ABSTRACT

Title of dissertation: EFFICIENT OPTIMIZATION ALGORITHMS FOR NONCONVEX MACHINE LEARNING PROBLEMS

Wenhan Xian, Doctor of Philosophy, 2024

Dissertation directed by: Professor Heng Huang
Department of Computer Science

In recent years, the success of the AI revolution has led to the training of larger neural networks on vast amounts of data to achieve superior performance. These powerful machine learning models have enabled the creation of remarkable AI products. Optimization, as the core of machine learning, becomes especially crucial because most machine learning problems can ultimately be formulated as optimization problems, which require minimizing a loss function with respect to model parameters based on training samples.

To enhance the efficiency of optimization algorithms, distributed learning has emerged as a popular solution for addressing large-scale machine learning tasks. In distributed learning, multiple worker nodes collaborate to train a global model. However, a key challenge in distributed learning is the communication cost. This thesis introduces a novel adaptive gradient algorithm with gradient sparsification to address this issue.

Another significant challenge in distributed learning is the communication overhead on the central parameter server. To mitigate this bottleneck, decentralized distributed (serverless) learning has been proposed, where each worker node only needs to communicate with its neighbors. This thesis investigates core nonconvex optimization problems in

decentralized settings, including constrained optimization, minimax optimization, and second-order optimality. Efficient optimization algorithms are proposed to solve these problems.

Additionally, the convergence analysis of minimax optimization under the generalized smooth condition is explored. A generalized algorithm is proposed, which can be applied to a broader range of applications.

EFFICIENT OPTIMIZATION ALGORITHMS FOR NONCONVEX MACHINE LEARNING PROBLEMS

by

Wenhan Xian

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2024

Advisory Committee:
Professor Heng Huang, Chair/Advisor
Professor Min Wu
Professor Tianyi Zhou
Professor Furong Huang
Professor Ang Li

Preface

In recent years, we have witnessed an AI revolution and the remarkable success of machine learning. Larger neural networks are being trained on increasingly vast amounts of data to achieve superior performance. These powerful machine learning models have enabled the creation of groundbreaking AI products, such as ChatGPT and Full Self-Driving (FSD) technology, which have significantly impacted our lives.

In this context, optimization, as the core of machine learning, becomes especially critical. Most machine learning problems can ultimately be formulated as optimization problems, requiring the minimization of a loss function based on training samples to obtain optimal model parameters. Given the large scale of models and the immense volume of data, there is an urgent need for designing fast and stable algorithms to solve these optimization problems.

This need has motivated my research on optimization in machine learning. In this dissertation, I propose more efficient optimization methods to address some of the core issues in machine learning.

Acknowledgments

It is nearly impossible to thank everyone who contributed to my successful completion of this dissertation. In fact, some of the people mentioned below might not remember their relevance in my progress, however, their impact was substantial and will not be forgotten.

First and foremost, this work would have been impossible without funding from The University of Maryland Department of Computer Science and Graduate School, and the University of Pittsburgh Department of Electrical and Computer Engineering. I would like to thank each of them for their support.

On more personal notes, I would like to thank my advisor Dr. Heng Huang. Heng has been a wonderful mentor, collaborator and friend. Aside from his help in successfully completing this dissertation, I am quite proud of the work we have done together. Bisides, I would like to thank my collaborators Bin Gu, Zhouyuan Huo, Feihu Huang, Yanfu Zhang, Lei Luo, Runxue Bao, Xidong Wu, Peiran Yu and Ziyi Chen, for always providing much needed direction and advice. I also acknowledge all group members in my laboratory for their friendship.

My family: my mother, father, grandmother and grandfather. This dissertation should really be dedicated to them because they are always my strongest support. I also would like to thank my cat Xiandan, who never forgets to remind me that there are more important things than work. Finally, I would like to thank my girlfriend Lingwei Li, who has accompanied me during the last two years of my Ph.D. program and who is together with me to overcome the most difficult times.

Table of Contents

Chapter 1: Introduction

In recent years, the success of machine learning has led to the training of larger neural networks on vast amounts of data to achieve better performance. These powerful machine learning models have enabled the creation of remarkable AI products. Optimization, as the core of machine learning, becomes especially crucial because most machine learning problems can ultimately be formulated as optimization problems. Typically, it is required to minimize a loss function based on the training samples to obtain the best model parameters. Apart from accuracy, training efficiency is also an important issue, especially for large-scale problems. Therefore, it is necessary to study more efficient optimization algorithms for machine learning problems.

1.1 Distributed Learning

Nowadays, with an increasing number of machine learning tasks requiring large models and datasets for optimal performance, distributed learning has emerged as a promising research area within the machine learning community. In distributed learning, multiple worker nodes collaborate to train a global model based on the data they can access, coordinated by a central server node. Since large-scale data are distributed and stored at different worker nodes and complex computations can be performed in parallel, distributed learning can tackle these challenging large-scale tasks effectively and efficiently.

1.2 Decentralized Learning

Despite the success of distributed learning, the conventional centralized scheme has a key bottleneck of communication, where the communication burden on the central server becomes larger as the number of nodes grows. For example, when the system has M workers, it will suffer from a communication complexity of $O(M)$. Thus, the decentralized distribution structure recently has attracted much attention in machine learning due to its communication efficiency compared with the centralized approach. Specifically, decentralized optimization adopts a pattern where each node maintains its own local data and model and only communicates with its neighbors. In fact, the communication complexity of decentralized learning at each iteration depends on the degree of graph topology (usually independent of the number of nodes).

1.3 Minimax Optimization

Minimax optimization has numerous applications in machine learning tasks such as Generative Adversarial Net (GAN) [32], adversarial training [78] and multi-agent reinforcement learning [116]. The fundamental idea behind minimax optimization is to find the optimal solution for a problem in the presence of an adversary, where one party seeks to minimize a payoff loss function while the other seeks to maximize it. Specifically, variable x aims to minimize an objective function $f(x,y) : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}$ while variable y tries to maximize the loss, which can be formulated as

$$\min_{x \in \mathscr{X}} \max_{y \in \mathscr{Y}} f(x,y), \tag{1.1}$$

where $\mathscr{X} \subseteq \mathbb{R}^{d_1}$ and $\mathscr{Y} \subseteq \mathbb{R}^{d_2}$.

1.4 Outline

In this dissertation, several efficient optimization algorithms are proposed to address specific machine learning problems. In Chapter 2, a decentralized communication-efficient Frank-Wolfe algorithm is proposed to solve decentralized constrained optimization problems. In Chapter 3, an accelerated decentralized minimax algorithm is proposed to solve decentralized minimax problems with better convergence rate. In Chapter 4, a communication-efficient adaptive gradient algorithm with gradient compression is proposed, which can reduce the communication cost of the distributed learning system. In Chapter 5, a decentralized perturbed gradient based algorithm is proposed, which can escape saddle point and achieve second-order stationary point efficiently in the decentralized setting. In Chapter 6, a generalized minimax algorithm with adaptive stepsize is proposed, which is guaranteed to converge under the generalized smooth condition.

Chapter 2: Decentralized Constrained Optimization

2.1 Introduction

Recently, decentralized learning becomes a popular research topic in machine learning and has been widely studied. For example, it was first studied to solve problems of computing aggregates among clients or be used for the sake of data locality and privacy [100, 130], where a centralized structure is not allowed. A general decentralized algorithm can be traced back to [85] that combines gradient descent method and Gossip-type consensus step. Subsequently, a degenerated case of decentralization that achieves huge success is federated optimization [52], which adopts star topology but enables data to be drawn from non-iid distribution. Besides, many other previous fully decentralized works such as [63, 106] that based on general network topology have shown that decentralized method is able to achieve more efficient communication without sacrificing the training result, indicating that decentralization is becoming competitive and advantageous in distributed learning rather than merely an alternate of centralization when centralization is not possible. [63] presented an important decentralized optimization work to verify that the decentralized method can outperform its centralized counterpart. [63] proposed an algorithm named Decentralized Parallel Stochastic Gradient Descent (D-PSGD) to directly compute the averaging value among each node with exact communication, which has the same convergence rate as centralized SGD in nonconvex optimization with non-identical data distribution.

To reduce the communication cost in distributed systems, gradient quantization [97] is

another effective method. Recently, many quantized gradient algorithms, such as QSGD [3], signSGD [6] and its variant [7], were developed and have shown excellent performance. In these algorithms, the number of bits transmitted in each communication round is reduced by packing and unpacking gradients. [3] proposed an unbiased quantization scheme and proved it can converge under convex and non-convex conditions. However, for other quantization methods like 1-bit quantization or signSGD, the unbiased assumption is not always satisfied. [49] proved that when applying signSGD with a scalar factor and error-feedback technique, the algorithm is guaranteed to converge in non-convex optimization. More recently, to further achieve communication efficiency, multiple quantized decentralized algorithms [22, 51, 94, 95] have been introduced. However, to the best of our knowledge, existing quantized decentralized algorithms for constrained problem are still very limited. In fact, large-scale constrained optimization problems are common in many machine learning applications, such as matrix completion and deep neural network compression.

To address this challenging issue, in this chapter, we focus on studying the quantized decentralized algorithm for solving the following constrained optimization problem:

$$\min_{x\in\Omega} \frac{1}{M}\sum_{i=1}^{M} f_i(x), \tag{2.1}$$

where $f_i(x)$ is a nonconvex smooth loss function, Ω is a convex and compact constraint set, M is the number of worker nodes. $f_i(x)$ is the objective function on node i and could have the stochastic expectation or finite sum formulations:

$$f_i(x) = \begin{cases} \mathbb{E}_{\xi\sim D_i} F^{(i)}(x;\xi), & \text{stochastic} \\ \frac{1}{n_i}\sum_{j=1}^{n_i} F_j^{(i)}(x), & \text{finite-sum} \end{cases} \tag{2.2}$$

where D_i is the data distributed on i-th node. Finite-sum objective function is a particular

case of stochastic problem where D_i consists of finite samples. We allow distributions D_i to be **non-identical**, which is more adaptive to general tasks in machine learning and is assumed in many previous decentralized analyses [63, 65, 106].

To solve the above constrained optimization problem, the Frank-Wolfe (a.k.a, conditional gradient or projection-free) method is one of the most efficient and popular algorithms, because the Frank-Wolfe method only requires to compute a linear oracle instead of the expensive projection operator applied in proximal gradient methods [31] and alternating direction method of multipliers [38]. In this chapter, thus, we focus on designing the communication-efficient quantized decentralized Frank-Wolfe algorithm to solve the problem (2.1). It is nontrivial to design such an algorithm. We first provide a counterexample to show that the vanilla quantized decentralized Frank-Wolfe algorithm usually diverges (please see the following Counterexample section). Thus, there exists an important research problems to be addressed:

Can we design a communication-efficient quantized decentralized Frank-Wolfe algorithm with convergence guarantee for non-convex optimization?

In this chapter, we address the above challenging question with a positive solution and propose a novel Decentralized Quantized Stochastic Frank-Wolfe (DQSFW) algorithm to solve the problem (2.1). By using the gradient tracking technique, we ensure that DQSFW can safely and quickly converge to a stationary point in non-convex optimization. Specifically, our DQSFW algorithm employs a 1-bit gradient quantization scheme. In summary, the main **contributions** of this chapter are given as follows:

(1) We propose a novel efficient Decentralized Quantized Stochastic Frank-Wolfe (DQSFW) method to solve the problem (2.1), which reduces communication cost but maintains good convergence speed.

(2) We derive the rigorous theoretical analysis for our DQSFW algorithm, and prove that

our DQSFW algorithm has the same gradient complexity $O(\varepsilon^{-4})$ as the SFW [93] (the sequential algorithm) and QFW [135] (the centralized algorithm), but with much lower communication cost.

(3) We provide a new intuitive counterexample to show that the decentralized optimization involving non-linear projection of gradient could lead to a potential divergent problem which also exists in many cases where we generalize other non-Frank-Wolfe methods to decentralized algorithms. To tackle this challenge, we utilize the gradient tracking technique to guarantee the convergence of our decentralized quantized Frank-Wolfe algorithm.

Notations

$\|\cdot\|_1$ denotes the L_1 norm of vector. $\|\cdot\|_2$ denotes the spectral norm of a matrix. $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. $\|\cdot\|_*$ denotes trace norm of a matrix. Let $\mathbf{1}$ be the column vector, each entry of which is one. Given a network with M nodes, we define a mixing matrix $W = (w_{ij}) \in \mathbb{R}^{M\times M}$ that represents the weights of neighbors in the communication round. For example, in D-PSGD [63], the consensus step on i-th node is formulated as

$$x_t^{(i)} = \sum_{i=1}^{M} w_{ij} x_t^{(j)}.$$

Generally, W is a symmetric doubly stochastic matrix that satisfies $W^T = W$ and $W\mathbf{1} = \mathbf{1}$. In the experiment section, we will consider a uniformly weighted ring-based network, whose

Algorithm	DeFW	QFW	DQSFW
Decentralized	$\surd$	$\times$	$\surd$
Stochastic	$\times$	$\surd$	$\surd$
Quantized	$\times$	$\surd$	$\surd$
Reference	[115]	[135]	Ours

Table 2.1: Comparison of related works.

mixing matrix is shown as Eq. (2.3).

$$W = \begin{bmatrix} 1/3 & 1/3 & 0 & \cdots & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 & \cdots & 0 \\ 0 & 1/3 & 1/3 & 1/3 & 0 & \cdots \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 0 & \cdots & 0 & 1/3 & 1/3 \end{bmatrix} \tag{2.3}$$

2.2 Related Works

2.2.1 Decentralized Frank-Wolfe

Decentralized Frank-Wolfe algorithm (DeFW) [115] was recently proposed to apply deterministic Frank-Wolfe method in decentralized structure. It is guaranteed to converge in both convex and non-convex problems. The authors compute net averaging on parameters and gradients. In previous work about decentralization such as [63], averaging of gradients is not required. Compared with DeFW, our DQSFW changes the deterministic algorithm to a stochastic one, which is more adaptive to large scale machine learning tasks. When the number of samples is very large, the full gradient is too expensive to calculate. Besides, we also take advantage of the technique of gradient quantization, which will further reduce the

cost of communication.

2.2.2 Quantized Frank-Wolfe

Quantized Frank-Wolfe algorithm (QFW) [135] was recently proposed to solve centralized distributed problems. It uses the the following momentum scheme as gradient estimator:

$$\bar{g}_t = (1-\rho_t)\bar{g}_{t-1} + \rho_t g_t \tag{2.4}$$

which is also used in [73] as a way to decrease the noise of gradient. In our algorithm, we combine this momentum scheme with the Gossip method. For gradient compressing, it adopts the s-Partition Encoding Scheme, which encodes the i-th coordinates g_i into an element from the set $\{\pm 1, \pm\frac{s-1}{s}, \cdots, \pm\frac{1}{s}, 0\}$. It requires $\log_2(s+1)$ bits to transfer each coordinate of the gradient. A scalar factor $\|g\|_\infty$ is also transmitted thus the total bits of quantized gradient is $32 + d \cdot \log_2(s+1)$ where d is the dimension of gradient. When s is large, the variance of this compressor will become small [135], which means the quantized gradient is more precise, but costs more bits. In this chapter, we use 1-bit signSGD scheme with a scalar factor [49, 51] shown as follows:

$$C(x) = \frac{\|x\|_1}{d}\text{sign}(x) \tag{2.5}$$

where d is the dimension of x. **Notice that** here signSGD is only a representative of feasible compressors. We can also use other compressor as long as it satisfies the Compressor Assumption in section 4, which is an important assumption in many theoretical analysis of related work about gradient quantization. For example, we can also use top-k SGD, which is a gradient sparsification method that automatically satisfies the Compressor Assumption.

We consider signSGD because it is efficient and convenient to implement.

The comparisons between DeFW, QFW and our DQSFW are summarized in Table 2.1. We can see that our DQSFW algorithm is the first to incorporate stochastic gradient descent and gradient quantization in decentralized Frank-Wolfe type algorithm.

2.3 Counterexample

In this section, we provide an intuitive counterexample that demonstrates the divergent trap if Frank-Wolfe method is simply generalized to the decentralized algorithm without making consensus on gradient when data at different nodes are drawn from **non-identical distributions**.

Given $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$, where $\mathbf{x} \in \Omega = \{(x,y)|x^2+y^2 \leq 1\}$, $f_1(\mathbf{x}) = x$ and $f_2(\mathbf{x}) = \sqrt{3}y$ (See Figure 2.1). We can calculate gradients $\mathbf{v}_1 = (1,0)$, $\mathbf{v}_2 = (0,\sqrt{3})$, $\mathbf{v} = (1,\sqrt{3})$. Since the gradient is never equal to 0, according to the Frank-Wolfe gap (see Eq. (9)), the only stationary point is $(-\frac{1}{2}, -\frac{\sqrt{3}}{2})$ (blue point), where the tangent of unit ball is vertical to direction $(1,\sqrt{3})$. However, if we update $\mathbf{x}$ by Frank-Wolfe algorithm on each node separately, the linear oracle will yield $\mathbf{d}_1 = (-1,0)$ and $\mathbf{d}_2 = (0,-1)$. Then we make consensus on $\mathbf{x}$ and get iteration formula $\mathbf{x}_{i+1} = (1-\gamma)\mathbf{x}_i + \gamma(-\frac{1}{2}, -\frac{1}{2})$. Sequence $\mathbf{x}_i$ eventually converges to point $(-\frac{1}{2}, -\frac{1}{2})$ (red point), which is not a stationary point.

It is reasonable to attribute the divergence to the non-commutative relationship between the linear oracle and addition. For SGD-based decentralized learning algorithms, convergence is achieved due to the commutative property of addition. The above divergence problem is also likely to occur in other variant algorithms of SGD that involve non-linear mappings of gradients in a decentralized system, not just Frank-Wolfe type methods. For example, adaptive gradient methods are a family of algorithms that adjust the learning rate according to the magnitude of the gradient. The Decentralized ADAM algorithm (DADAM)

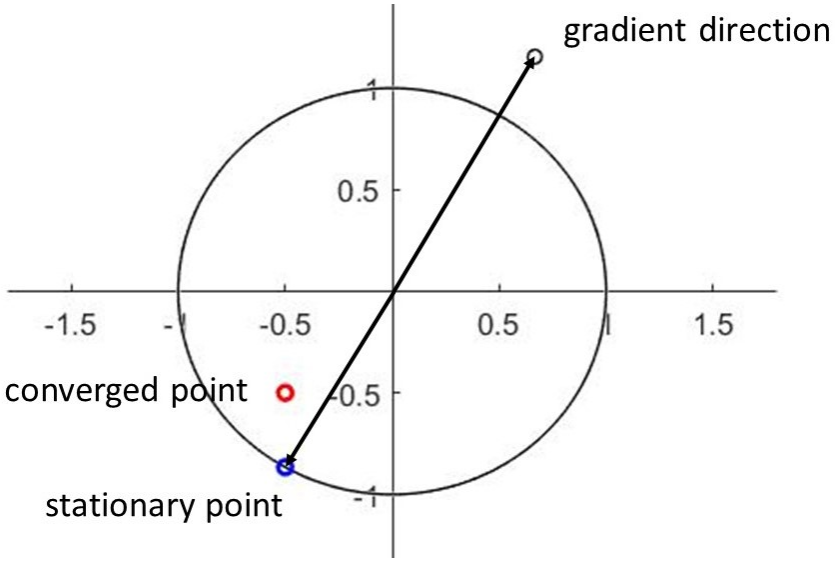


Figure 2.1: A graph to demonstrate the counter-example.

[83] was proved to converge under the criterion named local regret. Nonetheless, local regret can lead to a situation where each node converges to its own local stationary point, without sufficient cooperation across the entire system. This phenomenon highlights that when we generalize an algorithm involving steps that are not commutative with addition, similar divergence problems are likely to arise.

In DeFW [115], the gradients can be averaged directly by gradient tracking, a technique to accelerate consensus in distributed optimization [84, 126]. DIGing also considers the increment of gradient when averaging the non-quantized full gradient. **However**, in this chapter we have to face the variance of stochastic gradient and the noise of quantization. These issues do not occur in DeFW. Therefore, we have to use a new strategy to let them make a gradual consensus.

2.4 New Decentralized Quantized Stochastic Frank-Wolfe Algorithm

In this section, we propose a novel efficient Decentralized Quantized Stochastic Frank-Wolfe (DQSFW) algorithm to solve the problem (2.1) by using the gradient tracking tech-

nique [115, 126]. DQSFW algorithm is given in Algorithm 1.

In Algorithm 1, $x_t^{(i)}$ is a column vector that denotes the model parameter on i-th node in iteration t. We use upper case X_t to represent the matrix

$$X_t = [x_t^{(1)}, x_t^{(2)}, \ldots, x_t^{(M)}]$$

Inspired by Choco-Gossip algorithm [51], we also define a replicated $\hat{x}_t^{(i)}$ of $x_t^{(i)}$ on each node. The reason is when we apply gossip update, the exact value of model parameter on other nodes are unknown since there exists quantization and the communication is inexact. The replica $\hat{x}_t^{(i)}$ is an estimation of $x_t^{(i)}$, which is also updated at each iteration. And the consensus step is formulated as line 8 in Algorithm 1. According to the update of $\hat{x}_t^{(i)}$ (line 9 and line 10 in Algorithm 1), on all neighbors of the i-th node, the replica is added by an identical transmitted message $z_t^{(i)}$, which implies the values $\hat{x}_t^{(i)}$ on all neighbors of the i-th node are consistent. Therefore, replica $\hat{x}_t^{(i)}$ is well-defined. Similar to X_t, we also define matrices

$$\begin{aligned} \hat{X}_t &= [\hat{x}_t^{(1)}, \hat{x}_t^{(2)}, \ldots, \hat{x}_t^{(M)}], \\ \overline{X}_t &= [\bar{x}_t, \bar{x}_t, \ldots, \bar{x}_t] \end{aligned}$$

where $\bar{x}_t$ represent the mean value: $\bar{x}_t = \frac{1}{M}\sum_{i=1}^{M} x_t^{(i)}$.

$g_t^{(i)}$ is a stochastic gradient on i-th node calculated by selected samples and $v_t^{(i)}$ is our key estimation of the gradient on i-th node which is defined as Eq. (2.6) with a kind of momentum scheme. Here $\hat{v}_t^{(i)}$ is the replica of $v_t^{(i)}$ (see similar concept of $\hat{x}_t^{(i)}$). For initialization, we set $\hat{v}_{-1}^{(i)} = \mathbf{0}$ and $v_{-1}^{(i)} = g_0^{(i)}$. The definition and role of β_t will be discussed later in Remark 2. Our convergence analysis shows that, though our gradient estimator in line 4 is biased, the gradients on all nodes are getting close to the full gradient uniformly. To make consensus on

Algorithm 1 Decentralized Quantized Stochastic Frank-Wolfe (DQSFW)

Input: restricted domain Ω, matrix W, initial point $\hat{X}_0 = X_0 \in \Omega$
Parameter: η_t, γ_t, β_t, α_t, T
Output: $\bar{x}_{\hat{t}}$, where $\hat{t}$ is chosen uniformly from $\{0, 1, \ldots, T\}$

1: On i-th node:
2: **for** $t = 0, 1, \ldots, T-1$ **do**
3: Compute an estimation of the gradient $g_t^{(i)}$
4: Update $v_t^{(i)} = (1-\beta_t)v_{t-1}^{(i)} + \beta_t g_t^{(i)} + \alpha_t \sum_j w_{ij}(\hat{v}_{t-1}^{(j)} - \hat{v}_{t-1}^{(i)})$
5: Compute $q_t^{(i)} = C(v_t^{(i)} - \hat{v}_{t-1}^{(i)})$ and communicate with neighbors
6: Update replica $\hat{v}_t^{(j)} = \hat{v}_{t-1}^{(j)} + q_t^{(j)}$ for neighbor j
7: Calculate linear oracle $d_t^{(i)}$ such that $d_t^{(i)} = \arg\max_{d \in \Omega} \langle d, -v_t^{(i)} \rangle$
8: Update $x_{t+1}^{(i)} = (1-\eta_t)x_t^{(i)} + \eta_t d_t^{(i)} + \gamma_t \sum_j w_{ij}(\hat{x}_t^{(j)} - \hat{x}_t^{(i)})$
9: Compute $z_t^{(i)} = C(x_{t+1}^{(i)} - \hat{x}_t^{(i)})$ and communicate with neighbors
10: Update replica $\hat{x}_{t+1}^{(j)} = \hat{x}_t^{(j)} + z_t^{(j)}$ for neighbor j
11: **end for**

gradient and parameter, we adopt the gossip update [51] in line 4 and line 8 respectively.

$$v_t^{(i)} = (1-\beta_t)v_{t-1}^{(i)} + \beta_t g_t^{(i)} + \alpha_t \sum_j w_{ij}(\hat{v}_{t-1}^{(j)} - \hat{v}_{t-1}^{(i)}) \tag{2.6}$$

In line 5 and line 9 of Algorithm 1, we apply a gradient quantization method that satisfies the Compressor Assumption. As mentioned previously, the quantization scheme is not limited to the signSGD used in this chapter. Line 7 is the typical linear oracle in Frank-Wolfe method to get a direction $d_t^{(i)}$. In the Frank-Wolfe vanilla algorithm, the update

of $x_t^{(i)}$ should be $x_{t+1}^{(i)} = x_t^{(i)} + \eta_t(d_t^{(i)} - x_t^{(i)})$. For convenience, we define matrices

$$\begin{aligned} V_t &= [v_t^{(1)}, v_t^{(2)}, \ldots, v_t^{(M)}], \\ \hat{V}_t &= [\hat{v}_t^{(1)}, \hat{v}_t^{(2)}, \ldots, \hat{v}_t^{(M)}], \\ \overline{V}_t &= [\bar{v}_t, \bar{v}_t, \ldots, \bar{v}_t], \\ D_t &= [d_t^{(1)}, d_t^{(2)}, \ldots, d_t^{(M)}], \\ \overline{D}_t &= [\bar{d}_t, \bar{d}_t, \ldots, \bar{d}_t] \end{aligned}$$

where $\bar{v}_t$ and $\bar{d}_t$ are mean values

$$\bar{v}_t = \frac{1}{M}\sum_{i=1}^{M} v_t^{(i)}, \quad \bar{d}_t = \frac{1}{M}\sum_{i=1}^{M} d_t^{(i)}$$

By the doubly stochastic property of W, we have

$$\overline{X}_{t+1} = (1-\eta_t)\overline{X}_t + \eta_t \overline{D}_t \tag{2.7}$$

It is easy to verify that when $x_0 \in \Omega$, $\bar{x}_t \in \Omega$ for $\forall t$. Hence the constraint is always satisfied. Here we should note that we do not have to store all the replica in practice. We can regard $\sum_j w_{ij}(\hat{x}_t^{(j)} - \hat{x}_t^{(i)})$ as a term, and obtain iteration formula

$$\sum_j w_{ij}(\hat{x}_{t+1}^{(j)} - \hat{x}_{t+1}^{(i)}) = \sum_j w_{ij}(\hat{x}_t^{(j)} - \hat{x}_t^{(i)}) + \sum_j w_{ij}(z_t^{(j)} - z_t^{(i)}). \tag{2.8}$$

Therefore, we only need one buffer with the size of x_t to compute this term. So, it is with $\sum_j w_{ij}(\hat{v}_t^{(j)} - \hat{v}_t^{(i)})$. We will use Eq. (2.8) to save memory in our experiments.

2.5 Convergence Analysis

In this section, we study the convergence properties of our DQSFW algorithm. All proofs can be found in the Appendix. We begin by introducing some mild assumptions and the definition of the Frank-Wolfe gap [44]:

$$\mathscr{G}(x) = \max_{v \in \Omega} \langle v - x, -\nabla f(x) \rangle \tag{2.9}$$

The convergence criteria is $\mathbb{E}\|\mathscr{G}(x)\| \leq \varepsilon$.

Assumption 2.1. *(Lipschitz Gradient) There is a constant L such that for* $\forall i \in \{1, 2, \ldots, M\}$*, we have*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|. \tag{2.10}$$

Assumption 2.2. *(Compact Domain) There is a diameter D of domain* Ω*.*

Assumption 2.3. *(Lower Bound) Function* $f(x)$ *has the lower bound* $\inf_{x \in \Omega} f(x) = f^- > -\infty$*.*

Assumption 2.4. *(Spectral Gap) Given the doubly stochastic symmetric matrix W, we define* $\lambda_1, \lambda_2, \ldots, \lambda_M$ *to be its eigenvalues in descending order. Then* $\max\{|\lambda_2|, |\lambda_M|\} < 1$*. Let* $\rho = \max\{|\lambda_2|, |\lambda_M|\}$ *and* $\zeta = 1 - \lambda_M$*.*

Assumption 2.5. *(Compressor Assumption) Compressor* $C(\cdot)$ *satisfies* $\|C(x) - x\|^2 \leq (1 - \delta)\|x\|^2$*, where* $0 < \delta \leq 1$*.*

Assumption 2.6. *(Bounded Gradient and Bounded Variance) The generated gradient estimator* $g_t^{(i)}$ *satisfies* $\mathbb{E}[g_t^{(i)}] = \nabla f_i(x_t^{(i)})$*,* $\mathbb{E}\|g_t^{(i)} - \nabla f_i(x_t^{(i)})\|^2 \leq \sigma^2$*,* $\|\nabla F^{(i)}(x_t^{(i)}; \xi)\| \leq G$*.*

Based on above assumptions, we demonstrate three lemmas of our DQSFW algorithm. Lemma 1 and Lemma 2 estimate the consensus between model parameter X_t and gradient V_t respectively. Lemma 3 implies that our gradient estimator Eq. (2.6) on different node approaches the value of full gradient uniformly and gradually. The detailed proof of lemmas can be found in Appendix A.2.

Lemma 2.1. *Let* $\delta_0 = 1-\sqrt{1-\delta^2}$, $\delta_1 = 1-(1-\delta_0^2)^2$. $\alpha_t = \gamma_t = \gamma = min\{1, \frac{\delta_0}{\zeta}, \frac{(1-\rho)\delta_1}{2\zeta^2}\}$. $c_1 = (1-\rho)\gamma$, $c_2 = \delta$, $c_3 = min\{\frac{(1-\rho)\gamma}{2}, \frac{\delta_1}{2}\}$. $A = (1+c_1)(1-(1-\rho)\gamma)^2 + (1-\delta)(1+\frac{1}{c_2})\gamma^2\zeta^2$, $B = (1+\frac{1}{c_1})\gamma^2\zeta^2 + (1-\delta)(1+c_2)(1+\gamma\zeta)^2$. *Let* $Q = 8(1+\frac{1}{c_3})(A+B)$. *Set* $\eta_0 = \frac{c_3^3}{16(1+c_3)(A+B)}$ *and* $\eta_t = \frac{\eta_0}{(t+1)^\theta}$, $0 < \theta < 1$. *Then there exists a constant* R_1 *satisfying*

$$\|X_t - \overline{X_t}\|_F^2 + \|X_t - \hat{X}_t\|_F^2 \leq \frac{QR_1MD^2}{(t+1)^{2\theta}}$$

Lemma 2.2. *Let* $c_4 = c_3^2$, $\beta_0 = \frac{B(1+c_4)}{Ac_4}$ *and* $\beta_t = \frac{\beta_0}{(t+1)^{2\theta/3}}$. *Then there exists constant* R_2 *such that*

$$\|V_t - \overline{V_t}\|_F^2 + \|V_t - \hat{V}_t\|_F^2 \leq \frac{QR_2MG^2}{(t+1)^{2\theta/3}}$$

Lemma 2.3. *Denote* $\bar{v}_t = \frac{1}{M}\sum_{i=1}^M v_t^{(i)}$. *There exists constant S such that*

$$\mathbb{E}\|\bar{v}_t - \frac{1}{M}\sum_{i=1}^M \nabla f_i(x_t^{(i)})\|^2 \leq \frac{S}{(t+1)^{2\theta/3}}$$

Next, we will propose the main theorem of our convergence analysis. Please check the detailed proof in Appendix A.3.

Theorem 2.1. *Let Q,* R_1, R_2 *and S be the constants defined in Lemma 1 to Lemma 3. stepsize* η_t *is set as Lemma 1. Then we have*

$$\mathbb{E}[\mathscr{G}(\bar{x}_t)] \leq \mathbb{E}[\frac{f(\bar{x}_t - f(\bar{x}_{t+1})}{\eta_t})] + \frac{D\sqrt{2(S+QR_1L^2D^2)}}{(t+1)^{\theta/3}} + \frac{D\sqrt{QR_2}G}{(t+1)^{\theta/3}} + \frac{\eta_t L^2D^2}{2}$$

Theorem 2.2. *Suppose T iterations have been completed. Let $\hat{t}$ is chosen randomly with identical probability from $\{0,1,\ldots,T\}$. Set $\theta = \frac{3}{4}$. Then by* **Theorem 1** *we can obtain*

$$\mathbb{E}[\mathscr{G}(\bar{x}_{\hat{t}})] = O(\frac{1}{T^{1/4}}).$$

Remark 2.1. *Theorem 2.2 shows that our DQSFW algorithm reaches a gradient complexity of $O(\varepsilon^{-4})$ to achieve an ε-stationary point. And the Frank-Wolfe gap is asymptotically 0, rather than a neighborhood whose size depends on ε. This is because all parameters in our algorithm are independent of ε, while in SFW, the stepsize and the number of iterations are functions of ε.*

Remark 2.2. *$\theta = \frac{3}{4}$ is the best trade-off. If β_t is too large, the noise of quantization and the variance of stochastic gradient will cause bad consensus and then affect the convergence. If β_t is too small, the stepsize should also be small. Otherwise the averaged gradient cannot catch up with the changing of x, which will cause slow convergence. This trade-off is the challenge and intuition to define our gradient estimator as Eq. (2.6).*

Remark 2.3. *From the proof in supplementary material we can see the theoretical framework does not only work for signSGD, but also all compressors that satisfy Assumption 5.*

2.6 Experimental Results

To validate the efficiency of our new DQSFW algorithm, we performed the experiments on two constrained machine learning applications: matrix completion and model compression.

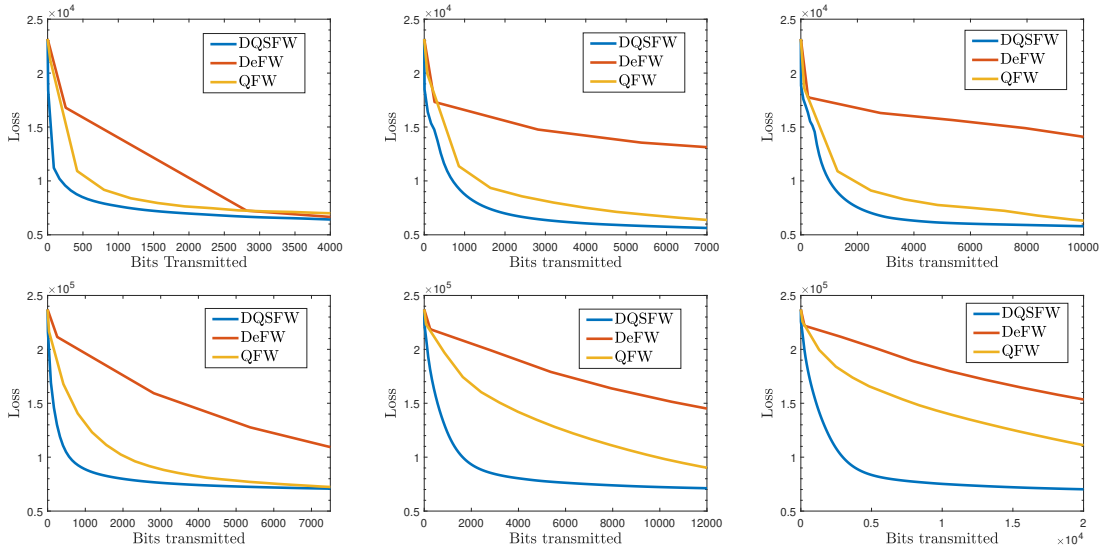


Figure 2.2: The experimental results of decentralized low-rank matrix completion for dataset MovieLens 100k and MovieLens 1m. Figure (a), (b) and (c) show the training loss with respect to bits transferred on MovieLens 100k with 20, 40 and 60 workers respectively. Figure (d), (e) and (f) show the training loss with respect to bits transferred on MovieLens 1m with 20, 40 and 60 workers respectively.

2.6.1 Decentralized Low-Rank Matrix Completion

Low-rank matrix completion is a model to solve a broad range of learning tasks, such as collaborative filtering [54] and multi-label learning [128]. The loss function of low-rank matrix completion problem has the following form:

$$\min_{X\in\mathbb{R}^{M\times N}} \sum_{(i,j)\in\Omega} \phi(X_{ij}-Y_{ij}), \quad \text{s.t. } \|X\|_* \leq C$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is the potential non-convex empirical loss function. Y is the target matrix and Ω is the set of observed entries. [115] also conducts this experiment with MSE loss function and robust Gaussian loss function. In our experiment, to verify that our algorithm works well for non-convex objective functions, we adopt the robust Gaussian loss function

$$\phi(z) = \frac{\sigma_0}{2}(1-\exp(-\frac{z^2}{\sigma_0})) \tag{2.11}$$

Name	User	Movie	Record
MovieLens 100k	943	1682	100000
MovieLens 1m	6040	3952	1000209

Table 2.2: Descriptions of MovieLens Datasets.

In our experiment, the parameter σ_0 is fixed as 2. We run our experiment on two benchmark datasets, MovieLens 100k and MovieLens 1m [35]. Both of two datasets are records of movie ratings from plenty of users, and are usually used to train recommendation systems. The descriptions of these two datasets are shown in Table 2.2. MovieLens 100k has 943 users, 1682 movies, and 100000 rating records. MovieLens 1M has 6040 users, 3952 movies, and 1000209 rating records. All ratings vary from 0 to 5. We scale them to the interval $[0,1]$. The rating records can be converted into matrix, where row represents user id and column represents movie id. Each record serves as an observation. As our purpose is to verify the performance of the optimization algorithm, we take all data for training.

For both datasets, we deploy our experiment on $M = 20, 40, 60$ MPI worker nodes, respectively, using mpi4py. Each node is an Intel Xeon E5-2660 machine within an infinity band network. We assign $1/M$ of the rating records to each worker. For MovieLens 100k, we set $C = 2000$ while for MovieLens 1m we set $C = 5000$.

In this task, the linear oracle can be obtained by singular value decomposition (SVD). Let the SVD of $v_t^{(i)}$ be $U \cdot S \cdot V^T$. Then the linear oracle $d = -C \cdot u \cdot v^T$ where u and v are the singular vectors corresponding to the largest singular value (also named leading vectors of SVD). In practice, we only need to compute the leading vectors, while in projected algorithms, we have to do the completed SVD.

We choose two other projection-free methods DeFW [115] and QFW [135] as baseline methods. For decentralized algorithms, we use a ring-based topology as the communication network because it is convenient to implement and achieves linear speedup in communication

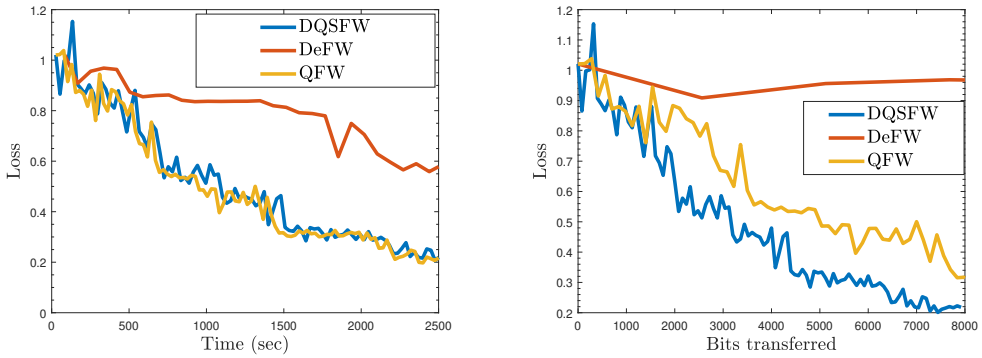


Figure 2.3: The experimental result of decentralized model compression for VGG11 neural network on dataset CIFAR 10. Figure (a) shows the training loss against time. Figure (b) shows the training loss against the number of bits transferred

[63]. For QFW, s of the s-partition encoding is set to 1. For all three algorithms, we set the stepsize $\eta_t = t^{-0.75}$. The results of low-rank matrix completion on MovieLens 100k and MovieLens 1m are shown in Figure 2.2. As many previous quantization works did (including QFW), we analyze the experimental results with respect to bits transferred, which means the number of bits sent or received on the busiest node. For decentralized algorithms, it can be any node, and for centralized algorithm, it is the master node. The number is divided by the size of x as it is always proportional to the size of x.

From the experimental results we can see that our DQSFW algorithm achieves the best performance on both datasets. Moreover, we can see that our algorithm becomes more competitive as the number of workers increases, which verifies the scalability of our method. With more workers, more sample gradients can be computed, but the number of gradients computed by DeFW stays the same.

2.6.2 Model Compression

Deep neural networks (DNNs) have achieved remarkable performance in many fields in recent years. But one of its shortcomings is the high cost of a large number of parameters. Thus, many attempts have been made to reduce the number of parameters in DNNs, such

as dropout and pruning. Among these methods, one solution is to add constraint to the parameters to make them more sparse compulsively [9, 68]. In [9], one popular method was proposed via adding the spectral norm constraint as follows:

$$\|W_l\|_2 \leq \tau \tag{2.12}$$

for each layer l. Because the model compression is attracting increasing attentions in both machine learning research and applications, in this experiment, we solve the model compression problem using the decentralized learning setting, and validate the performance of different decentralized quantized algorithms on this task. Here the linear oracle is $d = -\tau \cdot U \cdot V^T$, where $U \cdot S \cdot V^T$ is the SVD of $v_t^{(i)}$. This result can be achieved easily by the fact that trace norm and spectral norm are dual norms.

In our experiment, we run this task to compress the VGG11 network on the CIFAR 10 dataset, which has 50000 training samples and 10 labels, with constraint (2.12). We perform this task in decentralized settings where data are distributed on different nodes to verify our algorithms. Following [9], we use the cross-entropy loss function as a criterion and set $\tau = 0.8$. The experiment is implemented on 8 GTX1080 GPUs by PyTorch. Each GPU is treated as a single worker. The communication is based on NVIDIA NCCL.

We consider DeFW and QFW as baseline methods and ring-based topology as communication network. For DeFW and our DQSFW, the decentralized system is a uniform weight ring network. For QFW, s is set to 1. For all three algorithms, the stepsize is chosen as $\eta_t = \frac{1}{2}t^{-0.75}$. Due to the limitation of CUDA memory, we cannot compute the full gradient for DeFW. We calculate $1/5$ of the full gradient instead. This issue also indicates the limitation of DeFW algorithm.

The experimental results are visualized in Figure 2.3. To validate the efficiency of our algorithm, we compare the loss with respect to the bits transmitted. Similarly to the

matrix completion experiment, the number of bits transferred is divided by the size of the parameter. For decentralized algorithms, the number is counted on any node, while for a centralized algorithm, it is counted on the master node. According to the results, we can see that DeFW is almost infeasible for this task. In terms of the running time, QFW and DQSFW have similar performance. From the view of bits transferred, our DQSFW has the best performance among the three algorithms, which verifies the superior performance of our new algorithm.

2.7 Conclusion

In this chapter, we proposed a new Decentralized Quantized Stochastic Frank-Wolfe (DQSFW) algorithm to solve the non-convex constrained optimization problem. We revealed a potential divergence problem that is likely to occur in general decentralized training, not just for Frank-Wolfe-type methods, and also provided a solution by achieving consensus on the gradient. We derived a new theoretical analysis to prove that our algorithm can achieve the same gradient complexity $O(\varepsilon^{-4})$ as the Stochastic Frank-Wolfe (SFW) method with much lower communication cost, and the Frank-Wolfe gap is asymptotic to zero. The experimental results on two machine learning applications, matrix completion and deep neural network compression, validate the superior performance of our new algorithm.

Chapter 3: Decentralized Minimax Optimization

3.1 Introduction

In the past few decades, many works have studied the minimax optimization problem across various research fields, leading to the development of numerous methods. The most intuitive solution is the Gradient Descent Ascent (GDA) algorithm [24, 86] with equal stepsizes $\eta_x = \eta_y$. Asymptotic and nonasymptotic convergence analysis has been provided when f is convex in x and concave in y. Recently, many deterministic and stochastic gradient algorithms for nonconvex-strongly-concave and nonconvex-concave problems were proposed. Some algorithms improve the performance of the vanilla GDA method by adopting different steps in x and y, such as [37, 66], where the stepsize of y is typically larger than the stepsize of x. Some algorithms update x and y at different frequencies, such as [46, 77, 89]. These kind of algorithms usually involve a nested loop structure that updates y more frequently than x to make $f(x,y)$ close to the primal function $\Phi(x)$, which is defined by

$$\Phi(x) = \max_{y \in \mathscr{Y}} f(x,y). \tag{3.1}$$

As more large-scale machine learning problems arise, distributed training has become a popular and crucial framework due to its ability and efficiency in handling large data sets. It is desired to generalize minimax optimization to distributed training to solve large-scale minimax problems. In distributed optimization, the original centralized optimization suffers

from a bottleneck communication problem, i.e., the communication traffic on the busiest central node, especially when the network is large [63, 133]. To tackle this communication issue, decentralized optimization was proposed and has emerged as a promising technique. It is a distributed machine learning training paradigm that does not rely on a centralized network topology. Different worker nodes collaboratively utilize their own local data to implement large-scale training tasks, and at each iteration, they only have to communicate with their neighbors. Decentralized algorithms have been shown to enhance communication efficiency by avoiding the communication overhead problem. Decentralized methods are also advantageous when the network suffers from communication restrictions or has low bandwidth between some nodes and the central node. Moreover, it is an essential method in situations where data are geographically distributed and centralized data processing is not available, or there are concerns about preserving data privacy [130].

Recently many works were proposed to improve the performance of decentralized training. D-PSGD [63] theoretically justifies the potential advantage of the decentralized algorithm. D^2 [106] improves the convergence rate to outperform D-PSGD by eliminating the influence of data variance among different workers. D-SPIDER-SFO [90] incorporates D^2 and SPIDER [26, 117], which is a kind of variance reduction technique [48], to further reduce gradient complexity. DQSFW [119] studies decentralized constrained problem with Frank-Wolfe method. GT-HSGD [123] extends STORM to a decentralized setting, which is a variance-reduced approach that does not fetch a mega batch periodically. However, decentralized minimax optimization is still very limited, and existing methods suffer from a very high gradient complexity [70, 111]. Thus, we are motivated to design an accelerated decentralized algorithm for minimax problems.

In this chapter, thus, we propose a faster Decentralized Minimax Hybrid Stochastic Gradient Descent (DM-HSGD) algorithm to solve the following decentralized stochastic

minimax optimization problem:

$$\min_{x\in\mathbb{R}^{d_1}} \max_{y\in\mathscr{Y}} f(x,y) = \frac{1}{n}\sum_{i=1}^{n} f_i(x,y), \quad f_i(x,y) := \mathbb{E}_{\xi^{(i)}\sim D_i} F_i(x,y;\xi^{(i)}) \tag{3.2}$$

where n is the number of worker nodes, $\mathscr{Y}$ is a convex set. Here the local component objective function $F_i(x,y;\xi^{(i)})$ is L-smooth, nonconvex in x, and strongly-concave in y. D_i is the data distribution on the i-th node. In this chapter, the data distribution can be non-identical. The random variable $\xi^{(i)}$ is an index sampled from the local data. We summarize our contributions as follows.

(1) In this chapter, we propose a new accelerated decentralized stochastic first-order algorithm, named DM-HSGD, to solve decentralized nonconvex-strongly-concave minimax optimization problems. Our algorithm is the first stochastic gradient algorithm to solve general decentralized minimax problems on non-identically distributed data with theoretical guarantees. Moreover, our algorithm does not require a large batch size or nested loops, making it more practical and efficient to implement.

(2) We provide a completed proof to guarantee the convergence of our algorithm to solve decentralized stochastic minimax optimization. Under nonconvex-strongly-concave condition, our algorithm obtains SFO complexity of $O(\kappa^3\varepsilon^{-3})$ to search an ε-stationary point of function $\Phi(x) = \max_{y\in\mathscr{Y}} f(x,y)$. This result is faster than the complexity of previous decentralized minimax algorithms [70, 111]. Moreover, we also prove that our method achieves linear speedup as the number of workers n increases, which verifies its ability to solve large-scale problems.

The rest of this chapter is organized as follows. In Section 2, we introduce related works. In Section 3, we present our new DM-HSGD algorithm. In Section 4, we show the main theorems of convergence and complexity analysis. In Section 5, we discuss our experimental

Table 3.1: Comparison of Related Algorithms for Minimax Optimization

Name	SFO	Decentralized	Stochastic	Implementation	Reference
SGDA	$O(\kappa^3\varepsilon^{-4})$	$\times$	$\surd$	single-loop	[66]
SGDmax	$O(\kappa^3\varepsilon^{-4}\log(\frac{1}{\varepsilon}))$	$\times$	$\surd$	double-loop	[66]
SREDA	$O(\kappa^3\varepsilon^{-3})$	$\times$	$\surd$	double-loop	[77]
Acc-MDA	$O(\kappa^3\varepsilon^{-3})$	$\times$	$\surd$	single-loop	[39]
DPOSG	$O(\varepsilon^{-12})$	$\surd$ (iid)	$\surd$	single-loop	[70]
GT/DA	$O(N\varepsilon^{-2}\log(\frac{1}{\varepsilon}))$	$\surd$ (non-iid)	$\times$	double-loop	[111]
DM-HSGD	$O(\kappa^3\varepsilon^{-3})$	$\surd$ (non-iid)	$\surd$	single-loop	Ours

results, and Section 6 concludes this chapter.

3.2 Related Works

3.2.1 Centralized Minimax Optimization

In recent years, many algorithms for solving minimax optimization were proposed, and the majority of them were studied under the nonconvex-strongly-concave condition. SGDmax [46] is a double loop algorithm that achieves the SFO complexity of $O(\kappa^3\varepsilon^{-4}log(1/\varepsilon))$ where $\kappa = L/\mu$ is the condition number. Multistep GDA (MGDA) [89] is a double loop algorithm and HiBSA [75] is a single loop algorithm. Both MGDA and HiBSA are deterministic and hence can only solve finite-sum problems. Both of them achieve SFO complexity of $O(\kappa^4 N\varepsilon^{-2})$. Proximal Dual Implicit Accelerated Gradient (ProxDIAG) is a deterministic triple loop algorithm whose SFO complexity for the finite-sum problem is $O(\kappa^{1/2}N\varepsilon^{-2})$.

SGDA [66], Stochastic Recursive gradiEnt Descent Ascent (SREDA) [77], and Hybrid Variance-Reduced SGD [110] are more related to our work. SGDA is a single loop algorithm to solve nonconvex-strongly-concave and nonconvex-concave minimax problems. For nonconvex-strongly-concave problem, it requires $O(\kappa^3\varepsilon^{-4})$ SFO complexity to find an ε-stationary point of $\Phi(x)$. In this chapter, we will prove that our method achieves a better SFO complexity.

SREDA [77] is a double loop algorithm that achieves $O(\kappa^3\varepsilon^{-3})$ SFO complexity. It accelerates SGDA by using SPIDER, which is a variance reduction technique and uses the newest gradient information [26, 87]. SREDA also involves a separated initialization algorithm called PiSARAH [88] to ensure the convergence. More recently, [42] proposed an efficient mirror descent ascent algorithm for nonconvex-strongly-concave minimax optimization with nonsmooth regularization based on Bregman distance and variance reduced technique of SPIDER. In this chapter, we use another variance-reduced technique, named STORM or hybrid stochastic gradient descent [18], to accelerate the algorithm. We will discuss the challenges of using SPIDER on decentralized settings in Section 3.3. Unlike SREDA, our method only requires a large batch in the first iteration. Except for the first iteration, we can use either a single sample or a mini-batch to calculate the stochastic gradient. However, SREDA loads a mega-batch with size $O(\varepsilon^{-2})$ periodically (every q iterations) and needs $O(\varepsilon^{-1})$ gradient oracles at each iteration, which is not practical for large-scale problems. In addition, the maximizer in SREDA is a nested loop to update the variable y and if we count the loop of SPIDER then SREDA is actually a triple algorithm. On the contrary, there is no nested loop in our DM-HSGD, which makes our method more efficient and convenient to implement. Moreover, unlike SREDA, our method does not require a separate initialization algorithm to calculate a precise initial value for y.

Hybrid Variance-Reduced SGD algorithm also takes advantage of hybrid stochastic gradient descent to accelerate minimax optimization. For example, [39, 110] applied the Hybrid Variance-Reduced SGD to minimax problems. More recently, [33, 41] proposed some efficient adaptive gradient descent ascent methods for nonconvex-strongly-concave minimax optimization based on momentum techniques including Hybrid Variance-Reduced SGD.

3.2.2 Decentralized Minimax Optimization

At decentralized setting, most minimax algorithms were proposed for convex-concave problem [53, 80]. In [72] a nonconvex-nonconcave algorithm DPPSP was proposed. However, it is not gradient-based and the closed-form solution to the subproblem is not ensured in our problem. Hence we will not discuss it in this chapter. Decentralized Parallel Optimistic Stochastic Gradient (DPOSG) [70] is the first algorithm applicable to a general decentralized minimax problem with theoretical guarantees. It is a single loop minimax algorithm that generalizes Optimistic Stochastic Gradient (OSG) [17] to decentralized training. However, DPOSG has some obvious drawbacks. The first one is that the gradient complexity $O(\varepsilon^{-12})$ is too high and we are motivated to design a faster algorithm. The second one is that DPOSG only works in the case where the data distribution is identical. When the data distribution is non-identical, the Lemma 3 in [70] is not satisfied. Actually, the assumption of identical data distribution is not satisfied at most decentralized training tasks. Thus, in this chapter, we do not use this assumption.

More recently, [111] studied decentralized nonconvex-strongly-concave minimax problems and proposed a double-loop deterministic Gradient Tracking/Descent-Ascent algorithm which extends the vanilla GDA to decentralized setting and combines it with gradient tracking. It achieves a gradient complexity of $O(\varepsilon^{-2)}$. However, in large-scale machine learning tasks such as deep neural networks, the full gradient is generally unavailable and the application of deterministic algorithms is very restricted. If we convert Gradient Tracking/Descent-Ascent to stochastic gradient version, the SFO complexity should be at least $O(\varepsilon^{-4})$, which is the same result as SGD in nonconvex optimization. Under the same conditions, our new algorithm achieves a better SFO complexity of $O(\varepsilon^{-3})$.

[76] studied decentralized reinforcement learning problem based on distributed constrained Markov decision process model and proposed a decentralized policy gradient

optimization method named Safe Dec-PG, which achieves SFO complexity of $O(\varepsilon^{-4})$. However, the problem studied in [76] has a special form that is linear in y. In this chapter, we focus on general minimax problem. [8] is a simultaneous work of our work that studies a more general decentralized variational inequality problem with higher complexity. We summarize the comparison of related algorithms for general minimax optimization in Table 3.1. For decentralized algorithms DPOSG, GT/DA, and DM-HSGD, we also discuss whether they can converge on non-identical distributed data.

3.3 Proposed New Algorithm

3.3.1 Preliminaries

Before we propose our algorithms, we will introduce the notation used in this chapter and some important concepts. We use lower case $x_t^{(i)}$ and $y_t^{(i)}$ to represent the column vector parameters on the i-th worker node. We use upper case X_t and Y_t to represent the n-column matrix formed by $x_t^{(i)}$ and $y_t^{(i)}$ respectively, which means $X_t = [x_t^{(1)}, x_t^{(2)}, \ldots, x_t^{(n)}]$ and $Y_t = [y_t^{(1)}, y_t^{(2)}, \ldots, y_t^{(n)}]$. Column vectors $u_t^{(i)}$, $v_t^{(i)}$, $g_t^{(i)}$ and $h_t^{(i)}$ are gradient estimators used in our algorithms. The upper case U_t, V_t, G_t and H_t are matrices of which the i-th column is $u_t^{(i)}$, $v_t^{(i)}$, $g_t^{(i)}$ and $h_t^{(i)}$ respectively. Lower case with a bar represents the mean vector. The upper case with a bar represents the matrix in which each column is the mean vector. For example, $\bar{x}_t = \frac{1}{n}\sum_{i=1}^n x_t^{(i)}$ and $\bar{X}_t = [\bar{x}_t, \bar{x}_t, \ldots, \bar{x}_t]$. We define the optimal maximum value of y as:

$$y^*(\cdot) = \arg\max_{y \in \mathscr{Y}} f(\cdot, y), \quad \hat{y}_t = \arg\max_{y \in \mathscr{Y}} f(\bar{x}_t, y) \tag{3.3}$$

Note that when f is strongly-concave in y, $\hat{y}_t$ is unique. We also define:

$$\delta_t = \|\hat{y}_t - \bar{y}_t\|^2 \tag{3.4}$$

Bold number $\mathbf{0}$ and $\mathbf{1}$ are $n \times 1$ column vectors that each entry is 0 and 1, respectively. For matrices, we use $\|\cdot\|_F$ to denote Frobenius norm and $\|\cdot\|_2$ to denote spectral norm. We use ∇_x and ∇_y to denote the partial derivative with respect to x and y.

The mixing matrix W represents the weights of averaging among the topology of the communication network. It is doubly stochastic which satisfies:

$$W\mathbf{1} = W^T\mathbf{1} = \mathbf{1} \tag{3.5}$$

We should notice that here matrix W is not assumed to be symmetric so that the communication network is not restricted to undirected graph.

Algorithm 2 DM-HSGD

Input: mixing matrix W, initial value $x_0^{(i)} = x_0$, $y_0^{(i)} = y_0$, $v_{-1}^{(i)} = g_{-1}^{(i)} = \mathbf{0}$, $u_{-1}^{(i)} = h_{-1}^{(i)} = \mathbf{0}$
Parameter: stepsize η_x, η_y, weight β_x, β_y, batch size b_0, iteration T
Output: $\bar{x}_\zeta$, where ζ is chosen randomly from $\{1, 2, \cdots, T\}$

On i-th node:
for $t = 0, 1, \ldots, T-1$ **do**
  if $t = 0$ **then**
    $g_t^{(i)} = \nabla_x F_i(x_t^{(i)}, y_t^{(i)}; \xi_{x,t}^{(i)}), \quad |\xi_{x,t}^{(i)}| = b_0$
    $h_t^{(i)} = \nabla_y F_i(x_t^{(i)}, y_t^{(i)}; \xi_{y,t}^{(i)}), \quad |\xi_{y,t}^{(i)}| = b_0$
  else
    $g_t^{(i)} = \nabla_x F_i(x_t^{(i)}, y_t^{(i)}; \xi_t^{(i)}) + (1-\beta_x)(g_{t-1}^{(i)} - \nabla_x F_i(x_{t-1}^{(i)}, y_{t-1}^{(i)}; \xi_t^{(i)}))$
    $h_t^{(i)} = \nabla_y F_i(x_t^{(i)}, y_t^{(i)}; \xi_t^{(i)}) + (1-\beta_y)(h_{t-1}^{(i)} - \nabla_y F_i(x_{t-1}^{(i)}, y_{t-1}^{(i)}; \xi_t^{(i)}))$
  end if
  Communicate with neighbors and update gradient estimator as follows
  $v_t^{(i)} = \sum_{j=1}^n w_{ij}(v_{t-1}^{(j)} + g_t^{(j)} - g_{t-1}^{(j)})$
  $u_t^{(i)} = \sum_{j=1}^n w_{ij}(u_{t-1}^{(j)} + h_t^{(j)} - h_{t-1}^{(j)})$
  Communicate with neighbors and update model parameter as follows
  $x_{t+1}^{(i)} = \sum_{j=1}^n w_{ij}(x_t^{(j)} - \eta_x v_t^{(j)})$
  $y_{t+\frac{1}{2}}^{(i)} = \sum_{j=1}^n w_{ij}(y_t^{(j)} + \eta_y u_t^{(j)}), \quad y_{t+1}^{(i)} = P_{\mathscr{Y}}(y_{t+\frac{1}{2}}^{(i)})$
end for

3.3.2 Decentralized Minimax Hybrid Stochastic Gradient Descent

In this subsection, we introduce our new Decentralized Minimax Hybrid Stochastic Gradient Descent (DM-HSGD) algorithm. Our algorithm is a single loop minimax algorithm (summarized in Algorithm 2) which does not contain a nested loop structure.

The initial points of different nodes are the same, *i.e.* $x_0^{(i)} = x_0$ and $y_0^{(i)}$. $g_t^{(i)}$ and $h_t^{(i)}$ are the gradient estimators with respect to x and y on i-th node. $g_t^{(i)}$ and $h_t^{(i)}$ are computed in the same way as STORM [18]. When $t = 0$, we load a large batch with size b_0 to calculate the stochastic gradient (lines 4 and 5 in Algorithm 2). When $t > 0$, we can use either a single sample or a mini-batch to calculate the gradient (lines 7 and 8 in Algorithm 2). $g_t^{(i)}$ can also be written as

$$g_t^{(i)} = \beta_x \nabla_x F_i(x_t^{(i)}, y_t^{(i)}; \xi_t^{(i)}) + (1-\beta_x)\left(g_{t-1}^{(i)} - \nabla_x F_i(x_{t-1}^{(i)}, y_{t-1}^{(i)}; \xi_t^{(i)}) + \nabla_x F_i(x_t^{(i)}, y_t^{(i)}; \xi_t^{(i)})\right) \tag{3.6}$$

which is a linear combination of the gradient estimators of stochastic gradient descent (the first part) and SPIDER (the second part). As we have mentioned, SPIDER is a variance-reduced method that utilizes the latest gradient information. Thus, estimator Eq. (3.6) is also called hybrid stochastic gradient descent. It is the same with $h_t^{(i)}$. Then each worker communicates with their neighbors to compute gradient estimator $v_t^{(i)}$ and $u_t^{(i)}$. Here we use gradient tracking [21, 126] to reduce the consensus error (lines 11 and 12 in Algorithm 2). We will discuss why gradient tracking is necessary in our method in the next subsection. After we obtain $u_t^{(i)}$ and $v_t^{(i)}$, each worker communicates with their neighbors again and updates the model parameters x and y. Here $P_{\mathscr{Y}}(\cdot)$ represents the projection onto convex set $\mathscr{Y}$. In the theoretical analysis, we define $Y_{-\frac{1}{2}} = Y_0$.

3.3.3 Discussions on STORM and Gradient Tracking

In this subsection, we discuss the intuition behind our algorithm and explain why we chose STORM and gradient tracking rather than generalizing SREDA for the decentralized setting. The first reason is that SREDA requires a large or full batch periodically, which is expensive and often unavailable. Additionally, SREDA involves too many nested loops, making it inefficient and inconvenient. From a theoretical analysis perspective, normalization or projection is likely to cause divergence in decentralized training on non-identical data distributions, as indicated by Example 3.1. Therefore, in the context of this chapter, SPIDER will probably not converge to a stationary point. Moreover, SREDA adopts a smaller stepsize at the beginning and a larger stepsize at the end when $\|v_t\|$ becomes small enough. However, when the data distribution is non-identical, $\|v_t\|$ may not tend to 0, and the stepsize of SREDA will likely always remain small. In contrast, STORM can avoid these issues, so we use STORM to accelerate the decentralized minimax algorithm.

In the standard decentralized framework D-PSGD [63], the consensus error satisfies $\|X_t - \bar{X}_t\|_F \leq O(\varepsilon)$ when the stepsize η is $O(\varepsilon)$ and t is large enough. The following Example 3.2 is a simple example to show that this bound is tight and there are cases where consensus error $\|X_t - \bar{X}_t\|_F$ is exactly $\Theta(\eta)$ when the data distribution is non-identical. However, according to the analysis of STORM [18] without gradient tracking, the error term $e_t = \bar{g}_t - \nabla_x f(\bar{x}_t, \bar{y}_t)$ between the averaged update direction and the correct direction is supposed to satisfy:

$$\|e_t\|^2 \leq (1-\beta_x)\|e_{t-1}\|^2 + O(\eta_x^4)\,. \tag{3.7}$$

Nevertheless, the consensus error $\|X_t - \bar{X}_t\|_F^2$ is only $O(\eta_x^2)$ and cannot be as small as $O(\eta_x^4)$ if there is no gradient tracking. Therefore, to inherit the analysis framework of STORM, the gradient tracking in our algorithm is essential.

Example 3.1. *Assume $f(x) = f_1(x) + f_2(x)$, where $x = (a,b) \in R^2$. $f_1(x) = a$ and $f_2(x) = \sqrt{3}b$ are defined on two different nodes. Let W be the uniform weighted mixing matrix. We can compute $v_1 = (1,0)$ and $v_2 = (0,\sqrt{3})$. The ideal averaged gradient direction is $v^* = (1/2, \sqrt{3}/2)$. However, if we do normalization before making consensus, the obtained gradient estimator is $v = (1/2, 1/2)$, which is deviated from v^*.*

Example 3.2. *Suppose there are two sequences $\{p_t\}$ and $\{q_t\}$ defined on two different nodes with $p_0 = q_0$. They are updated by $p_{t+\frac{1}{2}} = p_t - \eta a$ and $q_{t+\frac{1}{2}} = q_t - \eta b$ in each iteration, respectively, where a and b are fixed gradient directions. As data distribution is non-identical, we have $a \neq b$. Assume the mixing matrix is*

$$W = \begin{bmatrix} 2/3 & 1/3 \\ 1/3 & 2/3 \end{bmatrix}$$

Then we have

$$p_{t+1} - q_{t+1} = \frac{1}{3}(p_t - q_t) - \frac{\eta}{3}(a-b) = \frac{1}{3^{t+1}}(p_0 - q_0) - \eta\left(\sum_{s=1}^{t+1}\frac{1}{3^s}\right)(a-b) = \frac{\eta}{2}\left(1 - \frac{1}{3^{t+1}}\right)(b-a) \tag{3.8}$$

Therefore, $\lim_{t\to\infty}\|p_t - q_t\| = \frac{\eta}{2}\|a-b\|$.

3.4 Convergence Analysis

In this section, we will show the main theorems of our convergence analysis. The theoretical results show that the SFO complexity of our algorithm is $O(\kappa^3\varepsilon^{-3})$, which is the same as the best result in centralized minimax problem [77]. First we will introduce the following assumptions.

Assumption 3.1. *(Lipschitz Gradient). Each component function $F_i(x,y;\xi)$ is L-smooth, which means there exists a constant L such that for any (x,y) and (x',y'), we have*

$$\|\nabla F_i(x,y;\xi)-\nabla F_i(x',y';\xi)\|^2 \leq L^2(\|x-x'\|^2+\|y-y'\|^2)$$

Assumption 3.2. *(Bounded Variance). The gradient of each component function $F_i(x,y;\xi)$ is an unbiased estimator of $\nabla f_i(x,y)$ and has bounded variance, i.e.,*

$$\mathbb{E}\|\nabla F_i(x,y;\xi)-\nabla f_i(x,y)\|^2 \leq \sigma < +\infty$$

Assumption 3.3. *(Lower Bound). The function $\Phi(\cdot)$ is lower bounded, i.e.,* $\inf_x \Phi(x) = \Phi^* > -\infty$.

Assumption 3.4. *(Spectral Gap). The doubly stochastic matrix W satisfies* $\|W-\frac{\mathbf{1}\mathbf{1}^T}{n}\|_2 = \lambda \in [0,1)$.

Assumption 3.5. *(Strongly Concave). The function $f_i(x,y)$ is μ-strongly-concave in y. That is, there exists a constant $\mu > 0$, for any x, y and y', we have*

$$f_i(x,y) \leq f(x,y') + \langle \nabla_y f(x,y'), y-y' \rangle - \frac{\mu}{2}\|y-y'\|^2$$

These are very common and mild assumptions that are frequently assumed in previous works. Assumptions 3.1, 3.2 and 3.3 are also used in minimax methods [77] and [66]. Assumption 3.4 is used in [123]. Typically, the spectral gap assumption is stated as W is symmetric and $|\lambda_2|<1$, $|\lambda_n|<1$ where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ are the eigenvalues of W [51, 63, 133]. Our Assumption 3.4 is automatically satisfied if the typical spectral gap assumption holds (see Lemma 16 in [51]). Assumption 3.5 is the definition of strong concavity.

In nonconvex-strongly-concave problems, we use ε-stationary point of $\Phi(x)$, *i.e.* $\|\nabla\Phi(x)\| \leq \varepsilon$ as the convergence criterion. From Lemma 4.3 in [66], we know that $\Phi(x)$ is differentiable and $(L+\kappa L)$-smooth and $y^*(\cdot)$ is κ-Lipschitz, which means $\|y^*(x_1)-y^*(x_2)\| \leq \kappa\|x_1-x_2\|$ for any $x_1, x_2 \in \mathbb{R}^{d_1}$. Furthermore, we have:

$$\nabla\Phi(\bar{x}_t) = \nabla_x f(\bar{x}_t, \hat{y}_t) + \nabla_y f(\bar{x}_t, \hat{y}_t) \cdot \partial y^*(\bar{x}_t) = \nabla_x f(\bar{x}_t, \hat{y}_t) \tag{3.9}$$

since $\nabla_y f(\bar{x}_t, \hat{y}_t) = 0$. This criterion is broadly used in the analysis of nonconvex-strongly-concave minimax optimization [66, 108]. Now we will provide the main theorems of our convergence analysis. Completed proof can be found in the Supplementary Material.

Theorem 3.1. *Let Assumptions 3.1 to 3.5 hold. When parameters* $\beta_x = \frac{\varepsilon \min\{1, n\varepsilon\}}{20}$, $\beta_y = \frac{\varepsilon \min\{1, n\varepsilon\}}{500\kappa^2}$, $\eta_x = \frac{(1-\lambda)^2 \min\{1, n\varepsilon\}}{2000\kappa^3 L}$, $\eta_y = \frac{(1-\lambda)^2 \min\{1, n\varepsilon\}}{500\kappa L}$, $b_0 = \frac{400}{\min\{1, n\varepsilon\}}$, $T = \frac{4000\kappa^3 \varepsilon^{-2}}{(1-\lambda)^2 \min\{1, n\varepsilon\}}$, *our Algorithm 2 satisfies*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla\Phi(\bar{x}_t)\|^2 \leq L(\Phi(x_0) - \Phi^*)\varepsilon^2 + \sigma^2\varepsilon^2 + L^2\delta_0\varepsilon^2 + \frac{\varepsilon^2}{n}\sum_{i=1}^{n} \mathbb{E}\|\nabla_x f_i(x_0, y_0)\|^2 + \frac{\varepsilon^2}{n}\sum_{i=1}^{n} \mathbb{E}\|\nabla_y f_i(x_0, y_0)\|^2 \tag{3.10}$$

Corollary 3.1. *When the parameters are defined as Theorem 3.1, we can see* $\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla\Phi(\bar{x}_t)\|^2 \leq O(\varepsilon^2)$. *Therefore, if* $n \leq O(\varepsilon^{-1})$, *the SFO complexity of Algorithm 2 is* $O(\kappa^3\varepsilon^{-3})$. *If* $n > O(\varepsilon^{-1})$, *the SFO complexity is* $O(\kappa^3 n\varepsilon^{-2})$. *Besides, from the proof of Theorem 3.1 we can see error* $\|\bar{y}_t - y^*(\bar{x}_t)\|^2$ *is also bounded by the right side of Eq. (3.10).*

Theorem 3.1 is the theoretical result when T is determined by ε. If the number of iteration T is not fixed, we have the following conclusion.

Theorem 3.2. *Let Assumptions 3.1 to 3.5 hold. We set the parameters as* $T = \frac{4000\kappa^3 T_0}{(1-\lambda)^2}$, $\beta_x = \frac{n^{1/3}}{20T_0^{2/3}}$, $\beta_y = \frac{n^{1/3}}{500\kappa^2 T_0^{2/3}}$, $\eta_x = \frac{(1-\lambda)^2 n^{2/3}}{2000\kappa^3 T_0^{1/3} L}$, $\eta_y = \frac{(1-\lambda)^2 n^{2/3}}{500\kappa T_0^{1/3} L}$, $b_0 = \frac{T_0^{1/3}}{n^{2/3}}$, *where we suppose*

$T_0 \geq 10n^2$. *Then our algorithm satisfies*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\Phi(\bar{x}_t)\|^2 \leq \frac{L(\Phi(x_0)-\Phi^*)+\sigma^2+L^2\delta_0}{(nT_0)^{2/3}} + \frac{\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\|\nabla_x f_i(x_0,y_0)\|^2}{T_0} + \frac{\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\|\nabla_y f_i(x_0,y_0)\|^2}{T_0} \tag{3.11}$$

Corollary 3.2. *From Theorem 3.2, we know* $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\Phi(\bar{x}_t)\|^2 \leq O(\frac{1}{(nT_0)^{2/3}}) + O(\frac{1}{T_0})$ *when parameters are defined as above. As we suppose* $T_0 \geq O(n^2)$*, the dominating term in the convergence rate is* $O(\frac{1}{(nT_0)^{2/3}})$*, which indicates the linear speedup of our algorithm.*

3.5 Experiments

3.5.1 Robust Logistic Regression

We conduct the experiment of decentralized robust logistic regression[1] task as the first experiment, which was proposed in [131] and was also conducted in the related work [77]. Given dataset $\{(a_i,b_i)\}_{i=1}^n$, where $a_i \in \mathbb{R}^d$ is the feature and $b_i \in \{-1,1\}$ is the label, the robust logistic regression problem is formulated as follows:

$$\min_{x\in\mathbb{R}^d}\max_{y\in\Delta_n} f(x,y) = \sum_{i=1}^{n} y_i l_i(x) - V(y) + g(x) \tag{3.12}$$

where y_i is the i-th component of variable y. $l_i(x)$ is the logistic loss function which is defined by $l_i(x) = \log(1+\exp(-b_i a_i^T x))$. $V(y)$ is a divergence measure defined by $V(y) = \frac{1}{2}\lambda_1\|ny-\mathbf{1}\|^2$. Δ_n represents the simplex in $\mathbb{R}^n$, which means

$$\Delta_n = \{y \in \mathbb{R}^n | 0 \leq y_i \leq 1, \sum_{i=1}^{n} y_i = 1\} \tag{3.13}$$

[1] `https://github.com/WH-XIAN/DM-HSGD`

$g(x)$ is a nonconvex regularization with form $g(x) = \lambda_2 \sum_{i=1}^{d} \frac{\alpha x_i^2}{1+\alpha x_i^2}$. Following the experimental settings in [77, 131], we let $\lambda_1 = \frac{1}{n^2}$, $\lambda_2 = 0.001$ and $\alpha = 10$ in our experiment.

Table 3.2: Descriptions of datasets used in our experiment

Name	a9a	covtype	ijcnn1	phishing	rcv1	w8a
N	32561	581012	49990	11055	20242	49749
d	123	54	22	68	47236	300

We conduct our experiment on six real-world training datasets "a9a", "covtype", "ijcnn1", "phishing","rcv1" and "w8a", which can be downloaded from LIBSVM[2] repository. The description of datasets is listed in Table 3.2 where N is the number of samples and d is the number of features. We implement our code on an MPI cluster where each node is equipped with 12-core Intel Xeon E5-2620 v3 2.40 GHz processor.

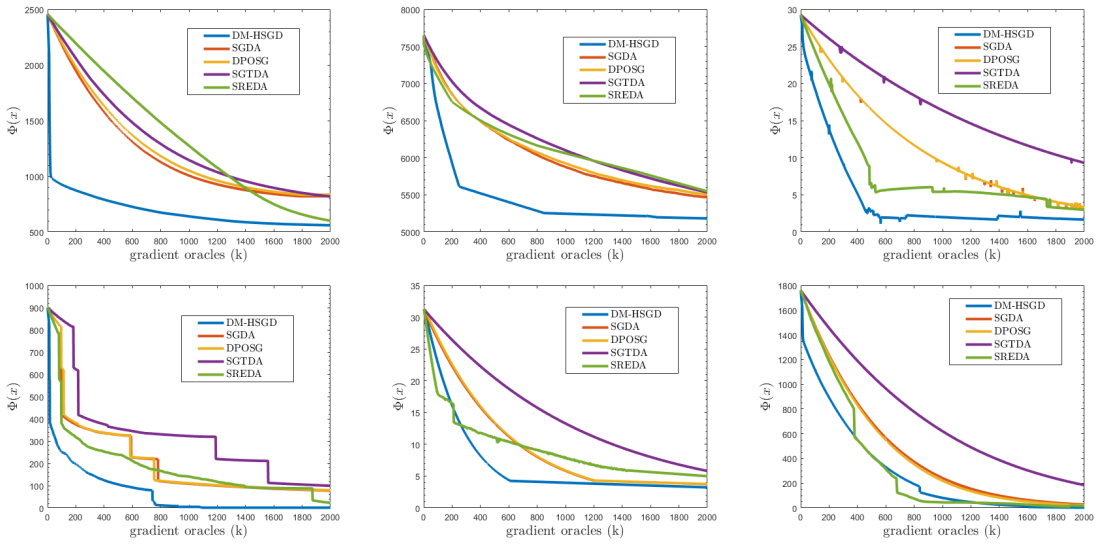


Figure 3.1: Results of our decentralized robust logistic regression task. Figure (a) to (f) show the value of $\Phi(x)$ with respect to the number of gradient oracles divided by 10^3. Figure (a), (b), (c), (d), (e) and (f) are experimental results on "a9a", "covtype", "ijcnn1", "phishing", "rcv1" and "w8a" respectively.

[2] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets`

We compare our DM-HSGD algorithm with baseline algorithms: SGDA [66], SREDA [77], DPOSG [70], and stochastic Gradient Tracking/Descent Ascent (SGTDA) [111]. We consider the algorithms for solving stochastic problems. We set the number of worker nodes to $n = 20$ and use the ring-based topology as the communication network. For each algorithm, we perform a grid search for learning rates η_x and η_y from $\{0.1, 0.01, 0.001, 0.0001\}$. The mini-batch size is set to 20. The number of iterations in the nested loop for double-loop algorithms is set to $K = 5$. For DM-HSGD, we set the batch size in the first iteration to $b_0 = 10000$. β_x and β_y are set to 0.01. For SREDA, we set $\varepsilon = 0.1$ in the factor $\frac{\varepsilon}{\|v_t\|}$, period $q = 50$ and large batch size $S_1 = 1000$. We compare the value of $\Phi(x)$ with respect to the number of gradient oracles among different algorithms, which can also be calculated by the projection onto simplex Δ_n. The experimental results are shown in Figure 3.1. From the experimental results in Figure 3.1, we can see our new DM-HSGD algorithm converges faster than other baseline algorithms, which verifies the performance of our method.

3.5.2 Policy Evaluation

Our second experiment is the decentralized policy evaluation (PE) task. PE is an important task in reinforcement learning, aiming to estimate the value function of a given policy. The most intuitive and frequently used method for PE is the temporal-difference (TD) method, which relies on the Bellman equation [20]. However, the traditional TD method, which is probably not a true gradient descent method as pointed out in [69] and [103], has been shown to be unstable in the case of off-policy sampling or nonlinear function approximation. [102] first proposed a method to optimize the objective function of the mean-squared projected Bellman error (MSPBE), and MSPBE is proven to achieve asymptotic convergence with arbitrary nonlinear smooth function approximation in [79]. In [114], the MSPBE objective function with nonlinear approximation is converted into a nonconvex-

strongly-concave minimax problem by Fenchel's duality. The problem can be formulated as:

$$\min_{\theta}\max_{w} L(\theta,w) = \frac{1}{nN_i}\sum_{i=1}^{n}\sum_{j=1}^{N_i} L_j^{(i)}(\theta,w),$$
$$L_j^{(i)}(\theta,w) = \langle w, [R_i(s_j,a_j)+\gamma V_\theta(s_{j+1})-V_\theta(s_j)]g_\theta(s_j)\rangle - \frac{1}{2}(w^T g_\theta(s_j))^2 \tag{3.14}$$

where s_j is a state and a_j is an action. R_i represents the reward and $\gamma \in (0,1)$ is the discount factor. V is a value function that maps the state space to a real number. θ is the parameter to estimate the value function. Function g_θ is the gradient of V_θ and parameter w is yield by Fenchel's duality.

Mountaincar [101] is a preliminary task in reinforcement learning. [114] and [116] ran offline PE task of this problem with primal-dual MSPBE, where the objective function is formulated as Eq. (3.14). Following the experimental settings in [114], we use Sarsa [101] to generate trajectories of transitions (s_i, a_i, s_{i+1}, r_i) with d features and $N = 5000$ samples on each worker node. We parameterize value function V_θ as a 2-layer neural network with H hidden neurons. We use Sigmoid function as activation and set discount factor to $\gamma = 0.95$. This experiment is run on an MPI cluster where each node is equipped with 12-core Intel Xeon E5-2620 v3 2.40 GHz processor.

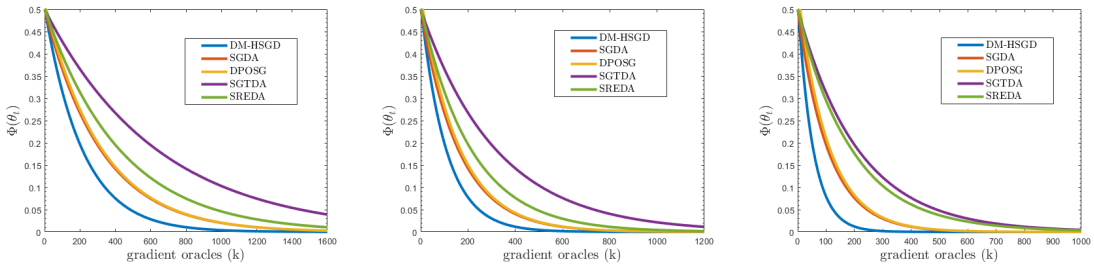


Figure 3.2: Results of our policy evaluation task. Figures (a), (b) and (c) show the value of $\Phi(\theta)$ with respect to the number of gradient oracles divided by 10^3. In Figures, (a) $d = 200$, $H = 50$; (b) $d = 300$, $H = 100$; (c) $d = 400$, $H = 200$.

We compare our DM-HSGD algorithm with baseline algorithms: SGDA [66], SREDA [77], DPOSG [70], and stochastic Gradient Tracking/Descent Ascent (SGTDA) [111]. We also consider algorithms for solving stochastic problems. We set the number of worker nodes to $n = 20$. We also use a ring-based topology with uniform weights as the communication network in this task. For each algorithm, we perform a grid search for learning rates η_x and η_y from $\{0.1, 0.01, 0.001, 0.0001\}$. The mini-batch size is set to 20. The number of iterations in the nested loop for double-loop algorithms is set to $K = 5$. For DM-HSGD, we set the batch size in the first iteration to $b_0 = 2500$. β_x and β_y are set to 0.01. For SREDA, we set $\varepsilon = 0.1$ in the factor $\frac{\varepsilon}{\|v_t\|}$, period $q = 50$ and large batch size $S_1 = 1000$. We compare the values of $\Phi(\theta)$ with respect to the number of gradient oracles among different algorithms, which can be calculated by quadratic optimization. The experimental results are shown in Figure 3.2.

Figure 3.2 (a), (b) and (c) show that our DM-HSGD algorithm achieves the fastest convergence regarding the number of gradient oracles. From our experimental results, we can also see that nested loop algorithms for minimax optimization usually consume more gradient complexities during the training process than single-loop algorithms.

3.6 Conclusion

In this chapter, we proposed a novel accelerated decentralized minimax algorithm, Decentralized Minimax Hybrid Stochastic Gradient Descent (DM-HSGD), to solve stochastic nonconvex-strongly-concave minimax optimization problems. We prove that our new method achieves SFO complexity of $O(\kappa^3\varepsilon^{-3})$, which outperforms existing results in decentralized minimax optimization and matches the state-of-the-art in centralized minimax optimization. Our method also achieves linear speedup with respect to the number of workers, demonstrating its ability to solve large-scale problems. We conducted experi-

ments on two machine learning tasks—decentralized robust logistic regression and policy evaluation—to validate the superior performance of our algorithm.

In future work, we will explore decentralized nonconvex-concave minimax optimization without the strong concavity to solve a broader range of problems, including loss functions that are linear in y. We will likely consider methods that add perturbations, such as Catalyst [132].

Chapter 4: Communication-Efficient Adaptive Gradient Algorithms

4.1 Introduction

Nowadays, as more and more data mining and machine learning applications take advantage of large-scale data, many learning models are trained in a distributed fashion across many worker nodes [60]. Specifically, the problem of these tasks can be formulated as:

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\xi_i \sim D_i} F_i(x; \xi_i), \tag{4.1}$$

where $f_i(x) = \mathbb{E}_{\xi_i \sim D_i} F_i(x; \xi_i)$ is the local objective function on the i-th node that is generally smooth and possibly nonconvex, and n is the number of worker nodes. Here D_i denotes the data distribution on the i-th node, and different $\{D_i\}_{i=1}^n$ are probably non-identical.

Although distributed training has shown a significant advantage, it still suffers from the communication bottleneck, especially when the network bandwidth is limited or the size of the model is large. To address this critical issue, many methods have been presented to reduce the cost of communication. Among these methods, one of the most popular ways is to compress the transmitted message in each communication round, such as gradient quantization [3, 97] and gradient sparsification [1, 4, 98], on which are the focus of this chapter. Other methods such as model compression [137] or decentralization [121] also alleviate the bottleneck issue.

Gradient quantization reduces the communication cost by lowering the float-point precision of gradients so that fewer bits will be transmitted. 1-bit Stochastic Gradient Descent (1-bit SGD) [97] is a classic and primitive gradient quantization work that uses 1-bit quantization and dramatically enhances communication efficiency. Quantized Stochastic Gradient Descent (QSGD) adopts stochastic randomized rounding to obtain an unbiased estimator after compression. SignSGD and its variant with momentum named Signum [6] only transmit the 1-bit gradient sign between the workers and the central node, which is convenient to implement.

Gradient sparsification is another widely-used strategy to decrease the communication cost which sparsifies the gradient instead of quantizing each element. The most popular way is to extract the top-k coordinates of local gradients and send them to the master node to estimate the overall mini-batch gradient. Some of these methods are also combined with other techniques, such as momentum correction and error-feedback.

Recently, more variants of gradient compression with theoretical guarantees have been proposed, such as SGD with Error-Feedback (EF-SGD) [49], Distributed SGD with Error-Feedback (dist-EF-SGD) [138] and SGD with Error Reset (CSER) [122]. In some recent works such as [122, 138], the aggregated gradient estimator is also compressed before being sent back to the workers. Some works also apply gradient compression to other optimizers such as the Frank-Wolfe algorithm [120].

In addition, to solve problem (4.1), we also need an efficient optimizer to search for the optimal solution. Among popular optimization methods, adaptive gradient algorithms [25, 109] have become one of the most important optimization algorithms to pursue higher efficiency or accuracy in a wide range of data mining and machine learning problems. In the family of adaptive gradient algorithms, Adam [50] is the most popular one that combines momentum and adaptive learning rate. Although it achieves great success in practice, several technical issues in the analysis were pointed out [92] and in some cases the algorithm could

diverge.

In [92], two variants of Adam, named AMSGrad and Adamnc, were proposed to fix the theoretical problems in the analysis of Adam. AMSGrad makes the quantity $\Gamma_{t+1} = (\frac{\sqrt{V_{t+1}}}{\alpha_{t+1}} - \frac{\sqrt{V_t}}{\alpha_t})$ positive to ensure the convergence, while Adamnc adopts an increasing parameter $\beta_{2,t} = 1 - \frac{1}{t}$.

Despite the success of gradient compression methods, it is hard to use them in distributed adaptive gradient method. So far, the application of gradient compression to adaptive gradient algorithm with theoretical guarantee is still limited. Quantized Adam [11] combines gradient quantization with Adamnc, which keeps track of the local momentum and variance terms on each worker node and uses quantization when averaging the parameter. Efficient-Adam [12] is similar to Quantized Adam where the gradient message sent back is also compressed. However, both Quantized Adam and Efficient-Adam are not proven to achieve linear speedup or convergence on non-iid data. APMSqueeze [105] and 1-bit Adam [104] are Adam-preconditioned momentum SGD algorithms with gradient compression. However, the variance term is fixed during the training process. Even though it is computed by Adam at the end of warm-up step, technically APMSqueeze and 1-bit Adam are not a true adaptive gradient method.

Therefore, it is difficult to apply gradient compression to adaptive gradient methods and maintain the excellent performance of distributed Adam-type algorithms. The challenge is that the original adaptive learning rate is adjustable based on global information such as the aggregated gradient. Although the compressed message is a good estimation of local gradient or momentum, the adaptive learning rate calculated by these inexact messages could be far away from the original one.

To address the challenging high communication cost limitation in distributed adaptive gradient methods, we propose a class of novel distributed Adam-type algorithms (called SketchedAMSGrad), based on the distributed version of the AMSGrad [92] algorithm and

the gradient sparsification technique named sketching [43, 61].

Our main contributions are summarized as follows.

(1) To efficiently address the communication bottleneck problem in distributed data mining, we propose a class of novel communication-efficient algorithms named SketchedAMSGrad with two averaging strategies: parameter averaging and gradient averaging. Our new methods can reduce the communication cost from $O(d)$ to $O(\log(d))$.

(2) We provide theoretical analysis based on mild assumptions to guarantee the convergence of our algorithms. Specifically, we prove that our SketchedAMSGrad algorithms have a convergence rate of $O(\frac{1}{\sqrt{nT}})$, which shows a linear speedup. Our theoretical analysis also allows the data distribution to be non-identical.

(3) To the best of our knowledge, our method is the first one to utilize the sketching technique to solve the communication bottleneck in distributed adaptive gradient methods. The experimental results on training various DNNs verify the performances of our algorithms, on both identical and non-identical distributed datasets.

4.2 Related Works

In the section, we review the related adaptive gradient algorithms with their compressed versions and introduce some preliminary background of sketching. The summary of properties of related methods is listed in Table 4.1. Top-k is considered as the compressor in the result of the convergence rate. The column 'Speedup' represents whether the algorithm achieves linear speedup. The column 'Adaptive' represents whether the algorithm adopts an adaptive gradient.

Table 4.1: Comparison of Related Algorithms with Compression

Name	Convergence rate	Speedup	Non-iid	Adaptive
Quantized-Adam [11]	$O(\frac{1}{\sqrt{T}})$	$\times$	$\times$	$\surd$
Efficient-Adam [12]	$O(\frac{1}{\sqrt{T}})$	$\times$	$\times$	$\surd$
APMSqueeze [105]	$O(\frac{1}{\sqrt{nT}} + \frac{1}{(k/d)^{2/3}T^{2/3}})$	$\surd$	$\surd$	$\times$
1-bit Adam [104]	$O(\frac{1}{\sqrt{nT}} + \frac{1}{(k/d)^{2/3}T^{2/3}})$	$\surd$	$\surd$	$\times$
SketchedAMSGrad (ours)	$O(\frac{1}{\sqrt{nT}} + \frac{1}{(k/d)^{2}T})$	$\surd$	$\surd$	$\surd$

4.2.1 Quantized-Adam and Efficient-Adam

Quantized-Adam [11] is proposed to combine quantization scheme with distributed Adam algorithm to reduce the communication cost. Specifically, on each worker, it owns a local momentum term $m_t^{(i)}$ and a local variance term $v_t^{(i)}$. These two terms are updated by the exponential moving averaging used in Adam-type algorithms. Gradient quantization is used to compress the term $m_t^{(i)}/\sqrt{v_t^{(i)}}$.

Efficient-Adam [12] is a similar work to Quantized Adam. The only difference is that Efficient-Adam compresses the updating term rather than the parameter. Both of these two algorithms are parameter averaging, since if there is no compression, they degenerate to an algorithm where each node is updated by Adam and then the model parameter is averaged. It is not mathematically equivalent to the typical distributed Adam algorithm where gradient averaging is used. Though in some cases parameter averaging is convenient to implement, it is likely to cause bad convergence or be detrimental to the model accuracy, especially when the optimizer relies on past local gradient. Actually, Quantized-Adam and Efficient-Adam fail to achieve linear speedup on non-iid data.

4.2.2 APMSqueeze and 1-bit Adam Algorithms

APMSqueeze [105] and 1-bit Adam [104] are communication-efficient Adam-preconditioned momentum SGD algorithms. Since the definitions of these two algorithms are similar and 1-bit Adam is the later work, in this chapter we will only discuss 1-bit Adam. In the warm-up stage, it calculates a variance term v_{T_w}. During the training process, v_{T_w} is fixed and serves as the exponential moving averages term v_t in regular Adam-type algorithms. However, since v_{T_w} is a fixed variable, 1-bit Adam is not technically an adaptive gradient method. In our method, the variance term v_t is dynamic and is computed by exponentially moving averaging. In addition, we do not need the warm-up stage with a separate communication-inefficient optimizer.

4.2.3 Sketching

Sketching [43] is a novel and promising gradient sparsicifation technique that compresses a gradient vector g into a sketch $S(g)$ of size $O(\log(d)\varepsilon^{-1})$ such that $S(g)$ can approximately recover all coordinates by $\hat{g}_i^2 = g_i^2 \pm \varepsilon\|g\|_2^2$. It originates from a data structure used in data streaming named Count Sketch [10] which is designed to find large coordinates in a vector g defined by a sequence of updates $\{(i_j, w_j)\}_{j=1}^n$.

In [43], sketching serves as a compressor that will approximately recover the true top-k coordinates of mini-batch gradient $\frac{1}{n}\sum_{i=1}^n g_t^{(i)}$ where n is the number of workers. In [96], the authors explicitly treat it as a compressor, and the sketching and unsketching operators are denoted by $\mathscr{S}$ and $\mathscr{U}$. For convenience, we also use these notation in this chapter. The sketching method reduces the communication cost to $O(\log(d))$, while gradient quantization only achieves a constant level reduction and the communication cost is still $O(d)$. The current best results for the quantization method achieve an approximate compression rate $32\times$ [7, 138]. Compared with the top-k method, one advantage of sketching is to recover

the true top-k coordinates, where the gradient estimator is $v_1 \approx Top_k(\frac{1}{n}\sum_{i=1}^n g_t^{(i)})$. Although applying the method in [138] can avoid the $O(n)$ return communication cost mentioned in [43], the gradient estimator $v_2 = Top_k(\frac{1}{n}\sum_{i=1}^n Top_k(g_t^{(i)}))$ is probably still far away from the true top-k coordinates. This issue can be reflected in the second dominant term in the convergence rate. In [138], the second dominant term is $O(\frac{1}{(k/d)^{4/3}T^{2/3}})$, which is claimed to be the price to pay for two-way compression and linear speedup. In 1-bit Adam the stepsize depends on the compression ratio and this term becomes $O(\frac{1}{(k/d)^{2/3}T^{2/3}})$ as we have mentioned. However, in Sketched-SGD and our algorithms, the corresponding term is $O(\frac{1}{(k/d)^2 T})$, which is smaller when T is large.

4.3 Sketched Adam-type Algorithms

4.3.1 SketchedAMSGrad (Parameter Averaging)

In this subsection, we will propose the SketchedAMSGrad (PA) algorithm using parameter averaging, the description of which is shown in Algorithm 3.

In Algorithm 3, we use the AMSGrad algorithm to update each worker node, based on the local momentum term $m_t^{(i)}$ and exponential moving averages of squared past gradients $v_t^{(i)}$. α_t is the stepsize and $\beta_1, \beta_2 \in (0,1)$ are the exponential moving average hyperparameters in the Adam-type algorithm. $\varepsilon > 0$ is the initial value of v_0 to avoid zero denominators. The multiplication, division, and square operation between vectors are component-wise. We use sketching to improve communication efficiency and average the parameters. We also use error-feedback to further accelerate the convergence.

For convenience, we also use the notation $\mathscr{S}$ and $\mathscr{U}$ defined in [96] to represent the sketch operator and the unsketching operator. They can be treated as a compressor that will approximately recover the true top-k coordinates. In practice, we use a second round communication which is also required in SketchedSGD [43]. After unsketching,

we get an estimation of the aggregated mini-batch gradient which is denoted by $\mathscr{U}(S_t)$. Then we select the largest Pk coordinates to extract their exact values before sketching from each worker during the second round communication. Finally, we select the top-k coordinates among these Pk coordinates as Δ_t and send them back to each worker. $\Delta_t^{(i)}$ contains the corresponding k coordinates in $m_t^{(i)}/\sqrt{\hat{v}_t^{(i)}} + \frac{\alpha_{t-1}}{\alpha_t} e_{t-1}^{(i)}$ and automatically satisfies $\Delta_t = \frac{1}{n}\sum_{i=1}^{n} \Delta_t^{(i)}$. Therefore, in each iteration, the total communication cost is $|S|+Pk+k$ and the compression rate is $2d/(|S|+Pk+k)$ where $|S|$ is the sketch size.

Using lemma 1 in [43] and replacing $\tilde{g}_t$ and $\bar{g}_t^i$ with Δ_t and $(m_t^{(i)}/\sqrt{\hat{v}_t^{(i)}} + \frac{\alpha_{t-1}}{\alpha_t} e_{t-1}^{(i)})$, we can obtain the following Lemma 4.1.

Lemma 4.1. *In Algorithm 3, let* $\tilde{\Delta}_t = \frac{1}{n}\sum_{i=1}^{n}(m_t^{(i)}/\sqrt{\hat{v}_t^{(i)}} + \frac{\alpha_{t-1}}{\alpha_t} e_{t-1}^{(i)})$*, and give sketch size* $\Theta(k\log(d/\delta))$*, with the probability* $\geq 1-\delta$*, we have*

$$\|\Delta_t - \tilde{\Delta}_t\|^2 \leq (1-\frac{k}{d})\|\tilde{\Delta}_t\|^2. \tag{4.2}$$

Lemma 4.1 indicates that Δ_t is an estimation of $\tilde{\Delta}_t$ and illustrates how the sketch technique can serve as a compressor.

4.3.2 SketchedAMSGrad (Gradient Averaging)

In the subsection, we propose the SketchedAMSGrad (GA) algorithm using gradient averaging, which is demonstrated in Algorithm 4.

In Algorithm 4, the meanings of the hyperparameters α_t, β_1 and β_2 are the same as those of Algorithm 3. We also keep track of the local momentum term $m_t^{(i)}$ on each node but the exponential moving averaging squared gradient v_t is defined on the master node. The index set $\mathscr{I}_t$ represents the coordinates updated in iteration t, which is obtained by the unsketching operator. Notation $h_t^{(i)} = (g_t^{(i)})_{\mathscr{I}_{t-1}}$ means for $\forall j \in \mathscr{I}_{t-1}$, $h_t^{(i)}$ maintains the j-th coordinate of $g_t^{(i)}$. Otherwise, if $j \notin \mathscr{I}_{t-1}$, the j-th coordinate of $h_t^{(i)}$ is 0. We define $\mathscr{I}_t$ in this way

Algorithm 3 SketchedAMSGrad (parameter averaging)

Input: initial value x_1, sketching operator $\mathscr{S}$ and unsketching operator $\mathscr{U}$.

Set: $m_0^{(i)} = \mathbf{0}$, $v_0^{(i)} = \hat{v}_0^{(i)} = \mathbf{ffl}$, $e_0^{(i)} = \mathbf{0}$ on i-th worker node.

for $t = 1$ **to** T **do**

On i-th worker node:

Estimate a stochastic gradient $g_t^{(i)}$;

Compute $m_t^{(i)} = \beta_1 m_{t-1}^{(i)} + (1-\beta_1) g_t^{(i)}$;

$v_t^{(i)} = \beta_2 v_{t-1}^{(i)} + (1-\beta_2)[g_t^{(i)}]^2$;

$\hat{v}_t^{(i)} = \max\{\hat{v}_{t-1}^{(i)}, v_t^{(i)}\}$;

Sketch $S_t^{(i)} = \mathscr{S}(m_t^{(i)} / \sqrt{\hat{v}_t^{(i)}} + \frac{\alpha_{t-1}}{\alpha_t} e_{t-1}^{(i)})$;

Send $S_t^{(i)}$ to the master node;

Send $\Delta_t^{(i)}$ to the master node after unsketching;

Compute $e_t^{(i)} = m_t^{(i)} / \sqrt{\hat{v}_t^{(i)}} + \frac{\alpha_{t-1}}{\alpha_t} e_{t-1}^{(i)} - \Delta_t^{(i)}$;

Receive Δ_t from the master node;

Update $x_{t+1} = x_t - \alpha_t \Delta_t$.

On the master node:

Aggregate $S_t = \frac{1}{n} \sum_{i=1}^n S_t^{(i)}$;

Unsketch $\Delta_t = \frac{1}{n} \sum_{i=1}^n \Delta_t^{(i)} = \text{Top-}k(\mathscr{U}(S_t))$;

Send Δ_t back to each worker node;

Update $x_{t+1} = x_t - \alpha_t \Delta_t$.

end for

because we want to accumulate the coordinates of the squared gradient that are just updated and we want to define an auxiliary sequence that makes the convergence analysis more convenient. The algorithm 4 is a gradient averaging algorithm because if no compressor is applied, this algorithm degenerates to the common distributed AMSGrad optimizer. In Algorithm 4 the unsketching operator $\mathscr{U}$ requires a vector $\hat{v}_t$ as another input and is used to recover the top-k coordinates of term $\tilde{\Delta}_t$, which is defined as follows.

$$\tilde{\Delta}_t = \frac{1}{n} \sum_{i=1}^n \tilde{\Delta}_t^{(i)}, \ \tilde{\Delta}_t^{(i)} = \hat{v}_t^{-1/2} (m_t^{(i)} + \frac{\alpha_{t-1}}{\alpha_t} e_{t-1}^{(i)}) \tag{4.3}$$

The index set of these k coordinates is denoted as $\mathscr{I}_t$. The implementation of $\mathscr{U}$ is shown in Algorithm 5, which is established on the original sketching and unsketching operator.

Algorithm 4 SketchedAMSGrad (gradient averaging)

Input: initial value x_1, sketching operator $\mathscr{S}$ and unsketching operator $\mathscr{U}$.

Set: $m_0^{(i)} = \mathbf{0}$, $e_0^{(i)} = \mathbf{0}$ on i-th worker node; $v_0 = \hat{v}_0$ on the master node; index set $\mathscr{I}_0 = \varnothing$.

for $t = 1$ **to** T **do**

On i-th worker node:

Estimate a stochastic gradient $g_t^{(i)}$;

Compute $m_t^{(i)} = \beta_1 m_{t-1}^{(i)} + (1-\beta_1) g_t^{(i)}$;

Send $h_t^{(i)} = (g_t^{(i)})_{\mathscr{I}_{t-1}}$ to the master node;

Sketch $S_t^{(i)} = \mathscr{S}(m_t^{(i)} + \frac{\alpha_{t-1}}{\alpha_t} e_{t-1}^{(i)})$;

Send $S_t^{(i)}$ to the master node;

Send $\Delta_t^{(i)}$ to the master node after unsketching;

Compute $e_t^{(i)} = m_t^{(i)} + \frac{\alpha_{t-1}}{\alpha_t} e_{t-1}^{(i)} - \Delta_t^{(i)}$;

Receive Δ_t from the master node;

Update $x_{t+1} = x_t - \alpha_t \Delta_t$.

On the master node:

Aggregate $h_t = \frac{1}{n}\sum_{i=1}^n h_t^{(i)}$;

Compute $v_t = \beta_2 v_{t-1} + (1-\beta_2) h_t^2$;

$\hat{v}_t = \max\{\hat{v}_{t-1}, v_t\}$;

Aggregate $S_t = \frac{1}{n}\sum_{i=1}^n S_t^{(i)}$;

Unsketch $\Delta_t = \frac{1}{n}\sum_{i=1}^n \Delta_t^{(i)} = \text{Top-}k(\mathscr{U}(S_t, \hat{v}_t))$;

Send Δ_t back to each worker node;

Update $x_{t+1} = x_t - \alpha_t \Delta_t$.

end for

According to the linear property of sketching $\mathscr{S}$, it is equivalent to compress $\tilde{\Delta}_t$ by $\mathscr{S}$ and then unsketch it by the normal unsketching operator. $\Delta_t^{(i)}$ contains the coordinates of $\tilde{\Delta}_t^{(i)}$ that belongs to index set $\mathscr{I}_t$ and $\Delta_t = \frac{1}{n}\sum_{i=1}^n \Delta_t^{(i)}$. Therefore, using lemma 1 in [43] and replacing $\tilde{g}_t$ and $\tilde{g}_t^i$ with Δ_t and $\tilde{\Delta}_t^{(i)}$, we reach our following Lemma 4.2.

Lemma 4.2. *With sketch size* $\Theta(k\log(d/\delta))$ *and with probability* $\geq 1-\delta$ *in Algorithm 4, we have*

$$\|\Delta_t - \tilde{\Delta}_t\|^2 \leq (1 - \frac{k}{d})\|\tilde{\Delta}_t\|^2 \tag{4.4}$$

Lemma 4.2 is the key lemma to the analysis of our Algorithm 4 which provides an estimation of term $m_t/\sqrt{\hat{v}_t}$. It is also a motivation to apply sketching in communication-

Algorithm 5 Unsketching Operator in Algorithm 4

Input: $r \times c$ sketch S, vector v, bucket hashes $\{h_j\}_{j=1}^{r}$, original unsketching operator $\mathscr{U}_0$.

```
for i = 1 to d do
  for j = 1 to r do
    S[j, h_j(i)] = S[j, h_j(i)] / sqrt(v_i)
  end for
end for
return U_0(S)
```

efficient Adam-type algorithms. As the top-k coordinates of $m_t/\sqrt{\hat{v}_t}$ and m_t are likely to change much, it is hard to estimate the Adam update term $m_t/\sqrt{\hat{v}_t}$ by the known vector $m_t^{(i)}$ on each node. However, the sketching technique makes it possible within the communication cost of $O(\log(d))$.

In Algorithm 4, thus, the total communication cost at each iteration is $|S|+Pk+2k$ and the compression rate is $2d/(|S|+Pk+2k)$ where $|S|$ is the sketch size.

In fact, our SketchedAMSGrad (GA) algorithm is compatible with 1-bit Adam algorithm. We can also regard v_{T_w} in the 1-bit Adam algorithm as the initial value of v_0 in Algorithm 4. The only difference is that in the theoretical analysis we need to replace the initial value ε with the v_{min} defined in the 1-bit Adam. Moreover, if we do not send h_t or update v_t, our algorithm is reduced to the 1-bit Adam with sketching compressor.

4.4 Convergence Analysis

In this section, we provide the convergence analysis of our algorithms. Due to the space limit, we will only provide the conclusions of Theorem 4.1 and Theorem 4.2. We begin with some mild assumptions.

Assumption 4.1. *(Lipschitz Gradient) There is a constant L such that for $\forall x, y \in R^d$, $\|\nabla f(x) - \nabla f(y)\| \leq L\|x-y\|$.*

Assumption 4.2. *(Lower Bound) Function $f(x)$ has the lower bound, i.e.,* $\inf_{x\in R^d} f(x) = f^* > -\infty$

Assumption 4.3. *(Bounded Gradient) There is a constant G such that for* $\forall i \in \{1,\cdots,n\}$, $\forall \xi_i \sim D_i$, *we have* $\|\nabla F_i(x;\xi_i)\|_\infty \leq G$.

These assumptions are commonly used in related works of Adam-type algorithms in nonconvex optimization [2, 14, 139].

4.4.1 SketchedAMSGrad (PA)

Theorem 4.1. *Assume that Assumption 1 to Assumption 3 are satisfied and data distribution* $\{D_i\}_{i=1}^n$ *are identical. In Algorithm 3, let* $\beta_1 < 1$, $\beta_2 < 1$, $\varepsilon > 0$ *and* $\alpha_t = \frac{\alpha}{\sqrt{1+T}}$, $\alpha > 0$. *Then we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(x_t)\|^2 \leq \frac{C_1}{\sqrt{T}} + \frac{C_2}{T},$$

where constants C_1 *and* C_2 *are independent of* T.

4.4.2 SketchedAMSGrad (GA)

Theorem 4.2. *Assume that Assumptions 1-3 are satisfied. In Algorithm 4, let* $\beta_1 < 1$, $\beta_2 < 1$, $\varepsilon > 0$ *and* $\alpha_t = \frac{\alpha}{\sqrt{1+T/n}}$, $\alpha > 0$. *Then we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(x_t)\|^2 \leq \frac{C_1}{\sqrt{nT}} + \frac{C_1+C_2}{T},$$

where constants C_1 *and* C_2 *are independent of* T.

Corollary 4.1. *In Theorem 4.2, we can see that the dominant term is* $O(\frac{1}{\sqrt{nT}})$, *which achieves a linear speedup compared with AMSGrad in nonconvex optimization [14].*

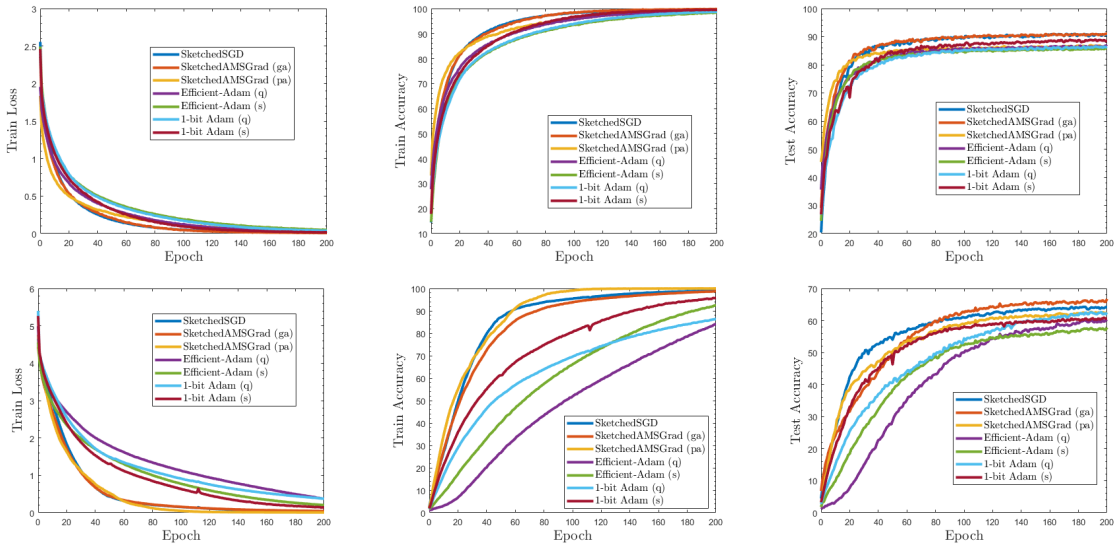


Figure 4.1: The experimental results of training ResNet-50 on CIFAR10 and CIFAR100. Figures (a), (b) and (c) show the experimental results on CIFAR10. Figures (d), (e) and (f) show the experimental results on CIFAR100. Figures (a) and (d) show the train loss value. Figures (b) and (e) show the train accuracy. Figures (c) and (f) show the test accuracy.

Remark 4.1. *In Algorithm 4, the data distribution D_i's are allowed to be non-identical. In both Algorithm 3 and Algorithm 4, β_1 and β_2 are constants in $(0,1)$, which is applicable to the common default settings that $\beta_1 = 0.9$ and $\beta_2 = 0.999$.*

Remark 4.2. *In [139], the $\log(T)$ term can be shaved using a fixed stepsize of $\alpha_t = O(\frac{1}{\sqrt{T}})$. Notice that in our methods, we can also remove the logarithm term by adopting such a stepsize of $\alpha_t = \alpha = O(\frac{1}{\sqrt{T}})$ and the convergence rate becomes $O(\frac{1}{\sqrt{nT}})$ (proof can be found in the Supplementary Material). However, in order to make the algorithm more flexible, we prefer the current stepsize and do not set an upper bound to the total number of iterations T.*

4.4.3 Discussion on the Compression Rate

Now we will discuss the how the compression rate, *i.e.*, the choice of k influences the convergence rate. In both Theorem 4.1 and Theorem 4.2, constant C_1 is independent on k. Hence, the dominating term is not affected by the compression rate. This result is the same as many other gradient compression methods. According to the definitions of

C_2 in Theorem 4.1 and Theorem 4.2, the second dominant term is affected by k in the form $O(\frac{1}{(k/d)^2 T})$ for the parameter averaging and gradient averaging SketchedAMSGrad algorithms.

4.5 Experiments

In this section we will show the experimental results of the distributed data mining task of image categorization to validate our methods. All experiments are run on a server with a 64-core Intel Xeon E5-2683 v4 2.10GHz processor and 4 Nvidia P40 GPUs. We simulate the edge-based training environment on the GPU server where the root process represents the edge server, each process represents an IoT device, and the dataset represents the captured data. The code is implemented by PyTorch 1.4.0 and CUDA 10.1.

4.5.1 ResNet on CIFAR

The experimental task is to train ResNet-50 [36] using CIFAR10 and CIFAR100 datasets [55], which are benchmark datasets for image classification tasks. Both CIFAR10 and CIFAR100 contain 60,000 32×32 pixel images with RGB channels, 50,000 of which is considered a training set and the other 10,000 of which is used for testing. The images are distributed evenly over 10 and 100 classes for CIFAR10 and CIFAR100, respectively. The ResNet-50 model has about 25M parameters. We use cross-entropy loss to train the neural network.

In our experiment, we compare our SketchedAMSGrad (PA) and SketchedAMSGrad (GA) with Sketched-SGD [43], Efficient-Adam [12] and 1-bit Adam [104]. For Efficient-Adam and 1-bit Adam, we consider both quantization and sparsification as a compressor. For gradient quantization, we adopt the following scheme used in [138] which is a variant

of SignSGD:

$$C(x) = \frac{\|x\|_1}{d} sign(x) \tag{4.5}$$

The number of workers in this task is set to 16. The batch size on each worker node is 32. Hence, the total batch size at each iteration is 512. We run 200 epochs in total. For each algorithm, we perform a grid search for the learning rate from $\{0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$ and ε from $\{1e-2, 1e-4, 1e-6\}$ and select the values that get the best training result. For Adam-type algorithms, β_1 and β_2 are set to be common choices as $\beta_1 = 0.9$ and $\beta_2 = 0.999$. For 1-bit Adam, similar to [104], we run 13 epochs to compute the Adam-preconditioned vector v_{T_w}. For sketching methods, the sketch is set to have 100,000 columns and 10 rows. We set $k = 50,000$ and $P = 8$. For Efficient-Adam and 1-bit Adam with top-k compressor, we choose $k = 750,000$. Therefore, all algorithms implemented in this task are communication-efficient and approximately achieve the same compression rate (about $32\times$ reduction).

Figure 4.1 shows the experimental results of this image classification task. Based on the result of the train loss value, we can see that the three sketching methods converge faster than other algorithms on both CIFAR10 and CIFAR100 dataset. When comparing the train accuracy, the sketching methods are still advantageous over other methods. Our parameter averaging and gradient averaging SketchedAMSGrad and SketchedSGD approximately have the same performance. On CIFAR100, our parameter averaging SketchedAMSGrad is slightly better in the train accuracy results. Based on the test accuracy results, our gradient averaging SketchedAMSGrad and SketchedSGD also outperform other algorithms in both datasets. On CIFAR100, our gradient averaging SketchedAMSGrad achieves the best performance in test accuracy. From this experiment we can see that although using compression on the returning message avoids the growing $O(n)$ communication cost issue of local top-k (mentioned in [43]), it probably encounters slow convergence since the estimator

is too far away from the true top-k coordinates.

Theoretically, when the sketch size is larger, the probability of recovering top-k coordinates is higher. The sketch size used in this experiment is 1,000,000. On CIFAR10, the test accuracy of our SketchedAMSGrad (GA) is 91.04%. When we increase the sketch size to 2,000,000 and 3,000,000, the test accuracy is increased by 0.24% and 0.39% respectively. Thus, we can see the influence of sketch size. If the sketch size is larger, our algorithm will probably show better performance.

4.6 Conclusion

In this chapter, we propose a class of communication-efficient distributed adaptive gradient algorithm named SketchedAMSGrad based on two averaging strategies to tackle the high communication cost issue for distributed training. Specifically, the communication cost of our algorithm in each iteration is reduced to $O(\log(d))$ from $O(d)$. Moreover, we proved that our algorithm achieves a fast convergence rate of $\tilde{O}(\frac{1}{\sqrt{nT}})$, which achieves the linear speedup compared with single-machine AMSGrad. In particular, our analysis of gradient averaging SketchedAMSGrad works for both identical and non-identical data distribution. To the best of our knowledge, our algorithm is the first one to apply the sketching technique to adaptive gradient methods.

Chapter 5: Second-Order Optimality in Decentralized Optimization

5.1 Introduction

Decentralized optimization is a class of distributed optimization that trains models in parallel across multiple worker nodes over a decentralized communication network. Decentralized optimization has recently attracted increased attention in machine learning and has emerged as a promising framework to solve large-scale tasks because of its ability to reduce communication costs. In the conventional centralized paradigm, all worker nodes need to communicate with the central node, resulting in a high communication cost on the central node when the number of nodes is large or the transmission between the center and some remote nodes suffers network latency. In contrast, decentralized optimization avoids these issues, as each worker node only communicates with its neighbors.

Although decentralized optimization has shown advantageous performance in many previous works ([64, 107]), the study of second-order optimality for decentralized stochastic optimization algorithms is still limited. Escaping the saddle point and finding local minima is a core problem in nonconvex optimization since saddle point is a category of first-order stationary point that can be reached by many gradient-based optimizers such as gradient descent, but it is not the expected point to minimize the objective function.

Perturbed gradient descent ([45]) and negative curvature descent ([5, 129]) are two primary pure gradient-based methods (not involving second-order derivatives) to achieve second-order optimality. Typically, the perturbed gradient descent method is composed of a

descent phase and an escaping phase. If the gradient norm is large, the algorithm will run the descent phase as normal. Otherwise, it will run the escaping phase to discriminate whether the candidate first-order stationary point is a saddle point or local minimum. The negative curvature descent method escapes the saddle point by computing the direction of negative curvature at the candidate point. If it is categorized as a saddle point, then the algorithm will update along the direction of negative curvature. Generally, it involves a nested loop to perform the negative curvature subroutine.

Currently, a solution to the second-order optimality of the decentralized problem in the deterministic setting has been proposed. Perturbed Decentralized Gradient Tracking (PDGT) ([112]) is a decentralized deterministic algorithm that adopts the perturbed gradient descent strategy to achieve a second-order stationary point. However, it is expensive to compute full gradients for large machine learning models. It is crucial to propose a stochastic algorithm to obtain second-order optimality for decentralized problems. Besides, there are some drawbacks of PDGT that make it less efficient and make it difficult to generalize to the stochastic setting. These drawbacks are also the key challenges in achieving second-order optimality for decentralized algorithms, which are listed as follows:

(1) PDGT runs a fixed number of iterations in the descent phase and escaping phase, such that the phases of all nodes can be changed simultaneously. This strategy works because the descent is easy to estimate in the deterministic setting. Nevertheless, the exact descent of stochastic algorithms over a fixed number of iterations is hard to be bounded because of randomness and noises. If the fixed number is not large enough, it is possible that the averaged model parameter is not a first-order stationary point. If the fixed number is as large as the expected number of iterations to achieve first-order stationary point, the algorithm will become less efficient, as it is probably stuck at a saddle point for a long time before drawing the perturbation, especially in the second and later descent phase. Specifically, applying a fixed number of iterations in each phase results in a complexity of at least $\tilde{O}(\varepsilon^{-4.5})$ (see

Appendix D.4), which is higher than $\tilde{O}(\varepsilon^{-3})$ of our method. Therefore, we are motivated to propose an algorithm that can change phases *adaptively* (based on the runtime gradient norm) and *independently* (not required to consider the status on other nodes or notify other nodes).

(2) In PDGT the perturbations on all nodes are drawn from the same random seed. Besides, a coordinating protocol involving broadcast and aggregation is used to compute the averaged model parameter and the descent of the overall loss function to discriminate the candidate point. These strategies, together with the fixed number of iterations, act as a hidden coordinator to make PDGT discriminate saddle point in the same way as centralized algorithms. However, when the number of worker nodes is large, it is time consuming to perform broadcast or aggregation over the whole decentralized network. Moreover, when generalized to stochastic setting, the changing of phase is not guaranteed to be synchronized. Furthermore, we will note in the Supplementary Material that the consensus error $\frac{1}{n}\sum_{i=1}^{n}\|x_t^{(i)} - \bar{x}_t\|^2$ is another factor that impacts the effectiveness of perturbed gradient descent, which is not present in centralized problems. All of the above issues are theoretical difficulties to study and ensure second-order optimality for decentralized stochastic algorithms.

([113]) proves the theoretical guarantee of second-order optimality for a decentralized stochastic algorithm with perturbed gradient descent. However, it does not provide a non-asymptotic analysis to estimate the convergence rate or gradient complexity. The effectiveness of the result relies on a sufficiently small learning rate, and it does not present a specific algorithm. The analysis is based on the assumption that the iteration formula can be approximated by a centralized update scheme when the learning rate is small enough. However, in practice, it is difficult to maintain an ideally small learning rate, and the iterative update process can be more complex, as previously mentioned. To our best knowledge, the second-order optimality issue of decentralized stochastic algorithms

with non-asymptotic analysis is still not solved. Therefore, we are motivated to study this important and challenging issue and raise the following questions.

Can we design a decentralized stochastic optimization algorithm with non-asymptotic analysis to find local minima efficiently? Is the algorithm still effective in discriminating the saddle point even if each node can change its phase adaptively and independently without coordinating protocols?

The answer is affirmative. In this chapter, we propose a novel gradient-based algorithm named PErturbed DEcentralized STORM ALgorithm (PEDESTAL) which is the first decentralized stochastic algorithm to find a second-order stationary point. We adopt perturbed gradient descent to ensure second-order optimality and use the STORM ([18]) estimator to accelerate convergence. We provide a completed convergence analysis to theoretically guarantee second-order optimality. More details about the reason for choosing perturbed gradient descent and technical difficulties are discussed in Section 5.3.2. Next, we will introduce the problem setup in this chapter.

We focus on the following decentralized optimization problem:

$$\min_x f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x),\ f_i(x) = \mathbb{E}_{\xi\sim D_i} F_i(x,\xi) \tag{5.1}$$

where n is the number of worker nodes in the decentralized network and f_i is the local loss function on i-th worker node. Here f_i is supposed to take the form of a stochastic expectation on the local data distribution D_i, which covers a variety of optimization problems including the finite-sum problem and the online problem. Data distributions on different nodes are allowed to be heterogeneous. The objective function f is nonconvex such that saddle points probably exist.

The goal of our method is to find $O(\varepsilon, \varepsilon_H)$-second-order stationary point of problem 5.1, which is defined by the point x satisfying $\|\nabla f(x)\| \leq \varepsilon$ and $\min eig(\nabla^2 f(x)) \geq -\varepsilon_H$, where

$eig(\cdot)$ represents the eigenvalues. The classic setting is $\varepsilon_H = \sqrt{\varepsilon}$.

We summarize the contributions of this chapter as follows.

- We propose a novel algorithm PEDESTAL, which is the first decentralized stochastic gradient-based algorithm to achieve second-order optimality with non-asymptotic analysis.
- We provide a new analysis framework to support changing phases adaptively and independently on each node without any coordinating protocols involving broadcast or aggregation. We also address certain technical difficulties unique to decentralized optimization to justify the effectiveness of perturbed gradient descent in discriminating saddle point.
- We prove that our PEDESTAL achieves the gradient complexity of $\tilde{O}(\varepsilon^{-3} + \varepsilon\varepsilon_H^{-8} + \varepsilon^4\varepsilon_H^{-11})$ to find $O(\varepsilon, \varepsilon_H)$-second-order stationary point. Particularly, PEDESTAL achieves the gradient complexity of $\tilde{O}(\varepsilon^{-3})$ in the classic setting $\varepsilon_H = \sqrt{\varepsilon}$, which matches state-of-the-art results of centralized counterparts or decentralized methods to find first-order stationary point.

5.2 Related Work

In this section we will introduce the background of related works. The comparison of important features is shown in Table 5.1. Here $\tilde{O}(\cdot)$ refers to the big O notation that hides the logarithmic terms.

5.2.1 Decentralized Algorithms for First-Order Optimality

Decentralized optimization is an efficient framework to solve problem 5.1 collaboratively by multiple worker nodes. In each iteration, a worker node only needs to communicate

with its neighbors. One of the best known decentralized stochastic algorithms is D-PSGD ([64]), which integrates average consensus with local stochastic gradient descent steps and shows a competitive result with centralized SGD. The ability to address Non-IID data is a limitation of D-PSGD and some variants of D-PSGD are studied to address the issue of data heterogeneity, such as D^2 ([107]) by storing the previous status and GT-DSGD ([125]) by using gradient tracking ([74, 127]). D-GET ([99]) and D-SPIDER-SFO ([90]) improve the gradient complexity of D-PSGD from $O(\varepsilon^{-4})$ to $O(\varepsilon^{-3})$ utilizing the variance-reduced gradient estimator SPIDER ([26]). GT-HSGD also achieves a gradient complexity of $O(\varepsilon^{-3})$ by combining gradient tracking and the STORM gradient estimator ([18]). SPIDER requires a large batchsize of $O(\varepsilon^{-1})$ on average and a mega batchsize of $O(\varepsilon^{-2})$ periodically. In contrast, STORM only requires a large batch in the first iteration. After that, the batchsize can be as small as $O(1)$, making STORM more efficient to implement in practice.

5.2.2 Centralized Algorithms for Second-Order Optimality

Perturbed gradient descent is a simple and effective method to escape saddle points and find local minima. PGD ([45]) is the representative of this family of algorithms, which achieves second-order optimality in the deterministic setting. It draws a perturbation when the gradient norm is small. If this point is a saddle point, the loss function value will decrease by a certain threshold within a specified number of iterations (*i.e.*, breaking the escaping phase) with high probability. Otherwise, the candidate point is regarded as a second-order stationary point. In stochastic setting, Perturbed SGD perturbs every iteration and suffers a high gradient complexity of $O(\varepsilon^{-8})$ to achieve $O(\varepsilon, \sqrt{\varepsilon})$-second-order stationary point and the gradient complexity hides a polynomial factor of dimension d. CNC-SGD requires a Correlated Negative Curvature assumption and the gradient complexity of $\tilde{O}(\varepsilon^{-5})$ to achieve classic second-order optimality. SSRGD ([62]) adopts the same two-phase scheme

as PGD but uses the moving distance as a criterion to discriminate the saddle point in the escaping phase. It also takes advantage of variance reduction to improve the gradient complexity to $\tilde{O}(\varepsilon^{-3.5})$. Pullback ([15]) proposes a pullback step to further enhance the gradient complexity to $\tilde{O}(\varepsilon^{-3})$, which matches the best result of reaching the first-order stationary point.

5.2.3 Stochastic Gradient Descent

A branch of study of stochastic gradient descent argues that SGD can avoid saddle point under certain conditions. However, that is completely different from the problem we focus on. In this chapter we propose a method that can find local minima effectively for a general problem 5.1, while escaping saddle point by stochastic gradient itself depends on some additional assumptions. For example, ([81]) requires that the gradient noise be uniformly excited. According to our experimental result in Section 5.5, we can see that in some cases stochastic gradient descent cannot escape the saddle point effectively or efficiently. Besides, the gradient noise in the variance reduced methods is reduced in order to accelerate the convergence. Our experimental results indicate that the gradient noise in variance reduced algorithms is not as good as SGD to serve as the perturbation to avoid saddle point. Therefore, it is necessary to study the second-order stationary point for variance reduced algorithms so as to enable both second-order optimality and fast convergence.

5.3 Method

5.3.1 Algorithm

In this section, we will introduce our PEDESTAL algorithm, which is demonstrated in Algorithm 6. Suppose there are n worker nodes in the decentralized communication network

Name	Averaged Batchsize	Gradient Complexity	Classic Setting
D-PSGD [64]	$O(1)$	$O(\varepsilon^{-4})$	-
GT-DSGD [125]	$O(1)$	$O(\varepsilon^{-4})$	-
D-GET [99]	$O(\varepsilon^{-1})$	$O(\varepsilon^{-3})$	-
D-SPIDER-SFO [90]	$O(\varepsilon^{-1})$	$O(\varepsilon^{-3})$	-
GT-HSGD [124]	$O(1)$	$O(\varepsilon^{-3})$	-
SGD+Neon2 [5]	$O(1)$	$\tilde{O}(\varepsilon^{-4}+\varepsilon^{-2}\varepsilon_H^{-3}+\varepsilon_H^{-5})$	$\tilde{O}(\varepsilon^{-4})$
SCSG+Neon2 [5]	$O(\varepsilon^{-0.5})$	$\tilde{O}(\varepsilon^{-10/3}+\varepsilon^{-2}\varepsilon_H^{-3}+\varepsilon_H^{-5})$	$\tilde{O}(\varepsilon^{-3.5})$
Natasha2+Neon2 [5]	$O(\varepsilon^{-2})$	$\tilde{O}(\varepsilon^{-3.25}+\varepsilon^{-3}\varepsilon_H^{-1}+\varepsilon_H^{-5})$	$\tilde{O}(\varepsilon^{-3.5})$
SPIDER-SFO$^+$ [26]	$O(\varepsilon^{-1})$	$\tilde{O}(\varepsilon^{-3}+\varepsilon^{-2}\varepsilon_H^{-2}+\varepsilon_H^{-5})$	$\tilde{O}(\varepsilon^{-3})$
Perturbed SGD [28]	$O(1)$	$O(\varepsilon^{-4}+\varepsilon_H^{-16})$	$O(\varepsilon^{-8})$
CNC-SGD [19]	$O(1)$	$\tilde{O}(\varepsilon^{-4}+\varepsilon_H^{-10})$	$\tilde{O}(\varepsilon^{-5})$
SSRGD [62]	$O(\varepsilon^{-1})$	$\tilde{O}(\varepsilon^{-3}+\varepsilon^{-2}\varepsilon_H^{-3}+\varepsilon^{-1}\varepsilon_H^{-4})$	$\tilde{O}(\varepsilon^{-3.5})$
Pullback [15]	$O(\varepsilon^{-1})$	$\tilde{O}(\varepsilon^{-3}+\varepsilon_H^{-6})$	$\tilde{O}(\varepsilon^{-3})$
PDGT [112]	Full	-	-
PEDESTAL-S (ours)	$O(1)$	$\tilde{O}(\varepsilon^{-3})$, $\varepsilon_H \geq \varepsilon^{0.2}$	-
PEDESTAL (ours)	$O(\varepsilon^{-3/4})$	$\tilde{O}(\varepsilon^{-3}+\varepsilon\varepsilon_H^{-8}+\varepsilon^4\varepsilon_H^{-11})$	$\tilde{O}(\varepsilon^{-3})$

Table 5.1: The comparison of important properties between related algorithms and our PEDESTAL. Column "Averaged Batchsize" is computed when $\varepsilon_H = \sqrt{\varepsilon}$. Column "Classic Setting" refers to the gradient complexity under the classic condition $\varepsilon_H = \sqrt{\varepsilon}$. The first group of algorithms are decentralized methods achieving first-order optimality. The second group of algorithms are centralized methods achieving second-order optimality. The last group of algorithms are decentralized methods achieving second-order optimality. PEDESTAL-S is a special case of PEDESTAL with $O(1)$ batchsize. The complexity of PDGT is not shown because it is not stochastic.

connected by a weight matrix W. The initial value of model parameters on all nodes are identical and equal to x_0. $x_t^{(i)}$, $v_t^{(i)}$ and $y_t^{(i)}$ are the model parameter, gradient estimator and gradient tracker on the i-th worker node in iteration t. $z_t^{(i)}$ is the temporary model parameter that is awaiting communication. $\bar{x}_t$, $\bar{v}_t$ and $\bar{y}_t$ are corresponding mean values over all nodes. Counter $esc^{(i)}$ counts the number of iterations in the current escaping phase on the i-th worker node, which is also the indicator of current phase. When it runs the descent phase on the i-th worker node $esc^{(i)}$ is set to -1; otherwise $esc^{(i)} \geq 0$.

In the first iteration, the gradient estimator is computed based on a large batch size with b_0. Beginning from the second iteration, the gradient estimator $v_t^{(i)}$ is calculated by small mini-batch of samples according to the update rule of STORM, which can be formulated by line 6 in Algorithm 6 where β is a hyperparameter of STORM algorithm. Notation $\nabla F_i(x_t^{(i)}, \xi_t^{(i)})$ represents the stochastic gradient obtained from a batch of samples $\xi_t^{(i)}$, which can be written as $\nabla F_i(x_t^{(i)}, \xi_t^{(i)}) = (1/|\xi_t^{(i)}|) \sum_{j \in \xi_t^{(i)}} F_i(x_t^{(i)}, j)$.

After calculating $v_t^{(i)}$, each worker node communicates with its neighbors and updates the gradient tracker $y_t^{(i)}$. Inspired by the framework of Perturbed Gradient Descent, our PEDESTAL method also consists of two phases, the descent phase and the escaping phase. If the worker node i is in the descent phase and the norm $\|y_t^{(i)}\|$ is smaller than the given threshold C_v, then it will draw a perturbation ξ uniformly from $B_0(r)$ and update $z_t^{(i)} = x_t^{(i)} + \xi$. The phase is switched to escaping phase and $esc^{(i)}$ is set to 0. The anchor $\tilde{x}^{(i)} = x_t^{(i)}$ is saved and will be used to discriminate whether the escaping phase is broken. After this iteration counter $esc^{(i)}$ will be added by 1 in each subsequent iteration until the moving distance from the anchor on i-th worker node (*i.e.*, $\|x_t^{(i)} - \tilde{x}^{(i)}\|$) is larger than the threshold C_d for some t, which breaks the escaping phase and turns back to the descent phase. If the condition of drawing perturbation is not satisfied, $z_t^{(i)}$ is updated by $z_t^{(i)} = x_t^{(i)} - \eta y_t^{(i)}$ no matter which phase is running currently.

If the i-th worker node's counter $esc^{(i)}$ is larger than the threshold C_T, it indicates that

Algorithm 6 Perturbed Decentralized STORM Algorithm (PEDESTAL)

Input: initial value $x_0^{(i)} = x_0$, $v_{-1}^{(i)} = \mathbf{0}$, $y_{-1}^{(i)} = \mathbf{0}$, $esc^{(i)} = -1$.
Parameter: b_0, b_1, η, β, r, C_v, C_d, C_T.

On i-th node:
for $t = 0, 1, \ldots, T-1$ **do**
  if $t = 0$ **then**
    Compute $v_0^{(i)} = \nabla F_i(x_0, \xi_0^{(i)})$ with $|\xi_0^{(i)}| = b_0$.
  else
    Compute $v_t^{(i)} = \nabla F_i(x_t^{(i)}, \xi_t^{(i)}) + (1-\beta)(v_{t-1}^{(i)} - \nabla F_i(x_{t-1}^{(i)}, \xi_t^{(i)}))$ with $|\xi_t^{(i)}| = b_1$.
  end if
  Communicate and update the gradient tracker: $y_t^{(i)} = \sum_{j=1}^n w_{ij}(y_{t-1}^{(j)} + v_t^{(j)} - v_{t-1}^{(j)})$.
  if $esc^{(i)} = -1$ and $\|y_t^{(i)}\| \le C_v$ **then**
    Draw a perturbation $\xi \sim B_0(r)$ and update $z_t^{(i)} = x_t^{(i)} + \xi$.
    Save $x_t^{(i)}$ as $\tilde{x}^{(i)}$ and set $esc^{(i)} = 0$.
  else
    Update $z_t^{(i)} = x_t^{(i)} - \eta y_t^{(i)}$.
  end if
  Communicate and update the model parameter: $x_{t+1}^{(i)} = \sum_{j=1}^n w_{ij} z_t^{(j)}$.
  if $esc^{(i)} \ge 0$ **then**
    Reset $esc^{(i)} = -1$ **if** $\|x_{t+1}^{(i)} - \tilde{x}^{(i)}\| > C_d$ **else** update $esc^{(i)} = esc^{(i)} + 1$.
  end if
end for

Return: $\bar{x}_{t-C_T}$ if there are at least $\frac{n}{10}$ nodes satisfying $esc^{(i)} \ge C_T$.

$\bar{x}_{t-C_T}$ is a candidate second-order stationary point. When at least $\frac{n}{10}$ nodes satisfy condition $esc^{(i)} \ge C_T$, the algorithm is terminated. Notice that the fraction is set to $\frac{1}{10}$ for convenience in the convergence analysis. Our algorithm also works for other constant fractions. From Algorithm 6 we can see that the decision to change phases on each node depends only on its own status, which is adaptive and independent. A coordination protocol, including broadcast or aggregation, is not required.

5.3.2 Discussion

Here we will discuss the insight of the algorithm design and compare the differences between our method and related works. Some novel improvements are the key to the

questions in Section 5.1.

Perturbed Gradient Descent or Negative Curvature Descent

Perturbed gradient descent and negative curvature descent are two of the most widely used pure first-order methods to find second-order stationary points. In PEDESTAL algorithm, we adopt the strategy of perturbed gradient descent rather than negative curvature descent because of the following reasons. First, negative curvature descent methods such as Neon ([129]) and Neon2 ([5]) involve a nested loop to execute the negative curvature subroutine to recognize if a first-order stationary point is a local minimum. However, in the decentralized setting, it is possible that the gradient norms on some nodes are smaller than the threshold, while others are not. Therefore, some nodes will execute the negative curvature subroutine but its neighbors may not. In this case neighbor nodes need to wait for the nodes running negative curvature subroutines, and there will be idle time on neighbor nodes. Besides, the analysis of negative curvature descent methods relies on the precision of the negative curvature direction. It is unknown if the theoretical results are still effective when only a fraction of nodes participate in the computation of negative curvature direction while the others use the gradient. In contrast, perturbed gradient descent only requires a simple operation of drawing perturbation, which is more suitable for decentralized algorithms.

Stepsize and Batchsize

In Pullback, a dynamic stepsize $\eta_t = \eta/\|v_t\|$ in the descent phase where $\eta = O(\varepsilon)$ and v_t is the gradient estimator. This stepsize is originated from SPIDER ([26]) which ensures its convergence by bounding the update distance $\|x_{t+1} - x_t\|$. In the escaping phase, Pullback adopts a larger stepsize of $O(1)$ in the escaping phase and a special pullback stepsize in the last iteration, which is the key to improve the gradient complexity. Unlike

pullback, in Algorithm 6 we adopt a consistent stepsize such that it keeps invariant even if the phase changes and all nodes always use the same stepsize. If there is no perturbation in iteration t, we have $\bar{x}_{t+1} = \bar{x}_t - \eta \bar{v}_t$, which is important for the convergence analysis. We discard the strategy in Pullback for two reasons. First, gradient normalization will probably cause divergence issues in decentralized optimization because in a centralized algorithm the gradient direction is $v_t/\|v_t\|$, which is equivalent to $v_c = \sum_{i=1}^n v_t^{(i)}/\|\sum_{i=1}^n v_t^{(i)}\|$. However, in the decentralized algorithm the average of $v_t^{(i)}$ is not available on local nodes. If the gradient normalization is done locally, we will get $v_d = \sum_{i=1}^n v_t^{(i)}/\|v_t^{(i)}\|$, which is different from v_c and the error is hard to estimate. In fact, both D-GET and D-SPIDER-SFO adopt the constant stepsize in SpiderBoost ([117]) to avoid performing the gradient normalization step. SPIDER needs the gradient normalization because $\|x_{t+1} - x_t\|$ is required to be small in the proof, while SpiderBoost improves the proof to bound $\|x_{t+1} - x_t\|$ by $\eta\|v_t\|$ which is canceled eventually. In our analysis, we also adopt the strategy in SpiderBoost. Second, in our algorithm, the change of phase occurred independently on each node. The phase-wise stepsize and pullback strategy will lead to different stepsizes among all nodes in one iteration, which will also cause potential convergence issues.

In ([15]), two versions of Pullback are proposed, *i.e.,* Pullback-SPIDER and Pullback-STORM using SPIDER and STORM as the gradient estimator, respectively. As introduced previously, one of the advantages of STORM is the ability to avoid large batch sizes. However, Pullback-STORM adopts a large batchsize of $O(\varepsilon^{-1})$ in each iteration, which violates the original intention of STORM. Besides, from Table 5.1 we can see that all algorithms achieving second-order optimality with $\tilde{O}(\varepsilon^{-3})$ gradient complexity require a large batchsize of $O(\varepsilon^{-1})$. Therefore, we propose a small batch version named PEDESTAL-S as a special case of PEDESTAL that only requires an averaged batchsize of $O(1)$.

Conditions of Termination

As a result of applying gradient tracking, we can bound $\frac{1}{n}\sum_{i=1}^{n}\|y_t^{(i)} - \bar{y}_t\|^2$ by $O(\varepsilon^2)$. Although we have such an estimation, it is still possible that the norm $\|y_t^{(i)}\|$ is as large as $O(\sqrt{n}\varepsilon)$ on some nodes when the entire decentralized network has already achieved optimality. Therefore, waiting for all nodes to reach the second-order stationary point is not an efficient strategy. This is the reason why we terminate our algorithm when only a fraction of worker nodes satisfy $esc^{(i)} \geq C_T$.

In SSRGD and Pullback, there is an upper bound of iteration numbers in the escaping phase. If the escaping phase is not broken in this number of iterations, then the candidate point is regarded as a second-order stationary point. If the escaping phase is broken, then the averaged moving distance is greater than a threshold and the loss function will be reduced by $O(\varepsilon^2)$ on average. This strategy guarantees that the algorithm will terminate with a certain gradient complexity. However, in our algorithm, worker nodes do not enter escaping phase simultaneously, and thus we do not set such an upper bound. In this case, the averaged moving distance cannot be lower bounded as C_T has no upper bound. Fortunately, we can complete our analysis by a different novel framework (see the proof outline in the Appendix). An alternative solution is to stop the update on the node that has run a certain number of iterations in the escaping phase while the algorithm continues. But that solution is also challenging since the relation between the first-achieved local optimal solution and the final global optimal solution is unknown and the analysis is nontrivial.

One remaining issue of the current termination strategy is that it involves global knowledge of how many worker nodes satisfy the termination condition. One solution is to run an additional process to track this global value. The cost of transmitting Boolean values is much less expensive than to broadcast the model. Another solution is to set a maximum iteration in practice. Generally we need to evaluate the model after certain epochs to see

if the training process is running smoothly, and we can save a checkpoint when we find a better evaluation result. Theoretical analysis ensures that an optimal solution can be visited if the number of iterations is as large as $O(\varepsilon^{-3})$.

Small Stuck Region

The theoretical guarantee of second-order optimality in SSRGD and Pullback is mainly credit to the lemma of small stuck region, which states that if there are two decoupled sequences x_t and x'_t with identical stochastic samples, $x_s = x'_s$ and $x_{s+1} - x'_{s+1} = r_0\mathbf{e_1}$ where $\mathbf{e_1}$ is the eigenvector corresponding to the smallest eigenvalue, then it satisfies $\max\{\|x_t - x_s\|,\ \|x'_t - x'_s\|\} \geq C_d$ for some $s \leq t \leq s + C_T$ with high probability. In SSRGD and Pullback, the averaged moving distance $\frac{1}{t-s}\sum_{\tau=s+1}^{t}\|x_{\tau+1} - x_\tau\|^2$ is used as the criterion to discriminate the saddle point because the small stuck region lemma can be applied in this way. However, in decentralized algorithms, some nodes enter the escaping phase before the candidate point $\bar{x}_s$ is achieved. Suppose node i enters the escaping phase in iteration s', then the averaged moving distance starting from iteration s on node i cannot be well estimated because the condition of not breaking the escaping phase on node i only guarantees the bound of averaged moving distance starting from s'. Therefore, in our method, we use the total moving distance $\|x_t^{(i)} - x_s^{(i)}\|$ as the criterion because we can obtain the estimation $\|x_t^{(i)} - x_s^{(i)}\| \leq 2C_d$ given $\|x_t^{(i)} - x_{s'}^{(i)}\| \leq C_d$ and $\|x_s^{(i)} - x_{s'}^{(i)}\| \leq C_d$. We can further complete our analysis by the small stuck region lemma. Actually, we do not require more memory because $\tilde{x}$ is the point to return in SSRGD and Pullback (hence should be saved). In practice, we can also return $\tilde{x}^{(i)}$ for any node i that draws perturbation in iteration $t - C_T$ as $\|x_t^{(i)} - \bar{x}_t\|$ can be well bounded. Besides, we discover that the consensus error $\frac{1}{n}\sum_{i=1}^{n}\|x_t^{(i)} - \bar{x}_t\|^2$ results in an extra term when proving the small stuck region lemma, which becomes another challenge. If the consensus error is not under control, it can drive x away from x_s or push x toward

x_s, regardless of what $\nabla f(x)$ is. In this manner, the stuck region cannot be estimated. In this work, we provide the corresponding proof to estimate this new term which exclusively occurred in the decentralized setting in our convergence analysis.

5.4 Convergence Analysis

5.4.1 Assumptions

In this section, we will provide the main theorem of our convergence analysis. First, we will introduce the assumptions used in this chapter. All assumptions used in this chapter are mild and commonly used in the analysis of related works.

Assumption 5.1. *(Lower Bound) The objective f is lower bounded, i.e.,* $\inf_x f(x) = f^* > -\infty$.

Assumption 5.2. *(Bounded Variance) The stochastic gradient of each local loss function is an unbiased estimator and has bounded variance, i.e., for any $i \in \{1,2,\cdots,n\}$ we have*

$$\mathbb{E}_\xi \nabla F_i(x,\xi) = \nabla f_i(x),\ \mathbb{E}_\xi \|\nabla F_i(x,\xi) - \nabla f_i(x)\|^2 \leq \sigma^2 \tag{5.2}$$

Assumption 5.3. *(Lipschitz Gradient) For all ξ and $i \in \{1,2,\cdots,n\}$, $F_i(x,\xi)$ has Lipschitz gradient, i.e., for any x_1 and x_2 we have $\|\nabla F_i(x_1,\xi) - \nabla F_i(x_2,\xi)\| \leq L\|x_1 - x_2\|$ with constant L.*

Assumption 5.4. *(Lipschitz Hessian) For all ξ and $i \in \{1,2,\cdots,n\}$, $F_i(x,\xi)$ has Lipschitz hessian, i.e., for any x_1 and x_2 we have $\|\nabla^2 F_i(x_1,\xi) - \nabla^2 F_i(x_2,\xi)\| \leq \rho\|x_1 - x_2\|$ with a constant ρ.*

Assumption 1, Assumption 2 and Assumption 3 are common assumptions used in the analysis of stochastic optimization algorithms. Assumption 4 is the standard assumption

to find second-order optimality, which is used in all algorithms that achieve second-order stationary point in Table 5.1.

Assumption 5.5. *(Spectral Gap) The decentralized network is connected by a doubly-stochastic weight matrix $W \in \mathbb{R}^{n\times n}$ that satisfies $W\mathbf{1}_n = W^T\mathbf{1}_n = \mathbf{1}_n$ and $\lambda := \|W - J\| \in [0,1)$.*

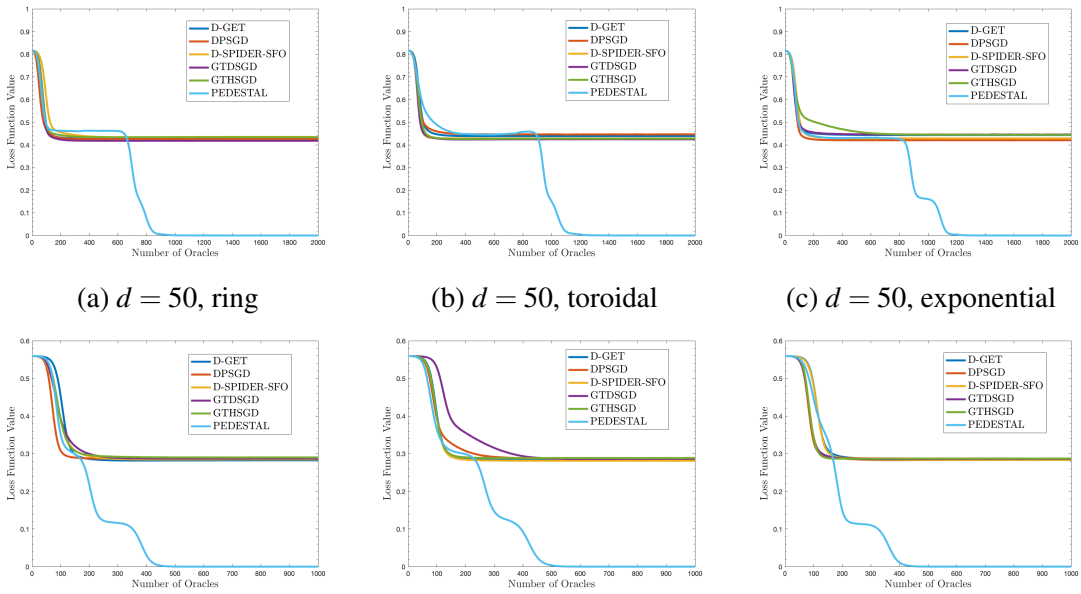
Here J is a $n \times n$ matrix with all elements equal to $\frac{1}{n}$. W is the weight matrix of the decentralized network where $w_{ij} > 0$ if node i and node j are connected, otherwise $w_{ij} = 0$. $\|\cdot\|$ denotes the spectral norm of matrix (*i.e.*, largest singular value). Notice that λ is a connectivity measurement of the network graph and is also the second largest singular value of W. We do not assume W to be symmetric and hence the communication network can be both a directed graph and an undirected graph. The spectral gap assumption is also commonly used in the analysis of decentralized algorithms.

5.4.2 Main Theorems

Let $\varepsilon_H = \varepsilon^\alpha$. When $\alpha \le 0.5$, we have the following Theorem 5.1.

Theorem 5.1. *Assume $\alpha \le 0.5$ and Assumption 1 to 5 are satisfied. Let $\theta = \min\{\frac{3-5\alpha}{2}, 1\}$. We set $\eta = \Theta(\frac{\varepsilon^\theta}{L})$, $\beta = \Theta(\varepsilon^{1+\theta})$, $b_0 = \Theta(\varepsilon^{-2})$, $b_1 = \Theta(\max\{\varepsilon^{2-\theta-5\alpha}, 1\})$, $r = \Theta(\varepsilon^{1+\theta})$, $C_v = \Theta(\varepsilon)$, $C_T = \tilde{\Theta}(\varepsilon^{-\theta-\alpha})$ and $C_d = \tilde{\Theta}(\varepsilon^{1-\alpha})$. Then our PEDESTAL algorithm will achieve $O(\varepsilon, \varepsilon_H)$-second-order stationary point with $\tilde{O}(\varepsilon^{-3})$ gradient complexity.*

The specific constants hidden in $\Theta(\cdot)$ will be shown in Appendix D.2, where the proof outline and the completed proof of Theorem 5.1 can also be found. From Theorem 5.1 we can see that our PEDESTAL-S with $b_1 = O(1)$ can achieve $O(\varepsilon, \varepsilon_H)$-second-order stationary point with $\tilde{O}(\varepsilon^{-3})$ gradient complexity for $\varepsilon_H \ge \varepsilon^{0.2}$. In the classic setting, our PEDESTAL achieves second-order stationary point with $\tilde{O}(\varepsilon^{-3})$ gradient complexity. When $\alpha > 0.5$,



(a) $d = 50$, ring (b) $d = 50$, toroidal (c) $d = 50$, exponential

(d) $d = 100$, ring (e) $d = 100$, toroidal (f) $d = 100$, exponential

Figure 5.1: Experimental results of the decentralized matrix sensing task on different network topology for $d = 50$ and $d = 100$. Data is assigned to worker nodes by random distribution. The y-axis is the loss function value and the x-axis is the number of gradient oracles divided by the number of data N.

i.e., $\varepsilon_H < \sqrt{\varepsilon}$, we have the following Theorem 5.2. Since the parameter settings are different and the $O(1)$ batchsize is only available in Theorem 5.1, we separate these two theorems. The proof of Theorem 5.2 can be found in Appendix D.4.

Theorem 5.2. *When $\varepsilon_H < \sqrt{\varepsilon}$ (i.e., $\alpha > 0.5$), we set $\eta = \tilde{\Theta}(\varepsilon^{\theta})$, $\beta = \Theta(\varepsilon^{1+\theta})$, $b_0 = \Theta(\varepsilon^{-1})$, $b_1 = \tilde{\Theta}(\varepsilon^{-\max\{4\alpha-1-\theta,\theta+\alpha\}})$, $r = \Theta(\varepsilon^{1+\theta})$, $C_v = \Theta(\varepsilon)$, $C_T = \tilde{\Theta}(\varepsilon^{-\theta-\alpha})$ and $C_d = \tilde{\Theta}(\varepsilon^{\alpha})$ where $\theta = \min\{\frac{3\alpha-1}{2}, 3\alpha-2\}$. Under Assumption 5.1 to 5.5, our PEDESTAL algorithm will achieve $O(\varepsilon, \varepsilon_H)$-second-order stationary point with $\tilde{O}(\varepsilon\varepsilon_H^{-8} + \varepsilon^4\varepsilon_H^{-11})$ gradient complexity.*

5.5 Experiments

In this section we will demonstrate our experimental results to validate the performance of our method. We conduct two tasks in our experiment, a matrix sensing task on a synthetic

data set and a matrix factorization task on a real-world data set. Both of these two tasks are non-spurious local minimum problems ([29, 30]), which means that all local minima are global minima. Thus, we conclude that an algorithm is stuck at saddle point if the loss function value does not achieve the global minimum. The source code is available in `https://github.com/WH-XIAN/PEDESTAL`.

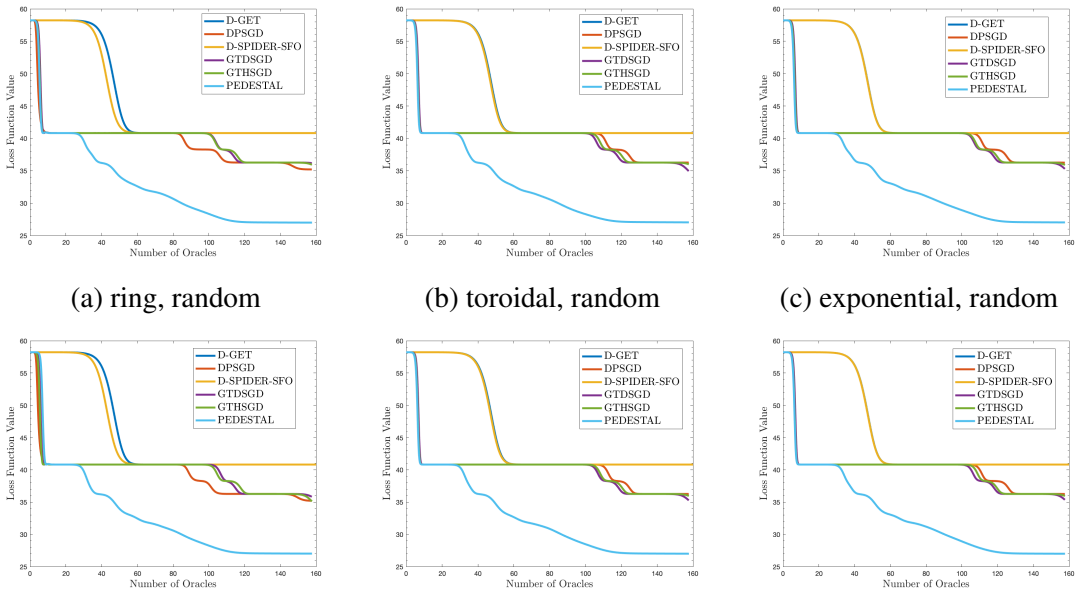
5.5.1 Matrix Sensing

We follow the experimental setup of ([15]) to solve a decentralized matrix sensing problem. The goal of this task is to recover a low-rank $d \times d$ symmetric matrix $M^* = U^*(U^*)^T$ where $U^* \in \mathbb{R}^{d\times r}$ for some small r. We set the number of worker nodes to $n = 20$. We generate a synthetic dataset with N sensing matrices $\{A_i\}_{i=1}^N$ and N corresponding observations $b_i = \langle A_i, M^* \rangle$. Here the inner product $\langle X, Y \rangle$ of two matrices X and Y is defined by the trace $tr(X^T Y)$. The decentralized optimization problem can be formulated by

$$\min_{U\in\mathbb{R}^{d\times r}} \sum_{i=1}^{n} L_i(U), \text{ where } L_i(U) = \frac{1}{2}\sum_{j=1}^{N_i} (\langle A_{ij}, UU^T\rangle - b_{ij})^2, \tag{5.3}$$

where N_i is the amount of data assigned to worker node i.

The number of rows in matrix U is set to $d = 50$ and $d = 100$, respectively, and the number of columns is set to $r = 3$. The ground truth low-rank matrix M^* equals $U^*(U^*)^T$ where each entry in U^* is generated independently by Gaussian distribution $\mathcal{N}(0, 1/d)$. We randomly generate $N = 20 \times n \times d$ samples of the sensing matrices $\{A_i\}_{i=1}^N$, $A_i \in \mathbb{R}^{d\times d}$ from the standard Gaussian distribution and calculate the corresponding labels $b_i = \langle A_i, M^* \rangle$. We consider two different types of data distribution, the random distribution and the Dirichlet distribution $Dir_{20}(0.3)$ to assign data to each worker node. We conduct experiments on three different types of network topology, *i.e.*, ring topology, toroidal topology (2-dimensional ring), and undirected exponential graph. The initial value of U is



(a) ring, random (b) toroidal, random (c) exponential, random

(d) ring, Dirichlet (e) toroidal, Dirichlet (f) exponential, Dirichlet

Figure 5.2: Experimental results of the decentralized matrix factorization task on different network topology on MovieLens-100k. The y-axis is the loss function value and the x-axis is the number of gradient oracles divided by the size of matrix $N \times l$.

set to $[u_0, \mathbf{0}, \mathbf{0}]$ where u_0 is the Gaussian distribution yield and multiplied by a scalar such that it satisfies $\|u_0\| \leq \max eig(M^*)$. We compare our PEDESTAL algorithm to decentralized baselines including D-PSGD, GTDSGD, D-GET, D-SPIDER-SFO and GTHSGD. In this experiment, the learning rate is chosen from $\{0.01, 0.001, 0.0001\}$. The batchsize is set to 10. For PEDESTAL and GTHSGD, the parameter β is set to 0.01. For D-GET and D-SPIDER-SFO, the period q is 100. For PEDESTAL, the threshold C_v is set to 0.0001. The perturbation radius r is set to 0.001. The threshold of the moving distance C_d is set to 0.01. The experimental results are shown in Figure 5.1. Due to space limitations, we only show the result of random data distribution in the main manuscript and leave the result of Dirichlet distribution to the Appendix D.1.

From the experimental result, we can see that all baselines are stuck at the saddle point and cannot escape it effectively. In contrast, our PEDESTAL reaches and escapes saddle points and finally finds the local minimum. We also calculate the smallest eigenvalue of

the Hessian matrix for each algorithm at the converged optimal point, which is left to the Supplementary Material because of space limit. According to the eigenvalue result, we can see that the smallest eigenvalue is much closer to 0 than all baselines. Therefore, our experiment verifies that our PEDESTAL achieves the best performance to escape the saddle point and find a local minimum.

5.5.2 Matrix Factorization

The second task in our experiment is matrix factorization, which aims to approximate a given matrix $M \in \mathbb{R}^{N\times l}$ by a low-rank matrix that can be decomposed to the product of two matrices $U \in \mathbb{R}^{N\times r}$ and $V \in \mathbb{R}^{l\times r}$ for some small r. The optimization problem can be formulated by

$$\min_{U\in\mathbb{R}^{N\times r}, V\in\mathbb{R}^{l\times r}} \|M - UV^T\|_F^2 := \sum_{i=1}^{N}\sum_{j=1}^{l}(M_{ij} - (UV^T)_{ij})^2 \tag{5.4}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and subscript ij refers to the element at i-th row and j-th column. In our experiment we solve this problem on the MovieLens-100k dataset ([35]). MovieLens-100k contains 100,000 ratings of 1682 movies provided by 943 users. Each rating is in the interval $[0,5]$ and scaled to $[0,1]$ in the experiment. This task can be regarded as an association task to predict users' potential ratings for unseen movies. In our experiment we set the number of worker node to $n = 50$. Each node is assigned the data from different group of users. Similar to the matrix sensing task, here we also use random distribution and Dirichlet distribution respectively to distribute users to worker nodes. And we also use ring topology, toroidal topology and undirected exponential graph as the communication network. The baselines are also D-PSGD, GTDSGD, D-GET, D-SPIDER-SFO and GTHSGD. In this experiment, the number of worker nodes is 50 and the rank of the matrix M is set to 25. The learning rate is chosen from $\{0.01, 0.001, 0.0001\}$. The batchsize is set to 100. For

PEDESTAL and GTHSGD, parameter β is set to 0.1. For D-GET and D-SPIDER-SFO, the period q is 100. For PEDESTAL, threshold C_v is set to 0.002. Perturbation radius r is set to 0.01. The threshold of moving distance C_d is set to 0.5. The experimental results are shown in Figure 5.2.

From the experimental results, we can see that our PEDESTAL achieves the best performance to escape saddle point and find second-order stationary point. All baselines cannot escape the saddle point effectively or efficiently. Particularly, variance-reduced methods D-GET and D-SPIDER-SFO show worse performance than SGD based algorithms D-PSGD and GTDSGD, which indicates that although reducing gradient noise can accelerate convergence, it also weakens the ability to escape saddle point. Therefore, our contribution is important since we make the fast convergence of variance reduction compatible with the capability to avoid the saddle point.

5.6 Conclusion

In this chapter, we propose a novel algorithm PEDESTAL to find local minima in nonconvex decentralized optimization. PEDESTAL is the first decentralized stochastic algorithm to achieve second-order optimality with non-asymptotic analysis. We improve the drawbacks in the previous deterministic counterpart to make phase change independently on each node and avoid consensus protocols of broadcast or aggregation. We prove that PEDESTAL can achieve $O(\varepsilon, \sqrt{\varepsilon})$-second-order stationary point with the gradient complexity of $\tilde{O}(\varepsilon^{-3})$, which matches the state-of-the-art results of centralized counterparts or the decentralized method to find the first-order stationary point. We also conduct the matrix sensing and matrix factorization tasks in our experiments to validate the performance of PEDESTAL.

Chapter 6: Generalized Smooth Minimax Optimization

6.1 Introduction

The minimax problem is attracting growing attention due to its widespread practical applications in machine learning such as Generative Adversarial Net (GAN) [32], adversarial training [78], robust optimization [13] and AUC maximization [27]. In minimax optimization, variable x aims to minimize a pay-off loss function $f(x,y):\mathbb{R}^{d_1}\times\mathbb{R}^{d_2}\to\mathbb{R}$ while variable y tries to maximize the loss, which can be formulated as

$$\min_{x\in\mathbb{R}^{d_1}}\max_{y\in\mathscr{Y}} f(x,y), \tag{6.1}$$

where $\mathscr{Y}\subseteq\mathbb{R}^{d_2}$ is a convex domain. In this chapter we consider the nonconvex strongly-concave problem where $f(x,y)$ in nonconvex in x and strongly-concave in y. In this case, the maximizer $y^*(x)=\arg\max_{y\in\mathscr{Y}} f(x,y)$ is unique and the primal objective function $\Phi(x)=f(x,y^*(x))$ can be well defined. The convergence criterion is to find a first-order stationary point of $\Phi(x)$ such that $\|\nabla\Phi(x)\|\leq\varepsilon$ for some tolerance ε. When considering stochastic problems, function $f(x,y)$ takes the form $f(x,y)=\mathbb{E}_{\xi\sim D}F(x,y;\xi)$ where $F(x,y;\xi)$ is the component loss function regarding sample ξ and D is the data distribution.

In recent years, the minimax optimization problem has been studied in a variety of research fields. Many deterministic and stochastic gradient-based methods with non-asymptotic convergence analysis for nonconvex strongly-concave minimax problems have

been developed. Among these methods, some algorithms adopt the single-loop structure that updates x and y at the same frequency, such as Gradient Descent Ascent (GDA) and Stochastic Gradient Descent Ascent (SGDA) [66]. Some algorithms update x and y at different frequencies, which involves a nested loop to search the optimal value of the maximizer y for the given x. Classic examples of double-loop minimax algorithms are GDmax and SGDmax [46]. Some methods adopt more sophisticated structures to achieve better theoretical results [67, 132]. In addition, some works also investigate the estimation of the lower bound of minimax problems [58, 136] and some algorithms have been proven to be optimal or near-optimal [67].

Although gradient-based minimax optimization algorithms have achieved huge success in the theoretical region, most of the analysis frameworks are based on the requirement of Lipschitz smoothness. Some works conduct the convergence analysis without the Lipschitz smooth assumption for convex or weakly-convex problems [91] and achieve competitive results, but the investigation for nonconvex generalized smooth minimax optimization is still limited. This drawback will restrict the applications of minimax optimization algorithms because in some cases the minimax structure breaks the Lipschitz smooth condition, such as distributionally robust optimization [47, 56, 131], and in some machine learning tasks the objective function itself does not satisfy the Lipschitz smoothness, such as phase retrieval [23, 82]. Counterexamples will be demonstrated in Section 6.2 to illustrate the divergence issue. Therefore, to fill this gap, we are motivated to investigate the convergence analysis of minimax algorithms under the relaxation of Lipschitz smooth assumption so that these algorithms can theoretically be guaranteed to work for a wider range of applications.

We summarize our contribution as follows.

- In this chapter we study the convergence analysis of minimax optimization algorithms without the assumption of Lipschitz smoothness. We provide some counterexamples

to reveal the divergence issue and propose the strategy to solve this problem.

- We prove that generalizations of classic minimax optimization algorithms (including single-loop algorithms GDA, SGDA, and double-loop algorithms GDmax, SGDmax) can still converge under the generalized smooth condition and the gradient complexity matches the Lipschitz smooth counterparts. We conduct a numerical experiment of robust logistic regression task to validate the practical performance of our method.

6.2 Preliminary

6.2.1 Minimax Optimization Algorithms

In recent years, many algorithms were proposed to solve the optimization of minimax, and many of them were studied under the nonconvex-strongly-concave condition. GDmax and its stochastic variant SGDmax [46] are representatives of double-loop minimax algorithms. In each iteration, they compute the estimation of the maximizer $y_{t+1} \approx y^*(x_t)$ via a nested loop and then update $x_{t+1} = x_t - \eta_x \nabla_x f(x_t, y_{t+1})$. GDmax can reach a first-order stationary point with $O(\kappa^2 \varepsilon^{-2} \log(1/\varepsilon))$ iterations, where $\kappa = L/\mu$ is the condition number, L is the Lipschitz constant and μ is the strong concavity constant. SGDmax achieves the stochastic first-order oracle (SFO) complexity of $O(\kappa^3 \varepsilon^{-4} \log(1/\varepsilon))$ to achieve a first-order stationary point. GDA and its stochastic variant SGDA [66] are representatives of single-loop minimax algorithms. In each iteration, they compute the partial derivatives with respect to x and y, respectively. Then the variables x and y are updated by $x_{t+1} = x_t - \eta_x \nabla_x f(x_t, y_t)$ and $y_{t+1} = y_t + \eta_y \nabla_y f(x_t, y_t)$. GDA reaches a first-order stationary point with $O(\kappa^2 \varepsilon^{-2})$ iterations, SGDA achieves the SFO complexity of $O(\kappa^3 \varepsilon^{-4})$ to achieve a first-order stationary point. These algorithms are fundamental optimizers to solve minimax optimization problem and hence we will conduct convergence analysis based on these algorithms. More

recently, some algorithms have been proposed to accelerate the convergence rate and reduce the gradient complexity of minimax optimization by variance reduction, such as SREDA ([77]) and Acc-MDA ([40]). Moreover, in the deterministic setting, some recently proposed algorithms ([67]) have already matched the optimal lower bound ([136]).

6.2.2 Counterexamples in Minimax Problems

In this section we will provide some counterexamples to illustrate the non-Lipschitz smoothness and divergence issue in minimax optimization. First, we will introduce some basic definitions about Lipschitz smoothness.

Definition 6.1. *A real-value function f is Lipschitz smooth if there exists a constant L such that*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \tag{6.2}$$

Definition 6.2. *A real-value function f is Lipschitz continuous if there exists a constant M such that*

$$\|f(x) - f(y)\| \leq M\|x - y\| \tag{6.3}$$

Example 1. We will take distributionally robust optimization as our first example, which is a classic application of minimax optimization. Distributionally robust optimization aims to make the training result of the original optimization problem more robust by introducing a perturbation and solving a minimax problem. In [131], an example of this task is formulated

as

$$\min_{x} \max_{y \in \Delta_n} f(x,y) = \sum_{i=1}^{n} y_i l_i(x) - V(y) \tag{6.4}$$

where n is the number of samples and $l_i(x)$ is the original loss function. Δ_n is the simplex in the n-dimensional Euclidean space and $V(y)$ denotes a divergence measure between two distributions, which could be chosen as $\sum_{i=1}^{n}(y_i - \frac{1}{n})^2$. In this case, we can see that problem (6.4) is a nonconvex-strongly-concave minimax problem. We assume that the original loss functions $l_i(x)$ are Lipschitz smooth but not Lipschitz continuous. Then we have

$$\|\nabla_y f(x,y) - \nabla_y f(x',y)\|^2 = \sum_{i=1}^{n} (l_i(x) - l_i(x'))^2 \tag{6.5}$$

If function f is Lipschitz smooth, we should have

$$\|\nabla_y f(x,y) - \nabla_y f(x',y)\|^2 \leq L^2 \|x - x'\|^2 \tag{6.6}$$

which implies each $l_i(x)$ is Lipschitz continuous and conflicts with our assumption. Hence, the objective function f is not Lipschitz smooth, even the original loss functions l_i are Lipschitz smooth, which shows that the minimax structure can probably break the condition of Lipschitz smoothness.

The convergence analysis of most current existing minimax algorithms is based on the Lipschitz smoothness assumption. However, this condition is not satisfied in many classic examples, such as robust optimization. This result motivates us to study the convergence of minimax algorithms without the requirement of Lipschitz smoothness.

Example 2. Next, we will provide a simple example to reveal the divergence issue when

Lipschitz smoothness is not satisfied. We define a minimax problem

$$\min_x \max_y f(x,y) = yx^2 - 0.5y^2 \tag{6.7}$$

where x and y are scalars. It is easy to check $y^*(x) = x^2$ and $\Phi(x) = 0.5x^4$. Thus, we have $\nabla\Phi(x) = 2x^3$. For any fixed stepsize $\eta > 0$, if we choose the initial value $x_0 \geq \frac{2}{\sqrt{\eta}}$ and apply a gradient descent algorithm, then we can prove $|\eta\nabla\Phi(x_t)| \geq |x_t|$ and $|x_{t+1}| \geq 2|x_t|$ for all $t \geq 0$. This implies that $|x_t| \geq 2^t|x_0|$ and the algorithm will diverge. In this chapter, we will discuss some generalized minimax algorithms to tackle the divergence issue.

6.2.3 Generalized Smoothness

Previous works studying nonconvex nonsmooth minimax optimization can be categorized into following branches. Some minimax algorithms adopt the zeroth-order strategy [40, 71, 118] to address the issue where the objective function is not differentiable or the gradient cannot be accessed. However, if the objective function is still differentiable, just not Lipschitz smooth, gradient-based methods are more efficient and effective than gradient-free methods. Some other works focus on nonconvex nonsmooth minimax problems with certain special structures. As an example, [42] considers the problem that is a nonconvex Lipschitz smooth loss function adding a convex nonsmooth regularization, which can be solved by proximal gradient. [59] considers a nonsmooth composite minimax problem where $f(\cdot, y)$ is the composition of a Lipschitz smooth function and Lipschitz continuous function. In this chapter, we do not assume any specific structures for the objective function.

In a concurrent work [34], the convergence analysis of a bilevel optimization algorithm is provided under the condition of unbounded smoothness, which is also applicable to minimax optimization. In [34], the lower level function that is used to calculate $y^*(x)$ is assumed to be Lipschitz smooth and the upper level function is assumed to be (L_0, L_1)-smooth [134],

which is defined as follows:

Definition 6.3. *A real-value function f is (L_0, L_1)-smooth if there exist constants L_0 and L_1 such that*

$$\|\nabla f(x) - \nabla f(y)\| \leq (L_0 + L_1 \|\nabla f(x)\|)\|x - y\| \tag{6.8}$$

We can see Lipschitz smoothness is a special case of (L_0, L_1)-smoothness where $L_1 = 0$. Recently, a variety of works have been proposed to study and generalize the requirement of Lipschitz smoothness [16, 57]. In [57], the definition of l-smoothness is proposed as follows:

Definition 6.4. *A real-value function f is l-smooth if there exists a non-decreasing continuous function $l(\cdot)$ such that*

$$\|\nabla f(x_1) - \nabla f(x_2)\| \leq l(\|\nabla f(x)\| + G) \cdot \|x_1 - x_2\| \tag{6.9}$$

for any x_1 and x_2 in $\mathscr{B}(x, \frac{G}{l(\|\nabla f(x)\|+G)})$ for any $G > 0$.

In [57], it is proven that the definition 6.9 is equivalent to $\|\nabla^2 f(x)\| \leq l(\|\nabla f(x)\|)$ almost everywhere. For nonconvex optimization problems, function l is required to be sub-quadratic but (L_0, L_1)-smoothness still can be regarded as a special case of l-smoothness where $l(u) = L_0 + L_1 u$. A common example of sub-quadratic function is $l(u) = L_0 + L_\rho u^\rho$ where $0 < \rho < 2$, which contain the case of $\rho = 1$. In this chapter, we extend the concept of l-smoothness to minimax optimization and propose the definition of l_x-l_y-smoothness in Definition 6.5. Therefore, the smoothness condition used in this chapter is more general than the assumption used in [34].

Definition 6.5. *A real-value function $f(x, y) : \mathbb{R}^{d_1} \times \mathscr{Y} \to \mathbb{R}$ is called l_x-l_y-smooth for non-decreasing continuous functions l_x and l_y if we have*

Algorithm 7 Generalized GDA or SGDA

Input: initial value x_0 and y_0

Parameter: learning rate η and η_y, maximum iteration T.

for $t = 0, 1, \ldots, T-1$ **do**
  Compute $v_t = \nabla_x f(x_t, y_t)$ (deterministic)
    or $v_t = \nabla_x F(x_t, y_t; \xi_t)$ (stochastic).
  Compute $u_t = \nabla_y f(x_t, y_t)$ (deterministic)
    or $u_t = \nabla_y F(x_t, y_t; \xi_t)$ (stochastic).
  Compute suitable stepsize parameter S_t.
  Update $x_{t+1} = x_t - (\eta / S_t) v_t$.
  Update $y_{t+1} = \Pi_{\mathscr{Y}}(y_t + \eta_y u_t)$.
end for

Algorithm 8 Generalized GDmax or SGDmax

Input: initial value x_0 and y_0

Parameter: learning rate η and η_y, nested loop size K, maximum iteration T.

for $t = 0, 1, \ldots, T-1$ **do**
  Compute $v_t = \nabla_x f(x_t, y_t)$ (deterministic)
    or $v_t = \nabla_x F(x_t, y_t; \xi_t)$ (stochastic).
  Compute suitable stepsize parameter S_t.
  Update $x_{t+1} = x_t - (\eta / S_t) v_t$.
  Let $y_{t,0} = y_t$.
  for $k = 0, 1, \ldots, K-1$ **do**
    Compute $u_{t,k} = \nabla_y f(x_{t+1}, y_{t,k})$ (deterministic)
      or $u_{t,k} = \nabla_y F(x_{t+1}, y_{t,k}; \xi_{t,k})$ (stochastic).
    Update $y_{t,k+1} = \Pi_{\mathscr{Y}}(y_{t,k} + \eta_y u_{t,k})$.
  end for
  Update $y_{t+1} = y_{t,K}$.
end for

$$\|\nabla_x f(z_1) - \nabla_x f(z_2)\| \le l_x(\|\nabla_x f(z_0)\| + G_1) \cdot \|z_1 - z_2\|$$

$$\|\nabla_y f(z_1) - \nabla_y f(z_2)\| \le l_y(\|\nabla_y f(z_0)\| + G_2) \cdot \|z_1 - z_2\|$$

for any z_1 *and* z_2 *in* $\mathscr{B}(z_0, r(z_0))$ *and any* $z_0 = [x_0; y_0]$, *where* $r(z_0) = \frac{G_1}{l_x(\|\nabla_x f(z_0)\| + G_1)} + \frac{G_2}{l_y(\|\nabla_y f(z_0)\| + G_2)}$ *for any given* $G_1 > 0$ *and* $G_2 > 0$.

Moreover, it can be proven that the two counterexamples belong to the category of our l_x-l_y-smoothness.

6.3 Algorithms

As revealed in our counterexample, vanilla gradient based algorithm fails to converge in minimax optimization when the Lipschitz smooth assumption does not hold. The reason for the divergence is due to the large gradient. Therefore, we will generalize these algorithms to tackle this issue by adopting a suitable stepsize strategy to control the moving distance in each iteration. We will apply this strategy to standard minimax optimizers GDA, SGDA, GDmax and SGDmax. The description of single-loop algorithms Generalized GDA (or SGDA) is shown in Algorithm 7. The description of double-loop algorithms Generalized GDmax (or SGDmax) is shown in Algorithm 8.

Let x_0 and y_0 be the initial values in Algorithm 7 and Algorithm 8. In our convergence analysis, we need to run an additional initialization process to obtain an approximation of the maximizer $y_0 \approx y^*(x_0)$ for the given initial value x_0 before the algorithms start. The specific conditions that y_0 needs to satisfy will also be discussed in the convergence analysis. This subproblem can be converted to a strongly-convex generalized Lipschitz smooth minimization problem and solved by optimizers such as GD, SGD or SPIDER [16, 26, 57].

In Algorithm 7, we adopt a suitable stepsize based on the norm of gradient to single-loop minimax algorithms GDA and SGDA. In each iteration, we compute the gradients $\nabla_x f(x_t, y_t)$, $\nabla_y f(x_t, y_t)$ or the corresponding stochastic gradients with respect to x and y, respectively. Then we update x_t and y_t by gradient descent ascent. When we update x_t, we adopt the suitable stepsize strategy to control the moving distance. We have multiple options to compute the suitable stepsize parameter S_t. It could be:

(1) $S_t = \|v_t\|$. (2) $S_t = \max\{\varepsilon, \frac{1}{t+1}\sum_{\tau=0}^{t}\|v_t\|\}$.

(3) $S_t \equiv S$. (4) $S_t = \max\{\varepsilon, (1-\beta)\|v_t\| + \beta S_{t-1}\}$.

When we choose option (1), the suitable stepsize strategy is turned out to be the gradient normalization method. When we choose option (2), we calculate the average of the historical gradient norm. When we choose option (3), the suitable stepsize will be a constant. Notice that it is different from the conventional constant stepsize because S probably has dependence on the initial value, and it is calculated after the algorithm starts. When we choose option (4), we calculate the exponential average of the historical gradient norm. When we update y_t, we adopt a constant stepsize such that $y_{t+1} = \Pi_{\mathscr{Y}}(y_t + \eta_y u_t)$. with a projection onto $\mathscr{Y}$.

In Algorithm 8, we apply the suitable stepsize strategy to double-loop minimax algorithms GDmax and SGDmax. In each iteration, we first calculate the gradient $\nabla_x f(x_t, y_t)$ with respect to x (or the corresponding stochastic gradient). We update $x_{t+1} = x_t - (\eta/S_t)v_t$ by an suitable stepsize η/S_t, where the options to compute S_t are the same as Algorithm 7. Then we run a nested loop to find an estimate of the maximizer $y_{t+1} \approx y^*(x_{t+1})$. Specifically, we apply an iterative gradient ascent algorithm $y_{t,k+1} = \Pi_{\mathscr{Y}}(y_{t,k} + \eta_y u_{t,k})$ where $u_{t,k}$ is the deterministic or stochastic gradient estimator to solve the maximization subproblem $\max_y f(x_{t+1}, y)$.

6.4 Convergence Analysis

6.4.1 Main Theorems

In this section, we will show the main theorems of our convergence analysis. The theoretical results indicate that our generalized GDA, SGDA, GDmax or SGDmax algorithms can converge under the generalized Lipschitz smooth condition and the gradient complexities to reach first-order stationary point are the same as Lipschitz smooth counterparts. First, we will introduce the following assumptions.

Assumption 6.1. *The primal function* Φ *is lower bounded, i.e.,* $\inf_x \Phi(x) = \Phi^* > -\infty$.

Assumption 6.2. *The loss function $f(x,y)$ is μ-strongly-concave w.r.t. y, i.e., there exists a constant $\mu > 0$ such that for any x, y and y', we have*

$$f(x,y) \leq f(x,y') + \langle \nabla_y f(x,y'), y - y' \rangle - \frac{\mu}{2}\|y - y'\|^2$$

Assumption 6.3. *The loss function $f(x,y)$ is l_x-l_y-smooth and function l_x is sub-quadratic.*

These assumptions are basic prerequisites for the convergence analysis of nonconvex strongly-concave minimax optimization. In nonconvex minimization problems [57], the function l is also required to be sub-quadratic.

We perform our convergence analysis based on two cases. The first case is $\mathscr{Y} = \mathbb{R}^{d_2}$, which results in an unconstrained optimization with respect to y. The second case is that $\mathscr{Y}$ is bounded, which implies f is Lipschitz smooth with respect to y, *i.e.*, there exists a constant L_y such that $l_y(\cdot) \equiv L_y$. We need these requirements because otherwise the value of $l_y(\|\nabla_y f(x, y^*(x))\|)$ is hard to estimate, which can lead to poor smoothness even approaching the maximizer y^*.

We provide the following essential definitions of notations that are frequently used in our analysis.

$$\begin{aligned} G_x &= \max\{u > 0 | u^2 \leq 8\kappa l_x(2u) \cdot (\Phi(x_0) - \Phi^*)\} \\ G_y &= \nabla_y f(x_0, y_0),\ y_t^* = y^*(x_t) \end{aligned} \tag{6.10}$$

Analysis Results of GDA

Similar to Lipschitz smooth minimax problems, we can define the condition number as $\kappa = l_y(4G_y)/\mu$. With Assumption 6.1 to 6.3, we can obtain the following Theorem for the generalized GDA algorithm.

Theorem 6.1. *Assume Assumptions 6.1, 6.2 and 6.3 are satisfied. Let the parameters* $\frac{\eta}{S_t} \leq \frac{C_0}{16\kappa^2 l_x(2G_x)}$ *for all* t, $\eta_y = \frac{1}{l_y(2G_y)}$ *and the initial value* $\|y_0 - y_0^*\| \leq \frac{C_0 G_x}{l_x(2G_x)}$ *where constant* $C_0 = \min\{1, \frac{l_x(2G_x)G_y}{l_y(0)G_x}\}$. *Then for the generalized GDA algorithm, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1}\frac{\|\nabla\Phi(x_t)\|^2}{S_t} \leq \frac{5(\Phi(x_0)-\Phi^*)}{\eta T} \tag{6.11}$$

When S_t is constant (option (3)), we can achieve the following Corollary 6.1 for generalized GDA, which indicates that under the condition of generalized Lipschitz smoothness, our generalized GDA algorithm can achieve the same gradient complexity to find the first-order stationary point as GDA does with Lipschitz smoothness.

Corollary 6.1. *When* S_t *is computed by option (3), let* $\eta = O(\frac{1}{\kappa^2 l_x(2G_x)})$, $\eta_y = O(\frac{1}{l_y(2G_y)})$, $T = O(\kappa^2\varepsilon^{-2})$ *and other conditions be the same as Theorem 6.1. Then the generalized GDA algorithm can find an* ε*-first-order stationary point with* $O(\kappa^2\varepsilon^{-2})$ *gradient oracles.*

When we choose other options to compute S_t, we can obtain the following theoretical results.

Corollary 6.2. *When* S_t *is computed by option (1) or (4), let* $\eta = O(\frac{\varepsilon}{\kappa^2 l_x(2G_x)})$, $\eta_y = O(\frac{1}{l_y(2G_y)})$, $T = O(\kappa^2\varepsilon^{-2})$ *and other conditions be the same as Theorem 6.1. Then the generalized GDA algorithm can find an* ε*-first-order stationary point with* $O(\kappa^2\varepsilon^{-2})$ *gradient oracles.*

Corollary 6.3. *When* S_t *is computed by option (2), let* $\eta = O(\frac{\varepsilon}{\kappa^2 l_x(2G_x)})$, $\eta_y = O(\frac{1}{l_y(2G_y)})$, $T = O(\kappa^2\varepsilon^{-2}\log(\frac{1}{\varepsilon}))$ *and other conditions be the same as Theorem 6.1. Then the generalized GDA algorithm can find an* ε*-first-order stationary point with* $O(\kappa^2\varepsilon^{-2}\log(\frac{1}{\varepsilon}))$ *gradient oracles.*

We can see that the complexity of the gradient oracle complexity to achieve first-order stationary points is the same as that of the GDA when the suitable stepsize parameter S_t is

computed by the gradient norm or the exponential moving average of the historical gradient norm. When S_t is computed using the averaged historical gradient norm, there will be an additional logarithmic term. However, our analysis is conducted under the condition of generalized Lipschitz smoothness, while the original analysis of GDA is based on Lipschitz smoothness.

Analysis Results of GDmax

For double-loop deterministic algorithm Generalized GDmax, we have the following Theorem 6.2.

Theorem 6.2. *Assume Assumptions 6.1, 6.2 and 6.3 are satisfied. Let parameters* $\frac{\eta}{S_t} \leq \frac{C_0}{16\kappa l_x(2G_x)}$ *for all t,* $\eta_y = \frac{1}{l_y(4G_y)}$, $K \geq \kappa\log(\frac{1}{\theta})$ *and initial value* $\|y_0 - y_0^*\| \leq \frac{C_0 G_x}{l_x(2G_x)}$ *where constant* $C_0 = \min\{1, \frac{l_x(2G_x)G_y}{l_y(2G_y)G_x}\}$ *and* $\theta = \min\{\frac{1}{4}, \frac{1}{l_x^2(2G_x)}\}$. *Then for the generalized GDmax algorithm, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1}\frac{\|\nabla\Phi(x_t)\|^2}{S_t} \leq \frac{5(\Phi(x_0)-\Phi^*)}{\eta T} \tag{6.12}$$

Similar to generalized GDA, we can prove under the condition of generalized Lipschitz smoothness, GDmax algorithm can achieve the same gradient complexity to find first-order stationary point as GDmax does with Lipschitz smoothness.

Corollary 6.4. *When* S_t *is computed by option (3), let* $\eta = O(\frac{1}{\kappa l_x(2G_x)})$, $\eta_y = O(\frac{1}{l_y(4G_y)})$, $K = O(\kappa)$, $T = O(\kappa\varepsilon^{-2})$ *and other conditions be the same as Theorem 6.2. Then the generalized GDmax algorithm can find an* ε*-first-order stationary point with* $O(\kappa^2\varepsilon^{-2})$ *gradient oracles.*

Corollary 6.5. *When* S_t *is computed by option (1) or (4), let* $\eta = O(\frac{\varepsilon}{\kappa l_x(2G_x)})$, $\eta_y = O(\frac{1}{l_y(4G_y)})$, $K = O(\kappa)$, $T = O(\kappa\varepsilon^{-2})$ *and other conditions be the same as Theorem 6.2.*

Then the generalized GDmax algorithm can find an ε-first-order stationary point with $O(\kappa^2\varepsilon^{-2})$ gradient oracles.

Corollary 6.6. *When S_t is computed by option (2), let $\eta = O(\frac{\varepsilon}{\kappa l_x(2G_x)})$, $\eta_y = O(\frac{1}{l_y(4G_y)})$, $K = O(\kappa)$, $T = O(\kappa\varepsilon^{-2}\log(\frac{1}{\varepsilon}))$ and other conditions are the same as Theorem 6.2. Then Generalized GDmax algorithm can find an ε-first-order stationary point with $O(\kappa^2\varepsilon^{-2}\log(\frac{1}{\varepsilon}))$ gradient oracles.*

Analysis Results of SGDA

For generalized stochastic algorithms SGDA and SGDmax, we assume the stochastic gradient oracle is unbiased, *i.e.*, $\mathbb{E}_\xi \nabla F(x,y;\xi) = \nabla f(x,y)$. We also need the following bounded variance assumption, which is a common assumption in the convergence analysis of stochastic gradient-based optimization algorithms.

Assumption 6.4. *The stochastic gradient oracle satisfies $\mathbb{E}_\xi \|\nabla F(x,y;\xi) - \nabla f(x,y)\|^2 \leq \sigma^2$ for some constant σ.*

In stochastic algorithms, let b_x and b_y denote the batchsize of stochastic gradient with respect to x and y, respectively. Due to the noise of stochastic gradient, there is no guarantee for the upper bound of gradient or function value. Thus, we cannot apply mathematical induction to estimate the upper bound along the trajectory, as we do in the deterministic case (see the sketch of the proof in the next subsection). However, we can still prove that generalized SGDA and SGDmax will converge with a high probability. In the stochastic case, we need to redefine the constant.

$$G_x = \max\{u > 0 | u^2 \leq 32\kappa l_x(2u)\cdot(\Phi(x_0) - \Phi^* + \sigma^2)/\delta\}$$

For Generalized SGDA, we have the following Theorem.

Theorem 6.3. *Assume Assumptions 6.1, 6.2, 6.3 and 6.4 are satisfied. Let the parameters* $\frac{\eta}{S_t} \leq \frac{\delta C_0}{48\kappa^2 l_x(2G_x)}$ *for all* t, $\eta_y = \frac{1}{l_y(2G_y)}$, $T = \frac{\kappa^2}{\delta^2\varepsilon^2}$, $b_x \geq \frac{\sigma^2}{G_x^2\varepsilon^2}$, $b_y \geq \max\{\frac{192\kappa\sigma^2 l_x^2(2G_x)}{\delta G_x^2 l_y^2(2G_y)}, \frac{\kappa l_x(2G_x)}{\delta^2 l_y^2(2G_y)\varepsilon^2}\}$ *and the initial value* $\|y_0 - y_0^*\| \leq \frac{\delta C_0 G_x}{8 l_x(2G_x)}$ *where constant* $C_0 = \min\{1, \frac{l_x(2G_x)G_y}{l_y(2G_y)G_x}\}$. *Then for the generalized SGDA algorithm, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1}\frac{\|\nabla\Phi(x_t)\|^2}{S_t} \leq \frac{10(\Phi(x_0) - \Phi^* + \sigma^2)}{\delta\eta T} \tag{6.13}$$

with probability at least $1-\delta$.

When S_t is constant ($S_t \equiv S$), we can obtain the following Corollary for Generalized SGDA, which results in the same stochastic first-order oracle complexity under the condition of relaxed Lipschitz smoothness as SGDA does with the requirement of Lipschitz smoothness.

Corollary 6.7. *When* S_t *is computed by option (3), let* $\eta = O(\frac{1}{\kappa^2 l_x(2G_x)})$, $\eta_y = O(\frac{1}{l_y(2G_y)})$, $b_x = O(\varepsilon^{-2})$, $b_y = O(\kappa\varepsilon^{-2})$, $T = O(\kappa^2\varepsilon^{-2})$ *and other conditions are the same as Theorem 6.3. Then the generalized SGDA algorithm can find an* ε*-first-order stationary point with SFO of* $O(\kappa^3\varepsilon^{-4})$.

When S_t is computed by option (1) or (4), we can reach the following conclusion which also achieves the same SFO complexity as SGDA does in the Lipschitz smooth case.

Corollary 6.8. *When* S_t *is computed by option (1) or (4), let* $\eta = O(\frac{\varepsilon}{\kappa^2 l_x(2G_x)})$, $\eta_y = O(\frac{1}{l_y(2G_y)})$, $b_x = O(\varepsilon^{-2})$, $b_y = O(\kappa\varepsilon^{-2})$, $T = O(\kappa^2\varepsilon^{-2})$ *and other conditions are the same as Theorem 6.3. Then the generalized SGDA algorithm can find an* ε*-first-order stationary point with SFO of* $O(\kappa^3\varepsilon^{-4})$.

When S_t is computed by option (4), we can obtain the following theoretical result, which causes an additional logarithm term in the SFO complexity.

Corollary 6.9. *When S_t is computed by option (2), let $\eta = O(\frac{\varepsilon}{\kappa^2 l_x(2G_x)})$, $\eta_y = O(\frac{1}{l_y(2G_y)})$, $b_x = O(\varepsilon^{-2})$, $b_y = O(\kappa\varepsilon^{-2})$, $T = O(\kappa^2\varepsilon^{-2}\log(\frac{1}{\varepsilon}))$ and other conditions are the same as Theorem 6.3. Then the generalized SGDA algorithm can find an ε-first-order stationary point with SFO of $O(\kappa^3\varepsilon^{-4}\log(\frac{1}{\varepsilon}))$.*

Name	a9a	covtype	diabetes	german	gisette	ijcnn1	mushrooms	phishing	w8a
Samples	32561	581012	768	1000	6000	141691	8124	11055	49749
Features	123	54	8	24	5000	22	112	68	300

Table 6.1: Descriptions of the LIBSVM binary classification datasets used in our experiment

Analysis Results of SGDmax

For the stochastic double-loop algorithm Generalized SGDmax, we have the following conclusions.

Theorem 6.4. *Assume Assumption 6.1, 6.2, 6.3 and 6.4 are satisfied. Let parameters $\frac{\eta}{S_t} \leq \frac{\delta C_0}{48\kappa l_x(2G_x)}$ for all t, $\eta_y = \frac{1}{l_y(4G_y)}$, $K \geq \kappa\log(\frac{1}{\theta})$, $T = \frac{\kappa}{\delta^2\varepsilon^2}$, $b_x \geq \frac{\sigma^2}{G_x^2\varepsilon^2}$, $b_y \geq \max\{\frac{24\kappa\sigma^2 l_x^2(2G_x)}{\delta G_x^2 l_y^2(4G_y)}, \frac{\kappa l_x(2G_x)}{\delta^2 l_y^2(4G_y)\varepsilon^2}\}$ and initial value $\|y_0 - y_0^*\| \leq \frac{\delta C_0 G_x}{8 l_x(2G_x)}$ where constant $C_0 = \min\{1, \frac{l_x(2G_x)G_y}{l_y(2G_y)G_x}\}$ and $\theta = \min\{\frac{1}{4}, \frac{1}{l_x^2(2G_x)}\}$. Then for the generalized SGDmax algorithm, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1}\frac{\|\nabla\Phi(x_t)\|^2}{S_t} \leq \frac{10(\Phi(x_0) - \Phi^* + \sigma^2)}{\delta\eta T} \tag{6.14}$$

with probability at least $1-\delta$.

Corollary 6.10. *When S_t is computed by option (3), let $\eta = O(\frac{1}{\kappa l_x(2G_x)})$, $\eta_y = O(\frac{1}{l_y(4G_y)})$, $K = O(\kappa)$, $T = O(\kappa\varepsilon^{-2})$ and other conditions are the same as Theorem 6.4. Then the generalized SGDmax algorithm can find an ε-first-order stationary point with SFO of $O(\kappa^3\varepsilon^{-4})$.*

Corollary 6.11. *When S_t is computed by option (1) or (4), let $\eta = O(\frac{\varepsilon}{\kappa l_x(2G_x)})$, $\eta_y = O(\frac{1}{l_y(4G_y)})$, $K = O(\kappa)$, $T = O(\kappa\varepsilon^{-2})$ and other conditions be the same as Theorem 6.4. Then the generalized SGDmax algorithm can find an ε-first-order stationary point with SFO of $O(\kappa^3\varepsilon^{-4})$.*

Corollary 6.12. *When S_t is computed by option (2), let $\eta = O(\frac{\varepsilon}{\kappa l_x(2G_x)})$, $\eta_y = O(\frac{1}{l_y(4G_y)})$, $K = O(\kappa)$, $T = O(\kappa\varepsilon^{-2}\log(\frac{1}{\varepsilon}))$ and other conditions be the same as Theorem 6.4. Then the generalized SGDmax algorithm can find an ε-first-order stationary point with SFO of $O(\kappa^3\varepsilon^{-4}\log(\frac{1}{\varepsilon}))$.*

These theoretical results indicate that, under the generalized Lipschitz smooth condition, our generalized SGDmax method can still converge and achieve the same SFO complexity as SGDmax does in the Lipschitz smooth case.

6.4.2 Sketch of Proof

In this subsection, we will provide the outline of our proof to illustrate the insight of our analysis. The completed proof is left to the Appendix. Due to the space limit, we will only demonstrate the sketch of the proof for the generalized GDA and SGDA algorithms. First, we can prove the smoothness for the functions $y^*(x)$ and $\Phi(x)$ (described in Lemma E.1 and Lemma E.2) such that $\|y^*(x) - y^*(x')\| \leq \kappa\|x - x'\|$ and

$$\|\nabla\Phi(x) - \nabla\Phi(x')\| \leq 2\kappa l_x(\|\nabla\Phi(x)\| + G) \cdot \|x - x'\|$$

if $\|x' - x\| \leq \frac{G}{l_x(\|\nabla_x f(x, y^*(x))\| + G)}$ for some $G \geq 0$. Then we can obtain Lemma E.3, which indicates that

$$\|\nabla\Phi(x)\|^2 \leq 4\kappa l_x(2\|\nabla\Phi(x)\|) \cdot (\Phi(x) - \Phi^*) \tag{6.15}$$

for $\forall x$. When function l_x is sub-quadratic, Eq. (6.15) provides an upper bound for $\|\nabla\Phi(x)\|$ that is dependent on the function value gap $(\Phi(x)-\Phi^*)$.

Next, we want to prove that $\|\nabla\Phi(x_t)\|\leq G_x$ for all $t\geq 0$ in Generalized GDA, which means the gradient is bounded along the trajectory x_t. With this conclusion, the values of $l_x(\cdot)$ that occur along the trajectory in the analysis can be bounded by $l_x(2G_x)$, and hence the rest of the proof will be simplified and relatively easy. In minimization optimization, this conclusion can be achieved directly by mathematical induction. However, in mimimax optimization, the exact value of $\nabla\Phi(x)$ is not available. It is estimated by $\nabla_x f(x_t,y_t)$, which yields an error term caused by $\|y_t-y_t^*\|$. The original proof framework of the minimization problem does not work in this case due to the existence of the error term. Besides, the error term will lead to an additional term that also has a dependence on G_x when bounding the function value gap $(\Phi(x)-\Phi^*)$. To solve this issue, we need to apply mathematical induction to $\nabla\Phi(x_t)$, $\nabla_x f(x_t,y_t)$, $\nabla_y f(x_t,y_t)$ and $\|y_t-y_t^*\|$ simultaneously to estimate the bound of these terms. This is one of the most challenging technical difficulties in our analysis. We can prove

$$\Phi(x_t)-\Phi^*\leq\Phi(x_0)-\Phi^*+\frac{G_x^2}{8\kappa l_x(2G_x)} \tag{6.16}$$

which will eventually finalize the mathematical induction.

In the stochastic case, the framework of mathematical induction in GDA does not work because neither the gradient norm nor the function value can be bounded when gradient noise exists. However, under these conditions, we can still prove the convergence of our Generalized SGDA with a probability of at least $1-\delta$. For Generalized SGDA, we define

$$G_x=\max\{u>0|u^2\leq 32\kappa l_x(2u)\cdot(\Phi(x_0)-\Phi^*+\sigma^2)/\delta\}$$

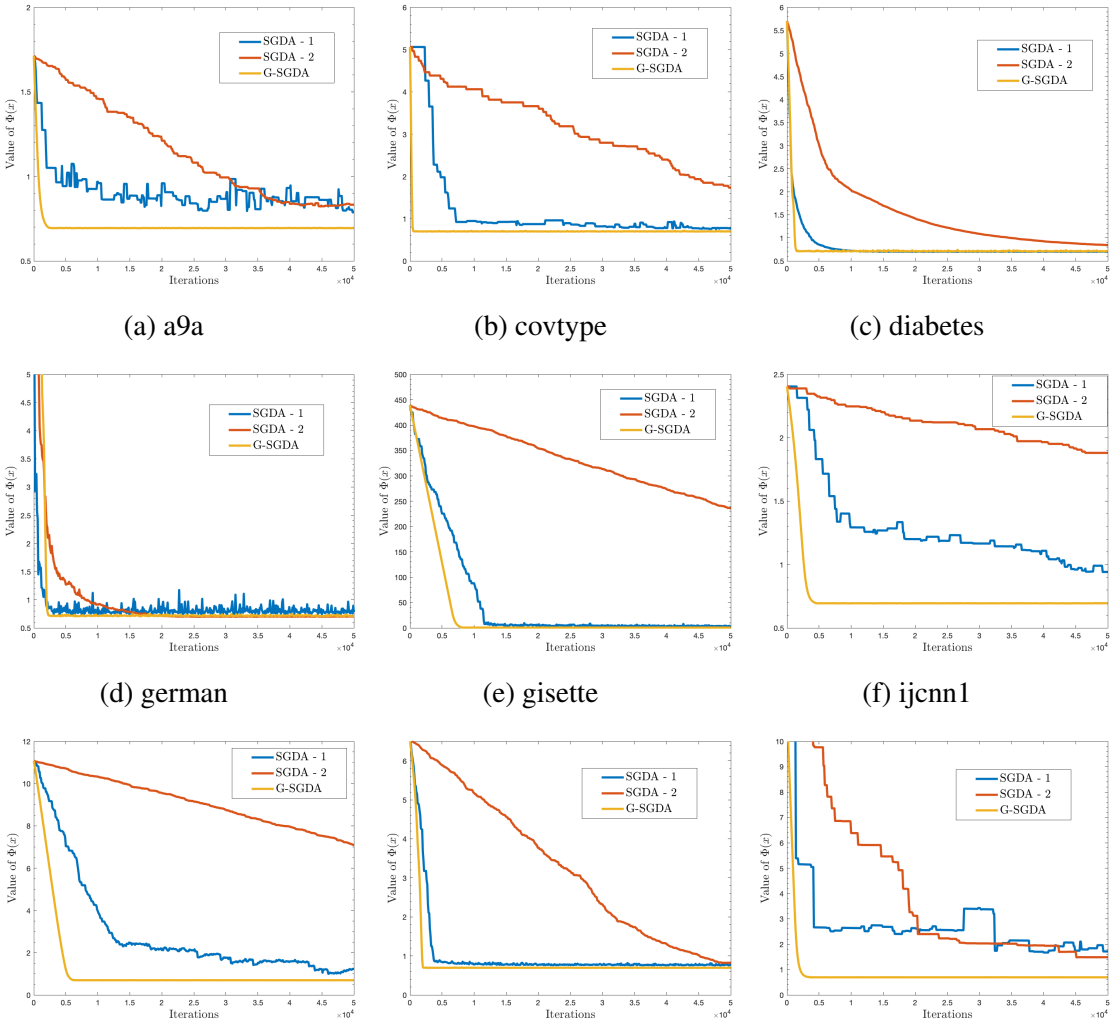
$$T_0=\min\{t|\Phi(x_t)-\Phi^*>F \text{ or } \|y_t^*-y_t\|>Y\}\wedge T$$

where $F = 8(\Phi(x_0) - \Phi^* + \sigma^2)/\delta$, $Y = \frac{C_0 G_x}{l_x(2G_x)}$ and $\wedge$ denotes the minimum operation. We want to prove that the probability of $T_0 < T$ is small. Notice that in minimization optimization we do not need to consider the upper bound of $\|y_t^* - y_t\|$, which is exclusive in minimax optimization. Based on the proof of the deterministic case, we can prove when $t < T_0$ all induction assumptions in the analysis of GDA are satisfied. Hence, we can obtain the estimations of expectations $\mathbb{E}\Phi(x_t) - \Phi^*$ and $\mathbb{E}\|y_t^* - y_t\|$ at iteration $t = T_0$. By Markov's inequality and union bound, we can prove that the probability of event $T_0 < T$ is smaller than $\frac{\delta}{2}$. Furthermore, by the union bound and the estimation of $\mathbb{E}\Phi(x_t)$ we can achieve the result in Theorem 6.3.

6.4.3 Discussion

In this subsection, we discuss the dependence of constants used in our convergence analysis. Since we run an additional initialization process to ensure $\|y_0 - y_0^*\| \leq C$ for some threshold C, we can obtain $G_y \leq \frac{1}{4}$ if there is no constraint with respect to y, *i.e.*, $\mathscr{Y} = \mathbb{R}^{d_2}$. Thus, we have $\kappa \leq \frac{l_y(1)}{\mu}$. If f is Lipschitz smooth with respect to y, we also have $\kappa \leq \frac{l_y(1)}{\mu}$. Hence, the condition number κ is a constant only depending on the function $l_y(\cdot)$. Inserting $\kappa \leq \frac{l_y(1)}{\mu}$ into the definition of G_x, we can see that G_x is a constant only depending on functions $l_x(\cdot)$, $l_y(\cdot)$, $\Phi(\cdot)$ and the initial value x_0. Besides, the initialization process can be regarded as a strongly-convex minimization subproblem, which aims to find an initial value satisfying $\|y_0 - y_0^*\|$ smaller than a constant tolerance. The complexity of this subproblem is shown to be within $O(\frac{l_y(2\|\nabla_y f(x_0, \tilde{y}_0)\|)}{\mu})$ where $\tilde{y}_0$ is the raw input of the variable y. Therefore, the complexity of the initialization process is dominated by the complexity to solve the entire minimax problem and therefore can be neglected.

Next, we will discuss the relation between parameters η and S_t. η can be regarded as a fixed stepsize parameter which is passed to the algorithm before it starts. S_t is the scale



(a) a9a (b) covtype (c) diabetes

(d) german (e) gisette (f) ijcnn1

(g) mushrooms (h) phishing (i) w8a

Figure 6.1: Experimental results of the loss function value of $\Phi(x)$ with respect to the number of iterations in the robust logistic regression task on dataset a9a, covtype, diabetes, german, gisette, ijcnn1, mushrooms, phishing, and w8a. SGDA-1 and SGDA-2 are the results of SGDA with two largest learning rates that make it converge. G-SGDA is the result of our Generalized SGDA Algorithm.

of suitable stepsize in iteration t which is computed during the execution of the algorithm. The ratio of $\frac{\eta}{S_t}$ should be bounded by a certain threshold according to our analysis. When S_t is chosen as a constant, the parameter η can also be a constant that does not depend on ε. When S_t is the gradient norm, the averaged historical gradient norm or exponential moving averaged historical gradient norm, the parameter η should be as small as $O(\varepsilon)$ with respect

to ε because S_t will gradually become as small as $O(\varepsilon)$.

6.5 Experiments

In this section, we will perform an experiment with the robust logistic regression task to validate the performance of our generalized minimax optimization methods with the suitable stepsize strategy. Recall the examples we have mentioned in Section 6.2, the problem can be formulated as

$$\min_{x\in\mathbb{R}^d}\max_{y\in\Delta_n} f(x,y) = \sum_{i=1}^{n} y_i l_i(x) - V(y) + g(x) \tag{6.17}$$

where $l_i(x)$ is the logistic loss function defined by $l_i(x) = \log(1+\exp(-b_i a_i^T x))$. $V(y)$ is a divergence measure defined by $V(y) = \frac{1}{2}\lambda_1 \|ny - \mathbf{1}\|^2$. The notation Δ_n represents the simplex in $\mathbb{R}^n$, that is,

$$\Delta_n = \{y \in \mathbb{R}^n | 0 \le y_i \le 1, \sum_{i=1}^{n} y_i = 1\} \tag{6.18}$$

Function $g(x)$ is the regularization term that takes the form $g(x) = \lambda_2 \sum_{i=1}^{d} \frac{\alpha x_i^2}{1+\alpha x_i^2}$. Following the experimental settings in [131], we set $\lambda_1 = \frac{1}{n^2}$, $\lambda_2 = 0.001$ and $\alpha = 10$ in our experiment.

We run the experiment and verify our method on 9 real-world datasets a9a, covtype, diabetes, german, gisette, ijcnn1, mushrooms, phishing, and w8a, which can be downloaded from the LIBSVM repository at `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets`. These datasets are frequently used in binary classification tasks. The description of these datasets is listed in Table 6.1.

We compare our generalized SGDA algorithm with suitable stepsize to the conventional constant stepsize SGDA. We choose option (1) to compute the suitable stepsize parameter S_t, which adopts gradient normalization. The size of the mini-batch is set to 50. For each

algorithm, we choose the best learning rates η and η_y from $\{0.1, 0.01, 0.001, 0.0001, 1e-5, 1e-6\}$ by grid search. We report the results of two of the largest learning rates that can make SGDA converge. We compare the value of $\Phi(x)$ with respect to the number of iterations in the training process. The value of $\Phi(x)$ can be calculated because $y^*(x)$ has a closed form in this problem and the projection operation onto a simplex is also available to compute. The code is available at `https://github.com/WH-XIAN/AS-SGDA`.

The experimental results are shown in Figure 6.1. SGDA-1 and SGDA-2 are the results of SGDA with the two largest learning rates from $\{0.1, 0.01, 0.001, 0.0001, 1e-5, 1e-6\}$ that make it converge. G-SGDA is the result of our Generalized SGDA method with the suitable stepsize strategy. From the results in Figure 6.1 we can see that our suitable stepsize strategy significantly improves the convergence speed and stability of SGDA algorithm on all datasets, which validates the effectiveness of our Generalized SGDA method.

6.6 Conclusion

In this chapter, we investigate the convergence analysis of minimax optimization algorithms under the relaxation of Lipschitz smooth condition. We provide some counterexamples to reveal that non-Lipschitz smoothness and divergence issues could occur in minimax problems. We propose some generalized minimax algorithms with the suitable stepsize strategy to tackle this issue. We prove that variants of fundamental minimax optimization algorithms GDA, SGDA, GDmax and SGDmax can still converge under the generalized Lipschitz smooth conditions and achieve the same gradient complexity or SFO complexity as their counterparts do in the Lipschitz smooth case. We conduct a numerical experiment of the robust logistic regression task to validate the practical performance of our methods.

Chapter 7: Conclusions

In this dissertation, I investigated and solved several core machine learning issues. In Chapter 2, I propose a novel efficient Decentralized Quantized Stochastic Frank-Wolfe (DQSFW) method to solve the decentralized constrained optimization problems, which requires less communication cost but still achieves a good convergence rate. In Chapter 3, I propose a new accelerated decentralized stochastic first-order algorithm DM-HSGD, to solve the decentralized nonconvex-strongly-concave minimax optimization problems. In Chapter 4, I propose a novel communication-efficient adaptive gradient algorithm named SketchedAMSGrad, which can reduce the communication cost from $O(d)$ to $O(\log(d))$. In Chapter 5, I propose a novel algorithm PEDESTAL, which is the first decentralized stochastic gradient-based algorithm to achieve second-order optimality with non-asymptotic analysis. In Chapter 6, I prove that generalizations of classic minimax optimization algorithms can still converge under the generalized smooth condition and the gradient complexity matches the Lipschitz smooth counterparts.

Appendix A: Appendix of Chapter 2

A.1 Proof of Auxiliary Propositions

Proposition A.1. *Suppose A and B are two matrices. Then it satisfies*

$$\|AB\|_F \leq \|A\|_2 \|B\|_F \tag{A.1}$$

This is a common inequality in linear algebra. So we omit the proof here.

Proposition A.2. *Suppose W is a doubly stochastic matrix satisfying Assumption 2.4. Then*

$$\|W - \frac{\mathbf{1}\mathbf{1}^T}{M}\|_2 \leq \rho \tag{A.2}$$

Proof. For both W and $\frac{\mathbf{1}\mathbf{1}^T}{M}$, $\mathbf{1}$ is an eigenvector of eigenvalue 1. According to Assumption 2.4, for $\forall \lambda_i, i \neq 1$, we have $\lambda_i \neq 1$. Let v_i be λ_i's eigenvector, then

$$v_i^T W \mathbf{1} = v_i^T \mathbf{1} = \mathbf{1}^T W v_i = \lambda_i \mathbf{1}^T v_i \tag{A.3}$$

hence $\mathbf{1}^T v_i = 0$. Therefore, the eigenvalues of $W - \frac{\mathbf{1}\mathbf{1}^T}{M}$ are $0, \lambda_2, \ldots, \lambda_M$ and $\|W - \frac{\mathbf{1}\mathbf{1}^T}{M}\|_2 \leq \rho$. □

Proposition A.3. *Suppose* $0 < p < 1$, $r > 0$, *series* A_t *satisfies* $A_{t+1} \leq (1-p)A_t + \frac{1}{(t+1)^r}$. *Then there exists constant R such that* $A_t \leq \frac{R}{(t+1)^r}$.

Proof. Let

$$t_0 = [\frac{1}{1-(1-\frac{p}{2})^{1/r}}]$$

For $t \leq t_0$ there exists constant R_1 such that $A_t \leq \frac{R_1}{(t+1)^r}$. When $t > t_0$, by the definition of t_0, we have

$$1 - \frac{p}{2} \leq (\frac{t+1}{t+2})^r$$

Suppose $R_2 \geq \frac{2}{p}$ and $A_{t_0} \leq \frac{R_2}{(t_0+1)^r}$. By mathematical deduction, assume we have $A_t \leq \frac{R_2}{(t+1)^r}$, $t \geq t_0$, then we also have

$$A_{t+1} \leq (1-p)A_t + \frac{1}{(t+1)^r} \leq \frac{(1-p)R_2+1}{(t+1)^r} \leq \frac{R_2}{(t+2)^r}$$

Therefore, let $R = max\{R_1, \frac{2}{p}\}$ and the conclusion will hold. □

A.2 Proof of Lemmas

A.2.1 Proof of Lemma 2.1

Proof. First we have

$$\frac{1-\delta_1}{1-\delta} = (1+\delta)(2-\sqrt{1-\delta^2})^2 > 1 \tag{A.4}$$

Therefore, $\delta_1 < \delta$. Then we have

$$\begin{aligned}
&\|X_{t+1} - \overline{X_{t+1}}\|_F^2 \\
&= \|X_{t+\frac{1}{2}} - \overline{X_{t+\frac{1}{2}}} + \gamma\hat{X}_t(W-I)\|_F^2 \\
&= \|X_{t+\frac{1}{2}} - \overline{X_{t+\frac{1}{2}}} + \gamma(\hat{X}_t - X_{t+\frac{1}{2}})(W-I) + \gamma X_{t+\frac{1}{2}}(W-I)\|_F^2 \\
&= \|X_{t+\frac{1}{2}} - \overline{X_{t+\frac{1}{2}}} + \gamma(\hat{X}_t - X_{t+\frac{1}{2}})(W-I) + \gamma(X_{t+\frac{1}{2}} - \overline{X_{t+\frac{1}{2}}})(W-I)\|_F^2 \\
&= \|(X_{t+\frac{1}{2}} - \overline{X_{t+\frac{1}{2}}})((1-\gamma)I + \gamma W) + \gamma(\hat{X}_t - X_{t+\frac{1}{2}})(W-I)\|_F^2
\end{aligned} \tag{A.5}$$

$$
\begin{aligned}
&\leq (1+c_1)\|(X_{t+\frac{1}{2}}-\overline{X_{t+\frac{1}{2}}})((1-\gamma)I+\gamma W)\|_F^2+(1+\frac{1}{c_1})\|\gamma(\hat{X}_t-X_{t+\frac{1}{2}})(W-I)\|_F^2\\
&=(1+c_1)\|(X_{t+\frac{1}{2}}-\overline{X_{t+\frac{1}{2}}})((1-\gamma)I+\gamma(W-\frac{\mathbf{1}\mathbf{1}^T}{M}))\|_F^2+(1+\frac{1}{c_1})\|\gamma(\hat{X}_t-X_{t+\frac{1}{2}})(W-I)\|_F^2\\
&\leq (1+c_1)(1-(1-\rho)\gamma)^2\|X_{t+\frac{1}{2}}-\overline{X_{t+\frac{1}{2}}}\|_F^2+(1+\frac{1}{c_1})\gamma^2\zeta^2\|\hat{X}_t-X_{t+\frac{1}{2}}\|_F^2 \qquad \text{(A.6)}
\end{aligned}
$$

The first inequality comes from Young's Inequality and the second inequality is derived by Proposition A.1 and Proposition A.2. By Assumption 2.5 we also have

$$
\begin{aligned}
&\|X_{t+1}-\hat{X}_{t+1}\|_F^2\\
=\;&\|X_{t+1}-\hat{X}_t-C(X_{t+1}-\hat{X}_t)\|_F^2\\
\leq\;&(1-\delta)\|X_{t+1}-\hat{X}_t\|_F^2\\
=\;&(1-\delta)\|X_{t+\frac{1}{2}}-\hat{X}_t+\gamma\hat{X}_t(W-I)\|_F^2\\
=\;&(1-\delta)\|(X_{t+\frac{1}{2}}-\hat{X}_t)((1+\gamma)I-\gamma W)-\gamma X_{t+\frac{1}{2}}(I-W)\|_F^2\\
=\;&(1-\delta)\|(X_{t+\frac{1}{2}}-\hat{X}_t)((1+\gamma)I-\gamma W)-\gamma(X_{t+\frac{1}{2}}-\overline{X_{t+\frac{1}{2}}})(I-W)\|_F^2\\
\leq\;&(1-\delta)(1+c_2)\|(X_{t+\frac{1}{2}}-\hat{X}_t)((1+\gamma)I-\gamma W)\|_F^2+(1-\delta)(1+\frac{1}{c_2})\|\gamma(X_{t+\frac{1}{2}}-\overline{X_{t+\frac{1}{2}}})(I-W)\|_F^2\\
\leq\;&(1-\delta)(1+c_2)(1+\gamma\zeta)^2\|X_{t+\frac{1}{2}}-\hat{X}_t\|_F^2+(1-\delta)(1+\frac{1}{c_2})\gamma^2\zeta^2\|X_{t+\frac{1}{2}}-\overline{X_{t+\frac{1}{2}}}\|_F^2 \qquad \text{(A.7)}
\end{aligned}
$$

Here the last two inequalities are also derived from Young's Inequality, Proposition A.1 and Proposition A.2. Let

$$
A=(1+c_1)(1-(1-\rho)\gamma)^2+(1-\delta)(1+\frac{1}{c_2})\gamma^2\zeta^2 \qquad \text{(A.8)}
$$

and

$$
B=(1+\frac{1}{c_1})\gamma^2\zeta^2+(1-\delta)(1+c_2)(1+\gamma\zeta)^2 \qquad \text{(A.9)}
$$

Then we have

$$
\begin{aligned}
&\|X_{t+1}-\overline{X_{t+1}}\|_F^2+\|X_{t+1}-\hat{X}_{t+1}\|_F^2\\
&\leq A\|X_{t+\frac{1}{2}}-\overline{X_{t+\frac{1}{2}}}\|_F^2+B\|\hat{X}_t-X_{t+\frac{1}{2}}\|_F^2\\
&\leq (1+c_3)(A\|X_t-\overline{X_t}\|_F^2+B\|X_t-\hat{X}_t\|_F^2)+(1+\frac{1}{c_3})\eta_t^2(A\|(D_t-X_t)-(\overline{D_t}-\overline{X_t})\|_F^2+B\|D_t-X_t\|_F^2)\\
&\leq (1+c_3)(A\|X_t-\overline{X_t}\|_F^2+B\|X_t-\hat{X}_t\|_F^2)+2(1+\frac{1}{c_3})\eta_t^2(A+B)(\|X_t-\overline{X_t}\|_F^2+MD^2)
\end{aligned}
\tag{A.10}
$$

Let $c_1=(1-\rho)\gamma$, $c_2=\delta$, $c_3=min\{\frac{(1-\rho)\gamma}{4},\frac{\delta_1}{4}\}$ and $p=c_3^2$. Let $\eta_t=\frac{\eta_0}{(t+1)^\theta}$, $\eta_0=\frac{c_3}{2}$. Denote $2(1+\frac{1}{c_3})(A+B)$ as Q. We obtain

$$
\|X_{t+1}-\overline{X_{t+1}}\|_F^2+\|X_{t+1}-\hat{X}_{t+1}\|_F^2\leq(1-p)(\|X_t-\overline{X_t}\|_F^2+\|X_t-\hat{X}_t\|_F^2)+Q\eta_t^2MD^2 \tag{A.11}
$$

According to Proposition A.3, there exists constant R_1 such that

$$
\|X_t-\overline{X_t}\|_F^2+\|X_t-\hat{X}_t\|_F^2\leq\frac{QR_1MD^2}{(t+1)^{2\theta}} \tag{A.12}
$$

which finishes the proof. □

A.2.2 Proof of Lemma 2.2

Proof. From Lemma 2.1 we know

$$
\begin{aligned}
\|X_{t+1}-X_t\|_F^2 &\leq 3\|X_{t+1}-\overline{X_{t+1}}\|_F^2+3\|X_t-\overline{X_t}\|_F^2+3\|\overline{X_{t+1}}-\overline{X_t}\|_F^2\\
&\leq \frac{3QR_1MD^2}{(t+2)^{2\theta}}+\frac{3QR_1MD^2}{(t+1)^{2\theta}}+3\eta_t^2\|\overline{D_t}-\overline{X_t}\|_F^2\\
&\leq \frac{(6QR_1+3\eta_0)MD^2}{(t+1)^{2\theta}}
\end{aligned}
\tag{A.13}
$$

The first inequality comes from Cauchy Schwartz Inequality. The last inequality comes from Assumption 2.2. Let $c_4 = c_3^2$. $\beta_0 = \frac{B(1+c_4)}{Ac_4}$ and $\beta_t = \frac{\beta_0}{(t+1)^{2\theta/3}}$. Mimic the proof of Lemma 2.1, we have

$$
\begin{aligned}
&\|V_t - \overline{V_t}\|_F^2 + \|V_t - \hat{V}_t\|_F^2 \\
\leq\ & A\|V_{t-\frac{1}{2}} - \overline{V_{t-\frac{1}{2}}}\|_F^2 + B\|\hat{V}_{t-1} - V_{t-\frac{1}{2}}\|_F^2 \\
\leq\ & (1+c_3)[A(1-\beta_t)^2\|V_{t-1} - \overline{V_{t-1}}\|_F^2 + B\|\hat{V}_{t-1} - V_{t-1} + \beta_t(V_{t-1} - \overline{V_{t-1}})\|_F^2] \\
&+ (1+\frac{1}{c_3})\beta_t^2(A\|G_t - \overline{G_t}\|_F^2 + B\|\overline{V_{t-1}} - G_t\|_F^2) \\
\leq\ & (1+c_3)[A(1-\beta_t)^2\|V_{t-1} - \overline{V_{t-1}}\|_F^2 + (1+c_4)B\|\hat{V}_{t-1} - V_{t-1}\|_F^2 + (1+\frac{1}{c_4})B\beta_t^2\|V_{t-1} - \overline{V_{t-1}}\|_F^2] \\
&+ (1+\frac{1}{c_3})\beta_t^2(A\|G_t - \overline{G_t}\|_F^2 + B\|\overline{V_{t-1}} - G_t\|_F^2) \\
\leq\ & (1+c_3)[A\|V_{t-1} - \overline{V_{t-1}}\|_F^2 + (1+c_4)B\|\hat{V}_{t-1} - V_{t-1}\|_F^2] + 4(1+\frac{1}{c_3})(A+B)\beta_t^2 MG^2 \\
\leq\ & (1-p^2)(\|V_{t-1} - \overline{V_{t-1}}\|_F^2 + \|V_{t-1} - \hat{V}_{t-1}\|_F^2) + Q\beta_t^2 MG^2 \qquad \text{(A.14)}
\end{aligned}
$$

Then, according to Proposition A.3 there exists constant R_2 such that

$$\|V_t - \overline{V_t}\|_F^2 + \|V_t - \hat{V}_t\|_F^2 \leq \frac{QR_2MG^2}{(t+1)^{2\theta/3}} \qquad \text{(A.15)}$$

which finishes the proof. □

A.2.3 Proof of Lemma 2.3

Proof. Eq. (A.14) implies

$$\frac{1}{M}\sum_{i=1}^{M}\|v_t^{(i)} - \frac{1}{M}\sum_{j=1}^{M}v_t^{(j)}\|^2 \leq \frac{QR_2G^2}{(t+1)^{2\theta/3}} \qquad \text{(A.16)}$$

The mean of $v_t^{(i)}$ satisfies

$$\frac{1}{M}\sum_{i=1}^{M} v_t^{(i)} = (1-\beta_t)\frac{1}{M}\sum_{i=1}^{M} v_{t-1}^{(i)} + \beta_t \frac{1}{M}\sum_{i=1}^{M} g_t^{(i)} \tag{A.17}$$

We can denote it as

$$\bar{v}_t = (1-\beta_t)\bar{v}_{t-1} + \beta_t \bar{g}_t \tag{A.18}$$

Then we have

$$\mathbb{E}\bar{g}_t = \frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_t^{(i)}) \tag{A.19}$$

and

$$\|\frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_{t+1}^{(i)}) - \frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_t^{(i)})\|^2 \leq \frac{L}{M}\|X_{t+1} - X_t\|_F^2 \leq \frac{(6QR_1 + 3\eta_0)LD^2}{(t+1)^{2\theta}} \tag{A.20}$$

where we have used Assumption 2.1. According to Eq. (A.19) and Assumption 2.6 we have

$$\begin{aligned}
&\mathbb{E}\|\bar{v}_t - \frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_t^{(i)})\|^2 \\
&= \mathbb{E}\|(1-\beta_t)(\bar{v}_{t-1} - \frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_{t-1}^{(i)})) + (1-\beta_t)(\frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_{t-1}^{(i)}) - \frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_t^{(i)})) \\
&\quad + \beta_t(\bar{g}_t - \frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_t^{(i)}))\|^2 \\
&= (1-\beta_t)^2\mathbb{E}\|\bar{v}_{t-1} - \frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_{t-1}^{(i)})\|^2 + (1-\beta_t)^2\mathbb{E}\|\frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_t^{(i)}) - \frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_{t-1}^{(i)})\|^2 \\
&\quad - 2(1-\beta_t)^2\mathbb{E}\langle \bar{v}_{t-1} - \frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_{t-1}^{(i)}), \frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_t^{(i)}) - \frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_{t-1}^{(i)})\rangle \\
&\quad + \beta_t^2\mathbb{E}\|\bar{g}_t - \frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_t^{(i)})\|^2 \\
&\leq (1-\beta_t)^2\mathbb{E}\|\bar{v}_{t-1} - \frac{1}{M}\sum_{i=1}^{M} \nabla f_i(x_{t-1}^{(i)})\|^2 + (1-\beta_t)^2\frac{(6QR_1 + 3\eta_0)LD^2}{t^{2\theta}}
\end{aligned}$$

$$+(1-\beta_t)^2\frac{\beta_t}{2}\mathbb{E}\|\bar{v}_{t-1}-\frac{1}{M}\sum_{i=1}^{M}\nabla f_i(x_{t-1}^{(i)})\|^2+(1-\beta_t)^2\frac{2}{\beta_t}\frac{(6QR_1+3\eta_0)LD^2}{t^{2\theta}}+\beta_t^2\sigma^2 \tag{A.21}$$

As $(1-\beta_t)(1+\frac{\beta_t}{2})<1$, we have

$$\begin{aligned}&\mathbb{E}\|\bar{v}_t-\frac{1}{M}\sum_{i=1}^{M}\nabla f_i(x_t^{(i)})\|^2\\ \leq\ &(1-\beta_t)\mathbb{E}\|\bar{v}_{t-1}-\frac{1}{M}\sum_{i=1}^{M}\nabla f_i(x_{t-1}^{(i)})\|^2+(1+\frac{2}{\beta_t})\frac{(6QR_1+3\eta_0)LD^2}{t^{2\theta}}+\beta_t^2\sigma^2\end{aligned} \tag{A.22}$$

There exists a constant integer t_0 such that

$$(1+t_0)^{1-2\theta/3}>\frac{2}{\beta_0} \tag{A.23}$$

For $t\leq t_0$, there exists a constant S_1 such that

$$\mathbb{E}\|\bar{v}_t-\frac{1}{M}\sum_{i=1}^{M}\nabla f_i(x_t^{(i)})\|^2\leq\frac{S_1}{(t+1)^{2\theta/3}} \tag{A.24}$$

Let $S=max\{S_1,\frac{48(2QR_1+\eta_0)LD^2}{\beta_0^2},4\sigma^2\}$. Then we use mathematical induction to prove

$$\mathbb{E}\|\bar{v}_t-\frac{1}{M}\sum_{i=1}^{M}\nabla f_i(x_t^{(i)})\|^2\leq\frac{S}{(t+1)^{2\theta/3}} \tag{A.25}$$

We have already know it holds when $t\leq t_0$. When $t>t_0$, suppose case $t-1$ satisfies the assumption. Then we have

$$\mathbb{E}\|\bar{v}_t-\frac{1}{M}\sum_{i=1}^{M}\nabla f_i(x_t^{(i)})\|^2$$

$$\leq \quad (1-\frac{\beta_0}{(t+1)^{2\theta/3}})\frac{S}{t^{2\theta/3}}+(1+\frac{2(t+1)^{2\theta/3}}{\beta_0})\frac{(6QR_1+3\eta_0)LD^2}{t^{2\theta}}+\frac{\beta_0^2\sigma^2}{(t+1)^{4\theta/3}} \tag{A.26}$$

Define function $h(x)=1+\frac{\beta_0}{2}x^{2\theta/3}-(1+x)^{2\theta/3}$. We can calculate the derivative

$$\begin{aligned} h'(x) &= \frac{2\theta}{3}(\frac{\beta_0}{2}x^{2\theta/3-1}-(1+x)^{2\theta/3-1}) \\ &= \frac{\theta\beta_0}{3(1+x)^{1-2\theta/3}}((1+\frac{1}{x})^{1-2\theta/3}-\frac{2}{\beta_0}) \end{aligned} \tag{A.27}$$

When $0<x\leq\frac{1}{t_0}$, $h'(x)>0$. Since $h(0)=0$, we get $h(x)\geq 0$ when $x\in[0,\frac{1}{t_0}]$. Therefore, as $t>t_0$, we have

$$(\frac{t+1}{t})^{2\theta/3}\leq 1+\frac{\beta_0}{2}\frac{1}{t^{2\theta/3}} \tag{A.28}$$

We can rewrite as

$$(1-\frac{\beta_0}{2(t+1)^{2\theta/3}})\frac{1}{t^{2\theta/3}}\leq\frac{1}{(t+1)^{2\theta/3}} \tag{A.29}$$

According to the definition of S, we also have

$$(1+\frac{2(t+1)^{2\theta/3}}{\beta_0})\frac{(6QR_1+3\eta_0)LD^2}{t^{2\theta}}+\frac{\beta_0^2\sigma^2}{(t+1)^{4\theta/3}}\leq\frac{\beta_0}{2(t+1)^{2\theta/3}}\frac{S}{t^{2\theta/3}} \tag{A.30}$$

Combine above two equations we reach the conclusion of our induction. □

A.3 Proof of Theorems

A.3.1 Proof of Theorem 2.1

Proof. By Assumption 2.1, we have

$$f(X_{t+1}\frac{\mathbf{1}}{M})\leq f(X_t\frac{\mathbf{1}}{M})+\langle\nabla f(X_t\frac{\mathbf{1}}{M}),X_{t+1}\frac{\mathbf{1}}{M}-X_t\frac{\mathbf{1}}{M}\rangle+\frac{L^2}{2}\|X_{t+1}\frac{\mathbf{1}}{M}-X_t\frac{\mathbf{1}}{M}\|^2$$

$$
\begin{aligned}
&\leq f(X_t\frac{\mathbf{1}}{M}) + \langle \nabla f(X_t\frac{\mathbf{1}}{M}), X_{t+1}\frac{\mathbf{1}}{M} - X_t\frac{\mathbf{1}}{M}\rangle + \frac{\eta_t^2 L^2 D^2}{2} \\
&= f(X_t\frac{\mathbf{1}}{M}) + \eta_t\langle \nabla f(X_t\frac{\mathbf{1}}{M}), D_t\frac{\mathbf{1}}{M} - X_t\frac{\mathbf{1}}{M}\rangle + \frac{\eta_t^2 L^2 D^2}{2}
\end{aligned} \tag{A.31}
$$

Define $\widehat{d_t}$ as

$$
\widehat{d_t} = argmin_{d\in\Omega}\langle d, \nabla f(X_t\frac{\mathbf{1}}{M})\rangle \tag{A.32}
$$

Term $\langle \nabla f(X_t\frac{\mathbf{1}}{M}), D_t\frac{\mathbf{1}}{M} - X_t\frac{\mathbf{1}}{M}\rangle$ can be estimated by

$$
\begin{aligned}
&\langle \nabla f(X_t\frac{\mathbf{1}}{M}), D_t\frac{\mathbf{1}}{M} - X_t\frac{\mathbf{1}}{M}\rangle \\
&= \frac{1}{M}\sum_{i=1}^{M}\langle \nabla f(X_t\frac{\mathbf{1}}{M}), d_t^{(i)} - X_t\frac{\mathbf{1}}{M}\rangle \\
&= \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\langle \nabla f(X_t\frac{\mathbf{1}}{M}), d_t^{(i)} - d_t^{(j)}\rangle + \langle \nabla f(X_t\frac{\mathbf{1}}{M}), d_t^{(j)} - X_t\frac{\mathbf{1}}{M}\rangle \\
&= \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\langle v_t^{(j)}, d_t^{(i)} - d_t^{(j)}\rangle + \langle \nabla f(X_t\frac{\mathbf{1}}{M}) - v_t^{(j)}, d_t^{(i)} - d_t^{(j)}\rangle \\
&\quad + \langle \nabla f(X_t\frac{\mathbf{1}}{M}) - v_t^{(j)}, d_t^{(j)} - X_t\frac{\mathbf{1}}{M}\rangle + \langle v_t^{(j)}, d_t^{(j)} - X_t\frac{\mathbf{1}}{M}\rangle \\
&\leq \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\langle v_t^{(j)}, d_t^{(i)} - d_t^{(j)}\rangle + \langle \nabla f(X_t\frac{\mathbf{1}}{M}) - v_t^{(j)}, d_t^{(i)} - d_t^{(j)}\rangle \\
&\quad + \langle \nabla f(X_t\frac{\mathbf{1}}{M}) - v_t^{(j)}, d_t^{(j)} - X_t\frac{\mathbf{1}}{M}\rangle + \langle v_t^{(j)}, \widehat{d_t} - X_t\frac{\mathbf{1}}{M}\rangle \\
&= \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\langle v_t^{(j)}, d_t^{(i)} - d_t^{(j)}\rangle + \langle \nabla f(X_t\frac{\mathbf{1}}{M}) - v_t^{(j)}, d_t^{(i)} - d_t^{(j)}\rangle \\
&\quad + \langle \nabla f(X_t\frac{\mathbf{1}}{M}) - v_t^{(j)}, d_t^{(j)} - \widehat{d_t}\rangle + \langle \nabla f(X_t\frac{\mathbf{1}}{M}), \widehat{d_t} - X_t\frac{\mathbf{1}}{M}\rangle \\
&= \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\langle v_t^{(j)}, d_t^{(i)} - d_t^{(j)}\rangle + \langle \nabla f(X_t\frac{\mathbf{1}}{M}) - v_t^{(j)}, d_t^{(i)} - \widehat{d_t}\rangle - \mathscr{G}_t \\
&= (\frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\langle v_t^{(j)}, d_t^{(i)} - d_t^{(j)}\rangle) + \langle \nabla f(X_t\frac{\mathbf{1}}{M}) - \bar{v}_t, \bar{d}_t - \widehat{d_t}\rangle - \mathscr{G}_t \\
&\leq -\mathscr{G}_t + D\|\nabla f(X_t\frac{\mathbf{1}}{M}) - \bar{v}_t\| + \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\langle v_t^{(j)}, d_t^{(i)} - d_t^{(j)}\rangle
\end{aligned} \tag{A.33}
$$

The first inequality is achieved by the definition of $\widehat{d_t}$ and the second inequality is achieved by the definition of $\mathscr{G}_t$. By Lemma 2.2 we have

$$\begin{aligned}
&\frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\langle v_t^{(j)}, d_t^{(i)} - d_t^{(j)}\rangle \\
&= \frac{1}{M}\sum_{j=1}^{M}\langle v_t^{(j)}, \bar{d}_t - d_t^{(j)}\rangle = \frac{1}{M}\sum_{j=1}^{M}\langle v_t^{(j)} - \bar{v}_t, \bar{d}_t - d_t^{(j)}\rangle \\
&\leq \frac{1}{M}\sum_{j=1}^{M} D\|v_t^{(j)} - \bar{v}_t\| \leq D\sqrt{\frac{\|V_t - \overline{V_t}\|_F^2}{M}} \leq \frac{\sqrt{QR_2}GD}{(t+1)^{\theta/3}}
\end{aligned} \tag{A.34}$$

Moreover, by Lemma 2.1 and Lemma 2.3 we can estimate

$$\begin{aligned}
\mathbb{E}\|\nabla f(X_t\frac{\mathbf{1}}{M}) - \bar{v}_t\|^2 &\leq 2\mathbb{E}\|\nabla f(X_t\frac{\mathbf{1}}{M}) - \frac{1}{M}\sum_{i=1}^{M}\nabla f_i(x_t^{(i)})\|^2 + 2\mathbb{E}\|\frac{1}{M}\sum_{i=1}^{M}\nabla f_i(x_t^{(i)}) - \bar{v}_t\|^2 \\
&\leq \frac{2L^2}{M}\|X_t - \overline{X_t}\|_F^2 + \frac{2S}{(t+1)^{2\theta/3}} \\
&\leq \frac{2QR_1L^2D^2}{(t+1)^{2\theta}} + \frac{2S}{(t+1)^{2\theta/3}} \leq \frac{2(S+QR_1L^2D^2)}{(t+1)^{2\theta/3}}
\end{aligned} \tag{A.35}$$

Therefore, we have

$$\mathbb{E}\|\nabla f(X_t\frac{\mathbf{1}}{M}) - \bar{v}_t\| \leq \frac{\sqrt{2(S+QR_1L^2D^2)}}{(t+1)^{\theta/3}} \tag{A.36}$$

According to Eq. (A.31), (A.33), (A.34) and (A.36) we have

$$\eta_t\mathbb{E}\mathscr{G}_t \leq \mathbb{E}(f(X_t\frac{\mathbf{1}}{M}) - f(X_{t+1}\frac{\mathbf{1}}{M})) + \eta_t\frac{D(\sqrt{2(S+QR_1L^2D^2)} + \sqrt{QR_2}G)}{(t+1)^{\theta/3}} + \frac{\eta_t^2L^2D^2}{2}$$

which reaches the conclusion of Theorem 2.1. □

A.3.2 Proof of Theorem 2.2

Proof. Let

$$f_t = \mathbb{E} f(X_t \frac{\mathbf{1}}{M}) \tag{A.37}$$

and $F_0, F_1, \ldots, F_T$ is a permutation of $f_0, f_1, \ldots, f_T$ in descending order. We also define $F_{t+1} = f_{t+1}$

Since we have Eq. (A.36) and $\eta_t = \frac{\eta_0}{(t+1)^\theta}$, telescoping over all iterations of t and taking expectation of $\hat{t}$, we obtain

$$\begin{aligned}\mathbb{E}\mathscr{G}_{\hat{t}} \leq & \frac{\sum_{t=0}^{T}(f_t - f_{t+1})(t+1)^{3/4}}{\eta_0(T+1)} + D(\sqrt{2(S+QR_1L^2D^2)} + \sqrt{QR_2}G)\frac{\sum_{t=0}^{T}\frac{1}{(t+1)^{1/4}}}{T+1} \\ & + \frac{L^2D^2}{2}\frac{\eta_0\sum_{t=0}^{T}\frac{1}{(t+1)^{3/4}}}{T+1}\end{aligned} \tag{A.38}$$

The numerator of the first term on right hand side of Eq. (A.38) equals

$$\sum_{t=0}^{T}(f_t - f_{t+1})(t+1)^{3/4} = \sum_{t=0}^{T} f_t((t+1)^{3/4} - t^{3/4}) - f_{T+1}(T+1)^{3/4} \tag{A.39}$$

Define

$$u(t) = (t+1)^{3/4} - t^{3/4} \tag{A.40}$$

We have

$$u'(t) = \frac{3}{4}(\frac{1}{(t+1)^{1/4}} - \frac{1}{t^{1/4}}) < 0 \tag{A.41}$$

which means function $u(t)$ is decreasing. By Rearranging Inequality,

$$\begin{aligned}\sum_{t=0}^{T} f_t((t+1)^{3/4}-t^{3/4}) &\leq \sum_{t=0}^{T} F_t((t+1)^{3/4}-t^{3/4}) \\ &= \sum_{t=0}^{T}(F_t-F_{t+1})(t+1)^{3/4}+F_{T+1}(T+1)^{3/4} \end{aligned} \tag{A.42}$$

As $F_t - F_{t+1} \geq 0$, we have

$$\begin{aligned}\frac{\sum_{t=0}^{T}(f_t-f_{t+1})(t+1)^{3/4}}{T+1} &\leq \frac{\sum_{t=0}^{T}(F_t-F_{t+1})(t+1)^{3/4}}{T+1} \leq \frac{\sum_{t=0}^{T}F_t-F_{t+1}}{(T+1)^{1/4}} \\ &= \frac{F_0-F_{T+1}}{(T+1)^{1/4}} \leq \frac{f^+-f^-}{(T+1)^{1/4}} \end{aligned} \tag{A.43}$$

where f^+ and f^- are upper and lower bound of f. According to the compact domain and bounded gradient assumptions, the upper bound always exists. Since $\frac{1}{(t+1)^{1/4}}$ is a descending function of t, we have estimation

$$\sum_{t=0}^{T}\frac{1}{(t+1)^{1/4}} \leq \int_0^{T+1}\frac{dx}{x^{1/4}} = \frac{4}{3}(T+1)^{3/4} \tag{A.44}$$

Similarly, we have

$$\sum_{t=0}^{T}\frac{1}{(t+1)^{3/4}} \leq 4(T+1)^{1/4} \tag{A.45}$$

By Eq. (A.39), (A.43), (A.44) and (A.45) we have

$$\mathbb{E}\mathscr{G}_{\hat{t}} = O(\frac{1}{T^{1/4}})$$

which reaches the conclusion of Theorem 2.2. □

Appendix B: Appendix of Chapter 3

B.1 Basic Lemmas

First, we introduce following basic lemmas, which are broadly used in the convergence analysis of optimization algorithms.

Lemma B.1. *Let vector X be a stochastic variable. Then we have*

$$0 \leq \mathbb{E}\|X - \mathbb{E}X\|^2 = \mathbb{E}\|X\|^2 - \|\mathbb{E}X\|^2 \leq \mathbb{E}\|X\|^2 \tag{B.1}$$

Lemma B.2. *Let $X_1, X_2, \cdots, X_n$ be n independent stochastic variables of which the means are 0. Then we have*

$$\mathbb{E}\|\sum_{i=1}^{n} X_i\|^2 = \sum_{i=1}^{n} \mathbb{E}\|X_i\|^2 \tag{B.2}$$

Lemma B.3. *Suppose A and B are two matrices. Then it satisfies*

$$\|AB\|_F \leq \|A\|_2 \|B\|_F \tag{B.3}$$

B.2 Important Conclusions

Next, we will propose and prove some conclusions that are important to the proof our main theorems.

Lemma B.4. *(Lemma 4.3 in paper [66])* $\Phi(x)$ *is* $(L+\kappa L)$*-smooth and* $y^*(\cdot)$ *is* κ*-Lipschitz, which means* $\|y^*(x_1)-y^*(x_2)\|\leq \kappa\|x_1-x_2\|$ *for any* $x_1,x_2\in\mathbb{R}^{d_1}$.

Proof. As $y^*(x_1)$ and $y^*(x_2)$ achieve the maximum, we have $\nabla_y f(x_1,y^*(x_1))=\mathbf{0}$ and $\nabla_y f(x_2,y^*(x_2))=\mathbf{0}$. Then we have

$$\begin{aligned}
&\|y^*(x_1)-y^*(x_2)\|\\
\leq&\frac{1}{\mu}\|\nabla_y f(x_1,y^*(x_1))-\nabla_y f(x_1,y^*(x_2))\|\\
=&\frac{1}{\mu}\|\nabla_y f(x_2,y^*(x_2))-\nabla_y f(x_1,y^*(x_2))\|\leq\frac{L}{\mu}\|x_1-x_2\|=\kappa\|x_1-x_2\|
\end{aligned}\tag{B.4}$$

where the first inequality is derived from μ-strong concavity and the second inequality is derived from L-smoothness. Since $\nabla\Phi(x)=\nabla_x f(x,y^*(x))$, from Assumption 3.1 we get

$$\|\nabla\Phi(x_1)-\nabla\Phi(x_2)\|\leq L\|x_1-x_2\|+L\|y^*(x_1)-y^*(x_2)\|\leq(L+\kappa L)\|x_1-x_2\|\tag{B.5}$$

which implies $\Phi(x)$ is $(L+\kappa L)$-smooth. □

Lemma B.5. *When* $\eta_y\leq\frac{1}{5L}$ *we have following estimation for* δ_t.

$$\begin{aligned}
\sum_{t=0}^{T-1}\delta_t\leq&\frac{4\kappa}{L\eta_y}\delta_0+\frac{18\eta_y}{\mu}\sum_{t=1}^{T-1}(1-\frac{\mu\eta_y}{4})^{T-t-1}\sum_{s=0}^{t-1}\|\bar{u}_s-\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_s^{(i)},y_s^{(i)})\|^2+\frac{72\kappa^2}{n}\sum_{t=0}^{T-1}(\|X_t-\bar{X}_t\|_F^2\\
&+\|Y_t-\bar{Y}_t\|_F^2)+\frac{20\kappa^4\eta_x^2}{L^2\eta_y^2}\sum_{t=0}^{T-1}\|\bar{v}_t\|^2-\frac{12}{5\mu^2}\sum_{t=0}^{T-1}(1-(1-\frac{\mu\eta_y}{4})^{T-t})\|\bar{u}_t\|^2
\end{aligned}\tag{B.6}$$

Proof. Define $z_t=\bar{y}_t+\theta\bar{u}_t$ for some constant θ. As function f is strongly-concave in y we

have

$$\begin{aligned} f(\bar{x}_t,\hat{y}_t) &\leq f(\bar{x}_t,\bar{y}_t)+\langle \nabla_y f(\bar{x}_t,\bar{y}_t),\hat{y}_t-\bar{y}_t\rangle-\frac{\mu}{2}\|\hat{y}_t-\bar{y}_t\|^2 \\ &= f(\bar{x}_t,\bar{y}_t)+\langle \bar{u}_t,\hat{y}_t-z_t\rangle+\langle \nabla_y f(\bar{x}_t,\bar{y}_t)-\bar{u}_t,\hat{y}_t-z_t\rangle \\ &\quad +\theta\langle \nabla_y f(\bar{x}_t,\bar{y}_t),\bar{u}_t\rangle-\frac{\mu}{2}\|\hat{y}_t-\bar{y}_t\|^2 \end{aligned} \tag{B.7}$$

By Assumption 3.1, we also have

$$-\frac{L\theta^2}{2}\|\bar{u}_t\|^2\leq f(\bar{x}_t,z_t)-f(\bar{x}_t,\bar{y}_t)-\theta\langle \nabla_y f(\bar{x}_t,\bar{y}_t),\bar{u}_t\rangle \tag{B.8}$$

Add Eq. (B.7) and Eq. (B.8) together we obtain

$$0\leq \langle \bar{u}_t,\hat{y}_t-z_t\rangle+\langle \nabla_y f(\bar{x}_t,\bar{y}_t)-\bar{u}_t,\hat{y}_t-z_t\rangle-\frac{\mu}{2}\|\hat{y}_t-\bar{y}_t\|^2+\frac{L\theta^2}{2}\|\bar{u}_t\|^2 \tag{B.9}$$

where we also use the definition of $\hat{y}_t$ so that $f(\bar{x}_t,\hat{y}_t)\geq f(\bar{x}_t,z_t)$.

$$\langle \bar{u}_t,\hat{y}_t-z_t\rangle=-\theta\|\bar{u}_t\|^2+\langle \bar{u}_t,\hat{y}_t-\bar{y}_t\rangle \tag{B.10}$$

Combining Eq. (B.9) and Eq. (B.10) we have

$$0\leq \langle \bar{u}_t,\hat{y}_t-\bar{y}_t\rangle-\frac{\mu}{2}\|\hat{y}_t-\bar{y}_t\|^2+\langle \nabla_y f(\bar{x}_t,\bar{y}_t)-\bar{u}_t,\hat{y}_t-z_t\rangle-(\theta-\frac{L\theta^2}{2})\|\bar{u}_t\|^2 \tag{B.11}$$

By Cauchy-Schwartz inequality we have

$$\langle \nabla_y f(\bar{x}_t,\bar{y}_t)-\bar{u}_t,\hat{y}_t-z_t\rangle\leq\frac{4}{\mu}\|\nabla_y f(\bar{x}_t,\bar{y}_t)-\bar{u}_t\|^2+\frac{\mu}{8}\|\hat{y}_t-\bar{y}_t\|^2+\frac{\mu\theta^2}{8}\|\bar{u}_t\|^2 \tag{B.12}$$

Therefore, we obtain

$$\begin{aligned}0 &\leq \langle \bar{u}_t, \hat{y}_t - \bar{y}_t \rangle - \frac{\mu}{4}\|\hat{y}_t - \bar{y}_t\|^2 + \frac{4}{\mu}\|\nabla_y f(\bar{x}_t, \bar{y}_t) - \bar{u}_t\|^2 - (\theta - \frac{L\theta^2}{2} - \frac{\mu\theta^2}{8})\|\bar{u}_t\|^2 \\ &\leq \langle \bar{u}_t, \hat{y}_t - \bar{y}_t \rangle - \frac{\mu}{4}\|\hat{y}_t - \bar{y}_t\|^2 + \frac{4}{\mu}\|\nabla_y f(\bar{x}_t, \bar{y}_t) - \bar{u}_t\|^2 - \frac{2}{5\mu}\|\bar{u}_t\|^2 \end{aligned} \tag{B.13}$$

where we let $\theta = \frac{4}{5\mu}$. As we have

$$2\eta_y \langle \bar{u}_t, \hat{y}_t - \bar{y}_t \rangle = \|\bar{y}_t - \hat{y}_t\|^2 + \|\bar{y}_{t+1} - \bar{y}_t\|^2 - \|\bar{y}_{t+1} - \hat{y}_t\|^2 \tag{B.14}$$

Eq. (B.13) is equivalent to

$$\|\bar{y}_{t+1} - \hat{y}_t\|^2 \leq (1 - \frac{\mu\eta_y}{2})\|\bar{y}_t - \hat{y}_t\|^2 + \|\bar{y}_{t+1} - \bar{y}_t\|^2 + \frac{8\eta_y}{\mu}\|\nabla_y f(\bar{x}_t, \bar{y}_t) - \bar{u}_t\|^2 - \frac{4\eta_y}{5\mu}\|\bar{u}_t\|^2 \tag{B.15}$$

When $L\eta_y \leq \frac{1}{5}$, from Eq. (B.15) we know

$$\|\bar{y}_{t+1} - \hat{y}_t\|^2 \leq (1 - \frac{\mu\eta_y}{2})\|\bar{y}_t - \hat{y}_t\|^2 + \frac{8\eta_y}{\mu}\|\nabla_y f(\bar{x}_t, \bar{y}_t) - \bar{u}_t\|^2 - \frac{3\eta_y}{5\mu}\|\bar{u}_t\|^2 \tag{B.16}$$

According to Young's inequality we have

$$\begin{aligned}\|\bar{y}_{t+1} - \hat{y}_{t+1}\|^2 &\leq (1 + \frac{\mu\eta_y}{4})\|\bar{y}_{t+1} - \hat{y}_t\|^2 + (1 + \frac{4}{\mu\eta_y})\|\hat{y}_{t+1} - \hat{y}_t\|^2 \\ &\leq (1 - \frac{\mu\eta_y}{4})\|\bar{y}_t - \hat{y}_t\|^2 + \frac{9\eta_y}{\mu}\|\nabla_y f(\bar{x}_t, \bar{y}_t) - \bar{u}_t\|^2 + \frac{5\kappa}{L\eta_y}\|\hat{y}_{t+1} - \hat{y}_t\|^2 - \frac{3\eta_y}{5\mu}\|\bar{u}_t\|^2 \\ &\leq (1 - \frac{\mu\eta_y}{4})\|\bar{y}_t - \hat{y}_t\|^2 + \frac{9\eta_y}{\mu}\|\nabla_y f(\bar{x}_t, \bar{y}_t) - \bar{u}_t\|^2 + \frac{5\kappa^3\eta_x^2}{L\eta_y}\|\bar{v}_t\|^2 - \frac{3\eta_y}{5\mu}\|\bar{u}_t\|^2 \end{aligned} \tag{B.17}$$

In the second inequality we use Eq. (B.16) and $L\eta_y \leq \frac{1}{5}$. The last inequality is because function $y^*(\cdot)$ is κ-Lipschitz. By Cauchy-Schwartz inequality and Assumption 3.1 we also

have

$$\|\nabla_y f(\bar{x}_t,\bar{y}_t)-\bar{u}_t\|^2\leq 2\|\bar{u}_t-\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_t^{(i)},y_t^{(i)})\|^2+\frac{2L^2}{n}(\|X_t-\bar{X}_t\|_F^2+\|Y_t-\bar{Y}_t\|_F^2) \tag{B.18}$$

Using the definition of δ_t and the recursion in Eq. (B.17) we obtain

$$\begin{aligned}\delta_t \leq (1-\frac{\mu\eta_y}{4})^t\delta_0+\frac{9\eta_y}{\mu}\sum_{s=0}^{t-1}(1-\frac{\mu\eta_y}{4})^{t-s-1}\|\bar{u}_s-\nabla_y f(\bar{x}_t,\bar{y}_t)\|^2+\frac{5\kappa^3\eta_x^2}{L\eta_y}\sum_{s=0}^{t-1}(1-\frac{\mu\eta_y}{4})^{t-s-1}\|\bar{v}_s\|^2\\ -\frac{3\eta_y}{5\mu}\sum_{s=0}^{t-1}(1-\frac{\mu\eta_y}{4})^{t-s-1}\|\bar{u}_s\|^2\end{aligned} \tag{B.19}$$

Summing above equation we have

$$\begin{aligned}\sum_{t=0}^{T-1}\delta_t \leq \frac{4\kappa}{L\eta_y}\delta_0+\frac{18\eta_y}{\mu}\sum_{t=1}^{T-1}(1-\frac{\mu\eta_y}{4})^{T-t-1}\sum_{s=0}^{t-1}\|\bar{u}_s-\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_s^{(i)},y_s^{(i)})\|^2+\frac{72\kappa^2}{n}\sum_{t=0}^{T-1}(\|X_t-\bar{X}_t\|_F^2\\ +\|Y_t-\bar{Y}_t\|_F^2)+\frac{20\kappa^4\eta_x^2}{L^2\eta_y^2}\sum_{t=0}^{T-1}\|\bar{v}_t\|^2-\frac{12}{5\mu^2}\sum_{t=0}^{T-1}(1-(1-\frac{\mu\eta_y}{4})^{T-t})\|\bar{u}_t\|^2\end{aligned} \tag{B.20}$$

where Eq. (B.18) is used. □

Lemma B.6. *For all $t\in\{0,1,\cdots,T\}$ we have $\bar{v}_t=\bar{g}_t$ and $\bar{u}_t=\bar{h}_t$.*

Proof. As matrix W is doubly stochastic, we have

$$\bar{v}_t=\bar{v}_{t-1}+\bar{g}_t-\bar{g}_{t-1} \tag{B.21}$$

which is equivalent to $\bar{v}_t-\bar{g}_t=\bar{v}_{t-1}-\bar{g}_{t-1}$. Since $\bar{u}_{-1}=\bar{g}_{-1}$, we have $\bar{v}_t=\bar{g}_t$ for all $t\in\{0,1,\cdots,T\}$. Similarly, we have $\bar{u}_t=\bar{h}_t$. □

Lemma B.7. *Let A_t, B_t be positive sequences satisfying*

$$A_{t+1}\leq(1-c)A_t+B_t \tag{B.22}$$

for some constant $c \in (0,1)$. Then for any positive integer T we have

$$\sum_{t=0}^{T} A_t \leq \frac{1}{c} A_0 + \frac{1}{c} \sum_{t=0}^{T-1} B_t \tag{B.23}$$

Proof. Using recursion on Eq. (B.22) we can obtain

$$A_t \leq (1-c)^t A_0 + \sum_{s=0}^{t-1} (1-c)^{t-s-1} B_s \tag{B.24}$$

for $\forall t \geq 0$. Sum above inequality and we achieve the desired conclusion Eq. (B.23), where we use the condtion A_t, B_t are positive and the fact that $\sum_{t=0}^{\infty}(1-c)^t = \frac{1}{c}$. □

Lemma B.8. *We can prove the following bound for gradient estimator $\bar{v}_t$ and $\bar{u}_t$.*

$$\begin{aligned}
\sum_{s=0}^{t-1} \mathbb{E}\|\bar{v}_s - \frac{1}{n}\sum_{i=1}^{n} \nabla_x f_i(x_s^{(i)}, y_s^{(i)})\|^2 &\leq \frac{\sigma^2}{n\beta_x b_0} + \frac{2\beta_x \sigma^2 t}{n} + \frac{12L^2}{n^2\beta_x} \sum_{s=0}^{t-1} (\mathbb{E}\|X_s - \bar{X}_s\|_F^2 + \mathbb{E}\|Y_s - \bar{Y}_s\|_F^2) \\
&\quad + \frac{6L^2}{n\beta_x} \sum_{s=0}^{t-2} (\eta_x^2 \mathbb{E}\|\bar{v}_s\|^2 + \eta_y^2 \mathbb{E}\|\bar{u}_s\|^2) \qquad \text{(B.25)} \\
\sum_{s=0}^{t-1} \mathbb{E}\|\bar{u}_s - \frac{1}{n}\sum_{i=1}^{n} \nabla_y f_i(x_s^{(i)}, y_s^{(i)})\|^2 &\leq \frac{\sigma^2}{n\beta_y b_0} + \frac{2\beta_y \sigma^2 t}{n} + \frac{12L^2}{n^2\beta_y} \sum_{s=0}^{t-1} (\mathbb{E}\|X_s - \bar{X}_s\|_F^2 + \mathbb{E}\|Y_s - \bar{Y}_s\|_F^2) \\
&\quad + \frac{6L^2}{n\beta_y} \sum_{s=0}^{t-2} (\eta_x^2 \mathbb{E}\|\bar{v}_s\|^2 + \eta_y^2 \mathbb{E}\|\bar{u}_s\|^2) \qquad \text{(B.26)}
\end{aligned}$$

for all $t \in \{1, 2, \cdots, T\}$.

Proof. By the definition of $g_t^{(i)}$ and Lemma B.6 we have

$$\begin{aligned}
&\bar{v}_t - \frac{1}{n}\sum_{i=1}^{n} \nabla_x f_i(x_t^{(i)}, y_t^{(i)}) \\
&= (1-\beta_x)(\bar{v}_{t-1} - \frac{1}{n}\sum_{i=1}^{n} \nabla_x f_i(x_{t-1}^{(i)}, y_{t-1}^{(i)})) + \frac{\beta_x}{n}\sum_{i=1}^{n} (\nabla_x F_i(x_t^{(i)}, y_t^{(i)}; \xi_t^{(i)}) - \nabla_x f_i(x_t^{(i)}, y_t^{(i)})) \\
&\quad + (1-\beta_x)\frac{1}{n}\sum_{i=1}^{n} \Big(\nabla_x F_i(x_t^{(i)}, y_t^{(i)}; \xi_t^{(i)}) - \nabla_x F_i(x_{t-1}^{(i)}, y_{t-1}^{(i)}; \xi_t^{(i)}) + \nabla_x f_i(x_{t-1}^{(i)}, y_{t-1}^{(i)}) \\
&\quad - \nabla_x f_i(x_t^{(i)}, y_t^{(i)})\Big) \qquad \text{(B.27)}
\end{aligned}$$

Taking expectation on $\xi_t^{(i)}$ the last two terms of Eq. (B.27) are 0. Therefore,

$$
\begin{aligned}
&\mathbb{E}\|\bar{v}_t - \frac{1}{n}\sum_{i=1}^{n}\nabla_x f_i(x_t^{(i)}, y_t^{(i)})\|^2 \\
&= (1-\beta_x)^2\mathbb{E}\|\bar{v}_{t-1} - \frac{1}{n}\sum_{i=1}^{n}\nabla_x f_i(x_{t-1}^{(i)}, y_{t-1}^{(i)})\|^2 + \mathbb{E}\|\frac{\beta_x}{n}\sum_{i=1}^{n}(\nabla_x F_i(x_t^{(i)}, y_t^{(i)}; \xi_t^{(i)}) \\
&\quad - \nabla_x f_i(x_t^{(i)}, y_t^{(i)})) + (1-\beta_x)\frac{1}{n}\sum_{i=1}^{n}\Big(\nabla_x F_i(x_t^{(i)}, y_t^{(i)}; \xi_t^{(i)}) - \nabla_x F_i(x_{t-1}^{(i)}, y_{t-1}^{(i)}; \xi_t^{(i)}) \\
&\quad\quad + \nabla_x f_i(x_{t-1}^{(i)}, y_{t-1}^{(i)}) - \nabla_x f_i(x_t^{(i)}, y_t^{(i)})\Big)\|^2 \\
&\leq (1-\beta_x)^2\mathbb{E}\|\bar{v}_{t-1} - \frac{1}{n}\sum_{i=1}^{n}\nabla_x f_i(x_{t-1}^{(i)}, y_{t-1}^{(i)})\|^2 + \frac{2\beta_x^2}{n^2}\sum_{i=1}^{n}\mathbb{E}\|\nabla_x F_i(x_t^{(i)}, y_t^{(i)}; \xi_t^{(i)}) \\
&\quad - \nabla_x f_i(x_t^{(i)}, y_t^{(i)})\|^2 + \frac{2(1-\beta_x)^2}{n^2}\sum_{i=1}^{n}\mathbb{E}\|\nabla_x F_i(x_t^{(i)}, y_t^{(i)}; \xi_t^{(i)}) - \nabla_x F_i(x_{t-1}^{(i)}, y_{t-1}^{(i)}; \xi_t^{(i)}) \\
&\quad + \nabla_x f_i(x_{t-1}^{(i)}, y_{t-1}^{(i)}) - \nabla_x f_i(x_t^{(i)}, y_t^{(i)})\|^2 \\
&\leq (1-\beta_x)^2\mathbb{E}\|\bar{v}_{t-1} - \frac{1}{n}\sum_{i=1}^{n}\nabla_x f_i(x_{t-1}^{(i)}, y_{t-1}^{(i)})\|^2 + \frac{2\beta_x^2\sigma^2}{n} + \frac{2L^2(1-\beta_x)^2}{n^2}(\mathbb{E}\|X_t - X_{t-1}\|_F^2 \\
&\quad + \mathbb{E}\|Y_t - Y_{t-1}\|_F^2)
\end{aligned}
\tag{B.28}
$$

The first inequality is obtained by Cauchy-Schwartz inequality. In the last inequality we use Lemma B.2 on the last two terms and then use Assumption 3.2, Lemma B.1 and Assumption 3.1. By Cauchy-Schwartz inequality we have estimations

$$\|X_t - X_{t-1}\|_F^2 \leq 3\|X_t - \bar{X}_t\|_F^2 + 3n\eta_x^2\|\bar{v}_{t-1}\|^2 + 3\|X_{t-1} - \bar{X}_{t-1}\|_F^2 \tag{B.29}$$

$$\|Y_t - Y_{t-1}\|_F^2 \leq 3\|Y_t - \bar{Y}_t\|_F^2 + 3n\eta_y^2\|\bar{u}_{t-1}\|^2 + 3\|Y_{t-1} - \bar{Y}_{t-1}\|_F^2 \tag{B.30}$$

Combining above two inequalities with Eq. (B.28) and Lemma B.7 we have

$$\begin{aligned}
&\sum_{s=0}^{t-1}\mathbb{E}\|\bar{v}_s-\frac{1}{n}\sum_{i=1}^{n}\nabla_x f_i(x_s^{(i)},y_s^{(i)})\|^2\\
\leq\ &\frac{1}{\beta_x}\mathbb{E}\|\bar{v}_0-\nabla_x f(x_0,y_0)\|^2+\frac{2\beta_x\sigma^2 t}{n}+\frac{12L^2}{n^2\beta_x}\sum_{s=0}^{t-1}(\mathbb{E}\|X_s-\bar{X}_s\|_F^2+\mathbb{E}\|Y_s-\bar{Y}_s\|_F^2)\\
&+\frac{6L^2}{n\beta_x}\sum_{s=0}^{t-2}(\eta_x^2\mathbb{E}\|\bar{v}_s\|^2+\eta_y^2\mathbb{E}\|\bar{u}_s\|^2)\\
\leq\ &\frac{\sigma^2}{n\beta_x b_0}+\frac{2\beta_x\sigma^2 t}{n}+\frac{12L^2}{n^2\beta_x}\sum_{s=0}^{t-1}(\mathbb{E}\|X_s-\bar{X}_s\|_F^2+\mathbb{E}\|Y_s-\bar{Y}_s\|_F^2)\\
&+\frac{6L^2}{n\beta_x}\sum_{s=0}^{t-2}(\eta_x^2\mathbb{E}\|\bar{v}_s\|^2+\eta_y^2\mathbb{E}\|\bar{u}_s\|^2) \qquad \text{(B.31)}
\end{aligned}$$

for all $t\in\{1,2,\cdots,T\}$. In the first inequality we use the fact $\frac{1}{1-(1-\beta_x)^2}\leq\frac{1}{\beta_x}$ when $\beta_x\leq 1$. The second inequality is because $\mathbb{E}\|\bar{v}_0-\nabla_x f(x_0,y_0)\|^2\leq\frac{\sigma^2}{nb_0}$ by Assumption 3.2 and Lemma B.2. Note that if we do not use Lemma B.7 on the last term we will get

$$\begin{aligned}
&\sum_{s=0}^{t-1}\mathbb{E}\|\bar{v}_s-\frac{1}{n}\sum_{i=1}^{n}\nabla_x f_i(x_s^{(i)},y_s^{(i)})\|^2\\
\leq\ &\frac{\sigma^2}{n\beta_x b_0}+\frac{2\beta_x\sigma^2 t}{n}+\frac{12L^2}{n^2\beta_x}\sum_{s=0}^{t-1}(\mathbb{E}\|X_s-\bar{X}_s\|_F^2+\mathbb{E}\|Y_s-\bar{Y}_s\|_F^2)\\
&+\frac{6L^2}{n\beta_x}\sum_{s=0}^{t-2}(1-(1-\beta_x)^{t-s-1})(\eta_x^2\mathbb{E}\|\bar{v}_s\|^2+\eta_y^2\mathbb{E}\|\bar{u}_s\|^2) \qquad \text{(B.32)}
\end{aligned}$$

Mimic above steps we can also prove the second conclusion in Lemma B.8. □

Lemma B.9. *The consensus error satisfies the following recursive relation*

$$\|X_{t+1}-\bar{X}_{t+1}\|_F^2\leq\frac{1+\lambda^2}{2}\|X_t-\bar{X}_t\|_F^2+\frac{2\lambda^2\eta_x^2}{1-\lambda^2}\|V_t-\bar{V}_t\|_F^2 \qquad \text{(B.33)}$$

$$\|Y_{t+1}-\bar{Y}_{t+1}\|_F^2\leq\frac{1+\lambda^2}{2}\|Y_t-\bar{Y}_t\|_F^2+\frac{2\lambda^2\eta_y^2}{1-\lambda^2}\|U_t-\bar{U}_t\|_F^2 \qquad \text{(B.34)}$$

Proof. Let $J = \frac{\mathbf{1}\mathbf{1}^T}{n}$. According to the update rule we have

$$\begin{aligned}
&\|X_{t+1} - \bar{X}_{t+1}\|_F^2 \\
&= \|(X_t - \eta_x V_t)W - (\bar{X}_t - \eta_x \bar{V}_t)\|_F^2 = \|(X_t - \bar{X}_t)(W-J) - \eta_x(V_t - \bar{V}_t)(W-J)\|_F^2 \\
&\leq \lambda^2 \|X_t - \bar{X}_t\|_F^2 + \lambda^2 \eta_x^2 \|V_t - \bar{V}_t\|_F^2 - 2\langle (X_t - \bar{X}_t)(W-J), \eta_x (V_t - \bar{V}_t)(W-J)\rangle \\
&\leq (\lambda^2 + \theta\lambda^2)\|X_t - \bar{X}_t\|_F^2 + (\frac{\lambda^2\eta_x^2}{\theta} + \lambda^2\eta_x^2)\|V_t - \bar{V}_t\|_F^2 \\
&\leq \frac{1+\lambda^2}{2}\|X_t - \bar{X}_t\|_F^2 + \frac{2\lambda^2\eta_x^2}{1-\lambda^2}\|V_t - \bar{V}_t\|_F^2
\end{aligned} \tag{B.35}$$

In the first inequality we use Assumption 3.4 and Lemma B.3. In the second inequality we use Young's inequality and θ is an arbitrary positive constant. Let $\theta = \frac{1-\lambda^2}{2\lambda^2}$ and we can get the last inequality. Similar to Eq. (B.35), we can obtain the following estimation

$$\begin{aligned}
\|Y_{t+1} - \bar{Y}_{t+1}\|_F^2 &= \|(Y_t + \eta_y U_t)W - (\bar{Y}_t + \eta_y \bar{U}_t)\|_F^2 \\
&\leq (\lambda^2 + \theta\lambda^2)\|Y_t - \bar{Y}_t\|_F^2 + (\frac{\lambda^2\eta_x^2}{\theta} + \lambda^2\eta_x^2)\|U_t - \bar{U}_t\|_F^2 \\
&\leq \frac{1+\lambda^2}{2}\|Y_t - \bar{Y}_t\|_F^2 + \frac{2\lambda^2\eta_y^2}{1-\lambda^2}\|U_t - \bar{U}_t\|_F^2
\end{aligned} \tag{B.36}$$

□

Lemma B.10. *For all $t \in \{0, 1, \cdots, T-1\}$ we have*

$$\begin{aligned}
\sum_{s=0}^{t}\mathbb{E}\|V_s - \bar{V}_s\|_F^2 &\leq \frac{2}{1-\lambda^2}\mathbb{E}\|V_0 - \bar{V}_0\|_F^2 + \frac{48\lambda^2L^2}{(1-\lambda^2)^2}\sum_{s=0}^{t}(\mathbb{E}\|X_s - \bar{X}_s\|_F^2 + \mathbb{E}\|Y_s - \bar{Y}_s\|_F^2) \\
&\quad + \frac{24n\lambda^2L^2}{(1-\lambda^2)^2}\sum_{s=0}^{t-1}\eta_y^2\mathbb{E}\|\bar{u}_s\|^2 + \frac{24n\lambda^2L^2}{(1-\lambda^2)^2}\sum_{s=0}^{t-1}\eta_x^2\mathbb{E}\|\bar{v}_s\|^2 \\
&\quad + \frac{8\lambda^2\beta_x^2}{(1-\lambda^2)^2}\sum_{s=0}^{t-1}\sum_{i=1}^{n}\mathbb{E}\|g_s^{(i)} - \nabla_x f_i(x_s^{(i)}, y_s^{(i)})\|^2 + \frac{6n\lambda^2\beta_x^2\sigma^2 t}{1-\lambda^2} \\
\sum_{s=0}^{t}\mathbb{E}\|U_s - \bar{U}_s\|_F^2 &\leq \frac{2}{1-\lambda^2}\mathbb{E}\|U_0 - \bar{U}_0\|_F^2 + \frac{48\lambda^2L^2}{(1-\lambda^2)^2}\sum_{s=0}^{t}(\mathbb{E}\|X_s - \bar{X}_s\|_F^2 + \mathbb{E}\|Y_s - \bar{Y}_s\|_F^2)
\end{aligned} \tag{B.37}$$

$$+\frac{24n\lambda^2L^2}{(1-\lambda^2)^2}\sum_{s=0}^{t-1}\eta_y^2\mathbb{E}\|\bar{u}_s\|^2+\frac{24n\lambda^2L^2}{(1-\lambda^2)^2}\sum_{s=0}^{t-1}\eta_x^2\mathbb{E}\|\bar{v}_s\|^2$$
$$+\frac{8\lambda^2\beta_y^2}{(1-\lambda^2)^2}\sum_{s=0}^{t-1}\sum_{i=1}^{n}\mathbb{E}\|h_s^{(i)}-\nabla_y f_i(x_s^{(i)},y_s^{(i)})\|^2+\frac{6n\lambda^2\beta_y^2\sigma^2 t}{1-\lambda^2} \quad \text{(B.38)}$$

Proof. By the definition of V_t, Assumption 3.4 and Lemma B.3, we have

$$\begin{aligned}
&\|V_{t+1}-\bar{V}_{t+1}\|_F^2\\
&=\|(V_t+G_{t+1}-G_t)W-(\bar{V}_t+\bar{G}_{t+1}-\bar{G}_t)\|_F^2\\
&=\|(V_t-\bar{V}_t)(W-J)+(G_{t+1}-G_t)(W-J)\|_F^2\\
&\leq\lambda^2\|V_t-\bar{V}_t\|_F^2+\lambda^2\|G_{t+1}-G_t\|_F^2+2\langle(V_t-\bar{V}_t)(W-J),(G_{t+1}-G_t)(W-J)\rangle \quad \text{(B.39)}
\end{aligned}$$

Review the definition of $g_t^{(i)}$

$$\begin{aligned}
g_{t+1}^{(i)}-g_t^{(i)}&=\nabla_x F_i(x_{t+1}^{(i)},y_{t+1}^{(i)};\xi_{t+1}^{(i)})-\nabla_x F_i(x_t^{(i)},y_t^{(i)};\xi_{t+1}^{(i)})-\beta_x(g_t^{(i)}-\nabla_x f_i(x_t^{(i)},y_t^{(i)}))\\
&\quad+\beta_x(\nabla_x F_i(x_t^{(i)},y_t^{(i)};\xi_{t+1}^{(i)})-\nabla_x f_i(x_t^{(i)},y_t^{(i)})) \quad \text{(B.40)}
\end{aligned}$$

and take expectation on $\xi_{t+1}^{(i)}$, then we have

$$\mathbb{E}[g_{t+1}^{(i)}-g_t^{(i)}]=\nabla_x f_i(x_{t+1}^{(i)},y_{t+1}^{(i)})-\nabla_x f_i(x_t^{(i)},y_t^{(i)})-\beta_x(g_t^{(i)}-\nabla_x f_i(x_t^{(i)},y_t^{(i)})) \quad \text{(B.41)}$$

Taking expectation on $\xi_{t+1}^{(i)}$ the last term of Eq. (B.39) can be bounded by

$$\begin{aligned}
&\mathbb{E}\langle(V_t-\bar{V}_t)(W-J),(G_{t+1}-G_t)(W-J)\rangle\\
&=\langle(V_t-\bar{V}_t)(W-J),\mathbb{E}[G_{t+1}-G_t](W-J)\rangle\leq\lambda\|V_t-\bar{V}_t\|_F\cdot\lambda\|\mathbb{E}[G_{t+1}-G_t]\|_F\\
&\leq\frac{1-\lambda^2}{4}\|V_t-\bar{V}_t\|_F^2+\frac{\lambda^4}{1-\lambda^2}\|\mathbb{E}[G_{t+1}-G_t]\|_F^2\\
&\leq\frac{1-\lambda^2}{4}\|V_t-\bar{V}_t\|_F^2+\frac{2\lambda^4}{1-\lambda^2}\sum_{i=1}^{n}\|\nabla_x f_i(x_{t+1}^{(i)},y_{t+1}^{(i)})-\nabla_x f_i(x_t^{(i)},y_t^{(i)})\|^2
\end{aligned}$$

$$
\begin{aligned}
&+\frac{2\lambda^4\beta_x^2}{1-\lambda^2}\sum_{i=1}^{n}\|g_t^{(i)}-\nabla_x f_i(x_t^{(i)},y_t^{(i)})\|^2\\
&\leq\frac{1-\lambda^2}{4}\|V_t-\bar{V}_t\|_F^2+\frac{2\lambda^4L^2}{1-\lambda^2}(\|X_{t+1}-X_t\|_F^2+\|Y_{t+1}-Y_t\|_F^2)\\
&+\frac{2\lambda^4\beta_x^2}{1-\lambda^2}\sum_{i=1}^{n}\|g_t^{(i)}-\nabla_x f_i(x_t^{(i)},y_t^{(i)})\|^2 \qquad \text{(B.42)}
\end{aligned}
$$

where the second inequality is resulted from Young's inequality, the third inequality is resulted from Cauchy-Schwartz inequality and the last inequality is resulted from Assumption 3.1. Besides, applying Cauchy-Schwartz inequality to Eq. (B.40) we have

$$
\begin{aligned}
&\mathbb{E}\|g_{t+1}^{(i)}-g_t^{(i)}\|^2\\
&\leq 3\mathbb{E}\|\nabla_x F_i(x_{t+1}^{(i)},y_{t+1}^{(i)};\xi_{t+1}^{(i)})-\nabla_x F_i(x_t^{(i)},y_t^{(i)};\xi_{t+1}^{(i)})\|^2+3\beta_x^2\mathbb{E}\|g_t^{(i)}-\nabla_x f_i(x_t^{(i)},y_t^{(i)})\|^2\\
&\quad+3\beta_x^2\mathbb{E}\|\nabla_x F_i(x_t^{(i)},y_t^{(i)};\xi_{t+1}^{(i)})-\nabla_x f_i(x_t^{(i)},y_t^{(i)})\|^2\\
&\leq 3L^2(\mathbb{E}\|x_{t+1}^{(i)}-x_t^{(i)}\|^2+\mathbb{E}\|y_{t+1}^{(i)}-y_t^{(i)}\|^2)+3\beta_x^2\mathbb{E}\|g_t^{(i)}-\nabla_x f_i(x_t^{(i)},y_t^{(i)})\|^2+3\beta_x^2\sigma^2 \qquad \text{(B.43)}
\end{aligned}
$$

where in the last inequality we use Assumption 3.1 and Assumption 3.2. Combining Eq. (B.39), (B.42) and (B.43) we can obtain

$$
\begin{aligned}
\mathbb{E}\|V_{t+1}-\bar{V}_{t+1}\|_F^2\leq\frac{1+\lambda^2}{2}\mathbb{E}\|V_t-\bar{V}_t\|_F^2+\frac{4\lambda^2L^2}{1-\lambda^2}(\mathbb{E}\|X_{t+1}-X_t\|_F^2+\mathbb{E}\|Y_{t+1}-Y_t\|_F^2)\\
+\frac{4\lambda^2\beta_x^2}{1-\lambda^2}\sum_{i=1}^{n}\mathbb{E}\|g_t^{(i)}-\nabla_x f_i(x_t^{(i)},y_t^{(i)})\|^2+3n\lambda^2\beta_x^2\sigma^2 \qquad \text{(B.44)}
\end{aligned}
$$

Then using Eq. (B.29) and (B.30) in above inequality we have

$$\begin{aligned}
&\mathbb{E}\|V_{t+1}-\bar{V}_{t+1}\|_F^2 \\
\leq\ &\frac{1+\lambda^2}{2}\mathbb{E}\|V_t-\bar{V}_t\|_F^2+\frac{12\lambda^2L^2}{1-\lambda^2}(\mathbb{E}\|X_{t+1}-\bar{X}_{t+1}\|_F^2+\mathbb{E}\|Y_{t+1}-\bar{Y}_{t+1}\|_F^2) \\
&+\frac{12\lambda^2L^2}{1-\lambda^2}(\mathbb{E}\|X_t-\bar{X}_t\|_F^2+\mathbb{E}\|Y_t-\bar{Y}_t\|_F^2)+\frac{12n\lambda^2L^2\eta_y^2}{1-\lambda^2}\mathbb{E}\|\bar{u}_t\|^2 \\
&+\frac{12n\lambda^2L^2\eta_x^2}{1-\lambda^2}\mathbb{E}\|\bar{v}_t\|^2+\frac{4\lambda^2\beta_x^2}{1-\lambda^2}\sum_{i=1}^{n}\mathbb{E}\|g_t^{(i)}-\nabla_x f_i(x_t^{(i)},y_t^{(i)})\|^2+3n\lambda^2\beta_x^2\sigma^2 \qquad \text{(B.45)}
\end{aligned}$$

By Lemma B.7, we can further achieve

$$\begin{aligned}
\sum_{s=0}^{t'}\mathbb{E}\|V_s-\bar{V}_s\|_F^2 \leq\ &\frac{2}{1-\lambda^2}\mathbb{E}\|V_0-\bar{V}_0\|_F^2+\frac{48\lambda^2L^2}{(1-\lambda^2)^2}\sum_{s=0}^{t'}(\mathbb{E}\|X_s-\bar{X}_s\|_F^2+\mathbb{E}\|Y_s-\bar{Y}_s\|_F^2) \\
&+\frac{24n\lambda^2L^2}{(1-\lambda^2)^2}\sum_{s=0}^{t'-1}\eta_y^2\mathbb{E}\|\bar{u}_s\|^2+\frac{24n\lambda^2L^2}{(1-\lambda^2)^2}\sum_{s=0}^{t'-1}\eta_x^2\mathbb{E}\|\bar{v}_s\|^2 \\
&+\frac{8\lambda^2\beta_x^2}{(1-\lambda^2)^2}\sum_{s=0}^{t'-1}\sum_{i=1}^{n}\mathbb{E}\|g_s^{(i)}-\nabla_x f_i(x_s^{(i)},y_s^{(i)})\|^2+\frac{6n\lambda^2\beta_x^2\sigma^2t'}{1-\lambda^2} \qquad \text{(B.46)}
\end{aligned}$$

for all $t' \in \{0,1,\cdots,T-1\}$. Here we should notice that term $\mathbb{E}\|X_{t+1}-\bar{X}_{t+1}\|_F^2$ in Eq. (B.45) is summed from $\mathbb{E}\|X_1-\bar{X}_1\|_F^2$ to $\mathbb{E}\|X_{t'}-\bar{X}_{t'}\|_F^2$, while term $\mathbb{E}\|X_t-\bar{X}_t\|_F^2$ is summed from $\mathbb{E}\|X_0-\bar{X}_0\|_F^2$ to $\mathbb{E}\|X_{t'-1}-\bar{X}_{t'-1}\|_F^2$. As $X_0=\bar{X}_0$, these two terms can be merged together. And it is the same with term $\mathbb{E}\|Y_{t+1}-\bar{Y}_{t+1}\|_F^2$. Mimic above steps and we can prove the conclusion for $\sum_{s=0}^{t'}\mathbb{E}\|U_s-\bar{U}_s\|_F^2$ in the similar way. □

Lemma B.11. *We can prove the gradient estimators $\bar{g}_t$ and $\bar{h}_t$ satisfy the following conclu-*

sion

$$\sum_{s=0}^{t}\sum_{i=1}^{n}\mathbb{E}\|g_s^{(i)}-\nabla_x f_i(x_s^{(i)},y_s^{(i)})\|^2 \le \frac{n\sigma^2}{\beta_x b_0}+2n\beta_x\sigma^2 t+\frac{12L^2}{\beta_x}\sum_{s=0}^{t}(\mathbb{E}\|X_s-\bar{X}_s\|_F^2+\mathbb{E}\|Y_s-\bar{Y}_s\|_F^2) + \frac{6nL^2}{\beta_x}\sum_{s=0}^{t-1}(\eta_x^2\mathbb{E}\|\bar{v}_s\|^2+\eta_y^2\mathbb{E}\|\bar{u}_s\|^2) \tag{B.47}$$

$$\sum_{s=0}^{t}\sum_{i=1}^{n}\mathbb{E}\|h_s^{(i)}-\nabla_y f_i(x_s^{(i)},y_s^{(i)})\|^2 \le \frac{n\sigma^2}{\beta_y b_0}+2n\beta_y\sigma^2 t+\frac{12L^2}{\beta_y}\sum_{s=0}^{t}(\mathbb{E}\|X_s-\bar{X}_s\|_F^2+\mathbb{E}\|Y_s-\bar{Y}_s\|_F^2) + \frac{6nL^2}{\beta_y}\sum_{s=0}^{t-1}(\eta_x^2\mathbb{E}\|\bar{v}_s\|^2+\eta_y^2\mathbb{E}\|\bar{u}_s\|^2) \tag{B.48}$$

for all $t \in \{0,1,\cdots,T-1\}$.

Proof. According to the definition of $g_t^{(i)}$ we have

$$\begin{aligned}
&g_t^{(i)}-\nabla_x f_i(x_t^{(i)},y_t^{(i)})\\
&=(1-\beta_x)(g_{t-1}^{(i)}-\nabla_x f_i(x_{t-1}^{(i)},y_{t-1}^{(i)}))+\beta_x(\nabla_x F_i(x_t^{(i)},y_t^{(i)};\xi_t^{(i)})-\nabla_x f_i(x_t^{(i)},y_t^{(i)}))\\
&\quad+(1-\beta_x)\Big(\nabla_x F_i(x_t^{(i)},y_t^{(i)};\xi_t^{(i)})-\nabla_x F_i(x_{t-1}^{(i)},y_{t-1}^{(i)};\xi_t^{(i)})\\
&\qquad+\nabla_x f_i(x_{t-1}^{(i)},y_{t-1}^{(i)})-\nabla_x f_i(x_t^{(i)},y_t^{(i)})\Big)
\end{aligned} \tag{B.49}$$

The last two terms of Eq. (B.49) is 0 after taking expectation of $\xi_t^{(i)}$. Hence we have

$$\begin{aligned}
&\mathbb{E}\|g_t^{(i)}-\nabla_x f_i(x_t^{(i)},y_t^{(i)})\|^2\\
&=(1-\beta_x)^2\mathbb{E}\|g_{t-1}^{(i)}-\nabla_x f_i(x_{t-1}^{(i)},y_{t-1}^{(i)})\|^2+\mathbb{E}\|\beta_x(\nabla_x F_i(x_t^{(i)},y_t^{(i)};\xi_t^{(i)})-\nabla_x f_i(x_t^{(i)},y_t^{(i)}))\\
&\quad+(1-\beta_x)\Big(\nabla_x F_i(x_t^{(i)},y_t^{(i)};\xi_t^{(i)})-\nabla_x F_i(x_{t-1}^{(i)},y_{t-1}^{(i)};\xi_t^{(i)})\\
&\qquad+\nabla_x f_i(x_{t-1}^{(i)},y_{t-1}^{(i)})-\nabla_x f_i(x_t^{(i)},y_t^{(i)})\Big)\|^2\\
&\le(1-\beta_x)^2\mathbb{E}\|g_{t-1}^{(i)}-\nabla_x f_i(x_{t-1}^{(i)},y_{t-1}^{(i)})\|^2+2\beta_x^2\mathbb{E}\|\nabla_x F_i(x_t^{(i)},y_t^{(i)};\xi_t^{(i)})-\nabla_x f_i(x_t^{(i)},y_t^{(i)})\|^2\\
&\quad+2(1-\beta_x)^2\mathbb{E}\|\nabla_x F_i(x_t^{(i)},y_t^{(i)};\xi_t^{(i)})-\nabla_x F_i(x_{t-1}^{(i)},y_{t-1}^{(i)};\xi_t^{(i)})\|^2
\end{aligned}$$

$$\leq (1-\beta_x)^2\mathbb{E}\|g_{t-1}^{(i)} - \nabla_x f_i(x_{t-1}^{(i)}, y_{t-1}^{(i)})\|^2 + 2\beta_x^2\sigma^2 + 2(1-\beta_x)^2L^2(\mathbb{E}\|x_t^{(i)} - x_{t-1}^{(i)}\|^2 + \mathbb{E}\|y_t^{(i)} - y_{t-1}^{(i)}\|^2) \tag{B.50}$$

where we use Cauchy-Schwartz inequality and Lemma B.1 in the first inequality and use Assumption 3.1 and Assumption 3.2 in the last inequality. Sum above inequality from $i = 1$ to n and we have

$$\sum_{i=1}^n \mathbb{E}\|g_t^{(i)} - \nabla_x f_i(x_t^{(i)}, y_t^{(i)})\|^2 \leq (1-\beta_x)^2 \sum_{i=1}^n \mathbb{E}\|g_{t-1}^{(i)} - \nabla_x f_i(x_{t-1}^{(i)}, y_{t-1}^{(i)})\|^2 + 2n\beta_x^2\sigma^2 + 2(1-\beta_x)^2L^2(\mathbb{E}\|X_t - X_{t-1}\|^2 + \mathbb{E}\|Y_t - Y_{t-1}\|^2) \tag{B.51}$$

Then by Eq. (B.29) and (B.30) we have

$$\begin{aligned}
&\sum_{i=1}^n \mathbb{E}\|g_t^{(i)} - \nabla_x f_i(x_t^{(i)}, y_t^{(i)})\|^2 \\
&\leq (1-\beta_x)^2 \sum_{i=1}^n \mathbb{E}\|g_{t-1}^{(i)} - \nabla_x f_i(x_{t-1}^{(i)}, y_{t-1}^{(i)})\|^2 + 2n\beta_x^2\sigma^2 \\
&\quad + 6(1-\beta_x)^2L^2(\mathbb{E}\|X_t - \bar{X}_t\|_F^2 + \mathbb{E}\|Y_t - \bar{Y}_t\|_F^2 + \mathbb{E}\|X_{t-1} - \bar{X}_{t-1}\|_F^2 \\
&\quad + \mathbb{E}\|Y_{t-1} - \bar{Y}_{t-1}\|_F^2) + 6n(1-\beta_x)^2L^2(\eta_x^2\mathbb{E}\|\bar{v}_{t-1}\|^2 + \eta_y^2\mathbb{E}\|\bar{u}_{t-1}\|^2)
\end{aligned} \tag{B.52}$$

Applying Lemma B.7 to Eq. (B.52), similar to Eq. (B.46), we can obtain

$$\begin{aligned}
&\sum_{s=0}^{t}\sum_{i=1}^{n}\mathbb{E}\|g_s^{(i)}-\nabla_x f_i(x_s^{(i)},y_s^{(i)})\|^2\\
\le&\frac{1}{\beta_x}\sum_{i=1}^{n}\mathbb{E}\|g_0^{(i)}-\nabla_x f_i(x_0^{(i)},y_0^{(i)})\|^2+\frac{12L^2}{\beta_x}\sum_{s=0}^{t}(\mathbb{E}\|X_s-\bar{X}_s\|_F^2+\mathbb{E}\|Y_s-\bar{Y}_s\|_F^2)\\
&+\frac{6nL^2}{\beta_x}\sum_{s=0}^{t-1}(\eta_x^2\mathbb{E}\|\bar{v}_s\|^2+\eta_y^2\mathbb{E}\|\bar{u}_s\|^2)+2n\beta_x\sigma^2 t\\
\le&\frac{n\sigma^2}{\beta_x b_0}+2n\beta_x\sigma^2 t+\frac{12L^2}{\beta_x}\sum_{s=0}^{t}(\mathbb{E}\|X_s-\bar{X}_s\|_F^2+\mathbb{E}\|Y_s-\bar{Y}_s\|_F^2)\\
&+\frac{6nL^2}{\beta_x}\sum_{s=0}^{t-1}(\eta_x^2\mathbb{E}\|\bar{v}_s\|^2+\eta_y^2\mathbb{E}\|\bar{u}_s\|^2)
\end{aligned}\tag{B.53}$$

for all $t\in\{0,1,\cdots,T-1\}$. Here the last inequality is derived by $\mathbb{E}\|g_0^{(i)}-\nabla_x f_i(x_0^{(i)},y_0^{(i)})\|^2\le\frac{\sigma^2}{b_0}$ due to Lemma B.2. The estimation of $h_t^{(i)}$ can be achieved in the same way as above. □

Lemma B.12. *Let $\eta_x\le\frac{(1-\lambda)^2}{500L}$ and $\eta_y\le\frac{(1-\lambda)^2}{500L}$. The consensus error can be bounded by*

$$\begin{aligned}
&\sum_{s=0}^{t}(\mathbb{E}\|X_s-\bar{X}_s\|_F^2+\mathbb{E}\|Y_s-\bar{Y}_s\|_F^2)\\
\le&\frac{16\lambda^2\eta_x^2}{(1-\lambda^2)^3}\mathbb{E}\|V_0-\bar{V}_0\|_F^2+\frac{16\lambda^2\eta_y^2}{(1-\lambda^2)^3}\mathbb{E}\|U_0-\bar{U}_0\|_F^2+\frac{576n\lambda^4L^2(\eta_x^2+\eta_y^2)}{(1-\lambda^2)^4}\sum_{s=0}^{t-2}(\eta_x^2\mathbb{E}\|\bar{v}_s\|^2\\
&+\eta_y^2\mathbb{E}\|\bar{u}_s\|^2)+\frac{64n\lambda^4(\beta_x\eta_x^2+\beta_y\eta_y^2)\sigma^2}{(1-\lambda^2)^4 b_0}+\frac{196n\lambda^4(\beta_x^2\eta_x^2+\beta_y^2\eta_y^2)\sigma^2 t}{(1-\lambda^2)^4}
\end{aligned}\tag{B.54}$$

for all $t\in\{0,1,\cdots,T\}$.

Proof. Combining Lemma B.7 and Lemma B.9, for all $t\in\{0,1,\cdots,T\}$ we have

$$\sum_{s=0}^{t}\|X_s-\bar{X}_s\|_F^2\le\frac{4\lambda^2\eta_x^2}{(1-\lambda^2)^2}\sum_{s=0}^{t-1}\|V_s-\bar{V}_s\|_F^2\tag{B.55}$$

Substitute the right side with Lemma B.10 we have

$$\begin{aligned}\sum_{s=0}^{t}\mathbb{E}\|X_s-\bar{X}_s\|_F^2 \leq\ & \frac{8\lambda^2\eta_x^2}{(1-\lambda^2)^3}\mathbb{E}\|V_0-\bar{V}_0\|_F^2+\frac{192\lambda^4L^2\eta_x^2}{(1-\lambda^2)^4}\sum_{s=0}^{t-1}(\mathbb{E}\|X_s-\bar{X}_s\|_F^2+\mathbb{E}\|Y_s-\bar{Y}_s\|_F^2)\\ &+\frac{96n\lambda^4L^2\eta_x^2}{(1-\lambda^2)^4}\sum_{s=0}^{t-2}\eta_y^2\mathbb{E}\|\bar{u}_s\|^2+\frac{96n\lambda^4L^2\eta_x^2}{(1-\lambda^2)^4}\sum_{s=0}^{t-2}\eta_x^2\mathbb{E}\|\bar{v}_s\|^2\\ &+\frac{32\lambda^4\beta_x^2\eta_x^2}{(1-\lambda^2)^4}\sum_{s=0}^{t-2}\sum_{i=1}^{n}\mathbb{E}\|g_s^{(i)}-\nabla_x f_i(x_s^{(i)},y_s^{(i)})\|^2+\frac{24n\lambda^4\beta_x^2\eta_x^2\sigma^2(t-1)}{(1-\lambda^2)^3}\end{aligned}\tag{B.56}$$

Apply Lemma B.11 and we get

$$\begin{aligned}&\sum_{s=0}^{t}\mathbb{E}\|X_s-\bar{X}_s\|_F^2\\ \leq\ & \frac{8\lambda^2\eta_x^2}{(1-\lambda^2)^3}\mathbb{E}\|V_0-\bar{V}_0\|_F^2+\frac{192\lambda^4L^2\eta_x^2}{(1-\lambda^2)^4}\sum_{s=0}^{t-1}(\mathbb{E}\|X_s-\bar{X}_s\|_F^2+\mathbb{E}\|Y_s-\bar{Y}_s\|_F^2)\\ &+\frac{96n\lambda^4L^2\eta_x^2}{(1-\lambda^2)^4}\sum_{s=0}^{t-2}(\eta_x^2\mathbb{E}\|\bar{v}_s\|^2+\eta_y^2\mathbb{E}\|\bar{u}_s\|^2)+\frac{32n\lambda^4\beta_x\eta_x^2\sigma^2}{(1-\lambda^2)^4b_0}+\frac{64n\lambda^4\beta_x^3\eta_x^2\sigma^2(t-2)}{(1-\lambda^2)^4}\\ &+\frac{24n\lambda^4\beta_x^2\eta_x^2\sigma^2(t-1)}{(1-\lambda^2)^3}+\frac{384\lambda^4\beta_xL^2\eta_x^2}{(1-\lambda^2)^4}\sum_{s=0}^{t-2}(\mathbb{E}\|X_s-\bar{X}_s\|_F^2+\mathbb{E}\|Y_s-\bar{Y}_s\|_F^2)\\ &+\frac{192n\lambda^4\beta_xL^2\eta_x^2}{(1-\lambda^2)^4}\sum_{s=0}^{t-3}(\eta_x^2\mathbb{E}\|\bar{v}_s\|^2+\eta_y^2\mathbb{E}\|\bar{u}_s\|^2)\\ \leq\ & \frac{8\lambda^2\eta_x^2}{(1-\lambda^2)^3}\mathbb{E}\|V_0-\bar{V}_0\|_F^2+\frac{576\lambda^4L^2\eta_x^2}{(1-\lambda^2)^4}\sum_{s=0}^{t-1}(\mathbb{E}\|X_s-\bar{X}_s\|_F^2+\mathbb{E}\|Y_s-\bar{Y}_s\|_F^2)\\ &+\frac{288n\lambda^4L^2\eta_x^2}{(1-\lambda^2)^4}\sum_{s=0}^{t-2}(\eta_x^2\mathbb{E}\|\bar{v}_s\|^2+\eta_y^2\mathbb{E}\|\bar{u}_s\|^2)+\frac{32n\lambda^4\beta_x\eta_x^2\sigma^2}{(1-\lambda^2)^4b_0}+\frac{98n\lambda^4\beta_x^2\eta_x^2\sigma^2t}{(1-\lambda^2)^4}\end{aligned}\tag{B.57}$$

where we use $\beta_x \leq 1$ to simplify the equation. Similarly, we have

$$\begin{aligned}
&\sum_{s=0}^{t} \mathbb{E}\|Y_s - \bar{Y}_s\|_F^2 \\
&\leq \frac{8\lambda^2\eta_y^2}{(1-\lambda^2)^3}\mathbb{E}\|U_0 - \bar{U}_0\|_F^2 + \frac{576\lambda^4 L^2\eta_y^2}{(1-\lambda^2)^4}\sum_{s=0}^{t-1}(\mathbb{E}\|X_s - \bar{X}_s\|_F^2 + \mathbb{E}\|Y_s - \bar{Y}_s\|_F^2) \\
&\quad + \frac{288n\lambda^4 L^2\eta_y^2}{(1-\lambda^2)^4}\sum_{s=0}^{t-2}(\eta_x^2\mathbb{E}\|\bar{v}_s\|^2 + \eta_y^2\mathbb{E}\|\bar{u}_s\|^2) + \frac{32n\lambda^4\beta_y\eta_y^2\sigma^2}{(1-\lambda^2)^4 b_0} + \frac{98n\lambda^4\beta_y^2\eta_y^2\sigma^2 t}{(1-\lambda^2)^4}
\end{aligned} \tag{B.58}$$

Add Eq. (B.57) and (B.57) together. Then we have

$$\begin{aligned}
&\sum_{s=0}^{t}(\mathbb{E}\|X_s - \bar{X}_s\|_F^2 + \mathbb{E}\|Y_s - \bar{Y}_s\|_F^2) \\
&\leq \frac{8\lambda^2\eta_x^2}{(1-\lambda^2)^3}\mathbb{E}\|V_0 - \bar{V}_0\|_F^2 + \frac{8\lambda^2\eta_y^2}{(1-\lambda^2)^3}\mathbb{E}\|U_0 - \bar{U}_0\|_F^2 + \frac{576\lambda^4 L^2(\eta_x^2+\eta_y^2)}{(1-\lambda^2)^4}\sum_{s=0}^{t-1}(\mathbb{E}\|X_s - \bar{X}_s\|_F^2 \\
&\quad + \mathbb{E}\|Y_s - \bar{Y}_s\|_F^2) + \frac{288n\lambda^4 L^2(\eta_x^2+\eta_y^2)}{(1-\lambda^2)^4}\sum_{s=0}^{t-2}(\eta_x^2\mathbb{E}\|\bar{v}_s\|^2 + \eta_y^2\mathbb{E}\|\bar{u}_s\|^2) \\
&\quad + \frac{32n\lambda^4(\beta_x\eta_x^2+\beta_y\eta_y^2)\sigma^2}{(1-\lambda^2)^4 b_0} + \frac{98n\lambda^4(\beta_x^2\eta_x^2+\beta_y^2\eta_y^2)\sigma^2 t}{(1-\lambda^2)^4}
\end{aligned} \tag{B.59}$$

As $\lambda < 1$, when $\eta_x \leq \frac{(1-\lambda)^2}{500L}$ and $\eta_y \leq \frac{(1-\lambda)^2}{500L}$ it satisfies

$$\frac{576\lambda^4 L^2(\eta_x^2+\eta_y^2)}{(1-\lambda^2)^4} \leq \frac{1}{2} \tag{B.60}$$

Therefore, Eq. (B.59) implies

$$\begin{aligned}
&\sum_{s=0}^{t}(\mathbb{E}\|X_s - \bar{X}_s\|_F^2 + \mathbb{E}\|Y_s - \bar{Y}_s\|_F^2) \\
&\leq \frac{16\lambda^2\eta_x^2}{(1-\lambda^2)^3}\mathbb{E}\|V_0 - \bar{V}_0\|_F^2 + \frac{16\lambda^2\eta_y^2}{(1-\lambda^2)^3}\mathbb{E}\|U_0 - \bar{U}_0\|_F^2 + \frac{576n\lambda^4 L^2(\eta_x^2+\eta_y^2)}{(1-\lambda^2)^4}\sum_{s=0}^{t-2}(\eta_x^2\mathbb{E}\|\bar{v}_s\|^2 \\
&\quad + \eta_y^2\mathbb{E}\|\bar{u}_s\|^2) + \frac{64n\lambda^4(\beta_x\eta_x^2+\beta_y\eta_y^2)\sigma^2}{(1-\lambda^2)^4 b_0} + \frac{196n\lambda^4(\beta_x^2\eta_x^2+\beta_y^2\eta_y^2)\sigma^2 t}{(1-\lambda^2)^4}
\end{aligned} \tag{B.61}$$

which reaches the conclusion of Lemma B.12. □

B.3 Proof of main Theorems

Now we will move forward to the main Theorems in our paper. Here we revise some constant coefficients in the statement, but it does not actually affect the result in our convergence analysis.

Theorem B.1. *(Restatement of Theorem 3.1) Let Assumptions 3.1 to 3.5 hold. When parameters* $\beta_x = \frac{\varepsilon \min\{1,n\varepsilon\}}{20}$, $\beta_y = \frac{\varepsilon \min\{1,n\varepsilon\}}{500\kappa^2}$, $\eta_x = \frac{(1-\lambda)^2 \min\{1,n\varepsilon\}}{15000\kappa^3 L}$, $\eta_y = \frac{(1-\lambda)^2 \min\{1,n\varepsilon\}}{1500\kappa L}$, $b_0 = \frac{400\kappa}{\min\{1,n\varepsilon\}}$, $T = \frac{30000\kappa^3\varepsilon^{-2}}{(1-\lambda)^2\min\{1,n\varepsilon\}}$, *our Algorithm 2 satisfies*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\Phi(\bar{x}_t)\|^2 \le L(\Phi(x_0)-\Phi^*)\varepsilon^2 + \sigma^2\varepsilon^2 + L^2\delta_0\varepsilon^2 + \frac{\varepsilon^2}{n}\sum_{i=1}^{n}\|\nabla_x f_i(x_0,y_0)\|^2 + \frac{\varepsilon^2}{n}\sum_{i=1}^{n}\|\nabla_y f_i(x_0,y_0)\|^2 \tag{B.62}$$

Proof. Since $\Phi(x)$ is $(\kappa L + L)$-smooth we have

$$\begin{aligned}
\Phi(\bar{x}_t) &\le \Phi(\bar{x}_{t-1}) - \eta_x\langle \bar{v}_{t-1}, \nabla\Phi(\bar{x}_{t-1})\rangle + \eta_x^2\kappa L\|\bar{v}_{t-1}\|^2 \\
&= \Phi(\bar{x}_{t-1}) - \frac{\eta_x}{2}\|\bar{v}_{t-1}\|^2 - \frac{\eta_x}{2}\|\nabla\Phi(\bar{x}_{t-1})\|^2 + \frac{\eta_x}{2}\|\bar{v}_{t-1}-\nabla\Phi(\bar{x}_{t-1})\|^2 + \eta_x^2\kappa L\|\bar{v}_{t-1}\|^2 \\
&\le \Phi(\bar{x}_{t-1}) - \frac{\eta_x}{2}\|\nabla\Phi(\bar{x}_{t-1})\|^2 - (\frac{\eta_x}{2} - \eta_x^2\kappa L)\|\bar{v}_{t-1}\|^2 + \eta_x\|\bar{v}_{t-1}-\nabla_x f(\bar{x}_{t-1},\bar{y}_{t-1})\|^2 \\
&\quad + \eta_x\|\nabla\Phi(\bar{x}_{t-1})-\nabla_x f(\bar{x}_{t-1},\bar{y}_{t-1})\|^2
\end{aligned} \tag{B.63}$$

where the last inequality is caused by Cauchy-Schwartz inequality. As we have $\nabla\Phi(\bar{x}_{t-1}) = \nabla_x f(\bar{x}_{t-1},\hat{y}_{t-1})$, by Assumption 3.1 the last term satisfies

$$\|\nabla\Phi(\bar{x}_{t-1})-\nabla_x f(\bar{x}_{t-1},\bar{y}_{t-1})\|^2 \le L^2\|\hat{y}_{t-1}-\bar{y}_{t-1}\|^2 = L^2\delta_{t-1} \tag{B.64}$$

Besides, according to Cauchy-Schwartz inequality we also have

$$\begin{aligned}
&\|\bar{v}_{t-1} - \nabla_x f(\bar{x}_{t-1}, \bar{y}_{t-1})\|^2 \\
&\leq 2\|\bar{v}_{t-1} - \frac{1}{n}\sum_{i=1}^{n} \nabla_x f_i(x_{t-1}^{(i)}, y_{t-1}^{(i)})\|^2 + 2\|\frac{1}{n}\sum_{i=1}^{n} \nabla_x f_i(x_{t-1}^{(i)}, y_{t-1}^{(i)}) - \nabla_x f(\bar{x}_{t-1}, \bar{y}_{t-1})\|^2 \\
&\leq 2\|\bar{v}_{t-1} - \frac{1}{n}\sum_{i=1}^{n} \nabla_x f_i(x_{t-1}^{(i)}, y_{t-1}^{(i)})\|^2 + \frac{2L^2}{n}(\|X_{t-1} - \bar{X}_{t-1}\|_F^2 + \|Y_{t-1} - \bar{Y}_{t-1}\|_F^2) \qquad \text{(B.65)}
\end{aligned}$$

Combine Eq. (B.63), (B.64), (B.65) and rearrange the inequality

$$\begin{aligned}
\|\nabla\Phi(\bar{x}_{t-1})\|^2 \leq \frac{2(\Phi(\bar{x}_{t-1}) - \Phi(\bar{x}_t))}{\eta_x} - (1 - 2\kappa L\eta_x)\|\bar{v}_{t-1}\|^2 + 2L^2\delta_{t-1} + \frac{4L^2}{n}(\|X_{t-1} - \bar{X}_{t-1}\|_F^2 \\
+ \|Y_{t-1} - \bar{Y}_{t-1}\|_F^2) + 4\|\bar{v}_{t-1} - \frac{1}{n}\sum_{i=1}^{n} \nabla_x f_i(x_{t-1}^{(i)}, y_{t-1}^{(i)})\|^2 \qquad \text{(B.66)}
\end{aligned}$$

Telescoping and taking expectation on Eq. (B.66) we have

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla\Phi(\bar{x}_t)\|^2 \\
&\leq \frac{2(\Phi(x_0) - \mathbb{E}\Phi(\bar{x}_T))}{\eta_x T} - \frac{(1 - 2\kappa L\eta_x)}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\bar{v}_t\|^2 + \frac{2L^2}{T}\sum_{t=0}^{T-1} \mathbb{E}\delta_t + \frac{4L^2}{nT}\sum_{t=0}^{T-1}(\mathbb{E}\|X_t - \bar{X}_t\|_F^2 \\
&+ \mathbb{E}\|Y_t - \bar{Y}_t\|_F^2) + \frac{4}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\bar{v}_t - \frac{1}{n}\sum_{i=1}^{n} \nabla_x f_i(x_t^{(i)}, y_t^{(i)})\|^2 \qquad \text{(B.67)}
\end{aligned}$$

Applying Assumption 3.3, Lemma B.5 and Lemma B.8 we have

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla\Phi(\bar{x}_t)\|^2 \\
&\leq \frac{2(\Phi(x_0) - \Phi^*)}{\eta_x T} - (1 - 2\kappa L\eta_x - \frac{40\kappa^4\eta_x^2}{\eta_y^2})\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\bar{v}_t\|^2 + \frac{8\kappa L^2\delta_0}{TL\eta_y} \\
&+ \frac{148\kappa^2 L^2}{nT}\sum_{t=0}^{T-1}(\mathbb{E}\|X_t - \bar{X}_t\|_F^2 + \mathbb{E}\|Y_t - \bar{Y}_t\|_F^2) + \frac{4}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\bar{v}_t - \frac{1}{n}\sum_{i=1}^{n} \nabla_x f_i(x_t^{(i)}, y_t^{(i)})\|^2 \\
&+ \frac{36\kappa L\eta_y}{T}\sum_{t=1}^{T-1}(1 - \frac{\mu\eta_y}{4})^{T-t-1}\sum_{s=0}^{t-1}\|\bar{u}_s - \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x_s^{(i)}, y_s^{(i)})\|^2
\end{aligned}$$

$$
\begin{aligned}
&-\frac{24\kappa^2}{5T}\sum_{t=0}^{T-1}(1-(1-\frac{\mu\eta_y}{4})^{T-t})\mathbb{E}\|\bar{u}_t\|^2 \\
&\leq \frac{2(\Phi(x_0)-\Phi^*)}{\eta_x T}-(1-2\kappa L\eta_x-\frac{40\kappa^4\eta_x^2}{\eta_y^2})\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\bar{v}_t\|^2+\frac{8\kappa L^2\delta_0}{TL\eta_y}+\frac{4\sigma^2}{nb_0T}(\frac{1}{\beta_x}+\frac{36\kappa^2}{\beta_y}) \\
&+\frac{8\sigma^2}{n}(\beta_x+36\kappa^2\beta_y)+\frac{4L^2}{nT}(47\kappa^2+\frac{12}{n\beta_x}+\frac{432\kappa^2}{n\beta_y})\sum_{t=0}^{T-1}(\mathbb{E}\|X_t-\bar{X}_t\|_F^2+\mathbb{E}\|Y_t-\bar{Y}_t\|_F^2) \\
&+\frac{24L^2}{n\beta_x T}\sum_{t=0}^{T-1}(1-(1-\beta_x)^{T-t})(\eta_x^2\mathbb{E}\|\bar{v}_t\|^2+\eta_y^2\mathbb{E}\|\bar{u}_t\|^2)+\frac{864\kappa^2L^2}{n\beta_y T}\sum_{t=0}^{T-1}(1-(1-\frac{\mu\eta_y}{4})^{T-t}) \\
&\cdot(\eta_x^2\mathbb{E}\|\bar{v}_t\|^2+\eta_y^2\mathbb{E}\|\bar{u}_t\|^2)-\frac{24\kappa^2}{5T}\sum_{t=0}^{T-1}(1-(1-\frac{\mu\eta_y}{4})^{T-t})\mathbb{E}\|\bar{u}_t\|^2 \qquad \text{(B.68)}
\end{aligned}
$$

where we use Eq. (B.32) in the last inequality. As

$$
\frac{1}{\beta_x}(1-(1-\beta_x)^{T-t})=\sum_{s=0}^{T-t-1}(1-\beta_x)^s \qquad \text{(B.69)}
$$

we know Eq. (B.69) is increasing when β_x is decreasing. Hence $\frac{1}{\beta_x}(1-(1-\beta_x)^{T-t})\leq \frac{300\kappa^2}{(1-\lambda)^2\beta_x}(1-(1-\frac{(1-\lambda)^2\beta_x}{300\kappa^2})^{T-t})$. According to the definition of β_x and η_y, we have $\frac{(1-\lambda)^2\beta_x}{300\kappa^2}\leq \frac{\mu\eta_y}{4}$ and

$$
\frac{24L^2}{n\beta_x T}(1-(1-\beta_x)^{T-t})\leq \frac{7200L^2\kappa^2}{n(1-\lambda)^2\beta_x T}(1-(1-\frac{\mu\eta_y}{4})^{T-t}) \qquad \text{(B.70)}
$$

Therefore, using the definition of β_x, β_y and η_y we obtain

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\Phi(\bar{x}_t)\|^2 \\
&\leq \frac{2(\Phi(x_0)-\Phi^*)}{\eta_x T}-(1-2\kappa L\eta_x-\frac{40\kappa^4\eta_x^2}{\eta_y^2})\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\bar{v}_t\|^2+\frac{8\kappa L^2\delta_0}{TL\eta_y}+\frac{4\sigma^2}{nb_0T}(\frac{1}{\beta_x}+\frac{36\kappa^2}{\beta_y}) \\
&+\frac{8\sigma^2}{n}(\beta_x+36\kappa^2\beta_y)+\frac{4L^2}{nT}(47\kappa^2+\frac{12}{n\beta_x}+\frac{432\kappa^2}{n\beta_y})\sum_{t=0}^{T-1}(\mathbb{E}\|X_t-\bar{X}_t\|_F^2+\mathbb{E}\|Y_t-\bar{Y}_t\|_F^2) \\
&+(\frac{24L^2\eta_x^2}{n\beta_x}+\frac{864\kappa^2L^2\eta_x^2}{n\beta_y})\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\bar{v}_t\|^2-\frac{\kappa L\eta_y}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\bar{u}_t\|^2 \qquad \text{(B.71)}
\end{aligned}
$$

Besides, according to Lemma B.12 we have

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\Phi(\bar{x}_t)\|^2\\
\leq\ &\frac{2(\Phi(x_0)-\Phi^*)}{\eta_x T}-(1-2\kappa L\eta_x-\frac{40\kappa^4\eta_x^2}{\eta_y^2})\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\bar{v}_t\|^2+\frac{8\kappa L^2\delta_0}{TL\eta_y}+\frac{4\sigma^2}{nb_0T}(\frac{1}{\beta_x}+\frac{36\kappa^2}{\beta_y})\\
&+\frac{8\sigma^2}{n}(\beta_x+36\kappa^2\beta_y)+\frac{4L^2}{nT}(47\kappa^2+\frac{12}{n\beta_x}+\frac{432\kappa^2}{n\beta_y})\Big(\frac{16\lambda^2\eta_x^2}{(1-\lambda^2)^3}\mathbb{E}\|V_0-\bar{V}_0\|_F^2\\
&+\frac{16\lambda^2\eta_y^2}{(1-\lambda^2)^3}\mathbb{E}\|U_0-\bar{U}_0\|_F^2+\frac{64n\lambda^4(\beta_x\eta_x^2+\beta_y\eta_y^2)\sigma^2}{(1-\lambda^2)^4b_0}+\frac{196n\lambda^4(\beta_x^2\eta_x^2+\beta_y^2\eta_y^2)\sigma^2T}{(1-\lambda^2)^4}\Big)\\
&+\frac{4L^2}{nT}(47\kappa^2+\frac{12}{n\beta_x}+\frac{432\kappa^2}{n\beta_y})\frac{576n\lambda^4L^2(\eta_x^2+\eta_y^2)}{(1-\lambda^2)^4}\sum_{t=0}^{T-1}(\eta_x^2\mathbb{E}\|\bar{v}_t\|^2+\eta_y^2\mathbb{E}\|\bar{u}_t\|^2)\\
&+(\frac{24L^2\eta_x^2}{n\beta_x}+\frac{864\kappa^2L^2\eta_x^2}{n\beta_y})\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\bar{v}_t\|^2-\frac{\kappa L\eta_y}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\bar{u}_t\|^2
\end{aligned}
\tag{B.72}
$$

When β_x, β_y, η_x and η_y are defined as Theorem B.1, we have

$$
\frac{4L^2}{nT}(47\kappa^2+\frac{12}{n\beta_x}+\frac{432\kappa^2}{n\beta_y})\frac{576n\lambda^4L^2(\eta_x^2+\eta_y^2)}{(1-\lambda^2)^4}\eta_y^2\leq\frac{\kappa L\eta_y}{2T}
\tag{B.73}
$$

and

$$
\begin{aligned}
1-2\kappa L\eta_x-\frac{40\kappa^4\eta_x^2}{\eta_y^2}-\frac{24L^2\eta_x^2}{n\beta_x}-\frac{864\kappa^2L^2\eta_x^2}{n\beta_y}&\\
-\frac{4L^2}{n}(47\kappa^2+\frac{12}{n\beta_x}+\frac{432\kappa^2}{n\beta_y})\frac{576n\lambda^4L^2(\eta_x^2+\eta_y^2)}{(1-\lambda^2)^4}\eta_x^2&\geq\frac{2}{5}
\end{aligned}
\tag{B.74}
$$

Thus, we obtain

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\Phi(\bar{x}_t)\|^2 \\
&\leq \frac{2(\Phi(x_0)-\Phi^*)}{\eta_x T}+\frac{8\kappa L^2\delta_0}{TL\eta_y}+\frac{4\sigma^2}{nb_0T}(\frac{1}{\beta_x}+\frac{36\kappa^2}{\beta_y})+\frac{8\sigma^2}{n}(\beta_x+36\kappa^2\beta_y) \\
&\quad+\frac{4L^2}{nT}(47\kappa^2+\frac{12}{n\beta_x}+\frac{432\kappa^2}{n\beta_y})\Big(\frac{16\lambda^2\eta_x^2}{(1-\lambda^2)^3}\mathbb{E}\|V_0-\bar{V}_0\|_F^2+\frac{16\lambda^2\eta_y^2}{(1-\lambda^2)^3}\mathbb{E}\|U_0-\bar{U}_0\|_F^2 \\
&\quad+\frac{64n\lambda^4(\beta_x\eta_x^2+\beta_y\eta_y^2)\sigma^2}{(1-\lambda^2)^4 b_0}+\frac{196n\lambda^4(\beta_x^2\eta_x^2+\beta_y^2\eta_y^2)\sigma^2 T}{(1-\lambda^2)^4}\Big)
\end{aligned} \tag{B.75}$$

By Assumption 3.4 and Cauchy-Schwartz inequality we also have

$$\mathbb{E}\|V_0-\bar{V}_0\|_F^2=\mathbb{E}\|G_0(W-J)\|_F^2\leq\lambda^2\mathbb{E}\|G_0\|_F^2\leq\frac{2n\lambda^2\sigma^2}{b_0}+2\lambda^2\sum_{i=1}^{n}\|\nabla_x f_i(x_0,y_0)\|^2 \tag{B.76}$$

Similarly, we have

$$\mathbb{E}\|U_0-\bar{U}_0\|_F^2\leq\frac{2n\lambda^2\sigma^2}{b_0}+2\lambda^2\sum_{i=1}^{n}\|\nabla_y f_i(x_0,y_0)\|^2 \tag{B.77}$$

Combine above three inequalities and substitute the parameters with their definitions. We achieve

$$\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\Phi(\bar{x}_t)\|^2 \leq L(\Phi(x_0)-\Phi^*)\varepsilon^2+L^2\delta_0\varepsilon^2+\sigma^2\varepsilon^2+\frac{\varepsilon^2}{n}\sum_{i=1}^{n}\|\nabla_x f_i(x_0,y_0)\|^2 \\
+\frac{\varepsilon^2}{n}\sum_{i=1}^{n}\|\nabla_y f_i(x_0,y_0)\|^2
\end{aligned} \tag{B.78}$$

where we use following inequalities for simplification.

$$\beta_x \geq \beta_y,\ \frac{144\kappa^2}{n\beta_y b_0 T} \leq \frac{144\kappa^2 \cdot 500\kappa^2(\min\{1,n\varepsilon\})^2\varepsilon^2}{n\varepsilon \min\{1,n\varepsilon\}400\kappa \cdot 30000\kappa^3} \leq \frac{3\varepsilon^2}{500}$$
$$4L^2(47\kappa^2 + \frac{12}{n\beta_x} + \frac{432\kappa^2}{n\beta_y}) \leq 200L^2\kappa^2 + \frac{1800L^2\kappa^2}{n\beta_y}$$
$$\frac{8\beta_x}{n} \leq \frac{8\varepsilon \cdot n\varepsilon}{20n} = \frac{2\varepsilon^2}{5},\ \frac{288\kappa^2\beta_y}{n} \leq \frac{288\kappa^2\varepsilon \cdot n\varepsilon}{500n\kappa^2} \leq \frac{288\varepsilon^2}{500}$$
$$\frac{L^2\beta_x\eta_x^2}{(1-\lambda)^4 b_0 T} \leq \frac{\varepsilon(\min\{1,n\varepsilon\})^5\varepsilon^2}{20 \cdot 400\kappa \cdot 30000\kappa^3(15000\kappa^3)^2},\ \frac{L^2\beta_y\eta_y^2}{(1-\lambda)^4 b_0 T} \leq \frac{\varepsilon(\min\{1,n\varepsilon\})^5\varepsilon^2}{500 \cdot 400\kappa \cdot 30000\kappa^3(1500\kappa)^2}$$
$$\frac{L^2\beta_x^2\eta_x^2}{(1-\lambda)^4} \leq \frac{\varepsilon^2(\min\{1,n\varepsilon\})^4}{400(15000\kappa^3)^2},\ \frac{L^2\beta_y^2\eta_y^2}{(1-\lambda)^4} \leq \frac{\varepsilon^2(\min\{1,n\varepsilon\})^4}{(500\kappa^2)^2(1500\kappa)^2} \tag{B.79}$$

□

Theorem B.2. *(Restatement of Theorem 3.2) Let Assumptions 3.1 to 3.5 hold. We set the parameters as* $T = \frac{30000\kappa^3 T_0}{(1-\lambda)^2}$, $\beta_x = \frac{n^{1/3}}{20T_0^{2/3}}$, $\beta_y = \frac{n^{1/3}}{500\kappa^2 T_0^{2/3}}$, $\eta_x = \frac{(1-\lambda)^2 n^{2/3}}{15000\kappa^3 T_0^{1/3} L}$, $\eta_y = \frac{(1-\lambda)^2 n^{2/3}}{1500\kappa T_0^{1/3} L}$, $b_0 = \frac{400\kappa T_0^{1/3}}{n^{2/3}}$, *where we suppose* $T_0 \geq 10n^2$. *Then our algorithm satisfies*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\Phi(\bar{x}_t)\|^2 \leq \frac{L(\Phi(x_0)-\Phi^*)+\sigma^2+L^2\delta_0}{(nT_0)^{2/3}} + \frac{\frac{1}{n}\sum_{i=1}^n \mathbb{E}\|\nabla_x f_i(x_0,y_0)\|^2}{T_0} + \frac{\frac{1}{n}\sum_{i=1}^n \mathbb{E}\|\nabla_y f_i(x_0,y_0)\|^2}{T_0} \tag{B.80}$$

Proof. When the parameters are defined as Theorem B.2, the conditions in Lemma B.5 and Lemma B.12 are also satisfied. Hence we can prove Eq. (B.68) and (B.72) still hold. When β_x, β_y, η_x and η_y are defined as Theorem B.2, we also have

$$\frac{4L^2}{nT}(47\kappa^2 + \frac{12}{n\beta_x} + \frac{432\kappa^2}{n\beta_y})\frac{576n\lambda^4L^2(\eta_x^2+\eta_y^2)}{(1-\lambda^2)^4}\eta_y^2 \leq \frac{\kappa L\eta_y}{2T} \tag{B.81}$$

and

$$1-2\kappa L\eta_x-\frac{40\kappa^4\eta_x^2}{\eta_y^2}-\frac{24L^2\eta_x^2}{n\beta_x}-\frac{864\kappa^2L^2\eta_x^2}{n\beta_y} \\ -\frac{4L^2}{n}(47\kappa^2+\frac{12}{n\beta_x}+\frac{432\kappa^2}{n\beta_y})\frac{576n\lambda^4L^2(\eta_x^2+\eta_y^2)}{(1-\lambda^2)^4}\eta_x^2\geq\frac{2}{5} \tag{B.82}$$

Similar to Theorem B.1, we can also obtain

$$\begin{aligned}&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\Phi(\bar{x}_t)\|^2\\ &\leq\frac{2(\Phi(x_0)-\Phi^*)}{\eta_xT}+\frac{8\kappa L^2\delta_0}{TL\eta_y}+\frac{4\sigma^2}{nb_0T}(\frac{1}{\beta_x}+\frac{36\kappa^2}{\beta_y})+\frac{8\sigma^2}{n}(\beta_x+36\kappa^2\beta_y)\\ &+\frac{4L^2}{nT}(47\kappa^2+\frac{12}{n\beta_x}+\frac{432\kappa^2}{n\beta_y})\Big(\frac{16\lambda^2\eta_x^2}{(1-\lambda^2)^3}\mathbb{E}\|V_0-\bar{V}_0\|_F^2+\frac{16\lambda^2\eta_y^2}{(1-\lambda^2)^3}\mathbb{E}\|U_0-\bar{U}_0\|_F^2\\ &+\frac{64n\lambda^4(\beta_x\eta_x^2+\beta_y\eta_y^2)\sigma^2}{(1-\lambda^2)^4b_0}+\frac{196n\lambda^4(\beta_x^2\eta_x^2+\beta_y^2\eta_y^2)\sigma^2T}{(1-\lambda^2)^4}\Big)\end{aligned} \tag{B.83}$$

Substitute the parameters with their definitions and we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\Phi(\bar{x}_t)\|^2\leq\frac{L(\Phi(x_0)-\Phi^*)+\sigma^2+L^2\delta_0}{(nT_0)^{2/3}}+\frac{\frac{1}{n}\sum_{i=1}^n\mathbb{E}\|\nabla_xf_i(x_0,y_0)\|^2}{T_0} \\ +\frac{\frac{1}{n}\sum_{i=1}^n\mathbb{E}\|\nabla_yf_i(x_0,y_0)\|^2}{T_0} \tag{B.84}$$

which achieves the conclusion of Theorem B.2. □

Appendix C: Appendix of Chapter 4

In this section, we provide the detailed theoretical analysis of our algorithms, and propose an other Adam-Type algorithm (i.e., Sketched-Adamnc). We first give some useful definitions and lemmas.

Lemma C.1. *Let m be a positive integer. Then we have*

$$\sum_{k=1}^{m} \frac{1}{k} \leq 1 + \log(m). \tag{C.1}$$

Proof. As function $f(x) = \frac{1}{x}$ is decreasing when $x > 0$, we have

$$\sum_{k=1}^{m} \frac{1}{k} = 1 + \sum_{k=2}^{m} \frac{1}{k} \leq 1 + \int_{1}^{m} \frac{dx}{x} = 1 + \log(m). \tag{C.2}$$

□

Lemma C.2. *Let X be a stochastic variable. Then we have*

$$\mathbb{E}[X - \mathbb{E}X]^2 = \mathbb{E}X^2 - (\mathbb{E}X)^2 \geq 0. \tag{C.3}$$

Lemma C.3. *Let $X_1, X_2, \cdots, X_k$ be independent stochastic variables with* 0 *means. Then we have*

$$\mathbb{E}\|\sum_{j=1}^{k} X_j\|^2 = \sum_{j=1}^{k} \mathbb{E}\|X_j\|^2. \tag{C.4}$$

C.1 Convergence Analysis of Sketched-AMSGrad (PA) Algorithm

In the subsection, we provide the detailed convergence analysis of our Sketched-AMSGrad (PA) algorithm.

In the following convergence analysis, we will define a useful auxiliary sequence $\tilde{x}_t$ such that $\tilde{x}_1 = x_1$ and

$$\tilde{x}_{t+1} = \tilde{x}_t - \alpha_t \frac{1}{n} \sum_{i=1}^{n} m_t^{(i)} / \sqrt{\hat{v}_t^{(i)}}. \tag{C.5}$$

Let $e_t = \frac{1}{n} \sum_{i=1}^{n} e_t^{(i)}$. The error compensation term $e_t^{(i)}$ is multiplied by a factor $\frac{\alpha_{t-1}}{\alpha_t}$ because it always satisfies

$$x_t - \tilde{x}_t = \alpha_{t-1} e_{t-1}. \tag{C.6}$$

Theorem C.1. *Assume that Assumption 1 to Assumption 3 are satisfied and data distribution* $\{D_i\}_{i=1}^n$ *are identical. In Algorithm 1, let* $\beta_1 < 1$, $\beta_2 < 1$, $\varepsilon > 0$ *and* $\alpha_t = \frac{\alpha}{\sqrt{1+T}}$, $\alpha > 0$. *Then we have*

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \|\nabla f(x_t)\|^2 \leq \frac{C_1}{\sqrt{T}} + \frac{C_2}{T},$$

where constants C_1 *and* C_2 *are independent of* T.

Proof. Let $A_t^{(i)} = \alpha_t [\hat{v}_t^{(i)}]^{-1/2} \nabla f(\tilde{x}_t)$ for $i = 1, \cdots, n$, $t = 1, \cdots, T$ and $A_0^{(i)} = A_1^{(i)}$. Since $m_t^{(i)} = \beta_1 m_{t-1}^{(i)} + (1-\beta_1) g_t^{(i)}$ and $m_0^{(i)} = \mathbf{0}$, it is easy to check the following equation always holds:

$$\sum_{t=1}^{T} \langle A_t^{(i)}, g_t^{(i)} \rangle = \frac{\beta_1}{1-\beta_1} \langle A_T^{(i)}, m_T^{(i)} \rangle + \sum_{t=1}^{T} \langle A_t^{(i)}, m_t^{(i)} \rangle + \frac{\beta_1}{1-\beta_1} \sum_{t=1}^{T} \langle A_t^{(i)} - A_{t+1}^{(i)}, m_t^{(i)} \rangle \tag{C.7}$$

The left hand side of Eq. (C.7) can be rewritten by

$$\langle A_t^{(i)}, g_t^{(i)} \rangle = \langle \alpha_{t-1} [\hat{v}_{t-1}^{(i)}]^{-1/2} \nabla f(x_t), g_t^{(i)} \rangle - \langle (\alpha_{t-1} [\hat{v}_{t-1}^{(i)}]^{-1/2} - \alpha_t [\hat{v}_t^{(i)}]^{-1/2}) \nabla f(\tilde{x}_t), g_t^{(i)} \rangle$$
$$- \langle \alpha_t [\hat{v}_{t-1}^{(i)}]^{-1/2} (\nabla f(x_t) - \nabla f(\tilde{x}_t)), g_t^{(i)} \rangle \tag{C.8}$$

Let $\xi_t^{(i)}$ be the sample index set at iteration t on node i. As data distribution D_i's are identical, we have

$$\mathbb{E}_{\xi_t^{(i)}} g_t^{(i)} = \nabla f(x_t) \tag{C.9}$$

Therefore, if we take expectation on term $\langle \alpha_{t-1} [\hat{v}_{t-1}^{(i)}]^{-1/2} \nabla f(x_t), g_t^{(i)} \rangle$ over $\xi_t^{(i)}$, we can replace the $g_t^{(i)}$ with $\nabla f(x_t)$. But this operation is not allowed on $\langle A_t^{(i)}, g_t^{(i)} \rangle$ because $\hat{v}_t^{(i)}$ is also dependent on $\xi_t^{(i)}$. We deal with it in this way because the previous value $\hat{v}_{t-1}^{(i)}$ is not determined by $\xi_t^{(i)}$. By using the above Assumption 1 (Lipschitz Gradient) and the definition of $\tilde{x}_t$ in (C.5), we have

$$\frac{1}{n} \sum_{i=1}^{n} \langle A_t^{(i)}, m_t^{(i)} \rangle = \langle \nabla f(\tilde{x}_t), \frac{1}{n} \sum_{i=1}^{n} \alpha_t [\hat{v}_t^{(i)}]^{-1/2} m_t^{(i)} \rangle = \langle \nabla f(\tilde{x}_t), \tilde{x}_{t+1} - \tilde{x}_t \rangle$$
$$\leq f(\tilde{x}_t) - f(\tilde{x}_{t+1}) + \frac{L}{2} \|\tilde{x}_{t+1} - \tilde{x}_t\|^2 \tag{C.10}$$

By Young's inequality we have

$$\frac{1}{n} \sum_{i=1}^{n} \langle A_T^{(i)}, m_T^{(i)} \rangle = \langle \nabla f(\tilde{x}_T), \frac{1}{n} \sum_{i=1}^{n} \alpha_T [\hat{v}_T^{(i)}]^{-1/2} m_T^{(i)} \rangle$$
$$\leq L \| \frac{1}{n} \sum_{i=1}^{n} \alpha_T [\hat{v}_T^{(i)}]^{-1/2} m_T^{(i)} \|^2 + \frac{1}{4L} \|\nabla f(\tilde{x}_T)\|^2 \leq L \|\tilde{x}_{T+1} - \tilde{x}_T\|^2 + \frac{G^2 d}{4L} \tag{C.11}$$

where the last inequality is derived by Assumption 3 (Bounded Gradient). With Assumption

3, we can also obtain

$$\begin{aligned}
&\langle A_t^{(i)} - A_{t+1}^{(i)}, m_t^{(i)} \rangle = \langle \alpha_t [\hat{v}_t^{(i)}]^{-1/2} \nabla f(\tilde{x}_t) - \alpha_{t+1} [\hat{v}_{t+1}^{(i)}]^{-1/2} \nabla f(\tilde{x}_{t+1}), m_t^{(i)} \rangle \\
&= \langle (\alpha_t [\hat{v}_t^{(i)}]^{-1/2} - \alpha_{t+1} [\hat{v}_{t+1}^{(i)}]^{-1/2}) \nabla f(\tilde{x}_{t+1}), m_t^{(i)} \rangle + \langle \nabla f(\tilde{x}_t) - \nabla f(\tilde{x}_{t+1}), \alpha_t [\hat{v}_t^{(i)}]^{-1/2} m_t^{(i)} \rangle \\
&\leq G^2 (\| \alpha_t [\hat{v}_t^{(i)}]^{-1/2} \|_1 - \| \alpha_{t+1} [\hat{v}_{t+1}^{(i)}]^{-1/2} \|_1) + \langle \nabla f(\tilde{x}_t) - \nabla f(\tilde{x}_{t+1}), \alpha_t [\hat{v}_t^{(i)}]^{-1/2} m_t^{(i)} \rangle
\end{aligned} \tag{C.12}$$

where in the last inequality we also use the fact that $\hat{v}_{t+1}^{(i)} \geq \hat{v}_t^{(i)}$ for each component and $\alpha_{t+1} \leq \alpha_t$. Sum i from 1 to n on Eq. (C.12) and we have

$$\begin{aligned}
&\frac{1}{n} \sum_{i=1}^{n} \langle A_t^{(i)} - A_{t+1}^{(i)}, m_t^{(i)} \rangle \\
&\leq \frac{G^2}{n} \sum_{i=1}^{n} (\| \alpha_t [\hat{v}_t^{(i)}]^{-1/2} \|_1 - \| \alpha_{t+1} [\hat{v}_{t+1}^{(i)}]^{-1/2} \|_1) + \langle \nabla f(\tilde{x}_t) - \nabla f(\tilde{x}_{t+1}), \frac{1}{n} \sum_{i=1}^{n} \alpha_t [\hat{v}_t^{(i)}]^{-1/2} m_t^{(i)} \rangle \\
&\leq \frac{G^2}{n} \sum_{i=1}^{n} (\| \alpha_t [\hat{v}_t^{(i)}]^{-1/2} \|_1 - \| \alpha_{t+1} [\hat{v}_{t+1}^{(i)}]^{-1/2} \|_1) + L \| \tilde{x}_{t+1} - \tilde{x}_t \|^2
\end{aligned} \tag{C.13}$$

where the last inequality is derived from Assumption 1. According to Assumption 3, $\alpha_t \leq \alpha_{t-1}$ and the element-wise ascent of $\hat{v}_t^{(i)} \geq \hat{v}_{t-1}^{(i)}$, we have

$$\langle (\alpha_{t-1} [\hat{v}_{t-1}^{(i)}]^{-1/2} - \alpha_t [\hat{v}_t^{(i)}]^{-1/2}) \nabla f(\tilde{x}_t), g_t^{(i)} \rangle \leq G^2 (\| \alpha_{t-1} [\hat{v}_{t-1}^{(i)}]^{-1/2} \|_1 - \| \alpha_t [\hat{v}_t^{(i)}]^{-1/2} \|_1) \tag{C.14}$$

Let $e_t = \frac{1}{n} \sum_{i=1}^{n} e_t^{(i)}$. According to the update rule of $e_t^{(i)}$ we have

$$\alpha_t e_t = \frac{1}{n} \sum_{i=1}^{n} \alpha_t m_t^{(i)} / \sqrt{\hat{v}_t^{(i)}} + \alpha_{t-1} e_{t-1} - \alpha_t \Delta_t \tag{C.15}$$

By the definition of $\tilde{x}_t$ in (C.5), we can estimate term $x_t - \tilde{x}_t$ by

$$\begin{aligned} x_t - \tilde{x}_t &= \sum_{s=1}^{t-1} \alpha_s \Big(\frac{1}{n}\sum_{i=1}^{n} m_s^{(i)} / \sqrt{\hat{v}_s^{(i)}} - \Delta_s\Big) \\ &= \sum_{s=1}^{t-1} \Big[\alpha_s \Big(\frac{1}{n}\sum_{i=1}^{n} (m_s^{(i)} / \sqrt{\hat{v}_s^{(i)}} + \frac{\alpha_{s-1}}{\alpha_s} e_{s-1}^{(i)}) - \Delta_s\Big) - \alpha_{s-1} e_{s-1}\Big] \\ &= \sum_{s=1}^{t-1} (\alpha_s e_s - \alpha_{s-1} e_{s-1}) = \alpha_{t-1} e_{t-1} \end{aligned} \tag{C.16}$$

Define

$$\gamma_0 = (1-\delta)(1-\frac{k}{d}) + \delta,\ \gamma = 1 - \frac{k}{2d}(1-\delta),\ \gamma_1 = \frac{(3-\gamma_0)\gamma_0}{1-\gamma_0} \tag{C.17}$$

We can see $0 < \gamma_0 < 1$. By Lemma 4.1 and Eq. (C.15) we have

$$\begin{aligned} \mathbb{E}\|\alpha_t e_t\|^2 &\le \gamma_0 \mathbb{E}\|\frac{1}{n}\sum_{i=1}^{n} \alpha_t m_t^{(i)} / \sqrt{\hat{v}_t^{(i)}} + \alpha_{t-1} e_{t-1}\|^2 \le \gamma_1 \mathbb{E}\|\frac{1}{n}\sum_{i=1}^{n} \alpha_t m_t^{(i)} / \sqrt{\hat{v}_t^{(i)}}\|^2 + \gamma \mathbb{E}\|\alpha_{t-1} e_{t-1}\|^2 \\ &\le \gamma_1 \sum_{s=1}^{t} \gamma^{t-s} \mathbb{E}\|\frac{1}{n}\sum_{i=1}^{n} \alpha_s [\hat{v}_s^{(i)}]^{-1/2} m_s^{(i)}\|^2 = \gamma_1 \sum_{s=1}^{t} \gamma^{t-s} \mathbb{E}\|\tilde{x}_{s+1} - \tilde{x}_s\|^2 \end{aligned} \tag{C.18}$$

Here the first inequality is because with probability $p > 1-\delta$, it satisfies

$$\|\alpha_t e_t\|^2 \le (1-\frac{k}{d})\ \|\frac{1}{n}\sum_{i=1}^{n} \alpha_t m_t^{(i)} / \sqrt{\hat{v}_t^{(i)}} + \alpha_{t-1} e_{t-1}\|^2$$

and otherwise with probability $p < \delta$, Δ_t is still some coordinates of $\tilde{\Delta}_t$. It always satisfies

$$\|\alpha_t e_t\| \le \|\frac{1}{n}\sum_{i=1}^{n} \alpha_t m_t^{(i)} / \sqrt{\hat{v}_t^{(i)}} + \alpha_{t-1} e_{t-1}\|$$

Hence we can get the first inequality of Eq. (C.18). In the third inequality of Eq. (C.18) we use Young's inequality. In the third inequality of Eq. (C.18) we apply recursion to the

second inequality. Next we can bound the last term in Eq. (C.8).

$$\begin{aligned}
&\mathbb{E}\langle \alpha_t[\hat{v}_{t-1}^{(i)}]^{-1/2}(\nabla f(x_t)-\nabla f(\tilde{x}_t)), g_t^{(i)}\rangle = \mathbb{E}\langle \alpha_t[\hat{v}_{t-1}^{(i)}]^{-1/2}(\nabla f(x_t)-\nabla f(\tilde{x}_t)), \nabla f(x_t)\rangle \\
&\leq \frac{1}{2}\mathbb{E}\langle \alpha_t[\hat{v}_{t-1}^{(i)}]^{-1/2}\nabla f(x_t), \nabla f(x_t)\rangle + \frac{\alpha_t L^2}{2\sqrt{\varepsilon}}\mathbb{E}\|x_t-\tilde{x}_t\|^2 \\
&= \frac{1}{2}\mathbb{E}\langle \alpha_t[\hat{v}_{t-1}^{(i)}]^{-1/2}\nabla f(x_t), g_t^{(i)}\rangle + \frac{\alpha_t L^2}{2\sqrt{\varepsilon}}\mathbb{E}\|x_t-\tilde{x}_t\|^2 \qquad \text{(C.19)}
\end{aligned}$$

where the inequality uses Assumption 1 and Cauchy-Schwartz inequality. The last equality takes expectation on $\xi_t^{(i)}$. The first term of Eq. (C.19) can be merge into the first term of Eq. (C.8). By Eq. (C.16) and (C.18), telescoping from $t=1$ to T we have

$$\begin{aligned}
\sum_{t=1}^{T}\mathbb{E}\|x_t-\tilde{x}_t\|^2 = \sum_{t=1}^{T}\mathbb{E}\|\alpha_{t-1}e_{t-1}\|^2 &\leq \gamma_1\sum_{t=1}^{T}\sum_{s=1}^{t-1}\gamma^{t-1-s}\mathbb{E}\|\tilde{x}_{s+1}-\tilde{x}_s\|^2 \\
&\leq \frac{\gamma_1}{1-\gamma}\sum_{t=1}^{T}\mathbb{E}\|\tilde{x}_{t+1}-\tilde{x}_t\|^2 \qquad \text{(C.20)}
\end{aligned}$$

Combining Eq. (C.7), (C.8), (C.10), (C.11), (C.13), (C.14), (C.19) and (C.20), summing t from 1 to T and averaging i from 1 to n, we have

$$\begin{aligned}
&\frac{1}{2n}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\langle \alpha_{t-1}[\hat{v}_{t-1}^{(i)}]^{-1/2}\nabla f(x_t), g_t^{(i)}\rangle \\
&\leq \frac{L\beta_1}{1-\beta_1}\mathbb{E}\|\tilde{x}_{T+1}-\tilde{x}_T\|^2 + \frac{\beta_1 G^2 d}{4L(1-\beta_1)} + f(\tilde{x}_1) - f(\tilde{x}_{T+1}) + \frac{L}{2}\sum_{t=1}^{T}\mathbb{E}\|\tilde{x}_{t+1}-\tilde{x}_t\|^2 \\
&\quad + \frac{\beta_1 G^2}{n(1-\beta_1)}\sum_{i=1}^{n}(\|\alpha_1[\hat{v}_1^{(i)}]^{-1/2}\|_1 - \mathbb{E}\|\alpha_{T+1}[\hat{v}_{T+1}^{(i)}]^{-1/2}\|_1) + \frac{\beta_1 L}{1-\beta_1}\sum_{t=1}^{T}\mathbb{E}\|\tilde{x}_{t+1}-\tilde{x}_t\|^2 \\
&\quad + \frac{G^2}{n}\sum_{i=1}^{n}(\|\alpha_0[\hat{v}_0^{(i)}]^{-1/2}\|_1 - \mathbb{E}\|\alpha_T[\hat{v}_T^{(i)}]^{-1/2}\|_1) + \frac{\alpha_t L^2\gamma_1}{2\sqrt{\varepsilon}(1-\gamma)}\sum_{t=1}^{T}\mathbb{E}\|\tilde{x}_{t+1}-\tilde{x}_t\|^2 \\
&\leq f(x_1) - f^* + \frac{\beta_1 G^2 d}{4L(1-\beta_1)} + \frac{G^2\alpha d}{\sqrt{T\varepsilon}(1-\beta_1)} + (\frac{(1+\beta_1)L}{1-\beta_1} + \frac{\alpha_t L^2\gamma_1}{2\sqrt{\varepsilon}(1-\gamma)})\sum_{t=1}^{T}\mathbb{E}\|\tilde{x}_{t+1}-\tilde{x}_t\|^2
\end{aligned} \qquad \text{(C.21)}$$

By Cauchy-Schwartz inequality we have

$$\begin{aligned}
&\sum_{t=1}^{T}\|\tilde{x}_{t+1}-\tilde{x}_t\|^2 \\
&\leq \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\|\alpha_t m_t^{(i)}/\sqrt{\hat{v}_t^{(i)}}\|^2 = \frac{(1-\beta_1)^2}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\|\sum_{s=1}^{t}\beta_1^{t-s}\alpha_t g_s^{(i)}/\sqrt{\hat{v}_t^{(i)}}\|^2 \\
&= \frac{(1-\beta_1)^2}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\sum_{s=1}^{t}\sum_{r=1}^{t}\beta_1^{2t-s-r}\langle\alpha_t g_s^{(i)}/\sqrt{\hat{v}_t^{(i)}},\alpha_t g_r^{(i)}/\sqrt{\hat{v}_t^{(i)}}\rangle \\
&\leq \frac{1-\beta_1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\sum_{s=1}^{t}\beta_1^{t-s}\|\alpha_s g_s^{(i)}/\sqrt{\hat{v}_s^{(i)}}\|^2 \leq \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\|\alpha_t g_t^{(i)}/\sqrt{\hat{v}_t^{(i)}}\|^2 \leq \frac{d}{1-\beta_2}\sum_{t=1}^{T}\alpha_t^2
\end{aligned} \tag{C.22}$$

where the second inequality is achieved by $2\langle a,b\rangle \leq \|a\|^2+\|b\|^2$, $\hat{v}_t^{(i)} \geq \hat{v}_s^{(i)}$ and $\alpha_t \leq \alpha_s$ for $t \geq s$. The last inequality is derived from Eq. (C.23) as follows.

$$v_t^{(i)} = (1-\beta_2)\sum_{s=1}^{t}\beta_2^{t-s}[g_s^{(i)}]^2 \tag{C.23}$$

Additionally, taking expectation on the left hand side of Eq. (C.21), by Assumption 3 we have

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\langle\alpha_{t-1}[\hat{v}_{t-1}^{(i)}]^{-1/2}\nabla f(x_t), g_t^{(i)}\rangle \geq \frac{\alpha}{G\sqrt{T}}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(x_t)\|^2 \tag{C.24}$$

Let

$$C_1 = \frac{2G(f(x_1)-f^*)}{\alpha} + \frac{\beta_1 G^3 d}{2L\alpha(1-\beta_1)} + \frac{4GL\alpha d}{(1-\beta_1)(1-\beta_2)}, \tag{C.25}$$

$$C_2 = \frac{GL^2\alpha^2 d\gamma_1}{\sqrt{\varepsilon}(1-\beta_2)(1-\gamma)} + \frac{2G^3 d}{\sqrt{\varepsilon}(1-\beta_1)}. \tag{C.26}$$

Then by Eq. (C.21), (C.22) and (C.24), we can obtain

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(x_t)\|^2 \leq \frac{C_1}{\sqrt{T}} + \frac{C_2}{T}. \tag{C.27}$$

□

C.2 Convergence Analysis of Sketched-AMSGrad (GA) Algorithm

In this subsection, we provide the detailed convergence analysis of our Sketched-AMSGrad (GA) algorithm. Similar to Algorithm 3, we also define $e_t = \frac{1}{n}\sum_{i=1}^{n} e_t^{(i)}$ and define an auxiliary sequence $\tilde{x}_t$ in the convergence analysis, which satisfies $\tilde{x}_1 = x_1$ and

$$\tilde{x}_{t+1} = \tilde{x}_t - \frac{1}{n}\sum_{i=1}^{n} \alpha_t \hat{v}_t^{-1/2} m_t^{(i)}. \tag{C.28}$$

We can prove that the auxiliary sequence $\tilde{x}_t$ satisfies the following Lemma C.4

Lemma C.4. *In Algorithm 4, we always have*

$$x_t - \tilde{x}_t = \alpha_{t-1}\hat{v}_{t-1}^{-1/2} e_{t-1}. \tag{C.29}$$

Proof. By the definition of $\tilde{x}_t$, Δ_t and e_t, we have

$$x_t - \tilde{x}_t = \sum_{s=1}^{t-1} \alpha_s(\hat{v}_s^{-1/2} m_s - \Delta_s) = \alpha_{t-1}\hat{v}_{t-1}^{-1/2} e_{t-1} + \sum_{s=1}^{t-2} \alpha_s(\hat{v}_s^{-1/2} - \hat{v}_{s+1}^{-1/2}) e_s. \tag{C.30}$$

As $\hat{v}_s \geq v_s$ for each element, the coordinate in v_{s+1} which is not updated at iteration $s+1$ keeps the same as v_s and is always smaller that the corresponding coordinate in $\hat{v}_s$. Moreover, since $\hat{v}_{s+1} = \max\{\hat{v}_s, v_{s+1}\}$, we reach the conclusion that for any index $j \notin \mathscr{I}_s$, the value of j-th coordinate in term $(\hat{v}_s^{-1/2} - \hat{v}_{s+1}^{-1/2})$ must be 0. On the other hand, by the definition of Δ_t and e_t, for any index $j \in \mathscr{I}_s$, the j-th coordinate of e_s is always 0. Therefore, term $(\hat{v}_s^{-1/2} - \hat{v}_{s+1}^{-1/2})$ and e_s are orthogonal and we can prove our Lemma C.4. □

Theorem C.2. *Assume that Assumptions 1-3 are satisfied. In Algorithm 2, let $\beta_1 < 1$, $\beta_2 < 1$,*

$\varepsilon > 0$ *and* $\alpha_t = \frac{\alpha}{\sqrt{1+T/n}}$, $\alpha > 0$. *Then we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(x_t)\|^2 \leq \frac{C_1}{\sqrt{nT}} + \frac{C_1 + C_2}{T},$$

where constants C_1 *and* C_2 *are independent of* T.

Proof. We define

$$g_t = \frac{1}{n}\sum_{i=1}^{n} g_t^{(i)},\ m_t = \frac{1}{n}\sum_{i=1}^{n} m_t^{(i)}. \tag{C.31}$$

It automatically satisfies

$$m_t = (1-\beta_1)m_{t-1} + \beta_1 g_t,\ \tilde{x}_{t+1} = \tilde{x}_t - \alpha_t \hat{v}_t^{-1/2} m_t \tag{C.32}$$

Let $A_t = \alpha_t \hat{v}_t^{-1/2}\nabla f(\tilde{x}_t)$ for $t = 1, \cdots, T$ and $A_0 = A_1$. By Eq. (C.32) and $m_0 = \mathbf{0}$, it is easy to check

$$\sum_{t=1}^{T}\langle A_t, g_t\rangle = \frac{\beta_1}{1-\beta_1}\langle A_T, m_T\rangle + \sum_{t=1}^{T}\langle A_t, m_t\rangle + \frac{\beta_1}{1-\beta_1}\sum_{t=1}^{T}\langle A_t - A_{t+1}, m_t\rangle \tag{C.33}$$

The left hand side of Eq. (C.33) can be rewritten by

$$\begin{aligned}\langle A_t, g_t\rangle = {} & \langle \alpha_{t-1}\hat{v}_{t-1}^{-1/2}\nabla f(x_t), g_t\rangle - \langle (\alpha_{t-1}\hat{v}_{t-1}^{-1/2} - \alpha_t \hat{v}_t^{-1/2})\nabla f(\tilde{x}_t), g_t\rangle \\ & - \langle \alpha_t \hat{v}_{t-1}^{-1/2}(\nabla f(x_t) - \nabla f(\tilde{x}_t)), g_t\rangle \end{aligned} \tag{C.34}$$

Similar to Sketched-AMSGrad (GA), we want to obtain $\|\nabla f(x_t)\|^2$ by taking expectation on g_t. However, we cannot do this by taking expectation directly on $\langle A_t, g_t\rangle$ because $\hat{v}_t$ is also determined by $\xi_t^{(i)}$. But the previous value $\hat{v}_{t-1}$ does not depend on $\xi_t^{(i)}$. Therefore we have

$$\mathbb{E}_{\xi_t}\langle \alpha_{t-1}\hat{v}_{t-1}^{-1/2}\nabla f(x_t), g_t\rangle = \langle \alpha_{t-1}\hat{v}_{t-1}^{-1/2}\nabla f(x_t), \nabla f(x_t)\rangle \tag{C.35}$$

By Young's inequality we have

$$\langle A_T, m_T \rangle = \langle \nabla f(\tilde{x}_T), \alpha_T \hat{v}_T^{-1/2} m_T \rangle \leq L \| \alpha_T \hat{v}_T^{-1/2} m_T \|^2 + \frac{1}{4L} \| \nabla f(\tilde{x}_T) \|^2 \tag{C.36}$$

Then using Assumption 3 we get

$$\langle A_T, m_T \rangle \leq L \| \alpha_T \hat{v}_T^{-1/2} m_T \|^2 + \frac{G^2 d}{4L} \tag{C.37}$$

The second term on the right side of Eq. (C.33) can be estimated by

$$\langle A_t, m_t \rangle = \langle \nabla f(\tilde{x}_t), \alpha_t \hat{v}_t^{-1/2} m_t \rangle = \langle \nabla f(\tilde{x}_t), \tilde{x}_t - \tilde{x}_{t+1} \rangle \leq f(\tilde{x}_t) - f(\tilde{x}_{t+1}) + \frac{L}{2} \| \tilde{x}_{t+1} - \tilde{x}_t \|^2 \tag{C.38}$$

where the last inequality is due to Assumption 1. According to Assumptions 1 and 3, Eq. (C.32), $\hat{v}_{t+1} \geq \hat{v}_t$ and $\alpha_{t+1} \leq \alpha_t$, we can obtain

$$\begin{aligned}
&\langle A_t - A_{t+1}, m_t \rangle \\
&= \langle \alpha_t \hat{v}_t^{-1/2} \nabla f(\tilde{x}_t) - \alpha_{t+1} \hat{v}_{t+1}^{-1/2} \nabla f(\tilde{x}_{t+1}), m_t \rangle \\
&= \langle \alpha_t \hat{v}_t^{-1/2} \nabla f(\tilde{x}_{t+1}) - \alpha_{t+1} \hat{v}_{t+1}^{-1/2} \nabla f(\tilde{x}_{t+1}), m_t \rangle + \langle \alpha_t \hat{v}_t^{-1/2} \nabla f(\tilde{x}_t) - \alpha_t \hat{v}_t^{-1/2} \nabla f(\tilde{x}_{t+1}), m_t \rangle \\
&= \langle (\alpha_t \hat{v}_t^{-1/2} - \alpha_{t+1} \hat{v}_{t+1}^{-1/2}) \nabla f(\tilde{x}_{t+1}), m_t \rangle + \langle \nabla f(\tilde{x}_t) - \nabla f(\tilde{x}_{t+1}), \alpha_t \hat{v}_t^{-1/2} m_t \rangle \\
&\leq G^2 (\| \alpha_t \hat{v}_t^{-1/2} \|_1 - \| \alpha_{t+1} \hat{v}_{t+1}^{-1/2} \|_1) + L \| \tilde{x}_{t+1} - \tilde{x}_t \|^2
\end{aligned} \tag{C.39}$$

Similarly, we can bound the second term on the right side of Eq. (C.34)

$$\mathbb{E} \langle (\alpha_{t-1} \hat{v}_{t-1}^{-1/2} - \alpha_t \hat{v}_t^{-1/2}) \nabla f(\tilde{x}_t), g_t \rangle \leq G^2 (\| \alpha_{t-1} \hat{v}_{t-1}^{-1/2} \|_1 - \| \alpha_t \hat{v}_t^{-1/2} \|_1) \tag{C.40}$$

Next, we will estimate term $\langle \alpha_t \hat{v}_t^{-1/2} (\nabla f(x_t) - \nabla f(\tilde{x}_t)), g_t \rangle$. Mimic the reasoning of

Eq. (C.18) with Lemma 4.2, we have

$$\mathbb{E}\|\alpha_t \hat{v}_t^{-1/2} e_t\|^2 \leq \gamma_0 \mathbb{E}\|\alpha_t \hat{v}_t^{-1/2} m_t + \alpha_{t-1} \hat{v}_t^{-1/2} e_{t-1}\|^2 \leq \gamma_1 \sum_{s=1}^{t} \gamma'^{-s} \mathbb{E}\|\alpha_s \hat{v}_s^{-1/2} m_s\|^2 \quad \text{(C.41)}$$

Taking expectation, we can estimate the last term of Eq. (C.34) by

$$\begin{aligned} &\mathbb{E}\langle \alpha_t \hat{v}_{t-1}^{-1/2} (\nabla f(x_t) - \nabla f(\tilde{x}_t)), g_t \rangle = \mathbb{E}\langle \alpha_t \hat{v}_{t-1}^{-1/2} (\nabla f(x_t) - \nabla f(\tilde{x}_t)), \nabla f(x_t) \rangle \\ &\leq \frac{1}{2} \mathbb{E}\langle \alpha_t [\hat{v}_{t-1}^{(i)}]^{-1/2} \nabla f(x_t), \nabla f(x_t) \rangle + \frac{\alpha_t L^2}{2\sqrt{\varepsilon}} \mathbb{E}\|x_t - \tilde{x}_t\|^2 \end{aligned} \quad \text{(C.42)}$$

The inequality results from Cauchy-Schwartz inequality and Assumption 1. Sum the last term of Eq. (C.42) from $t = 1$ to T and we have

$$\begin{aligned} &\sum_{t=1}^{T} \mathbb{E}\|x_t - \tilde{x}_t\|^2 = \sum_{t=1}^{T} \mathbb{E}\|\alpha_{t-1} \hat{v}_{t-1}^{-1/2} e_{t-1}\|^2 \\ &\leq \gamma_1 \sum_{t=1}^{T} \sum_{s=1}^{t-1} \gamma'^{t-1-s} \mathbb{E}\|\alpha_s \hat{v}_s^{-1/2} m_s\|^2 \leq \frac{\gamma_1}{(1-\gamma)} \sum_{t=1}^{T} \mathbb{E}\|\alpha_t \hat{v}_t^{-1/2} m_t\|^2 \end{aligned} \quad \text{(C.43)}$$

Combine Eqs. (C.33), (C.34), (C.37), (C.38), (C.39), (C.40), (C.42) and (C.43). Take expectation and we have

$$\begin{aligned} &\frac{1}{2} \sum_{t=1}^{T} \mathbb{E}\langle \alpha_{t-1} \hat{v}_{t-1}^{-1/2} \nabla f(x_t), \nabla f(x_t) \rangle \\ &\leq f(\tilde{x}_1) - f(\tilde{x}_{T+1}) + \frac{\beta_1 G^2 d}{4L(1-\beta_1)} + (\frac{L}{2} + \frac{2\beta_1 L}{1-\beta_1}) \sum_{t=1}^{T} \mathbb{E}\|\tilde{x}_t - \tilde{x}_{t+1}\|^2 \\ &+ \frac{\beta_1 G^2}{1-\beta_1} (\|\alpha_1 \hat{v}_1^{-1/2}\|_1 - \mathbb{E}\|\alpha_{T+1} \hat{v}_{T+1}^{-1/2}\|_1) + G^2 (\|\alpha_0 \hat{v}_0^{-1/2}\|_1 - \mathbb{E}\|\alpha_T \hat{v}_T^{-1/2}\|_1) \\ &+ \frac{\alpha_t L^2 \gamma_1}{2\sqrt{\varepsilon}(1-\gamma)} \sum_{t=1}^{T} \mathbb{E}\|\alpha_t \hat{v}_t^{-1/2} m_t\|^2 + \frac{L}{2} \sum_{t=1}^{T} \mathbb{E}\|\alpha_t \hat{v}_t^{-1/2} g_t\|^2. \end{aligned} \quad \text{(C.44)}$$

As $\hat{v}_{t+1} \geq \hat{v}_t$ and $\alpha_{t+1} \leq \alpha_t$, we have

$$\begin{aligned}
&\sum_{t=1}^{T} \|\alpha_t \hat{v}_t^{-1/2} m_t\|^2 \\
&= (1-\beta_1)^2 \sum_{t=1}^{T} \|\sum_{s=1}^{t} \beta_1^{t-s} \alpha_t \hat{v}_t^{-1/2} g_s\|^2 = (1-\beta_1)^2 \sum_{t=1}^{T} \sum_{s,j=1}^{t} \beta_1^{2t-s-j} \langle \alpha_t \hat{v}_t^{-1/2} g_s, \alpha_t \hat{v}_t^{-1/2} g_j \rangle \\
&\leq (1-\beta_1) \sum_{t=1}^{T} \sum_{s=1}^{t} \beta_1^{t-s} \|\alpha_s \hat{v}_s^{-1/2} g_s\|^2 \leq \sum_{t=1}^{T} \|\alpha_t \hat{v}_t^{-1/2} g_t\|^2. \qquad \text{(C.45)}
\end{aligned}$$

From Lemma C.2 and Lemma C.3 we know

$$\mathbb{E}\|g_t - \nabla f(x_t)\|^2 = \mathbb{E}\|\frac{1}{n}\sum_{i=1}^{n}(g_t^{(i)} - \nabla f_i(x_t))\|^2 \leq \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\|g_t^{(i)} - \nabla f_i(x_t)\|^2 \leq \frac{G^2 d}{n}. \quad \text{(C.46)}$$

According to Eq. (C.32), (C.44), (C.45), (C.46) and Assumption 2 we can obtain

$$\begin{aligned}
&\frac{1}{2}\sum_{t=1}^{T} \mathbb{E}\langle \alpha_{t-1} \hat{v}_{t-1}^{-1/2} \nabla f(x_t), \nabla f(x_t) \rangle \\
&\leq f(x_1) - f^* + \frac{\beta_1 G^2 d}{4L(1-\beta_1)} + \frac{G^2 \alpha_t}{\sqrt{\varepsilon}(1-\beta_1)} + C_0 \sum_{t=1}^{T} \mathbb{E}\|\alpha_t \hat{v}_t^{-1/2} g_t\|^2 \\
&\leq f(x_1) - f^* + \frac{\beta_1 G^2 d}{4L(1-\beta_1)} + \frac{G^2 \alpha_t}{\sqrt{\varepsilon}(1-\beta_1)} + \frac{2C_0}{\varepsilon}\sum_{t=1}^{T} \alpha_t^2 (\mathbb{E}\|\nabla f(x_t)\|^2 + \mathbb{E}\|g_t - \nabla f(x_t)\|^2) \\
&\leq f(x_1) - f^* + \frac{\beta_1 G^2 d}{4L(1-\beta_1)} + \frac{G^2 \alpha_t}{\sqrt{\varepsilon}(1-\beta_1)} + \frac{2C_0 G^2 d}{n\varepsilon}\sum_{t=1}^{T} \alpha_t^2 + \frac{2C_0}{\varepsilon}\sum_{t=1}^{T} \alpha_t^2 \mathbb{E}\|\nabla f(x_t)\|^2.
\end{aligned} \qquad \text{(C.47)}$$

where

$$C_0 = \frac{(1+\beta_1)L}{1-\beta_1} + \frac{\alpha_t L^2 \gamma_1}{2\sqrt{\varepsilon}(1-\gamma)} \qquad \text{(C.48)}$$

The left side of Eq. (C.47) can be lower bounded by

$$\sum_{t=1}^{T} \mathbb{E}\langle \alpha_{t-1} \hat{v}_{t-1}^{-1/2} \nabla f(x_t), g_t \rangle \geq \sum_{t=1}^{T} \frac{\alpha_t}{G} \mathbb{E}\|\nabla f(x_t)\|^2. \qquad \text{(C.49)}$$

Since $\alpha_t \leq \frac{\varepsilon}{8C_0G}$ when T is large, we have

$$\frac{\alpha}{4G\sqrt{1+\frac{T}{n}}}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(x_t)\|^2 \leq f(x_1)-f^* + \frac{\beta_1 G^2 d}{4L(1-\beta_1)} + \frac{G^2\alpha_t}{\sqrt{\varepsilon}(1-\beta_1)} + \frac{2C_0G^2d}{n\varepsilon}\sum_{t=1}^{T}\alpha_t^2 \tag{C.50}$$

Let

$$C_1 = \frac{4G(f(x_1)-f^*)}{\alpha} + \frac{\beta_1 G^3 d}{L\alpha(1-\beta_1)} + \frac{8(1+\beta_1)LG^3 d\alpha}{\varepsilon(1-\beta_1)}, \tag{C.51}$$

$$C_2 = \frac{4L^2G^3 d\alpha^2\gamma_1}{\varepsilon^{3/2}(1-\gamma)} + \frac{4G^3}{\sqrt{\varepsilon}(1-\beta_1)}. \tag{C.52}$$

Then we can reach the conclusion

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(x_t)\|^2 \leq \sqrt{1+\frac{T}{n}}\left(\frac{C_1}{T}+\frac{C_2}{T\sqrt{1+T/n}}\right) \leq \frac{C_1}{\sqrt{nT}} + \frac{C_1+C_2}{T}. \tag{C.53}$$

In the last inequality we use the fact that $\sqrt{1+x} \leq 1+\sqrt{x}$. □

Next we will provide the proof for Remark 4.2.

Proof. When we set $\alpha = \alpha_t = \sqrt{\frac{n}{T}}$, Eq. (C.47) and (C.49) still hold since we only use the relation $\alpha_t \geq \alpha_{t+1}$. When it satisfies $\sqrt{\frac{n}{T}} \leq \frac{\varepsilon}{4C_0G}$ we have

$$\frac{\sqrt{n}}{2G\sqrt{T}}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(x_t)\|^2 \leq f(x_1)-f^* + \frac{\beta_1 G^2 d}{4L(1-\beta_1)} + \frac{G^2\sqrt{n}}{\sqrt{\varepsilon}(1-\beta_1)\sqrt{T}} + \frac{2C_0G^2d}{\varepsilon} \tag{C.54}$$

Therefore, we can obtain

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(x_t)\|^2 \leq \frac{2G(f(x_1)-f^*)}{\sqrt{nT}} + \frac{\beta_1 G^3 d}{2L(1-\beta_1)\sqrt{nT}} + \frac{4C_0G^3d}{\varepsilon\sqrt{nT}} + \frac{4G^3}{\sqrt{\varepsilon}(1-\beta_1)T} \tag{C.55}$$

Algorithm 9 Sketched-Adamnc Algorithm

Input: initial value x_1, sketching operator $\mathscr{S}$ and unsketching operator $\mathscr{U}$
Set: $m_0^{(i)} = \mathbf{0}$, $e_0^{(i)} = \mathbf{0}$ on i-th worker node
for $t = 1$ **to** T **do**
  On i-th worker node:
    Estimate a stochastic gradient $g_t^{(i)}$, and then compute $m_t^{(i)} = \beta_1 m_{t-1}^{(i)} + (1-\beta_1) g_t^{(i)}$;
    $v_t^{(i)} = \frac{1}{t}\sum_{j=1}^{t} [g_j^{(i)}]^2 + \varepsilon \mathbf{1}$;
    Sketch $S_t^{(i)} = \mathscr{S}(m_t^{(i)} / \sqrt{v_t^{(i)}} + \frac{\alpha_{t-1}}{\alpha_t} e_{t-1}^{(i)})$, and then send $S_t^{(i)}$ to the master node;
    Send $\Delta_t^{(i)}$ to the master node after unsketching;
    $e_t^{(i)} = m_t^{(i)} / \sqrt{v_t^{(i)}} + \frac{\alpha_{t-1}}{\alpha_t} e_{t-1}^{(i)} - \Delta_t^{(i)}$;
    Receive Δ_t from the master node, and then update $x_{t+1} = x_t - \alpha_t \Delta_t$;
  On the master node:
    Aggregate $S_t = \frac{1}{n}\sum_{i=1}^{n} S_t^{(i)}$;
    Unsketch $\Delta_t = \frac{1}{n}\sum_{i=1}^{n} \Delta_t^{(i)} = \text{Top-}k(\mathscr{U}(S_t))$;
    Send Δ_t back to each worker node, and then update $x_{t+1} = x_t - \alpha_t \Delta_t$.
end for

which indicates that the convergence rate is $O(\frac{1}{\sqrt{nT}})$. □

C.3 Sketched-Adamnc Algorithm

In the section, we propose another Adam-type algorithm with sketching technique. Since our Sketched-AMSGrad (GA) algorithm has better theoretical properties such as linear speedup and convergence with non-identical data distribution. But the parameter averaging scheme is more convenient to implement and more compatible with other Adam-type algorithms. For example, besides AMSGrad, we can also apply sketching method to Adamnc [92] algorithm using the parameter averaging scheme. The description of Sketched-Adamnc algorithm is shown in Algorithm 9.

Similar to Lemma 4.1, we have the following lemma

Lemma C.5. *In Algorithm 4 we define* $\tilde{\Delta}_t = \frac{1}{n}\sum_{i=1}^{n}(m_t^{(i)} / \sqrt{v_t^{(i)}} + \frac{\alpha_{t-1}}{\alpha_t} e_{t-1}^{(i)})$. *With sketch*

size $\Theta(klog(d/\delta))$ *and with probability* $\geq 1-\delta$*, we have*

$$\|\tilde{\Delta}_t - \Delta_t\|^2 \leq (1-\frac{k}{d})\|\tilde{\Delta}_t\|^2. \tag{C.56}$$

Theorem C.3. *Assume that the Assumptions 1-3 are satisfied and data distribution* D_i*'s are identical. Let* $\beta_1 < 1$*,* $\beta_2 < 1$*,* $\varepsilon > 0$ *and* $\alpha_t = \frac{\alpha}{\sqrt{t}}$*,* $\alpha > 0$*. Then there exist constant* C_1 *and* C_2 *such that Algorithm 4 satisfies:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(x_t)\|^2 \leq \frac{C_1}{\sqrt{T}} + \frac{C_2(1+\log(T))}{\sqrt{T}} \tag{C.57}$$

Proof. Define auxiliary sequence $\tilde{x}_t$ such that $\tilde{x}_1 = x_1$ and

$$\tilde{x}_{t+1} = \tilde{x}_t - \frac{1}{n}\sum_{i=1}^{n}\alpha_t m_t^{(i)}/\sqrt{v_t^{(i)}} \tag{C.58}$$

Let $A_t^{(i)} = \alpha_t[v_t^{(i)}]^{-1/2}\nabla f(\tilde{x}_t)$ for $i = 1,\cdots,n$, $t = 1,\cdots,T$ and $A_0^{(i)} = A_1^{(i)}$. Since $\alpha_t = \frac{\alpha}{\sqrt{t}}$ and

$$v_t^{(i)} = \frac{1}{t}\sum_{j=1}^{t}[g_j^{(i)}]^2 + \varepsilon\mathbf{1} \tag{C.59}$$

we always have

$$\alpha_{t+1}[v_{t+1}^{(i)}]^{-1/2} \leq \alpha_t[v_t^{(i)}]^{-1/2} \tag{C.60}$$

Similar to the proof of Theorem 4.1, we also have

$$\sum_{t=1}^{T}\langle A_t^{(i)}, g_t^{(i)}\rangle = \frac{\beta_1}{1-\beta_1}\langle A_T^{(i)}, m_T^{(i)}\rangle + \sum_{t=1}^{T}\langle A_t^{(i)}, m_t^{(i)}\rangle + \frac{\beta_1}{1-\beta_1}\sum_{t=1}^{T}\langle A_t^{(i)} - A_{t+1}^{(i)}, m_t^{(i)}\rangle \tag{C.61}$$

$$\begin{aligned}\langle A_t^{(i)}, g_t^{(i)}\rangle = \langle \alpha_{t-1}[v_{t-1}^{(i)}]^{-1/2}\nabla f(x_t), g_t^{(i)}\rangle - \langle(\alpha_{t-1}[v_{t-1}^{(i)}]^{-1/2} - \alpha_t[v_t^{(i)}]^{-1/2})\nabla f(x_t), g_t^{(i)}\rangle \\ -\langle \alpha_t[v_t^{(i)}]^{-1/2}(\nabla f(x_t) - \nabla f(\tilde{x}_t)), g_t^{(i)}\rangle\end{aligned} \tag{C.62}$$

$$\frac{1}{n}\sum_{i=1}^{n}\langle A_T^{(i)}, m_T^{(i)}\rangle \leq L\|\tilde{x}_{T+1}-\tilde{x}_T\|^2+\frac{G^2 d}{4L} \tag{C.63}$$

$$\frac{1}{n}\sum_{i=1}^{n}\langle A_t^{(i)}, m_t^{(i)}\rangle \leq f(\tilde{x}_t)-f(\tilde{x}_{t+1})+\frac{L}{2}\|\tilde{x}_{t+1}-\tilde{x}_t\|^2 \tag{C.64}$$

$$\frac{1}{n}\sum_{i=1}^{n}\langle A_t^{(i)}-A_{t+1}^{(i)}, m_t^{(i)}\rangle \leq \frac{G^2}{n}\sum_{i=1}^{n}(\|\alpha_t[v_t^{(i)}]^{-1/2}\|_1-\|\alpha_{t+1}[v_{t+1}^{(i)}]^{-1/2}\|_1)+L\|\tilde{x}_{t+1}-\tilde{x}_t\|^2 \tag{C.65}$$

$$\langle(\alpha_{t-1}[\hat{v}_{t-1}^{(i)}]^{-1/2}-\alpha_t[\hat{v}_t^{(i)}]^{-1/2})\nabla f(x_t), g_t^{(i)}\rangle \leq G^2(\|\alpha_{t-1}[\hat{v}_{t-1}^{(i)}]^{-1/2}\|_1-\|\alpha_t[\hat{v}_t^{(i)}]^{-1/2}\|_1) \tag{C.66}$$

In above two inequalities we use the relation of Eq. (C.60). Let $e_t=\frac{1}{n}\sum_{i=1}^{n}e_t^{(i)}$. We also have

$$\alpha_t e_t=\frac{1}{n}\sum_{i=1}^{n}\alpha_t m_t^{(i)}/\sqrt{v_t^{(i)}}+\alpha_{t-1}e_{t-1}-\alpha_t\Delta_t \tag{C.67}$$

and

$$x_t-\tilde{x}_t=\alpha_{t-1}e_{t-1} \tag{C.68}$$

Define

$$\theta=1-(1-\delta)(1-\frac{k}{d})-\delta, \quad \gamma=1-\theta^2 \tag{C.69}$$

and mimic the reasoning of Eq. (C.41)

$$\begin{aligned}
\mathbb{E}\|\alpha_t e_t\|^2 &\leq \Big((1-\delta)(1-\frac{k}{d})+\delta\Big)\mathbb{E}\|\frac{1}{n}\sum_{i=1}^{n}\alpha_t m_t^{(i)}/\sqrt{v_t^{(i)}}+\alpha_{t-1}e_{t-1}\|^2 \\
&\leq \gamma\mathbb{E}\|\alpha_{t-1}e_{t-1}\|^2+(1-\theta)(1+\frac{1}{\theta})\mathbb{E}\|\frac{1}{n}\sum_{i=1}^{n}\alpha_t m_t^{(i)}/\sqrt{v_t^{(i)}}\|^2 \\
&\leq \frac{\gamma}{\theta}\sum_{s=1}^{t}\gamma^{t-s}\mathbb{E}\|\frac{1}{n}\sum_{i=1}^{n}\alpha_s[v_s^{(i)}]^{-1/2}m_s^{(i)}\|^2 \qquad \text{(C.70)}
\end{aligned}$$

Then we can estimate the last term of Eq. (C.62)

$$\begin{aligned}
&\langle\alpha_t[v_t^{(i)}]^{-1/2}(\nabla f(x_t)-\nabla f(\tilde{x}_t)),g_t^{(i)}\rangle \leq \|\nabla f(x_t)-\nabla f(\tilde{x}_t)\|\cdot\|\alpha_t[v_t^{(i)}]^{-1/2}g_t^{(i)}\| \\
\leq\ & \frac{1}{2L}\|\nabla f(x_t)-\nabla f(\tilde{x}_t)\|^2+\frac{L}{2}\|\alpha_t[v_t^{(i)}]^{-1/2}g_t^{(i)}\|^2 \leq \frac{L}{2}\|x_t-\tilde{x}_t\|^2+\frac{L}{2}\|\alpha_t[v_t^{(i)}]^{-1/2}g_t^{(i)}\|^2
\end{aligned} \qquad \text{(C.71)}$$

The second inequality results from Young's inequality and the last inequality results from Assumption 1. Sum Eq. (C.71) from $t=1$ to T, average i and take expectation

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\langle\alpha_t[v_t^{(i)}]^{-1/2}(\nabla f(x_t)-\nabla f(\tilde{x}_t)),g_t^{(i)}\rangle \\
\leq\ & \frac{L}{2}\sum_{t=1}^{T}\mathbb{E}\|x_t-\tilde{x}_t\|^2+\frac{L}{2n}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\|\alpha_t[v_t^{(i)}]^{-1/2}g_t^{(i)}\|^2 \\
=\ & \frac{L}{2}\sum_{t=1}^{T}\mathbb{E}\|\alpha_{t-1}e_{t-1}\|^2+\frac{L}{2n}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\|\alpha_t[v_t^{(i)}]^{-1/2}g_t^{(i)}\|^2 \\
\leq\ & \frac{L\gamma}{2\theta}\sum_{t=1}^{T}\sum_{s=1}^{t-1}\gamma^{t-1-s}\mathbb{E}\|\frac{1}{n}\sum_{i=1}^{n}\alpha_s[v_s^{(i)}]^{-1/2}m_s^{(i)}\|^2+\frac{L}{2n}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\|\alpha_t[v_t^{(i)}]^{-1/2}g_t^{(i)}\|^2 \\
\leq\ & \frac{L\gamma}{2\theta(1-\gamma)}\sum_{t=1}^{T}\mathbb{E}\|\frac{1}{n}\sum_{i=1}^{n}\alpha_t[v_t^{(i)}]^{-1/2}m_t^{(i)}\|^2+\frac{L}{2n}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\|\alpha_t[v_t^{(i)}]^{-1/2}g_t^{(i)}\|^2 \\
\leq\ & \frac{L\gamma}{2n\theta(1-\gamma)}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\|\alpha_t[v_t^{(i)}]^{-1/2}m_t^{(i)}\|^2+\frac{L}{2n}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\|\alpha_t[v_t^{(i)}]^{-1/2}g_t^{(i)}\|^2 \qquad \text{(C.72)}
\end{aligned}$$

Combine Eq. (C.61), (C.62), (C.63), (C.64), (C.65), (C.66) and (C.72) and we get

$$
\begin{aligned}
& \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\langle \alpha_{t-1}[v_{t-1}^{(i)}]^{-1/2}\nabla f(x_t), g_t^{(i)}\rangle \\
\leq\ & \frac{L\beta_1}{1-\beta_1}\mathbb{E}\|\tilde{x}_{T+1}-\tilde{x}_T\|^2+\frac{\beta_1 G^2 d}{4L(1-\beta_1)}+f(\tilde{x}_1)-f(\tilde{x}_{T+1})+\frac{L}{2}\sum_{t=1}^{T}\mathbb{E}\|\tilde{x}_{t+1}-\tilde{x}_t\|^2 \\
& +\frac{\beta_1 G^2}{n(1-\beta_1)}\sum_{i=1}^{n}(\|\alpha_1[v_1^{(i)}]^{-1/2}\|_1-\mathbb{E}\|\alpha_{T+1}[v_{T+1}^{(i)}]^{-1/2}\|_1)+\frac{\beta_1 L}{1-\beta_1}\sum_{t=1}^{T}\mathbb{E}\|\tilde{x}_{t+1}-\tilde{x}_t\|^2 \\
& +\frac{G^2}{n}\sum_{i=1}^{n}(\|\alpha_0[v_0^{(i)}]^{-1/2}\|_1-\mathbb{E}\|\alpha_T[v_T^{(i)}]^{-1/2}\|_1)+\frac{L}{2n}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\|\alpha_t[v_t^{(i)}]^{-1/2}g_t^{(i)}\|^2 \\
& +\frac{L\gamma}{2n\theta(1-\gamma)}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\|\alpha_t[v_t^{(i)}]^{-1/2}m_t^{(i)}\|^2 \\
\leq\ & f(x_1)-f^*+\frac{\beta_1 G^2 d}{4L(1-\beta_1)}+\frac{G^2\alpha d}{\sqrt{\varepsilon}(1-\beta_1)}+(\frac{L}{2}+\frac{2\beta_1 L}{1-\beta_1})\sum_{t=1}^{T}\mathbb{E}\|\tilde{x}_{t+1}-\tilde{x}_t\|^2 \\
& +\frac{L\gamma}{2n\theta(1-\gamma)}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\|\alpha_t[v_t^{(i)}]^{-1/2}m_t^{(i)}\|^2+\frac{L}{2n}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\|\alpha_t[v_t^{(i)}]^{-1/2}g_t^{(i)}\|^2
\end{aligned}
\tag{C.73}
$$

By Eq. (C.60) and Cauchy-Schwartz inequality we have

$$
\begin{aligned}
& \sum_{t=1}^{T}\|\tilde{x}_{t+1}-\tilde{x}_t\|^2 \\
\leq\ & \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\|\alpha_t[v_t^{(i)}]^{-1/2}m_t^{(i)}\|^2=\frac{(1-\beta_1)^2}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\|\sum_{s=1}^{t}\beta_1^{t-s}\alpha_t[v_t^{(i)}]^{-1/2}g_s^{(i)}\|^2 \\
=\ & \frac{(1-\beta_1)^2}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\sum_{s=1}^{t}\sum_{r=1}^{t}\beta_1^{2t-s-r}\langle \alpha_t[v_t^{(i)}]^{-1/2}g_s^{(i)}, \alpha_t[v_t^{(i)}]^{-1/2}g_r^{(i)}\rangle \\
\leq\ & \frac{1-\beta_1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\sum_{s=1}^{t}\beta_1^{t-s}\|\alpha_s[v_s^{(i)}]^{-1/2}g_s^{(i)}\|^2\leq\frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\|\alpha_t[v_t^{(i)}]^{-1/2}g_t^{(i)}\|^2
\end{aligned}
\tag{C.74}
$$

Define

$$
C_0=\frac{(1+\beta_1)L}{1-\beta_1}+\frac{L\gamma}{2\theta(1-\gamma)} \tag{C.75}
$$

Then we have

$$\begin{aligned} & \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\langle \alpha_{t-1}[v_{t-1}^{(i)}]^{-1/2}\nabla f(x_t), g_t^{(i)}\rangle \\ \leq \;& f(x_1)-f^* + \frac{\beta_1 G^2 d}{4L(1-\beta_1)} + \frac{G^2\alpha d}{\sqrt{\varepsilon}(1-\beta_1)} + \frac{C_0}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\|\alpha_t[v_t^{(i)}]^{-1/2}g_t^{(i)}\|^2 \end{aligned} \tag{C.76}$$

Let $g_{t,j}^{(i)}$ be the j-th coordinate of vector $g_t^{(i)}$. Then we have

$$\sum_{t=1}^{T}\|\alpha_t[v_t^{(i)}]^{-1/2}g_t^{(i)}\|^2 = \sum_{t=1}^{T}\sum_{j=1}^{d}\alpha^2\frac{[g_{t,j}^{(i)}]^2}{\sum_{s=1}^{t}[g_{s,j}^{(i)}]^2+t\varepsilon} \leq \sum_{j=1}^{d}\alpha^2\sum_{t=1}^{T}\frac{[g_{t,j}^{(i)}]^2}{\sum_{s=1}^{t}[g_{s,j}^{(i)}]^2} \tag{C.77}$$

Let $a_t = \sum_{s=1}^{t}[g_{s,j}^{(i)}]^2$, $t = 1,\cdots,T$. Since function $f(x) = \frac{1}{x}$ is decreasing, we have

$$\frac{[g_{t,j}^{(i)}]^2}{a_t} \leq \int_{a_{t-1}}^{a_t}\frac{dx}{x} \tag{C.78}$$

when $t \geq 2$. Therefore, we have bound

$$\sum_{t=1}^{T}\frac{[g_{t,j}^{(i)}]^2}{\sum_{s=1}^{t}[g_{s,j}^{(i)}]^2} \leq 1+\int_{a_1}^{a_T}\frac{dx}{x} \leq 1+\log(a_T) \leq 1+2\log(G)+\log(T) \tag{C.79}$$

where the last inequality is resulted from Assumption 3. Taking expectation on $g_t^{(i)}$ on the left of Eq. (C.76), by Assumption 3 we have

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\mathbb{E}\langle \alpha_{t-1}[v_{t-1}^{(i)}]^{-1/2}\nabla f(x_t), g_t^{(i)}\rangle \geq \frac{\alpha}{G\sqrt{T}}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(x_t)\|^2 \tag{C.80}$$

Let

$$C_1 = \frac{G(f(x_1)-f^*)}{\alpha} + \frac{\beta_1 G^3 d}{4L\alpha(1-\beta_1)} + \frac{G^3 d}{\sqrt{\varepsilon}(1-\beta_1)} + 2C_0 G d \alpha \log(G) \quad \text{(C.81)}$$

$$C_2 = C_0 G d \alpha \quad \text{(C.82)}$$

Combining Eq. (C.76), (C.77), (C.79) and (C.80) we can reach the conclusion

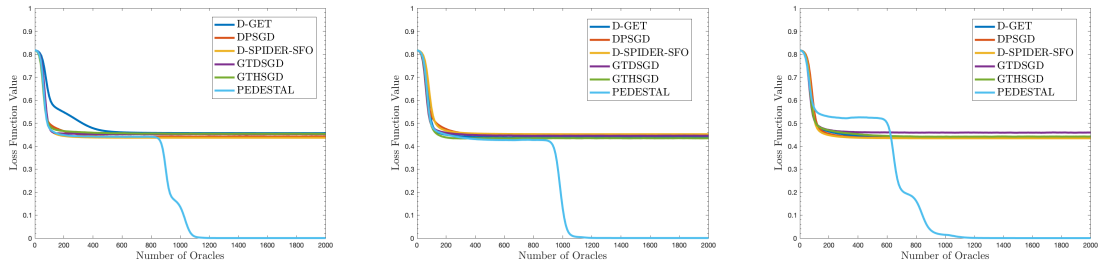
$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\nabla f(x_t)\|^2 \leq \frac{C_1}{\sqrt{T}} + \frac{C_2(1+\log(T))}{\sqrt{T}} \quad \text{(C.83)}$$

□

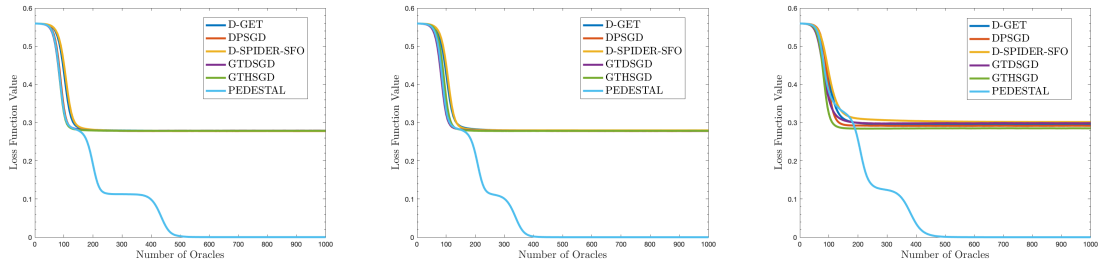
Appendix D: Appendix of Chapter 5

D.1 Additional Experimental Results

The experimental results of Dirichlet distribution of the matrix sensing task is shown in Figure D.1. The smallest eigenvalue at the converged point for each algorithm is shown in Table D.1 and Table D.2.



(a) $d = 50$, ring (b) $d = 50$, toroidal (c) $d = 50$, exponential



(d) $d = 100$, ring (e) $d = 100$, toroidal (f) $d = 100$, exponential

Figure D.1: Experimental results of the decentralized matrix sensing task on different network topology for $d = 50$ and $d = 100$. Data is assigned to worker nodes by Dirichlet distribution. The y-axis is the loss function value and the x-axis is the number of gradient oracles divided by the number of data N.

	D-PSGD	GTDSGD	D-GET	D-SPIDER-SFO	GTHSGD	PEDESTAL
$d = 50$, ring	-0.0332	-0.0327	-0.0333	-0.0328	-0.0329	$\mathbf{-1.78e^{-5}}$
$d = 50$, toroidal	-0.0331	-0.0334	-0.0334	-0.0327	-0.0329	$\mathbf{-4.18e^{-5}}$
$d = 50$, exponential	-0.0323	-0.0330	-0.0331	-0.0332	-0.0333	$\mathbf{-1.09e^{-6}}$
$d = 100$, ring	-0.0184	-0.0184	-0.0184	-0.0184	-0.0185	$\mathbf{-2.07e^{-6}}$
$d = 100$, toroidal	-0.0185	-0.0186	-0.0185	-0.0184	-0.0184	$\mathbf{-2.25e^{-7}}$
$d = 100$, exponential	-0.0184	-0.0184	-0.0186	-0.0184	-0.0184	$\mathbf{-3.07e^{-5}}$

Table D.1: Smallest eigenvalue of hessian matrix at the converged point (random data distribution).

	D-PSGD	GTDSGD	D-GET	D-SPIDER-SFO	GTHSGD	PEDESTAL
$d = 50$, ring	-0.0332	-0.0337	-0.0332	-0.0325	-0.0330	$\mathbf{-3.60e^{-6}}$
$d = 50$, toroidal	-0.0334	-0.0324	-0.0329	-0.0325	-0.0327	$\mathbf{-2.29e^{-5}}$
$d = 50$, exponential	-0.0334	-0.0326	-0.0333	-0.0330	-0.0328	$\mathbf{-3.97e^{-5}}$
$d = 100$, ring	-0.0184	-0.0184	-0.0184	-0.0185	-0.0183	$\mathbf{-4.48e^{-5}}$
$d = 100$, toroidal	-0.0184	-0.0184	-0.0184	-0.0184	-0.0185	$\mathbf{-1.24e^{-5}}$
$d = 100$, exponential	-0.0186	-0.0185	-0.0186	-0.0183	-0.0185	$\mathbf{-3.63e^{-6}}$

Table D.2: Smallest eigenvalue of hessian matrix at the converged point (Dirichlet data distribution).

D.2 Proof of Theorem 5.1

D.2.1 Notation

We define matrix $X_t = [x_t^{(1)}, \cdots, x_t^{(n)}] \in \mathbb{R}^{d\times n}$ where $x_t^{(i)}$ is the model parameter on i-th worker node with dimension d and n is the number of worker nodes. Similarly we have $Y_t = [y_t^{(1)}, \cdots, y_t^{(n)}]$, $Z_t = [z_t^{(1)}, \cdots, z_t^{(n)}]$ and $V_t = [v_t^{(1)}, \cdots, v_t^{(n)}]$. Let $\omega_t = \|\bar{x}_{t+1} - \bar{x}_t\|^2$ and $\Omega_t = Z_t - X_t$. Define $p_t = n_t/n$ where n_t is the number of worker nodes drawing perturbation in iteration t.

D.2.2 Outline

In this section we will provide the proof outline of Theorem 5.1. First, we prove some basic lemmas to estimate gradient noise and consensus error, which will be used frequently in later proof. The gradient noise is estimated by Lemma D.1, the proof of which can be

found in Section D.3.1. The consensus error is estimated by Lemma D.2, the proof of which can be found in Section D.3.2.

Lemma D.1. *(Gradient Noise) Under Assumption 5.2 and Assumption 5.3 we have*

$$(a)\quad \frac{1}{nT}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\|v_t^{(i)}-\nabla f_i(x_t^{(i)})\|^2 \le \frac{16\log(4/\delta)\beta\sigma^2}{b_1}+\frac{384\log(4/\delta)L^2}{nb_1\beta T}\sum_{t=0}^{T-1}\|X_t-\bar{X}_t\|_F^2 + \frac{192\log(4/\delta)L^2}{b_1\beta T}\sum_{t=0}^{T-1}\omega_t+\frac{2\log(4/\delta)\sigma^2}{\beta b_0 T}$$

$$(b)\quad \frac{1}{T}\sum_{t=1}^{T}\|\bar{v}_t-\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_t^{(i)})\|^2 \le \frac{16\log(4/\delta)\beta\sigma^2}{nb_1}+\frac{384\log(4/\delta)L^2}{n^2b_1\beta T}\sum_{t=1}^{T}\|X_t-\bar{X}_t\|_F^2 + \frac{192\log(4/\delta)L^2}{nb_1\beta T}\sum_{t=0}^{T-1}\omega_t+\frac{2\log(4/\delta)\sigma^2}{n\beta b_0 T}$$

Lemma D.2. *(Consensus Error) Let* $\eta \le \frac{(1-\lambda)^2\varepsilon^\theta}{600\log(4/\delta)\lambda^2 L}$, $\beta = C_1^{-1}\varepsilon^{1+\theta}$ *and* $b_1 \ge C_1\varepsilon^{-1+\theta}$ *where* $C_1 \ge 1$ *is a constant. Under Assumption 5.2, 5.3 and 5.5 we have*

$$(a)\quad \frac{1}{T}\sum_{t=1}^{T}\|X_t-\bar{X}_t\|_F^2 \le \frac{160000n\log(4/\delta)L^2\eta^2\lambda^4}{(1-\lambda)^4\min\{b_1\beta,1\}T}\sum_{t=0}^{T-1}\omega_t+\frac{12288n\log(4/\delta)\beta\eta^2\lambda^4\sigma^2}{(1-\lambda)^4 b_1} + \frac{2000n\log(4/\delta)\eta^2\lambda^4\sigma^2}{(1-\lambda)^4\beta b_0 T}+\frac{128\lambda^4\eta^2}{(1-\lambda)^3 T}\sum_{i=1}^{n}\|\nabla f_i(x_0)\|^2+\sum_{t=0}^{T-1}\frac{64n\lambda^2 p_t(\eta^2C_v^2+r^2)}{(1-\lambda)^2 T}$$

$$(b)\quad \frac{1}{T}\sum_{t=0}^{T-1}\|Y_t-\bar{Y}_t\|_F^2 \le \frac{4644\log(4/\delta)nL^2\lambda^2}{(1-\lambda)\min\{b_1\beta,1\}T}\sum_{t=0}^{T-1}\omega_t+\frac{384\log(4/\delta)n\lambda^2\beta\sigma^2}{(1-\lambda)b_1} + \frac{50\log(4/\delta)n\lambda^2\sigma^2}{(1-\lambda)\beta b_0 T}+\frac{8\lambda^2}{(1-\lambda)T}\sum_{i=1}^{n}\|\nabla f_i(x_0)\|^2+\sum_{t=0}^{T-1}\frac{150000\log(4/\delta)nL^2\lambda^4 p_t(\eta^2C_v^2+r^2)}{(1-\lambda)^3\min\{b_1\beta,1\}T}$$

Next we will prove that PEDESTAL will terminate in certain number of iterations. Under Assumption 5.2, 5.3 and 5.5, we can prove the following Lemma D.3. The proof is demonstrated in Section D.3.3.

Lemma D.3. *(Descent) Let* $\eta \le \frac{(1-\lambda)^2\varepsilon^\theta}{600\log(4/\delta)\lambda^2 L}$, $\beta = C_1^{-1}\varepsilon^{1+\theta}$, $b_1 \ge C_1\varepsilon^{-1+\theta}$ *and* $b_0 =$

$C_1\varepsilon^{-1}$ *where* $C_1 \geq 1$ *is a constant. Under Assumption 5.2, 5.3 and 5.5 we have*

$$f(\bar{x}_T) \leq f(x_0) + \frac{\sigma^2}{L} + \frac{1}{nL}\sum_{i=1}^{n}\|\nabla f_i(x_0)\|^2 - \sum_{t=0}^{T-1}\mathscr{D}_t$$

where

$$\mathscr{D}_t = \frac{1}{16\eta}\omega_t + \frac{(1-\lambda)^2}{256n\eta}\sum_{i=1}^{n}\|x_{t+1}^{(i)} - x_t^{(i)}\|^2 + \frac{\eta}{2n}\sum_{i=1}^{n}\|y_t^{(i)}\|^2 - \frac{200\eta\varepsilon^2\sigma^2}{(1-\lambda)^2C_1^2} - \frac{7p_t(\eta^2C_v^2 + r^2)}{4\eta}$$

Here we call $\mathscr{D}_t$ the descent of iteration t. We categorize all iterations into three types:

$$\text{type-A: } p_t \geq \frac{1}{5}, \quad \text{type-B: } p_t < \frac{1}{5} \text{ and } \frac{1}{n}\sum_{i=1}^{n}\|y_t^{(i)}\|^2 \geq \frac{4C_v^2}{5}, \quad \text{type-C: otherwise}$$

When at least $\frac{n}{5}$ nodes drawing perturbation in iteration t, then it is type-A. There are two cases where p_t is small: most nodes in the descent phase or most nodes in the escaping phase. An iteration is type-B if $p_t < \frac{1}{5}$ and $\frac{1}{n}\sum_{i=1}^{n}\|y_t^{(i)}\|^2 \geq \frac{4C_v^2}{5}$, which represents the case where most nodes are in the descent phase. And type-C iteration represents the case where most nodes are in the escaping phase. Next we will estimate type-A and type-C iteration with the following Lemma D.4.

Lemma D.4. *Let* $\eta \leq \frac{(1-\lambda)^2\varepsilon^\theta}{600\log(4/\delta)\lambda^2 L}$, $\beta = C_1^{-1}\varepsilon^{1+\theta}$, $b_1 \geq C_1\varepsilon^{-1+\theta}$, $b_0 = C_1\varepsilon^{-1}$, $C_d = C_2\eta C_T\varepsilon$, $C_v = \frac{(1-\lambda)C_2\varepsilon}{200}$ *and* $r \leq \eta C_v/4$ *where* $C_1 = \frac{20000\sigma}{(1-\lambda)^2C_2}$ *and* C_2 *is a constant. Under Assumption 5.2, 5.3 and 5.5, we can find disjoint intervals* $\mathscr{I} = \mathscr{I}_1 \cup \cdots \cup \mathscr{I}_k$ *such that the indexes of all type-A and type-C iterations except the last* C_T *iterations are contained in* $\mathscr{I}$ *and the descent over* $\mathscr{I}$ *can be estimated by*

$$\sum_{t\in\mathscr{I}}\mathscr{D}_t \geq |\mathscr{I}|\cdot\frac{(1-\lambda)^2C_2^2\eta\varepsilon^2}{10000}$$

where $|\cdot|$ *denotes the total number of the set.*

Besides, for all type-B iteration t, we have the following estimation

Lemma D.5. *Let parameter and assumption settings be the same as Lemma D.4, then for all type-B iteration t we have*

$$\mathscr{D}_t \geq \frac{(1-\lambda)^2 C_2^2 \eta \varepsilon^2}{8000000}$$

With Lemma D.4, Lemma D.5 and Assumption 5.1, we can conclude that PEDESTAL will terminate in $\tilde{O}(\varepsilon^{-2-\theta})+C_T$ iterations. As the last two negative terms in $\mathscr{D}_t$ are canceled by $\frac{1}{n}\sum_{i=1}^n \|x_{t+1}^{(i)} - x_t^{(i)}\|^2$ and $\frac{1}{n}\sum_{i=1}^n \|y_t^{(i)}\|^2$ respectively in Lemma D.4 and Lemma D.5, we have $\frac{1}{\eta}\sum_{t=0}^{T-1} \omega_t \leq O(1)$. Hence by Lemma D.2 we know the consensus error $\frac{1}{n}\|X_t - \bar{X}_t\|_F^2$ can be bounded by $O(\varepsilon^{1+\theta})$ on average. Besides, from the parameter setting we can see C_v is $\Theta(\varepsilon)$, which ensures the first-order optimality of the decentralized algorithm.

Finally, we will prove PEDESTAL is able to achieve second-order stationary point. First, we will give the small stuck region lemma in decentralized setting. Recall that $\varepsilon_H = \varepsilon^\alpha$ is the tolerance of second-order stationary point. The proof is in Section D.3.6.

Lemma D.6. *(Small Stuck Region) Suppose n_s worker nodes draw perturbation in iteration s and $-\gamma = \min eig(\nabla^2 f(\bar{x}_s)) \leq -\varepsilon_H$. Let $\eta \leq \frac{(1-\lambda)^2\varepsilon^\theta}{1000\sqrt{n}\log(C_T)\log(4/\delta)\lambda^2 L}$, $\beta = C_1^{-1}\varepsilon^{1+\theta}$, $b_1 \geq 1000C_1\varepsilon^{2-\theta-5\alpha}$, $C_d = C_2\eta C_T \varepsilon^\mu$ and $C_T = \log(12nC_d/r_0)/(\eta\gamma)$ where $C_1 = \frac{20000}{(1-\lambda)^2 C_2}$, $C_2 \leq \frac{1-\lambda}{2000\log(4/\delta)\rho\log(C_d)}$ and $\mu = \max\{1, 2\alpha\}$. Let X_t and X_t' be two coupled decentralized sequences by running PEDESTAL from X_s with $X_s = X_s'$, $x_{s+1}^{(i)} = x_{s+1}^{(i)'}$ if node i does not draw perturbation in iteration s and $x_{s+1}^{(i)} = x_{s+1}^{(i)'} + r_0\mathbf{e_1}$ otherwise. Here $\mathbf{e_1}$ is the eigenvector with respect to the smallest eigenvalue γ. Define $d_i = \max_{s\leq t\leq s+C_T}\{\|x_t^{(i)} - x_s^{(i)}\|, \|x_t^{(i)'} - x_s^{(i)}\|\}$. Then there are at least $\frac{9n}{10}$ nodes such that $d_i \geq 2C_d$.*

In decentralized small stuck region lemma, the consensus error will lead to a new term (see Eq. (D.30)) and make the proof more complicated. In our proof, we use the condition

of $\varepsilon_H \geq \varepsilon$, *i.e.,* $\alpha \leq 1$. For smaller ε_H the batchsize b_1 is required to set larger. With Lemma D.6, we can prove that when PEDESTAL is terminated, it finds a local minimum with high probability.

Lemma D.7. *Let $r_0 = \delta r/\sqrt{d}$ where d is the dimension of model parameter. Other parameters are the same as Lemma D.6. Suppose PEDESTAL is terminated in iteration $s + C_T$. Then $\bar{x}_s$ is a second-order stationary point with probability at least $1 - \delta$.*

Lemma D.7 provides the guarantee of second-order optimality of PEDESTAL. When $\varepsilon_H \geq \sqrt{\varepsilon}$, *i.e.,* $\alpha \leq 0.5$ (including the classic setting $\varepsilon_H = \sqrt{\varepsilon}$), the parameter settings of all lemmas are consistent and the main theorem is proven. The total gradient complexity is

$$\tilde{O}(\varepsilon^{-2-\theta} \cdot \varepsilon^{-1+\theta}) = \tilde{O}(\varepsilon^{-3})$$

When $\alpha = 0.5$, we have $\theta = 0.25$ and $b_1 = \Theta(\varepsilon^{-0.75})$. When $\alpha \leq 0.2$, we can set $\theta = 1$ and $b_1 = O(1)$, which is result of PEDESTAL-S. In Section D.4 we will provide the analysis of the case $\alpha > 0.5$ with a different parameter setting of θ and b_1. We can achieve the gradient complexity of

$$\tilde{O}(\varepsilon^{-3} + \varepsilon\varepsilon_H^{-8} + \varepsilon^4\varepsilon_H^{-11}) \tag{D.1}$$

over all cases of ε_H.

D.3 Proof of Lemmas

D.3.1 Proof of Lemma D.1

Proof. According to the definition of $v_t^{(i)}$, we have

$$\frac{v_{t+1}^{(i)} - \nabla f_i(x_{t+1}^{(i)})}{(1-\beta)^{t+1}} - \frac{v_t^{(i)} - \nabla f_i(x_t^{(i)})}{(1-\beta)^t} = \frac{\beta(\nabla F_i(x_{t+1}^{(i)}, \xi_{t+1}^{(i)}) - \nabla f_i(x_{t+1}^{(i)}))}{(1-\beta)^{t+1}} + \frac{(\nabla F_i(x_{t+1}^{(i)}, \xi_{t+1}^{(i)}) - \nabla f_i(x_{t+1}^{(i)})) - (\nabla F_i(x_t^{(i)}, \xi_{t+1}^{(i)}) - \nabla f_i(x_t^{(i)}))}{(1-\beta)^t} \quad \text{(D.2)}$$

where $|\xi_{t+1}^{(i)}| = b_1$. The expectation of the right side of Eq. (D.2) over $\xi_{t+1}^{(i)}$ is 0. Using Cauchy-Schwartz inequality, Assumption 5.2 and Assumption 5.3 we have

$$\begin{aligned} &\|\frac{\beta(\nabla F_i(x_{t+1}^{(i)}, j) - \nabla f_i(x_{t+1}^{(i)}))}{(1-\beta)^{t+1}} + \frac{(\nabla F_i(x_{t+1}^{(i)}, j) - \nabla f_i(x_{t+1}^{(i)})) - (\nabla F_i(x_t^{(i)}, j) - \nabla f_i(x_t^{(i)}))}{(1-\beta)^t}\|^2 \\ &\leq \frac{2\beta^2\sigma^2}{(1-\beta)^{2t+2}} + \frac{8L^2\|x_{t+1}^{(i)} - x_t^{(i)}\|^2}{(1-\beta)^{2t}} \end{aligned} \quad \text{(D.3)}$$

for each $j \in \xi_{t+1}^{(i)}$. Thus, applying Azuma-Hoeffding inequality to Eq. (D.2) we can obtain

$$\begin{aligned} &\|v_t^{(i)} - \nabla f_i(x_t^{(i)}) - (1-\beta)^t(v_0^{(i)} - \nabla f_i(x_0))\|^2 \\ &\leq \frac{4\log(4/\delta)}{b_1}(2\beta\sigma^2 + 8L^2\sum_{s=0}^{t-1}(1-\beta)^{2(t-s)}\|x_{s+1}^{(i)} - x_s^{(i)}\|^2) \end{aligned} \quad \text{(D.4)}$$

with probability $1-\delta$. Here we use the fact that $\sum_{s=0}^{+\infty}(1-\beta)^s = \frac{1}{\beta}$. Using Cauchy-Schwartz inequality to Eq. (D.4) we have

$$\begin{aligned} \|v_t^{(i)} - \nabla f_i(x_t^{(i)})\|^2 \leq \frac{16\log(4/\delta)}{b_1}(\beta\sigma^2 + 4L^2\sum_{s=0}^{t-1}(1-\beta)^{2(t-s)}\|x_{s+1}^{(i)} - x_s^{(i)}\|^2) \\ + 2(1-\beta)^{2t}\|v_0^{(i)} - \nabla f_i(x_0))\|^2 \end{aligned} \quad \text{(D.5)}$$

Sum Eq. (D.5), we obtain

$$\begin{aligned}
&\frac{1}{\log(4/\delta)nT}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\|v_t^{(i)}-\nabla f_i(x_t^{(i)})\|^2 \\
&\leq \frac{16\beta\sigma^2}{b_1}+\frac{64L^2}{nb_1\beta T}\sum_{t=0}^{T-2}\|X_{t+1}-X_t\|_F^2+\frac{2\sigma^2}{\beta b_0 T} \\
&\leq \frac{16\beta\sigma^2}{b_1}+\frac{384L^2}{nb_1\beta T}\sum_{t=0}^{T-1}\|X_t-\bar{X}_t\|_F^2+\frac{192L^2}{b_1\beta T}\sum_{t=0}^{T-1}\omega_t+\frac{2\sigma^2}{\beta b_0 T} \qquad \text{(D.6)}
\end{aligned}$$

which finishes the proof of (a) in Lemma D.1. In the first inequality of Eq. (D.6) we apply Azuma-Hoeffding inequality to $v_0^{(i)}-\nabla f_i(x_0)$. In the second inequality we apply Cauchy-Schwartz inequality and use the fact $x_{t+1}^{(i)}-x_t^{(i)}=(x_{t+1}^{(i)}-\bar{x}_{t+1})-(x_t^{(i)}-\bar{x}_t)+(\bar{x}_{t+1}-\bar{x}_t)$. Mimic above steps and we can achieve the inequality (b) in Lemma D.1. The term n in the denominator is derived by the fact that $\xi_t^{(i)}$'s on different nodes are independent. □

D.3.2 Proof of Lemma D.2

Proof. As $Y_t=W(Y_{t-1}+V_t-V_{t-1})$, we have

$$\begin{aligned}
&\|Y_t-\bar{Y}_t\|_F^2 \\
&=\|(W-J)(Y_{t-1}-\bar{Y}_{t-1})+(W-J)(V_t-V_{t-1})\|_F^2 \\
&\leq \lambda^2\|Y_{t-1}-\bar{Y}_{t-1}\|_F^2+2\langle(W-J)Y_t,(W-J)(V_t-V_{t-1})\rangle+\lambda^2\|V_t-V_{t-1}\|_F^2 \\
&\leq \frac{1+\lambda^2}{2}\|Y_{t-1}-\bar{Y}_{t-1}\|_F^2+\frac{\lambda^2+\lambda^4}{1-\lambda^2}\|V_t-V_{t-1}\|_F^2 \\
&\leq \frac{1+\lambda^2}{2}\|Y_{t-1}-\bar{Y}_{t-1}\|_F^2+\frac{3\lambda^2(1+\lambda^2)}{1-\lambda^2}\sum_{i=1}^{n}(\|v_t^{(i)}-\nabla f_i(x_t^{(i)})\|^2+\|v_{t-1}^{(i)}-\nabla f_i(x_{t-1}^{(i)})\|^2) \\
&\quad+\frac{9L^2\lambda^2(1+\lambda^2)}{1-\lambda^2}(\|X_t-\bar{X}_t\|_F^2+\|X_{t-1}-\bar{X}_{t-1}\|_F^2+n\omega_{t-1}) \qquad \text{(D.7)}
\end{aligned}$$

where the first inequality is derived by Assumption 5.5, the second inequality is derived by Young's inequality and the last inequality is derived by Cauchy-Schwartz inequality and

Assumption 5.3. When $t=0$, by Azuma-Hoeffding inequality we can get

$$\|Y_0-\bar{Y}_0\|_F^2 \leq 2\lambda^2\sum_{i=1}^{n}\|\nabla f_i(x_0)\|^2+\frac{8\log(4/\delta)n\lambda^2\sigma^2}{b_0} \tag{D.8}$$

with probability $1-\delta$. As $X_{t+1}=W(X_t+\Omega_t)$, by Assumption 5.5 and Young's inequality we have

$$\begin{aligned}
&\|X_{t+1}-\bar{X}_{t+1}\|_F^2\\
&\leq \frac{1+\lambda^2}{2}\|X_t-\bar{X}_t\|_F^2+\frac{2\lambda^2}{1-\lambda^2}\|\Omega_t-\bar{\Omega}_t\|_F^2\\
&\leq \frac{1+\lambda^2}{2}\|X_t-\bar{X}_t\|_F^2+\frac{4\eta^2\lambda^2}{1-\lambda^2}\|Y_t-\bar{Y}_t\|_F^2+\frac{4\lambda^2}{1-\lambda^2}\|\Omega_t-\bar{\Omega}_t-\eta(Y_t-\bar{Y}_t)\|_F^2\\
&\leq \frac{1+\lambda^2}{2}\|X_t-\bar{X}_t\|_F^2+\frac{4\eta^2\lambda^2}{1-\lambda^2}\|Y_t-\bar{Y}_t\|_F^2+\frac{8n\lambda^2p_t(\eta^2C_v^2+r^2)}{1-\lambda^2}
\end{aligned} \tag{D.9}$$

where the second inequality is obtained by Cauchy-Schwartz inequality and the last inequality is because when node i draws perturbation it must satisfy $\|y_t^{(i)}\|\leq C_v$. Note that $X_0=\bar{X}_0$. Sum Eq. (D.9), we have

$$\begin{aligned}
&\sum_{t=1}^{T}\|X_t-\bar{X}_t\|_F^2\\
&\leq \frac{8\eta^2\lambda^2}{(1-\lambda^2)^2}\sum_{t=0}^{T-1}\|Y_t-\bar{Y}_t\|_F^2+\frac{16n\lambda^2(\eta^2C_v^2+r^2)T}{(1-\lambda^2)^2}\\
&\leq \frac{288L^2\eta^2\lambda^4(1+\lambda^2)}{(1-\lambda^2)^4}\sum_{t=0}^{T-1}\|X_t-\bar{X}_t\|_F^2+\frac{96\eta^2\lambda^4(1+\lambda^2)}{(1-\lambda^2)^4}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\|v_t^{(i)}-\nabla f_i(x_t^{(i)})\|^2\\
&\quad+\frac{144nL^2\eta^2\lambda^4(1+\lambda^2)}{(1-\lambda^2)^4}\sum_{t=0}^{T-1}\omega_t+\frac{16\lambda^2\eta^2}{(1-\lambda^2)^3}\|Y_0-\bar{Y}_0\|_F^2+\sum_{t=0}^{T-1}\frac{16n\lambda^2p_t(\eta^2C_v^2+r^2)}{(1-\lambda^2)^2}
\end{aligned} \tag{D.10}$$

where the last inequality comes from Eq. (D.7). When $\eta \leq \frac{(1-\lambda)^2}{40\lambda^2 L}$ we have $\frac{288L^2\eta^2\lambda^4(1+\lambda^2)}{(1-\lambda^2)^4} \leq \frac{1}{2}$ and

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\|X_t-\bar{X}_t\|_F^2 \\
&\leq \frac{192\eta^2\lambda^4(1+\lambda^2)}{(1-\lambda^2)^4 T}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\|v_t^{(i)}-\nabla f_i(x_t^{(i)})\|^2+\frac{288nL^2\eta^2\lambda^4(1+\lambda^2)}{(1-\lambda^2)^4 T}\sum_{t=0}^{T-1}\omega_t \\
&\quad+\frac{64\lambda^4\eta^2}{(1-\lambda^2)^3 T}\sum_{i=1}^{n}\|\nabla f_i(x_0)\|^2+\frac{256\log(4/\delta)n\lambda^4\eta^2\sigma^2}{(1-\lambda^2)^3 b_0 T}+\sum_{t=0}^{T-1}\frac{32n\lambda^2 p_t(\eta^2 C_v^2+r^2)}{(1-\lambda^2)^2 T} \\
&\leq \frac{73728\log(4/\delta)L^2\eta^2\lambda^4(1+\lambda^2)}{(1-\lambda^2)^4 b_1\beta T}\sum_{t=0}^{T-1}\|X_t-\bar{X}_t\|_F^2+\frac{24576n\log(4/\delta)L^2\eta^2\lambda^4(1+\lambda^2)}{(1-\lambda^2)^4 b_1\beta T}\sum_{t=0}^{T-1}\omega_t \\
&\quad+\frac{n\log(4/\delta)\eta^2\lambda^4(1+\lambda^2)\sigma^2}{(1-\lambda^2)^4}\left(\frac{3072\beta}{b_1}+\frac{384}{\beta b_0 T}\right)+\frac{288nL^2\eta^2\lambda^4(1+\lambda^2)}{(1-\lambda^2)^4 T}\sum_{t=0}^{T-1}\omega_t \\
&\quad+\frac{64\lambda^4\eta^2}{(1-\lambda^2)^3 T}\sum_{i=1}^{n}\|\nabla f_i(x_0)\|^2+\frac{256\log(4/\delta)n\lambda^4\eta^2\sigma^2}{(1-\lambda^2)^3 b_0 T}+\sum_{t=0}^{T-1}\frac{32n\lambda^2 p_t(\eta^2 C_v^2+r^2)}{(1-\lambda^2)^2 T}
\end{aligned}
\tag{D.11}
$$

where the last inequality is achieved by Lemma D.1. According to the parameter setting, we have

$$
\frac{73728\log(4/\delta)L^2\eta^2\lambda^4(1+\lambda^2)}{(1-\lambda^2)^4 b_1\beta} \leq \frac{1}{2}
$$

Therefore, we have

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\|X_t-\bar{X}_t\|_F^2 \\
&\leq \frac{160000n\log(4/\delta)L^2\eta^2\lambda^4}{(1-\lambda)^4\min\{b_1\beta,1\}T}\sum_{t=0}^{T-1}\omega_t+\frac{12288n\log(4/\delta)\beta\eta^2\lambda^4\sigma^2}{(1-\lambda)^4 b_1}+\frac{2000n\log(4/\delta)\eta^2\lambda^4\sigma^2}{(1-\lambda)^4\beta b_0 T} \\
&\quad+\frac{128\lambda^4\eta^2}{(1-\lambda)^3 T}\sum_{i=1}^{n}\|\nabla f_i(x_0)\|^2+\sum_{t=0}^{T-1}\frac{64n\lambda^2 p_t(\eta^2 C_v^2+r^2)}{(1-\lambda)^2 T}
\end{aligned}
\tag{D.12}
$$

where we have used the condition $\lambda \leq 1$ to simplify the inequality. Moreover, sum Eq. (D.7) and we can achieve

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\|Y_t-\bar{Y}_t\|_F^2 \\
&\leq \frac{12\lambda^2}{(1-\lambda)T}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\|v_t^{(i)}-\nabla f_i(x_t^{(i)})\|^2+\frac{36L^2\lambda^2}{(1-\lambda)T}\sum_{t=0}^{T-1}\|X_t-\bar{X}_t\|_F^2+\frac{18nL^2\lambda^2}{(1-\lambda)T}\sum_{t=0}^{T-1}\omega_t \\
&\quad+\frac{2}{(1-\lambda)T}\|Y_0-\bar{Y}_0\|_F^2 \\
&\leq \frac{36L^2\lambda^2}{(1-\lambda)T}(1+\frac{128\log(4/\delta)}{b_1\beta})\sum_{t=0}^{T-1}\|X_t-\bar{X}_t\|_F^2+\frac{18nL^2\lambda^2}{(1-\lambda)T}(1+\frac{128\log(4/\delta)}{b_1\beta})\sum_{t=0}^{T-1}\omega_t \\
&\quad+\frac{192\log(4/\delta)n\lambda^2\beta\sigma^2}{(1-\lambda)b_1}+\frac{25\log(4/\delta)n\lambda^2\sigma^2}{(1-\lambda)\beta b_0 T}+\frac{4\lambda^2}{(1-\lambda)T}\sum_{i=1}^{n}\|\nabla f_i(x_0)\|^2 \\
&\leq \frac{4644\log(4/\delta)L^2\lambda^2}{(1-\lambda)\min\{b_1\beta,1\}T}\sum_{t=0}^{T-1}\|X_t-\bar{X}_t\|_F^2+\frac{2322\log(4/\delta)nL^2\lambda^2}{(1-\lambda)\min\{b_1\beta,1\}T}\sum_{t=0}^{T-1}\omega_t \\
&\quad+\frac{192\log(4/\delta)n\lambda^2\beta\sigma^2}{(1-\lambda)b_1}+\frac{25\log(4/\delta)n\lambda^2\sigma^2}{(1-\lambda)\beta b_0 T}+\frac{4\lambda^2}{(1-\lambda)T}\sum_{i=1}^{n}\|\nabla f_i(x_0)\|^2 \\
&\leq \frac{37152\log(4/\delta)L^2\eta^2\lambda^4}{(1-\lambda)^3\min\{b_1\beta,1\}T}\sum_{t=0}^{T-1}\|Y_t-\bar{Y}_t\|_F^2+\frac{2322\log(4/\delta)nL^2\lambda^2}{(1-\lambda)\min\{b_1\beta,1\}T}\sum_{t=0}^{T-1}\omega_t \\
&\quad+\frac{192\log(4/\delta)n\lambda^2\beta\sigma^2}{(1-\lambda)b_1}+\sum_{t=0}^{T-1}\frac{74304\log(4/\delta)nL^2\lambda^4 p_t(\eta^2C_v^2+r^2)}{(1-\lambda)^3\min\{b_1\beta,1\}T}+\frac{25\log(4/\delta)n\lambda^2\sigma^2}{(1-\lambda)\beta b_0 T} \\
&\quad+\frac{4\lambda^2}{(1-\lambda)T}\sum_{i=1}^{n}\|\nabla f_i(x_0)\|^2 \qquad \text{(D.13)}
\end{aligned}
$$

where the second inequality uses Lemma D.1 and Eq. (D.8). The last inequality uses the sum of Eq. (D.9). As $\frac{37152\log(4/\delta)L^2\eta^2\lambda^4}{(1-\lambda)^3\min\{b_1\beta,1\}} \leq \frac{1}{2}$, we have

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\|Y_t-\bar{Y}_t\|_F^2 \\
&\leq \frac{4644\log(4/\delta)nL^2\lambda^2}{(1-\lambda)\min\{b_1\beta,1\}T}\sum_{t=0}^{T-1}\omega_t + \frac{384\log(4/\delta)n\lambda^2\beta\sigma^2}{(1-\lambda)b_1} + \frac{50\log(4/\delta)n\lambda^2\sigma^2}{(1-\lambda)\beta b_0 T} \\
&\quad + \frac{8\lambda^2}{(1-\lambda)T}\sum_{i=1}^{n}\|\nabla f_i(x_0)\|^2 + \sum_{t=0}^{T-1}\frac{150000\log(4/\delta)nL^2\lambda^4 p_t(\eta^2C_v^2+r^2)}{(1-\lambda)^3\min\{b_1\beta,1\}T}
\end{aligned} \tag{D.14}$$

which finishes the proof. □

D.3.3 Proof of Lemma D.3

Proof. By Assumption 5.3 we have

$$\begin{aligned}
f(\bar{x}_{t+1}) &\leq f(\bar{x}_t) + \langle\nabla f(\bar{x}_t), \bar{x}_{t+1}-\bar{x}_t\rangle + \frac{L}{2}\|\bar{x}_{t+1}-\bar{x}_t\|^2 \\
&= f(\bar{x}_t) + \langle\nabla f(\bar{x}_t), -\eta\bar{v}_t\rangle + \langle\nabla f(\bar{x}_t), \bar{x}_{t+1}-\bar{x}_t+\eta\bar{v}_t\rangle + \frac{L}{2}\|\bar{x}_{t+1}-\bar{x}_t\|^2 \\
&= f(\bar{x}_t) - \frac{\eta}{2}\|\bar{v}_t\|^2 - \frac{\eta}{2}\|\nabla f(\bar{x}_t)\|^2 + \frac{\eta}{2}\|\bar{v}_t-\nabla f(\bar{x}_t)\|^2 + \frac{\eta}{2}\|\nabla f(\bar{x}_t)\|^2 \\
&\quad + \frac{1}{2\eta}\|\bar{x}_{t+1}-\bar{x}_t+\eta\bar{v}_t\|^2 - \frac{1}{2\eta}\|\bar{x}_{t+1}-\bar{x}_t+\eta\bar{v}_t-\eta\nabla f(\bar{x}_t)\|^2 + \frac{L}{2}\|\bar{x}_{t+1}-\bar{x}_t\|^2 \\
&\leq f(\bar{x}_t) - \frac{\eta}{2}\|\bar{v}_t\|^2 + \frac{\eta}{2}\|\bar{v}_t-\nabla f(\bar{x}_t)\|^2 + \frac{1}{2\eta}\|\bar{x}_{t+1}-\bar{x}_t+\eta\bar{v}_t\|^2 + \frac{L\omega_t}{2} \\
&\quad - \frac{1}{2\eta}\omega_t - \frac{\eta}{2}\|\bar{v}_t-\nabla f(\bar{x}_t)\|^2 + \frac{1}{4\eta}\omega_t + \eta\|\bar{v}_t-\nabla f(\bar{x}_t)\|^2 \\
&\leq f(\bar{x}_t) - \frac{1}{4\eta}\omega_t - \frac{\eta}{2}\|\bar{v}_t\|^2 + \frac{p_t(\eta^2C_v^2+r^2)}{\eta} + \frac{L\omega_t}{2} + 2\eta\|\bar{v}_t - \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_t^{(i)})\|^2 \\
&\quad + \frac{L^2\eta}{n}\|X_t-\bar{X}_t\|_F^2
\end{aligned} \tag{D.15}$$

where the first inequality is obtained by Young's inequality and the last inequality is obtained by Cauchy-Schwartz inequality, Assumption 5.3 and the fact that perturbation is only drawn

when $\|y_t^{(i)}\| \leq C_v$ and n_t nodes draw perturbation in iteration t. Sum Eq. (D.15) and apply Lemma D.1, we have

$$\begin{aligned} f(\bar{x}_T) \leq f(x_0) - \frac{1}{4\eta}\sum_{t=0}^{T-1}\omega_t - \frac{\eta}{2}\sum_{t=0}^{T-1}\|\bar{v}_t\|^2 + (1+\frac{768\log(4/\delta)}{nb_1\beta})\frac{L^2\eta}{n}\sum_{t=0}^{T-1}\|X_t-\bar{X}_t\|_F^2 \\ + \sum_{t=0}^{T-1}\frac{p_t(\eta^2C_v^2+r^2)}{\eta} + \frac{32\log(4/\delta)\beta\eta T\sigma^2}{nb_1} + (1+\frac{384\log(4/\delta)L\eta}{nb_1\beta})\sum_{t=0}^{T-1}L\omega_t \\ + \frac{4\log(4/\delta)\eta\sigma^2}{n\beta b_0} \end{aligned} \tag{D.16}$$

According to the update of gradient tracker, we have $\bar{y}_t = \bar{v}_t$. By Lemma D.10 we have

$$\frac{1}{n}\sum_{i=1}^{n}\|x_{t+1}^{(i)} - x_t^{(i)}\|^2 = \omega_t + \frac{1}{n}\|(X_{t+1}-\bar{X}_{t+1}) - (X_t-\bar{X}_t)\|_F^2 \tag{D.17}$$

$$\frac{1}{n}\sum_{i=1}^{n}\|y_t^{(i)}\|^2 = \|\bar{y}_t\|^2 + \frac{1}{n}\|Y_t-\bar{Y}_t\|_F^2 \tag{D.18}$$

Divide the term $\|\bar{v}_t\|^2$ in Eq. (D.16) into three portions and we get

$$\begin{aligned} & f(\bar{x}_T) \\ & \leq f(x_0) - \frac{1}{8\eta}\sum_{t=0}^{T-1}\omega_t - \frac{(1-\lambda)^2}{256\eta}\sum_{t=0}^{T-1}\omega_t - \frac{\eta}{2}\sum_{t=0}^{T-1}\|\bar{v}_t\|^2 + (1+\frac{768\log(4/\delta)}{nb_1\beta})\frac{L^2\eta}{n}\sum_{t=0}^{T-1}\|X_t-\bar{X}_t\|_F^2 \\ & \quad + \sum_{t=0}^{T-1}\frac{p_t(\eta^2C_v^2+r^2)}{\eta} + \frac{32\log(4/\delta)\beta\eta T\sigma^2}{nb_1} + (1+\frac{384\log(4/\delta)L\eta}{nb_1\beta})\sum_{t=0}^{T-1}L\omega_t + \frac{4\log(4/\delta)\eta\sigma^2}{n\beta b_0} \\ & \leq f(x_0) - \frac{1}{8\eta}\sum_{t=0}^{T-1}\omega_t - \frac{(1-\lambda)^2}{256\eta}\sum_{t=0}^{T-1}\omega_t - \frac{\eta}{2n}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\|y_t^{(i)}\|^2 + \frac{\eta}{2n}\sum_{t=0}^{T-1}\|Y_t-\bar{Y}_t\|_F^2 \\ & \quad + \sum_{t=0}^{T-1}\frac{p_t(\eta^2C_v^2+r^2)}{\eta} + (1+\frac{768\log(4/\delta)}{nb_1\beta})\frac{L^2\eta}{n}\sum_{t=0}^{T-1}\|X_t-\bar{X}_t\|_F^2 + \frac{32\log(4/\delta)\beta\eta T\sigma^2}{nb_1} \\ & \quad + (1+\frac{384\log(4/\delta)L\eta}{nb_1\beta})\sum_{t=0}^{T-1}L\omega_t + \frac{4\log(4/\delta)\eta\sigma^2}{n\beta b_0} \\ & \leq f(x_0) - \frac{1}{8\eta}\sum_{t=0}^{T-1}\omega_t - \frac{(1-\lambda)^2}{256n\eta}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\|x_{t+1}^{(i)} - x_t^{(i)}\|^2 - \frac{\eta}{2n}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\|y_t^{(i)}\|^2 \\ & \quad + (1+\frac{768\log(4/\delta)}{nb_1\beta} + \frac{(1-\lambda)^2}{128L^2\eta^2})\frac{L^2\eta}{n}\sum_{t=0}^{T-1}\|X_t-\bar{X}_t\|_F^2 + \frac{\eta}{2n}\sum_{t=0}^{T-1}\|Y_t-\bar{Y}_t\|_F^2 \end{aligned}$$

$$
\begin{aligned}
&+(1+\frac{384\log(4/\delta)L\eta}{nb_1\beta})\sum_{t=0}^{T-1}L\omega_t+\sum_{t=0}^{T-1}\frac{p_t(\eta^2C_v^2+r^2)}{\eta}+\frac{32\log(4/\delta)\beta\eta T\sigma^2}{nb_1}+\frac{4\log(4/\delta)\eta\sigma^2}{n\beta b_0}\\
&\leq f(x_0)-\frac{1}{8L\eta}\sum_{t=0}^{T-1}L\omega_t-\frac{(1-\lambda)^2}{256n\eta}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\|x_{t+1}^{(i)}-x_t^{(i)}\|^2-\frac{\eta}{2n}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\|y_t^{(i)}\|^2\\
&+A_1\sum_{t=0}^{T-1}L\omega_t+A_2\frac{T\beta\eta\sigma^2}{b_1}+A_3\frac{\eta\sigma^2}{\beta b_0}+A_4\sum_{t=0}^{T-1}\frac{p_t(\eta^2C_v^2+r^2)}{\eta}+A_5\frac{\eta}{n}\sum_{i=1}^{n}\|\nabla f_i(x_0)\|^2 \qquad \text{(D.19)}
\end{aligned}
$$

In the second inequality we use Eq. (D.18). In the third inequality we use Eq. (D.17) and Cauchy-Schwartz inequality. In the last inequality we use Lemma D.2 and the coefficients are

$$
\begin{aligned}
A_1&=1+\frac{384\log(4/\delta)L\eta}{nb_1\beta}+(1+\frac{768\log(4/\delta)}{nb_1\beta}+\frac{(1-\lambda)^2}{128L^2\eta^2})\frac{160000\log(4/\delta)L^3\eta^3\lambda^4}{(1-\lambda)^4\min\{b_1\beta,1\}}+\frac{774\log(4/\delta)L\eta\lambda^2}{(1-\lambda)}\\
A_2&=\frac{32\log(4/\delta)}{n}+(1+\frac{768\log(4/\delta)}{nb_1\beta}+\frac{(1-\lambda)^2}{128L^2\eta^2})\frac{12288\log(4/\delta)L^2\eta^2\lambda^4}{(1-\lambda)^4}+\frac{64\log(4/\delta)\lambda^2}{(1-\lambda)}\\
A_3&=\frac{4\log(4/\delta)}{n}+(1+\frac{768\log(4/\delta)}{nb_1\beta}+\frac{(1-\lambda)^2}{128L^2\eta^2})\frac{2000\log(4/\delta)L^2\eta^2\lambda^4}{(1-\lambda)^4}+\frac{10\log(4/\delta)\lambda^2}{(1-\lambda)}\\
A_4&=1+(1+\frac{768\log(4/\delta)}{nb_1\beta}+\frac{(1-\lambda)^2}{128L^2\eta^2})\frac{64\lambda^2L^2\eta^2}{(1-\lambda)^2}+\frac{25000\log(4/\delta)L^2\eta^2\lambda^4}{(1-\lambda)^3}\\
A_5&=(1+\frac{768\log(4/\delta)}{nb_1\beta}+\frac{(1-\lambda)^2}{128L^2\eta^2})\frac{128\lambda^4L^2\eta^2}{(1-\lambda)^3}+\frac{2\lambda^2}{1-\lambda}
\end{aligned}
$$

According to the parameter setting, we have $A_1\leq\frac{1}{16L\eta}$, $A_2\leq\frac{200\log(4/\delta)}{(1-\lambda)^2}$, $A_3\leq\frac{40\log(4/\delta)}{(1-\lambda)^2}$, $A_4\leq\frac{7}{4}$ and $A_5\leq\frac{5}{1-\lambda}$. Therefore, we have

$$
\begin{aligned}
f(\bar{x}_T)&\leq f(x_0)+\frac{40\log(4/\delta)\eta\sigma^2}{(1-\lambda)^2\beta b_0}+\frac{5\eta}{(1-\lambda)n}\sum_{i=1}^{n}\|\nabla f_i(x_0)\|^2-\sum_{t=0}^{T-1}\mathscr{D}_t\\
&\leq f(x_0)+\frac{\sigma^2}{L}+\frac{1}{nL}\sum_{i=1}^{n}\|\nabla f_i(x_0)\|^2-\sum_{t=0}^{T-1}\mathscr{D}_t \qquad \text{(D.20)}
\end{aligned}
$$

where

$$
\mathscr{D}_t=\frac{1}{16\eta}\omega_t+\frac{(1-\lambda)^2}{256n\eta}\sum_{i=1}^{n}\|x_{t+1}^{(i)}-x_t^{(i)}\|^2+\frac{\eta}{2n}\sum_{i=1}^{n}\|y_t^{(i)}\|^2-\frac{200\eta\varepsilon^2\sigma^2}{(1-\lambda)^2C_1^2}-\frac{7p_t(\eta^2C_v^2+r^2)}{4\eta} \qquad \text{(D.21)}
$$

which reaches the conclusion. □

D.3.4 Proof of Lemma D.4

Proof. For convenience, the iteration that draws perturbation is considered to be included in the escaping phase. If an iteration belongs to type-A, *i.e.*, $p_t \geq \frac{1}{5}$, then at least $n/5$ worker nodes are in the escaping phase. If an iteration belongs to type-C, we have $\frac{1}{n}\sum_{i=1}^{n}\|y_t^{(i)}\|^2 \leq \frac{4C_v^2}{5}$. Therefore, there are at least $\frac{n}{5}$ worker nodes satisfying $\|y_t^{(i)}\| \leq C_v$, which also indicates that at least $\frac{n}{5}$ worker nodes are in the escaping phase. Then if iteration t is either type-A or type-C, there must be $n/5$ worker nodes in the escaping phase. We denote the set of these $n/5$ worker nodes as $\mathscr{E}_t$. Furthermore, if this iteration t is not one of the last C_T iterations before termination, then there must exist $n/10$ worker nodes out of $\mathscr{E}_t$ such that they have not met the condition $esc^{(i)} \geq C_T$ and will break the escaping phase before meeting the condition because of the termination criterion in Algorithm 6. We use $\mathscr{B}_t$ to denote these worker nodes.

For each $i \in \mathscr{B}_t$, we have an interval $[a_t^{(i)}, b_t^{(i)}]$ such that $t \in [a_t^{(i)}, b_t^{(i)}]$ and node i enters escaping phase in iteration $a_t^{(i)}$ and breaks escaping phase in iteration $b_t^{(i)}$. Besides, we also have

$$b_t^{(i)} - a_t^{(i)} \leq C_T \quad \text{and} \quad \|x_{b_t^{(i)}}^{(i)} - x_{a_t^{(i)}}^{(i)}\| \geq C_d$$

Then by Cauchy-Schwartz inequality we have

$$C_d^2 \leq \|x_{b_t^{(i)}}^{(i)} - x_{a_t^{(i)}}^{(i)}\|^2 \leq C_T \sum_{t=a_t^{(i)}}^{b_t^{(i)}} \|x_{t+1}^{(i)} - x_t^{(i)}\|^2 \tag{D.22}$$

Let $a_t = \min_i\{a_t^{(i)}\}$ and $b_t = \max_i\{b_t^{(i)}\}$. It is easy to check that $b_t - a_t \leq 2C_T$. Next, we will perform the refining step. If $t < t'$ are two iterations that are either type-A or type-C and $t' \in [a_t, b_t]$, then we make $a_{t'} = a_t$ and $b_{t'} = b_t$. Let $\mathscr{I} = \cup_t [a_t, b_t]$ for all type-A and type-C

iterations t. Then I can be written as disjoint union of

$$\mathscr{I} = \mathscr{I}_1 \cup \mathscr{I}_2 \cup \cdots \cup \mathscr{I}_k \tag{D.23}$$

because if $a_t \leq a_{t'} \leq b_t$ then $[a_t, b_t]$ and $[a_{t'}, b_{t'}]$ can be merged into one interval. Now we can see for each iteration t that is either type-A or type-C and t is not one of the last C_T iterations, we have $t \in \mathscr{I}$. Next we will estimate the descent over $\mathscr{I}$. Without loss of generality, we consider an interval $\mathscr{I}_j$. $\mathscr{I}_j$ can be expressed by union $\mathscr{J}_1 \cup \cdots \cup \mathscr{J}_l$ where $\mathscr{J}_m = [a_{t_m}, b_{t_m}]$ for some t_m, $m = 1, \cdots, l$. Because of the refining step, we have each t_m is only included in interval $\mathscr{J}_m$ and the intersection of any three intervals in $\mathscr{J}_1, \cdots, \mathscr{J}_l$ is $\emptyset$. According to Eq. (D.22) we have

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{t \in \mathscr{J}_m} \|x_{t+1}^{(i)} - x_t^{(i)}\|^2 \geq \frac{C_d^2}{10 C_T} \tag{D.24}$$

since $|\mathscr{B}_t| \geq \frac{n}{10}$. Next, we will consider the intersection of $\mathscr{J}_m$ and $\mathscr{J}_{m+1}$. Notice that when estimating Eq. (D.24) we only add the terms $\|x_{t+1}^{(i)} - x_t^{(i)}\|^2$ on nodes $i \in \mathscr{B}_{t_m}$ and in the intervals $[a_{t_m}^{(i)}, b_{t_m}^{(i)}]$. Therefore, for any node $i \notin \mathscr{B}_{t_m} \cap \mathscr{B}_{t_{m+1}}$, the terms used to estimate Eq. (D.24) will not be added repeatedly. If $i \in \mathscr{B}_{t_m} \cap \mathscr{B}_{t_{m+1}}$, we have $[a_{t_m}^{(i)}, b_{t_m}^{(i)}]$ and $[a_{t_{m+1}}^{(i)}, b_{t_{m+1}}^{(i)}]$ are disjoint because $t_{m+1} \in [a_{t_{m+1}}^{(i)}, b_{t_{m+1}}^{(i)}]$ but $t_{m+1} \notin [a_{t_m}^{(i)}, b_{t_m}^{(i)}]$ and a node cannot draw perturbation before breaking the escaping phase. Hence we can sum Eq. (D.24) over m and achieve

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{t \in \mathscr{I}_j} \|x_{t+1}^{(i)} - x_t^{(i)}\|^2 \geq \frac{l C_d^2}{10 C_T} \tag{D.25}$$

Since the length of each $\mathscr{J}_m$ is not larger than $2C_T$, we have

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{t\in\mathscr{I}_j}\|x_{t+1}^{(i)}-x_t^{(i)}\|^2\geq\frac{|\mathscr{I}_j|C_d^2}{20C_T^2}\text{ and }\frac{1}{n}\sum_{i=1}^{n}\sum_{t\in\mathscr{I}}\|x_{t+1}^{(i)}-x_t^{(i)}\|^2\geq\frac{|\mathscr{I}|C_d^2}{20C_T^2}\tag{D.26}$$

Combining Eq. (D.26) and Lemma D.3, we can estimate the descent over $\mathscr{I}$ by

$$\sum_{t\in\mathscr{I}}\mathscr{D}_t\geq|\mathscr{I}|\Big(\frac{(1-\lambda)^2C_d^2}{5120\eta C_T^2}-\frac{200\eta\varepsilon^2\sigma^2}{(1-\lambda)^2C_1^2}-\frac{7(\eta^2C_v^2+r^2)}{4\eta}\Big)\geq|\mathscr{I}|\cdot\frac{(1-\lambda)^2C_2^2\eta\varepsilon^2}{10000}\tag{D.27}$$

according to the parameter setting. □

D.3.5 Proof of Lemma D.5

Proof. According to Lemma D.3 and the definition of type-B iteration, we have

$$\mathscr{D}_t\geq\frac{\eta C_v^2}{20}-\frac{200\eta\varepsilon^2}{(1-\lambda)^2C_1^2}-\frac{7r^2}{20\eta}\geq\frac{\eta C_v^2}{40}-\frac{200\eta\varepsilon^2\sigma^2}{(1-\lambda)^2C_1^2}\geq\frac{(1-\lambda)^2C_2^2\eta\varepsilon^2}{8000000}\tag{D.28}$$

for all type-B iteration t where we have used the parameter setting. □

D.3.6 Proof of Lemma D.6

Proof. Suppose the conclusion is not true and we will find the conflict. Thus, we have the assumption that there are at least $\frac{n}{10}$ worker nodes satisfying $d_i\leq 2C_d$. First, we define

$$w_t^{(i)}=x_t^{(i)}-x_t^{(i)\prime},\ w_t=\bar{x}_t-\bar{x}_t',\ \mathscr{H}=\nabla^2 f(\bar{x}_s),\ \mathscr{H}^{(i)}=\nabla^2 f_i(\bar{x}_s),\ \mathscr{H}_t^{(i)}=\nabla^2 F_i(x_s^{(i)},\xi_t^{(i)})$$

$$\zeta_t = \frac{1}{n}\sum_{i=1}^{n}(\nabla F_i(x_t^{(i)}, \xi_t^{(i)}) - \nabla F_i(\bar{x}_t, \xi_t^{(i)})) - (\nabla F_i(x_t^{(i)'}, \xi_t^{(i)}) - \nabla F_i(\bar{x}'_t, \xi_t^{(i)}))$$
$$-(1-\beta)(\nabla F_i(x_{t-1}^{(i)}, \xi_t^{(i)}) - \nabla F_i(\bar{x}_{t-1}, \xi_t^{(i)})) - (\nabla F_i(x_{t-1}^{(i)'}, \xi_t^{(i)}) - \nabla F_i(\bar{x}'_{t-1}, \xi_t^{(i)}))$$

$$\nu_t = \bar{v}_t - \nabla f(\bar{x}_t) - (\bar{v}'_t - \nabla f(\bar{x}'_t)) - \zeta_t$$

and

$$\bar{\Delta}_t = \int_0^1 (\nabla^2 f(\bar{x}'_t + \theta(\bar{x}_t - \bar{x}'_t)) - \mathscr{H})d\theta$$
$$\Delta_t^{(i)} = \int_0^1 (\nabla^2 f_i(\bar{x}'_t + \theta(\bar{x}_t - \bar{x}'_t)) - \mathscr{H}^{(i)})d\theta$$

Then we have

$$\begin{aligned} w_t &= w_{t-1} - \eta(\bar{v}_{t-1} - \bar{v}'_{t-1}) \\ &= w_{t-1} - \eta(\nabla f(\bar{x}_{t-1}) - \nabla f(\bar{x}'_{t-1}) + \bar{v}_{t-1} - \nabla f(\bar{x}_{t-1}) - \bar{v}'_{t-1} + \nabla f(\bar{x}'_{t-1})) \\ &= w_{t-1} - \eta\left[(\bar{x}_{t-1} - \bar{x}'_{t-1})\int_0^1 \nabla^2 f(\bar{x}'_{t-1} + \theta(\bar{x}_{t-1} - \bar{x}'_{t-1}))d\theta + \nu_{t-1} + \zeta_{t-1}\right] \\ &= (I - \eta\mathscr{H})w_{t-1} - \eta(\bar{\Delta}_{t-1}w_{t-1} + \nu_{t-1} + \zeta_{t-1}) \end{aligned} \tag{D.29}$$

Here term ζ_t is yield from consensus error and does not exist in centralized algorithms. Applying recursion to Eq. (D.29), we can obtain

$$w_t = (I - \eta\mathscr{H})^{t-s-1}w_{s+1} - \eta\sum_{\tau=s+1}^{t-1}(I - \eta\mathscr{H})^{t-\tau-1}(\bar{\Delta}_\tau w_\tau + \nu_\tau + \zeta_\tau) \tag{D.30}$$

Let $q_t = \eta \sum_{\tau=s+1}^{t-1} (I - \eta \mathscr{H})^{t-\tau-1} (\bar{\Delta}_\tau w_\tau + \nu_\tau + \zeta_\tau)$. We will prove

$$\|q_t\| \leq \frac{1}{2}(1+\eta\gamma)^{t-s-1} p_s r_0 \tag{D.31}$$

which leads to

$$\frac{1}{2}(1+\eta\gamma)^{t-s-1} p_s r_0 \leq \|w_t\| \leq \frac{3}{2}(1+\eta\gamma)^{t-s-1} p_s r_0 \tag{D.32}$$

because $\|(I - \eta \mathscr{H})^{t-s-1} w_{s+1}\| = (1+\eta\gamma)^{t-s-1} p_s r_0$ according to the definition of w_{s+1}. We define $\bar{d} = \max_{s \leq t \leq s+C_T} \{\|\bar{x}_t - \bar{x}_s\|, \|\bar{x}'_t - \bar{x}_s\|\}$. Since at least $\frac{n}{10}$ nodes satisfy $d_i \leq 2C_d$, $C_d = \tilde{O}(\varepsilon^{1-\alpha})$ and the averaged consensus error is bounded by $O(\varepsilon^{2(1+\theta)})$, we have

$$d_i \leq 3C_d \quad \text{and} \quad \bar{d} \leq \frac{1}{n}\sum_{i=1}^{n} d_i \leq 3C_d \tag{D.33}$$

To achieve Eq. (D.31), it is sufficient to prove

$$\eta \sum_{\tau=s+1}^{t-1} (1+\eta\gamma)^{t-\tau-1} \|\bar{\Delta}_\tau w_\tau\| + \|\nu_\tau\| + \|\zeta_\tau\| \leq \frac{1}{2}(1+\eta\gamma)^{t-s-1} p_s r_0 \tag{D.34}$$

$$\|\nu_t\| \leq \sqrt{\frac{4\log(4/\delta)}{b_1}} \cdot \frac{(1+\eta\gamma)^{t-s-1} L p_s r_0}{t-s} + \frac{1}{12\eta C_T}(1+\eta\gamma)^{t-s-1} p_s r_0 \tag{D.35}$$

$$\|\zeta_t\| \leq 8\left(\frac{1+\lambda^2}{2}\right)^{\frac{t-s-1}{2}} L\sqrt{p_s} r_0 + \frac{L\eta(1+\eta\gamma)^{t-s-1} L p_s r_0}{\sqrt{b_1}(t-s)} + \frac{1}{12\eta C_T}(1+\eta\gamma)^{t-s-1} p_s r_0 \tag{D.36}$$

which can be derived by induction. When $t = s+1$, the left side of Eq. (D.34) is 0 and thus the inequality is satisfied. Suppose Eq. (D.34) holds for $t \leq t_0$. When $t = t_0 + 1$, we have

$$\begin{aligned}
&\eta \sum_{\tau=s+1}^{t-1} (1+\eta\gamma)^{t-\tau-1} \|\bar{\Delta}_\tau w_\tau\| \\
&\leq \frac{3}{2}\eta\rho\bar{d} \sum_{\tau=s+1}^{t-1} (1+\eta\gamma)^{t-s-2} p_s r_0 \leq 5\eta\rho C_d C_T (1+\eta\gamma)^{t-s-2} p_s r_0 \\
&\leq \frac{1}{6}(1+\eta\gamma)^{t-s-1} p_s r_0
\end{aligned} \tag{D.37}$$

where we use Assumption 5.4 and the case of $t \leq t_0$ in the first two inequalities. We use the parameter setting of C_d in the last inequality. Next, we will estimate the terms related to ν_t. By Azuma-Hoeffding inequality we know Eq. (D.35) is satisfied when $t = s+1$. We define

$$\begin{aligned}
\varepsilon_{t,i} &= (\nabla F_i(\bar{x}_{t+1}, \xi_{t+1}^{(i)}) - \nabla f_i(\bar{x}_{t+1})) - (1-\beta)(\nabla F_i(\bar{x}_t, \xi_{t+1}) - \nabla f_i(\bar{x}_t)) \\
\varepsilon'_{t,i} &= (\nabla F_i(\bar{x}'_{t+1}, \xi_{t+1}^{(i)}) - \nabla f_i(\bar{x}'_{t+1})) - (1-\beta)(\nabla F_i(\bar{x}'_t, \xi_{t+1}) - \nabla f_i(\bar{x}'_t))
\end{aligned}$$

Then according to the definition of ν_t we have

$$\nu_{t+1} = (1-\beta)\nu_t + \frac{1}{n}\sum_{i=1}^{n}(\varepsilon_{t,i} - \varepsilon'_{t,i}) = \frac{1}{n}\sum_{\tau=s}^{t}(1-\beta)^{t-\tau}\sum_{i=1}^{n}(\varepsilon_{\tau,i} - \varepsilon'_{\tau,i}) \tag{D.38}$$

Define

$$\begin{aligned}
\tilde{\Delta}_{t,1}^{(i)} &= \int_0^1 (\nabla^2 F_i(\bar{x}'_t + \theta(\bar{x}_t - \bar{x}'_t), \xi_t^{(i)}) - \mathscr{H}_t^{(i)}) d\theta \\
\tilde{\Delta}_{t,2}^{(i)} &= \int_0^1 (\nabla^2 F_i(\bar{x}'_{t-1} + \theta(\bar{x}_{t-1} - \bar{x}'_{t-1}), \xi_t^{(i)}) - \mathscr{H}_t^{(i)}) d\theta \\
\hat{\Delta}_{t,1}^{(i)} &= \int_0^1 (\nabla^2 F_i(x_t^{(i)'} + \theta(x_t^{(i)} - x_t^{(i)'}), \xi_t^{(i)}) - \mathscr{H}_t^{(i)}) d\theta \\
\hat{\Delta}_{t,2}^{(i)} &= \int_0^1 (\nabla^2 F_i(x_{t-1}^{(i)'} + \theta(x_{t-1}^{(i)} - x_{t-1}^{(i)'}), \xi_t^{(i)}) - \mathscr{H}_t^{(i)}) d\theta
\end{aligned} \tag{D.39}$$

Then we have

$$
\begin{aligned}
&\varepsilon_{t,i} - \varepsilon'_{t,i} \\
&= \mathscr{H}^{(i)}_{t+1} w_{t+1} + \tilde{\Delta}^{(i)}_{t+1,1} w_{t+1} - \mathscr{H}^{(i)} w_{t+1} - \Delta^{(i)}_{t+1} w_{t+1} + (1-\beta)(\mathscr{H}^{(i)} w_t + \Delta^{(i)}_t w_t) \\
&\quad - (1-\beta)(\mathscr{H}^{(i)}_{t+1} w_t + \tilde{\Delta}^{(i)}_{t+1,2} w_t) \\
&= (\mathscr{H}^{(i)}_{t+1} - \mathscr{H})(w_{t+1} - (1-\beta) w_t) + (\tilde{\Delta}^{(i)}_{t+1,1} - \Delta^{(i)}_{t+1}) w_{t+1} + (1-\beta)(\Delta^{(i)}_t - \tilde{\Delta}^{(i)}_{t+1,2}) w_t
\end{aligned} \tag{D.40}
$$

According to Assumption 5.3 and Assumption 5.4, we have

$$
\|\varepsilon_{t,i} - \varepsilon'_{t,i}\| \leq 2L\|w_{t+1} - w_t\| + (2\beta L + 3\rho C_d)\|w_t\| + 3\rho C_d \|w_{t+1}\| \tag{D.41}
$$

Applying Azuma-Hoeffding inequality to Eq. (D.38), with Eq. (D.41) we can obtain

$$
\begin{aligned}
\|\nu_t\|^2 &\leq \frac{4\log(4/\delta)}{nb_1} \sum_{\tau=s}^{t-1} [2L\|w_{\tau+1} - w_\tau\| + (2\beta L + 3\rho C_d)\|w_\tau\| + 3\rho C_d \|w_{\tau+1}\|]^2 \\
&\leq \frac{48\log(4/\delta)}{nb_1} \sum_{\tau=s+1}^{t} (L^2 \|w_\tau - w_{\tau-1}\|^2 + 5\rho^2 C_d^2 \|w_\tau\|^2)
\end{aligned} \tag{D.42}
$$

since β is $\Theta(\varepsilon^{1+\theta})$ and C_d is $\Theta(\varepsilon^{1-\alpha})$. According to Eq. (D.30), we have

$$
\begin{aligned}
&L\|w_\tau - w_{\tau-1}\| \\
&= L\| -\eta \mathscr{H}(I - \eta\mathscr{H})^{\tau-s-2} w_{s+1} - \eta \sum_{\tau'=s+1}^{\tau-2} \eta \mathscr{H}(I - \eta\mathscr{H})^{\tau'-s-2} (\bar{\Delta}_{\tau'} w_{\tau'} + \nu_{\tau'} + \zeta_{\tau'}) \\
&\quad + \eta(\bar{\Delta}_{\tau-1} w_{\tau-1} + \nu_{\tau-1} + \zeta_{\tau-1})\| \\
&\leq L\eta\gamma(1+\eta\gamma)^{\tau-s-2} p_s r_0 + \frac{L\eta\gamma}{2}(1+\eta\gamma)^{\tau-s-2} p_s r_0 + L\eta\|\bar{\Delta}_{\tau-1} w_{\tau-1} + \nu_{\tau-1} + \zeta_{\tau-1}\| \\
&\leq 2L\eta\gamma(1+\eta\gamma)^{\tau-s-2} p_s r_0 + L\eta\|\bar{\Delta}_{\tau-1} w_{\tau-1} + \nu_{\tau-1} + \zeta_{\tau-1}\|
\end{aligned} \tag{D.43}
$$

In the first inequality, the first term is derived by the definition of w_{s+1}. The second term is derived by the supposition that Eq. (D.34) holds for $t \leq t_0$ and the fact that Eq. (D.34) implies

$$\eta \sum_{\tau=s+1}^{t-1} (1+\eta\gamma)^{t-\tau-1} \|\bar{\Delta}_\tau w_\tau + \nu_\tau + \zeta_\tau\| \leq \frac{1}{2}(1+\eta\gamma)^{t-s-1} p_s r_0 \tag{D.44}$$

Combining Eq. (D.42) and Eq. (D.43), we have

$$\begin{aligned}
\|\nu_t\|^2 &\leq \frac{48\log(4/\delta)}{nb_1} \sum_{\tau=s+1}^{t} (L^2\|w_\tau - w_{\tau-1}\|^2 + 5\rho^2 C_d^2 \|w_\tau\|^2) \\
&\leq \frac{270\log(4/\delta)\rho^2 C_d^2}{nb_1\eta\gamma}(1+\eta\gamma)^{2(t-s-1)} p_s^2 r_0^2 + \frac{192\log(4/\delta)L^2\eta\gamma}{nb_1}(1+\eta\gamma)^{2(t-s-1)} p_s^2 r_0^2 \\
&\quad + \frac{96\log(4/\delta)L^2\eta^2}{nb_1} \sum_{\tau=s+1}^{t-2} \|\bar{\Delta}_\tau w_\tau + \nu_\tau + \zeta_\tau\|^2 \\
&\leq \frac{300\log(4/\delta)\rho^2 C_d^2}{nb_1\eta\gamma}(1+\eta\gamma)^{2(t-s-1)} p_s^2 r_0^2 + \frac{192\log(4/\delta)L^2\eta\gamma}{nb_1}(1+\eta\gamma)^{2(t-s-1)} p_s^2 r_0^2 \\
&\quad + \frac{4\log(4/\delta)L^2}{nb_1\eta\gamma C_T^2}(1+\eta\gamma)^{2(t-s-1)} p_s^2 r_0^2 + \frac{5000\log^2(4/\delta)L^4\eta^2 p_s^2 r_0^2}{b_1^2} \sum_{\tau=s+1}^{t-2} \frac{(1+\eta\gamma)^{2(\tau-s-1)}}{(\tau-s)^2} \\
&\leq \frac{1}{288\eta^2 C_T^2}(1+\eta\gamma)^{2(t-s-1)} p_s^2 r_0^2 + \frac{800\log^2(4/\delta)L^2\eta\gamma}{nb_1}(1+\eta\gamma)^{2(t-s-1)} p_s^2 r_0^2 \\
&\quad + \frac{5000\log^2(4/\delta)L^4\eta^2 p_s^2 r_0^2}{b_1^2} \sum_{\tau=s+1}^{s+\frac{L\eta}{b_1\gamma}} \frac{(1+\eta\gamma)^{2(\tau-s-1)}}{(\tau-s)^2} \\
&\leq \frac{10000\log(4/\delta)L^4\eta^4}{b_1^4\gamma^2} \cdot \frac{(1+\eta\gamma)^{2(t-s-1)}L^2 p_s^2 r_0^2}{(t-s)^2} + \frac{1}{144\eta^2 C_T^2}(1+\eta\gamma)^{2(t-s-1)} p_s^2 r_0^2 \\
&\leq \frac{(1+\eta\gamma)^{2(t-s-1)}L^2 p_s^2 r_0^2}{(t-s)^2} + \frac{1}{144\eta^2 C_T^2}(1+\eta\gamma)^{2(t-s-1)} p_s^2 r_0^2
\end{aligned} \tag{D.45}$$

The exponential term in Eq. (D.36) can be addressed by the following strategy. When $t \geq \tilde{O}(\frac{1}{1-\lambda})$, the term can be dominated by other terms such as $\frac{1}{\eta C_T}$. When $t < \tilde{O}(\frac{1}{1-\lambda})$, it

can be bounded by

$$\frac{L^2\eta^2\log(4/\delta)p_s r_0^2}{n(1-\lambda)b_1} \leq \frac{L^2\eta^2(t-s)^2\log(4/\delta)p_s^2 r_0^2}{(1-\lambda)b_1(t-s)^2} \tag{D.46}$$

The term $t-s$ in the numerator will be bounded by η in this case and hence it can be merged to the first term in Eq. (D.35). In the third inequality of Eq. (D.45), we split the last term into two parts: $\tau - s > \frac{L\eta}{b_1\gamma}$ and $\tau - s \leq \frac{L\eta}{b_1\gamma}$. Since $\int_t^{+\infty}\frac{dx}{x^2} = \frac{1}{t}$, we can merge the case $\tau - s > \frac{L\eta}{b_1\gamma}$ into the second term and estimate the rest one where $\tau - s$ is small. According to the choice of θ, we have $b_1 \geq \Theta(\varepsilon^{2-\theta-5\alpha})$ and $\frac{\eta^2 C_d^2 C_T^3}{b_1} \leq O(1)$ and hence get the estimation in Eq. (D.45). We should notice that we use the relation $\frac{\eta}{b_1\gamma} \leq O(1)$ in our proof, which is automatically satisfied. By Eq. (D.45) we can reach the conclusion in Eq. (D.35). Furthermore, we have

$$\begin{aligned}
&\eta\sum_{\tau=s+1}^{t-1}(1+\eta\gamma)^{t-\tau-1}\|v_\tau\| \\
\leq\ & L\eta(1+\eta\gamma)^{t-s-1}p_s r_0(\sum_{\tau=s+1}^{t-1}\frac{1}{\tau-s}) + \frac{1}{12}(1+\eta\gamma)^{t-s-1}p_s r_0 \\
\leq\ & L\eta\log(C_T)(1+\eta\gamma)^{t-s-1}p_s r_0 + \frac{1}{12}(1+\eta\gamma)^{t-s-1}p_s r_0 \leq \frac{1}{6}(1+\eta\gamma)^{t-s-1}p_s r_0
\end{aligned} \tag{D.47}$$

The last step to prove Eq. (D.34) is to estimate the term corresponding to ζ_t, which is a new term only occurred in decentralized algorithms. Recall the definitions in Eq. (D.39), we have

$$\begin{aligned}
\zeta_t &= \frac{1}{n}\sum_{i=1}^n\Big[(\mathscr{H}_t^{(i)}+\hat{\Delta}_{t,1}^{(i)})w_t^{(i)} - (\mathscr{H}_t^{(i)}+\tilde{\Delta}_{t,1}^{(i)})w_t - (1-\beta)((\mathscr{H}_t^{(i)}+\hat{\Delta}_{t,2}^{(i)})w_{t-1}^{(i)} - (\mathscr{H}_t^{(i)}+\tilde{\Delta}_{t,2}^{(i)})w_{t-1})\Big] \\
&= \frac{1}{n}\sum_{i=1}^n\mathscr{H}_t^{(i)}[(w_t^{(i)}-w_t) - (1-\beta)(w_{t-1}^{(i)}-w_{t-1})] + \frac{1}{n}\sum_{i=1}^n\hat{\Delta}_{t,1}^{(i)}(w_t^{(i)}-w_t) \\
&\quad + \frac{1}{n}\sum_{i=1}^n(\hat{\Delta}_{t,1}^{(i)}-\tilde{\Delta}_{t,1}^{(i)})w_t - \frac{1-\beta}{n}\sum_{i=1}^n\hat{\Delta}_{t,2}^{(i)}(w_{t-1}^{(i)}-w_{t-1}) - \frac{1-\beta}{n}\sum_{i=1}^n(\hat{\Delta}_{t,2}^{(i)}-\tilde{\Delta}_{t,2}^{(i)})w_{t-1}
\end{aligned} \tag{D.48}$$

Then by Assumption 5.3, Assumption 5.4, Eq. (D.33), Lemma D.10 and Cauchy-Schwartz inequality, we have

$$\begin{aligned}\|\zeta_t\|^2 \leq \frac{4L^2}{n}(\|X_t-\bar{X}_t-(X_t'-\bar{X}_t')\|_F^2+\|X_{t-1}-\bar{X}_{t-1}-(X_{t-1}'-\bar{X}_{t-1}')\|_F^2)\\ +144\rho^2C_d^2(\|w_t\|^2+\|w_{t-1}\|^2)\end{aligned} \tag{D.49}$$

It is sufficient to prove

$$\begin{aligned}\frac{L}{\sqrt{n}}\|X_t-\bar{X}_t-(X_t'-\bar{X}_t')\|_F \leq 2(\frac{1+\lambda^2}{2})^{\frac{t-s-1}{2}}L\sqrt{p_s}r_0+\frac{1}{48\eta C_T}(1+\eta\gamma)^{t-s-1}p_sr_0\\ +\frac{L\eta(1+\eta\gamma)^{t-s-1}Lp_sr_0}{4\sqrt{b_1}(t-s)}\end{aligned} \tag{D.50}$$

because of Eq. (D.49) and the parameter setting. Eq. (D.50) can also be proven by induction. When $t=s+1$ the condition is satisfied. Next we will estimate $\|X_t-\bar{X}_t-(X_t'-\bar{X}_t')\|_F^2$. By Assumption 5.5 and Young's inequality we have

$$\begin{aligned}&\|X_t-\bar{X}_t-(X_t'-\bar{X}_t')\|_F^2\\ &=\|(W-J)[(X_{t-1}-\bar{X}_{t-1}-(X_{t-1}'-\bar{X}_{t-1}'))-\eta(Y_{t-1}-\bar{Y}_{t-1}-(Y_{t-1}'-\bar{Y}_{t-1}'))]\|_F^2\\ &\leq\frac{1+\lambda^2}{2}\|X_{t-1}-\bar{X}_{t-1}-(X_{t-1}'-\bar{X}_{t-1}')\|_F^2+\frac{2\eta^2\lambda^2}{1-\lambda^2}\|Y_{t-1}-\bar{Y}_{t-1}-(Y_{t-1}'-\bar{Y}_{t-1}')\|_F^2\\ &\leq\frac{2\eta^2\lambda^2}{1-\lambda^2}\sum_{\tau=s+1}^{t-1}(\frac{1+\lambda^2}{2})^{t-\tau-1}\|Y_\tau-\bar{Y}_\tau-(Y_\tau'-\bar{Y}_\tau')\|_F^2\\ &\quad+(\frac{1+\lambda^2}{2})^{t-s-1}\|X_{s+1}-\bar{X}_{s+1}-(X_{s+1}'-\bar{X}_{s+1}')\|_F^2\\ &=\frac{2\eta^2\lambda^2}{1-\lambda}\sum_{\tau=s+1}^{t-1}(\frac{1+\lambda^2}{2})^{t-\tau-1}\|Y_\tau-\bar{Y}_\tau-(Y_\tau'-\bar{Y}_\tau')\|_F^2+(\frac{1+\lambda^2}{2})^{t-s-1}\lambda^2(n-n_s)p_sr_0^2\end{aligned} \tag{D.51}$$

where we apply recursion in the second inequality and use the definition of the decoupled

sequences in the last equality. Similarly, by recursion we also have

$$\begin{aligned}
&\|Y_t - \bar{Y}_t - (Y_t' - \bar{Y}_t')\|_F^2 \\
&\leq \frac{1+\lambda^2}{2}\|Y_{t-1} - \bar{Y}_{t-1} - (Y_{t-1}' - \bar{Y}_{t-1}')\|_F^2 + \frac{\lambda^2+\lambda^4}{1-\lambda^2}\|V_t - V_{t-1} - (V_t' - V_{t-1}')\|_F^2 \\
&\leq \frac{2\lambda^2}{1-\lambda}\sum_{\tau=s+1}^{t}(\frac{1+\lambda^2}{2})^{t-\tau}\|V_\tau - V_{\tau-1} - (V_\tau' - V_{\tau-1}')\|_F^2
\end{aligned} \tag{D.52}$$

Combining above two inequalities, we achieve

$$\begin{aligned}
&\|X_t - \bar{X}_t - (X_t' - \bar{X}_t')\|_F^2 \\
&\leq \frac{2\eta^2\lambda^4}{(1-\lambda)^2}\sum_{\tau=s+1}^{t-1}(\frac{1+\lambda^2}{2})^{t-\tau-1}(t-\tau)\|V_\tau - V_{\tau-1} - (V_\tau' - V_{\tau-1}')\|_F^2 \\
&\quad + (\frac{1+\lambda^2}{2})^{t-s-1}\lambda^2(n-n_s)p_s r_0^2
\end{aligned} \tag{D.53}$$

According to the update rule of $v_t^{(i)}$ we have

$$\begin{aligned}
&v_t^{(i)} - v_{t-1}^{(i)} - (v_t^{(i)'} - v_{t-1}^{(i)'}) - (1-\beta)(v_{t-1}^{(i)} - v_{t-2}^{(i)} - (v_{t-1}^{(i)'} - v_{t-2}^{(i)'})) \\
&= \nabla F_i(x_t^{(i)}, \xi_t^{(i)}) - (1-\beta)\nabla F_i(x_{t-1}^{(i)}, \xi_t^{(i)}) - \nabla F_i(x_t^{(i)'}, \xi_t^{(i)}) + (1-\beta)\nabla F_i(x_{t-1}^{(i)'}, \xi_t^{(i)}) \\
&\quad - [\nabla F_i(x_{t-1}^{(i)}, \xi_{t-1}^{(i)}) - (1-\beta)\nabla F_i(x_{t-2}^{(i)}, \xi_{t-1}^{(i)}) - \nabla F_i(x_{t-1}^{(i)'}, \xi_{t-1}^{(i)}) \\
&\quad + (1-\beta)\nabla F_i(x_{t-2}^{(i)'}, \xi_{t-1}^{(i)})]
\end{aligned} \tag{D.54}$$

Then mimic the estimation of ν_t, we can obtain

$$\begin{aligned}
&\|V_t - V_{t-1} - (V_t' - V_{t-1}')\|_F^2 \\
&\leq \frac{32\log(4/\delta)}{b_1}\sum_{\tau=s}^{t-1}\sum_{i=1}^{n}[2L\|w_{\tau+1}^{(i)} - w_\tau^{(i)}\| + (2\beta L + 3\rho C_d)\|w_\tau^{(i)}\| + 3\rho C_d\|w_{\tau+1}^{(i)}\|]^2 \\
&\quad + 4L^2\sum_{i=1}^{n}\|w_t^{(i)} - w_{t-1}^{(i)}\|^2 + 36\rho^2 C_d^2\sum_{i=1}^{n}\|w_t^{(i)}\|^2
\end{aligned} \tag{D.55}$$

Combining above inequalities and the parameter setting of β, we can obtain

$$
\begin{aligned}
&\frac{1}{n}\|X_t-\bar{X}_t-(X'_t-\bar{X}'_t)\|_F^2-(\frac{1+\lambda^2}{2})^{t-s-1}\lambda^2 p_s r_0^2\\
\leq&(\frac{2000\log(4/\delta)\eta^2\rho^2C_d^2(t-s)\lambda^4}{(1-\lambda)^2b_1}+\frac{72\eta^2\rho^2C_d^2\lambda^4}{(1-\lambda)^2})\sum_{\tau=s+1}^{t-1}(\frac{1+\lambda^2}{2})^{t-\tau-1}\frac{(t-\tau)}{n}\sum_{i=1}^{n}\|w_\tau^{(i)}\|^2\\
&+(\frac{500\log(4/\delta)L^2\eta^2(t-s)\lambda^4}{(1-\lambda)^2b_1}+\frac{8L^2\eta^2\lambda^4}{(1-\lambda)^2})\sum_{\tau=s+1}^{t-1}(\frac{1+\lambda^2}{2})^{t-\tau-1}\frac{(t-\tau)}{n}\sum_{i=1}^{n}\|w_\tau^{(i)}-w_{\tau-1}^{(i)}\|^2\\
\leq&\frac{L^2\eta^2\lambda^4}{(1-\lambda)^2}(32+\frac{2000\log(4/\delta)(t-s)}{b_1})\sum_{\tau=s+1}^{t-1}(\frac{1+\lambda^2}{2})^{t-\tau-1}\frac{(t-\tau)}{n}\sum_{i=1}^{n}\|X_\tau-\bar{X}_\tau-(X'_\tau-\bar{X}'_\tau)\|_F^2\\
&+(\frac{2000\log(4/\delta)\eta^2\rho^2C_d^2(t-s)\lambda^4}{(1-\lambda)^2b_1}+\frac{72\eta^2\rho^2C_d^2\lambda^4}{(1-\lambda)^2})\sum_{\tau=s+1}^{t-1}(\frac{1+\lambda^2}{2})^{t-\tau-1}(t-\tau)\|w_\tau\|^2\\
&+(\frac{500\log(4/\delta)L^2\eta^2(t-s)\lambda^4}{(1-\lambda)^2b_1}+\frac{8L^2\eta^2\lambda^4}{(1-\lambda)^2})\sum_{\tau=s+1}^{t-1}(\frac{1+\lambda^2}{2})^{t-\tau-1}(t-\tau)\|w_\tau-w_{\tau-1}\|^2 \quad \text{(D.56)}
\end{aligned}
$$

Using Eq. (D.32), Eq. (D.43) and Eq. (D.50) we have

$$
\begin{aligned}
&\frac{L^2}{n}\|X_t-\bar{X}_t-(X'_t-\bar{X}'_t)\|_F^2\\
\leq&B_1t^2(\frac{1+\lambda^2}{2})^{t-s-1}L^2p_sr_0^2+B_2(1+\eta\gamma)^{2(t-s-1)}L^2p_s^2r_0^2\sum_{\tau=s+1}^{t-1}(\frac{(1+\lambda^2)(1+\eta\gamma)^2}{2})^{t-\tau-1}(t-\tau)\\
&+B_3(1+\eta\gamma)^{2(t-s-1)}L^2p_s^2r_0^2\sum_{\tau=s+1}^{t-1}(\frac{(1+\lambda^2)(1+\eta\gamma)^2}{2})^{t-\tau-1}\frac{t-\tau}{(\tau-s)^2}\\
&+(\frac{1+\lambda^2}{2})^{t-s-1}L^2p_sr_0^2 \quad \text{(D.57)}
\end{aligned}
$$

where

$$
\begin{aligned}
B_1=&\frac{4L^2\eta^2}{(1-\lambda)^2}(32+\frac{2000\log(4/\delta)(t-s)}{b_1})+384L^2\eta^2(\frac{500\log(4/\delta)L^2\eta^2(t-s)}{(1-\lambda)^2b_1}+\frac{8L^2\eta^2}{(1-\lambda)^2})\\
B_2=&\frac{1}{72(1-\lambda)^2C_T^2}(2+\frac{125\log(4/\delta)(t-s)}{b_1})+\frac{4500\log(4/\delta)\eta^2\rho^2C_d^2(t-s)}{(1-\lambda)^2b_1}+\frac{162\eta^2\rho^2C_d^2}{(1-\lambda)^2}\\
&+(8L^2\eta^2\gamma^2+54L^2\eta^2\rho^2C_d^2+\frac{1}{6C_T^2})(\frac{500\log(4/\delta)L^2\eta^2(t-s)}{(1-\lambda)^2b_1}+\frac{8L^2\eta^2}{(1-\lambda)^2})\\
B_3=&\frac{48\log(4/\delta)L^2\eta^2}{b_1}(\frac{500\log(4/\delta)L^2\eta^2(t-s)}{(1-\lambda)^2b_1}+\frac{8L^2\eta^2}{(1-\lambda)^2})
\end{aligned}
$$

$$+\frac{L^4\eta^4(1+\eta\gamma)^{2(t-s-1)}}{(1-\lambda)^2 b_1}(32+\frac{2000\log(4/\delta)(t-s)}{b_1}) \tag{D.58}$$

If $t \geq \tilde{O}(\frac{1}{1-\lambda})$, $t^2(\frac{1+\lambda^2}{2})^{t-s-1}$ is small and the first term of Eq. (D.57) can be merged to the second term. Otherwise if $t < \tilde{O}(\frac{1}{1-\lambda})$, it can be merged to the last term according to the parameter setting of η and b_1. When ε is small, we have $\frac{(1+\lambda^2)(1+\eta\gamma)^2}{2} \leq \frac{3+\lambda^2}{4}$. Hence the second term of Eq. (D.57) can be bounded by Lemma D.8. The third term of Eq. (D.57) can be estimated by Lemma D.9 (the case of $t < \tilde{O}(\frac{1}{1-\lambda})$ can be addressed by the parameter setting of η and b_1). Therefore, we can prove

$$\begin{aligned}\frac{L^2}{n}\|X_t - \bar{X}_t - (X'_t - \bar{X}'_t)\|_F^2 \leq\ & 4(\frac{1+\lambda^2}{2})^{t-s-1}L^2 p_s r_0^2 + \frac{1}{2304\eta C_T}(1+\eta\gamma)^{2(t-s-1)}p_s^2 r_0^2 \\ & + \frac{L^2\eta^2(1+\eta\gamma)^{2(t-s-1)}L^2 p_s^2 r_0^2}{16 b_1 (t-s)^2}\end{aligned} \tag{D.59}$$

because of the parameter setting. We should notice that here we also use the relation $\frac{\eta}{b_1\gamma} \leq O(1)$, which is always satisfied according to the setting of b_1. Based on Eq. (D.49) and Eq. (D.59), it is easy to check that ζ_t satisfies Eq. (D.36). Moreover, we have

$$\begin{aligned}&\eta\sum_{\tau=s+1}^{t-1}(1+\eta\gamma)^{t-\tau-1}\|\zeta_\tau\| \\ \leq\ & L\eta(1+\eta\gamma)^{t-s-1}p_s r_0(8\sum_{\tau=s+1}^{t-1}(\frac{4+\lambda^2}{5})^{t-s-1} + \sum_{\tau=s+1}^{t-1}\frac{1}{\tau-s}) + \frac{1}{12}(1+\eta\gamma)^{t-s-1}p_s r_0 \\ \leq\ & L\eta(\frac{80}{1-\lambda} + \log(C_T))(1+\eta\gamma)^{t-s-1}p_s r_0 + \frac{1}{12}(1+\eta\gamma)^{t-s-1}p_s r_0 \\ \leq\ & \frac{1}{6}(1+\eta\gamma)^{t-s-1}p_s r_0\end{aligned} \tag{D.60}$$

where the first inequality is derived by

$$\frac{1+\lambda^2}{2} \leq (\frac{3+\lambda^2}{4})^2 \quad \text{and} \quad \frac{(3+\lambda^2)(1+\eta\gamma)}{4} \leq \frac{4+\lambda^2}{5} \tag{D.61}$$

Now combining Eq. (D.37), Eq. (D.47) and Eq. (D.60), we can reach the conclusion in Eq. (D.34) and finish the proof of the induction. Recall the assumption at the beginning, we have

$$\frac{1}{2}(1+\eta\gamma)^{C_T} p_s r_0 \leq w_{C_T} \leq 2\bar{d} \leq 6C_d \tag{D.62}$$

since $\|\bar{x}_t - \bar{x}'_t\| \leq \|\bar{x}_t - \bar{x}_s\| + \|\bar{x}'_t - \bar{x}_s\|$. Eq. (D.62) implies that

$$C_T \leq \frac{\log(12C_d/(p_s r_0))}{\log(1+\eta\gamma)} < \frac{2\log(12nC_d/r_0)}{\eta\gamma} \tag{D.63}$$

which conflicts with the definition of C_T. Therefore, the proof of Lemma D.6 is finished. □

D.3.7 Proof of Lemma D.7

Proof. If node i enters the escaping phase in iteration s' before iteration s and does not break it in iteration $s+C_T$, then for $s \leq t \leq s+C_T$, we have $\|x_t^{(i)} - x_s^{(i)}\| \leq \|x_t^{(i)} - x_{s'}^{(i)}\| + \|x_s^{(i)} - x_{s'}^{(i)}\| \leq 2C_d$. Therefore, there are at least $\frac{n}{10}$ worker nodes satisfying $\max_{s\leq t\leq s+C_T} \|x_t^{(i)} - x_s^{(i)}\| \leq 2C_d$.

Suppose $\min eig(\nabla^2 f(\bar{x}_s)) \leq -\varepsilon_H$ and $\mathbf{e_1}$ is the corresponding eigenvector. Let $\mathscr{S}_i$ denote the region of the perturbation on node i that PEDESTAL will terminate in iteration $s+C_T$, *i.e.,* $\frac{n}{10}$ workers will not break the escaping phase. Then by Lemma D.6 we can conclude that there must exist one worker node such that the projection of $\mathscr{S}_i$ onto direction $\mathbf{e_1}$ is smaller than r_0. Since the perturbation ξ_i is drawn from uniform distribution, the probability of $\xi_i \in \mathscr{S}_i$ can be bounded by

$$Pr(\xi_i \in \mathscr{S}_i) \leq \frac{r_0 V(Ball(d-1,r))}{V(Ball(d,r))} \leq \delta \tag{D.64}$$

where $V(\cdot)$ denotes the volume and $Ball(d,r)$ denotes the d-dimensional ball with radius r.

The last inequality is achieved by the definition of r_0. Therefore, we can prove that $\bar{x}_s$ is a second-order stationary point with probability at least $1-\delta$. □

D.4 Additional Theoretical Result

In this section we will provide some additional theoretical result of our PEDESTAL algorithm. First we will demonstrate the convergence analysis of the case $\varepsilon_H < \sqrt{\varepsilon}$, *i.e.,* $\alpha > 0.5$. Next, we will discuss the strategy of using fixed number of iterations in each descent and escaping phase, which motivates the design of PEDESTAL.

D.4.1 Smaller Tolerance for Second-Order Optimality

When $\varepsilon_H < \sqrt{\varepsilon}$, the conclusions of previous Lemmas are still satisfied except Lemma D.4. In this case, $C_d = C_2 \eta C_T \varepsilon^{\mu}$ where $\mu = 2\alpha > 1$. Parameter C_d should be smaller than the original setting in Lemma D.4, which results in more iterations to converge. Fortunately, the analysis of Lemma D.4 can be adjusted and we can achieve Theorem 5.2. The proof is provided as follows.

Proof. The fourth term of $\mathscr{D}_t$ in Lemma D.3 is derived by $\frac{\eta\beta\sigma^2}{b_1}$ and at this time we will set $b_1 \geq \varepsilon^{-(2\mu-1-\theta)}$ so that the ε term is replaced by ε^{μ}. The last term of $\mathscr{D}_t$ can be written as

$$\sum_{t=0}^{T-1} \frac{7p_t(\eta^2 C_v^2 + r^2)}{4\eta} = \frac{1}{n} \sum_{(t,i)\in\mathscr{P}} \frac{7(\eta^2 C_v^2 + r^2)}{4\eta} \tag{D.65}$$

where $\mathscr{P}$ is the set of all pairs of (t,i) such that node i draws perturbation in iteration t. We can divide $\mathscr{P}$ into two parts. $\mathscr{P}_1$ contains all pairs of (t,i) such that node i breaks the escaping phase within M iterations, where M is an integer to be decided later. The rest part is denoted by $\mathscr{P}_2$.

For any $(t,i) \in \mathscr{P}_1$, suppose node i breaks escaping phase in iteration $t+m$, where $m \leq M$. Then node i will never draw perturbation between iteration t and iteration $t+M$. Mimic the steps of Eq. (D.22), by Cauchy-Schwartz inequality we can obtain

$$\sum_{\tau=t}^{t+m} \|x_{\tau+1}^{(i)} - x_{\tau}^{(i)}\|^2 \geq \frac{C_d^2}{M} \tag{D.66}$$

Let $M = \varepsilon^{-2-2\theta+2\alpha}$. Then we have

$$\frac{(1-\lambda)^2}{512\eta} \sum_{\tau=t}^{t+m} \|x_{\tau+1}^{(i)} - x_{\tau}^{(i)}\|^2 \geq \frac{7(\eta^2 C_v^2 + r^2)}{4\eta} \tag{D.67}$$

and

$$\frac{(1-\lambda)^2}{512n\eta} \sum_{\tau=t}^{t+m} \sum_{i=1}^{n} \|x_{\tau+1}^{(i)} - x_{\tau}^{(i)}\|^2 \geq \frac{1}{n} \sum_{(t,i)\in\mathscr{P}_1} \frac{7(\eta^2 C_v^2 + r^2)}{4\eta} \tag{D.68}$$

by the parameter setting of C_v. On the other hand, if $(t,i) \in \mathscr{P}_2$, then node i will not break the escaping phase in M steps and hence the perturbation step will not execute, either. Therefore, we have estimation

$$\frac{1}{n} \sum_{(t,i)\in\mathscr{P}_2} \frac{7(\eta^2 C_v^2 + r^2)}{4\eta} \leq \sum_{t=0}^{T-1} \frac{7(\eta^2 C_v^2 + r^2)}{4M\eta} \tag{D.69}$$

With Eq. (D.68), Eq. (D.69) and the new setting of b_1, the descent in Lemma D.3 can be improved to

$$\mathscr{D}_t = \frac{1}{16\eta}\omega_t + \frac{(1-\lambda)^2}{512n\eta} \sum_{i=1}^{n} \|x_{t+1}^{(i)} - x_t^{(i)}\|^2 + \frac{\eta}{2n} \sum_{i=1}^{n} \|y_t^{(i)}\|^2 - \frac{200\eta\varepsilon^{2\mu}\sigma^2}{(1-\lambda)^2 C_1^2} - \frac{7(\eta^2 C_v^2 + r^2)}{4M\eta}$$

When $\theta \geq 3\alpha - 2$, we have $\frac{\varepsilon^2}{M} \leq \varepsilon^{2\mu}$ and Lemma D.4 still holds but the conclusion is

changed to

$$\sum_{t\in\mathscr{I}} \mathscr{D}_t \geq |\mathscr{I}|\cdot\frac{(1-\lambda)^2 C_2^2 \eta \varepsilon^{2\mu}}{10000}$$

In this case, PEDESTAL algorithm will terminate in $\tilde{O}(\varepsilon^{-\theta-2\mu})$ iterations. In Lemma D.6 and Lemma D.7 we need the relations

$$\frac{\eta^2 C_d^2 C_T^3}{b_1} \leq O(1), \quad \frac{\eta}{b_1 \varepsilon_H} \leq O(1) \tag{D.70}$$

which implies $b_1 \geq \tilde{O}(\varepsilon^{-\theta-\alpha})$. Therefore, we set $b_1 = \tilde{\Theta}(\varepsilon^{-\max\{4\alpha-1-\theta,\theta+\alpha\}})$ with the condition $\theta \geq 3\alpha-2$. When $\alpha \leq 1$, we set $\theta = \frac{3\alpha-1}{2}$, which satisfies $\theta \geq 3\alpha-2$ and

$$4\alpha-1-\theta = \theta+\alpha = \frac{5\alpha-1}{2} \tag{D.71}$$

The gradient complexity in this case is

$$\tilde{O}(\varepsilon^{-\frac{11\alpha-1}{2}}\cdot\varepsilon^{-\frac{5\alpha-1}{2}}) = \tilde{O}(\varepsilon^{-8\alpha+1}) \tag{D.72}$$

When $\alpha > 1$, we have $\theta = 3\alpha-2$ and $b_1 = \tilde{\Theta}(\varepsilon^{-(4\alpha-2)})$. The gradient complexity is

$$\tilde{O}(\varepsilon^{-(7\alpha-2)}\cdot\varepsilon^{-(4\alpha-2)}) = \tilde{O}(\varepsilon^{-11\alpha+4}) \tag{D.73}$$

which finishes the proof of Theorem 5.2. □

Therefore, the gradient complexity over all cases of α can by written by

$$\tilde{O}(\varepsilon^{-3} + \varepsilon\varepsilon_H^{-8} + \varepsilon^4\varepsilon_H^{-11}) \tag{D.74}$$

D.4.2 Phases with Fixed Number of Iterations

If a decentralized stochastic perturbed gradient descent method adopt the strategy of fixed number of iterations in each phase, the gradient complexity in the descent phase should be at least $O(\varepsilon^{-3})$ to ensure the first-order stationary point. But the total descent of a descent phase could be small because it is possible that it is stuck at a saddle point after only a few steps. Hence we need to consider the descent in the escaping phase. According to Lemma D.3 and Lemma D.4 we can see the descent of an escaping phase is $O(\frac{C_d^2}{\eta C_T})$. As the conditions $\eta C_d C_T \leq O(1)$ and $C_T = \tilde{O}(\frac{1}{\eta \varepsilon_H})$ are required in Lemma D.6, we can obtain that the total descent of an escaping phase is no larger than $\tilde{O}(\varepsilon_H^3)$. In the classic setting of $\varepsilon_H = \sqrt{\varepsilon}$, the total descent of an escaping is upper bounded by $\tilde{O}(\varepsilon^{1.5})$. Consequently, the total gradient complexity to achieve $(\varepsilon, \sqrt{\varepsilon})$-second-order stationary point is at least $\tilde{O}(\varepsilon^{-4.5})$, which is worse than the result of our PEDESTAL.

D.5 Auxiliary Lemmas

Lemma D.8. *Let* $0 < a < 1$. *Then we have*

$$\sum_{\tau=1}^{t} \tau a^{\tau-1} = \frac{1-a^t}{(1-a)^2} - \frac{t a^t}{1-a}$$

Lemma D.9. *Let* $0 < a < 1$. *When* $t \geq \tilde{O}(\frac{1}{1-a})$, *we have*

$$\sum_{\tau=1}^{t} \frac{\tau a^{\tau-1}}{(t+1-\tau)^2} \leq \frac{8}{t^2(1-a)^2}$$

Proof. When $\tau \leq \frac{t}{2}$, by Lemma D.8 we have

$$\sum_{\tau \leq t/2} \frac{\tau a^{\tau-1}}{(t+1-\tau)^2} \leq \frac{4}{t^2(1-a)^2} \tag{D.75}$$

When $\tau > \frac{t}{2}$, we have

$$\sum_{\tau>t/2} \frac{\tau a^{\tau-1}}{(t+1-\tau)^2} \leq \sum_{\tau>t/2} \tau a^{\tau-1} \leq a^{t/2}(\frac{t}{2(1-a)} + \frac{1}{(1-a)^2}) \tag{D.76}$$

Therefore, we can reach the conclusion when $t \geq \tilde{O}(\frac{1}{1-a})$. □

Lemma D.10. *(Definition of Variance) For any random variable X, we have*

$$\mathbb{E}[X - \mathbb{E}X]^2 = \mathbb{E}X^2 - (\mathbb{E}X)^2$$

Lemma D.11. *(Lemma D.1 in ([15])) Let $\varepsilon_{1:k} \in \mathbb{R}^d$ be a vector-valued martingale difference sequence with respect to $\mathscr{F}_k$, i.e., for each $k \in [K]$, $\mathbb{E}[\varepsilon_k|\mathscr{F}_k] = 0$ and $\|\varepsilon_k\| \leq B_k$, then with probability $1-\delta$ we have*

$$\|\sum_{k=1}^{K} \varepsilon_k\|^2 \leq 4\log(4/\delta) \sum_{k=1}^{K} B_k^2 \tag{D.77}$$

Appendix E: Appendix of Chapter 6

E.1 Convergence Analysis of Generalized GDA

First, we will provide the proof for the following essential Lemmas.

Lemma E.1. *For any x and x' that satisfy $\|x'-x\| \leq \frac{G}{l_x(\|\nabla_x f(x,y^*(x))\|+G)}$ for some $G \geq 0$, we have*

$$\|y^*(x)-y^*(x')\| \leq \kappa \|x-x'\| \tag{E.1}$$

Proof. Since $y^*(\cdot)$ is the maximizer, for $\forall y \in \mathscr{Y}$ we have

$$\langle y-y^*(x), \nabla_y f(x,y^*(x))\rangle \leq 0,\ \langle y-y^*(x'), \nabla_y f(x',y^*(x'))\rangle \leq 0 \tag{E.2}$$

Sum these two inequalities together and we can obtain

$$\langle y^*(x)-y^*(x'), \nabla_y f(x',y^*(x')) - \nabla_y f(x,y^*(x))\rangle \leq 0 \tag{E.3}$$

As function f is strongly concave with respect to y, we have

$$\mu \|y^*(x)-y^*(x')\|^2 \leq \langle y^*(x)-y^*(x'), \nabla_y f(x',y^*(x')) - \nabla_y f(x',y^*(x))\rangle \tag{E.4}$$

Combine above two inequalities and we achieve

$$\mu\|y^*(x)-y^*(x')\|^2 \leq \langle y^*(x)-y^*(x'), \nabla_y f(x,y^*(x)) - \nabla_y f(x',y^*(x))\rangle \tag{E.5}$$

When $\|x'-x\| \leq \frac{G}{l_x(\|\nabla_x f(x,y^*(x))\|+G)}$ for some $G>0$, by Assumption 6.3 we have

$$\|\nabla_y f(x,y^*(x)) - \nabla_y f(x',y^*(x))\| \leq l_y(\|\nabla_y f(x,y^*(x))\|)\cdot\|x-x'\| \tag{E.6}$$

When $\mathscr{Y} = \mathbb{R}^{d_2}$, we have $\|\nabla_y f(x,y^*(x))\| = 0$. As function $l_y(\cdot)$ is non-decreasing, we have $l_y(0) \leq l_y(4G_y)$. When f is Lipschitz smooth with respect to y, function $l_y(\cdot)$ is constant L_y and we still have $l_y(2G_y) = L_y$. Combine Eq. (E.5) and (E.6), we can reach the conclusion in Lemma E.1. □

Lemma E.2. *For any x and x' that satisfy $\|x'-x\| \leq \frac{G}{l_x(\|\nabla\Phi(x)\|+G)}$ for some $G \geq 0$, we have*

$$\|\nabla\Phi(x) - \nabla\Phi(x')\| \leq 2\kappa l_x(\|\nabla\Phi(x)\|+G)\cdot\|x-x'\| \tag{E.7}$$

$$\Phi(x') \leq \Phi(x) + \langle\nabla\Phi(x), x'-x\rangle + \kappa l_x(\|\nabla\Phi(x)\|+G)\cdot\|x-x'\|^2 \tag{E.8}$$

$$\Phi(x') \geq \Phi(x) + \langle\nabla\Phi(x), x'-x\rangle - \kappa l_x(\|\nabla\Phi(x)\|+G)\cdot\|x-x'\|^2 \tag{E.9}$$

Proof. By Lemma E.1 and Assumption 6.3 we have

$$\begin{aligned}\|\nabla\Phi(x') - \nabla\Phi(x)\| &= \|\nabla_x f(x',y^*(x')) - \nabla_x f(x,y^*(x))\| \\ &\leq l_x(\|\nabla\Phi(x)\|+G)\cdot(\|x'-x\| + \|y^*(x')-y^*(x)\|) \\ &\leq 2\kappa l_x(\|\nabla\Phi(x)\|+G)\cdot\|x'-x\| \end{aligned} \tag{E.10}$$

Hence for any $z(t) = x + t(x' - x)$, we have

$$\|\nabla\Phi(z(t)) - \nabla\Phi(x)\| \leq 2t\kappa l_x(\|\nabla\Phi(x)\| + G) \cdot \|x' - x\| \tag{E.11}$$

Since we have the equation

$$\Phi(x') = \Phi(x) + \langle\nabla\Phi(x), x' - x\rangle + \int_0^1 \langle\nabla\Phi(z(t)) - \Phi(x), x' - x\rangle dt \tag{E.12}$$

we can obtain

$$\begin{aligned}\|\Phi(x') - \Phi(x) - \langle\nabla\Phi(x), x' - x\rangle\| &\leq 2\kappa l_x(\|\nabla\Phi(x)\| + G) \cdot \|x' - x\|^2 \cdot \int_0^1 t dt \\ &\leq \kappa l_x(\|\nabla\Phi(x)\| + G) \cdot \|x - x'\|^2 \end{aligned} \tag{E.13}$$

which leads to the last two inequalities in Lemma E.2. □

Lemma E.3. *For any x, we have*

$$\|\nabla\Phi(x)\|^2 \leq 4\kappa l_x(2\|\nabla\Phi(x)\|) \cdot (\Phi(x) - \Phi^*) \tag{E.14}$$

Proof. Let $x' = x - \frac{\nabla\Phi(x)}{2\kappa l_x(2\|\nabla\Phi(x)\|)}$. By Lemma E.2 we have

$$\Phi^* \leq \Phi(x') \leq \Phi(x) - \frac{\|\nabla\Phi(x)\|^2}{4\kappa l_x(2\|\nabla\Phi(x)\|)} \tag{E.15}$$

which implies the conclusion of Lemma E.3. □

Lemma E.4. *For $\forall x$ and y, we have*

$$\|\nabla_y f(x,y)\|^2 \leq 2 \cdot l_y(2\|\nabla_y f(x,y)\|) \cdot (f(x, y^*(x)) - f(x,y)) \tag{E.16}$$

Proof. By Assumption 6.3 and the definition of maximizer $y^*(\cdot)$ we have

$$\begin{aligned} f(x,y^*(x)) &\geq f(x,y+\frac{\nabla_y f(x,y)}{l_y(2\|\nabla_y f(x,y)\|)}) \\ &\geq f(x,y)+\frac{1}{l_y(2\|\nabla_y f(x,y)\|)}\|\nabla_y f(x,y)\|^2-\frac{1}{2\cdot l_y(2\|\nabla_y f(x,y)\|)}\|\nabla_y f(x,y)\|^2 \\ &= f(x,y)+\frac{1}{2\cdot l_y(2\|\nabla_y f(x,y)\|)}\|\nabla_y f(x,y)\|^2 \end{aligned} \tag{E.17}$$

which implies the conclusion in Lemma E.4. □

Lemma E.5. *Let* $\eta_t=\frac{\eta}{S_t}\leq\frac{C_0}{16\kappa^2 l_x(2G_x)}$, $\eta_y=\frac{1}{l_y(2G_y)}$ *and* $\|y_0-y_0^*\|\leq\frac{C_0G_x}{l_x(2G_x)}$. *When* $\mathscr{Y}=\mathbb{R}^{d_2}$, *we have* $\|\nabla\Phi(x_t)\|\leq G_x$, $\|\nabla_x f(x_t,y_t)\|\leq 2G_x$, $\|\nabla_y f(x_t,y_t)\|\leq G_y$ *and* $\|y_t-y_t^*\|\leq\frac{C_0G_x}{l_x(2G_x)}$ *for all* $t\geq 0$, *where constant* $C_0=\min\{1,\frac{l_x(2G_x)G_y}{l_y(2G_y)G_x}\}$.

Proof. We apply mathematical induction to prove the conclusions in Lemma E.5. According to Lemma E.3 and the definition of G_x, we have $\|\nabla\Phi(x_0)\|\leq G_x$. As $\|y_0-y_0^*\|\leq\frac{G_x}{l_x(2G_x)}\leq\frac{G_x}{l_x(\|\nabla\Phi(x_0)\|+G_x)}$, by Assumption 6.3 we can further obtain

$$\|\nabla_x f(x_0,y_0)-\nabla\Phi(x_0)\|\leq l_x(2G_x)\cdot\|y_0-y_0^*\|\leq G_x \tag{E.18}$$

which implies $\|\nabla_x f(x_0,y_0)\|\leq 2G_x$. Hence the conditions of case $t=0$ are satisfied.

Assume that the conclusions are satisfied for case $t\leq\tau$. When $t=\tau+1$, by the requirement of η_t we have

$$\eta_\tau\|\nabla_x f(x_\tau,y_\tau)\|\leq 2\eta_\tau G_x\leq\frac{G_x}{l_x(2G_x)}\leq\frac{G_x}{l_x(\|\nabla\Phi(x_\tau)\|+G_x)} \tag{E.19}$$

where we have used the induction assumption $\|\nabla_x f(x_\tau,y_\tau)\|\leq 2G_x$ and $\|\nabla\Phi(x_\tau)\|\leq G_x$.

Then we can apply Lemma E.1 and Lemma E.2 to achieve

$$\|y^*_{\tau+1} - y^*_\tau\| \leq \kappa \|x_{\tau+1} - x_\tau\| \tag{E.20}$$

and

$$\begin{aligned}
\Phi(x_{\tau+1}) &\leq \Phi(x_\tau) + \langle \nabla\Phi(x_\tau), x_{\tau+1} - x_\tau \rangle + \kappa l_x(2G_x) \cdot \|x_{\tau+1} - x_\tau\|^2 \\
&= \Phi(x_\tau) - \eta_\tau \langle \nabla\Phi(x_\tau), \nabla_x f(x_\tau, y_\tau) \rangle + \kappa l_x(2G_x) \cdot \eta_\tau^2 \|\nabla_x f(x_\tau, y_\tau)\|^2 \\
&= \Phi(x_\tau) - \frac{\eta_\tau}{2}\|\nabla\Phi(x_\tau)\|^2 + \frac{\eta_\tau}{2}\|\nabla_x f(x_\tau, y_\tau) - \nabla\Phi(x_\tau)\|^2 - \frac{\eta_\tau}{2}(1 - 2\kappa l_x(2G_x) \cdot \eta_\tau)\|\nabla_x f(x_\tau, y_\tau)\|^2 \\
&\leq \Phi(x_\tau) - \frac{\eta_\tau}{2}\|\nabla\Phi(x_\tau)\|^2 + \frac{\eta_\tau}{2}\|\nabla_x f(x_\tau, y_\tau) - \nabla\Phi(x_\tau)\|^2
\end{aligned} \tag{E.21}$$

where the last inequality is obtained by the condition of η_τ. Next we will prove $\|y_{\tau+1} - y^*_{\tau+1}\| \leq \frac{G_x}{l_x(2G_x)}$. According to the update rule of y and the non-expansion property of projection, we have

$$\begin{aligned}
\|y^*_\tau - y_{\tau+1}\|^2 &= \|y^*_\tau - \Pi_{\mathscr{Y}}(y_\tau + \eta_y \nabla_y f(x_\tau, y_\tau))\|^2 \\
&\leq \|y^*_\tau - y_\tau - \eta_y \nabla_y f(x_\tau, y_\tau)\|^2 \\
&= \|y^*_\tau - y_\tau\|^2 - 2\eta_y \langle \nabla_y f(x_\tau, y_\tau), y^*_\tau - y_\tau \rangle + \eta_y^2 \|\nabla_y f(x_\tau, y_\tau)\|^2
\end{aligned} \tag{E.22}$$

As function f is strongly-concave with respect to y, we have

$$\langle \nabla_y f(x_\tau, y_\tau), y^*_\tau - y_\tau \rangle \geq \frac{\mu}{2}\|y^*_\tau - y_\tau\|^2 + f(x_\tau, y^*_\tau) - f(x_\tau, y_\tau) \tag{E.23}$$

Combine Eq. (E.22), (E.23) and Lemma E.4, we have

$$\begin{aligned}\|y_\tau^* - y_{\tau+1}\|^2 &\le (1-\mu\eta_y)\|y_\tau^* - y_\tau\|^2 - 2\eta_y(1-\eta_y \cdot l_y(2G_y))(f(x_\tau, y_\tau^*) - f(x_\tau, y_\tau)) \\ &\le (1-\mu\eta_y)\|y_\tau^* - y_\tau\|^2 \le (1-\frac{1}{\kappa})\|y_\tau^* - y_\tau\|^2 \end{aligned} \tag{E.24}$$

where we have used the induction assumption $\|\nabla_y f(x_\tau, y_\tau)\| \le G_y$ and $\eta_y = \frac{1}{l_y(2G_y)}$. Combine Eq. (E.20), (E.24), the induction assumption $\|y_\tau - y_\tau^*\| \le \frac{G_x}{l_x(2G_x)}$ and $\|\nabla_x f(x_\tau, y_\tau)\| \le 2G_x$, we have

$$\begin{aligned}\|y_{\tau+1}^* - y_{\tau+1}\| &\le \|y_\tau^* - y_{\tau+1}\| + \|y_{\tau+1}^* - y_\tau^*\| \\ &\le (1-\frac{1}{2\kappa})\|y_\tau^* - y_\tau\| + \kappa\eta_\tau\|\nabla_x f(x_\tau, y_\tau)\| \le (1-\frac{1}{2\kappa})\frac{G_x}{l_x(2G_x)} + 2\kappa\eta_\tau G_x \le \frac{G_x}{l_x(2G_x)} \end{aligned} \tag{E.25}$$

where we have used the requirement of η_τ in the last inequality. As $\|y_\tau - y_\tau^*\| \le \frac{G_x}{l_x(2G_x)} \le \frac{G_x}{l_x(\|\nabla\Phi(x_\tau)\| + G_x)}$, by Assumption 6.3 we can obtain

$$\|\nabla_x f(x_\tau, y_\tau) - \nabla\Phi(x_\tau)\|^2 \le l_x^2(2G_x) \cdot \|y_\tau - y_\tau^*\|^2 \tag{E.26}$$

By Young's inequality we have

$$\begin{aligned}\|y_\tau - y_\tau^*\|^2 &\le (1+\frac{1}{2\kappa-1})\|y_{\tau-1}^* - y_\tau\|^2 + 2\kappa\|y_\tau^* - y_{\tau-1}^*\|^2 \\ &\le \frac{2\kappa}{2\kappa-1} \cdot \frac{\kappa-1}{\kappa}\|y_{\tau-1} - y_{\tau-1}^*\|^2 + 2\kappa^3\eta_{\tau-1}^2\|\nabla_x f(x_{\tau-1}, y_{\tau-1})\|^2 \\ &\le (1-\frac{1}{2\kappa} + 4\kappa^3\eta_{\tau-1}^2 l_x^2(2G_x))\|y_{\tau-1} - y_{\tau-1}^*\|^2 + 4\kappa^3\eta_{\tau-1}^2\|\nabla\Phi(x_{\tau-1})\|^2 \\ &\le (1-\frac{1}{4\kappa})\|y_{\tau-1} - y_{\tau-1}^*\|^2 + 4\kappa^3\eta_{\tau-1}^2\|\nabla\Phi(x_{\tau-1})\|^2 \end{aligned} \tag{E.27}$$

where the second inequality is derived by the same way as Eq. (E.24) and (E.20); the third

inequality is derived by Cauchy-Schwartz inequality and Assumption 6.3; the last inequality is derived by the condition of η_t. Let $\gamma = 1 - \frac{1}{4\kappa}$. Applying recursion to Eq. (E.27), we can obtain

$$\|y_\tau - y_\tau^*\|^2 \leq \gamma^\tau \|y_0 - y_0^*\|^2 + 4\kappa^3 \sum_{s=0}^{\tau-1} \gamma^{\tau-1-s} \eta_s^2 \|\nabla\Phi(x_s)\|^2 \tag{E.28}$$

Inserting Eq. (E.26) and (E.28) into (E.21), we have

$$\Phi(x_{\tau+1}) \leq \Phi(x_\tau) - \frac{\eta_\tau}{2}\|\nabla\Phi(x_\tau)\|^2 + \frac{\eta_\tau l_x^2(2G_x)}{2}\left(\gamma^\tau \|y_0 - y_0^*\|^2 + 4\kappa^3 \sum_{s=0}^{\tau-1} \gamma^{\tau-1-s}\eta_s^2 \|\nabla\Phi(x_s)\|^2\right) \tag{E.29}$$

Applying recursion to above inequality, we can achieve

$$\begin{aligned}
\Phi(x_{\tau+1}) &\leq \Phi(x_0) - \sum_{t=0}^{\tau} \frac{\eta_t}{2}\left(1 - 4\kappa^3 \eta_t^2 l_x^2(2G_x) \sum_{s=t}^{\tau-1} \gamma^{s-t}\right)\|\nabla\Phi(x_t)\|^2 + \frac{l_x^2(2G_x)\cdot\|y_0 - y_0^*\|^2}{2}\sum_{t=0}^{\tau}\gamma^t\eta_t \\
&\leq \Phi(x_0) - \sum_{t=0}^{\tau} \frac{\eta_t}{2}\left(1 - 16\kappa^4 \eta_t^2 l_x^2(2G_x)\right)\|\nabla\Phi(x_t)\|^2 + \frac{G_x^2}{32\kappa^2 l_x(2G_x)}\sum_{t=0}^{\tau}\gamma^t \\
&\leq \Phi(x_0) - \sum_{t=0}^{\tau} \frac{15\eta_t}{32}\|\nabla\Phi(x_t)\|^2 + \frac{G_x^2}{8\kappa l_x(2G_x)}
\end{aligned} \tag{E.30}$$

where we have used the setup of η_t and $\|y_0 - y_0^*\|$. According to the definition of G_x, we have

$$\Phi(x_{\tau+1}) - \Phi^* \leq (\Phi(x_0) - \Phi^*) - \sum_{t=0}^{\tau} \frac{15\eta_t}{32}\|\nabla\Phi(x_t)\|^2 + (\Phi(x_0) - \Phi^*) \leq 2(\Phi(x_0) - \Phi^*) \tag{E.31}$$

Combining Eq. (E.31), Lemma E.3 and the definition of G_x, we can reach the conclusion that $\|\nabla\Phi(x_{\tau+1})\| \leq G_x$. As $\|y_{\tau+1} - y_{\tau+1}^*\| \leq \frac{G_x}{l_x(2G_x)} \leq \frac{G_x}{l_x(\|\nabla\Phi(x_{\tau+1})\| + G_x)}$, by Assumption 6.3

we can obtain

$$\|\nabla_x f(x_{\tau+1}, y_{\tau+1}) - \nabla\Phi(x_{\tau+1})\| \leq l_x(2G_x) \cdot \|y_{\tau+1} - y^*_{\tau+1}\| \leq G_x \tag{E.32}$$

which implies $\|\nabla_x f(x_{\tau+1}, y_{\tau+1})\| \leq 2G_x$. Finally, we need to estimate $\|\nabla_y f(x_{\tau+1}, y_{\tau+1})\|$. We have

$$\begin{aligned}\|\nabla_y f(x_{\tau+1}, y_{\tau+1})\| &= \|\nabla_y f(x_{\tau+1}, y_{\tau+1}) - \nabla_y f(x_{\tau+1}, y^*_{\tau+1})\| \\ &\leq l_y(0) \cdot \|y_{\tau+1} - y^*_{\tau+1}\| \leq \frac{C_0 l_y(0) G_x}{l_x(2G_x)} \leq G_y\end{aligned} \tag{E.33}$$

which is obtained by the definition of constant C_0. □

Lemma E.6. *Let* $\eta_t = \frac{\eta}{S_t} \leq \frac{C_0}{16\kappa^2 l_x(2G_x)}$, $\eta_y = \frac{1}{l_y(2G_y)}$ *and* $\|y_0 - y^*_0\| \leq \frac{C_0 G_x}{l_x(2G_x)}$. *When* $l_y(\cdot) \equiv L_y$, *we have* $\|\nabla\Phi(x_t)\| \leq G_x$, $\|\nabla_x f(x_t, y_t)\| \leq 2G_x$ *and* $\|y_t - y^*_t\| \leq \frac{C_0 G_x}{l_x(2G_x)}$ *for all* $t \geq 0$, *where constant* $C_0 = \min\{1, \frac{l_x(2G_x)G_y}{l_y(0)G_x}\}$.

Proof. Different from the case $\mathscr{Y} = \mathbb{R}^{d_2}$, we do not need the upper bound for $\nabla_y f(x_t, y_t)$. In Lemma E.5 the only place that needs the condition is Eq. (E.24), which requires $l_y(2\|\nabla_y f(x_\tau, y_\tau)) \leq l_y(2G_y)$. However, when f is Lipschitz smooth with respect to y, this condition is always satisfied since $l_y(\cdot) \equiv L_y$. The rest part of proof is the same as Lemma E.5. □

With Lemma E.5, Lemma E.6 and Eq. (E.31), we can reach the conclusion in Theorem 6.1. When $S_t \equiv S$, the result of Corollary 6.1 can be directly achieved by Theorem 6.1. Next, we will prove other corollaries using different options to compute S_t. By Cauchy-Schwartz inequality, we have the following conclusion based on Theorem 6.1.

Lemma E.7. *Suppose the conditions in Theorem 6.1 are satisfied. Then we have*

$$(\sum_{t=0}^{T-1}\|\nabla\Phi(x_t)\|)^2 \leq \frac{5(\Phi(x_0)-\Phi^*)(\sum_{t=0}^{T-1}S_t)}{\eta} \tag{E.34}$$

We also need the following Lemma E.8

Lemma E.8. *Suppose the conditions in Theorem 6.1 are satisfied. Then we have*

$$\sum_{t=0}^{T-1}\|\nabla_x f(x_t,y_t)\| \leq 2\sum_{t=0}^{T-1}\|\nabla\Phi(x_t)\| + 4\kappa G_x \tag{E.35}$$

Proof. By the proof of Lemma E.5 we have

$$\|\nabla_x f(x_t,y_t)-\nabla\Phi(x_t)\| \leq l_x(2G_x)\cdot\|y_t-y_t^*\| \tag{E.36}$$

By Eq. (E.24) we have

$$\begin{aligned}\|y_t-y_t^*\| &\leq \|y_{t-1}^*-y_t\|+\|y_t^*-y_{t-1}^*\| \leq (1-\frac{1}{2\kappa})\cdot\|y_{t-1}-y_{t-1}^*\|+\kappa\eta_{t-1}\|\nabla_x f(x_{t-1},y_{t-1})\| \\ &\leq (1-\frac{1}{2\kappa}+\kappa\eta_{t-1}l_x(2G_x))\cdot\|y_{t-1}-y_{t-1}^*\|+\kappa\eta_{t-1}\|\nabla\Phi(x_{t-1})\| \\ &\leq (1-\frac{1}{4\kappa})\cdot\|y_{t-1}-y_{t-1}^*\|+\kappa\eta_{t-1}\|\nabla\Phi(x_{t-1})\| \end{aligned} \tag{E.37}$$

Let $\gamma = 1-\frac{1}{4\kappa}$. Applying recursion to above inequality and we can obtain

$$\|y_t-y_t^*\| \leq \gamma^t\|y_0-y_0^*\|+\kappa\sum_{s=0}^{t-1}\gamma^{t-1-s}\eta_s\|\nabla\Phi(x_s)\| \tag{E.38}$$

Summing Eq. (E.36) and combining with Eq. (E.38), we achieve

$$\sum_{t=0}^{T-1}\|\nabla_x f(x_t,y_t)-\nabla\Phi(x_t)\| \le 4\kappa l_x(2G_x)\cdot\|y_0-y_0^*\|+\kappa l_x(2G_x)\sum_{t=0}^{T-1}\eta_t\|\nabla\Phi(x_t)\|\cdot\sum_{s=t}^{T-1}\gamma^{s-t}$$
$$\le 4\kappa G_x+\frac{1}{4}\sum_{t=0}^{T-1}\|\nabla\Phi(x_t)\| \tag{E.39}$$

Hence we can reach the conclusion of Lemma E.8. □

When S_t is computed by option (1) or (4), we have $\sum_{t=0}^{T-1}S_t \le \sum_{t=0}^{T-1}\|\nabla_x f(x_t,y_t)\|+T\varepsilon$ and

$$(\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla\Phi(x_t)\|)^2 \le \frac{10(\Phi(x_0)-\Phi^*)}{\eta T}(\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla\Phi(x_t)\|)+\frac{\varepsilon(\Phi(x_0)-\Phi^*)}{T}+\frac{4\kappa G_x(\Phi(x_0)-\Phi^*)}{T^2} \tag{E.40}$$

The first term on the right side is the dominant term when we have $\eta = O(\frac{\varepsilon}{\kappa^2})$ and $T = O(\kappa^2\varepsilon^{-2})$. We have

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla\Phi(x_t)\| \le \frac{20(\Phi(x_0)-\Phi^*)}{\eta T} \tag{E.41}$$

which implies the result in Corollary 6.2. When S_t is computed by option (2), we have we have $\sum_{t=0}^{T-1}S_t \le \log(T)\sum_{t=0}^{T-1}\|\nabla_x f(x_t,y_t)\|+T\varepsilon$. Mimic above steps and we can achieve Corollary 6.3. Therefore, we have completed the convergence analysis of the Generalized GDA algorithm.

E.2 Convergence Analysis of Generalized GDmax

Lemma E.9. *Let* $\eta_t = \frac{\eta}{S_t} \le \frac{C_0}{16\kappa l_x(2G_x)}$, $\eta_y = \frac{1}{l_y(4G_y)}$, $K = \kappa\log(\frac{1}{\theta})$ *and* $\|y_0-y_0^*\| \le \frac{C_0G_x}{l_x(2G_x)}$. *When* $\mathscr{Y} = \mathbb{R}^{d_2}$, *we have* $\|\nabla\Phi(x_t)\| \le G_x$, $\|\nabla_x f(x_t,y_t)\| \le 2G_x$, $\|\nabla_y f(x_t,y_t)\| \le G_y$ *and* $\|y_t -$

$y_t^*\| \leq \frac{C_0 G_x}{l_x(2G_x)}$ *for all* $t \geq 0$*, where constant* $C_0 = \min\{1, \frac{l_x(2G_x)G_y}{l_y(2G_y)G_x}\}$ *and* $\theta = \min\{\frac{1}{4}, \frac{1}{l_x^2(2G_x)}\}$.

Proof. It is easy to check that the following result in Lemma E.5 are still satisfied.

$$\begin{aligned}\Phi(x_{\tau+1}) &\leq \Phi(x_\tau) - \frac{\eta_\tau}{2}\|\nabla\Phi(x_\tau)\|^2 + \frac{\eta_\tau}{2}\|\nabla_x f(x_\tau, y_\tau) - \nabla\Phi(x_\tau)\|^2 \\ &\leq \Phi(x_\tau) - \frac{\eta_\tau}{2}\|\nabla\Phi(x_\tau)\|^2 + \frac{\eta_\tau l_x^2(2G_x)}{2}\|y_\tau - y_\tau^*\|^2 \end{aligned} \tag{E.42}$$

The difference is the way to estimate $\|y_\tau - y_\tau^*\|$. As we have

$$\eta_\tau\|\nabla_x f(x_\tau, y_\tau)\| \leq 2\eta_\tau G_x \leq \frac{G_x}{l_x(2G_x)} \leq \frac{G_x}{l_x(\|\nabla\Phi(x_\tau)\| + G_x)} \tag{E.43}$$

by Assumption 6.3 we can achieve

$$\begin{aligned}\|\nabla_y f(x_{\tau+1}, y_\tau)\| &\leq \|\nabla_y f(x_{\tau+1}, y_\tau) - \nabla_y f(x_\tau, y_\tau)\| + \|\nabla_y f(x_\tau, y_\tau)\| \\ &\leq l_y(2G_y)\cdot\eta_\tau\|\nabla_x f(x_\tau, y_\tau)\| + G_y \leq 2G_y \end{aligned} \tag{E.44}$$

where we have used the definition of η_τ and constant C_0 in the last inequality. $y_{\tau+1}$ is computed via a nested loop, which can be regarded as a strongly-convex minimization subproblem starting at $-f(x_{\tau+1}, y_\tau)$. According to the result in minimization problem (theorem 4.3 in [57]), when we set $\eta_y = \frac{1}{l_y(4G_y)}$ and $K = \kappa\log(\frac{1}{\theta})$, we have

$$\begin{aligned}\|y_{\tau+1} - y_{\tau+1}^*\|^2 &\leq \theta\|y_\tau - y_{\tau+1}^*\|^2 \leq 2\theta\|y_{\tau+1}^* - y_\tau^*\|^2 + 2\theta\|y_\tau - y_\tau^*\|^2 \\ &\leq 2\theta\kappa^2\eta_\tau^2\|\nabla_x f(x_\tau, y_\tau)\|^2 + 2\theta\|y_\tau - y_\tau^*\|^2 \leq \frac{C_0^2 G_x^2}{l_x^2(2G_x)} \end{aligned} \tag{E.45}$$

where the last inequality is achieved when $\theta \leq \frac{1}{4}$. From Eq. (E.45) we can also obtain

$$\begin{aligned}\|y_{\tau+1} - y^*_{\tau+1}\|^2 &\leq (2\theta + 4\theta\kappa^2\eta_\tau^2 l_x^2(2G_x))\|y_\tau - y^*_\tau\|^2 + 4\theta\kappa^2\eta_\tau^2 l_x^2(2G_x)\|\nabla\Phi(x_\tau)\|^2 \\ &\leq 3\theta\|y_\tau - y^*_\tau\|^2 + \frac{\theta}{64}\|\nabla\Phi(x_\tau)\|^2 \end{aligned} \tag{E.46}$$

where we have used the setup of η_t to simplify the inequality. Applying recursion to Eq. (E.42) and (E.46), we can obtain

$$\begin{aligned}\Phi(x_{\tau+1}) &\leq \Phi(x_0) - \sum_{t=0}^{\tau} \frac{\eta_t}{2}\Big(1 - \frac{\theta l_x^2(2G_x)}{16}\Big)\|\nabla\Phi(x_t)\|^2 + \frac{G_x^2}{8\kappa l_x(2G_x)} \\ &\leq \Phi(x_0) - \sum_{t=0}^{\tau} \frac{15\eta_t}{32}\|\nabla\Phi(x_t)\|^2 + \frac{G_x^2}{8\kappa l_x(2G_x)} \end{aligned} \tag{E.47}$$

where we have used $\theta \leq \min\{\frac{1}{4}, \frac{1}{l_x^2(2G_x)}\}$, $\|y_0 - y_0^*\| \leq \frac{G_x}{l_x(2G_x)}$ and $\eta_t \leq \frac{1}{16\kappa l_x(2G_x)}$. According to the definition of G_x and Lemma E.3,, we can obtain $\|\nabla\Phi(x_{\tau+1})\| \leq G_x$. As $\|y_{\tau+1} - y^*_{\tau+1}\| \leq \frac{G_x}{l_x(2G_x)} \leq \frac{G_x}{l_x(\|\nabla\Phi(x_{\tau+1})\|+G_x)}$, by Assumption 6.3 we can obtain

$$\|\nabla_x f(x_{\tau+1}, y_{\tau+1}) - \nabla\Phi(x_{\tau+1})\| \leq l_x(2G_x)\cdot\|y_{\tau+1} - y^*_{\tau+1}\| \leq G_x \tag{E.48}$$

which implies $\|\nabla_x f(x_{\tau+1}, y_{\tau+1})\| \leq 2G_x$. Finally, we need to estimate $\|\nabla_y f(x_{\tau+1}, y_{\tau+1})\|$. We have

$$\begin{aligned}\|\nabla_y f(x_{\tau+1}, y_{\tau+1})\| &= \|\nabla_y f(x_{\tau+1}, y_{\tau+1}) - \nabla_y f(x_{\tau+1}, y^*_{\tau+1})\| \\ &\leq l_y(0)\cdot\|y_{\tau+1} - y^*_{\tau+1}\| \leq \frac{C_0 l_y(0) G_x}{l_x(2G_x)} \leq G_y \end{aligned} \tag{E.49}$$

which is obtained by the definition of constant C_0. Hence we have finished the mathematical induction. □

Lemma E.10. *Let* $\eta_t = \frac{\eta}{S_t} \leq \frac{C_0}{16\kappa l_x(2G_x)}$, $\eta_y = \frac{1}{l_y(4G_y)}$, $K = \kappa\log(\frac{1}{\theta})$ *and* $\|y_0 - y_0^*\| \leq \frac{C_0 G_x}{l_x(2G_x)}$.

When $l_y(\cdot) \equiv L_y$, we have $\|\nabla\Phi(x_t)\| \leq G_x$, $\|\nabla_x f(x_t, y_t)\| \leq 2G_x$ and $\|y_t - y_t^*\| \leq \frac{C_0 G_x}{l_x(2G_x)}$ for all $t \geq 0$, where constant $C_0 = \min\{1, \frac{l_x(2G_x)G_y}{l_y(2G_y)G_x}\}$ and $\theta = \min\{\frac{1}{4}, \frac{1}{l_x^2(2G_x)}\}$.

Proof. Different from Lemma E.9, in this case we do not need to estimate an upper bound for $\nabla_y f(x_t, y_t)$. In the case of Lemma E.9, the upper bound of $\nabla_y f(x_t, y_t)$ is required because when solving the l_y-smooth strongly-convex subproblem, the stepsize and complexity are affected by the initial gradient norm. But when the function is Lipschitz smooth, the requirement is unnecessary and we can set the stepsize to $\eta_y = \frac{1}{L_y}$. □

Based on Lemma E.9, Lemma E.10 and Eq. (E.47), we can reach the conclusions of Theorem 6.2 and Corollary 6.4. Mimic the steps of Lemma E.7 and Lemma E.8, we can prove the results in Corollary 6.5 and Corollary 6.6.

E.3 Convergence Analysis of Generalized SGDA

In stochastic algorithms, we need the following auxiliary Lemmas.

Lemma E.11. *Let vector X be a stochastic variable. Then we have*

$$0 \leq \mathbb{E}\|X - \mathbb{E}X\|^2 = \mathbb{E}\|X\|^2 - \|\mathbb{E}X\|^2 \leq \mathbb{E}\|X\|^2 \tag{E.50}$$

Lemma E.12. *Let $X_1, X_2, \cdots, X_n$ be n independent stochastic variables of which the means are 0. Then we have*

$$\mathbb{E}\|\sum_{i=1}^{n} X_i\|^2 = \sum_{i=1}^{n} \mathbb{E}\|X_i\|^2 \tag{E.51}$$

Next, we will provide the proof for Theorem 6.3. Here we will only consider the case of $\mathscr{Y} = \mathbb{R}^{d_2}$ because the operations for the case $l_y(\cdot) \equiv L_y$ is similar to deterministic algorithms. For convenience, we denote $\eta_t = \frac{\eta}{S_t}$. Recall that in the stochastic case constant

G_x is re-defined as follows:

$$G_x = \max\{u > 0 | u^2 \leq 32\kappa l_x(2u) \cdot (\Phi(x_0) - \Phi^* + \sigma^2)/\delta\}$$

Proof. First, we define

$$T_0 = \min\{ \min\{t \mid \Phi(x_t) - \Phi^* > \frac{8(\Phi(x_0) - \Phi^* + \sigma^2)}{\delta} \text{ or } \|y_t - y_t^*\| > \frac{C_0 G_x}{l_x(2G_x)}\}, T\} \tag{E.52}$$

We will prove that the probability of $T_0 < T$ is small. According to the definition of G_x and T_0, we know when $t < T_0$, we have $\|\nabla\Phi(x_t)\| \leq G_x$ and $\|y_t - y_t^*\| \leq \frac{C_0 G_x}{l_x(2G_x)}$. From the proof of Lemma E.5, it can also be checked that $\|\nabla_x f(x_t, y_t)\| \leq 2G_x$ and $\|\nabla_y f(x_t, y_t)\| \leq G_y$. By the update rule of y we have

$$\begin{aligned}
\|y_t^* - y_{t+1}\|^2 &= \|y_t^* - \Pi_{\mathscr{Y}}(y_t + \eta_y u_t)\|^2 \\
&\leq \|y_t^* - y_t - \eta_y u_t\|^2 \\
&= \|y_t^* - y_t\|^2 - 2\eta_y \langle u_t - \nabla_y f(x_t, y_t), y_t^* - y_t - \eta_y \nabla_y f(x_t, y_t)\rangle - 2\eta_y \langle \nabla_y f(x_t, y_t), y_t^* - y_t\rangle \\
&\quad + \eta_y^2 \|\nabla_y f(x_t, y_t)\|^2 + \eta_y^2 \|u_t - \nabla_y f(x_t, y_t)\|^2
\end{aligned} \tag{E.53}$$

When $t < T_0$, taking expectation on ξ_t, by Eq. (E.23), Lemma E.4 and Lemma E.12 we have

$$\mathbb{E}\|y_t^* - y_{t+1}\|^2 \leq (1 - \frac{1}{\kappa})\mathbb{E}\|y_t^* - y_t\|^2 + \frac{\eta_y^2 \sigma^2}{b_y} \tag{E.54}$$

Hence by Young's inequality we have

$$\begin{aligned}\mathbb{E}\|y_{t+1}^*-y_{t+1}\|^2 &\le (1+\frac{1}{2\kappa-1})\mathbb{E}\|y_t^*-y_{t+1}\|^2+2\kappa\mathbb{E}\|y_{t+1}^*-y_t^*\|^2\\ &\le (1-\frac{1}{2\kappa})\mathbb{E}\|y_t^*-y_t\|^2+2\kappa^3\eta_t^2\mathbb{E}\|v_t\|^2+\frac{\eta_y^2\sigma^2}{b_y}\\ &\le (1-\frac{1}{2\kappa}+6\kappa^3\eta_t^2l_x^2(2G_x))\mathbb{E}\|y_t^*-y_t\|^2+6\kappa^3\eta_t^2\mathbb{E}\|\nabla\Phi(x_t)\|^2+\frac{6\kappa^3\eta_t^2\sigma^2}{b_x}+\frac{\eta_y^2\sigma^2}{b_y}\\ &\le (1-\frac{1}{4\kappa})\mathbb{E}\|y_t^*-y_t\|^2+6\kappa^3\eta_t^2G_x^2+\frac{6\kappa^3\eta_t^2\sigma^2}{b_x}+\frac{\eta_y^2\sigma^2}{b_y} \qquad \text{(E.55)}\end{aligned}$$

Applying recursion and the setup of η_t, we can achieve

$$\mathbb{E}\|y_t^*-y_t\|^2\le (1-\frac{1}{4\kappa})^t\|y_0^*-y_0\|^2+\frac{\delta C_0^2G_x^2}{96l_x^2(2G_x)}+\frac{\delta C_0^2\sigma^2}{96l_x^2(2G_x)b_x}+\frac{4\kappa\eta_y^2\sigma^2}{b_y}\le\frac{\delta C_0^2G_x^2}{16l_x^2(2G_x)} \qquad \text{(E.56)}$$

for $t\le T_0$ where we have used $b_x\ge\frac{\sigma^2}{G_x^2}$, $b_y\ge\frac{192\kappa\sigma^2l_x^2(2G_x)}{\delta G_x^2l_y^2(2G_y)}$ and the condition of $\|y_0^*-y_0\|$.

Mimic the steps in Eq. (E.21), we can also obtain

$$\begin{aligned}\Phi(x_{t+1}) &\le \Phi(x_t)+\langle\nabla\Phi(x_t),x_{t+1}-x_t\rangle+\kappa l_x(2G_x)\cdot\|x_{t+1}-x_t\|^2\\ &= \Phi(x_t)-\eta_t\langle\nabla\Phi(x_t),v_t\rangle+\kappa l_x(2G_x)\cdot\eta_t^2\|v_t\|^2 \qquad \text{(E.57)}\end{aligned}$$

for $t<T_0$. Taking expectation on ξ_t, by Lemma E.12 we have

$$\begin{aligned}\mathbb{E}\Phi(x_{t+1}) &\le \mathbb{E}\Phi(x_t)-\frac{\eta_t}{2}\mathbb{E}\|\nabla\Phi(x_t)\|^2+\frac{\eta_t}{2}\mathbb{E}\|\nabla_x f(x_t,y_t)-\nabla\Phi(x_t)\|^2+\kappa l_x(2G_x)\cdot\eta_t^2\mathbb{E}\|v_t-\nabla_x f(x_t,y_t)\|^2\\ &\quad-\frac{\eta_t}{2}(1-2\kappa l_x(2G_x)\cdot\eta_t)\mathbb{E}\|\nabla_x f(x_t,y_t)\|^2\\ &\le \mathbb{E}\Phi(x_t)-\frac{\eta_t}{2}\mathbb{E}\|\nabla\Phi(x_t)\|^2+\frac{\eta_t}{2}\mathbb{E}\|\nabla_x f(x_t,y_t)-\nabla\Phi(x_t)\|^2+\frac{\kappa l_x(2G_x)\cdot\eta_t^2\sigma^2}{b_x}\\ &\le \mathbb{E}\Phi(x_t)-\frac{\eta_t}{2}\mathbb{E}\|\nabla\Phi(x_t)\|^2+\frac{\eta_tl_x^2(2G_x)}{2}\mathbb{E}\|y_t^*-y_t\|^2+\frac{\kappa l_x(2G_x)\cdot\eta_t^2\sigma^2}{b_x} \qquad \text{(E.58)}\end{aligned}$$

From Eq. (E.55) we can also achieve

$$\mathbb{E}\|y_t^* - y_t\|^2 \leq (1 - \frac{1}{4\kappa})\mathbb{E}\|y_{t-1}^* - y_{t-1}\|^2 + 6\kappa^3\eta_t^2\mathbb{E}\|\nabla\Phi(x_{t-1})\|^2 + \frac{6\kappa^3\eta_t^2\sigma^2}{b_x} + \frac{\eta_y^2\sigma^2}{b_y} \tag{E.59}$$

Let $\gamma = 1 - \frac{1}{\kappa}$ and apply recursion to Eq. (E.59), then we can obtain

$$\mathbb{E}\|y_t^* - y_t\|^2 \leq \gamma^t\|y_0^* - y_0\|^2 + 6\kappa^3\sum_{s=0}^{t-1}\gamma^{t-1-s}\eta_s^2\mathbb{E}\|\nabla\Phi(x_s)\|^2 + \frac{\delta^2 C_0^2\sigma^2}{96l_x^2(2G_x)b_x} + \frac{4\kappa\eta_y^2\sigma^2}{b_y} \tag{E.60}$$

Insert Eq. (E.60) into Eq. (E.58) and summing over t. We have

$$\begin{aligned}\mathbb{E}\Phi(x_{t+1}) \leq \Phi(x_0) - \sum_{s=0}^{t-1}\frac{\eta_s}{2}\Big(1 - \frac{\delta^2 C_0^2}{96}\Big)\mathbb{E}\|\nabla\Phi(x_s)\|^2 + \frac{\delta C_0 l_x(2G_x)}{24\kappa}\|y_0^* - y_0\|^2 + \frac{\delta^2 C_0^2\sigma^2(t+1)}{2304\kappa^3 l_x(2G_x)b_x} \\ + \frac{\delta^3 C_0^3\sigma^2(t+1)}{9216\kappa^2 l_x(2G_x)b_x} + \frac{\eta_y^2 l_x(2G_x)\sigma^2(t+1)}{24\kappa b_y}\end{aligned} \tag{E.61}$$

As $T = \frac{\kappa^2}{\delta^2\varepsilon^2}$, $b_x \geq \frac{\sigma^2}{G_x^2\varepsilon^2}$, $b_y \geq \max\{\frac{192\kappa\sigma^2 l_x^2(2G_x)}{\delta G_x^2 l_y^2(2G_y)}, \frac{\kappa l_x(2G_x)}{\delta^2 l_y^2(2G_y)\varepsilon^2}\}$, we have

$$\mathbb{E}\Phi(x_{t+1}) \leq \Phi(x_0) - \sum_{s=0}^{t-1}\frac{95\eta_s}{192}\mathbb{E}\|\nabla\Phi(x_s)\|^2 + \frac{\delta G_x^2}{32\kappa l_x(2G_x)} + \sigma^2 \tag{E.62}$$

According to the definition of G_x, for all $t \leq T_0$ we have

$$\mathbb{E}\Phi(x_t) - \Phi^* \leq 2(\Phi(x_0) - \Phi^* + \sigma^2) \tag{E.63}$$

If $T_0 < T$, then we have $\Phi(x_t) - \Phi^* > \frac{8(\Phi(x_0) - \Phi^* + \sigma^2)}{\delta}$ or $\|y_t - y_t^*\| > \frac{C_0 G_x}{l_x(2G_x)}$ at $t = T_0$.

According to Markov's inequality and Eq. (E.56), we have

$$Pr(\|y_t - y_t^*\|^2 > \frac{C_0^2 G_x^2}{l_x^2(2G_x)} | t = T_0) \leq \mathbb{E}\|y_t^* - y_t\|^2 / (\frac{C_0^2 G_x^2}{l_x^2(2G_x)}) \leq \frac{\delta}{4} \tag{E.64}$$

According to Markov's inequality and Eq. (E.63), we have

$$Pr(\Phi(x_t) - \Phi^* > \frac{8(\Phi(x_0) - \Phi^* + \sigma^2)}{\delta} | t = T_0) \leq (\mathbb{E}\Phi(x_t) - \Phi^*) / \frac{8(\Phi(x_0) - \Phi^* + \sigma^2)}{\delta} \leq \frac{\delta}{4} \tag{E.65}$$

By union bound we have

$$Pr(T_0 < T) \leq Pr(\|y_t - y_t^*\|^2 > \frac{C_0^2 G_x^2}{l_x^2(2G_x)} | t = T_0) + Pr(\Phi(x_t) - \Phi^* > \frac{8(\Phi(x_0) - \Phi^* + \sigma^2)}{\delta} | t = T_0) \leq \frac{\delta}{2} \tag{E.66}$$

If $T_0 = T$, by Eq. (E.62) we have

$$\frac{1}{T} \sum_{t=0}^{T-1} \frac{\mathbb{E}\|\nabla \Phi(x_t)\|^2}{S_t} \leq \frac{5(\Phi(x_0) - \Phi^* + \sigma^2)}{\eta T} \tag{E.67}$$

By Markov's inequality, we have

$$\frac{1}{T} \sum_{t=0}^{T-1} \frac{\|\nabla \Phi(x_t)\|^2}{S_t} \leq \frac{10(\Phi(x_0) - \Phi^* + \sigma^2)}{\delta \eta T} \tag{E.68}$$

with probability at least $1 - \frac{\delta}{2}$. By union bound, we can finish the proof of Theorem 6.3. □

When $S_t \equiv S$, we can set $\eta_t = \frac{\delta C_0}{48\kappa^2 l_x(2G_x)}$. By Theorem 6.3, we have

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla \Phi(x_t)\|^2 \leq \frac{480(\Phi(x_0) - \Phi^* + \sigma^2)\varepsilon^2}{C_0} \tag{E.69}$$

which reaches the conclusion of Corollary 6.7. Mimic the steps in Lemma E.7 and Lemma E.8, we can prove the results in Corollary 6.8 and Corollary 6.9.

E.4 Convergence Analysis of Generalized SGDmax

In this section we will provide the proof for Theorem 6.4. Here we will only consider the case of $\mathscr{Y} = \mathbb{R}^{d_2}$ because the operations for the case $l_y(\cdot) \equiv L_y$ is similar to deterministic algorithms. For convenience, we denote $\eta_t = \frac{\eta}{S_t}$.

Proof. Similar to the analysis of SGDA, we define

$$T_0 = \min\{\ \min\{t \mid \Phi(x_t) - \Phi^* > \frac{8(\Phi(x_0) - \Phi^* + \sigma^2)}{\delta} \text{ or } \|y_t - y_t^*\| > \frac{C_0 G_x}{l_x(2G_x)}\}, T\} \tag{E.70}$$

We will prove that the probability of $T_0 < T$ is small. When $t < T_0$, according the proof of Lemma E.9 it can be checked that all induction assumptions still hold. Hence we still have

$$\mathbb{E}\Phi(x_{t+1}) \le \mathbb{E}\Phi(x_t) - \frac{\eta_t}{2}\mathbb{E}\|\nabla\Phi(x_t)\|^2 + \frac{\eta_t l_x^2(2G_x)}{2}\mathbb{E}\|y_t^* - y_t\|^2 + \frac{\kappa l_x(2G_x)\cdot\eta_t^2\sigma^2}{b_x} \tag{E.71}$$

as what we have done in Eq. (E.58). According to the update rule of y, we have

$$\begin{aligned}
&\|y_t^* - y_{t-1,k+1}\|^2 \\
&= \|y_t^* - \Pi_{\mathscr{Y}}(y_{t-1,k} + \eta_y u_{t-1,k})\|^2 \\
&\le \|y_t^* - y_{t-1,k} - \eta_y u_{t-1,k}\|^2 \\
&= \|y_t^* - y_{t-1,k}\|^2 - 2\eta_y\langle u_{t-1,k} - \nabla_y f(x_t, y_{t-1,k}), y_t^* - y_{t-1,k} - \eta_y\nabla_y f(x_t, y_{t-1,k})\rangle \\
&\quad - 2\eta_y\langle\nabla_y f(x_t, y_{t-1,k}), y_t^* - y_{t-1,k}\rangle + \eta_y^2\|\nabla_y f(x_t, y_{t-1,k})\|^2 + \eta_y^2\|u_{t-1,k} - \nabla_y f(x_t, y_{t-1,k})\|^2
\end{aligned} \tag{E.72}$$

Taking expectation and we achieve

$$\mathbb{E}\|y_t^* - y_{t-1,k+1}\|^2 \leq (1-\frac{1}{\kappa})\mathbb{E}\|y_t^* - y_{t-1,k}\|^2 + \frac{\eta_y^2\sigma^2}{b_y} \tag{E.73}$$

Apply recursion to above inequality and we can obtain

$$\mathbb{E}\|y_t^* - y_t\|^2 \leq (1-\frac{1}{\kappa})^K\mathbb{E}\|y_t^* - y_{t-1}\|^2 + \frac{\kappa\eta_y^2\sigma^2}{b_y} \tag{E.74}$$

As $K = \kappa\log(\frac{1}{\theta})$, we have

$$\begin{aligned}
\mathbb{E}\|y_t^* - y_t\|^2 &\leq \theta\mathbb{E}\|y_t^* - y_{t-1}\|^2 + \frac{\kappa\eta_y^2\sigma^2}{b_y} \\
&\leq 2\theta\mathbb{E}\|y_{t-1}^* - y_{t-1}\|^2 + 2\theta\mathbb{E}\|y_t^* - y_{t-1}^*\|^2 + \frac{\kappa\eta_y^2\sigma^2}{b_y} \\
&\leq 2\theta\mathbb{E}\|y_{t-1}^* - y_{t-1}\|^2 + 2\theta\kappa^2\eta_{t-1}^2\mathbb{E}\|v_{t-1}\|^2 + \frac{\kappa\eta_y^2\sigma^2}{b_y} \\
&\leq (2\theta + 6\theta\kappa^2\eta_{t-1}^2 l_x^2(2G_x))\mathbb{E}\|y_{t-1}^* - y_{t-1}\|^2 + 6\theta\kappa^2\eta_{t-1}^2\mathbb{E}\|\nabla\Phi(x_{t-1})\|^2 + \frac{6\theta\kappa^2\eta_{t-1}^2\sigma^2}{b_x} + \frac{\kappa\eta_y^2}{b}
\end{aligned} \tag{E.75}$$

Since we have $\eta_t \leq \frac{\delta C_0}{48\kappa l_x(2G_x)}$, $b_x \geq \frac{\sigma^2}{G_x^2}$, $b_y \geq \frac{24\kappa\sigma^2 l_x(2G_x)}{\delta G_x^2 l_y^2(4G_y)}$, $\theta \leq \frac{1}{4}$ and $\|\nabla\Phi(x_t)\| \leq G_x$ when $t < T_0$, we can obtain

$$\mathbb{E}\|y_t^* - y_t\|^2 \leq \|y_0^* - y_0\|^2 + \frac{\delta^2 C_0^2 G_x^2}{48 l_x^2(2G_x)} + \frac{\delta^2 C_0^2 G_x^2}{48 l_x^2(2G_x)} + \frac{\delta C_0^2 G_x^2}{6 l_x^2(2G_x)} \leq \frac{\delta C_0^2 G_x^2}{4 l_x^2(2G_x)} \tag{E.76}$$

for $t \leq T_0$. Besides, from Eq. (E.75) we can also obtain

$$\mathbb{E}\|y_t^* - y_t\|^2 \leq (\frac{3}{4})^t\|y_0^* - y_0\|^2 + \frac{\theta\delta^2 C_0^2}{384 l_x^2(2G_x)}\sum_{s=0}^{t-1}(\frac{3}{4})^{t-1-s}\mathbb{E}\|\nabla\Phi(x_s)\|^2 + \frac{\theta\delta^2 C_0^2\sigma^2}{96 l_x^2(2G_x)b_x} + \frac{4\kappa\eta_y^2\sigma^2}{b_y} \tag{E.77}$$

Combining with Eq. (E.71) and summing over t, we achieve

$$\mathbb{E}\Phi(x_t) \leq \Phi(x_0) - \sum_{s=0}^{t-1} \frac{\eta_s}{2}(1 - \frac{\delta^2 C_0^2}{384})\mathbb{E}\|\nabla\Phi(x_s)\|^2 + \frac{\delta C_0 l_x(2G_x)}{24\kappa}\|y_0^* - y_0\|^2 + \frac{\delta^2 C_0^2 \sigma^2 t}{2304\kappa l_x(2G_x) b_x} + \frac{\delta^3 C_0^3 \sigma^2 t}{9216\kappa l_x(2G_x) b_x} + \frac{\eta_y^2 l_x(2G_x)\sigma^2 t}{24 b_y} \tag{E.78}$$

for $t \leq T_0$. As $T = \frac{\kappa}{\delta^2\varepsilon^2}$, $b_x \geq \frac{\sigma^2}{G_x^2\varepsilon^2}$, $b_y \geq \max\{\frac{24\kappa\sigma^2 l_x^2(2G_x)}{\delta G_x^2 l_y^2(4G_y)}, \frac{\kappa l_x(2G_x)}{\delta^2 l_y^2(4G_y)\varepsilon^2}\}$, we have

$$\mathbb{E}\Phi(x_{t+1}) \leq \Phi(x_0) - \sum_{s=0}^{t-1} \frac{95\eta_s}{192}\mathbb{E}\|\nabla\Phi(x_s)\|^2 + \frac{\delta G_x^2}{32\kappa l_x(2G_x)} + \sigma^2 \tag{E.79}$$

According to the definition of G_x, for all $t \leq T_0$ we have

$$\mathbb{E}\Phi(x_t) - \Phi^* \leq 2(\Phi(x_0) - \Phi^* + \sigma^2) \tag{E.80}$$

If $T_0 < T$, then we have $\Phi(x_t) - \Phi^* > \frac{8(\Phi(x_0)-\Phi^*+\sigma^2)}{\delta}$ or $\|y_t - y_t^*\| > \frac{C_0 G_x}{l_x(2G_x)}$ at $t = T_0$. According to Markov's inequality and Eq. (E.76), we have

$$Pr(\|y_t - y_t^*\|^2 > \frac{C_0^2 G_x^2}{l_x^2(2G_x)} | t = T_0) \leq \mathbb{E}\|y_t^* - y_t\|^2 / (\frac{C_0^2 G_x^2}{l_x^2(2G_x)}) \leq \frac{\delta}{4} \tag{E.81}$$

According to Markov's inequality and Eq. (E.80), we have

$$Pr(\Phi(x_t) - \Phi^* > \frac{8(\Phi(x_0) - \Phi^* + \sigma^2)}{\delta} | t = T_0) \leq (\mathbb{E}\Phi(x_t) - \Phi^*) / \frac{8(\Phi(x_0) - \Phi^* + \sigma^2)}{\delta} \leq \frac{\delta}{4} \tag{E.82}$$

By union bound we have

$$Pr(T_0 < T) \leq Pr(\|y_t - y_t^*\|^2 > \frac{C_0^2 G_x^2}{l_x^2(2G_x)} | t = T_0) + Pr(\Phi(x_t) - \Phi^* > \frac{8(\Phi(x_0) - \Phi^* + \sigma^2)}{\delta} | t = T_0) \leq \frac{\delta}{2} \tag{E.83}$$

If $T_0 = T$, by Eq. (E.79) we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\frac{\mathbb{E}\|\nabla\Phi(x_t)\|^2}{S_t} \leq \frac{5(\Phi(x_0)-\Phi^*+\sigma^2)}{\eta T} \tag{E.84}$$

By Markov's inequality, we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\frac{\|\nabla\Phi(x_t)\|^2}{S_t} \leq \frac{10(\Phi(x_0)-\Phi^*+\sigma^2)}{\delta\eta T} \tag{E.85}$$

with probability at least $1-\frac{\delta}{2}$. By union bound, we can finish the proof of Theorem 6.4. □

The rest proof for Corollary 6.10, Corollary 6.11 and Corollary 6.12 is similar to the analysis of SGDA. Hence we will omit that part of proof to avoid redundancy.

Bibliography

[1] Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. *arXiv preprint arXiv:1704.05021*, 2017.

[2] Ahmet Alacaoglu, Yura Malitsky, Panayotis Mertikopoulos, and Volkan Cevher. A new regret analysis for adam-type algorithms. In *International conference on machine learning*, pages 202–210. PMLR, 2020.

[3] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in neural information processing systems*, 30, 2017.

[4] Dan Alistarh, Torsten Hoefler, Mikael Johansson, Nikola Konstantinov, Sarit Khirirat, and Cédric Renggli. The convergence of sparsified gradient methods. *Advances in Neural Information Processing Systems*, 31, 2018.

[5] Zeyuan Allen-Zhu and Yuanzhi Li. Neon2: Finding local minima via first-order oracles. *Advances in Neural Information Processing Systems*, 31, 2018.

[6] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018.

[7] Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd with majority vote is communication efficient and fault tolerant. *arXiv preprint arXiv:1810.05291*, 2018.

[8] Aleksandr Beznosikov, Pavel Dvurechensky, Anastasia Koloskova, Valentin Samokhin, Sebastian U Stich, and Alexander Gasnikov. Decentralized local stochastic extra-gradient for variational inequalities. *arXiv preprint arXiv:2106.08315*, 2021.

[9] Alberto Bietti, Grégoire Mialon, Dexiong Chen, and Julien Mairal. A kernel perspective for regularizing deep neural networks. In *International Conference on Machine Learning*, pages 664–674. PMLR, 2019.

[10] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. *International Colloquium on Automata, Languages, and Programming*, 2002.

[11] Congliang Chen, Li Shen, Haozhi Huang, and Wei Liu. Quantized adam with error feedback. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(5):1–26, 2021.

[12] Congliang Chen, Li Shen, Wei Liu, and Zhi-Quan Luo. Efficient-adam: Communication-efficient distributed adam with complexity analysis. *arXiv preprint arXiv:2205.14473*, 2022.

[13] Robert S Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. *Advances in Neural Information Processing Systems*, 30, 2017.

[14] Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. *arXiv preprint arXiv:1808.02941*, 2018.

[15] Zixiang Chen, Dongruo Zhou, and Quanquan Gu. Faster perturbed stochastic gradient methods for finding local minima. In Sanjoy Dasgupta and Nika Haghtalab, editors, *Proceedings of The 33rd International Conference on Algorithmic Learning Theory*, volume 167 of *Proceedings of Machine Learning Research*, pages 176–204. PMLR, 29 Mar–01 Apr 2022.

[16] Ziyi Chen, Yi Zhou, Yingbin Liang, and Zhaosong Lu. Generalized-smooth nonconvex optimization is as efficient as smooth nonconvex optimization. *arXiv preprint arXiv:2303.02854*, 2023.

[17] Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. In *Conference on Learning Theory*, pages 6–1. JMLR Workshop and Conference Proceedings, 2012.

[18] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *Neural Information Processing Systems (NeurIPS)*, 2019.

[19] Hadi Daneshmand, Jonas Kohler, Aurelien Lucchi, and Thomas Hofmann. Escaping saddles with stochastic gradients. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1155–1164, 10–15 Jul 2018.

[20] Christoph Dann, Gerhard Neumann, and Jan Peters. Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research*, 2014.

[21] 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.

[22] Thinh T Doan, Siva Theja Maguluri, and Justin Romberg. Fast convergence rates of distributed subgradient methods with adaptive quantization. *IEEE Transactions on Automatic Control*, 66(5):2191–2205, 2020.

[23] Jan Drenth. *Principles of protein X-ray crystallography*. Springer Science & Business Media, 2007.

[24] Simon S Du and Wei Hu. Linear convergence of the primal-dual gradient method for convex-concave saddle point problems without strong convexity. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 196–205. PMLR, 2019.

[25] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[26] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in neural information processing systems*, 31, 2018.

[27] Wei Gao, Rong Jin, Shenghuo Zhu, and Zhi-Hua Zhou. One-pass auc optimization. In *International conference on machine learning*, pages 906–914. PMLR, 2013.

[28] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points — online stochastic gradient for tensor decomposition. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 797–842, Paris, France, 03–06 Jul 2015. PMLR.

[29] Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International Conference on Machine Learning*, pages 1233–1242. PMLR, 2017.

[30] Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[31] Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1-2):267–305, 2016.

[32] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[33] Zhishuai Guo, Yi Xu, Wotao Yin, Rong Jin, and Tianbao Yang. On stochastic moving-average estimators for non-convex optimization. *arXiv preprint arXiv:2104.14840*, 2021.

[34] Jie Hao, Xiaochuan Gong, and Mingrui Liu. Bilevel optimization under unbounded smoothness: A new algorithm and convergence analysis, 2024.

[35] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

[36] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[37] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[38] Feihu Huang, Songcan Chen, and Heng Huang. Faster stochastic alternating direction method of multipliers for nonconvex optimization. In *International Conference on Machine Learning*, pages 2839–2848, 2019.

[39] Feihu Huang, Shangqian Gao, Jian Pei, and Heng Huang. Accelerated zeroth-order and first-order momentum methods from mini to minimax optimization. *Journal of Machine Learning Research*, 22:1–70, 2021.

[40] Feihu Huang, Shangqian Gao, Jian Pei, and Heng Huang. Accelerated zeroth-order and first-order momentum methods from mini to minimax optimization. *The Journal of Machine Learning Research*, 23(1):1616–1685, 2022.

[41] Feihu Huang and Heng Huang. Adagda: Faster adaptive gradient descent ascent methods for minimax optimization. *arXiv preprint arXiv:2106.16101*, 2021.

[42] Feihu Huang, Xidong Wu, and Heng Huang. Efficient mirror descent ascent methods for nonsmooth minimax problems. *Advances in Neural Information Processing Systems*, 34, 2021.

[43] Nikita Ivkin, Daniel Rothchild, Enayat Ullah, Ion Stoica, Raman Arora, et al. Communication-efficient distributed sgd with sketching. *Advances in Neural Information Processing Systems*, 32, 2019.

[44] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, pages 427–435. PMLR, 2013.

[45] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pages 1724–1732. PMLR, 2017.

[46] Chi Jin, Praneeth Netrapalli, and Michael Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? In *International conference on machine learning*, pages 4880–4889. PMLR, 2020.

[47] Jikai Jin, Bohang Zhang, Haiyang Wang, and Liwei Wang. Non-convex distributionally robust optimization: Non-asymptotic analysis. *Advances in Neural Information Processing Systems*, 34:2771–2782, 2021.

[48] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

[49] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. Error feedback fixes signsgd and other gradient compression schemes. In *International Conference on Machine Learning*, pages 3252–3261. PMLR, 2019.

[50] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[51] 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.

[52] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[53] Alec Koppel, Felicia Y Jakubiec, and Alejandro Ribeiro. A saddle point algorithm for networked online convex optimization. *IEEE Transactions on Signal Processing*, 63(19):5149–5164, 2015.

[54] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[55] A. Krizhevsky. Learning multiple layers of features from tiny images. *Technical Report TR-2009*, 2009.

[56] Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. *Advances in Neural Information Processing Systems*, 33:8847–8860, 2020.

[57] Haochuan Li, Jian Qian, Yi Tian, Alexander Rakhlin, and Ali Jadbabaie. Convex and non-convex optimization under generalized smoothness. *arXiv preprint arXiv:2306.01264*, 2023.

[58] Haochuan Li, Yi Tian, Jingzhao Zhang, and Ali Jadbabaie. Complexity lower bounds for nonconvex-strongly-concave min-max optimization. *Advances in Neural Information Processing Systems*, 34:1792–1804, 2021.

[59] Jiajin Li, Linglingzhi Zhu, and Anthony Man-Cho So. Nonsmooth composite nonconvex-concave minimax optimization. *arXiv preprint arXiv:2209.10825*, 2022.

[60] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania, and S. Chintala. Pytorch distributed: Experiences on accelerating data parallel training. *Proceedings of the VLDB Endowment*, 13(12), 2020.

[61] Tian Li, Zaoxing Liu, Vyas Sekar, and Virginia Smith. Privacy for free: Communication-efficient learning with differential privacy using sketches. *arXiv preprint arXiv:1911.00972*, 2019.

[62] Zhize Li. Ssrgd: Simple stochastic recursive gradient descent for escaping saddle points. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[63] 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. *Advances in neural information processing systems*, 30, 2017.

[64] 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 I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[65] Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *International Conference on Machine Learning*, pages 3043–3052. PMLR, 2018.

[66] Tianyi Lin, Chi Jin, and Michael Jordan. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*, pages 6083–6093. PMLR, 2020.

[67] Tianyi Lin, Chi Jin, and Michael I Jordan. Near-optimal algorithms for minimax optimization. In *Conference on Learning Theory*, pages 2738–2779. PMLR, 2020.

[68] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. Sparse convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 806–814, 2015.

[69] Bo Liu, Ji Liu, Mohammad Ghavamzadeh, Sridhar Mahadevan, and Marek Petrik. Finite-sample analysis of proximal gradient td algorithms. *Conference on Uncertainty in Artificial Intelligence*, 2015.

[70] Mingrui Liu, Wei Zhang, Youssef Mroueh, Xiaodong Cui, Jarret Ross, Tianbao Yang, and Payel Das. A decentralized parallel algorithm for training generative adversarial nets. *Advances in Neural Information Processing Systems*, 33:11056–11070, 2020.

[71] Sijia Liu, Songtao Lu, Xiangyi Chen, Yao Feng, Kaidi Xu, Abdullah Al-Dujaili, Minyi Hong, and Una-May O'Reilly. Min-max optimization without gradients: Convergence and applications to adversarial ml. *arXiv preprint arXiv:1909.13806*, 2019.

[72] Weijie Liu, Aryan Mokhtari, Asuman Ozdaglar, Sarath Pattathil, Zebang Shen, and Nenggan Zheng. A decentralized proximal point-type method for saddle point problems. *arXiv preprint arXiv:1910.14380*, 2019.

[73] Francesco Locatello, Alp Yurtsever, Olivier Fercoq, and Volkan Cevher. Stochastic frank-wolfe for composite convex minimization. *Advances in Neural Information Processing Systems*, 32, 2019.

[74] 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.

[75] Songtao Lu, Ioannis Tsaknakis, Mingyi Hong, and Yongxin Chen. Hybrid block successive approximation for one-sided non-convex min-max problems: algorithms and applications. *IEEE Transactions on Signal Processing*, 68:3676–3691, 2020.

[76] Songtao Lu, Kaiqing Zhang, Tianyi Chen, Tamer Basar, and Lior Horesh. Decentralized policy gradient descent ascent for safe multi-agent reinforcement learning. *Association for the Advancement of Artificial Intelligence (AAAI)*, 2021.

[77] Luo Luo, Haishan Ye, Zhichao Huang, and Tong Zhang. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. *Advances in Neural Information Processing Systems*, 33:20566–20577, 2020.

[78] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[79] Hamid Reza Maei, Csaba Szepesvári, Shalabh Bhatnagar, Doina Precup, David Silver, and Richard S. Sutton. Convergent temporal-difference learning with arbitrary smooth function approximation. *Neural Information Processing Systems (NeurIPS)*, 2010.

[80] David Mateos-Núnez and Jorge Cortés. Distributed subgradient methods for saddle-point problems. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 5462–5467. IEEE, 2015.

[81] Panayotis Mertikopoulos, Nadav Hallak, Ali Kavis, and Volkan Cevher. On the almost sure convergence of stochastic gradient descent in non-convex problems. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1117–1128. Curran Associates, Inc., 2020.

[82] Jianwei Miao, Pambos Charalambous, Janos Kirz, and David Sayre. Extending the methodology of x-ray crystallography to allow imaging of micrometre-sized non-crystalline specimens. *Nature*, 400(6742):342–344, 1999.

[83] Parvin Nazari, Davoud Ataee Tarzanagh, and George Michailidis. Dadam: A consensus-based distributed adaptive gradient method for online optimization. *IEEE Transactions on Signal Processing*, 70:6065–6079, 2022.

[84] Angelia Nedic, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.

[85] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[86] Arkadi Nemirovski. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[87] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International conference on machine learning*, pages 2613–2621. PMLR, 2017.

[88] Lam M Nguyen, Katya Scheinberg, and Martin Takáč. Inexact sarah algorithm for stochastic optimization. *Optimization Methods and Software*, 36(1):237–258, 2021.

[89] Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D Lee, and Meisam Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. *Advances in Neural Information Processing Systems*, 32, 2019.

[90] Taoxing Pan, Jun Liu, and Jie Wang. D-spider-sfo: A decentralized optimization algorithm with faster convergence rate for nonconvex problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1619–1626, 2020.

[91] Hassan Rafique, Mingrui Liu, Qihang Lin, and Tianbao Yang. Weakly-convex–concave min–max optimization: provable algorithms and applications in machine learning. *Optimization Methods and Software*, 37(3):1087–1121, 2022.

[92] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

[93] Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic frank-wolfe methods for nonconvex optimization. In *2016 54th annual Allerton conference on communication, control, and computing (Allerton)*, pages 1244–1251. IEEE, 2016.

[94] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, and Ramtin Pedarsani. An exact quantized decentralized gradient descent algorithm. *IEEE Transactions on Signal Processing*, 67(19):4934–4947, 2019.

[95] Amirhossein Reisizadeh, Hossein Taheri, Aryan Mokhtari, Hamed Hassani, and Ramtin Pedarsani. Robust and communication-efficient collaborative learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[96] Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. Fetchsgd: Communication-efficient federated learning with sketching. In *International Conference on Machine Learning*, pages 8253–8265. PMLR, 2020.

[97] Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Fifteenth annual conference of the international speech communication association*, 2014.

[98] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. *Advances in neural information processing systems*, 31, 2018.

[99] Haoran Sun, Songtao Lu, and Mingyi Hong. Improving the sample and communication complexity for decentralized non-convex optimization: Joint gradient estimation and tracking. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9217–9228. PMLR, 13–18 Jul 2020.

[100] S Sundhar Ram, Angelia Nedić, and Venugopal V Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of optimization theory and applications*, 147:516–545, 2010.

[101] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[102] Richard S. Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. *International Conference on Machine Learning (ICML)*, 2009.

[103] Csaba Szepesvári. *Algorithms for Reinforcement Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning, 2010.

[104] Hanlin Tang, Shaoduo Gan, Ammar Ahmad Awan, Samyam Rajbhandari, Conglong Li, Xiangru Lian, Ji Liu, Ce Zhang, and Yuxiong He. 1-bit adam: Communication efficient large-scale training with adam's convergence speed. *arXiv preprint arXiv:2102.02888*, 2021.

[105] Hanlin Tang, Shaoduo Gan, Samyam Rajbhandari, Xiangru Lian, Ji Liu, Yuxiong He, and Ce Zhang. Apmsqueeze: A communication efficient adam-preconditioned momentum sgd algorithm. *arXiv preprint arXiv:2008.11343*, 2020.

[106] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. d^2: Decentralized training over decentralized data. In *International Conference on Machine Learning*, pages 4848–4856. PMLR, 2018.

[107] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. d^2: Decentralized training over decentralized data. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4848–4856. PMLR, 10–15 Jul 2018.

[108] Kiran K Thekumparampil, Prateek Jain, Praneeth Netrapalli, and Sewoong Oh. Efficient algorithms for smooth minimax optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

[109] Tijmen Tieleman and Geoffrey Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *COURSERA Neural Networks Mach. Learn*, 17, 2012.

[110] Quoc Tran Dinh, Deyi Liu, and Lam Nguyen. Hybrid variance-reduced sgd algorithms for minimax problems with nonconvex-linear function. *Advances in Neural Information Processing Systems*, 33:11096–11107, 2020.

[111] Ioannis Tsaknakis, Mingyi Hong, and Sijia Liu. Decentralized min-max optimization: Formulations, algorithms and applications in network poisoning attack. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5755–5759. IEEE, 2020.

[112] Isidoros Tziotis, Constantine Caramanis, and Aryan Mokhtari. Second order optimality in decentralized non-convex optimization via perturbed gradient tracking. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21162–21173. Curran Associates, Inc., 2020.

[113] Stefan Vlaski and Ali H Sayed. Second-order guarantees in centralized, federated and decentralized nonconvex optimization. *arXiv preprint arXiv:2003.14366*, 2020.

[114] Hoi-To Wai, Mingyi Hong, Zhuoran Yang, Zhaoran Wang, and Kexin Tang. Variance reduced policy evaluation with smooth function approximation. *Advances in Neural Information Processing Systems*, 32, 2019.

[115] Hoi-To Wai, Jean Lafond, Anna Scaglione, and Eric Moulines. Decentralized frank–wolfe algorithm for convex and nonconvex problems. *IEEE Transactions on Automatic Control*, 62(11):5522–5537, 2017.

[116] Hoi-To Wai, Zhuoran Yang, Zhaoran Wang, and Mingyi Hong. Multi-agent reinforcement learning via double averaging primal-dual optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[117] Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. Spiderboost: A class of faster variance-reduced algorithms for nonconvex optimization. *arXiv*, 2018, 2018.

[118] Zhongruo Wang, Krishnakumar Balasubramanian, Shiqian Ma, and Meisam Razaviyayn. Zeroth-order algorithms for nonconvex minimax problems with improved complexities. *arXiv preprint arXiv:2001.07819*, 2020.

[119] Wenhan Xian, Feihu Huang, and Heng Huang. Communication-efficient frank-wolfe algorithm for nonconvex decentralized distributed learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10405–10413, 2021.

[120] Wenhan Xian, Feihu Huang, and Heng Huang. Communication-efficient frank-wolfe algorithm for nonconvex decentralized distributed learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):10405–10413, May 2021.

[121] Wenhan Xian, Feihu Huang, Yanfu Zhang, and Heng Huang. A faster decentralized algorithm for nonconvex minimax problems. In *Advances in Neural Information Processing Systems*, volume 34, pages 25865–25877, 2021.

[122] Cong Xie, Shuai Zheng, Sanmi Koyejo, Indranil Gupta, Mu Li, and Haibin Lin. Cser: Communication-efficient sgd with error reset. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12593–12603. Curran Associates, Inc., 2020.

[123] Ran Xin, Usman Khan, and Soummya Kar. A hybrid variance-reduced method for decentralized stochastic non-convex optimization. In *International Conference on Machine Learning*, pages 11459–11469. PMLR, 2021.

[124] Ran Xin, Usman Khan, and Soummya Kar. A hybrid variance-reduced method for decentralized stochastic non-convex optimization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11459–11469. PMLR, 18–24 Jul 2021.

[125] Ran Xin, Usman A. Khan, and Soummya Kar. An improved convergence analysis for decentralized online stochastic non-convex optimization. *IEEE Transactions on Signal Processing*, 69:1842–1858, 2021.

[126] Jinming Xu, Shanying Zhu, Yeng Chai Soh, and Lihua Xie. Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 2055–2060. IEEE, 2015.

[127] Jinming Xu, Shanying Zhu, Yeng Chai Soh, and Lihua Xie. Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 2055–2060, 2015.

[128] Miao Xu, Rong Jin, and Zhi-Hua Zhou. Speedup matrix completion with side information: Application to multi-label learning. In *Advances in neural information processing systems*, pages 2301–2309, 2013.

[129] Yi Xu, Rong Jin, and Tianbao Yang. First-order stochastic algorithms for escaping from saddle points in almost linear time. *Advances in neural information processing systems*, 31, 2018.

[130] Feng Yan, Shreyas Sundaram, SVN Vishwanathan, and Yuan Qi. Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties. *IEEE Transactions on Knowledge and Data Engineering*, 25(11):2483–2493, 2012.

[131] Yan Yan, Yi Xu, Qihang Lin, Lijun Zhang, and Tianbao Yang. Stochastic primal-dual algorithms with faster convergence than $O(1/\sqrt{T})$ for problems without bilinear structure. *arXiv:1904.10112*, 2019.

[132] Junchi Yang, Siqi Zhang, Negar Kiyavash, and Niao He. A catalyst framework for minimax optimization. In *NeurIPS*, 2020.

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

[134] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. *arXiv preprint arXiv:1905.11881*, 2019.

[135] Mingrui Zhang, Lin Chen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Quantized frank-wolfe: Faster optimization, lower communication, and projection free. In *International Conference on Artificial Intelligence and Statistics*, pages 3696–3706. PMLR, 2020.

[136] Siqi Zhang, Junchi Yang, Cristóbal Guzmán, Negar Kiyavash, and Niao He. The complexity of nonconvex-strongly-concave minimax optimization. In *Uncertainty in Artificial Intelligence*, pages 482–492. PMLR, 2021.

[137] Yanfu Zhang, Shangqian Gao, and Heng Huang. Exploration and estimation for model compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 487–496, October 2021.

[138] Shuai Zheng, Ziyue Huang, and James Kwok. Communication-efficient distributed blockwise momentum sgd with error-feedback. *Advances in Neural Information Processing Systems*, 32, 2019.

[139] Dongruo Zhou, Jinghui Chen, Yuan Cao, Yiqi Tang, Ziyan Yang, and Quanquan Gu. On the convergence of adaptive gradient methods for nonconvex optimization. *arXiv preprint arXiv:1808.05671*, 2018.