frac{3L}{2}\eta_g^2\mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\tilde{\Delta}_i^t-\overline{\Delta}_i^t\right\|^2\right] \qquad \text{(D.11)}
\end{aligned}
$$

When $\eta_g\eta_l\leq\min\{\frac{\sqrt{P}}{\sqrt{48QQ}},\frac{P}{6QL(P-1)}\}$ and $\eta_l\leq\frac{1}{\sqrt{60QL}}$, the above inequality simplifies to

$$
\begin{aligned}
\mathbb{E}[f(x_{t+1})]\leq& f(x_t)-\frac{\eta_g\eta_l\overline{\alpha}^tQ}{4}\|\nabla f(x_t)\|^2\\
&+\frac{5\eta_g\eta_l^3\overline{\alpha}^t}{2}(1+\frac{12\eta_l\eta_g}{P})L^2Q^2(\sigma_l^2+6Q\sigma_g^2)\\
&+\eta_g\left\langle\nabla f(x_t),\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\Delta_i^t-\tilde{\Delta}_i^t\right]\right\rangle+\eta_g\left\langle\nabla f(x_t),\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\tilde{\Delta}_i^t-\overline{\Delta}_i^t\right]\right\rangle\\
&+\frac{3L}{N}\eta_g^2\eta_l^2(\overline{\alpha}^t)^2Q\sigma_l^2+\frac{L}{2}\eta_g^2\frac{1}{P}\sigma^2 d\\
&+\frac{3L}{2}\eta_g^2\mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\Delta_i^t-\tilde{\Delta}_i^t\right\|^2\right]+\frac{3L}{2}\eta_g^2\mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\tilde{\Delta}_i^t-\overline{\Delta}_i^t\right\|^2\right] \qquad \text{(D.12)}
\end{aligned}
$$

Sum over t from 1 to T, divide both sides by $T\eta_g\eta_lQ/4$, and rearrange, we have

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\overline{\alpha}^t\|\nabla f(x_t)\|^2]\\
\leq&\frac{4}{T\eta_g\eta_lQ}(\mathbb{E}[f(x_1)]-\mathbb{E}[f(x_{T+1})])\\
&+10\eta_l^2L^2Q(1+\frac{12\eta_l\eta_g}{P})(\sigma_l^2+6Q\sigma_g^2)\frac{1}{T}\sum_{t=1}^{T}\overline{\alpha}^t+\frac{12L}{P}\eta_g\eta_l\sigma_l^2\frac{1}{T}\sum_{t=1}^{T}(\overline{\alpha}^t)^2+2L\frac{\eta_g}{\eta_lQP}d\sigma^2\\
&+\frac{1}{T}\sum_{t=1}^{T}\frac{4}{\eta_lQ}\mathbb{E}\left[\left\langle\nabla f(x_t),\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\Delta_i^t-\tilde{\Delta}_i^t\right]\right\rangle+\left\langle\nabla f(x_t),\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\tilde{\Delta}_i^t-\overline{\Delta}_i^t\right]\right\rangle\right]\\
&+\frac{6L}{\eta_lQ}\eta_g\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\Delta_i^t-\tilde{\Delta}_i^t\right\|^2\right]+\frac{6L}{\eta_lQ}\eta_g\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{1}{P}\sum_{i\in\mathcal{P}_t}\tilde{\Delta}_i^t-\overline{\Delta}_i^t\right\|^2\right]. \qquad \text{(D.13)}
\end{aligned}
$$

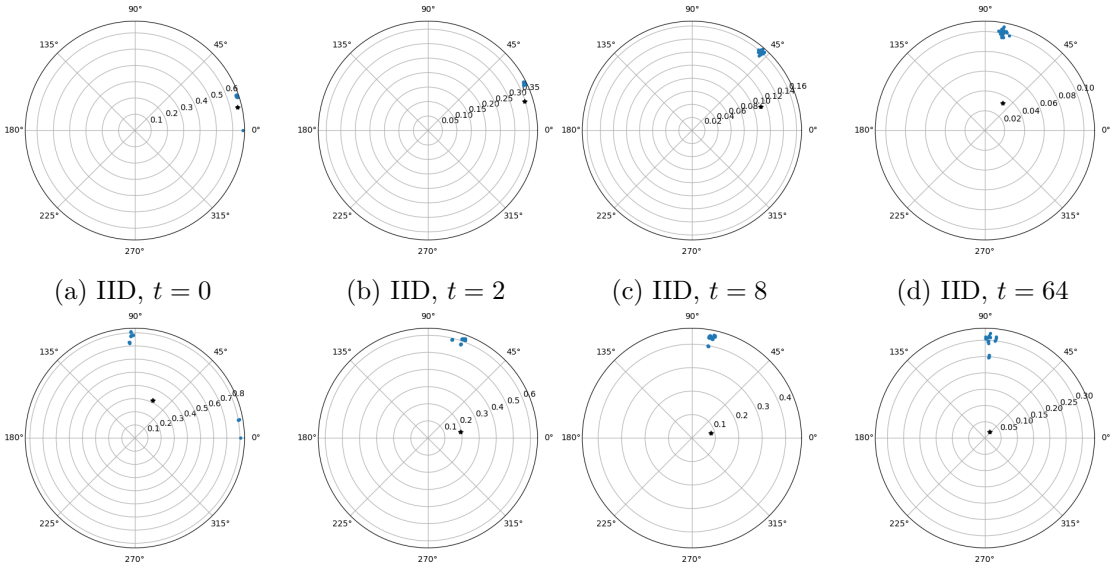
Upper-bounding the last four terms using $\|g_i^{t,q}\|\leq G$ yields the desired result.

D.2 Additional Numerical Experiments

In this part, we provide additional numerical results of Chapter 6

D.2.1 Update Distributions

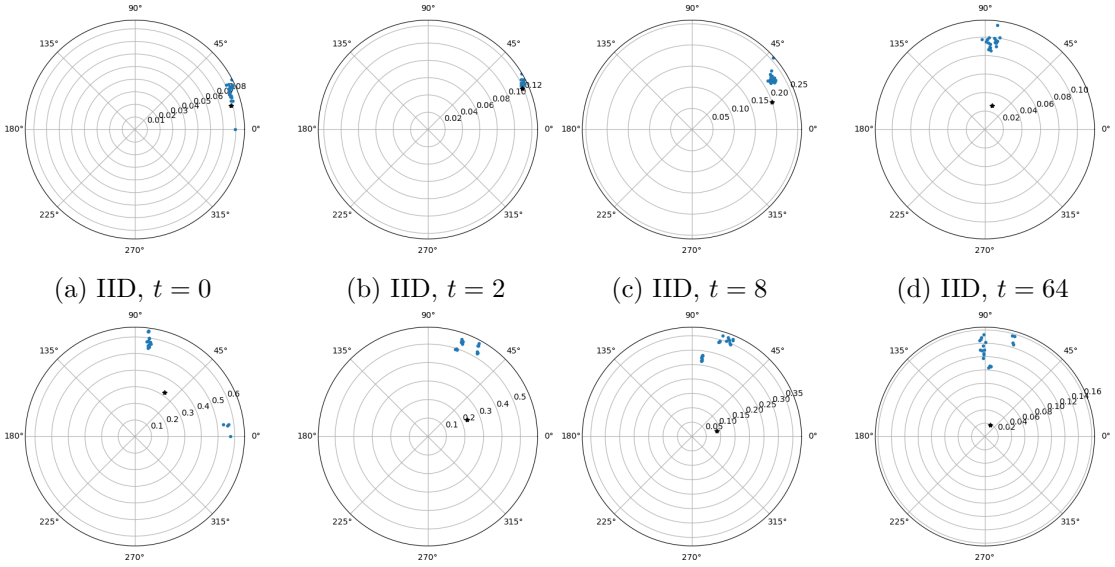
In this part, we plot the change of the distributions of the update differences of different algorithms listed Chapter 6. Notice that in all models and datasets, the distributions of the magnitude in the IID cases are more concentrated than the corresponding Non-IID cases. Also, the distributions of the same model trained on EMNIST dataset are more concentrated than trained on Cifar-10 dataset.



(a) IID, $t = 0$ (b) IID, $t = 2$ (c) IID, $t = 8$ (d) IID, $t = 64$

(e) Non-IID, $t = 0$ (f) Non-IID, $t = 2$ (g) Non-IID, $t = 8$ (h) Non-IID, $t = 64$

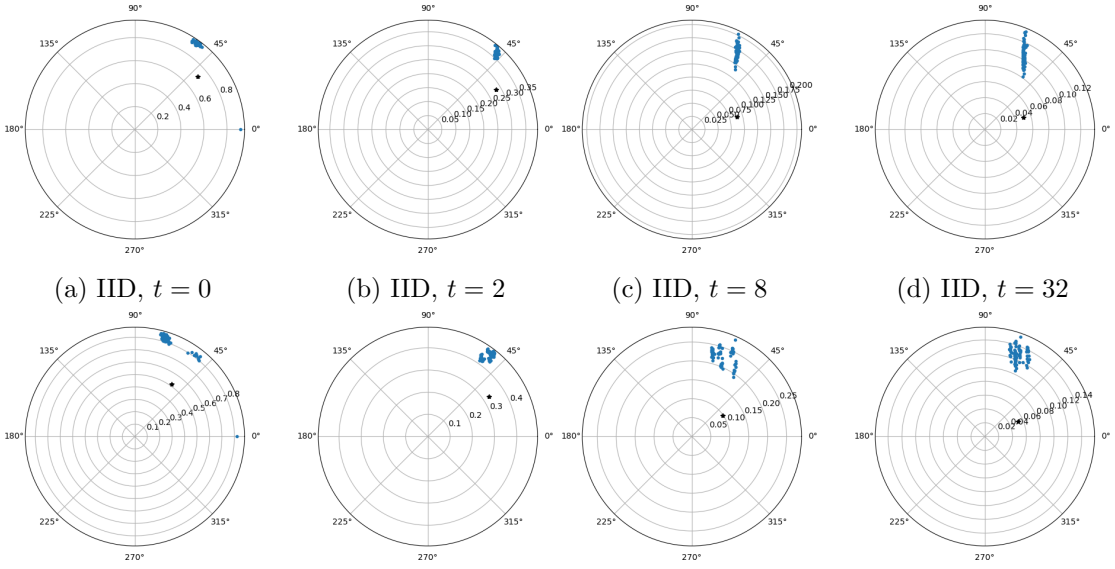
Figure D.1: The distribution of local updates for MLP on IID and Non-IID data at different communication rounds for EMNIST dataset. Each blue dot corresponds to the local update from one client. The black dot shows the magnitude and the cosine angle of global model update at iteration t.



(a) IID, $t = 0$ (b) IID, $t = 2$ (c) IID, $t = 8$ (d) IID, $t = 64$

(e) Non-IID, $t = 0$ (f) Non-IID, $t = 2$ (g) Non-IID, $t = 8$ (h) Non-IID, $t = 64$

Figure D.2: The distribution of local updates for AlexNet on IID and Non-IID data at different communication rounds for EMNIST dataset. Each blue dot corresponds to the local update from one client. The black dot shows the magnitude and the cosine angle of global local model update at iteration t.



(a) IID, $t = 0$ (b) IID, $t = 2$ (c) IID, $t = 8$ (d) IID, $t = 32$

(e) Non-IID, $t = 0$ (f) Non-IID, $t = 2$ (g) Non-IID, $t = 8$ (h) Non-IID, $t = 32$

Figure D.3: The distribution of local updates for ResNet-18 on IID and Non-IID data at different communication rounds for EMNIST dataset. Each blue dot corresponds to the local update from one client. The black dot shows the magnitude and the cosine angle of global local model update at iteration t.

D.3 Quadratic Example

D.3.1 Proof of Claim 3

Given a fixed clipping threshold c, consider the following quadratic problem

$$f(x) = \sum_{i=1}^{3} \frac{1}{2}(x - b_i)^2,$$

where we have $N = 3$ clients. By applying model clipping to FedAvg, one round update can be expressed as:

$$\begin{aligned} x^+ &= \frac{1}{3}\sum_{i=1}^{3} \text{clip}(\lambda x + (1-\lambda)b_i, c), \\ \lambda &= (1-\eta_l)^Q \in (0,1), \end{aligned} \tag{D.14}$$

where η_l is the local stepsize.

Suppose that the algorithm converges, then we will have solution $x^+ = x = x^\infty$. This implies that

$$\frac{1}{3}\sum_{i=1}^{3} \text{clip}(\lambda x^\infty + (1-\lambda)b_i, c) = x^\infty. \tag{D.15}$$

Let us set $b_1 = b_2 = -0.5c, b_3 = kc$, then it is easy to verify that the optimal solution of the problem is given by $x^\star = \frac{(k-1)c}{3} > 0$. However, when $k > 4$, from (D.15) we can see that $x^\infty \leq c$ and $x^\star > c$. Therefore, the only possibility is that $x^\infty = \frac{\lambda}{3-2\lambda}c \leq c \neq x^\star$, and this holds true for any $\lambda \in (0, 1)$. So the stationary solution of FedAvg with model clipping to this problem will not converge to the original optimal solution no matter how we choose Q and η_l.

D.3.2 Proof of Claim 4

First, we prove that using difference clipping, FedAvg can converge to global optimal by carefully selecting Q and η_l. Consider the following convex quadratic problem

$$f(x) = \sum_{i=1}^{N} \frac{1}{2}(A_i x - b_i)^2.$$

By applying FedAvg with update difference clipping, one round of update can be expressed as:

$$x^+ = x - \frac{1}{N}\sum_{i=1}^{N} \text{clip}(\Lambda_i \nabla f_i(x), c), \quad \text{where } \Lambda_i = (I - (I - \eta_l A_i^T A_i)^Q)(A_i^T A_i)^{-1}. \tag{D.16}$$

In order for the problem to converge to the original problem, it is easy to verify that the following condition has to hold:

$$\sum_{i=1}^{N} \text{clip}(\Lambda_i \nabla f_i(x^\star), c) = 0.$$

The above example can be viewed as using gradient descent to optimize a problem with the following gradient

$$\nabla f_i'(x) = \begin{cases} \Lambda_i \nabla f_i(x) & \|\Lambda_i \nabla f_i(x)\| \leq c, \\ \frac{c\Lambda_i \nabla f_i(x)}{\|\Lambda_i \nabla f_i(x)\|} & \text{otherwise.} \end{cases} \tag{D.17}$$

Note that in general it is hard to write down the exact local problems f_i' that satisfies the above condition, but when $x \in \mathbb{R}$ is a scalar, $f_i'(x)$ is the Huberized loss of $\Lambda_i f_i(x)$ [54]

$$f_i'(x) = \begin{cases} \Lambda_i f_i(x) & \text{if } |\Lambda_i A_i (A_i x - b_i)| \leq c, \\ c\left|\frac{\Lambda_i}{A_i} f_i(x)\right| - \frac{1}{2}c^2 & \text{otherwise.} \end{cases} \tag{D.18}$$

In general, the re-weighted problem does not have the same solution as the original problem, but we can select η_l and Q (determined by on $x^\star$ and f_i's) so that $f'(x)$ has the same solution as $f(x)$. For example, one set of parameters that satisfy the above requirement is $Q = 1, \eta_l = 1/\max_i\{\|\nabla f_i(x^\star)\|\}$. In this case, $\Lambda_i = I\eta_l$, and when η_l is small enough, the clipping will not be activate when $x = x^\star$ and $\sum_{i=1}^N \text{clip}(\Lambda_i \nabla f_i(x^\star), c) = \sum_{i=1}^N \eta_l \nabla f_i(x^\star) = 0$.

Next, we show that Clipping-enabled FedAvg can outperform the non-clipped version. Note that when $Q > 1$, even when η is small such that the clipping is not activated, the algorithm will not converge to the original solution. So in general one cannot draw the conclusion about whether clipping helps or hurts the performance of FedAvg. Consider the following problem:

$$f(x) = \sum_{i=1}^{3} f_i(x), \quad \text{where } f_1(x) = \frac{1}{2}(x-4)^2,\ f_2(x) = \frac{1}{2}(2x-1)^2,\ f_3(x) = \frac{1}{2}(6x+1)^2. \tag{D.19}$$

As $\nabla f(x) = (x-4)+(4x-2)+(36x+6) = 41x$, the optimal solution of this problem is $x^\star = 0$. Table D.1 show the stationary points of FedAvg under different choice of parameters. When $Q = 1$, FedAvg is equivalent to SGD and clipping hurts the performance of FedAvg. However, when Q is large, clipped FedAvg has a better performance than the non-clipped version, in the sense that the stationary solution it obtains are closer to the global optimal solution $x^* = 0$.

	$Q = 1$	$Q = \infty$
$c = \infty$	$x^\infty = 0$	$x^\infty = \frac{13}{9}$
$c = 1$	$x^\infty = \frac{1}{2}$	$x^\infty = \frac{2}{3}$

Table D.1: Stationary points of FedAvg with gradient clipping for (D.19) under different parameter settings.