Communication-Efficient Optimization Algorithms for Decentralized Machine Learning

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Boyue Li

B.S., Electronic Engineering, Tsinghua University
M.S., Language Technologies, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

May 2023

© Boyue Li, 2023
All rights reserved.

Acknowledgements

I would like to thank the generous support by ONR under N00014-18-1-2142 and N00014-19-1-2404, ARO under W911NF-18-1-0303, AFOSR/AFRL under FA8750-20-2-0504, NSF under ECCS-1818571, CCF-1806154, CCF-1901199 and CCF-2007911, as well as the Wei Shen and Xuehong Zhang Presidential Fellowship at Carnegie Mellon University.

I am deeply grateful for my advisor and the chair of the committee, Dr. Yuejie Chi, for her mentorship and support through my PhD journey. Your passion for research and dedication to students have been inspiring me since the first day we work together. I would like to extend my sincere thanks to my other committee members: Dr. Guannan Gu, Dr. Mingyi Hong and Dr. Soummya Kar, for giving invaluable feedback to help shaping my thesis.

I would like to acknowledge everyone in Yuejie's group: Dr. Yuanxin Li, Dr. Maxime Ferreira Da Costa, Dr. Harlin Lee, Dr. Vincent J. Monardo, Dr. Tian Tong, Shicong Cen, Laixi Shi, Diogo Cardoso, Pedro Valdeira, Jiin Woo, Harry Dong, Dr. Zhize Li, Lingjing Kong, Zixin Wen and He Wang, my life in Porter Hall basement was brighter because of all of you.

Finally, special thanks to my lovely fiancée, it would be impossible for me to get through the COVID-19 pandemic without you. And thanks to my parents for your unconditional support over the years.

Boyue Li

Abstract

Emerging applications in multi-agent environments such as internet-of-things, networked sensing, large-scale machine learning and federated learning, have attracting increasing attention for decentralized optimization algorithms that are resource efficient in both computation and communication while being able to protect data privacy. This thesis considers the prototypical setting where the agents work collaboratively to minimize the sum of local loss functions by only communicating with their neighbors over a predetermined network topology. We propose four decentralized optimization algorithms, with the intertwined goals of achieving communication efficiency, computation efficiency, as well as data privacy through carefully designed update procedures. For all algorithms, we provide theoretical convergence guarantees and perform extensive numerical experiments to support the analyses.

First, we propose a Newton-type algorithm called `Network-DANE` for decentralized problems with strongly convex objectives, which utilizes gradient tracking and extra mixing (i.e., multiple mixing rounds per iteration) to extend the celebrated server/client algorithm DANE to the decentralized setting. Our analysis shows that, similar to DANE, `Network-DANE` achieves linear convergence guarantees for general smooth strongly convex and quadratic objective functions, and can provably harness data similarity across agents to accelerate convergence, which highlights its communication efficiency. We further extend `Network-DANE` by allowing a nonsmooth penalty term for composite optimization problems, and by using stochastic variance-reduced local updates for computation efficiency.

Next, for more general decentralized nonconvex empirical risk minimization (ERM) problems, we propose `DESTRESS`, which matches the optimal incremental first-order oracle (IFO) complexity of centralized algorithms for finding first-order stationary points, while maintaining communication efficiency. In addition to gradient tracking and extra mixing, `DESTRESS` also incorporates randomly activated stochastic recursive mini-batch gradient updates to avoid unnecessary computations, which allows the improvement upon prior decentralized algorithms over a wide range of parameter regimes.

Then, we consider communication compression to further improve communication efficiency for decentralized nonconvex optimization problems, which leads to the development of `BEER`. This algorithm

also leverages stochastic gradient tracking, and in addition incorporates communication compression together with error feedback to improve communication quality, which allows it to maintain communication efficiency even when data distribution over agents is highly heterogeneous.

Finally, based on `BEER`, we propose `PORTER` for decentralized nonconvex optimization, which can provably converge to global first-order stationary points while preserving each agent's privacy under the notion of differential privacy. `PORTER` utilizes stochastic gradient tracking, communication compression together with error feedback as `BEER` does, and further leverages Gaussian perturbation with gradient clipping to preserve privacy for arbitrary objective functions.

In summary, our work emphasizes 1) the effectiveness of gradient tracking in estimating global gradients, 2) by using extra mixing, communication compression and error feedback, the overall efficiency can be substantially improved, and 3) privacy can be preserved thorough gradient clipping and Gaussian perturbation. The key algorithm design ideas can also be applied, in a systematic manner, to design new resource-efficient decentralized optimization algorithms.

Keywords: decentralized optimization, communication and computation efficiency, gradient tracking, variance reduction, communication compression, error feedback, differential privacy

Contents

List of Tables

List of Figures

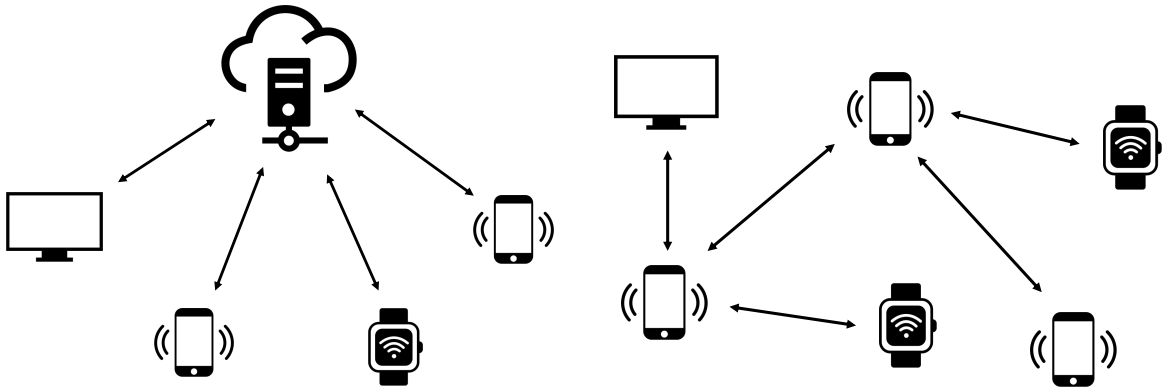
Chapter 1

Introduction

Distributed optimization has been a classic topic [BT89] yet is attracting significant attention recently in machine learning due to its numerous applications such as distributed training [BPC+11], multi-agent learning [NOP10], and federated learning [KMR15, KMY+16, MMR+17]. At least two facts contribute towards this resurgence of interest: (1) the scale of modern datasets has oftentimes far exceeded the **computation** capacity of a single machine and requires coordination across multiple machines; (2) **communication** and **privacy** constraints disfavor information sharing in a centralized manner and necessitates distributed infrastructures.

Broadly speaking, in terms of communication patterns, there are two distributed settings that have received the most interest as illustrated in Figure 1.1: 1) the *server/client* setting, which assumes the existence of a central parameter server that can aggregate and share information with all agents; and 2) the *decentralized* setting, where each agent is only permitted to communicate with its neighbors over a locally connected network specified by a communication graph. For both settings, each agent only has access to a disjoint subset of the data samples and aims to work collaboratively to optimize the global objective function $f(\boldsymbol{x})$, by only exchanging information with the parameter server or its neighbors (in other words, no centralized coordination is present). It is in general more challenging to developing algorithms for the latter setting, which is the focus of this thesis.

For a typical decentralized optimization algorithm, agents generally alternate between (1) communication, which propagates local information and enforces consensus and (2) computation, which updates individual optimization variables and improves convergence using information received from neighbors. Compared to the server/client setting, [LZZ+17] suggests decentralized algorithms can effectively avoid communication jams on busy agents, e.g., the parameter server, and be more efficient in wall-clock time. We can roughly break down an algorithm's total running cost C_{total} to the total number of iterations T



(a) The server/client setting, where all devices (clients) connect to a remote central server.

(b) The decentralized setting, where all devices connect with its neighbors without a central server.

Figure 1.1: Illustration of two distributed settings: (a) the server/client setting, and (b) the decentralized setting. This thesis focuses on the decentralized setting.

multiplied by the sum of per-iteration communication cost C_{comm} and per-iteration computation cost C_{comp}, i.e.

$$C_{\text{total}} = T \times (C_{\text{comm}} + C_{\text{comp}}). \tag{1.1}$$

For example, when reducing total time spent to reach certain accuracy is the top priority, the running cost can be defined as wall-clock time. Horizontally scaling the system by adding more agents to run in parallel may reduce T, but it will increase the communication cost C_{comm} in the meantime, which may lead to a worse total cost C_{total} in some cases. Therefore, achieving a desired level of resource efficiency for a decentralized algorithm often requires careful and delicate trade-offs between computation and communication costs, as these objectives are often conflicting in nature.

Privacy is another crucial constraint in large-scale decentralized optimization, because these applications often involve transferring computation outcomes on sensitive data that may contain personal or confidential information to external agents. Without carefully designed privacy-preserving protocols, systems may risk to directly reveal sensitive data or be vulnerable to other forms of attacks, e.g., linkage attacks [DR14]. A notable example of linkage attacks is the identification of the medical records of the governor of Massachusetts from anonymized medical data with voter registration data [Lam01], which emphasizes the importance of privacy-preserving.

1.1 Problem formulation

In this thesis, we investigate decentralized minimization problems, with the aim of achieving communication and computation efficiency simultaneously, as well as satisfying privacy-preserving constraints. In this section, we formally define the problem and related concepts.

Objective functions Consider the following minimization problem:

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimize}} \quad f(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}), \tag{1.2}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ denotes the optimization variable, $f_i(\cdot)$ denotes the local objective function at the i-th agent ($1 \le i \le n$) of n agents. In many machine learning applications, for example, empirical risk minimization (ERM) problems, local objective functions often have a finite-sum structure that is defined as

$$f_i(\boldsymbol{x}) := \frac{1}{m}\sum_{\boldsymbol{z}\in\mathcal{M}_i} \ell(\boldsymbol{x};\boldsymbol{z}), \tag{1.3}$$

where $\ell(\boldsymbol{x};\boldsymbol{z})$ denotes the sample loss of the sample $\boldsymbol{z}$ at $\boldsymbol{x}$, $\mathcal{M}_i$ denotes the dataset at agent i, $m = |\mathcal{M}_i|$ denotes the number of data samples at each agent.[1] In addition, we define the full dataset $\mathcal{M} = \cup_{i=1}^{n}\mathcal{M}_i$ and total sample size $N = |\mathcal{M}| = mn$. The communication pattern of the network is specified by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of all agents, and two agents can exchange information if and only if there is an edge in $\mathcal{E}$ connecting them.

General assumptions We assume Assumptions 1 and 2 for all analysis in this thesis. Assumption 1 guarantees the existence of nontrivial solution(s) of the global optimization problem. Assumption 2 limits the change of sample gradients, which also implies the local objective functions $f_i(\cdot)$ and global objective function $f(\cdot)$ have Lipschitz gradients.

Assumption 1 (Bounded global objective function)**.** *The global objective function $f(\boldsymbol{x})$ is bounded below, i.e.,*

$$f^\star = \inf_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) > -\infty.$$

Assumption 2 (L-smooth sample loss function)**.** *A sample loss function $\ell(\boldsymbol{x};\boldsymbol{z})$ is L-smooth if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ and $\forall \boldsymbol{z} \in \mathcal{Z}$, the following inequality holds:*

$$\left\|\nabla\ell(\boldsymbol{x};\boldsymbol{z}) - \nabla\ell(\boldsymbol{y};\boldsymbol{z})\right\|_2 \le L\|\boldsymbol{x}-\boldsymbol{y}\|_2.$$

Information mixing The information mixing between agents is conducted by updating the local information via a weighted sum of information from neighbors, which is characterized by a mixing (gossiping) matrix. Concerning this matrix is an important quantity called the mixing rate, defined in Definition 1.

Definition 1 (Mixing matrix and mixing rate)**.** *The* mixing matrix *is a matrix $\boldsymbol{W} = [w_{ij}] \in \mathbb{R}^{n\times n}$, such that $w_{ij} = 0$ if agent i and j are not connected according to the communication graph $\mathcal{G}$. Furthermore, $\boldsymbol{W}\mathbf{1}_n = \mathbf{1}_n$ and $\boldsymbol{W}^\top\mathbf{1}_n = \mathbf{1}_n$. The* mixing rate *of a mixing matrix $\boldsymbol{W}$ is defined as*

$$\alpha := \left\|\boldsymbol{W} - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right\|_{\mathsf{op}} \in [0, 1). \tag{1.4}$$

[1] We assuming the data is distributed equally among all agents, but it can be easily generalized to the unequal splitting case.

The mixing rate indicates the speed of information shared across the network. For example, for a fully connected network, choosing $\boldsymbol{W} = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ leads to $\alpha = 0$. For general networks and mixing matrices, [NOR18, Proposition 5] provides comprehensive bounds on $1-\alpha$—also known as the spectral gap—for various graphs. For instance, one has $\alpha \asymp 1$ with high probability in an Erdős-Rényi random graph if the graph is connected. In practice, FDLA matrices [XB04] are more favorable because it can achieve a much smaller mixing rate, but they usually contain negative elements and are not symmetric. Our analysis can handle arbitrary mixing matrices as long as their row/column sums equal to one and its mixing rate is smaller than 1.

Convergence metrics For strongly convex optimization algorithms, a global optimum is guaranteed to exist. Thus, we consider the distance of output to the global optimum defined in Definition 2, where a smaller ν means better performance.

Definition 2 (ν-accurate solution)**.** *The output of a deterministic optimization algorithm $\boldsymbol{x} \in \mathbb{R}^d$ is a ν-solution of a differentiable strongly convex function $f(\boldsymbol{x})$ if*

$$\|\boldsymbol{x} - \boldsymbol{x}^\star\|_2^2 \leq \nu^2,$$

where $\boldsymbol{x}^\star = \arg\min f(\boldsymbol{x})$.

For nonconvex optimization algorithms, we consider the norm of the output's gradient as the quality metric. Definition 3 defines a ν-first-order stationary point, where a smaller ν indicates better convergence.

Definition 3 (ν-first-order stationary point)**.** *The output of an optimization algorithm $\boldsymbol{x} \in \mathbb{R}^d$ is a ν-first-order stationary point of a differentiable function $f(\boldsymbol{x})$ if*

- $\|\nabla f(\boldsymbol{x})\|_2^2 \leq \nu^2$ *for deterministic optimization algorithms.*
- $\mathbb{E}\|\nabla f(\boldsymbol{x})\|_2^2 \leq \nu^2$ *for stochastic optimization algorithms.*

1.2 Contributions

Our main contribution is the design and analysis of four resource-efficient primal-only decentralized optimization algorithms: `Network-DANE` (cf. Algorithm 1), `DESTRESS` (cf. Algorithm 4), `BEER` (cf. Algorithm 5) and `PORTER` (cf. Algorithm 7) as summarized in Table 1.1. While all algorithms share a similar framework based on gradient tracking, each one focuses on a unique perspective of decentralized ERM problems, resulting in distinct algorithm designs that can achieve overall resource efficiency and address problem-specific constraints. Proposed algorithms encompass a variety of methods, such as approximate

Algorithm	Chapter	Communication efficiency	Computation efficiency	Communication compression	Differential privacy
`Network-DANE`	2	✓			
`DESTRESS`	3	✓	✓		
`BEER`	4	✓	✓	✓	
`PORTER`	5	✓	✓	✓	✓

Table 1.1: Proposed algorithms, corresponding chapters and contribution highlights.

Newton-type methods, stochastic variance-reduced methods, communication compression, error feedback and privacy perturbation.

In addition, our work indicates that by performing a judiciously chosen amount of local communication and computation per iteration, the overall efficiency can be remarkably boosted while simultaneously meeting problem-specific constraints. Extensive numerical experiments are provided to corroborate our theoretical findings, and to demonstrate the practical efficacy of the proposed algorithms over competitive baselines. All code can be accessed at

`https://github.com/liboyue/Network-Distributed-Algorithm`.

1.2.1 Decentralized Newton-style algorithm

This subsection highlights our contributions of designing the decentralized Newton-type method `Network-DANE` (cf. Chapter 2) [LCCC20], which converges linearly for smooth strongly convex objectives and quadratic objectives, and can be extended to nonsmooth composite objectives using a proximal operator.

Algorithm development We start by studying an approximate Newton-type method called DANE [SSZ14], which is among the most popular communication-efficient server/client optimization algorithms for solving ERM problems. Then, we develop `Network-DANE`, which generalizes DANE to the decentralized (network) setting. The main challenge in developing such an algorithm is to track and update a faithful estimate of the global gradient at each agent, despite the lack of centralized information aggregation. Towards this end, we leverage *dynamic average consensus* (originally proposed in the control literature [ZM10] and later adopted in decentralized optimization [QL18, NOS17, DLS16]) to track and correct locally aggregated gradients at each agent — a scheme commonly referred to as *gradient tracking*. We then employ the corrected gradient in local computations, according to the subroutine adapted from DANE. This simple idea allows one to adapt approximate Newton-type methods to network-distributed optimization, without communicating Hessian matrices.

Algorithm	Communication rounds	Extra averaging	Loss functions	β
EXTRA [SLWY15a]	$\kappa^2 \log(1/\epsilon)$	✗	Strongly convex	Arbitrary
DGD [QL18]	$\frac{\kappa^2 \log(1/\epsilon)}{(1-\alpha)^2}$	✗	Strongly convex	
`Network-DANE` (Algorithm 1)	$\frac{\kappa(\beta/\sigma+1)\log(1/\epsilon)}{(1-\alpha)^2}$	✗	Quadratic	Arbitrary
	$\log\kappa \cdot \frac{(\beta^2/\sigma^2+1)\log(1/\epsilon)}{(1-\alpha)^{1/2}}$	✓		
	$\frac{\kappa^2 \log(1/\epsilon)}{(1-\alpha)^2}$	✗	Strongly convex	
	$\log\kappa \cdot \frac{\kappa(\beta/\sigma+1)\log(1/\epsilon)}{(1-\alpha)^{1/2}}$	✓		
`Network-SVRG` (Algorithm 3)	$\log\kappa \cdot \frac{\log(1/\epsilon)}{(1-\alpha)^{1/2}}$	✓	Strongly convex	$\beta \leq \sigma/200$

Table 1.2: Communication complexity of the proposed algorithms for quadratic and strongly convex losses to reach ϵ-accuracy. Here, σ, L and $\kappa = L/\sigma$ are the strong convexity, smoothness, and condition number of the local loss functions. In addition, $\beta \leq L$ is the homogeneity parameter gauging the similarities of the local loss functions defined in (4), and α is the mixing rate defined in (1.4). Here, we assume the extra averaging step is implemented via Chebyshev acceleration scheme [AS14]. EXTRA [SLWY15a] and DGD [QL18] are listed as baselines. The big-O notation (defined in Section 1.4) is omitted for simplicity.

Our ideas for designing `Network-DANE` can be extended, in a systematic manner, to obtain decentralized versions of other algorithms developed for the server/client setting, by modifying the local computation step properly. As a notable example, we develop `Network-SVRG`, which performs variance-reduced stochastic optimization locally to enable further computational savings [JZ13]. We also demonstrate that `Network-DANE` can be extended to the proximal setting for nonsmooth composite optimization in a straightforward manner.

Efficiency analysis `Network-DANE` achieves an intriguing trade-off between communication and computation efficiency. During every iteration, each agent only communicates the parameter and gradient estimate to its neighbors, and is therefore communication-efficient globally; moreover, the local subproblem at each agent can be solved efficiently with accelerated gradient methods, and is thus computation-efficient locally. When the network exhibits a high degree of locality, we show that by allowing multiple rounds of local mixing within each iteration, an improved overall communication complexity can be achieved as it accelerates the rate of convergence. Theoretically, we establish the linear convergence of `Network-DANE` for strongly convex and quadratic losses, and show that incorporating extra mixing leads to great improvements in communication complexity. For `Network-SVRG`, we establish its linear convergence for the case of smooth strongly convex losses with extra rounds of averaging. Our analysis is highly nontrivial, as it needs to deal with the tight couplings of optimization and network consensus errors through a carefully designed linear

system of Lyapunov functions, especially in the context of approximate Newton-type methods which are known be harder to handle than simple gradient-type methods.

Table 1.2 summarizes the convergence rates of the proposed algorithm and baseline algorithms. Let σ, L, $\kappa = L/\sigma$ and β denote the strong convexity, smoothness, condition number of the local loss functions and homogeneity parameter. For general strongly convex losses, `Network-DANE` matches the results of both EXTRA [SLWY15a] and DGD [QL18] without extra mixing, but achieves a significantly improved communication complexity when incorporating extra mixing by a factor of $O\big((1-\alpha)^{3/2}(\beta/\sigma+1)\kappa^{-1}\log\kappa\big)$. The resulting convergence rate attains the optimal network dependency $O\big((1-\alpha)^{-1/2}\big)$, and improves the condition number dependency from $O(\kappa^2)$ to $O\big((\beta/\sigma+1)\kappa\log\kappa\big)$, which can be a significant improvement in the case that data similarity parameter β is small. For example, in the big data scenario, each agent can access a large number of i.i.d. data samples, which results in a small homogeneity parameter. Moreover, `Network-DANE` reaches a condition number free (up to a log factor) communication complexity for quadratic losses, which emphasizes the potential of the Newton-type method and extra mixing. `Network-SVRG` reaches a condition number free (up to a log factor) communication complexity as well for strongly convex objectives, providing β is small enough, at a much lower computation complexity than `Network-DANE` due to its stochastic nature. Our results shed light on the impacts of data homogeneity and network connectivity upon the rate of convergence.

1.2.2 Decentralized stochastic recursive gradient algorithm

This subsection highlights our contributions of the development of DEcentralized STochastic RECurSive gradient methodS (`DESTRESS`, cf. Chapter 3) [LLC22], which is a resource-efficient algorithm for decentralized nonconvex ERM problems. `DESTRESS` provably finds first-order stationary points of the global objective function $f(\boldsymbol{x})$ with the optimal incremental first-order (IFO) oracle complexity, i.e., the complexity of evaluating sample gradients, but at a much lower communication complexity compared to existing decentralized algorithms over a wide range of parameter regimes.

Algorithm development `DESTRESS` tries to improve computation and communication simultaneously to achieve overall resource efficiency. To reduce local computation, `DESTRESS` harnesses the finite-sum structure of the empirical risk function by performing stochastic variance-reduced recursive gradient updates [NvP+22, FLLZ18, WJZ+19, Li19, LR21b, LBZR21, ZXG18]—an approach that is shown to be optimal in terms of IFO complexity in the centralized setting—in a randomly activated manner to further improve computational efficiency when the local sample size is limited. To reduce communication, `DESTRESS` employs gradient tracking [ZM10] with a few mixing rounds per iteration, which helps accelerate the

Algorithms	Setting	Per-agent IFO Complexity	Communication Rounds
SVRG [AZH16, RHS+16]	centralized	$N+\frac{N^{2/3}L}{\epsilon^2}$	-
SCSG/SVRG+ [LJCJ17, LL18]	centralized	$N+\frac{N^{2/3}L}{\epsilon^2}$	-
SNVRG [ZXG18]	centralized	$N+\frac{N^{1/2}L}{\epsilon^2}$	-
SARAH/SPIDER/SpiderBoost [NvP+22, FLLZ18, WJZ+19]	centralized	$N+\frac{N^{1/2}L}{\epsilon^2}$	-
SSRGD/ZeroSARAH/PAGE [Li19, LR21b, LBZR21]	centralized	$N+\frac{N^{1/2}L}{\epsilon^2}$	-
D-GET [SLH20]	decentralized	$m+\frac{1}{(1-\alpha)^2}\cdot\frac{m^{1/2}L}{\epsilon^2}$	Same as IFO
GT-SARAH [XKK22a]	decentralized	$m+\max\left(\frac{1}{(1-\alpha)^2},\left(\frac{m}{n}\right)^{1/2},\frac{(m/n+1)^{1/3}}{1-\alpha}\right)\cdot\frac{L}{\epsilon^2}$	Same as IFO
DESTRESS (Algorithm 4)	decentralized	$m+\frac{(m/n)^{1/2}L}{\epsilon^2}$	$\frac{1}{(1-\alpha)^{1/2}}\cdot\left((mn)^{1/2}+\frac{L}{\epsilon^2}\right)$

Table 1.3: The per-agent IFO complexities and communication complexities to find ϵ-first-order stationary points by stochastic variance-reduced algorithms for nonconvex ERM problems. The first five algorithms are designed for the centralized setting, and the remaining D-GET, GT-SARAH and DESTRESS are for the decentralized setting. m, n, L are defined in Section 1.1 and α is the mixing rate defined in (1.4). The big-O notation (defined in Section 1.4) and logarithmic terms are omitted for simplicity.

convergence through better information sharing [LCCC20]; the extra mixing scheme can be implemented using Chebyshev acceleration [AS14] (detailed in Section 2.1.2) to further improve the communication efficiency.

Efficiency analysis In a nutshell, to find an ϵ-first-order stationary point (cf. Definition 3), where $\boldsymbol{x}^{\text{out}}$ is the output of DESTRESS, and the expectation is taken with respect to the randomness of the algorithm, DESTRESS requires:

- $O\big(m+(m/n)^{1/2}L/\epsilon^2\big)$ per-agent IFO calls, which is *network-independent*; and
- $O\Big(\frac{\log\left((n/m)^{1/2}+2\right)}{(1-\alpha)^{1/2}}\cdot\big((mn)^{1/2}+\frac{L}{\epsilon^2}\big)\Big)$ rounds of communication,

where L is the smoothness parameter of the sample loss, $\alpha\in[0,1)$ is the mixing rate of the mixing matrix. DESTRESS is resource-efficient for it reaches optimal computation complexity with state-of-the-art communication complexity.

Table 1.3 summarizes convergence guarantees of representative stochastic variance-reduced algorithms for finding first-order stationary points across centralized and decentralized communication settings.

- In terms of the computation complexity, the overall IFO complexity of DESTRESS—when summed

over all agents—becomes

$$n \cdot O\big(m + (m/n)^{1/2} L/\epsilon^2\big) = O\big(mn + (mn)^{1/2} L/\epsilon^2\big) = O\big(N + N^{1/2} L/\epsilon^2\big),$$

matching the optimal IFO complexity of centralized algorithms (e.g., SPIDER [FLLZ18], PAGE [LBZR21]) and distributed server/client algorithms (e.g., D-ZeroSARAH [LR21b]). However, the state-of-the-art decentralized algorithm GT-SARAH [XKK22a] is not able to achieve this optimal IFO complexity for all situations (see Table 1.3). To the best of our knowledge, DESTRESS is the first algorithm to achieve the optimal IFO complexity for the decentralized setting regardless of network topology and sample size.

- When it comes to the communication complexity, it is observed that the communication rounds of DESTRESS can be decomposed into the sum of an ϵ-independent term and an ϵ-dependent term (up to a logarithmic factor), i.e.,

$$\underbrace{\frac{1}{(1-\alpha)^{1/2}} \cdot (mn)^{1/2}}_{\epsilon-\text{independent}} + \underbrace{\frac{1}{(1-\alpha)^{1/2}} \cdot \frac{L}{\epsilon^2}}_{\epsilon-\text{dependent}};$$

similar decompositions also apply to competing decentralized algorithms. DESTRESS significantly improves the ϵ-dependent term of D-GET and GT-SARAH by at least a factor of $\frac{1}{(1-\alpha)^{3/2}}$, and therefore, saves more communications over poorly connected networks. Further, the ϵ-independent term of DESTRESS is also smaller than that of D-GET/GT-SARAH as long as the local sample size is sufficiently large, i.e., $m = \Omega\big(\frac{n}{1-\alpha}\big)$, which also holds for a wide variety of application scenarios.

In sum, DESTRESS harnesses the ideas of random client activation, variance reduction, gradient tracking and extra mixing in a sophisticated manner to achieve a scalable decentralized algorithm for nonconvex empirical risk minimization that is competitive in both computation and communication over existing approaches.

1.2.3 Decentralized stochastic algorithm with communication compression

This subsection highlights our contributions of the development of Better comprEssion for dEcentRalized optimization (BEER, cf. Chapter 4) [ZLL+22], which is a communication-efficient algorithm for decentralized nonconvex optimization problems. Communication compression is a well-established method to improve communication efficiency for server/client distributed optimization algorithms. However, for the decentralized setting, most existing algorithms, e.g. [KSJ19, TLQ+19, SDGD21], require strong assumptions as bounded gradient assumption or bounded dissimilarity assumption to guarantee convergence. BEER

Algorithm	Communication rounds	Per-agent IFO complexity	Extra assumption
SQuARM-SGD [SDGD21]	$\frac{n\tau^2}{\epsilon^2} + \frac{\sigma^2}{bn\epsilon^4}$	$\frac{n\tau^2}{\epsilon^2} + \frac{\sigma^2}{n\epsilon^4}$	Bounded gradient [(1)]
DeepSqueeze [TLQ+19]	$\frac{\beta}{\epsilon^3} + \frac{\sigma^2}{bn\epsilon^4}$	$\frac{\beta}{\epsilon^3} + \frac{\sigma^2}{n\epsilon^4}$	Bounded dissimilarity [(2)]
CHOCO-SGD [KSJ19]	$\frac{\tau}{\epsilon^3} + \frac{\sigma^2}{bn\epsilon^4}$	$\frac{\tau}{\epsilon^3} + \frac{\sigma^2}{n\epsilon^4}$	Bounded gradient [(1)]
BEER (Algorithm 5)	$\frac{1}{\epsilon^2}$	$\frac{1}{\epsilon^2} + \frac{\sigma^2}{\epsilon^4}$	-

Table 1.4: Comparison of iteration (communication) complexity, per-agent IFO complexity, and extra assumptions in addition to BEER's assumptions, for existing decentralized stochastic nonconvex optimization algorithms to reach ϵ-first-order stationary points. σ^2 and b denote the local gradient variance and batch size, respectively.
[(1)] The bounded gradient assumption is defined as $\forall \boldsymbol{x} \in \mathbb{R}^d$, $\mathbb{E}_{z_i \sim \mathcal{Z}_i} \|\nabla \ell(\boldsymbol{x}, \boldsymbol{z}_i)\|_2^2 \leq \tau^2$.
[(2)] The bounded dissimilarity assumption is defined as $\forall \boldsymbol{x} \in \mathbb{R}^d$, $\frac{1}{n}\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|_2^2 \leq \beta^2$.

removes theses strong assumptions, which enables it to converge under arbitrary data heterogeneity where other algorithms fail. BEER enjoys a faster convergence rate to find first-order stationary points of the global objective function $f(\boldsymbol{x})$ than existing algorithms that use communication compression. We establish convergence analysis for BEER using stochastic gradient for nonconvex optimization problem, and further extend this convergence to problems satisfying the Polyak-Łojasiewicz (PL) condition.

Algorithm development We develop BEER, in hope to match the convergence rate of centralized stochastic gradient descent (SGD) without aforementioned strong assumptions and under arbitrary data distributions, by leveraging gradient tracking and error feedback [RSF21]. For each variable that needs to be communicated, BEER tracks and maintains a control sequence that serves as compressed surrogate by communicating and accumulating a compressed error from the variable and the control sequence, which leads to an improved convergence rate.

Efficiency analysis To find an ϵ-first-order stationary point, using both full gradient and stochastic gradient (with proper batch size), BEER takes

- $O(\epsilon^{-2})$ iterations for optimizing nonconvex objective functions,
- $O\big(\kappa \log(1/\epsilon)\big)$ iterations for optimizing PL objectives, where κ is the condition number,

while only requires bounded local stochastic gradient variance assumption and global L-smoothness assumption.

Table 1.4 summarizes the iteration complexities, per-agent IFO complexities and extra assumptions for BEER and baseline algorithms for optimizing nonconvex objectives using minibatch stochastic gradients,

where the IFO complexities are results after using appropriate batch sizes, extra assumptions are those needed in addition to BEER's assumptions.

Let σ^2 denote the local gradient variance. When all algorithms use full gradient updates, which is equal to $\sigma = 0$, in terms of the variance-independent terms in the iteration complexities, BEER improves the dependency on ϵ from $O(\epsilon^{-3})$ to $O(\epsilon^{-2})$ and matches the centralized SGD.

When the local gradient variance is present, using appropriate batch sizes for all algorithms, BEER still attains the best iteration complexity. Furthermore, as shown in Table 1.4, SQuARM-SGD and CHOCO-SGD all require bounded gradient assumption, and DeepSqueeze requires bounded dissimilarity assumption, which results in a significant performance degeneration when those assumptions are not satisfied. On the contrary, BEER maintains the same convergence guarantees under arbitrary data heterogeneity, which highlights BEER's adaptivity to more optimization problems.

1.2.4 Decentralized private stochastic algorithm

This subsection highlights our contributions of developing PORTER (cf. Chapter 5), which is a resource-efficient and differentially private decentralized stochastic optimization algorithm for nonconvex ERM problems. PORTER is the first private decentralized optimization algorithm to incorporate communication compression and gradient clipping, to improve communication efficiency and converge for arbitrary objective functions without the bounded gradient assumption, respectively. We establish convergence analysis both with and without the bounded gradient assumption, and show explicit dependency on the mixing rate and compression parameter.

Algorithm development The focus of PORTER is preserving the privacy of each agent while being resource-efficient. To address the privacy concern, PORTER applies gradient clipping to ensure gradients are bounded for any objective function, then adds privacy perturbation to clipped gradients. To improve communication efficiency, PORTER incorporates gradient tracking [ZM10] without extra mixing to construct an estimate of the global gradient at each agent, and utilizes communication compression together with error feedback [RSF21] to reduce the amount of data to be communicated without hurting the convergence rate. Lastly, to improve computation efficiency, PORTER queries a mini-batch of samples to compute stochastic gradient per agent at each iteration.

Privacy constraints PORTER achieves (ϵ, δ)-local differential privacy (LDP) (cf. Theorem 9), which protects privacy leakage between any two agents, hence is a stronger guarantee than conventional differential privacy that only protects privacy leakage to external adversaries. Privacy is protected by adding Gaussian

Algorithm	Privacy	Compression operator	Bounded gradient	Utility	Communication rounds
DP-SGD [ACG+16]	DP	-	✓	ϕ_m	-
DDP-SRM [WJEG19]	DP	-	✓	$\frac{1}{n}\phi_m$	$n^2 d\phi_m^{-1}$
Soteria-SGD (1) [LZLC22]	LDP	Unbiased	✓	$(1+\theta^{1/2})\left(\frac{1+\omega}{n}\right)^{1/2}\phi_m$	$(1+\theta^{1/2})\left(\frac{n}{1+\omega}\right)^{2/3} d\phi_m^{-1}$
PORTER (Algorithm 8)	LDP	General	✓	$\frac{1}{(1-\alpha)^{8/3}\rho^{4/3}}\phi_m$	ϕ_m^{-2}
PORTER (Algorithm 7)	-	General	✗	$\frac{1}{(1-\alpha)^{8/3}\rho^{4/3}}\phi_m^{1/2}$	$(1-\alpha)^{8/3}\rho^{4/3}\phi_m^{-1}$

Table 1.5: Comparison of final utility upper bounds communication complexities of different stochastic optimization algorithms that achieves (ϵ, δ)-DP/LDP. $\phi_m = \frac{\sqrt{d\log(1/\delta)}}{m\epsilon}$ is the baseline utility. Big-O notation (defined in Section 1.4) is omitted for simplicity. DP-SGD is a single-machine optimization algorithm that serves as a baseline, to show the overhead brought in by the distributed setting. DDP-SRM and Soteria-SGD are server/client distributed algorithms, but DDP-SRM doesn't use communication compression.
(1) Here $\theta = (1+\omega)^{3/2} n^{-1/2}$.

perturbations to clipped stochastic gradients at each agent [DR14], where the variance of the perturbation is decided according to the target final utility and the norm of gradients (which is bounded by the clipping parameter τ).

Efficiency analysis Table 1.5 presents final utilities and corresponding communication complexities for PORTER and baseline algorithms under (ϵ, δ)-DP/LDP, where we show explicit convergence rates of PORTER for general compression operators (cf. Definition 5). When assuming the bounded gradient assumption (cf. Assumption 8), PORTER reaches $O\big((1-\alpha)^{-8/3}\rho^{-4/3}\phi_m\big)$ utility in $O(\phi_m^{-2})$ iterations. Otherwise, PORTER reaches $O\big((1-\alpha)^{-4/3}\rho^{-2/3}\phi_m^{1/4}\big)$ ℓ_2 utility in no more than $O\big((1-\alpha)^{4/3}\rho^{2/3}\phi_m^{-1}\big)$ iterations. This convergence analysis is the first that specifies explicate dependency on mixing rate and compression parameter for decentralized private optimization algorithms.

To begin with, we first assume Assumption 8 holds, Using general compression operators (cf. Definition 5), PORTER reaches $O\big((1-\alpha)^{-8/3}\rho^{-4/3}\phi_m\big)$ squared ℓ_2 utility in ϕ_m^{-2} iterations. The iteration complexity is relatively higher compared to the $O(\phi_m^{-1})$ complexity of baseline algorithms, due to extra errors induced by the decentralized setting.

However, the bounded gradient assumption is rarely met in practice, which makes it necessary for private optimization algorithms to apply gradient clipping to converge on more general objective functions. We propose a novel framework to analyze algorithms that uses gradient clipping without bounded gradient assumption, and show that using general compression operators, Algorithm 7 can reach $O\big((1-\alpha)^{-4/3}\rho^{-2/3}\phi_m^{1/4}\big)$ ℓ_2 utility in no more than $O\big((1-\alpha)^{4/3}\rho^{2/3}\phi_m^{-1}\big)$ iterations.

To conclude, PORTER is a promising approach to efficiently and privately optimizing decentralized nonconvex optimization problems by leveraging gradient tracking, communication compression, error feedback, gradient clipping and privacy perturbation in a refined manner, which is backed by novel convergence analysis that shows explicit rates under various assumptions.

1.3 Related works

Gradient tracking Gradient tracking [ZM10, QL18] provides a systematic approach to estimate the global gradient at each agent, which allows one to easily design decentralized optimization algorithms based on existing centralized algorithms. This idea is first incorporated to adjust distributed gradient descent (DGD) to ensure its linear convergence using a constant step size [NOS17, QL18, LSY19, XXK17, YYZS18, SS19, XSKK19]. Using the same idea, [LCCC20] extends Newton-style algorithms as well as stochastic variance-reduced algorithms with performance guarantees for optimizing strongly convex functions, and [ZY19, SLH20, XKK22b, XKK22a, LLC22, HSZ+22, LY22] design stochastic optimization algorithms for nonconvex problems.

Newton-type methods for distributed optimization Distributed Approximate NEwton-type Method (DANE) [SSZ14] exhibits appealing performance in both theory and practice. AIDE [RKR+16] relaxes the local optimization problem to an inexact solver and applies acceleration techniques, which improves the communication complexity to match the lower bound. CEASE [FGW21] further extends DANE to optimizing objective functions with proximal terms and improves the analysis.

Stochastic variance-reduced methods Many variants of stochastic variance-reduced gradient based methods have been proposed for finite-sum optimization for finding first-order stationary points, including but not limited to SVRG [JZ13, AZH16, RHS+16], SCSG [LJCJ17], SVRG+ [LL18], SAGA [DBLJ14], SARAH [NLST17, NvP+22], SPIDER [FLLZ18], SpiderBoost [WJZ+19], SSRGD [Li19], ZeroSARAH [LR21b] and PAGE [LBZR21]. SVRG/SVRG+/SCSG/SAGA utilize stochastic variance-reduced gradients as a corrected estimator of the full gradient, but can only achieve a sub-optimal IFO complexity of $O(N + N^{2/3}L/\epsilon)$. Other algorithms such as SARAH, SPIDER, SpiderBoost, SSRGD and PAGE adopt stochastic recursive gradients to improve the IFO complexity to $O(N + N^{1/2}L/\epsilon)$, which is optimal indicated by the lower bound [FLLZ18, LBZR21].

Decentralized stochastic nonconvex optimization There has been a flurry of recent activities in decentralized nonconvex optimization. D-PSGD [LZZ+17] and SGP [ALBR19] extend stochastic gradient descent

(SGD) to solve the nonconvex decentralized expectation minimization problems with sub-optimal rates. However, due to the noisy stochastic gradients, D-PSGD can only use diminishing step size to ensure convergence, and SGP uses a small step size on the order of $1/K$, where K denotes the total iterations, [KDG03, XB04, Sha07, BJ13, LZZ+17, WJ21] also propose decentralized algorithms with similar structures. D^2 [TLY+18] introduces a variance-reduced correction term to D-PSGD, which allows a constant step size and hence reaches a better convergence rate.

GT-SAGA [XKK22b] further uses SAGA-style updates and reaches a convergence rate that matches SAGA [DBLJ14, RSPS16]. However, GT-SAGA requires to store a variable table, which leads to a high memory complexity. D-GET [SLH20] and GT-SARAH [XKK22a] adopt equivalent recursive local gradient estimators to enable the use of constant step sizes without extra memory usage. The IFO complexity of GT-SARAH is optimal in the restrictive range $m \gtrsim \frac{n}{(1-\alpha)^6}$, while DESTRESS achieves the optimal IFO over all parameter regimes.

In addition to variance reduction techniques, performing multiple mixing steps between local updates can greatly improve the dependence of the network in convergence rates, which is equivalent of communicating over a better-connected communication graph for the agents, which in turn leads to a faster convergence (and a better overall efficiency) due to better information mixing. This technique is applied by a number of recent literatures [BBKW19, PLW20, BBW21, LCCC20, HAD+21, IW22, SBB+17, SBB+18, LFYL20, YZLZ20, GF20, LDS21].

Communication compression Communication efficiency is critical to decentralized optimization algorithms, as communication can quickly become bottleneck of the system as the number of agents and the size of the model increase. This has led to the development of communication compression (or quantization) technique, which can significantly reduce the communication burden by transferring compressed information without hurting the convergence too much. [DSZOR15, AGL+17] adopt gradient compression to create a server/client distributed stochastic gradient descent, however, the large variance of compressed gradients leads to a sub-optimal convergence rate. [SFD+14] first proposes using error feedback to compensate for the variance induced by compression. [SCJ18, AHJ+18, MGTR19, LKQR20, GBLR21, LR21a] all equip similar mechanism to improve convergence for server/client distributed optimization algorithms, and [RSF21] proposes EF21 that formalizes the error feedback mechanism and reaches a $O(1/T)$ convergence rate for smooth nonconvex objective functions. [TGZ+18, KSJ19, SDGD21, TMHP20, ZLL+22, YCC+23, LLP23] further extend communication compression schemes to the decentralized setting.

On the other hand, many compressed methods are proposed recently such as [AGL+17, KFJ18, TGZ+18, SCJ18, KSJ19, LKQR20, GBLR21, LR21a, RSF21, FSG+21, ZBLR21].

Private optimization algorithms The concern of leaking sensitive data has been increasing with the rapid development of large-scale machine learning systems. To address this concern, the concept of differential privacy is widely adopted [DMNS06, Dwo08, DR14], which is the possibility of a system to leak information under an adversarial attack. A popular approach to protect privacy is adding a noise to the model or gradients, so that the algorithm will converge to an "inexact" solution. [ACG+16, WYX17, INS+19, FKT20, CWH20] apply this method to design differentially private optimization algorithms for the single server setting, while [HDJ+20, ACCÖ21, NBD22, DLC+22] consider differential privacy for server/client distributed setting. However, merely protecting the privacy of all agents against external adversary is insufficient in the decentralized setting. It is also necessary to protect each agent's privacy from leaking to other agents, which leads to the development of locally differentially private server/client and decentralized optimization algorithms [DJW13, ABCP13, CABP13, XYD19, CSU+19, WXDX20, ZZY+21, LZLC22, ZCH+22, ZKL18, ZKL20, CEBM22, ZP23, MS23].

Gradient clipping Gradient clipping has gained significant attention in recent years. Earlier works e.g. [PMB13, BBP13, KLL16, KFI17, YGG17], use gradient clipping as a pure heuristic to solve gradient vanishing/exploding problems without theoretical understandings. Then, [ZHSJ20, ZKV+20, ZJFW20, RLDJ23] introduce theoretical analysis trying to understand the impact on the convergence rate when an algorithm adopts gradient clipping. With the recent advancements in differentially private optimization algorithms, a series of works, e.g. [CWH20, ZCH+22, DKX+22, **?**, **?**], apply this technique to limit the magnitude of gradients, so that the variance of privacy perturbation can be decided without the bounded gradient assumption.

1.4 Notation

Throughout this thesis, we use lowercase and uppercase boldface letters to represent vectors and matrices, respectively. We use $\|\cdot\|_{\mathsf{op}}$ for matrix operator norm, $\|\cdot\|_{\mathsf{F}}$ for Frobenius norm, $\|\cdot\|_2$ for 2-norm, $\otimes$ for the Kronecker product, $\odot$ for the element-wise multiplication, $\boldsymbol{I}_n$ for the n-dimensional identity matrix, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ for the inner product of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, $\mathbf{1}_n$ for the n-dimensional all-one vector and $\mathbf{0}_{d\times n}$ for the $(d \times n)$-dimensional zero matrix. For two real functions $f(\cdot)$ and $g(\cdot)$ defined on $\mathbb{R}^+$, we say $f(x) = O\big(g(x)\big)$ or $f(x) \lesssim g(x)$ if there exists some universal constant $M > 0$ such that $f(x) \leq Mg(x)$. The notation $f(x) = \Omega\big(g(x)\big)$ or $f(x) \gtrsim g(x)$ means $g(x) = O\big(f(x)\big)$.

Let $\boldsymbol{x}_i \in \mathbb{R}^d$ be the optimization variable at agent i. We define the matrix of all optimization variables

and the average vector as

$$\boldsymbol{X} := [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n} \quad \overline{\boldsymbol{x}} := \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i. \tag{1.5}$$

In addition, for a matrix of variables, we introduce the distributed gradient $\nabla F(\boldsymbol{X}) \in \mathbb{R}^{d \times n}$ as

$$\nabla F(\boldsymbol{X}) := \left[\nabla f_1(\boldsymbol{x}_1), \nabla f_2(\boldsymbol{x}_2), \ldots, \nabla f_n(\boldsymbol{x}_n)\right] \in \mathbb{R}^{d \times n}, \tag{1.6}$$

and the global gradient of the matrix $\nabla f(\boldsymbol{X}) \in \mathbb{R}^{d \times n}$ as

$$\nabla f(\boldsymbol{x}) := [\nabla f(\boldsymbol{x}_1), \cdots, \nabla f(\boldsymbol{x}_n)]. \tag{1.7}$$

1.5 Thesis organization

The rest of this thesis is organized as following: Chapters 2 to 5 develop and analyze `Network-DANE`, `DESTRESS`, `BEER` and `PORTER`, respectively, with extensive numerical experiments; Chapter 6 concludes this thesis and proposes future directions.

Chapter 2

Decentralized Newton-style algorithm

There is growing interest in large-scale machine learning and optimization over decentralized networks, e.g., in the context of multi-agent learning and federated learning. Due to the imminent need to alleviate the communication burden, the investigation of communication-efficient distributed optimization algorithms — particularly for empirical risk minimization — has flourished in recent years. A large fraction of these algorithms has been developed for the server/client setting, relying on the presence of a central parameter server that can communicate with all agents.

This chapter decentralized optimization problems defined by (1.2), where objective functions don't have finite-sum structure, and each agent is only allowed to aggregate information from its neighbors over a network (namely, no centralized coordination is present). By properly adjusting the global gradient estimate via local averaging in conjunction with proper correction, we develop a communication-efficient approximate Newton-type method, called `Network-DANE`, which generalizes DANE to accommodate decentralized scenarios. Our key ideas can be applied, in a systematic manner, to obtain decentralized versions of other server/client distributed algorithms. A notable example is the development of `Network-SVRG`, which employs variance reduction at each agent to further accelerate local computation. We establish linear convergence of `Network-DANE` and `Network-SVRG` for strongly convex losses, which shed light on the impacts of data homogeneity, network connectivity, and local averaging upon the rate of convergence. We further extend `Network-DANE` to composite optimization by allowing a nonsmooth penalty term. Numerical evidence is provided to demonstrate the appealing performance of our algorithms over competitive baselines, in terms of both communication and computation efficiency. Our work suggests that by performing a judiciously chosen amount of local communication and computation per iteration, the overall efficiency can be substantially improved.

This chapter is based on our previous publication [LCCC20].

2.1 Preliminaries

This section introduces two important concepts that are crucial to `Network-DANE` and `Network-SVRG`, and the DANE algorithm that inspires the development of `Network-DANE`.

2.1.1 Dynamic average consensus

Assume that each agent generates some *time-varying* quantity $r_j^{(t)}$ (e.g., the current local parameter or gradient estimates). Let $\boldsymbol{r}^{(t)} = [r_1^{(t)}, \cdots, r_n^{(t)}]^\top$. To track the dynamic average $\frac{1}{n}\sum_{j=1}^n r_j^{(t)} = \frac{1}{n}\mathbf{1}_n^\top \boldsymbol{r}^{(t)}$ at each agent, [ZM10] proposes a simple tracking algorithm: suppose each agent maintains an estimate $q_j^{(t)}$ in the t-th iteration, and the network collectively adopts the following update rule

$$\boldsymbol{q}^{(t)} = \boldsymbol{W}\boldsymbol{q}^{(t-1)} + \boldsymbol{r}^{(t)} - \boldsymbol{r}^{(t-1)}, \tag{2.1}$$

where $\boldsymbol{q}^{(t)} = [q_1^{(t)}, \cdots, q_n^{(t)}]^\top$. The first term $\boldsymbol{W}\boldsymbol{q}^{(t-1)}$ represents the standard local information mixing operation (meaning that each agent updates its own estimate by a weighted average of its neighbors' estimates), the second term $\boldsymbol{r}^{(t)} - \boldsymbol{r}^{(t-1)}$ tracks the temporal difference. A crucial property of (2.1) is

$$\mathbf{1}_n^\top \boldsymbol{q}^{(t)} = \mathbf{1}_n^\top \boldsymbol{r}^{(t)}, \tag{2.2}$$

which indicates that the average of $\{q_i^{(t)}\}_{1\le i\le n}$ dynamically tracks the average of $\{r_i^{(t)}\}_{1\le i\le n}$. We shall adapt this procedure in our algorithmic development, in the hope of reliably tracking the global gradients (i.e., the average of the local, and often time-varying, gradients at all agents).

2.1.2 Chebyshev's acceleration

Performing one round of mixing for $\boldsymbol{x} \in \mathbb{R}^{nd}$ using mixing matrix $\boldsymbol{W}$, the resulting vector at agent i is $\tilde{\boldsymbol{x}}_i = \sum_j w_{ij}\boldsymbol{x}_j$. If the communication network is not well-connected, we may need to perform $K > 1$ rounds of communications to improve the communication quality. If implemented as repeatedly mixing the mixed variable, the result will be equivalent as mixing using $\boldsymbol{W}^K$, and the resulting vector at agent i is $\tilde{\boldsymbol{x}}_i = \sum_j (\boldsymbol{W}^K)_{ij}\boldsymbol{x}_j$.

However, the simple implementation is not optimal, as we can construct a polynomial of $\boldsymbol{W}$ to further minimize the mixing rate of the resulting mixing matrix. Let $P_k(\boldsymbol{W}) = c_k\boldsymbol{W}^k + c_{k-1}\boldsymbol{W}^{k-1} + \ldots + c_1\boldsymbol{W} + c_0$. Optimize the mixing rate of $P_k(\boldsymbol{W})$ leads to the solution

$$P_k(x) = 1 - \frac{T_k(x/\alpha_0)}{T_k(1/\alpha_0)},$$

where $T_k(x)$ is the Chebyshev polynomial defined by

$$T_0(x) = 1$$

$$T_1(x) = x$$
$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x).$$

When α is close to 1, the mixing rate of $P_K(\boldsymbol{W})$ is approximately $\alpha_{\text{cheby}} \sim 1 - \sqrt{2(1-\alpha_0)}$.

2.1.3 Additional notation

We define the following (nd)-dimensional stacked vectors

$$\boldsymbol{x}^{(t)} := [\boldsymbol{x}_1^{(t)\top}, \cdots, \boldsymbol{x}_n^{(t)\top}]^\top, \quad \boldsymbol{y}^{(t)} := [\boldsymbol{y}_1^{(t)\top}, \cdots, \boldsymbol{y}_n^{(t)\top}]^\top, \quad \boldsymbol{s}^{(t)} := [\boldsymbol{s}_1^{(t)\top}, \cdots, \boldsymbol{s}_n^{(t)\top}]^\top. \tag{2.3}$$

With a slight abuse of nations, we introduce the stacked distributed gradient $\nabla F(\boldsymbol{x}) \in \mathbb{R}^{nd}$ and the stacked global gradient $\nabla f(\boldsymbol{x}) \in \mathbb{R}^{nd}$ of an (nd)-dimensional vector $\boldsymbol{x}$ as follows:

$$\nabla F(\boldsymbol{x}) := [\nabla f_1(\boldsymbol{x}_1)^\top, \cdots, \nabla f_n(\boldsymbol{x}_n)^\top]^\top, \quad \nabla f(\boldsymbol{x}) := [\nabla f(\boldsymbol{x}_1)^\top, \cdots, \nabla f(\boldsymbol{x}_n)^\top]^\top. \tag{2.4}$$

2.1.4 The DANE algorithm

The DANE algorithm is a popular communication-efficient approximate Newton method developed for the server/client model [SSZ14]. Here, we review some key features of DANE. (i) Each agent performs an update using both the local loss function $f_j(\cdot)$ and the gradient $\nabla f(\cdot)$ of the global loss function (obtained via the parameter server). (ii) In the t-th iteration, the j-th agent solves the following problem to update its local estimate $\boldsymbol{x}_j^{(t)}$:

$$\boldsymbol{x}_j^{(t)} = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathbb{R}^d} \left\{ f_j(\boldsymbol{x}) - \left\langle \nabla f_j(\overline{\boldsymbol{x}}^{(t)}) - \nabla f(\overline{\boldsymbol{x}}^{(t)}), \boldsymbol{x} \right\rangle + \frac{\mu}{2} \|\boldsymbol{x} - \overline{\boldsymbol{x}}^{(t)}\|_2^2 \right\}, \tag{2.5}$$

where $\mu \geq 0$ is the regularization parameter.[1] Implementing this algorithm requires two rounds of communications per iteration.

(a) The parameter server first collects all local estimates $\{\boldsymbol{x}_j^{(t-1)}\}_{1 \leq j \leq n}$ and computes the average global parameter estimate $\overline{\boldsymbol{x}}^{(t)} = \frac{1}{n} \sum_{j=1}^n \boldsymbol{x}_j^{(t-1)}$; this is then sent back to all agents.

(b) The parameter server collects all local gradients evaluated at the point $\overline{\boldsymbol{x}}^{(t)}$, computes the global gradient $\nabla f(\overline{\boldsymbol{x}}^{(t)}) = \frac{1}{n} \sum_{j=1}^n \nabla f_j(\overline{\boldsymbol{x}}^{(t)})$, and shares it with all agents.

The DANE algorithm has been demonstrated as a competitive baseline whose communication efficiency improves, in some sense, with the increase of data size [SSZ14]; see [FGW21] for its proximal variation and

[1] In [SSZ14], the second term in (2.5) takes the form $\nabla f_j(\overline{\boldsymbol{x}}^{(t)}) - \tilde{\eta} \nabla f(\overline{\boldsymbol{x}}^{(t)})$. We set $\tilde{\eta} = 1$ without loss of generality following the analysis in [FGW21].

improved theoretical analysis. To see the reason why DANE is an approximate Newton-type algorithm, consider the case when the local loss functions in all agents are quadratic and takes the form (2.10). The local optimization subproblem (2.5) in DANE can be solved in closed form, with $\boldsymbol{x}_j^{(t)}$ given by[2]

$$\boldsymbol{x}_j^{(t)} = \overline{\boldsymbol{x}}^{(t)} - \big(\underbrace{\boldsymbol{H}_j + \mu \boldsymbol{I}_d}_{\text{local Hessian}} \big)^{-1} \nabla f(\overline{\boldsymbol{x}}^{(t)}). \tag{2.6}$$

Clearly, this can be interpreted as

$$\boldsymbol{x}_j^{(t)} = \text{local parameter estimate} - \big(\text{local Hessian}\big)^{-1} \big(\text{global gradient}\big),$$

which is an approximate Newton-type update rule (since we invoke the local Hessian to approximate the true global Hessian). It is worth noting that the algorithm proceeds without communicating the local Hessians.

2.2 The `Network-DANE` algorithm

The DANE algorithm was originally developed for the server/client setting. In the network setting, however, agents can no longer compute (2.5) locally, due to the absence of centralization enabled by the parameter server; more specifically, agents have access to neither $\overline{\boldsymbol{x}}^{(t)}$ nor $\nabla f(\overline{\boldsymbol{x}}^{(t)})$, both of which are required when solving (2.5). To address this lack of global information, one might naturally wonder whether we can simply replace global averaging by local averaging; that is, replacing $\overline{\boldsymbol{x}}^{(t)}$ and $\nabla f(\overline{\boldsymbol{x}}^{(t)})$ by $\frac{1}{|\mathcal{N}_j|} \sum_{i \in \mathcal{N}_j} \boldsymbol{x}_i^{(t-1)}$ and $\frac{1}{|\mathcal{N}_j|} \sum_{i \in \mathcal{N}_j} \nabla f_i(\boldsymbol{x}_i^{(t-1)})$, respectively, in the j-th agent. However, this simple idea fails to guarantee convergence in local agents. For instance, the local estimation errors may stay flat (but nonvanishing) — as opposed to converging to zero — as the iterations progress, primarily due to imperfect information sharing.

With this convergence issue in mind, our key idea is composed of the following components.

- The first ingredient is to maintain an additional estimate of the global gradient in each agent — denoted by $\boldsymbol{s}_j^{(t)}$ in the j-th agent. This additional gradient estimate is updated via dynamic average consensus (2.8), in the hope of tracking the global gradient evaluated at $\boldsymbol{y}_j^{(t)}$ in the j-th agent ($1 \le j \le n$), i.e., $\boldsymbol{s}_j^{(t)}$ attempts to track $\nabla f(\boldsymbol{y}_j^{(t)})$. Here, $\boldsymbol{y}_j^{(t)}$ stands for the parameter estimate obtained by local neighborly averaging in the t-th iteration (see Algorithm 1 for details). As the algorithm converges, $\{\boldsymbol{y}_j^{(t)}\}_{1 \le j \le n}$ is expected to reach consensus, allowing $\boldsymbol{s}_j^{(t)}$ $(1 \le j \le n)$ to converge to the true global gradient as well.

[2] See [SSZ14] for a short derivation.

Algorithm 1 `Network-DANE`

input: initial parameter estimates $\boldsymbol{x}_j^{(0)} \in \mathbb{R}^d$ $(1 \le j \le n)$, regularization parameter μ.

initialization: set $\boldsymbol{y}_j^{(0)} = \boldsymbol{x}_j^{(0)}$, $\boldsymbol{s}_j^{(0)} = \nabla f_j(\boldsymbol{y}_j^{(0)})$ for all agents $1 \le j \le n$.

for $t = 1, 2, \cdots$ **do**

  for Agents $1 \le j \le n$ in parallel **do**

    Set $\boldsymbol{y}_j^{(t),0} = \boldsymbol{x}_j^{(t-1)}$ and $\boldsymbol{s}_j^{(t),0} = \boldsymbol{s}_j^{(t-1)}$.

    for $k = 1, 2, \ldots, K$ **do**

      Receive information $\boldsymbol{y}_i^{(t),k-1}$ and $\boldsymbol{s}_i^{(t),k-1}$ from its neighbors $i \in \mathcal{N}_j$.

      Aggregate parameter estimates from neighbors:

$$\boldsymbol{y}_j^{(t),k} = \sum\nolimits_{i \in \mathcal{N}_j} w_{ji} \boldsymbol{y}_i^{(t),k-1}, \quad \boldsymbol{s}_j^{(t),k} = \sum\nolimits_{i \in \mathcal{N}_j} w_{ji} \boldsymbol{s}_i^{(t),k-1} \tag{2.7}$$

    Set the local parameter estimate to $\boldsymbol{y}_j^{(t)} = \boldsymbol{y}_j^{(t),K}$.

    Update the global gradient estimate by aggregated local information and gradient tracking:

$$\boldsymbol{s}_j^{(t)} = \boldsymbol{s}_j^{(t),K} + \underbrace{\nabla f_j(\boldsymbol{y}_j^{(t)}) - \nabla f_j(\boldsymbol{y}_j^{(t-1)})}_{\text{gradient tracking}}. \tag{2.8}$$

    Update the parameter estimate by solving:

$$\boldsymbol{x}_j^{(t)} = \operatorname*{argmin}_{\boldsymbol{z} \in \mathbb{R}^d} \left\{ f_j(\boldsymbol{z}) - \langle \nabla f_j(\boldsymbol{y}_j^{(t)}) - \boldsymbol{s}_j^{(t)}, \boldsymbol{z} \rangle + \frac{\mu}{2} \| \boldsymbol{z} - \boldsymbol{y}_j^{(t)} \|_2^2 \right\}. \tag{2.9}$$

- In addition, we also allow multiple rounds of mixing within each iteration, i.e., (2.7), which is helpful in accelerating convergence when the network exhibits a high degree of locality. In essence, by applying K rounds of mixing, we improve the mixing rate from α to α^K. As we shall see later, choosing a proper (but not too large) K suffices to achieve the desired trade-off between the rate of information sharing and iteration complexity, which helps reduce the overall communication and computation cost. This step of extra averaging can be implemented in an efficient manner via the Chebyshev acceleration scheme [AS14, SBB+17].

Armed with such improved global gradient estimates, we propose to solve a modified local optimization subproblem (2.9) in `Network-DANE`, which approximates the original Newton-type problem (2.5) by replacing $\nabla f(\bar{\boldsymbol{x}}^{(t)})$ with the local surrogate $\boldsymbol{s}_j^{(t)}$. The proposed local subproblem (2.9) is convex and can be solved efficiently via, say, Nesterov's accelerated gradient methods. The whole algorithm is presented in

Algorithm 1.

Remark 1. *It is certainly possible to employ more general mixing matrices in (2.7). For instance, in mobile computing scenarios with moving agents, one might prefer using time-varying mixing matrices in order to accommodate the topology changes over time. We omit such extensions for brevity.*

2.3 Convergence guarantees

2.3.1 Assumptions, metrics and parameters

This section formally introduces additional assumptions, key parameters, and error metrics required for convergence analysis of `Network-DANE`.

To begin with, we introduce Assumptions 3 and 4 that characterize local objective functions.

Assumption 3 (strongly convex and smooth local objective function)**.** *The local objective function $f_i(\boldsymbol{x})$ at each agent is strongly convex and smooth, namely, $\sigma \boldsymbol{I} \preceq \nabla^2 f_j(\boldsymbol{x}) \preceq L\boldsymbol{I}$ $(1 \le j \le n)$ for some quantities $0 < \sigma \le L$. Define $\kappa = L/\sigma$ is the condition number.*

Assumption 4 (quadratic local objective function)**.** *The local objective function $f_i(\boldsymbol{x})$ at each agent is quadratic w.r.t. $\boldsymbol{x}$, i.e., taking the form of*

$$f_i(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{H}_i \boldsymbol{x} + \boldsymbol{b}_i^\top \boldsymbol{x} + c_i, \tag{2.10}$$

where $\boldsymbol{b}_i \in \mathbb{R}^d$, $c_i \in \mathbb{R}$, and $\boldsymbol{H}_i = \nabla^2 f_i(\boldsymbol{x}) \in \mathbb{R}^{d\times d}$ is a fixed symmetric and positive semidefinite matrix.

Next, we define the homogeneity parameter following [CZC+20, FGW21].

Definition 4 (Homogeneity parameter)**.** *Let $f(\boldsymbol{x})$ and $f_j(\boldsymbol{x})$ be as defined in (1.2). The homogeneity parameter β is defined as*

$$\beta := \max_{1\le j\le n} \beta_j \qquad \text{with } \beta_j := \sup_{\boldsymbol{x}\in\mathbb{R}^d} \left\|\nabla^2 f_j(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{x})\right\|. \tag{2.11}$$

As it turns out, β is bounded by the smoothness parameter of $f(\boldsymbol{x})$, i.e., $\beta \le L$.[3] On the other end, as the local loss functions f_j's become similar with each other, β will become smaller. Therefore, β is a key quantity measuring the similarity of data across agents.

[3]To prove this inequality, we note from the minimax theorem of eigenvalues and the triangle inequality that

$$\beta \le \max_j \left\{ \sup_{\substack{\boldsymbol{x}\in\mathbb{R}^d,\\ \|\boldsymbol{v}\|_2=1}} \boldsymbol{v}^\top \left(\tfrac{n-1}{n}\nabla^2 f_j(\boldsymbol{x})\right)\boldsymbol{v} - \inf_{\substack{\boldsymbol{x}\in\mathbb{R}^d,\\ \|\boldsymbol{v}\|_2=1}} \boldsymbol{v}^\top \left(\tfrac{1}{n}\sum_{i:i\ne j}\nabla^2 f_i(\boldsymbol{x})\right)\boldsymbol{v} \right\} = (1-\tfrac{1}{n})(L-\sigma) \le L.$$

Remark 2. *If the local data follow certain statistical models, it is possible to show that β decreases as the local data size m grows. For example, [SSZ14] shows that if the data samples at all agents are i.i.d. and Assumption 2 holds, then with probability at least $1-\delta$ over the samples, we have $\beta < \sqrt{\frac{32L^2}{m}\log\frac{nd}{\delta}}$ – implying β decreases at the rate of $1/\sqrt{m}$.*

In addition to notation defined in Section 1.4, we define an extra (nd)-dimensional vector $\boldsymbol{y}$ analogous to (1.5).

To characterize the convergence behavior of our algorithm, we need to simultaneously track several interrelated error metrics as follows:

(1) the convergence error: $\left\|\overline{\boldsymbol{y}}^{(t)} - \boldsymbol{y}^\star\right\|_2$;

(2) the parameter consensus error: $\left\|\boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t)}\right\|_2$;

(3) the gradient estimation error: $\left\|\boldsymbol{s}^{(t)} - \mathbf{1}_n \otimes \nabla f(\boldsymbol{y}^{(t)})\right\|_2$.

In this thesis, an algorithm is said to converge linearly at a rate $\rho \in (0,1)$ if there exists some constant $C > 0$ such that the following holds for all $t \geq 1$:

$$\max\left\{\sqrt{n}\left\|\overline{\boldsymbol{y}}^{(t)} - \boldsymbol{y}^\star\right\|_2, \left\|\boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t)}\right\|_2, L^{-1}\left\|\boldsymbol{s}^{(t)} - \nabla f(\boldsymbol{y}^{(t)})\right\|_2\right\} \leq C\rho^t.$$

In addition, an algorithm is said to reach ϵ-accuracy if the left-hand side of the above expression is bounded by ϵ.

2.3.2 Convergence guarantees of `Network-DANE` for quadratic loss

This section establishes linear convergence of `Network-DANE` when the objective functions are quadratic.

Theorem 1 (`Network-DANE` under quadratic loss, arbitrary K)**.** *Suppose that Assumptions 1, 3 and 4 hold. Set $K > 0$, the effective mixing rate becomes α^K. Set μ large enough so that $\sigma + \mu \geq \frac{140L}{(1-\alpha^K)^2}\left(\frac{\beta}{\sigma}+1\right)$. Then `Network-DANE` converges linearly at a rate ρ_1 obeying*

$$\rho_1 := \max\left\{\frac{1+\theta_1}{2}, \alpha^K + \frac{140\kappa}{1-\alpha^K}\left(\frac{\sigma+\beta}{\sigma+\mu}\right), \frac{1+\alpha^K}{2} + \frac{2\beta}{\sigma+\mu}\right\}, \tag{2.12}$$

where θ_1 is defined by

$$\theta_1 := 1 - \frac{\sigma}{\sigma+\mu} + \frac{L}{L+\mu}\frac{\beta^2}{(\sigma+\mu)(\sigma+\mu-\beta)}. \tag{2.13}$$

Remark 3. *It turns out that $\theta_1 \in (0,1)$ is the convergence rate of DANE in the server/client setting under quadratic losses [SSZ14, Theorem 1].*

It is worth noting that we do not optimized constants in the above theorem, as our primary focus is the order of convergence rate. If the regularization parameter μ is sufficiently large, one can guarantee that $\theta_1 < 1$ and hence DANE converges at a linear rate when optimizing quadratic losses [SSZ14]. We can clearly see that (2.12) is always greater than θ_1, which is the price we pay for consensus under the network setting. Fortunately, by properly setting μ, we can still guarantee that $\rho_1 < 1$, which in turn enables linear convergence of `Network-DANE`.

In view of (2.12), if the network is sufficiently connected (i.e., α is small), or if the data are sufficiently homogeneous (i.e., β is small), we can use a smaller parameter μ, which makes θ_1 (defined in (2.13)) smaller and results in faster convergence. In summary, `Network-DANE` takes fewer iterations to converge when α and β are both small. After some basic calculations, the complexity of `Network-DANE` for quadratic losses is formalized in the following corollary.

Corollary 1. *Set $\mu + \sigma = \frac{180L}{(1-\alpha^K)^2}(\frac{\beta}{\sigma} + 1)$. Under the assumptions of Theorem 1, one has*

$$\rho_1 \leq 1 - \left(\frac{1-\alpha^K}{20}\right)^2 \frac{1}{\kappa} \frac{1}{(\beta/\sigma + 1)}. \tag{2.14}$$

To reach ϵ-accuracy, `Network-DANE` takes at most $O\left(\frac{\kappa(\beta/\sigma+1)\log(1/\epsilon)}{(1-\alpha^K)^2}\right)$ iterations, and $O\left(K \cdot \frac{\kappa(\beta/\sigma+1)\log(1/\epsilon)}{(1-\alpha^K)^2}\right)$ communication rounds.

If we set the number of local averaging rounds to be $K = 1$, then the iteration complexity can be directly compared with other existing results. If the homogeneous parameter β obeys $\beta = O(\sigma)$, then the convergence rate can be improved to $O\big(\kappa \log(1/\epsilon)/(1-\alpha)^2\big)$; this is much faster than the corrected DGD [QL18] with gradient tracking, which converges in $O(\kappa^2 \log(1/\epsilon)/(1-\alpha)^2)$ iterations. The convergence rate of `Network-DANE` degenerates to that of DGD [QL18] with gradient tracking under the worst condition $\beta = \Theta(L)$. This observation highlights the communication efficiency of `Network-DANE` by harnessing the homogeneity of data across different agents. We emphasize that this is an important feature of our analysis, where the convergence rate adapts with respect to the data homogeneity.

Benefits of extra local averaging (i.e., $K > 1$). The careful reader might have noticed that the rate established above scales poorly with respect to the network parameter, namely, $(1-\alpha)^{-1}$, when $K = 1$. One remedy is to consider the case with $K > 1$, where `Network-DANE` performs K rounds of communications per iteration. On the one hand, the effective network parameter becomes α^K that can be made arbitrarily small by setting K sufficiently large, thus leading to faster convergence; on the other hand, the total number of communications is K times larger than the number of iterations, meaning that we might end up with a higher communication complexity. As an example, invoking Corollary 1, the total communication cost to

reach ϵ-accuracy is given by

$$O\big(K \cdot \kappa(1+\beta/\sigma)\log(1/\epsilon)/(1-\alpha^K)^2\big).$$

Therefore, by judiciously choosing K, it is possible to significantly improve the overall communication complexity, especially when α is close to 1. For example, by setting $K \asymp 1/\log(1/\alpha) = O(1/(1-\alpha))$, we can ensure $\alpha^K \asymp 1/2$ and reduce the communication complexity to $O\big(\kappa \cdot (\beta/\sigma+1)\log(1/\epsilon)/(1-\alpha)\big)$, thus improving the dependence with the graph topology.

The following theorem shows an improved result following a refined analysis, which improves the dependence simultaneously with respect to both κ and $(1-\alpha)^{-1}$.

Theorem 2 (`Network-DANE` under quadratic loss, optimized K)**.** *Instate the assumptions of Theorem 1. Set K and μ large enough so that $\alpha^K \le 1/(2\kappa)$ and $\sigma+\mu \ge 360\sigma\left(\frac{\beta^2}{\sigma^2}+1\right)$. To reach ϵ-accuracy, `Network-DANE` takes at most $O\left((\beta^2/\sigma^2+1)\log(1/\epsilon)\right)$ iterations, and $O\left(\log\kappa \cdot \frac{(\beta^2/\sigma^2+1)\log(1/\epsilon)}{1-\alpha}\right)$ communications rounds.*

When we set K as suggested in Theorem 2, the iteration complexity becomes independent of the network topology. Moreover, it matches the rate of DANE in the server/client setting [SSZ14] when $\beta = O(\sigma)$, which is $O(\log(1/\epsilon))$ and further independent of the condition number κ.

In terms of network dependence, the communication complexity improves from $O\big(1/(1-\alpha)^2\big)$ to $O\big(1/(1-\alpha)\big)$. By implementing the extra averaging step in an efficient manner via the well-known Chebyshev acceleration scheme [AS14, SBB+17], the dependence of the communication complexity with respect to $(1-\alpha)^{-1}$ can be further improved to $O\left((1-\alpha)^{-1/2}\right)$. The final communication complexity of `Network-DANE` for quadratic losses thus becomes

$$O\left(\log\kappa \cdot \frac{(\beta^2/\sigma^2+1)\log(1/\epsilon)}{(1-\alpha)^{1/2}}\right).$$

Therefore, the total amount of communication is significantly reduced using extra averaging, where it scales only logarithmically with respect to κ.

2.3.3 Convergence guarantees of `Network-DANE` for strongly convex loss

This section establishes the linear convergence of `Network-DANE` for general smooth and strongly convex loss functions, where the rate is worse than that for quadratic losses.

Theorem 3. *Assume Assumptions 1 and 3 hold. Set $K > 0$, and μ large enough so that $\sigma+\mu \ge \frac{170\kappa L}{(1-\alpha^K)^2}$. Then `Network-DANE` converges linearly at a rate ρ_2 obeying*

$$\rho_2 := \max\left\{\frac{1+\theta_2}{2}, \alpha^K + \frac{170\kappa}{1-\alpha^K}\left(\frac{L}{\sigma+\mu}\right), \frac{1+\alpha^K}{2} + \frac{2\beta}{\sigma+\mu}\right\}, \tag{2.15}$$

where θ_2 is given by

$$\theta_2 := 1 - \frac{\sigma}{\sigma+\mu} + \frac{\beta}{\sigma+\mu}\sqrt{1-\left(\frac{\mu}{\sigma+\mu}\right)^2}. \tag{2.16}$$

Remark 4. *Note that $\theta_2 \in (0,1)$ is precisely the convergence rate of DANE in the server/client setting [FGW21, Theorem 3.1].*

Similar to Theorem 1, one can guarantee $\theta_2 < 1$ and $\rho_2 < 1$ by setting the regularization parameter μ sufficiently large. Therefore, `Network-DANE` can converge at a linear rate for a general class of smooth and strongly convex problems. Comparing the convergence rates of `Network-DANE` derived for the above two different losses (i.e., comparing (2.13) with (2.16)), we see that: when the loss functions are non-quadratic, θ_2 is generally greater than θ_1[4]. This happens since the Hessian matrices associated with the non-quadratic loss functions may vary across different points, which is also the reason why the convergence rate of `Network-DANE` derived for the general case degenerates to the worst-case rate. After some basic calculations, the complexity of `Network-DANE` under strongly convex losses is formalized by the following corollary.

Corollary 2. *Set $\sigma + \mu = \frac{180\kappa L}{(1-\alpha^K)^2}$. Under the assumptions of Theorem 3, one has*

$$\rho_2 \leq 1 - \left(\frac{1-\alpha^K}{20}\right)^2 \frac{1}{\kappa^2}. \tag{2.17}$$

To reach ϵ-accuracy, `Network-DANE` takes at most $O\left(\frac{\kappa^2\log(1/\epsilon)}{(1-\alpha^K)^2}\right)$ iterations and $O\left(K\cdot\frac{\kappa^2\log(1/\epsilon)}{(1-\alpha^K)^2}\right)$ communication rounds.

When $K = 1$, the communication complexity of `Network-DANE` is $O\left(\frac{\kappa^2\log(1/\epsilon)}{(1-\alpha)^2}\right)$, which is rather pessimistic and does not improve with data homogeneity. Similar to Theorem 2, we can improve this by optimizing K properly. We have the following theorem, which is parallel to Theorem 2.

Theorem 4 (`Network-DANE` under strongly convex loss, optimized K)**.** *Instate the assumptions of Theorem 3. Set K and μ large enough so that $\alpha^K \leq 1/(2\kappa)$ and $\sigma + \mu \geq 360L\left(\frac{\beta}{\sigma}+1\right)$. To reach ϵ-accuracy, `Network-DANE` takes at most $O\left(\kappa(\beta/\sigma+1)\log(1/\epsilon)\right)$ iterations and $O\left(\log\kappa\cdot\frac{\kappa(\beta/\sigma+1)\log(1/\epsilon)}{1-\alpha}\right)$ communication rounds.*

The improved rate in Theorem 4 improves as the local data become more homogeneous, recovering a feature that has been highlighted previously. Similar to earlier discussions in Section 2.1.2, by using the Chebyshev acceleration scheme [AS14, SBB+17], the final communication complexity of `Network-DANE` for strongly convex losses becomes

$$O\left(\log\kappa\cdot\frac{\kappa(\beta/\sigma+1)\log(1/\epsilon)}{(1-\alpha)^{1/2}}\right).$$

[4]This is because $\sqrt{\frac{\sigma^2+2\sigma\mu}{(\sigma+\mu)^2}} \geq \frac{\sigma}{\sigma+\mu}$.

Remark 5. *The homogeneity parameter β defined in Definition 4 measures the largest deviation of local Hessians from the global Hessian. A refined analysis using local deviation β_j is possible by permitting different regularization parameters μ_j in (2.9) for different agents.*

2.4 Extension to nonsmooth composite optimization

`Network-DANE` can be extended for nonsmooth composite optimization, by properly adjusting the local optimization step, leveraging proximal variants of DANE [FGW21] and SVRG [XZ14]. For simplicity, we present the proximal variant of `Network-DANE` and leave its theoretical analysis to future work.

Consider the following regularized empirical risk minimization problem:

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimize}} \quad f(\boldsymbol{x}) + g(\boldsymbol{x}) \triangleq \frac{1}{N}\sum_{i=1}^{N} \ell(\boldsymbol{x};\boldsymbol{z}_i) + g(\boldsymbol{x}), \tag{2.18}$$

where $f(\cdot)$ and $f_j(\cdot)$ are defined as in (1.2), and $g(\cdot)$ is a deterministic convex regularizer that can be nonsmooth. This type of problem has wide applications, where it is desirable to promote additional structures or incorporate prior knowledge about the solution through adding a deterministic regularization term $g(\boldsymbol{x})$. We can extend `Network-DANE` to solve (2.18) by adding the proximal term into the local optimization step, as detailed in Algorithm 2, which is a direct extension of Algorithm 1. Section 2.6 numerically verifies the effectiveness of Algorithm 2.

Algorithm 2 `Network-DANE` for nonsmooth composite optimization

1 Replace the local optimization sub-problem (2.9) of `Network-DANE` by the following:

2 **Input:** $\boldsymbol{y}_j^{(t)}$, $\boldsymbol{s}_j^{(t)}$, regularization parameter μ.

3 Update the parameter estimate by solving:

$$\boldsymbol{x}_j^{(t)} = \underset{\boldsymbol{z}\in\mathbb{R}^d}{\operatorname{argmin}} \left\{ f_j(\boldsymbol{z}) + g(\boldsymbol{z}) - \langle \nabla f_j(\boldsymbol{y}_j^{(t)}) - \boldsymbol{s}_j^{(t)}, \boldsymbol{z}\rangle + \frac{\mu}{2}\|\boldsymbol{z} - \boldsymbol{y}_j^{(t)}\|_2^2 \right\}. \tag{2.19}$$

2.5 Extension with variance reduction

The design of `Network-DANE` suggests a systematic approach to obtain decentralized versions of other algorithms. We illustrate this by reducing local computation of `Network-DANE` using variance reduction. Stochastic variance reduction methods are a popular class of stochastic optimization algorithms, developed to allow for constant step sizes and faster convergence in finite-sum optimization [JZ13, XZ14, NLST17]. It is therefore natural to ask whether such variance reduction techniques can be leveraged in a network setting to further save local computation without compromising communication.

Algorithm 3 `Network-SVRG`

Replace the local optimization subproblem (2.9) of `Network-DANE` by the following:

Input: $\boldsymbol{y}_j^{(t)}$, $\boldsymbol{s}_j^{(t)}$, step size δ, number of local iterations S.

Initialization: set $\boldsymbol{u}_j^{(t),0} = \boldsymbol{y}_j^{(t)}$, $\boldsymbol{v}_j^{(t),0} = \boldsymbol{s}_j^{(t)}$.

for $s = 1, ..., S$ **do**

$\boldsymbol{u}_j^{(t),s} = \boldsymbol{u}_j^{(t),s-1} - \delta \boldsymbol{v}_j^{(t),s-1}$.

Sample $\boldsymbol{z}$ from $\mathcal{M}_j$ uniformly at random, then,

$$\boldsymbol{v}_j^{(t),s} = \nabla \ell(\boldsymbol{u}_j^{(t),s}; \boldsymbol{z}) - \nabla \ell(\boldsymbol{u}_j^{(t),0}; \boldsymbol{z}) + \boldsymbol{v}_j^{(t),0}. \tag{2.20}$$

Choose the new parameter estimate $\boldsymbol{x}_j^{(t)}$ from $\{\boldsymbol{u}_j^{(t),1}, \cdots, \boldsymbol{u}_j^{(t),S}\}$ uniformly at random.

Inspired by the connection between DANE and SVRG [KMR15], we introduce `Network-SVRG` in Algorithm 3, a decentralized version of SVRG [JZ13] tailored to the network setting, with the assistance of gradient tracking. In particular, the inner loops of SVRG [JZ13] are adopted to replace the local computation subproblem (2.9) of `Network-DANE`, where the reference to the global gradient is replaced by $\boldsymbol{s}_j^{(t)}$ to calculate the variance-reduced stochastic gradient.

The convergence analysis of Algorithm 3 is more challenging due to the biased stochastic gradient involved in each local iteration. Theorem 5 establishes the linear convergence of `Network-SVRG` for strongly convex losses, as long as β is sufficiently small and the number of mixing rounds K is sufficiently large. Again, we have not strived to improve the pre-constants specified in the theorem.

Theorem 5. *Assume Assumptions 1 and 3 hold and $\beta/\sigma \leq 1/200$. Set K large enough such that $\alpha^K \asymp 1/\kappa$ and S large enough, `Network-SVRG` converges linearly. To reach ε-accuracy, `Network-SVRG` takes at most $O(\log(1/\varepsilon))$ iterations and $O\left(\log\kappa \cdot \frac{\log(1/\varepsilon)}{1-\alpha}\right)$ communication rounds.*

The proof of Theorem 5 can be found in Appendix A.4. Theorem 5 implies that: as long as the local data are sufficiently similar (so that β does not exceed the order of σ), by performing $O(\log\kappa/(1-\alpha))$ rounds of local communication per iteration, `Network-SVRG` converges in $O(\log(1/\varepsilon))$ iterations independent of κ. This performance guarantee matches its counterpart in the server/client setting [CZC+20]. Altogether, `Network-SVRG` achieves appealing computation and communication complexities simultaneously. By further adopting the Chebyshev acceleration scheme [AS14, SBB+17], the final communication complexity of `Network-SVRG` is at most

$$O\left(\log\kappa \cdot \frac{\log(1/\varepsilon)}{(1-\alpha)^{1/2}}\right).$$

It is straightforward to extend this idea to obtain decentralized variants of other stochastic variance

reduced algorithms such as Katyusha [AZ17], by replacing the local computation step (2.9) by the inner loop update rules of the stochastic methods of interest. For the sake of brevity, this paper does not pursue such "plug-and-play" extensions.

Remark 6. *Our convergence theory of* `Network-SVRG` *requires* $\beta \lesssim \sigma$*, which is consistent with its counterpart in the server/client setting [CZC+20]. In contrast,* `Network-DANE` *is guaranteed to converge linearly in the entire range of* β *by setting* μ *sufficiently large. One scheme to relax this requirement, as analyzed in [CZC+20], is to add a regularization term, similar to the last term in (2.9), that penalizes the distance to the previous estimate. However, this might come at a price of slower convergence. We leave this to future investigation.*

2.6 Numerical experiments

We evaluate the performance of the proposed algorithms[5] for solving both strongly convex and nonconvex problems, in order to demonstrate the appealing performance in terms of communication-computation trade-offs.

Throughout this section, we set the number of agents $n = 20$. We use symmetric fastest distributed linear averaging (FDLA) matrices [XB04] generated according to the communication graph as the mixing matrix $\boldsymbol{W}$ for aggregating $\boldsymbol{x}_j^{(t)}$ in (2.7). For aggregating $\boldsymbol{s}_j^{(t)}$ in (2.7), we use a convex combination of $\boldsymbol{I}$ and $\boldsymbol{W}$ such that its diagonal elements are greater than 0.1, which makes the algorithm more stable in practice. The same regularization parameter μ is used for DANE and `Network-DANE`. We generate connected random communication graphs using an Erdős-Rényi graph with the probability of connectivity $p = 0.3$ (if not specified). For each experiment, we use the same random starting point $\boldsymbol{x}^{(0)}$ and mixing matrix $\boldsymbol{W}$ for all algorithms. To solve the local optimization subproblems, we use Nesterov's accelerated gradient descent for at most 100 iterations for DANE and `Network-DANE`.

2.6.1 Experiments on synthetic data

We conduct five synthetic numerical experiments based on linear regression to investigate the performance of our algorithms. The same data generation method is used for all synthetic experiments. We generate $m = 1000$ samples of dimension $d = 40$, denoted by $\boldsymbol{A}_i$, randomly from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ i.i.d. for each agent, where $\boldsymbol{\Sigma}$ is a diagonal matrix with $\boldsymbol{\Sigma}_{ii} = i^{-\varrho}$. By changing ϱ, we can change the condition number κ. Data samples are generated according to the linear model $\boldsymbol{b}_i = \boldsymbol{A}_i \boldsymbol{x}_0 + \boldsymbol{\xi}_i$, with a random signal $\boldsymbol{x}_0$ and i.i.d. noise $\boldsymbol{\xi}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. For DANE and `Network-DANE`, we set $\mu = 5 \times 10^{-10}$ when $\kappa = 10$ and $\mu = 5 \times 10^{-4}$ when

[5] In our experiments of `Network-SVRG`, we use the last iterate $\boldsymbol{u}_j^{(t),S}$ as the new parameter estimate locally, which is more practical; though our analysis only handles the case where the new parameter estimate is selected uniformly at random from previous iterates.

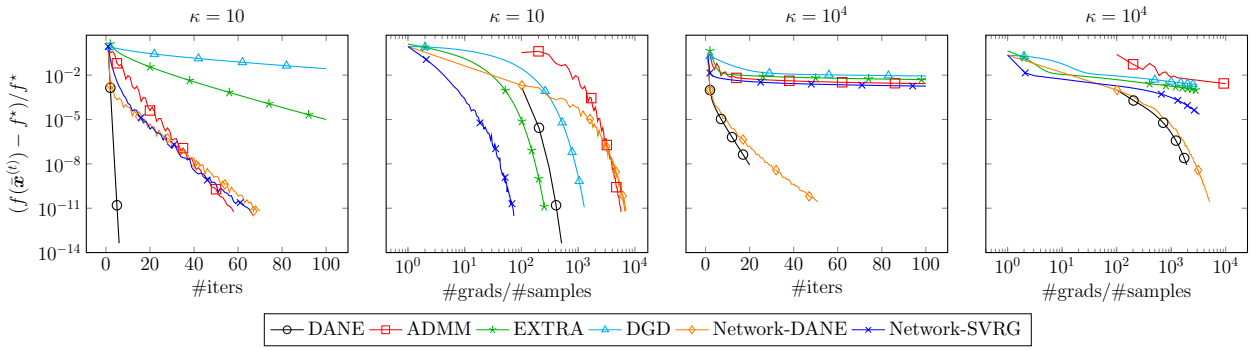


Figure 2.1: The relative optimality gap with respect to the number of iterations and gradient evaluations under different conditioning $\kappa = 10$ (left two panels) and $\kappa = 10^4$ (right two panels) for linear regression.

$\kappa = 10^4$. For `Network-SVRG`, we set the step size $\delta = 0.1/(L + \sigma + 2\mu)$ and the number of local iterations $S = 0.05m$.

Comparison with existing algorithms. To make a fair comparison with other algorithms, no extra local averaging is adopted in this experiment, i.e., the number of mixing rounds is set to $K = 1$. The loss function at each agent is given as $f_i(\boldsymbol{x}) = \frac{1}{2m}\|\boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i\|_2^2$. We plot the relative optimality gap, given as $(f(\bar{\boldsymbol{x}}^{(t)}) - f^\star)/f^\star$, where $\bar{\boldsymbol{x}}^{(t)}$ is the average parameter of all agents at the t-th iteration, and $f^\star$ is the optimal value. We compare the proposed `Network-DANE` (cf. Algorithm 1) and `Network-SVRG` (cf. Algorithm 3) with the server/client algorithms DANE [SSZ14] and ADMM [BPC+11],[6] and two popular network-distributed gradient descent algorithms, referred to as DGD [QL18] and EXTRA [SLWY15a].

Figure 2.1 shows the relative optimality gap with respect to the number of iterations as well as the number of gradient evaluations under different condition numbers $\kappa = 10$ and $\kappa = 10^4$ for linear regression. In both experiments, `Network-DANE` and `Network-SVRG` outperform DGD and EXTRA in terms of the numbers of communication rounds. `Network-SVRG` has similar communication rounds with ADMM but only communicates locally. `Network-DANE` is quite insensitive to the condition number, performing almost as well as the DANE algorithm in the ill-conditioned case, but operates in a fully decentralized setting. `Network-SVRG` further outperforms other algorithms in terms of gradient evaluations in most settings, especially for well-conditioned cases.

Benefits of extra local mixing (communication) per iteration. We conduct synthetic experiments to investigate the communication-computation trade-off observed in Theorem 4 when employing multiple rounds of mixing within every iteration. Following the suggestion of the theory, we use a poorly connected

[6] We apply ADMM to the constrained optimization problem, which amounts to the centrally-distributed setting, $\min_{\boldsymbol{x}_i} \frac{1}{n}\sum f_i(\boldsymbol{x}_i)$ s.t. $\boldsymbol{x}_i = \boldsymbol{x}$. Note that ADMM can also be applied to the network-distributed setting, which is not shown here since our network algorithms already outperform ADMM in the centrally-distributed setting.

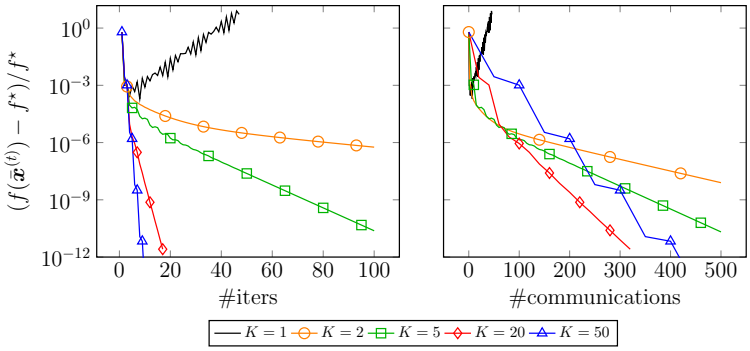


Figure 2.2: The relative optimality gap with respect to the number of iterations and communication rounds under different rounds of mixing K for `Network-DANE` over a poorly connected graph.

network with mixing rate $\alpha_0 = 0.944$ for communication, which is generated by an Erdős-Rényi graph with $p = 0.2$. For illustration, we consider the relative optimality gap for a linear regression problem with $\kappa = 10$, with respect to the number of iterations and communication rounds for `Network-DANE` and `Network-SVRG`, under different values of K (no Chebyshev acceleration is employed), shown in Figure 2.2. Due to poor connectivity, `Network-DANE` and `Network-SVRG` fail to converge when using moderate parameters. However, by using a larger K, due to improvement in consensus, `Network-DANE` converge faster in terms of the number of iterations. Notice that after certain threshold, further increasing K will not improve the convergence rate in terms of communication rounds.

Effects of local computation for `Network-SVRG`. We conduct an experiment to analyze the effect of different numbers of local stochastic iterations for `Network-SVRG`. Throughout this experiment, we run our algorithms on a linear regression problem with $\kappa = 10$ and Erdős-Rényi graph ($p = 0.2$) as the communication graph. Figure 2.3 shows the number of communication rounds and the number of gradient evaluations till converge for different numbers of local iterations. It is clear that with too few local iterations, `Network-SVRG` converges very slow and requires more communication. As soon as S is above a threshold, which is around $0.05m$ local iterations, the communication rounds no longer decrease. Therefore, in our experiments, we set the number of local iterations as $S = 0.05m$ to ensure satisfactory convergence rate while using an economical amount of local computation.

Effects of network topology. We conduct another experiment to compare the effect of network topology on linear regression problem with $\kappa = 10$. We generate communication graphs with different topology settings. Figure 2.4 shows the relative optimality gap with respect to the number of iterations and gradient evaluations for `Network-DANE` for Erdős-Rényi graph ($p = 0.3$), a 4×5 grid graph, a star graph, and a ring graph. The performance degrades as the network becomes less connected (where $1 - \alpha_0$ becomes small)

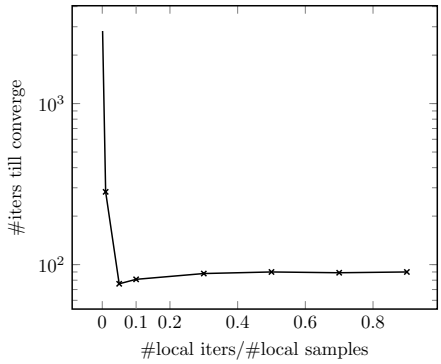


Figure 2.3: Number of communication rounds till converge with respect to different numbers of local iterations for `Network-SVRG`.

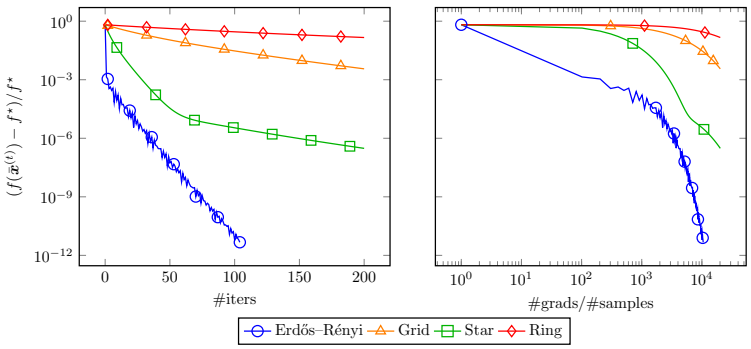


Figure 2.4: Performance of `Network-DANE` under different network topologies.

[NOR18].

Experiments for nonsmooth composite optimization We consider the ℓ_1-norm regularized linear regression, where the loss function of each agent is given as $\tilde{f}_i(\boldsymbol{x}) = f_i(\boldsymbol{x}) + g(\boldsymbol{x}) = \frac{1}{2m}\|\boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i\|_2^2 + 0.01\|\boldsymbol{x}\|_1$, and the communication graph are generated in the same way as Figure 2.1. The condition number κ is also defined in the same way as earlier. We compare the performance of `Network-DANE` with CEASE [FGW21], which is the proximal version of DANE in the server/client setting, ADMM, and PG-EXTRA, which is the proximal version of EXTRA [SLWY15b]. For CEASE and `Network-DANE`, we set $\mu = 10^{-4}$ when $\kappa = 10$ and $\mu = 10^{-1}$ when $\kappa = 10^4$, and use FISTA [BT09] to solve the ℓ_1-norm regularized local problems for computation efficiency. Figure 2.5 plots the relative optimality gap $\|\overline{\boldsymbol{x}}^{(t)} - \boldsymbol{x}^\star\|_2/\|\boldsymbol{x}^\star\|_2$ with respect to the number of iterations and the number of gradient evaluations for different algorithms under different condition numbers. In both experiments, `Network-DANE` outperformed ADMM and PG-EXTRA in both metrics, and achieves similar convergence behavior as CEASE, though at a slower rate due to optimizing over a decentralized topology.

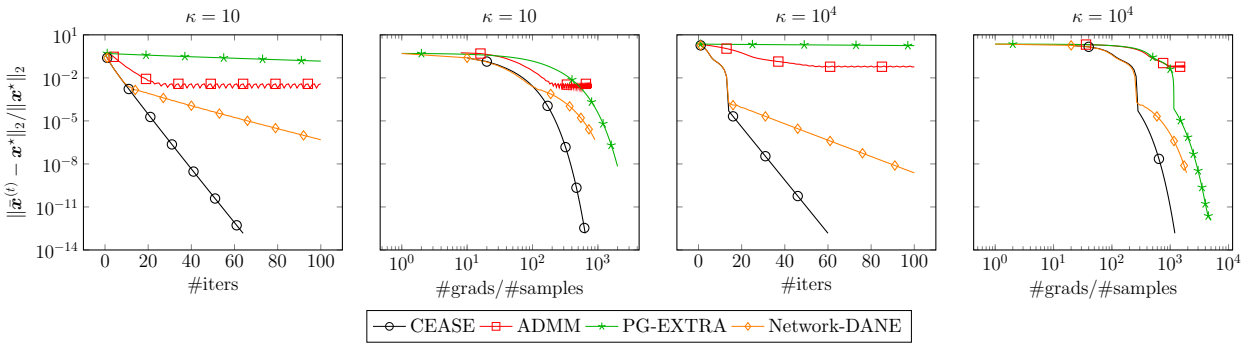


Figure 2.5: The relative optimality gap with respect to the number of iterations and gradient evaluations under different conditioning $\kappa = 10$ (left two panels) and $\kappa = 10^4$ (right two panels) for linear regression with ℓ_1-norm regularization.

2.6.2 Experiments on real data

We perform two experiments on real data to further evaluate the performance of the proposed algorithms for both convex and nonconvex problems.

Binary classification using logistic regression. We use regularized logistic regression to solve a binary classification problem using the Gisette dataset.[7] We split the Gisette dataset to $n = 20$ agents, where each agent receives $m = 300$ training samples of dimension $d = 5000$. The loss function at each agent is given as

$$f_i(\boldsymbol{x}) = \frac{1}{m}\sum_{j=1}^{m} \log\left(1 + (2b_i^{(j)} - 1)\exp(\boldsymbol{x}^\top \boldsymbol{a}_i^{(j)})\right) + \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2,$$

where $\boldsymbol{a}_i^{(j)} \in \mathbb{R}^d$ and $b_i^{(j)} \in \{0, 1\}$ are samples stored at agent i. For DANE and `Network-DANE`, we set $\mu = 5 \times 10^{-9}$ when $\kappa = 2$ and $\mu = 5 \times 10^{-1}$ when $\kappa = 100$. The condition number is controlled by changing the regularization λ. Figure 2.6 shows the results. In both cases, our algorithms exhibit compelling performance over other decentralized optimization algorithms especially in terms of communication efficiency.

1-hidden-layer neural network training. Though our theory only applies to the strongly convex case, we examine `Network-SVRG` in the nonconvex case, by training a one-hidden-layer neural network with 64 hidden neurons and sigmoid activations for a classification task using the MNIST dataset. We split $60,000$ training samples to 20 agents and use an Erdős-Rényi graph with $p = 0.3$ for communications. Figure 2.7 plots the training loss and testing accuracy against the number of iterations and gradient evaluations for different algorithms, where centralized ADMM and decentralized stochastic algorithm (DSGD) are plotted as baselines. Being more communication-efficient than DSGD, and more computation-efficient than ADMM, `Network-SVRG` reach a desirable balance between computation and communication efficacies.

[7] The dataset can be found at https://archive.ics.uci.edu/ml/datasets/Gisette.

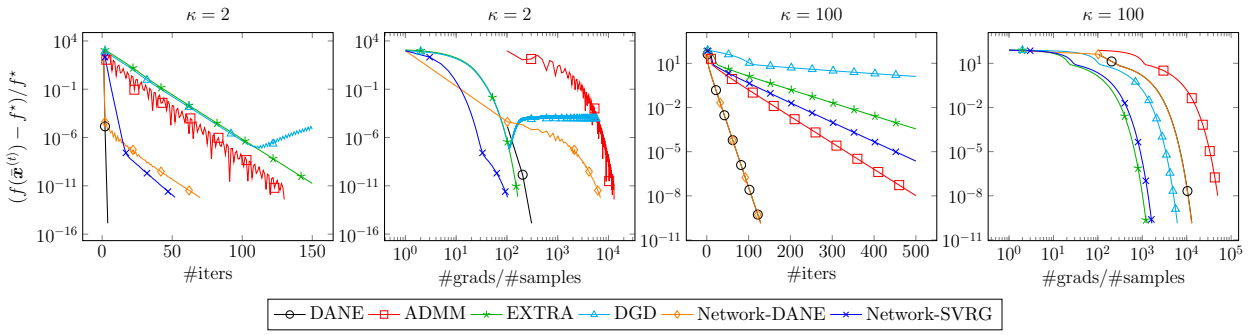


Figure 2.6: The relative optimality gap with respect to the number of iterations and gradient evaluations under different conditioning $\kappa = 2$ (left two panels) and $\kappa = 100$ (right two panels) for logistic regression using the Gisette dataset.

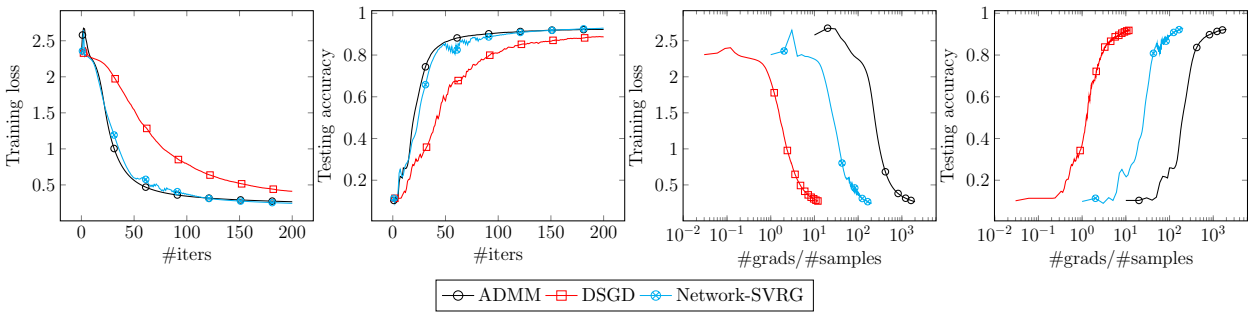


Figure 2.7: The training loss and testing accuracy with respect to the number of iterations (left two panels) and gradient evaluations (right two panels) for different algorithms on the MNIST dataset.

Chapter 3

Decentralized stochastic recursive gradient algorithm

Emerging applications in multi-agent environments such as internet-of-things, networked sensing, autonomous systems and federated learning, call for decentralized algorithms for finite-sum optimizations that are resource-efficient in terms of both computation and communication. In this chapter, we consider the prototypical setting where the agents work collaboratively to minimize the sum of local loss functions (cf. (1.3)) by only communicating with their neighbors over a predetermined network topology. We develop a new algorithm, called DEcentralized STochastic RECurSive gradient methodS (DESTRESS) for nonconvex finite-sum optimization, which matches the optimal incremental first-order oracle (IFO) complexity of centralized algorithms for finding first-order stationary points, while maintaining communication efficiency. Detailed theoretical and numerical comparisons corroborate that the resource efficiencies of DESTRESS improve upon prior decentralized algorithms over a wide range of parameter regimes. DESTRESS leverages several key algorithm design ideas including randomly activated stochastic recursive gradient updates with mini-batches for local computation, gradient tracking with extra mixing (i.e., multiple gossiping rounds) for per-iteration communication, together with careful choices of hyper-parameters and new analysis frameworks to provably achieve a desirable computation-communication trade-off.

This chapter is based on our previous publication [LLC22].

3.1 The DESTRESS algorithm

We propose DESTRESS for finding first-order order stationary points of nonconvex finite-sum problems. Throughout, we define (nd)-dimensional stacked vectors $\boldsymbol{u}^{(t)}$, $\boldsymbol{v}^{(t)}$, $\boldsymbol{g}^{(t)}$ and $\boldsymbol{s}^{(t)}$ analogously to Section 2.1.3. Motivated by stochastic recursive gradient methods in the centralized setting, DESTRESS has a nested loop

structure:

1. The inner loop refines the parameter estimate $\boldsymbol{u}^{(t),0} = \boldsymbol{x}^{(t-1)}$ by performing randomly activated stochastic recursive gradient updates (3.1), where the stochastic recursive gradient $\boldsymbol{v}^{(t),s}$ is updated in (3.1b) and (3.1c) via mixing mini-batch stochastic gradients from activated agents' local datasets.

2. The outer loop adopts dynamic average consensus to estimate and track the global gradient $\nabla F(\boldsymbol{x}^{(t)})$ at each agent by $\boldsymbol{s}^{(t)}$ in (3.2), which allows the next inner loop to start from a less noisy starting gradient $\boldsymbol{v}^{(t+1),0} = \boldsymbol{s}^{(t)}$. A key property of (3.2)—which is a direct consequence of dynamic average consensus—is that the average of $\boldsymbol{s}^{(t)}$ equals to the dynamic average of local gradients, i.e., $\bar{\boldsymbol{s}}^{(t)} = \frac{1}{n}\sum_{i\in[n]} \boldsymbol{s}_i^{(t)} = \frac{1}{n}\sum_{i\in[n]} \nabla f_i(\boldsymbol{x}_i^{(t)})$.

To enable better information sharing and faster convergence, inspired by [LCCC20], we allow DESTRESS to perform a few rounds of mixing or gossiping whenever communication takes place. Specifically, DESTRESS performs K_{out} and K_{in} mixing steps for the outer and inner loops respectively per iteration, which is equivalent to using

$$\boldsymbol{W}_{\mathsf{out}} = \boldsymbol{W}^{K_{\mathsf{out}}} \qquad \text{and} \qquad \boldsymbol{W}_{\mathsf{in}} = \boldsymbol{W}^{K_{\mathsf{in}}}$$

as mixing matrices, and correspondingly a network with better connectivity; see (3.2), (3.1a) and (3.1c). Note that Algorithm 4 is written in matrix notation, where the mixing steps are described by $\boldsymbol{W}_{\mathsf{in}} \otimes \boldsymbol{I}_n$ or $\boldsymbol{W}_{\mathsf{out}} \otimes \boldsymbol{I}_n$ and applied to all agents simultaneously. The extra mixing steps can be implemented by Chebyshev acceleration [AS14] with improved communication efficiency.

Compared with existing decentralized algorithms based on stochastic variance-reduced algorithms such as D-GET [SLH20] and GT-SARAH [XKK22a], DESTRESS utilizes different gradient estimators and communication protocols: First, DESTRESS produces a sequence of reference points $\{\boldsymbol{x}^{(t)}\}$ that converge to a global first-order stationary point and corresponding global gradient estimates $\{\boldsymbol{s}^{(t)}\}$ that are updated by full gradient computations, so that inner loops can refine $\boldsymbol{x}^{(t)}$ using stochastic recursive gradients based on accurate gradient estimates; second, the communication and computation in DESTRESS are paced differently due to the introduction of extra mixing, which allow a more flexible trade-off schemes between different types of resources; last but not least, the random activation of stochastic recursive gradient updates further saves local computation, especially when the local sample size is small compared to the number of agents.

[1]The stochastic gradients will not be computed if $\lambda_i^{(t),s} = 0$.

Algorithm 4 DESTRESS for decentralized nonconvex finite-sum optimization

input: initial parameter $\overline{\boldsymbol{x}}^{(0)}$, step size η, activation probability p, batch size b, number of outer loops T, number of inner loops S, and number of communication (extra mixing) steps K_{in} and K_{out}.

initialization: set $\boldsymbol{x}_i^{(0)} = \overline{\boldsymbol{x}}^{(0)}$ and $\boldsymbol{s}_i^{(0)} = \nabla f(\overline{\boldsymbol{x}}^{(0)})$ for all agents $1 \le i \le n$.

for $t = 1, \ldots, T$ **do**

  Set inner loop initial parameters $\boldsymbol{u}^{(t),0} = \boldsymbol{x}^{(t-1)}$ and $\boldsymbol{v}^{(t),0} = \boldsymbol{s}^{(t-1)}$.

  for $s = 1, ..., S$ **do**

    Each agent i samples a mini-batch $\mathcal{Z}_i^{(t),s}$ of size b from $\mathcal{M}_i$ uniformly at random, $\lambda_i^{(t),s} \sim \mathcal{B}(p)$ where $\mathcal{B}(p)$ denotes the Bernoulli distribution with parameter p,[1] and then performs the following updates:

$$\boldsymbol{u}^{(t),s} = (\boldsymbol{W}_{\mathsf{in}} \otimes \boldsymbol{I}_d)(\boldsymbol{u}^{(t),s-1} - \eta \boldsymbol{v}^{(t),s-1}), \tag{3.1a}$$

$$\boldsymbol{g}_i^{(t),s} = \frac{\lambda_i^{(t),s}}{pb} \sum_{z_i \in \mathcal{Z}_i^{(t),s}} \Big(\nabla \ell(\boldsymbol{u}_i^{(t),s}; \boldsymbol{z}_i) - \nabla \ell(\boldsymbol{u}_i^{(t),s-1}; \boldsymbol{z}_i) \Big) + \boldsymbol{v}_i^{(t),s-1}, \tag{3.1b}$$

$$\boldsymbol{v}^{(t),s} = (\boldsymbol{W}_{\mathsf{in}} \otimes \boldsymbol{I}_d) \boldsymbol{g}^{(t),s}. \tag{3.1c}$$

  Set the new parameter estimate $\boldsymbol{x}^{(t)} = \boldsymbol{u}^{(t),S}$.

  Update the global gradient estimate by aggregated local information and gradient tracking:

$$\boldsymbol{s}^{(t)} = (\boldsymbol{W}_{\mathsf{out}} \otimes \boldsymbol{I}_d) \Big(\boldsymbol{s}^{(t-1)} + \nabla F(\boldsymbol{x}^{(t)}) - \nabla F(\boldsymbol{x}^{(t-1)}) \Big) \tag{3.2}$$

output: $\boldsymbol{x}^{\mathsf{out}} \sim \mathrm{Uniform}(\{\boldsymbol{u}_i^{(t),s-1} | i \in [n], t \in [T], s \in [S]\})$.

3.2 Convergence guarantees

This section presents the performance guarantees of DESTRESS. Due to the nonconvexity, first-order algorithms are generally guaranteed to converge to only first-order stationary points of the global loss function $f(\cdot)$, which is defined in Definition 3.

Theorem 6 shows that DESTRESS converges in expectation to an approximate first-order stationary point, under suitable parameter choices.

Theorem 6 (First-order optimality). *Assume Assumptions 1 and 2 hold. Set $p \in (0, 1]$, K_{in}, K_{out}, S, b and η to be positive and satisfy*

$$\alpha^{K_{\mathsf{in}}} \le p \quad \text{and} \quad \eta L \le \frac{(1 - \alpha^{K_{\mathsf{in}}})^3 (1 - \alpha^{K_{\mathsf{out}}})}{10\big(1 + \alpha^{K_{\mathsf{in}}} \alpha^{K_{\mathsf{out}}} \sqrt{npb}\big)\big(\sqrt{S/(npb)} + 1\big)}. \tag{3.3}$$

The output produced by Algorithm 4 satisfies

$$\mathbb{E} \big\| \nabla f(\boldsymbol{x}^{\mathsf{out}}) \big\|_2^2 < \frac{4}{\eta T S} \Big(\mathbb{E}[f(\overline{\boldsymbol{x}}^{(0)})] - f^* \Big). \tag{3.4}$$

If there is only one agent, i.e., $n = 1$, the mixing rate will be $\alpha = 0$, we can choose $K_{\mathsf{in}} = K_{\mathsf{out}} = p = 1$, and Theorem 6 reduces to [NvP+22, Theorem 1], its counterpart in the centralized setting. For

general decentralized settings with arbitrary mixing schedules, Theorem 6 provides a comprehensive characterization of the convergence rate, where an ϵ-first-order stationary point can be found in expectation in a total of

$$TS = O\left(\frac{\mathbb{E}[f(\bar{\boldsymbol{x}}^{(0)})] - f^*}{\eta\epsilon^2}\right)$$

iterations, where T is the number of outer iterations and S is the number of inner iterations. Clearly, a larger step size η, as allowable by (3.3), hints on a smaller iteration complexity, and hence a smaller IFO complexity.

There are two conditions in (3.3). On one end, K_{in} needs to be large enough (i.e., perform more rounds of extra mixing) to counter the effect when p is small (i.e., we compute less stochastic gradients every iteration), or when α is close to 1 (i.e., the network is poorly connected). On the other end, the step size η needs to be small enough to account for the requirement of the step size in the centralized setting, as well as the effect of imperfect communication due to decentralization. For well connected networks where $\alpha \ll 1$, the terms introduced by the decentralized setting will diminish—indicating the iteration complexity is close to that of the centralized setting. For poorly connected networks, carefully designing the mixing matrix and other parameters can ensure a desirable trade-off between convergence speed and communication cost. The following corollary provides specific parameter choices for DESTRESS to achieve the optimal per-agent IFO complexity.

Corollary 3 (Complexity for finding first-order stationary points)**.** *Under conditions of Theorem 6, set* $S = \left\lceil \sqrt{mn} \right\rceil$, $b = \lceil \sqrt{m/n} \rceil$, $p = \frac{\sqrt{m/n}}{\lceil \sqrt{m/n} \rceil}$, $K_{\mathsf{out}} = \left\lceil \frac{\log(\sqrt{npb}+1)}{(1-\alpha)^{1/2}} \right\rceil$, $K_{\mathsf{in}} = \left\lceil \frac{\log(2/p)}{(1-\alpha)^{1/2}} \right\rceil$, $\eta = \frac{1}{640L}$, *and implement the mixing steps using Chebyshev's acceleration [AS14]. To reach an ϵ-first-order stationary point, in expectation,* DESTRESS *takes* $O\left(m + \frac{(m/n)^{1/2}L}{\epsilon^2}\right)$ *IFO calls per agent, and* $O\left(\frac{\log\left((n/m)^{1/2}+2\right)}{(1-\alpha)^{1/2}} \cdot \left((mn)^{1/2} + \frac{L}{\epsilon^2}\right)\right)$ *rounds of communication.*

DESTRESS achieves a network-independent IFO complexity that matches the optimal complexity in the centralized setting. In addition, when the accuracy $\epsilon^2 \lesssim L/(mn)^{1/2}$, DESTRESS reaches a communication complexity of $O(\frac{1}{(1-\alpha)^{1/2}} \cdot \frac{L}{\epsilon^2})$, which is independent of the sample size.

It is worthwhile to further highlight the role of the random activation probability p in achieving the optimal IFO by allowing "fractional" batch size. Note that the batch size is set as $b = \lceil \sqrt{m/n} \rceil$, where m is the local sample size, and n is the number of agents.

1. When the local sample size is large, i.e., $m \geq n$, we can approximate $b \approx \sqrt{m/n}$ and $p \approx 1$. In fact, Corollary 3 continues to hold with $p = 1$ in this regime.

	Erdős-Rényi graph	2-D grid graph	Path graph
$1-\alpha$ (spectral gap)	1	$\frac{1}{n\log n}$	$\frac{1}{n^2}$
D-GET [SLH20]	$m+\frac{m^{1/2}L}{\epsilon^2}$	$m+\frac{m^{1/2}n^2L}{\epsilon^2}$	$m+\frac{m^{1/2}n^4L}{\epsilon^2}$
GT-SARAH [XKK22a]	$m+\max\left\{1,\ (\frac{m}{n})^{1/3},\ (\frac{m}{n})^{1/2}\right\}\cdot\frac{L}{\epsilon^2}$	$m+\max\left\{n^2,\ m^{1/3}n^{2/3},\ (\frac{m}{n})^{1/2}\right\}\cdot\frac{L}{\epsilon^2}$	$m+\max\left\{n^4,\ m^{1/3}n^{5/3},\ (\frac{m}{n})^{1/2}\right\}\cdot\frac{L}{\epsilon^2}$
DESTRESS (Algorithm 4)	$(mn)^{1/2}+\frac{L}{\epsilon^2}$	$m^{1/2}n+\frac{n^{1/2}L}{\epsilon^2}$	$(mn^3)^{1/2}+\frac{nL}{\epsilon^2}$
Improvement factors for ϵ-independent term	$(\frac{m}{n})^{1/2}$	$\frac{m^{1/2}}{n}$	$\frac{m^{1/2}}{n^{3/2}}$
Improvement factors for ϵ-dependent term	$\max\left\{1,\ (\frac{m}{n})^{1/3},\ (\frac{m}{n})^{1/2}\right\}$	$\max\left\{n^{3/2},\ m^{1/3}n^{1/6},\ \frac{m^{1/2}}{n}\right\}$	$\max\left\{n^3,\ m^{1/3}n^{2/3},\ \frac{m^{1/2}}{n^{3/2}}\right\}$

Table 3.1: Detailed comparisons of the communication complexities of D-GET, GT-SARAH and DESTRESS under three graph topologies, where the last two rows delineate the improve factors of DESTRESS over existing algorithms. The communication savings become significant especially when $m=\Omega\big(\frac{n}{1-\alpha}\big)$. The complexities are simplified by plugging the bound on the spectral gap $1-\alpha$ from [NOR18, Proposition 5]. m, n, L are defined in Section 1.1 and α is the mixing rate defined in (1.4). The big-O (defined in Section 1.4) notation and logarithmic terms are omitted for simplicity.

2. However, when the number of agents is large, i.e., $n>m$, the batch size $b=1$ and $p=\sqrt{m/n}<1$, which mitigates the potential computation waste by only selecting a subset of agents to perform local computation, compared to the case when we naively set $p=1$.

Therefore, by introducing random activation, we can view $pb=\sqrt{m/n}$ as the effective batch size at each agent, which allows fractional values and leads to the optimal IFO complexity in all scenarios.

To gain further insights in terms of the communication savings of DESTRESS, Table 3.1 further compares the communication complexities of decentralized algorithms for finding first-order stationary points under three common network settings, which highlights the communication improvement over existing works.

3.3 Numerical experiments

This section provides numerical experiments on real datasets to evaluate our proposed algorithm DESTRESS with comparisons against two existing baselines: DSGD [NO09, LZZ+17] and GT-SARAH [XKK22a].

For all experiments, we shuffle the datasets and normalize the samples by subtracting the mean and dividing the standard deviation. We use the same number of agents, FDLA matrices, communication graphs and Chebyshev's acceleration scheme as in Section 2.6. In addition, since $m\gg n$ in all experiments, we set $p=1$ for simplicity. To ensure convergence, DSGD adopts a diminishing step size schedule. All parameters are tuned manually for the best performance. We defer detailed descriptions of baseline algorithms to Appendix B.1.

3.3.1 Logistic regression with nonconvex regularization

To begin with, we employ logistic regression with nonconvex regularization to solve a binary classification problem using the Gisette dataset.[2] We split the Gisette dataset to $n = 20$ agents, where each agent receives $m = 300$ training samples. The sample loss function is given as

$$\ell(\boldsymbol{x}; \{\boldsymbol{f}, l\}) = \log\left(1 + (2l-1)\exp(\boldsymbol{x}^\top \boldsymbol{f})\right) + \lambda \sum_{i=1}^{d} \frac{x_i^2}{1+x_i^2},$$

where $\{\boldsymbol{f}, l\}$ represents a training tuple, $\boldsymbol{f} \in \mathbb{R}^d$ is the feature vector and $l \in \{0, 1\}$ is the label, and λ is the regularization parameter. For this experiment, we set $\lambda = 0.1$. Table 3.2 specifies parameter setting we use for each graph.

Algorithms	DSGD		DESTRESS						GT-SARAH		
Parameters	η_0	b	η	p	K_{in}	K_{out}	b	S	η	b	S
Erdős-Rényi	1	10	0.01	1	2	2	10	10	0.001	10	10
Grid	1	10	0.01	1	2	3	10	10	0.001	10	10
Path	0.1	10	0.01	1	8	8	10	10	0.0001	10	10

Table 3.2: Parameter settings for the experiments on regularized logistic regression in Figure 3.1.

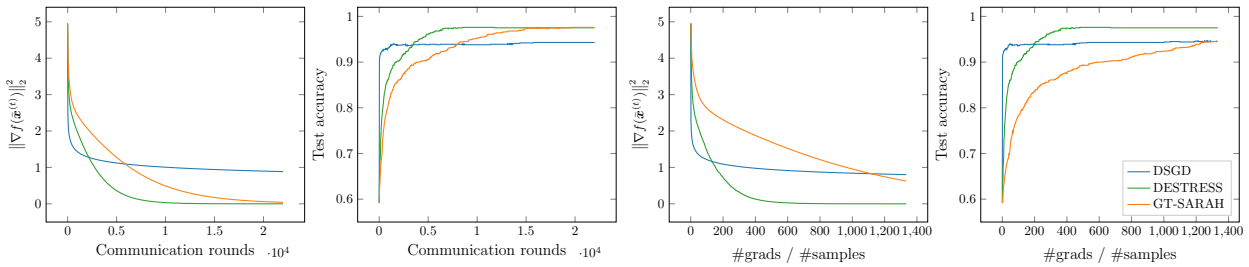
Figure 3.1 shows the train gradient norm and testing accuracy for all algorithms. DESTRESS significantly outperforms other algorithms both in terms of communication and computation. It is worth noting that, DSGD converges very fast at the beginning of training, but cannot sustain the progress due to the diminishing schedule of step sizes. On the contrary, the variance-reduced algorithms can converge with a constant step size, and hence converge better overall. Moreover, due to the refined gradient estimation and information mixing designs, DESTRESS can bear a larger step size than GT-SARAH, which leads to the fastest convergence and best overall performance. In addition, a larger number of extra mixing steps leads to a better performance when the communication graph is less connected.

3.3.2 One-hidden-layer neural network training

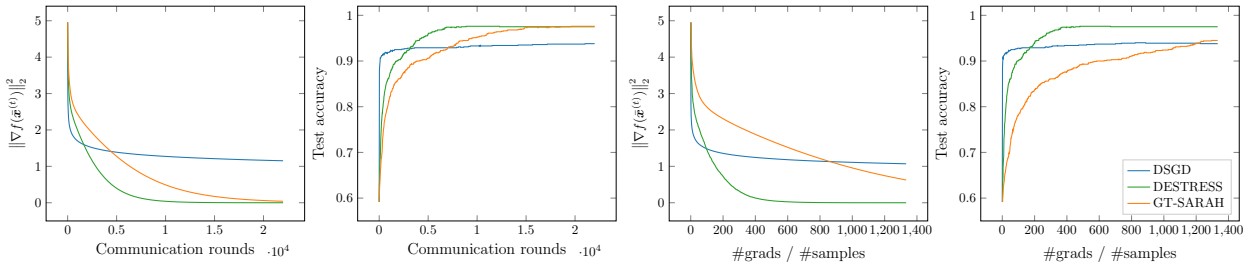
Next, we compare the performance of DESTRESS to DSGD and GT-SARAH for training a one-hidden-layer neural network with 64 hidden neurons and sigmoid activations for classifying the MNIST dataset [Den12]. We evenly split the MNIST dataset to $n = 20$ agents, where each agent receives $m = 3,000$ training samples. Table 3.3 specifies parameter setting we use for each graph.

Figure 3.2 plots the training gradient norm and testing accuracy against the number of communication rounds and gradient evaluations for all algorithms. DESTRESS significantly outperforms GT-SARAH in terms of computation and communication costs due to the larger step size and extra mixing. Different

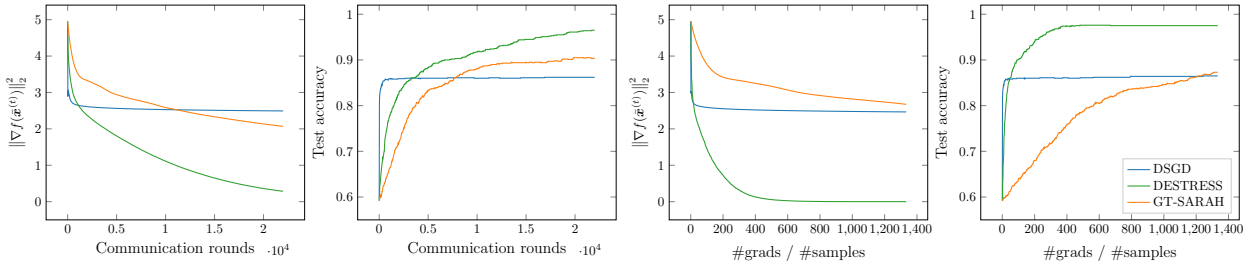
[2]The dataset can be accessed at https://archive.ics.uci.edu/ml/datasets/Gisette.



(a) Erdős-Rényi graph



(b) Grid graph



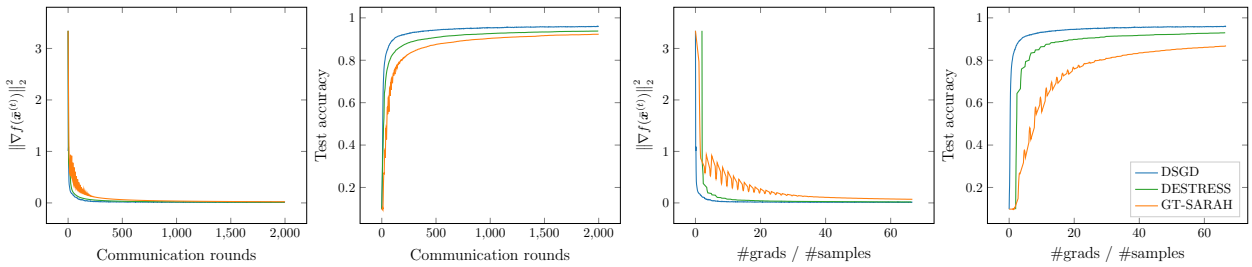
(c) Path graph

Figure 3.1: The train gradient norm and testing accuracy with respect to the number of communication rounds (left two panels) and gradient evaluations (right two panels) for DSGD, GT-SARAH and DESTRESS when training logistic regression model with nonconvex regularization on the Gisette dataset. Due to the initial full-gradient computation, the gradient evaluations of DESTRESS and GT-SARAH do not start from 0.

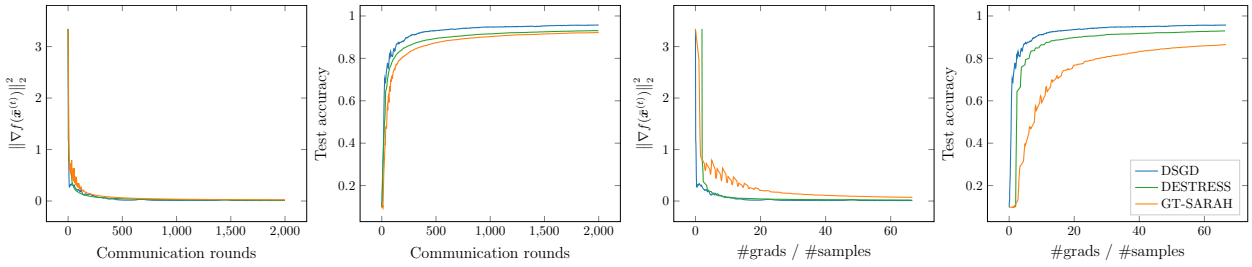
Algorithms	DSGD		DESTRESS						GT-SARAH		
Parameters	η_0	b	η	p	K_{in}	K_{out}	b	S	η	b	S
Erdős-Rényi	1	100	1	1	2	2	100	10	0.1	100	10
Grid	1	100	1	1	3	4	100	10	0.1	100	10
Path	0.1	100	1	1	8	10	100	10	0.0001	100	10

Table 3.3: Parameter settings for the experiments on neural network training in Figure 3.2.

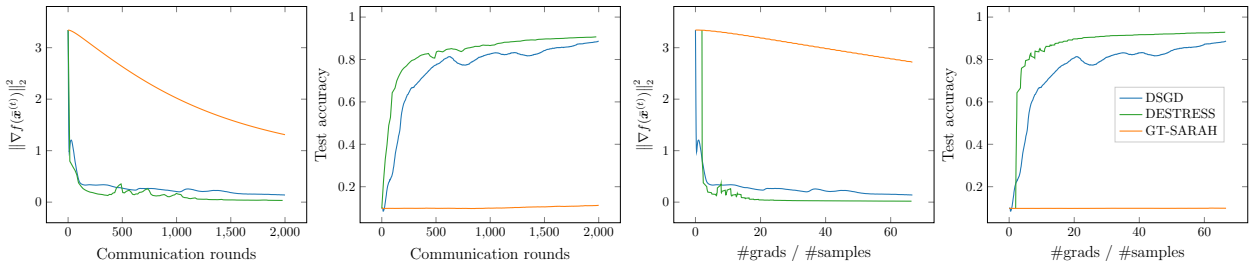
from the previous experiment, DSGD performs the best for Erdős-Rényi graph and grid graph that are well connected, while converges slower than DESTRESS on path graph.



(a) Erdős-Rényi graph



(b) Grid graph



(c) Path graph

Figure 3.2: The train gradient norm and testing accuracy with respect to the number of communication rounds (left two panels) and gradient evaluations (right two panels) for DSGD, GT-SARAH and DESTRESS when training a one-hidden-layer neural network on the MNIST dataset. Due to the initial full-gradient computation, the gradient evaluations of DESTRESS and GT-SARAH do not start from 0.

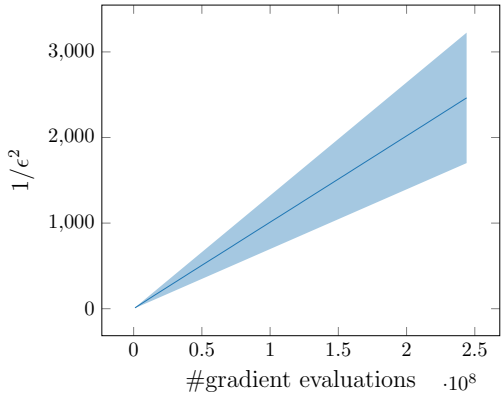


Figure 3.3: The convergence precision $1/\epsilon^2$ with respect to the number of total gradient evaluations for neural network training averaged over 64 experiments. The shade shows the variance.

3.3.3 Convergence and gradient computations

The last experiment investigates the convergence precision $1/\epsilon^2$ of DESTRESS with respect to the number of gradient evaluations. Under the same experimental setup, we conduct 64 different runs where each run starts from a different initial point. The convergence precision is computed by the inverse of the running average of the squared gradient norms. The results, including mean and variance, are shown in Figure 3.3, which numerically validate the linear relation indicated by Corollary 3.

Chapter 4

Decentralized stochastic algorithm with communication compression

Communication efficiency has been widely recognized as the bottleneck for large-scale decentralized machine learning applications in multi-agent or federated environments. To tackle the communication bottleneck, there have been many efforts to design algorithms with communication compression for decentralized nonconvex optimization (cf. (1.3)), where the clients are only allowed to communicate a small amount (commonly measured in bits) of quantized information with their neighbors over a predefined graph topology. Despite significant efforts, the state-of-the-art algorithm in the nonconvex setting still suffers from a slower rate of convergence $O(G/\epsilon^3)$ compared with their uncompressed counterpart to reach an ϵ-first-order stationary point for nonconvex objectives, where G measures the data heterogeneity across different clients. We propose `BEER`, which adopts communication compression with gradient tracking, and show it converges at a *faster rate* of $O(1/\epsilon^2)$. This significantly improves over the state-of-the-art rate, by matching the rate without compression even under arbitrary data heterogeneity. Numerical experiments are also provided to corroborate our theory and confirm the practical superiority of `BEER` in the data heterogeneous regime.

This chapter is based our previous publication [ZLL+22].

4.1 Preliminaries

4.1.1 Assumptions

We first introduce Assumptions 5 and 6, which are imposed on the global loss function. Assumption 5 only assumes global loss function is L-smooth, which makes it weaker than Assumption 2. The Polyak-

Łojasiewicz (PL) condition [Pol63] described in Assumption 6 can lead to linear convergence even when the function is nonconvex. It's worth noting that Assumption 6 is weaker than strong convexity.

Assumption 5 (L-smooth objective function). *A function $f(\boldsymbol{x})$ is L-smooth if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, the following inequality holds for some $L \geq 0$:*

$$\left\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\right\|_2 \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2.$$

Assumption 6 (PL condition). *A function $f(\boldsymbol{x})$ satisfies the Polyak-Łojasiewicz condition if $\forall \boldsymbol{x} \in \mathbb{R}^d$, the following inequality holds for some $\mu > 0$:*

$$\left\|\nabla f(\boldsymbol{x})\right\|_2^2 \geq 2\mu\big(f(\boldsymbol{x}) - f^\star\big),$$

where $f^\star = \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$.

Assumption 7 is a standard assumption for stochastic optimization algorithms, which implies a mini-batch stochastic gradient of batch size b has a variance of σ^2/b.

Assumption 7 (Bounded local variance). *The variance of a local stochastic gradient at a uniformly randomly selected data sample $\boldsymbol{z}_i$ is bounded if $\forall i \in [n]$ and $\forall \boldsymbol{x} \in \mathbb{R}^d$, the following inequality holds:*

$$\mathbb{E}_{\boldsymbol{z}_i\sim\mathcal{M}_i}\left\|\nabla \ell(\boldsymbol{x}; \boldsymbol{z}_i) - \nabla f_i(\boldsymbol{x})\right\|_2^2 \leq \sigma^2.$$

4.1.2 Compression operators

Using compression operators to compress gradients or any data needed to be communicated can substantially improve communication efficiency [TGZ+18, SCJ18, KSJ19, RSF21, FSG+21].

Definition 5 defines a randomized general compression operator that only guarantees the expected compression error $\mathbb{E}\left\|\mathcal{C}(\boldsymbol{x}) - \boldsymbol{x}\right\|_2^2$ is less than the magnitude of original message $\|\boldsymbol{x}\|_2^2$. If the compression operator is unbiased, it falls into the unbiased compression operator category [AGL+17, KFJ18, MGTR19, LR20] defined in Definition 5.

Definition 5 (General compression operator). *A randomized map $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ is a ρ-compression operator if $\forall \boldsymbol{x} \in \mathbb{R}^d$ and some $\rho \in [0, 1]$, the following inequality holds:*

$$\mathbb{E}\left\|\mathcal{C}(\boldsymbol{x}) - \boldsymbol{x}\right\|_2^2 \leq (1-\rho)\|\boldsymbol{x}\|_2^2.$$

Definition 6 (Unbiased compression operator). *A randomized map $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ is an ω-unbiased compression operator if $\forall \boldsymbol{x} \in \mathbb{R}^d$, and there exists some $\omega > 0$, the followings hold:*

$$\mathbb{E}[\mathcal{C}(\boldsymbol{x})] = \boldsymbol{x} \quad \textit{and} \quad \mathbb{E}\left\|\mathcal{C}(\boldsymbol{x}) - \boldsymbol{x}\right\|_2^2 \leq \omega\|\boldsymbol{x}\|_2^2.$$

It is possible to convert a general (biased) compression operator to an unbiased one. Example 1 shows an example that random sparsification can be converted between these two categories by applying a constant scaling. Examples 2 and 3 are two examples of general compression operators.

Example 1 (Random sparsification). *Random sparsification keeps an element from a d-dimensional vector with probability* $\frac{k}{d}$. *Let* $\boldsymbol{u} \in \mathbb{R}^d$ *where* $u_i \sim B\left(\frac{k}{d}\right)$.

- *Biased random sparsification is defined as* $\mathrm{random}_{k,biased}(\boldsymbol{x}) = \boldsymbol{u} \odot \boldsymbol{x}$, *which satisfies Definition 5 with* $\rho = \frac{k}{d}$.
- *Unbiased random sparsification is defined as* $\mathrm{random}_{k,unbiased}(\boldsymbol{x}) = \frac{d}{k}\boldsymbol{u} \odot \boldsymbol{x}$, *which satisfies Definition 6 with* $\omega = \frac{d}{k} - 1$.

Example 2 (Random dithering). *Random dithering [AGL+17] quantizes the message to b bits after adding random noise, which is defined as*

$$\mathrm{gsgd}_b(\boldsymbol{x}) := \frac{\|\boldsymbol{x}\|_2}{\tau} \cdot \mathrm{sign}(\boldsymbol{x}) \cdot 2^{-(b-1)} \cdot \left\lfloor \frac{2^{(b-1)}|\boldsymbol{x}|}{\|\boldsymbol{x}\|_2} + \boldsymbol{u} \right\rfloor$$

where $\tau = 1 + \min\left\{\frac{d}{2^{2(b-1)}}, \frac{\sqrt{d}}{2^{(b-1)}}\right\}$, *and* $\boldsymbol{u}$ *is the random dithering vector uniformly sampled from* $[0,1]^d$. gsgd_b *satisfies Definition 5 with* $\rho = 1/\tau$.

Example 3 (top_k). top_k *[AHJ+18, SCJ18] keeps k elements that have the largest absolute values and sets other elements to* 0, *which is defined as*

$$\mathrm{top}_k(\boldsymbol{x}) := \boldsymbol{x} \odot \boldsymbol{u}(\boldsymbol{x}),$$

where $[\boldsymbol{u}(\boldsymbol{x})]_i = 1$ *if the absolute value of the i-th element is one of the k-largest absolute values, otherwise* $[\boldsymbol{u}(\boldsymbol{x})]_i = 0$. top_k *satisfies Definition 5 with* $\rho = k/d$.

4.1.3 Additional notation

We denote a gradient of local objective function as $\widetilde{\nabla} f_i(\boldsymbol{x}) := \nabla \ell(\boldsymbol{x}; \boldsymbol{z}_i)$, where $\boldsymbol{z}_i \in \mathcal{Z}_i$ is uniformly randomly sampled from local dataset $\mathcal{Z}_i$. In addition, for a batch of b i.i.d. uniform random samples $\{\boldsymbol{z}_{i,k} \in \mathcal{Z}_i\}_{k\in[b]}$, the mini-batch stochastic gradient is defined as $\widetilde{\nabla}_b f_i(\boldsymbol{x}) := \frac{1}{b}\sum_{k=1}^b \nabla \ell(\boldsymbol{x}; \boldsymbol{z}_{i,k})$.

We define the distributed stochastic gradient and mini-batch stochastic gradient $\widetilde{\nabla} F(\boldsymbol{X})$, $\widetilde{\nabla}_b F(\boldsymbol{X})$, and distributed compression operator $\mathcal{C}(\boldsymbol{X})$ analogously to (1.7).

4.2 The BEER algorithm

This section presents BEER (cf. Algorithm 5) for decentralized nonconvex optimization with compressed communication.

At the t-th iteration, BEER maintains the current model estimates $\boldsymbol{X}^{(t)}$ and the global gradient estimates $\boldsymbol{V}^{(t)}$ across the clients. At the crux of its design, BEER also tracks and maintains two control sequences $\boldsymbol{H}^{(t)}$ and $\boldsymbol{G}^{(t)}$ that serve as compressed surrogates of $\boldsymbol{X}^{(t)}$ and $\boldsymbol{V}^{(t)}$, respectively. In particular, these two control sequences are updated by aggregating the received compressed messages alone (cf. Line 5 and Line 8).

It then boils down to how to carefully update these quantities in each iteration with communication compression. To begin, note that for each client i, BEER not only maintains its own parameters $\{\boldsymbol{x}_i^{(t)}, \boldsymbol{v}_i^{(t)}, \boldsymbol{h}_i^{(t)}, \boldsymbol{g}_i^{(t)}\}$, but also the control variables from its neighbors, namely, $\{\boldsymbol{h}_j^{(t)}\}_{j\in\mathcal{N}(i)}$ and $\{\boldsymbol{g}_j^{(t)}\}_{j\in\mathcal{N}(i)}$.

Each iteration can be roughly broken into three steps.

- **Update model estimate:** Each agent i first updates its model $\boldsymbol{x}_i^{(t+1)}$ according to Line 3, by a gradient-style update with a correction term using compressed surrogates of models, i.e., $\{\boldsymbol{h}_j^{(t)}\}_{j\in\mathcal{N}(i)}$. This update step incorporates aggregated information and compensates for the compression errors, thus leads to better consensus among agents and improved communication efficiency.

- **Update global gradient estimate:** Each client i updates the global gradient estimate $\boldsymbol{v}_i^{(t+1)}$ according to Line 6, where the last correction term—based on the difference of the gradients at consecutive models—is known as a trick called *gradient tracking* [QL18, DLS16, NOS17]. The use of gradient tracking is critical: as shall be seen momentarily, it contributes to the key difference from CHOCO-SGD that enables the fast rate of $O(1/\epsilon^2)$ without any bounded dissimilarity or bounded gradient assumptions. Indeed, if we remove the control sequence $\boldsymbol{G}^{(t)}$ and substitute Lines 6-8 by $\boldsymbol{V}^{(t+1)} = \tilde{\nabla}_b F(\boldsymbol{X}^{(t+1)})$, we recover CHOCO-SGD from BEER.

- **Update compressed surrogates with communication:** To update $\{\boldsymbol{h}_j^{(t)}\}_{j\in\mathcal{N}(i)}$, each client i first computes a compressed message $\boldsymbol{q}_{h,i}^{(t+1)}$ that encodes the difference $\boldsymbol{x}_i^{(t+1)} - \boldsymbol{h}_i^{(t)}$, and broadcasts to its neighbors (cf. Line 4). Then, each client i updates $\{\boldsymbol{h}_j^{(t)}\}_{j\in\mathcal{N}(i)}$ by aggregating the received compressed messages $\{\boldsymbol{q}_{h,j}^{(t+1)}\}_{j\in\mathcal{N}(i)}$ following Line 5. The updates of $\{\boldsymbol{g}_j^{(t)}\}_{j\in\mathcal{N}(i)}$ can be performed similarly. Moreover, all the compressed messages can be sent in a single communication round at one iteration, i.e., the communications in Lines 4 and 7 can be performed at once. This leverages EF21 [RSF21] for communication compression, which is a *better and simpler* algorithm that deals with biased compression operators compared with the error feedback (or error compensation, EF/EC) framework [KRSJ19, SK20]. Using the control sequence $\boldsymbol{G}^{(t)}$, BEER does not need to apply EF/EC explicitly and can deal with the error implicitly.

Algorithm 5 BEER: Better comprEssion for dEcentRalized optimization

Input: Initial point $\boldsymbol{X}^{(0)} = \boldsymbol{x}^{(0)}\mathbf{1}^\top$, $\boldsymbol{G}^{(0)} = \boldsymbol{H}^{(0)} = \mathbf{0}_{d\times n}$, $\boldsymbol{V}^{(0)} = \nabla F(\boldsymbol{X}^{(0)})$, step size η, mixing step size γ, minibatch size b.

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

$\boldsymbol{X}^{(t+1)} = \boldsymbol{X}^{(t)} + \gamma \boldsymbol{H}^{(t)}(\boldsymbol{W} - \boldsymbol{I}) - \eta \boldsymbol{V}^{(t)}$

$\boldsymbol{Q}_h^{(t+1)} = \mathcal{C}(\boldsymbol{X}^{(t+1)} - \boldsymbol{H}^{(t)})$ ▷ *agent i sends $\boldsymbol{q}_{h,i}^{(t+1)}$ to all its neighbors*

$\boldsymbol{H}^{(t+1)} = \boldsymbol{H}^{(t)} + \boldsymbol{Q}_h^{(t+1)}$

$\boldsymbol{V}^{(t+1)} = \boldsymbol{V}^{(t)} + \gamma \boldsymbol{G}^{(t)}(\boldsymbol{W} - \boldsymbol{I}) + \widetilde{\nabla}_b F(\boldsymbol{X}^{(t+1)}) - \widetilde{\nabla}_b F(\boldsymbol{X}^{(t)})$

$\boldsymbol{Q}_g^{(t+1)} = \mathcal{C}(\boldsymbol{V}^{(t+1)} - \boldsymbol{G}^{(t)})$ ▷ *agent i sends $\boldsymbol{q}_{g,i}^{(t+1)}$ to all its neighbors*

$\boldsymbol{G}^{(t+1)} = \boldsymbol{G}^{(t)} + \boldsymbol{Q}_g^{(t+1)}$

4.3 Convergence guarantees

This section presents convergence guarantees of BEER for nonconvex objective functions (cf. Section 4.3.1) and PL objective functions (cf. Section 4.3.2). The convergence analysis is based on a Lyapunov function tailored for BEER, given by

$$\Phi^{(t)} = \mathbb{E}[f(\bar{\boldsymbol{x}}^{(t)})] - f^\star + \frac{c_1 L}{n}\Omega_1^{(t)} + \frac{c_2(1-\alpha)^2}{nL}\Omega_2^{(t)} + \frac{c_3 L}{n}\Omega_3^{(t)} + \frac{c_4(1-\alpha)^4}{nL}\Omega_4^{(t)}, \tag{4.1}$$

where $\{c_i\}$ are some constants depend on specific settings, $\mathbb{E}[f(\bar{\boldsymbol{x}}^{(t)})] - f^\star$ represents the sub-optimality gap, and the errors $\{\Omega_i^{(t)}\}$ are defined by

$$\begin{aligned} \Omega_1^{(t)} &:= \mathbb{E}\big\|\boldsymbol{H}^{(t)} - \boldsymbol{X}^{(t)}\big\|_{\mathsf{F}}^2, \qquad \Omega_2^{(t)} := \mathbb{E}\big\|\boldsymbol{G}^{(t)} - \boldsymbol{V}^{(t)}\big\|_{\mathsf{F}}^2, \\ \Omega_3^{(t)} &:= \mathbb{E}\big\|\boldsymbol{X}^{(t)} - \bar{\boldsymbol{x}}^{(t)}\mathbf{1}^\top\big\|_{\mathsf{F}}^2, \qquad \Omega_4^{(t)} := \mathbb{E}\big\|\boldsymbol{V}^{(t)} - \bar{\boldsymbol{v}}^{(t)}\mathbf{1}^\top\big\|_{\mathsf{F}}^2. \end{aligned} \tag{4.2}$$

Here, $\Omega_1^{(t)}$ and $\Omega_2^{(t)}$ denote the compression errors for $\boldsymbol{X}^{(t)}$ and $\boldsymbol{V}^{(t)}$ when approximated using the compressed surrogates $\boldsymbol{H}^{(t)}$ and $\boldsymbol{G}^{(t)}$, respectively, and $\Omega_3^{(t)}$ and $\Omega_4^{(t)}$ denote the consensus errors of $\boldsymbol{X}^{(t)}$ and $\boldsymbol{V}^{(t)}$.

4.3.1 Convergence for nonconvex objective functions

Theorem 7 presents the convergence results of BEER for nonconvex objective functions

Theorem 7 (Convergence in the nonconvex setting)**.** *Suppose Assumptions 1, 5 and 7 hold. There exist absolute constants $c_1, c_2, c_3, c_4, c_\gamma, c_\eta > 0$, such that if we set $\gamma = c_\gamma \rho(1-\alpha)$, $\eta = c_\eta \gamma (1-\alpha)^2/L$, for the Lyapunov function $\Phi^{(t)}$ defined in (4.1), it holds*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2^2 \le \frac{2(\Phi^{(0)} - \Phi^{(T)})}{\eta T} + \frac{36 c_4 \sigma^2}{c_\gamma b \rho L}.$$

Theorem 7 shows the convergence results for BEER.

Full gradient case When we can access the full gradient, which is equivalent to $\sigma^2 = 0$, BEER converges at a rate of $O(\epsilon^{-2})$. This rate is faster than the $O(\epsilon^{-3})$ rate by CHOCO-SGD [KSJ19] and DeepSqueeze [TLQ+19], and the $O(\epsilon^{-4})$ rate by SQuARM-SGD [SDGD21].

More specifically, to achieve an ϵ-first-order stationary point, BEER needs

$$O\left(\frac{1}{(1-\alpha)^3\rho\epsilon^2}\right)$$

communication rounds, where α and ρ are the mixing rate (cf. Definition 1) and the compression parameter (cf. Definition 5), respectively. In comparison, the state-of-the-art algorithm CHOCO-SGD [KSJ19] converges in $O\left(\frac{G}{(1-\alpha)^2\rho\epsilon^3}\right)$ communication rounds, with G being the bounded gradient parameter, namely, $\mathbb{E}_{z_i\sim\mathcal{M}_i}\|\nabla\ell(\boldsymbol{x};\boldsymbol{z}_i)\|^2 \leq G^2$. Therefore, BEER improves over CHOCO-SGD not only in terms of a better dependency on ϵ, but also removing the bounded gradient assumption, which is significant since in practice, G can be excessively large due to data heterogeneity across the clients.

The dependency on ρ of BEER is consistent with other compression schemes, such as CHOCO-SGD, DeepSqueeze and SQuARM-SGD for the nonconvex setting, as well as LEAD [KKJ+21] and EF-C-GT [LLHP22] for the strongly convex setting.

As for the dependency on $(1-\alpha)^{-1}$, BEER is slightly worse than CHOCO-SGD, where CHOCO-SGD has a dependency of $O\big((1-\alpha)^{-2}\big)$ whereas BEER has a dependency of $O((1-\alpha)^{-3})$. This degeneration is also seen in the analysis of uncompressed decentralized algorithms using gradient tracking [SLH20, XKK22b], where the rate $O((1-\alpha)^{-2})$ is worse than the rate of $O((1-\alpha)^{-1})$ for basic decentralized SGD algorithms [KDG03, LZZ+17] by a factor of $1-\alpha$. In addition, both BEER and CHOCO-SGD use small mixing step size γ to guarantee convergence, which makes the dependency on $(1-\alpha)^{-1}$ worse than their uncompressed counterparts.

Stochastic gradient case In the presence of local variance, the squared gradient norm of BEER has an additional term that scales on the order of $O\left(\frac{\sigma^2}{\rho b}\right)$ (ignoring other parameters). By choosing a sufficiently large minibatch size b, i.e., $b \geq O\left(\frac{\sigma^2}{\rho\epsilon^2}\right)$, BEER maintains the iteration complexity

$$O\left(\frac{1}{(1-\alpha)^3\rho\epsilon^2}\right)$$

to reach an ϵ-first-order stationary point, without the bounded gradient assumption, thus inheriting similar advantages over CHOCO-SGD as discussed earlier. In terms of local computation, the gradient oracle complexity on a single client of BEER is

$$O\left(\frac{1}{(1-\alpha)^3\rho\epsilon^2} + \frac{\sigma^2}{(1-\alpha)^3\rho^2\epsilon^4}\right).$$

While our focus is on communication efficiency, to gain more insights, Table 1.4 summarizes both the communication rounds and the gradient complexity for different decentralized stochastic methods. While BEER does not require the bounded gradient assumption, it may lead to a worse gradient complexity in the data homogeneous setting due to the use of large minibatch size. Fortunately, this only impacts the local computation cost, and does not exacerbate the communication complexity, which is often the bottleneck. It is of great interest to further refine the design and analysis of BEER in terms of the gradient complexity.

4.3.2 Convergence for PL objectives functions

Theorem 8 (Convergence under PL condition)**.** *Suppose Assumptions 1 and 5 to 7 hold. There exist absolute constants* $c_1, c_2, c_3, c_4, c_\gamma, c_\eta > 0$*, such that if we set* $\gamma = c_\gamma \rho(1-\alpha)$*,* $\eta = c_\eta \gamma (1-\alpha)^2/L$*, for the Lyapunov function* $\Phi^{(t)}$ *in* (4.1)*, it holds*

$$\Phi^{(T)} \leq (1-\mu\eta)^T \Phi^{(0)} + \frac{36 c_4 \sigma^2}{c_\gamma L b \rho}.$$

Theorem 8 shows the convergence guarantees for BEER under the PL condition. If we can access full local gradients ($\sigma = 0$), BEER converges linearly to the global optimum $f^\star$ at a rate of $O\big(\kappa \log(1/\epsilon)\big)$. When using stochastic gradients, BEER converges linearly to a neighborhood of size $O\big(\frac{\sigma^2}{\rho b}\big)$ around the global optimum.

4.4 Numerical experiments

This section presents numerical experiments on real-world datasets to showcase BEER's superior ability to handle data heterogeneity over agents, by running each experiment on unshuffled datasets and comparing the performances with the state-of-the-art baseline algorithms both with and without communication compression.

For all experiments, we split unshuffled datasets evenly to 10 agents that are connected by a ring topology, so that we can simulate the scenario with high data heterogeneity across agents. Approximately, for the a9a dataset, 5 agents receive data with label 1 and others receive data with label 0; for the MNIST dataset, agent i receives data with label i. We use the FDLA matrix [XB04] as the mixing matrix to perform weighted information aggregation to accelerate convergence. For each experiment, all algorithms are initialized to the same staring point, and uses best-tuned learning rates and batch sizes.

Sections 4.4.1 and 4.4.2 compares BEER with 1) CHOCO-SGD [KSJ19], which is the state-of-the-art nonconvex decentralized optimizing algorithm using communication compression, and 2) DSGD [LZZ+17] and D^2 [TLY+18], which are decentralized optimization algorithms without compression. Section 4.4.3 further evaluates the impact of communication network and compression operators.

4.4.1 Logistic regression with nonconvex regularization

We first run experiments on logistic regression with a nonconvex regularizer [WJZ+19] on the a9a dataset [CL11]. Similar to Section 3.3.1, following [WJZ+19], the objective function over a datum $(\boldsymbol{a}, b)$ is defined as

$$\ell(\boldsymbol{x}; \{\boldsymbol{f}, l\})) = \log\left(1 + l \exp(-\boldsymbol{x}^\top \boldsymbol{f})\right) + \lambda \sum_{i=1}^{d} \frac{x_i^2}{1 + x_i^2},$$

where $\{\boldsymbol{f}, l\}$ represents a training tuple, $\boldsymbol{f} \in \mathbb{R}^d$ is the feature vector and $l \in \{0, 1\}$ is the label, and λ is the regularization parameter which is set to $\lambda = 0.05$.

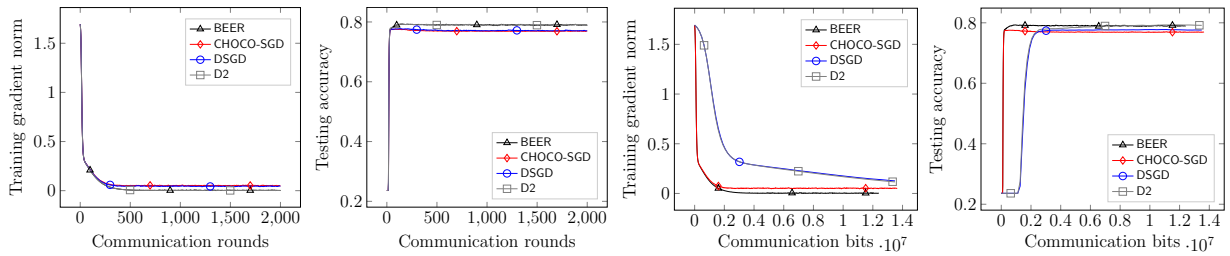


Figure 4.1: The training gradient norm and testing accuracy against communication rounds (left two panels) and communication bits (right two panels) for logistic regression with nonconvex regularization on unshuffled a9a dataset. Both BEER and CHOCO-SGD employ the biased gsgd_5 compression [AGL+17].

Figure 4.1 plots the training gradient norm and testing accuracy against communication rounds and communication bits for logistic regression with nonconvex regularization. The algorithms with communication compression (BEER and CHOCO-SGD [KSJ19]) converge faster than the uncompressed algorithms (DSGD [LZZ+17] and D^2 [TLY+18]) in terms of the communication bits. However, CHOCO-SGD fails to converge to a small gradient norm due to its inability to tolerate a high level of data dissimilarity across different agents. In contrast, BEER and D^2 converge to the smallest gradient norm, while BEER outperforms D^2 in terms of communication bits. The results for testing accuracy are similar, where BEER achieves the best testing accuracy and is the fastest.

4.4.2 One-hidden-layer neural network training

Similar to Section 3.3.2, we evaluate BEER by training a 1-hidden layer neural network on the MNIST dataset [LJB+95]. The network uses 64 hidden neurons, sigmoid activation functions and cross-entropy loss, where the loss function over a training pair $\{\boldsymbol{f}, l\}$ is defined as

$$\ell(\boldsymbol{x}; (\boldsymbol{f}, l)) = \mathsf{CrossEntropy}(\mathsf{softmax}(\boldsymbol{W}_2\ \mathsf{sigmoid}(\boldsymbol{W}_1 \boldsymbol{f} + \boldsymbol{c}_1) + \boldsymbol{c}_2), l).$$

Here the model parameter is defined by $\boldsymbol{x} = \text{vec}(\boldsymbol{W}_1, \boldsymbol{c}_1, \boldsymbol{W}_2, \boldsymbol{c}_2)$, where the dimensions of the network parameters $\boldsymbol{W}_1$, $\boldsymbol{c}_1$, $\boldsymbol{W}_2$, $\boldsymbol{c}_2$ are 64×784, 64×1, 10×64, and 10×1, respectively.

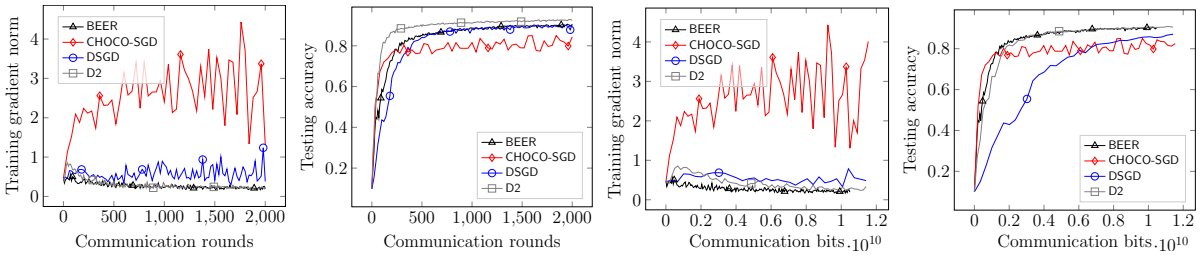


Figure 4.2: The training gradient norm and testing accuracy against communication rounds (left two panels) and communication bits (right two panels) for classification on unshuffled MNIST dataset using a 1-hidden-layer neural network. Both BEER and CHOCO-SGD employ the biased gsgd_{20} compression [AGL+17].

Figure 4.2 plots the training gradient norm and testing accuracy against communication rounds and communication bits for 1-hidden-layer neural network training. In terms of the final training gradient norm, BEER converges to a solution comparable to D^2 but at a lower communication cost, while CHOCO-SGD and DSGD cannot converge due to the high data heterogeneity. In terms of testing accuracy, BEER and D^2 have very similar performance, and outperform CHOCO-SGD and DSGD.

4.4.3 Network topology and compression operators

We further investigate the impact of communication network topology and compression operators on the performance of BEER. We follow the same setup as Section 4.4.1 to run logistic regression with nonconvex regularization ($\lambda = 0.05$) on the unshuffled a9a dataset, by splitting it evenly to 40 agents. All experiments use the same best-tuned step size $\eta = 0.5$, batch size $b = 100$ and $\gamma = 0.7$.

Impacts of network topology Figure 4.3 shows the training gradient norm and testing accuracy of BEER with respect to the communication rounds over different network topologies using the gsgd_5 compression [AGL+17]. Experimented topologies are ring topology ($\alpha = 0.978$), star topology ($\alpha = 0.951$), grid topology ($\alpha = 0.937$), and Erdős-Rényi topology with connectivity probability $p = 0.5$ and $p = 0.2$ ($\alpha = 0.49$ and $\alpha = 0.23$, respectively). Despite the huge differences in mixing rates, BEER can use nearly the same hyper-parameters to obtain similar performance. The experiments complement our theoretical analysis and show that BEER may converge way better in practice despite its cubic dependency of $(1-\alpha)^{-1}$ in Theorem 7.

Impacts of compression schemes Figure 4.4 shows the training gradient norm and testing accuracy of BEER with respect to the communication rounds and communication bits on a ring topology using different compression schemes: no compression, gsgd_5 (cf. Example 2) and top_{10} (cf. Example 3). The

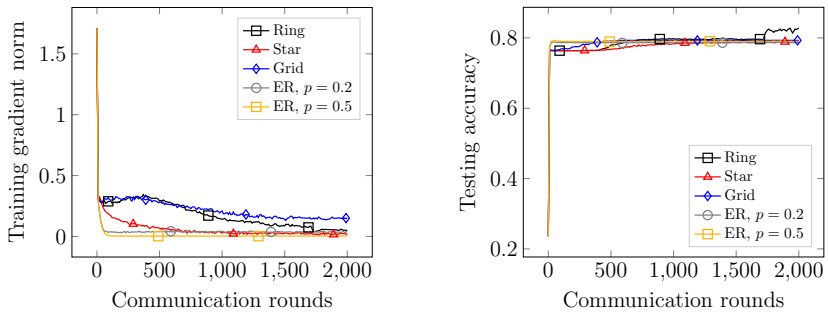


Figure 4.3: The training gradient norm and testing accuracy against communication rounds for BEER using the biased gsgd_5 compression [AGL+17] for logistic regression with nonconvex regularization on unshuffled a9a dataset.

compression parameters are chosen such that each compression operator transfers similar number of bits per communication round. All experiments use the same best-tuned step size $\eta = 0.5$, batch size $b = 100$ and $\gamma = 0.7$, except that we use $\eta = 0.005$ and $\gamma = 0.8$ for top_{10} compression.

In terms of communication bits, using compression operators improves over the uncompressed baseline, because all algorithms with compression converge to a solution with lower gradient norm and higher testing accuracy at a lower communication cost. In terms of communication rounds and testing accuracy, different compression operators can lead to significantly behaviors. For example, using gsgd_5 compression operator leads to faster converges than without compression, but using the top_{10} compression operator leads to slower converges than without compression. In sum, BEER with gsgd_5 reaches the highest final testing accuracy while behaves similar to BEER without compression in terms of communication rounds, which indicates the benefit of using communication compression.

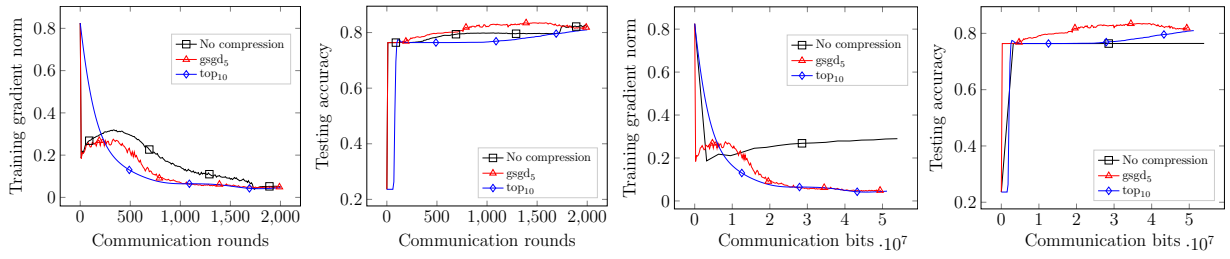


Figure 4.4: The training gradient norm and testing accuracy against communication rounds (top two panels) and communication bits (bottom two panels) for BEER using different compression schemes for logistic regression with nonconvex regularization on unshuffled a9a dataset.

Chapter 5

Decentralized private stochastic algorithm with communication compression

To enable large-scale machine learning in bandwidth-hungry environments such as wireless networks, significant progress has been made recently in designing communication-efficient federated learning algorithms with the aid of communication compression. On the other end, privacy-preserving, especially at the client level, is another important desideratum that has not been addressed simultaneously in the presence of advanced communication compression techniques yet. In this chapter, based on `BEER`, we propose `PORTER`, a decentralized stochastic optimization algorithm for decentralized nonconvex ERM problems (cf. (1.3)) which enhances the communication efficiency of private decentralized learning with communication compression, by exploiting general compression operators, gradient clipping and local differential privacy.

We provide the *first* theoretical analysis of private decentralized optimization algorithms using communication compression and gradient clipping, and show explicit dependency on the mixing rate and compression parameter. Furthermore, we provide numerical evidence that shows `PORTER` converges in similar communication rounds without sacrificing privacy nor utility compared to server/client private optimization algorithm `SoteriaFL-SGD` [LZLC22].

5.1 Preliminaries

5.1.1 Local differential privacy

In decentralized learning systems, all agents share potentially sensitive information with their neighbors. If some agents are exploited by adversaries, the system will face a risk of privacy leaking even though

the system-level privacy is protected. Therefore, we introduce local differential privacy (LDP) defined Definition 7 following [DJW13, ABCP13, CABP13, XYD19, WXDX20, ZZY+21, LZLC22], which protects each agent's privacy from leaking to other agents.

Definition 7 (Local differential privacy (LDP)). *A randomized mechanism $\mathcal{M}: \mathcal{Z} \to \mathcal{R}$ with domain $\mathcal{Z}$ and range $\mathcal{R}$ satisfies (ϵ, δ)-local differential privacy for client i, if for any two neighboring dataset $\boldsymbol{Z}_i, \boldsymbol{Z}_i^{'} \in \mathcal{Z}$ at client i and for any subset of outputs $\boldsymbol{R} \subseteq \mathcal{R}$, it holds that*

$$\mathbb{P}\big(\mathcal{M}(\boldsymbol{Z}_i) \in \boldsymbol{R}\big) \leq e^{\epsilon}\mathbb{P}\big(\mathcal{M}(\boldsymbol{Z}_i^{'}) \in \boldsymbol{R}\big) + \delta. \tag{5.1}$$

The two datasets $\boldsymbol{Z}_i$ and $\boldsymbol{Z}_i^{'}$ are called neighboring if they are only different by one data point at client i.

Definition 7 is a more stricter privacy guarantee because it can imply general differential privacy (DP). Consequently, LDP requires a larger perturbation variance than general DP. To identify the impact of the decentralized LDP setting compared to centralized DP setting, we define the baseline utility

$$\phi_m = \frac{\sqrt{d\log(1/\delta)}}{m\epsilon}, \tag{5.2}$$

which can be understood as the final utility of a centralized system with m data samples that guarantees (ϵ, δ)-DP. For typical private problems, the local sample size m has to be large enough for the privacy perturbation to work, we impose a mild assumption that $\phi_m < 1$. For example, the problem defined in (1.3) has in total mn data samples, running an (ϵ, δ)-DP algorithm on one server that can access all data will achieve $\frac{1}{n}\phi_m$ utility in $n\phi_m^{-1}$ iterations.

5.1.2 The `SoteriaFL` algorithm framework

This section reviews `SoteriaFL` [LZLC22] (cf. Algorithm 6) as it has inspired the development of `PORTER`. `SoteriaFL` is a unified framework for differentially private optimization algorithms with unbiased communication compression in the server/client setting, which employs shifted communication to compensate for the error induced by compression, and can use various local sub-routines to produce an estimate of global gradients, e.g., stochastic gradient descent (SGD) or stochastic variance reduced gradient (SVRG).

During each iteration, each agent 1) computes an estimate of the global gradient using a local sub-routine, 2) adds a Gaussian perturbation to the gradient estimate to preserve privacy, and 3) communicates the perturbed gradient with its neighbors using the shifted communication scheme. Instead of directly compressing the perturbed gradients, `SoteriaFL` maintains a reference $\boldsymbol{s}_i^{(t)}$ and compresses the shifted message $\boldsymbol{g}_i^{(t)} - \boldsymbol{s}_i^{(t)}$ at each agent. This extra shift operation allows `SoteriaFL` achieve much better convergence behavior (fewer communication rounds) than existing algorithms.

Algorithm 6 `SoteriaFL` (a unified framework for compressed private FL)

1 initial point $\boldsymbol{x}^{(0)}$, step size η_t, shift step size γ_t, variance σ_p^2, initial reference $\boldsymbol{s}_i^{(0)} = \boldsymbol{0}_d$
2 **for** $t = 0, 1, 2, \ldots, T$ **do**
3 **for each node** $i \in [n]$ **do in parallel**
4 Compute local gradient estimator $\tilde{\boldsymbol{g}}_i^{(t)}$ ▷ *It allows many methods, e.g., SGD, SVRG, and SAGA.*
5 *Privacy*: $\boldsymbol{g}_i^{(t)} = \tilde{\boldsymbol{g}}_i^{(t)} + \boldsymbol{\xi}_i^{(t)}$, where $\boldsymbol{\xi}_i^{(t)} \sim \mathcal{N}(\boldsymbol{0}, \sigma_p^2 \boldsymbol{I})$
6 *Compression*: let $\boldsymbol{v}_i^{(t)} = \mathcal{C}_i^t(\boldsymbol{g}_i^{(t)} - \boldsymbol{s}_i^{(t)})$ and send to the server ▷ *Shifted compression.*
7 Update shift $\boldsymbol{s}_i^{t+1} = \boldsymbol{s}_i^{(t)} + \gamma_t \mathcal{C}_i^t(\boldsymbol{g}_i^{(t)} - \boldsymbol{s}_i^{(t)})$
8 **end each node** Server aggregates compressed information $\boldsymbol{v}^t = \boldsymbol{s}^t + \frac{1}{n}\sum_{i=1}^n \boldsymbol{v}_i^t$
9 $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_t \boldsymbol{v}^t$
10 $\boldsymbol{s}^{t+1} = \boldsymbol{s}^t + \gamma_t \frac{1}{n}\sum_{i=1}^n \boldsymbol{v}_i^t$

`SoteriaFL-SGD` is obtained by using mini-batch stochastic gradients as the gradient estimator for Line 4. The theoretical analysis shows explicit dependency on the compression parameter, which helps to better understand the communication-utility trade-off for private server/client optimization algorithms, as show in Table 1.5. Compare the centralized DP setting, the server/client algorithm differentially private algorithm DDP-SRM [WJEG19] reaches the same $\frac{1}{n}\phi_m$ utility but at a worse $n^2 d\phi_m^{-1}$ iteration complexity. `SoteriaFL-SGD` is also a server/client algorithm but with LDP guarantee, which reaches an even worse utility but with potentially less iterations due to the stronger privacy constraint.

5.1.3 Gradient clipping

In practice, gradient clipping is frequently adopted to ensure the gradients are within a predetermined region, so that the variance of privacy perturbation can be decided accordingly. The clipping operator we adopt is a smooth clipping operator defined in Definition 8, which scales a vector into a ball of radius τ centered at the origin. Another widely used clipping operator is the piece-wise linear clipping operator defined in Definition 9, which scales inputs whenever its gradient norm is larger than τ and does nothing otherwise.

Definition 8 (Smooth clipping operator). *For $\boldsymbol{x} \in \mathbb{R}^d$, the clipping operator is defined as*

$$\mathrm{Clip}_\tau(\boldsymbol{x}) = \frac{\tau}{\tau + \|\boldsymbol{x}\|_2}\boldsymbol{x}.$$

For $\boldsymbol{X} \in \mathbb{R}^{d\times n}$, the distributed clipping operator is defined as

$$\mathrm{Clip}_\tau(\boldsymbol{X}) = [\mathrm{Clip}_\tau(\boldsymbol{x}_1), \ldots, \mathrm{Clip}_\tau(\boldsymbol{x}_n)].$$

Definition 9 (Piece-wise linear clipping operator). *For $\boldsymbol{x} \in \mathbb{R}^d$, the clipping operator is defined as*

$$\mathrm{Clip}_\tau(\boldsymbol{x}) = \boldsymbol{x} \min\left\{1, \tau/\|\boldsymbol{x}\|_2\right\}.$$

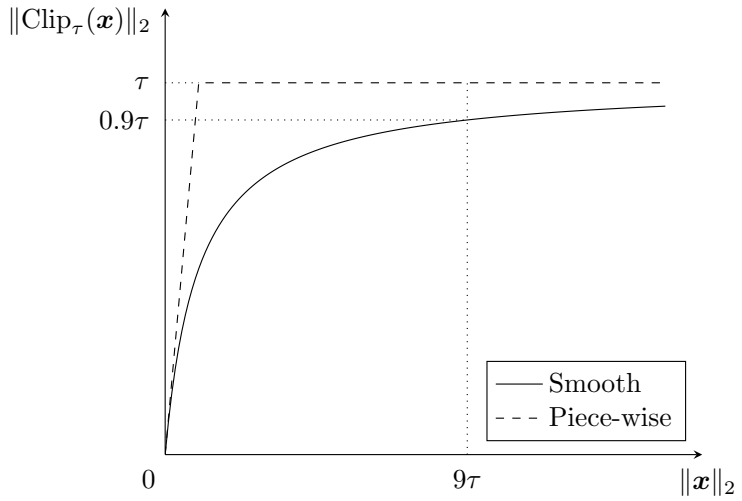


Figure 5.1: Illustration of input norm and clipped norm for the smooth clipping operator (Definition 8) and piece-wise linear clipping operator (Definition 9), where τ is the clipping parameter.

Figure 5.1 plots the norm of a vector before and after clipping for these two clipping operators, which shows most fundamental difference is when $\|\boldsymbol{x}\|_2 = \tau$, the piece-wise linear compression operator is not smooth, which leads to more difficulties in theoretical analysis. When $\|\boldsymbol{x}\|_2$ is small, these clipping operators behave like identity transformation, which keeps the stationary point property of objective functions.

5.2 The PORTER algorithm

We propose PORTER (Algorithm 7), a novel stochastic private decentralized optimization algorithm for finding ν-stationary points of nonconvex finite-sum problems based on BEER, where gradient estimates $\boldsymbol{V}$, stochastic gradients $\boldsymbol{G}$, perturbation noise $\boldsymbol{E}$, compressed surrogate $\boldsymbol{Q}_x$ and $\boldsymbol{Q}_v$, and their corresponding agent-wise values are defined analogously to (1.5).

Key ingredients of PORTER are: 1) gradient clipping, which ensures the norm of clipped gradients are bounded by τ; 2) privacy perturbation, which adds a Gaussian noise to clipped gradients to achieve privacy constraints; 3) compression with error feedback [RSF21], which accelerates the convergence with biased compression operators; and 4) stochastic gradient tracking, tracks the global gradient locally at each agent. The algorithm details explained in Section 4.2 also applies to PORTER, except PORTER doesn't explicitly show compressed messages as Lines 4 and 7, and PORTER adds perturbation to the stochastic

Algorithm 7 PORTER

input: $\bar{\boldsymbol{x}}^{(0)}$, η, γ, τ, b, σ_p, T

initialize: $\boldsymbol{V}^{(0)} = \boldsymbol{Q}_v^{(0)} = \boldsymbol{G}_p^{(0)} = \boldsymbol{0}_{d\times n}$, $\boldsymbol{Q}_x^{(0)} = \boldsymbol{X}^{(0)} = \bar{\boldsymbol{x}}^{(0)}\boldsymbol{1}_n^\top$

for $t = 1, \ldots, T$ **do**

$\boldsymbol{G}_\tau^{(t)} = \frac{1}{b}\sum_{\boldsymbol{Z}\in\mathcal{Z}^{(t)}} \mathrm{Clip}_\tau(\nabla\ell(\boldsymbol{X}^{(t-1)};\boldsymbol{Z}))$

$\boldsymbol{G}_p^{(t)} = \boldsymbol{G}_\tau^{(t)} + \boldsymbol{E}^{(t)}$, where $\boldsymbol{e}_i^{(t)} \sim \mathcal{N}(\boldsymbol{0}_d, \sigma_p^2\boldsymbol{I}_d)$

$\boldsymbol{Q}_v^{(t)} = \boldsymbol{Q}_v^{(t-1)} + \mathcal{C}(\boldsymbol{V}^{(t-1)} - \boldsymbol{Q}_v^{(t-1)})$ ▷ *Communication*

$\boldsymbol{V}^{(t)} = \boldsymbol{V}^{(t-1)} + \gamma\boldsymbol{Q}_v^{(t)}(\boldsymbol{W} - \boldsymbol{I}_n) + \boldsymbol{G}_p^{(t)} - \boldsymbol{G}_p^{(t-1)}$

$\boldsymbol{Q}_x^{(t)} = \boldsymbol{Q}_x^{(t-1)} + \mathcal{C}(\boldsymbol{X}^{(t-1)} - \boldsymbol{Q}_x^{(t-1)})$ ▷ *Communication*

$\boldsymbol{X}^{(t)} = \boldsymbol{X}^{(t-1)} + \gamma\boldsymbol{Q}_x^{(t)}(\boldsymbol{W} - \boldsymbol{I}_n) - \eta\boldsymbol{V}^{(t)}$

output: $\boldsymbol{x}_{out} \sim \mathrm{Uniform}(\{\boldsymbol{x}_i^{(t)} | i \in [n], t \in [T]\})$.

gradients. PORTER initializes gradient-related variables to $\boldsymbol{0}_d$ and other variables to the same random value $\bar{\boldsymbol{x}}^{(0)}$, which improves the algorithm's stability in early iterations and simplifies analysis but has no impact on convergence rates.

5.3 Local differential privacy guarantee

Algorithm 7 adds a Gaussian noise $\boldsymbol{e}_i^{(t)} \sim \mathcal{N}(\boldsymbol{0}_d, \sigma_p^2\boldsymbol{I}_d)$ to ensure privacy. Theorem 9 proves our algorithm is (ϵ, δ)-private when setting the variance of Gaussian perturbation to be $\sigma_p^2 = T\tau^2 b^2 \phi_m^2/d$. The proof is deferred to Appendix D.1.

Theorem 9 (Local differential privacy). *Let $\phi_m = \frac{\sqrt{d\log(1/\delta)}}{m\epsilon}$. For any $\epsilon \le T/m^2$ and $\delta \in (0,1)$, Algorithm 7 is (ϵ, δ)-LDP after T iterations if we set*

$$\sigma_p^2 = \frac{T\tau^2 b^2 \log(1/\delta)}{m^2\epsilon^2} = T\tau^2 b^2 \phi_m^2/d. \tag{5.3}$$

Using clipping operators guarantees all gradients' ℓ_2 norms are bounded by τ, i.e., $\forall i, t$, $\left\|\boldsymbol{g}_{\tau,i}^{(t)}\right\|_2 \le \tau$. Therefore, Theorem 9 holds for PORTER throughout.

5.4 Convergence with bounded gradient assumption

This section theoretically analyzes the convergence properties of PORTER under bounded gradient assumption without gradient clipping. Section 5.3 shows PORTER is (ϵ, δ)-local differential private when using a specific perturbation variance $\sigma_p^2 = T\tau^2 b^2 \phi_m^2/d$, then presents convergence analysis.

Assumption 8 (Bounded sample gradient). *For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ and any datum $\boldsymbol{z}$ in dataset $\mathcal{Z}$, it holds*

$$\|\nabla \ell(\boldsymbol{x}; \boldsymbol{z})\|_2 \leq \tau.$$

With Assumption 8, PORTER can omit the clipping operator and reduce to Algorithm 8. Here, we assume $b = 1$.

Algorithm 8 PORTER with bounded gradient assumption

input: $\bar{\boldsymbol{x}}^{(0)}$, η, γ, τ, σ_p, T
initialize: $\boldsymbol{V}^{(0)} = \boldsymbol{Q}_v^{(0)} = \boldsymbol{G}_p^{(0)} = \boldsymbol{0}_{d\times n}$, $\boldsymbol{Q}_x^{(0)} = \boldsymbol{X}^{(0)} = \bar{\boldsymbol{x}}^{(0)}\boldsymbol{1}_n^\top$
for $t = 1, \ldots, T$ **do**
  $\boldsymbol{G}_\tau^{(t)} = \nabla \ell(\boldsymbol{X}^{(t-1)}; \boldsymbol{Z}^{(t)})$
  $\boldsymbol{G}_p^{(t)} = \boldsymbol{G}_\tau^{(t)} + \boldsymbol{E}^{(t)}$, where $\boldsymbol{e}_i^{(t)} \sim \mathcal{N}(\boldsymbol{0}_d, \sigma_p^2 \boldsymbol{I}_d)$
  $\boldsymbol{Q}_v^{(t)} = \boldsymbol{Q}_v^{(t-1)} + \mathcal{C}(\boldsymbol{V}^{(t-1)} - \boldsymbol{Q}_v^{(t-1)})$ ▷ *Communicate*
  $\boldsymbol{V}^{(t)} = \boldsymbol{V}^{(t-1)} + \gamma \boldsymbol{Q}_v^{(t)}(\boldsymbol{W} - \boldsymbol{I}_n) + \boldsymbol{G}_p^{(t)} - \boldsymbol{G}_p^{(t-1)}$
  $\boldsymbol{Q}_x^{(t)} = \boldsymbol{Q}_x^{(t-1)} + \mathcal{C}(\boldsymbol{X}^{(t-1)} - \boldsymbol{Q}_x^{(t-1)})$ ▷ *Communicate*
  $\boldsymbol{X}^{(t)} = \boldsymbol{X}^{(t-1)} + \gamma \boldsymbol{Q}_x^{(t)}(\boldsymbol{W} - \boldsymbol{I}_n) - \eta \boldsymbol{V}^{(t)}$
output: $\boldsymbol{x}_{out} \sim \text{Uniform}(\{\boldsymbol{x}_i^{(t)} | i \in [n], t \in [T]\})$.

Theorem 10 presents the convergence result of PORTER using general compression operators (Definition 5). The proof is deferred to Appendix D.2.

Theorem 10 (Convergence using general compression operators). *Assume Assumptions 1, 2 and 8 hold. Use general compression operators (Definition 5). Set $\gamma = O\big((1-\alpha)\rho\big)$, $\eta = O\big(\gamma^{4/3}(1-\alpha)^{4/3}\phi_m/L\big)$, $T = \phi_m^{-2}$, and $\sigma_p^2 = T\tau^2\phi_m^2/d$. Let $\Delta = \mathbb{E}\big[f(\bar{\boldsymbol{x}}^{(0)})\big] - f^\star$, Algorithm 8 reaches $O\big((1-\alpha)^{-8/3}\rho^{-4/3}\phi_m \cdot \max\big\{\tau^2, L\Delta\big\}\big)$ utility in T iterations.*

We can compare each term of the final utility in Theorem 10 with the results of the server/client algorithm SoteriaFL-SGD, due to a lack of theoretical analysis of comparable algorithms. 1) PORTER will always reach the same final utility, whose dependency on the compression operator is $O(\rho^{-4/3})$. We can convert SoteriaFL-SGD's rate (cf. Table 1.5) using the approximation $\rho = 1/(1+\omega)$. When n is not large enough compared to ω, or using aggressive compression (ω is large), SoteriaFL-SGD reaches $O\big(n^{-1}\rho^{-2}\phi_m\big)$ utility, which is worse than PORTER, even though PORTER is a decentralized optimization algorithm. 2) In terms of dependency on the baseline utility ϕ_m, due to extra complexities induced by the decentralized setting, PORTER takes $O(\phi_m^{-2})$ iterations to reach $O(\phi_m)$ utility, which is worse than other algorithms, e.g., SoteriaFL-SGD only takes $O(\phi_m^{-1})$ iterations.

5.5 Convergence without bounded gradient assumption

When assuming the bounded gradient assumption (cf. Assumption 8), PORTER can skip the clipping operator, and $\boldsymbol{G}_{\tau,i}^{(t)}$ becomes an unbiased estimator of the local gradient $\nabla f_i(\boldsymbol{x}_i^{(t)})$. However, Assumption 8 is rarely met in reality. For example, the gradient of a quadratic loss function is not bounded. Therefore, it is of interest to examine convergence without the strong bounded gradient assumption, where we utilize the gradient clipping operator $\mathrm{Clip}_\tau(\cdot)$ to ensure gradients are bounded.

However, it is necessary to introduce milder assumptions, Assumptions 9 and 10, which limit the deviation of local objective functions to the global objective function. Because stochastic gradients at different agents will lose correct scaling after clipping, which will break the stationary point property at local minima. For example, consider optimizing a one-dimensional objective function over a 3-agent fully connected communication graph. Initialize every agent with $\overline{\boldsymbol{x}}^{(0)} = \boldsymbol{x}^\star$, where $\boldsymbol{x}^\star$ is one of global local minima. Assume local gradients are $g_1 = 1$, $g_2 = 1$ and $g_3 = -2$, which leads to a global gradient of $\mathbf{0}_3$. If we apply gradient clipping with $\tau = 1$, $\mathrm{Clip}_1(g_1) = 0.5$, $\mathrm{Clip}_1(g_2) = 0.5$ and $\mathrm{Clip}_1(g_3) = -0.66$, and the global gradient is $g' = 0.33 \neq 0$.

Assumption 9 (Bounded gradient dissimilarity)**.** $\forall \boldsymbol{x} \in \mathbb{R}^d$ *and* $i \in [n]$, $\left\|\nabla f(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x})\right\|_2 \leq \frac{1}{12}\left\|\nabla f(\boldsymbol{x})\right\|_2$.

Assumption 10 (Bounded local variance)**.** $\forall \boldsymbol{x} \in \mathbb{R}^d$ *and* $i \in [n]$, $\mathbb{E}_{\boldsymbol{z}\sim\mathcal{Z}_i}\|\nabla f_i(\boldsymbol{x}) - \nabla \ell(\boldsymbol{x};\boldsymbol{z})\|_2^2 \leq \sigma_g^2$.

As an initial step to understand the clipping operation, we study a variant of PORTER, which applies the clipping to the mini-batch gradient, i.e. Line 4 is modified to

$$\boldsymbol{G}^{(t)} = \frac{1}{b}\sum_{\boldsymbol{Z}\in\mathcal{Z}^{(t)}} \nabla \ell(\boldsymbol{X}^{(t-1)};\boldsymbol{Z}) \tag{5.4a}$$

$$\boldsymbol{G}_\tau^{(t)} = \mathrm{Clip}_\tau(\boldsymbol{G}^{(t)}), \tag{5.4b}$$

which is also a widely used operation in deep learning training. However, the privacy guarantee does not apply to this variation anymore. Theorem 11 describes the convergence behavior of the modified version of Algorithm 7 in this case. The proof is deferred to Appendix D.3.

Theorem 11 (Convergence without bounded gradient assumptions)**.** *Assume Assumptions 1, 2, 9 and 10 hold and use general compression operators (Definition 5). Let* $\Delta = \mathbb{E}\left[f(\overline{\boldsymbol{x}}^{(0)})\right] - f^\star$ *and* $\nu = O\big(\rho^{-\frac{2}{3}}(1-\alpha)^{-\frac{4}{3}}\sigma_g^{1/2}\phi_m^{1/4}\big)$. *Set* $\gamma = O((1-\alpha)\rho)$, $\eta = O\big(L^{-1}\gamma^{\frac{4}{3}}(1-\alpha)^{\frac{4}{3}}\sqrt{bL\Delta\phi_m}/\tau\big)$, $b = O(\sigma_g^2\nu^{-2})$, $\tau = \nu$, $\sigma_p^2 = T\tau^2\phi_m^2/d$, *and* $T = (b\phi_m)^{-1} = O\big(\rho^{\frac{4}{3}}(1-\alpha)^{\frac{8}{3}}\phi_m^{-1}\big)$, *Algorithm 7 with mini-batch clipping (5.4) reaches a* ν*-first-order stationary point in no more than* T *iterations, i.e.,*

$$\min_{t\in[T]} \mathbb{E}\left\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\right\|_2 \leq \nu. \tag{5.5}$$

Theorem 11 establishes a simple framework for analyzing algorithms that use gradient clipping without strong assumptions, which results in a clean deterministic bound of the iterations to reach a ν-stationary point. In contrast, most works that analyze gradient clipping require stronger assumptions. We believe our technique can be applied to analyze the original version of PORTER too, which will be left to future work.

5.6 Numerical experiments

This section presents numerical experiments on the a9a dataset to numerically examine the performance of PORTER, with comparison to the state-of-the-art server/client private optimization algorithm SoteriaFL-SGD, which also utilizes communication compression and guarantees local differential privacy. More specifically, we evaluate the convergence of utility and accuracy in terms of communication rounds and communication bits, to analyze the privacy-utility-communication trade-offs of different optimization algorithms.

For all experiments, we split shuffled datasets evenly to 10 agents that are connected by an Erdős-Rényi random graph with connecting probability $p = 0.8$. We use the FDLA matrix [XB04] as the mixing matrix to perform weighted information aggregation to accelerate convergence. We use biased random sparsification (cf. Example 1) for all algorithms where $k = \lfloor \frac{d}{20} \rfloor$, i.e., the compressor randomly selects 5% elements from each vector. We also apply gradient clipping with $\tau = 1$ to all algorithms for simplicity. For each experiment, all algorithms are initialized to the same staring point, and use best-tuned learning rates, batch size 1 and $\sigma_p = \frac{\tau\sqrt{T\log(1/\delta)}}{m\epsilon}$.

5.6.1 Logistic regression with nonconvex regularization

We run experiments on logistic regression with nonconvex regularization [WJZ+19] on the a9a dataset [CL11] for different privacy settings. Similar to Section 3.3.1, following [WJZ+19], the objective function is defined as

$$\ell(\boldsymbol{x}; \{\boldsymbol{f}, l\}) = \log\left(1 + l\exp(-\boldsymbol{x}^\top \boldsymbol{f})\right) + \lambda \sum_{i=1}^{d} \frac{x_i^2}{1 + x_i^2},$$

where $\{\boldsymbol{f}, l\}$ represents a training tuple, $\boldsymbol{f} \in \mathbb{R}^d$ is the feature vector and $l \in \{0, 1\}$ is the label, and λ is the regularization parameter which is set to $\lambda = 0.2$.

Figure 5.2 and Figure 5.3 show the convergence results of PORTER and SoteriaFL-SGD for logistic regression with nonconvex regularization on the a9a dataset to reach $(10^{-2}, 10^{-3})$-LDP and $(10^{-1}, 10^{-3})$-LDP, respectively. Under $(10^{-2}, 10^{-3})$-LDP, which is a more stricter privacy setting, PORTER converges faster than SoteriaFL-SGD in test accuracy, while PORTER converges slightly slower in train utility. Under $(10^{-1}, 10^{-3})$-LDP, PORTER performs SoteriaFL-SGD slightly worse than SoteriaFL-SGD. The results highlight PORTER's

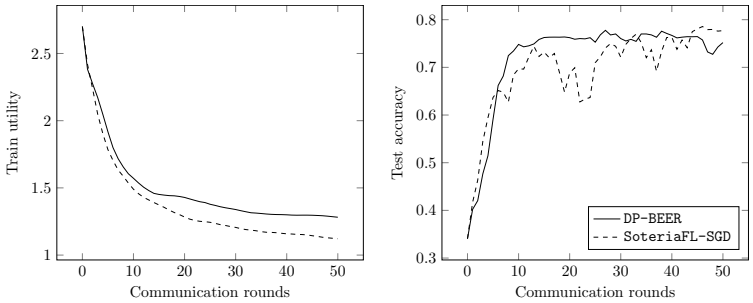


Figure 5.2: The train utility and test accuracy vs. communication rounds for logistic regression with nonconvex regularization on the a9a dataset when guaranteeing $(10^{-2}, 10^{-3})$-LDP. Both PORTER and SoteriaFL-SGD employ $\text{random}_{162,\textit{biased}}$ compression (cf. Example 1).

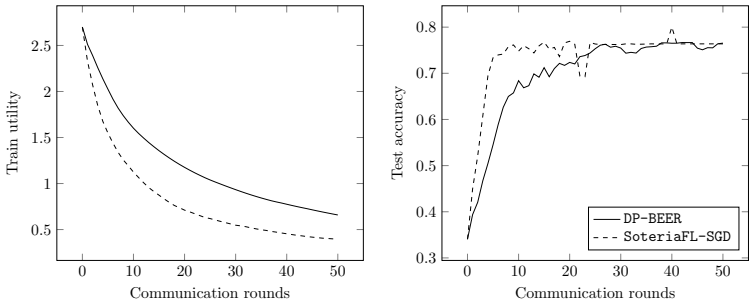


Figure 5.3: The train utility and test accuracy vs. communication rounds for logistic regression with nonconvex regularization on the a9a dataset when guaranteeing $(10^{-1}, 10^{-3})$-LDP. Both PORTER and SoteriaFL-SGD employ $\text{random}_{162,\textit{biased}}$ compression (cf. Example 1).

communication efficiency by showing it can achieve similar performance as its server/client counterpart, i.e. SoteriaFL-SGD, especially under strict privacy constraints.

5.6.2 One-hidden-layer neural network training

Similar to Section 4.4.2, we evaluate PORTER by training a one-hidden layer neural network on the MNIST dataset [LJB+95]. The network uses 64 hidden neurons, sigmoid activation functions and cross-entropy loss, where the loss function over a training pair $\{\boldsymbol{f}, l\}$ is defined as

$$\ell(\boldsymbol{x}; (\boldsymbol{f}, l)) = \mathsf{CrossEntropy}(\mathsf{softmax}(\boldsymbol{W}_2\ \mathsf{sigmoid}(\boldsymbol{W}_1 \boldsymbol{f} + \boldsymbol{c}_1) + \boldsymbol{c}_2), l).$$

Here the model parameter is defined by $\boldsymbol{x} = \text{vec}(\boldsymbol{W}_1, \boldsymbol{c}_1, \boldsymbol{W}_2, \boldsymbol{c}_2)$, where the dimensions of the network parameters $\boldsymbol{W}_1$, $\boldsymbol{c}_1$, $\boldsymbol{W}_2$, $\boldsymbol{c}_2$ are 64×784, 64×1, 10×64, and 10×1, respectively.

Figures 5.4 and 5.5 show the convergence results of PORTER and SoteriaFL-SGD for training a one-hidden-layer neural network on the MNIST dataset to reach $(10^{-2}, 10^{-3})$-LDP and $(10^{-1}, 10^{-3})$-LDP, respectively. PORTER Under both privacy settings, PORTER converges at a similar rate as SoteriaFL-SGD in train utility. However, in terms of convergence in test accuracy, PORTER outperforms SoteriaFL-SGD

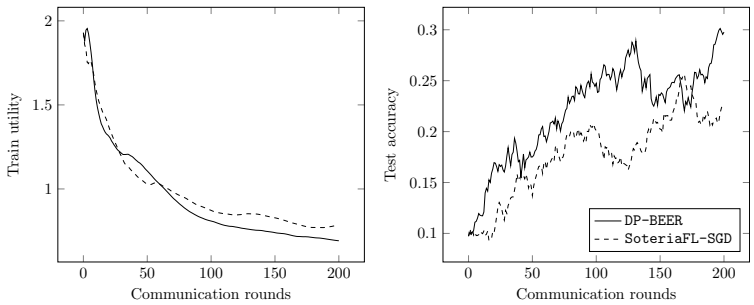


Figure 5.4: The train utility and test accuracy vs. communication rounds for training a one-hidden-layer neural network on the MNIST dataset when guaranteeing $(10^{-2}, 10^{-3})$-LDP. Both PORTER and SoteriaFL-SGD employ $\text{random}_{2583,\textit{biased}}$ compression (cf. Example 1).

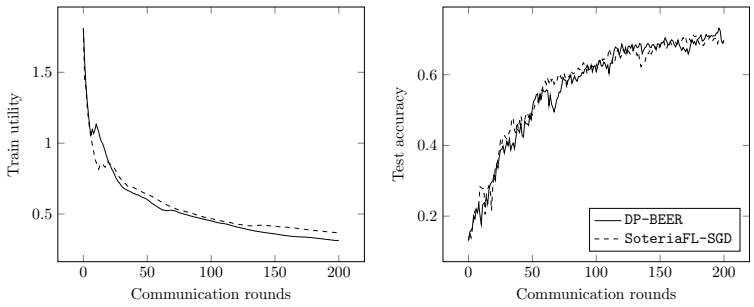


Figure 5.5: The train utility and test accuracy vs. communication rounds for training a one-hidden-layer neural network on the MNIST dataset when guaranteeing $(10^{-1}, 10^{-3})$-LDP. Both PORTER and SoteriaFL-SGD employ $\text{random}_{2583,\textit{biased}}$ compression (cf. Example 1).

under the more stricter $(10^{-2}, 10^{-3})$-LDP, while the two algorithms have similar performance under the other setting. This experiment again empathizes PORTER's communication efficiency by comparing to the server/client algorithm SoteriaFL-SGD.

Chapter 6

Conclusions

The main focuses of this thesis are theoretical and practical communication and computation efficiency of decentralized optimization algorithms, where these algorithms leverage techniques such as gradient tracking, Newton-type updates, stochastic recursive gradients, communication compression and differential privacy.

We start from the `Network-DANE` algorithm in Chapter 2, which adapts Newton-type updates to the decentralized setting using gradient tracking and extra mixing, and achieves a condition number free (up to a log factor) communication complexity for quadratic objectives. `Network-DANE` indicates that it is possible to perform more local computation, i.e., solving a strongly convex optimization local problem at each agent, and more communication per iteration, i.e., extra mixing, to reduce overall communication complexity.

Next, we propose `DESTRESS`, a stochastic recursive gradient algorithm, in Chapter 3. `DESTRESS` employs recursive stochastic gradients, stochastic gradient tracking and extra mixing, which results in an optimal IFO-complexity for arbitrary nonconvex ERM problems and improved communication complexity upon existing works. `DESTRESS` reaches both communication and computation efficiency at the same time, which emphasizes the efficacy of gradient tracking, extra mixing and stochastic algorithms.

Then, we propose `BEER` and `PORTER` in Chapters 4 and 5, respectively. While these two algorithms share a similar structure that uses gradient tracking, communication compression and error feedback, they focus on different perspectives. `BEER` achieves an improved communication complexity but at the cost of worse IFO complexity, while being able to converge under high data heterogeneity. On top of `BEER`, `PORTER` adds gradient clipping and privacy perturbation to gradients to protect the privacy of each agent, with explicit utility and communication complexity. The development of `BEER` and `PORTER` shows that by using gradient tracking and communication compression, the efficiency of decentralized optimization algorithms can be significantly improved, and these algorithms can be easily extended to suit new problems, e.g.,

privacy-preserving decentralized optimization.

Appendix A

Appendix for Chapter 2

A.1 Derivation of Equation (2.6)

We observe that

$$f_j(\boldsymbol{x}) - \left\langle \nabla f_j(\overline{\boldsymbol{x}}^{(t)}), \boldsymbol{x} \right\rangle = \tfrac{1}{2}\boldsymbol{x}^\top \boldsymbol{H}_j \boldsymbol{x} - \boldsymbol{x}^\top \boldsymbol{H}_j \overline{\boldsymbol{x}}^{(t)} + \text{constant} = \tfrac{1}{2}\left(\boldsymbol{x} - \overline{\boldsymbol{x}}^{(t)}\right)^\top \boldsymbol{H}_j \left(\boldsymbol{x} - \overline{\boldsymbol{x}}^{(t)}\right) + \text{constant},$$

which allows us to derive a closed-form expression for $\boldsymbol{x}_j^{(t)}$ as follows

$$\begin{aligned}
\boldsymbol{x}_j^{(t)} &= \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \left\{ \frac{1}{2}\left(\boldsymbol{x} - \overline{\boldsymbol{x}}^{(t)}\right)^\top \boldsymbol{H}_j \left(\boldsymbol{x} - \overline{\boldsymbol{x}}^{(t)}\right) + \left\langle \nabla f\left(\overline{\boldsymbol{x}}^{(t)}\right), \boldsymbol{x} - \overline{\boldsymbol{x}}^{(t)} \right\rangle + \frac{\mu}{2}\left\|\boldsymbol{x} - \overline{\boldsymbol{x}}^{(t)}\right\|_2^2 \right\} \\
&= \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \left\{ \frac{1}{2}\left(\boldsymbol{x} - \overline{\boldsymbol{x}}^{(t)}\right)^\top \left(\boldsymbol{H}_j + \mu \boldsymbol{I}\right) \left(\boldsymbol{x} - \overline{\boldsymbol{x}}^{(t)}\right) + \left\langle \nabla f\left(\overline{\boldsymbol{x}}^{(t)}\right), \boldsymbol{x} - \overline{\boldsymbol{x}}^{(t)} \right\rangle \right\} \\
&= \overline{\boldsymbol{x}}^{(t)} - \left(\boldsymbol{H}_j + \mu \boldsymbol{I}_d\right)^{-1} \nabla f\left(\overline{\boldsymbol{x}}^{(t)}\right).
\end{aligned}$$

A.2 Proof of Theorem 1 and Theorem 2

This sections proves the convergence rate of `Network-DANE` for quadratic losses. When local and global loss functions are quadratic, we can solve (2.9) explicitly. Specifically, Algorithm 1 can be alternatively written as Algorithm 9 below.

For simplicity, we first proceed with the proof as if $K = 1$, then replace α with α^K in the last step, and we let $\overline{\boldsymbol{H}} = \nabla^2 f(\boldsymbol{x}) = \frac{1}{n}\sum_{j=1}^n \boldsymbol{H}_j$ be the Hessian of the global loss function. From the definition of the homogeneity parameter β, we have $\|\overline{\boldsymbol{H}} - \boldsymbol{H}_j\|_2 \leq \beta$ for all $j = 1, \ldots, n$. In addition, we recall the notation in Definition 2, (2.3) and (2.4), and define the error vector as follows

$$\boldsymbol{e}^{(t)} = \begin{bmatrix} \sqrt{n}\|\overline{\boldsymbol{y}}^{(t)} - \boldsymbol{y}^\star\|_2 \\ \|\boldsymbol{y}^{(t)} - \boldsymbol{1}_n \otimes \overline{\boldsymbol{y}}^{(t)}\|_2 \\ L^{-1}\|\boldsymbol{s}^{(t)} - \nabla f(\boldsymbol{y}^{(t)})\|_2 \end{bmatrix}. \tag{A.2}$$

Algorithm 9 `Network-DANE` for quadratic losses (2.10)

for $t = 1, 2, \cdots$ **do**

$$\boldsymbol{y}^{(t)} = (\boldsymbol{W} \otimes \boldsymbol{I}_d)\boldsymbol{x}^{(t-1)}, \tag{A.1a}$$
$$\boldsymbol{s}^{(t)} = (\boldsymbol{W} \otimes \boldsymbol{I}_d)\boldsymbol{s}^{(t-1)} + \boldsymbol{H}\big(\boldsymbol{y}^{(t)} - \boldsymbol{y}^{(t-1)}\big), \tag{A.1b}$$
$$\boldsymbol{x}^{(t)} = \boldsymbol{y}^{(t-1)} - (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1}\boldsymbol{s}^{(t-1)}, \tag{A.1c}$$

where $\boldsymbol{y}^{(t)}$ and $\boldsymbol{s}^{(t)}$ are defined in (2.3), $\boldsymbol{H} := \mathrm{diag}(\boldsymbol{H}_1, \cdots, \boldsymbol{H}_n) \in \mathbb{R}^{nd \times nd}$, and $\boldsymbol{H}_i$ is defined in (2.10).

Establishing the convergence of `Network-DANE` relies on characterization of the per-iteration dynamics of $\boldsymbol{e}^{(t)}$ for quadratic losses. Towards this end, we state the following key lemma — which is established in Appendix A.5 — that plays a crucial role in the analysis.

Lemma 1. *Let $\eta = \frac{1}{\sigma+\mu}$ and $\gamma = \frac{L}{L+\mu}$. Suppose that Assumptions 3 and 4 hold. Then one has*

$$\boldsymbol{e}^{(t)} \leq \underbrace{\begin{bmatrix} \theta_1 & \gamma\eta\beta + \eta\beta & \eta^2 L\beta \\ \alpha\gamma\eta\beta & \alpha + \alpha\eta L & \alpha\eta L \\ \frac{\beta}{L} + \theta_1\frac{\beta}{L} + \alpha\gamma\eta\beta\frac{\beta}{L} & \alpha\frac{\beta}{L} + \alpha + 1 + \gamma\eta\beta\frac{\beta}{L} + \eta\beta\frac{\beta}{L} + \alpha\frac{\beta}{L} + \alpha\eta\beta & \alpha + \gamma\eta\beta\frac{\beta}{L} + \alpha\eta\beta \end{bmatrix}}_{=:\boldsymbol{G}} \boldsymbol{e}^{(t-1)}. \tag{A.3}$$

Here, $\boldsymbol{a} \leq \boldsymbol{b}$ indicates that $a_i \leq b_i$ for all entries i.

In what follows, we invoke this result to establish Theorems 1 and 2 separately.

A.2.1 Proof of Theorem 1

By the choice of μ stated in Theorem 1, we can show that

$$\gamma < 1 \qquad \text{and} \qquad \eta\beta \leq \eta L < 1. \tag{A.4}$$

In view of Lemma 1, we can obtain

$$\boldsymbol{e}^{(t)} \leq \boldsymbol{G}_1 \boldsymbol{e}^{(t-1)}$$

with a simplified matrix

$$\boldsymbol{G}_1 := \begin{bmatrix} \theta_1 & 2\eta\beta & \eta^2 L\beta \\ \alpha\gamma\eta\beta & \alpha + \alpha\eta L & \alpha\eta L \\ 3\frac{\beta}{L} & 7 & \alpha + 2\eta\beta \end{bmatrix}, \tag{A.5}$$

where $\boldsymbol{e}^{(t)}$ is defined in (A.2). We first invoke an argument from [WYWH18] to show that $\boldsymbol{e}^{(t)}$ converges linearly at a rate not exceeding $\rho(\boldsymbol{G}_1)$. Given that $\boldsymbol{G}_1$ is a positive matrix (i.e. all of its entries are strictly

greater than zero), one can invoke the Perron-Frobenius Theorem to show that: there exists a real-valued positive number $\rho(\boldsymbol{G}_1) \in \mathbb{R}$ — the spectral radius of $\boldsymbol{G}_1$ — such that (i) $\rho(\boldsymbol{G}_1)$ is an algebraically simple eigenvalue of $\boldsymbol{G}_1$ associated with a strictly positive eigenvector $\boldsymbol{\chi}$, (ii) all other eigenvalues of $\boldsymbol{G}_1$ are strictly smaller in magnitude than $\rho(\boldsymbol{G}_1)$. Therefore, there exists some constant $C > 0$ such that $\boldsymbol{e}_0 \leq C\boldsymbol{\chi}$, and consequently,

$$\boldsymbol{e}^{(1)} \leq \boldsymbol{G}_1 \boldsymbol{e}^{(0)} \leq C\boldsymbol{G}_1\boldsymbol{\chi} = C\rho(\boldsymbol{G}_1)\boldsymbol{\chi}. \tag{A.6}$$

Invoking this argument recursively for all t, we arrive at

$$\boldsymbol{e}^{(t)} \leq C\big(\rho(\boldsymbol{G}_1)\big)^t \boldsymbol{\chi}. \tag{A.7}$$

Therefore, the rest of this proof boils down to upper bounding $\rho(\boldsymbol{G}_1)$. Rearrange the characteristic polynomial of $\boldsymbol{G}_1$, given by

$$\begin{aligned} f_1(\lambda) =& \det\left(\lambda \boldsymbol{I} - \boldsymbol{G}_1\right) \\ =& (\lambda - \theta_1)p_1(\lambda) + \alpha\gamma\eta^2\beta^2(2\alpha + 4\eta\beta - 2\theta_1 - 7\eta L) - 3\eta^2\beta^2(\alpha - \alpha\eta L + \theta_1), \end{aligned} \tag{A.8}$$

where $p_1(\lambda)$ is the following function obtained by direct computation

$$p_1(\lambda) = (\lambda - \alpha - \alpha\eta L)(\lambda - \alpha - 2\eta\beta) - 7\alpha\eta L - 2\alpha\gamma\eta^2\beta^2 - 3\eta^2\beta^2. \tag{A.9}$$

From the Perron-Frobenius Theorem, we know that $\rho(\boldsymbol{G}_1)$ is a simple positive root of $f_1(\lambda)$ (so that $f_1(\rho(\boldsymbol{G}_1)) = 0$). However, it is difficult to compute it directly. In what follows, we seek to first upper bound $\rho(\boldsymbol{G}_1)$ by

$$\rho_1 := \lambda_0 = \max\left\{\frac{1+\theta_1}{2}, \alpha + \frac{140\eta L}{1-\alpha}\Big(\frac{\beta}{\sigma} + 1\Big), \frac{1+\alpha}{2} + 2\eta\beta\right\}, \tag{A.10}$$

then demonstrate that $\lambda_0 < 1$, which ensures linear convergence, and finally replace α with α^K for the final result.

Step 1: bounding $\rho(\boldsymbol{G}_1)$ by λ_0. The following calculation aims to verify the fact that: for all $\lambda \geq \lambda_0$, one has $f_1(\lambda) > 0$, and hence $\rho(\boldsymbol{G}_1) \leq \lambda_0$. Recall the definition of θ_1 in (2.13). When $\lambda \geq \lambda_0 \geq \frac{1+\theta_1}{2}$, one has

$$\begin{aligned} \lambda - \theta_1 \geq& \frac{1-\theta_1}{2} \\ =& \frac{1}{2}\frac{\sigma}{\sigma+\mu}\Big(1 - \frac{L}{L+\mu}\frac{\beta}{\sigma+\mu-\beta}\frac{\beta}{\sigma}\Big) \\ \geq& \frac{1}{4}\frac{\sigma}{\sigma+\mu}. \end{aligned} \tag{A.11}$$

In order for the last inequality to hold, we must make sure that

$$\begin{cases} \sigma+\mu \geq \frac{3\beta^2}{\sigma}, & \text{if } \beta \geq \sigma; \\ \sigma+\mu \geq 3\sigma, & \text{otherwise.} \end{cases} \tag{A.12}$$

Note that the above relationship is guaranteed by the condition $\sigma+\mu \geq \frac{140L}{(1-\alpha)^2}\left(\frac{\beta}{\sigma}+1\right)$. When $\lambda \geq \lambda_0$, using (A.4), we can lower bound the first term of $p_1(\lambda)$ by

$$\begin{aligned}(\lambda-\alpha-\alpha\eta L)(\lambda-\alpha-2\eta\beta) \geq & \frac{1-\alpha}{2}\left(\frac{140\eta L}{1-\alpha}\left(\frac{\beta}{\sigma}+1\right)-\alpha\eta L\right) \\ > & 69\eta L\left(\frac{\beta}{\sigma}+1\right).\end{aligned}$$

We can lower bound $p_1(\lambda)$ by incorporating (A.4) as

$$\begin{aligned} p_1(\lambda) &= (\lambda-\alpha-\alpha\eta L)(\lambda-\alpha-2\eta\beta)-7\alpha\eta L-2\alpha\gamma\eta^2\beta^2-3\eta^2\beta^2 \\ &> 69\eta L\left(\frac{\beta}{\sigma}+1\right)-12\eta L \\ &> 68\kappa\eta\beta. \end{aligned} \tag{A.13}$$

As a result of (A.11) and (A.13), when $\lambda \geq \lambda_0$, the characteristic polynomial (A.8) satisfies

$$\begin{aligned} f_1(\lambda) &\geq (\lambda-\theta_1)p_1(\lambda)+\alpha\gamma\eta^2\beta^2(2\alpha+4\eta\beta-2\theta_1-7\eta L)-3\eta^2\beta^2(\alpha-\alpha\eta L+\theta_1) \\ &> \frac{1}{4}\eta\sigma\cdot 68\kappa\eta\beta-9\alpha\gamma\eta^2\beta^2-3\eta^2\beta^2(\alpha+\theta_1) \\ &> 17\eta\beta\eta L-9\alpha\gamma\eta^2\beta^2-6\eta^2\beta^2>0. \end{aligned}$$

Therefore, any λ that exceeds λ_0 cannot be a root of $f_1(\cdot)$. This implies that the spectral radius $\rho(\boldsymbol{G}_1)$, of necessity, obeys $\rho(\boldsymbol{G}_1)<\lambda_0$.

Step 2: bounding λ_0. This step verifies that all three terms in (A.10) are smaller than 1, thus leading to the conclusion $\lambda_0<1$.

- First, observe that if (A.12) is satisfied, we have $\frac{1+\theta_1}{2} \leq 1-\frac{1}{4}\eta\sigma<1$.
- When $\sigma+\mu \geq \frac{140L}{(1-\alpha)^2}\left(\frac{\beta}{\sigma}+1\right)$, the second term in (A.10) obeys $\alpha+\frac{140\eta L}{1-\alpha}\left(\frac{\beta}{\sigma}+1\right) \leq 1$.
- Finally, the third term in (A.10) is also less than 1, since

$$\frac{1+\alpha}{2}+2\eta\beta \leq \frac{1+\alpha}{2}+\frac{(1-\alpha)^2}{70}\frac{\beta}{\frac{\beta}{\sigma}+1}\frac{1}{L} \leq \frac{1+\alpha}{2}+\frac{(1-\alpha)^2}{70} \leq 1-\frac{1-\alpha}{2}+\frac{1-\alpha}{70}<1.$$

Step 3: replacing α with α^K. This step gives the final results as

$$\rho_1=\max\left\{\frac{1+\theta_1}{2}, \alpha^K+\frac{140\eta L}{1-\alpha^K}\left(\frac{\beta}{\sigma}+1\right), \frac{1+\alpha^K}{2}+2\eta\beta\right\}.$$

A.2.2 Proof of Theorem 2

By the assumption $\sigma+\mu \geq 360\sigma\Big(\frac{\beta^2}{\sigma^2}+1\Big)$ and $\alpha^K \leq \frac{1}{2\kappa}$ as we can prove that $\eta\beta<1$ and $\alpha^K\eta L \leq \frac{1}{2}$. The characteristic polynomial (A.8) in Appendix A.2.1 can then be lower bounded by

$$\begin{aligned}
f_1(\lambda) =& \det\left(\lambda \boldsymbol{I}-\boldsymbol{G}_1\right) \\
=&(\lambda-\theta_1)\Big((\lambda-\alpha^K-\alpha^K\eta L)(\lambda-\alpha^K-2\eta\beta)-7\alpha^K\eta L-2\alpha^K\gamma\eta^2\beta^2-3\eta^2\beta^2\Big) \\
&+\alpha^K\gamma\eta^2\beta^2(2\alpha^K+4\eta\beta-2\theta_1-7\eta L)-3\eta^2\beta^2(\alpha^K-\alpha^K\eta L+\theta_1) \\
\geq&(\lambda-\theta_1)\Big((\lambda-\alpha^K-\frac{1}{2}\eta\sigma)(\lambda-\alpha^K-2\eta\beta)-\frac{7}{2}\eta\sigma-\eta\sigma\eta^2\beta^2-3\eta^2\beta^2\Big) \\
&+\alpha^K\gamma\eta^2\beta^2(2\alpha^K+4\eta\beta-2\theta_1-7\eta L)-3\eta^2\beta^2(\alpha^K-\alpha^K\eta L+\theta_1), \qquad \text{(A.14)}
\end{aligned}$$

provided that λ obeys

$$\lambda \geq \max\left\{\frac{1+\theta_1}{2}, \alpha^K+180\eta\sigma\Big(\frac{\beta^2}{\sigma^2}+1\Big), \frac{1+\alpha^K}{2}+2\eta\beta\right\}.$$

Given that all conditions in (A.12) are satisfied, we can show $\eta^2\beta^2 \leq \eta\sigma\cdot\frac{\beta^2}{360\sigma^2(\beta^2/\sigma^2+1)}<\eta\sigma<1$. One can thus continue to lower bound (A.14) by

$$\begin{aligned}
f_1(\lambda) &> (\lambda-\theta_1)\Big((\lambda-\alpha^K-\frac{1}{2}\eta\sigma)(\lambda-\alpha^K-2\eta\beta)-8\eta\sigma\Big)-11\eta^2\beta^2 \\
&> \frac{1}{4}\eta\sigma\Big\{\frac{1}{4}\Big[180\eta\sigma\Big(\frac{\beta^2}{\sigma^2}+1\Big)-\frac{1}{2}\eta\sigma\Big]-8\eta\sigma\Big\}-11\eta^2\beta^2 \\
&> \frac{1}{4}\eta\sigma\Big\{45\eta\beta\frac{\beta}{\sigma}+44\eta\sigma-8\eta\sigma\Big\}-11\eta^2\beta^2 \\
&> \frac{45}{4}\eta\beta-11\eta^2\sigma^2 \\
&> 0.
\end{aligned}$$

Consequently, following similar arguments as in Appendix A.2.1, we can show that: under the conditions of Theorem 2, the spectral radius of $\boldsymbol{G}_1$ can be upper bounded by

$$\rho(\boldsymbol{G}_1) \leq 1-\frac{C}{\frac{\beta^2}{\sigma^2}+1},$$

where C is some sufficiently small positive constant. This immediately tells us that: to reach ϵ-accuracy, `Network-DANE` takes at most $O\left(\left(\frac{\beta^2}{\sigma^2}+1\right)\log(1/\epsilon)\right)$ iterations. For each iteration, `Network-DANE` needs

$$K \asymp \frac{\log(1/2\kappa)}{\log\alpha} \lesssim \frac{\log\kappa}{1-\alpha}$$

rounds of communication, where we have used the elementary inequality $1-\alpha<\log(1/\alpha)$. Putting all this together leads to a communication complexity at most $O\left(\log\kappa\cdot\frac{(\beta^2/\sigma^2+1)\log(1/\epsilon)}{1-\alpha}\right)$.

A.3 Proofs of Theorem 3 and Theorem 4

This sections establishes the convergence rate of `Network-DANE` for smooth and strongly convex loss functions, following the analysis approach adopted in the proof of Theorem 1. Similarly, we first proceed with the proof as if $K = 1$, then replace α with α^K in the last step. The following key lemma plays a crucial role, which characterizes the per-iteration dynamics of the proposed `Network-DANE` for general smooth strongly convex losses. The proof of this lemma is deferred to Appendix A.6.

Lemma 2. *Recall the notation in Lemma 1. Suppose that Assumption 3 holds, and* $\left(\frac{\beta}{\sigma+\mu}\right)^2 \le \frac{\sigma}{\sigma+2\mu}$. *One has*

$$\boldsymbol{e}^{(t)} \le \underbrace{\begin{bmatrix} \theta_2 & \eta L & \gamma\eta L \\ \alpha\gamma\eta L & \alpha+\alpha\eta L & \alpha\eta L \\ \frac{\beta}{L}+\theta_2\frac{\beta}{L}+\alpha\gamma\eta\beta & \alpha+1+\alpha\frac{\beta}{L}+\eta\beta+\alpha\frac{\beta}{L}+\alpha\eta\beta & \alpha+\gamma\eta\beta+\alpha\eta\beta \end{bmatrix}}_{=:\boldsymbol{G}'} \boldsymbol{e}^{(t-1)}. \tag{A.15}$$

Here, $\boldsymbol{e}^{(t)}$ *is the error vector defined in (A.2), and the notation* $\boldsymbol{a} \le \boldsymbol{b}$ *indicates that* $a_i \le b_i$ *for all entries* i.

A.3.1 Proof of Theorem 3

Under the conditions of Theorem 3, the inequalities stated in (A.4) remain valid. In addition, when $\sigma+\mu = \frac{170\kappa L}{(1-\alpha)^2}$, we can verify that

$$\left(\frac{\beta}{\sigma+\mu}\right)^2 = \frac{(1-\alpha)^4\beta^2}{170^2\kappa^2L^2} \le \frac{(1-\alpha)^2}{170^2\kappa^2} < \frac{1}{2}\cdot\frac{(1-\alpha)^2}{170\kappa^2} = \frac{1}{2}\cdot\frac{\sigma}{\sigma+\mu} < \frac{\sigma}{\sigma+2\mu}.$$

When $\sigma+\mu \ge \frac{170\kappa L}{(1-\alpha)^2}$, the LHS decreases faster than the RHS, thus the requirement of Lemma 2 is met. In view of Lemma 2 as well as the fact $\theta_2 \le 1$, we can replace $\boldsymbol{G}'$ by a simplified matrix that dominates $\boldsymbol{G}'$:

$$\boldsymbol{G}_2 := \begin{bmatrix} \theta_2 & 2\eta L & \gamma\eta L \\ \alpha\gamma\eta L & \alpha+\alpha\eta L & \alpha\eta L \\ 3\frac{\beta}{L} & 7 & \alpha+2\eta\beta \end{bmatrix}. \tag{A.16}$$

The above matrix $\boldsymbol{G}_2$ is similar to $\boldsymbol{G}_1$ in (A.5) in the quadratic case, except that the quantity β in the first two rows of $\boldsymbol{G}_1$ is replaced by L (thus leading to a worse convergence rate).

Similar to the proof of Theorem 1, we shall upper bound $\rho(\boldsymbol{G}_2)$ — the spectral radius of $\boldsymbol{G}_2$. To locate the eigenvalues of $\boldsymbol{G}_2$, we rearrange the characteristic polynomial of $\boldsymbol{G}_2$ as follows

$$\begin{aligned} f_2(\lambda) =& \det(\lambda\boldsymbol{I}-\boldsymbol{G}_2) \\ =& (\lambda-\theta_2)p_2(\lambda) + \alpha\gamma\eta^2L^2(2\alpha+4\eta\beta-2\theta_2-7\gamma) - 3\eta\beta(2\alpha\eta L-\gamma(\alpha+\alpha\eta L-\theta_2)), \end{aligned} \tag{A.17}$$

where $p_2(\lambda)$ is the following function obtained by direct computation

$$p_2(\lambda) = (\lambda - \alpha - \alpha\eta L)(\lambda - \alpha - 2\eta\beta) - 7\alpha\eta L - 2\alpha\gamma\eta^2 L^2 - 3\gamma\eta\beta.$$

From the Perron-Frobenius Theorem, $\rho(\boldsymbol{G}_2)$ is a simple positive root of the equation $f_2(\lambda) = 0$. However, it is hard to calculate it directly. In what follows, we seek to first upper bound $\rho(\boldsymbol{G}_2)$ by

$$\rho_2 := \lambda_0 = \max\left\{\frac{1+\theta_2}{2},\ \alpha + \frac{170\kappa\eta L}{1-\alpha}, \frac{1+\alpha}{2} + 2\eta\beta\right\}, \tag{A.18}$$

and then demonstrate that $\lambda_0 < 1$, which in turn ensures linear convergence.

Step 1: bounding $\rho(\boldsymbol{G}_2)$ by λ_0. The following calculation aims to verify the fact that $f_2(\lambda) > 0$ holds for all $\lambda \geq \lambda_0$, so that $\rho(\boldsymbol{G}_2) \leq \lambda_0$. Recalling the definition of θ_2 in Lemma 2, we see that when $\lambda \geq \lambda_0 \geq \frac{1+\theta_2}{2}$,

$$\begin{aligned}
\lambda - \theta_2 &\geq \frac{1-\theta_2}{2} \\
&= \frac{1}{2}\eta\left(\sigma - \beta\sqrt{(1-\eta\mu)(1+\eta\mu)}\right) \\
&\geq \frac{1}{2}\eta\left(\sigma - \beta\sqrt{2(1-\eta\mu)}\right) > \frac{1}{4}\eta\sigma,
\end{aligned} \tag{A.19}$$

where we have used the fact $\eta\mu < 1$ to reach the second inequality. For the last inequality to hold, we need to make sure

$$\begin{cases} \sigma + \mu \geq \frac{10\beta^2}{\sigma}, & \beta \geq \sigma \\ \sigma + \mu \geq 10\sigma, & \text{otherwise} \end{cases} \tag{A.20}$$

which is guaranteed by the assumption $\sigma + \mu \geq \frac{170\kappa L}{(1-\alpha)^2}$.

Similarly, when $\lambda \geq \lambda_0$, the first term of $p_2(\lambda)$ can be lower bounded by

$$(\lambda - \alpha - \alpha\eta L)(\lambda - \alpha - 2\eta\beta) \geq \frac{1-\alpha}{2}\left(\frac{170\kappa\eta L}{1-\alpha} - \alpha\eta L\right) > 80\kappa\eta L.$$

Then, using (A.4) we can bound $p_2(\lambda)$ by

$$\begin{aligned}
p_2(\lambda) &= (\lambda - \alpha - \alpha\eta L)(\lambda - \alpha - 2\eta\beta) - 7\alpha\eta L - 2\alpha\gamma\eta^2 L^2 - 3\gamma\eta\beta \\
&> 80\kappa\eta L - 12\eta L \geq 68\kappa\eta L.
\end{aligned} \tag{A.21}$$

By virtue of (A.19) and (A.21), it is seen that when $\lambda \geq \lambda_0$, the characteristic polynomial $f_2(\lambda)$ in (A.17) satisfies

$$f_2(\lambda) > \frac{1}{4}\eta\sigma \cdot 68\kappa\eta L - 8\alpha\gamma\eta^2 L^2 - 9\eta\beta\eta L > 0.$$

Therefore, any λ that exceeds λ_0 cannot possibly be a root of $f_2(\cdot)$. This implies that the spectral radius necessarily obeys $\rho(\boldsymbol{G}_2) < \lambda_0$.

Step 2: bounding λ_0. This step verifies that the three terms in the expression of λ_0 in (A.18) is smaller than 1, allowing us to conclude that $\lambda_0 < 1$.

- First, observe that if (A.20) is satisfied, then we have $\frac{1+\theta_2}{2} \leq 1 - \frac{1}{4}\eta\sigma < 1$.
- When $\sigma + \mu \geq \frac{170\kappa L}{(1-\alpha)^2}$, the second term is $\alpha + \frac{170\kappa\eta L}{1-\alpha} \leq 1$.
- We conclude the proof by checking that the third term is also less than 1, namely,

$$\frac{1+\alpha}{2} + 2\eta\beta \leq \frac{1+\alpha}{2} + \frac{(1-\alpha)^2}{85}\frac{1}{\kappa}\frac{\beta}{L} \leq \frac{1+\alpha}{2} + \frac{(1-\alpha)^2}{85} \leq 1 - \frac{1-\alpha}{2} + \frac{1-\alpha}{85}.$$

Step 3: replacing α with α^K. This step gives the final results as

$$\rho_2 = \max\left\{\frac{1+\theta_2}{2},\ \alpha^K + \frac{170\kappa\eta L}{1-\alpha^K},\ \frac{1+\alpha^K}{2} + 2\eta\beta\right\}.$$

A.3.2 Proof of Theorem 4

We first verify the assumption of Lemma 2. When $\sigma + \mu = 360L\left(\frac{\beta}{\sigma} + 1\right)$,

$$\left(\frac{\beta}{\sigma+\mu}\right)^2 = \frac{\beta^2}{360^2L^2(\frac{\beta}{\sigma}+1)^2} \leq \frac{\beta}{360^2\kappa L(\frac{\beta}{\sigma}+1)} < \frac{1}{2}\cdot\frac{1}{360\kappa(\frac{\beta}{\sigma}+1)} = \frac{1}{2}\cdot\frac{\sigma}{\sigma+\mu} < \frac{\sigma}{\sigma+2\mu}.$$

Therefore, Lemma 2 still holds.

By the assumption $\alpha^K \leq \frac{1}{2\kappa}$, we can further lower bound the characteristic polynomial (A.17) in Appendix A.3.1 as follows:

$$\begin{aligned}
f_2(\lambda) =\ & \det\left(\lambda \boldsymbol{I} - \boldsymbol{G}_2\right) \\
=\ & (\lambda-\theta_2)\Big((\lambda-\alpha^K-\alpha^K\eta L)(\lambda-\alpha^K-2\eta\beta) - 7\alpha^K\eta L - 2\alpha^K\gamma\eta^2L^2 - 3\gamma\eta\beta\Big) \\
& + \alpha^K\gamma\eta^2L^2\left(2\alpha^K + 4\eta\beta - 2\theta_2 - 7\gamma\right) - 3\eta\beta\left(2\alpha^K\eta L - \gamma(\alpha^K + \alpha^K\eta L - \theta_2)\right) \\
\geq\ & (\lambda-\theta_2)\Big((\lambda-\alpha^K-\frac{1}{2}\eta\sigma)(\lambda-\alpha^K-2\eta\beta) - \frac{7}{2}\eta\sigma - \eta\sigma\eta^2L^2 - 3\gamma\eta\beta\Big) \\
& - \eta\sigma\eta^2L^2(\theta_2 + \frac{7}{2}\gamma) - 3\eta\beta\Big(\eta\sigma + \gamma\theta_2\Big) \\
>\ & (\lambda-\theta_2)\Big((\lambda-\alpha^K-\frac{1}{2}\eta\sigma)(\lambda-\alpha^K-2\eta\beta) - 8\eta\sigma\Big) - 5\eta\sigma\eta^2L^2 - 6\eta\beta\eta L, \qquad \text{(A.22)}
\end{aligned}$$

providing λ obeys

$$\lambda \geq \max\left\{\frac{1+\theta_2}{2}, \alpha^K + 180\eta L\Big(\frac{\beta}{\sigma}+1\Big), \frac{1+\alpha^K}{2} + 2\eta\beta\right\}.$$

We can further lower bound (A.22) by

$$f_2(\lambda) \geq \frac{1}{4}\eta\sigma\left\{\frac{1}{4}\Big[180\eta L\Big(\frac{\beta}{\sigma}+1\Big) - \frac{1}{2}\eta\sigma\Big] - 8\eta\sigma\right\} - 5\eta\sigma\eta^2L^2 - 6\eta\beta\eta L > 0,$$

as long as μ satisfies $\sigma + \mu \geq 360L(\frac{\beta}{\sigma} + 1)$. Therefore, following similar arguments as adopted in Appendix A.3.1, the spectral radius of $\boldsymbol{G}_2$ can be upper bounded by

$$\rho(\boldsymbol{G}_2) \leq 1 - \frac{C}{\kappa(\frac{\beta}{\sigma} + 1)},$$

where C is a small positive constant. Consequently, to reach ϵ-accuracy, `Network-DANE` takes at most $O\left(\kappa(\frac{\beta}{\sigma} + 1)\log(1/\epsilon)\right)$ iterations and $O\left(\log\kappa \cdot \frac{\kappa(\beta/\sigma+1)\log(1/\epsilon)}{1-\alpha}\right)$ communication rounds.

A.4 Proof of Theorem 5

The proof strategy of Theorem 5 is similar in spirit to the convergence proof of `Network-DANE`, where we will carefully build a linear system that tracks the coupling of the consensus error and the optimization error. Under the assumptions in Theorem 5, we can assume that $1 - 3\alpha\kappa - 3\beta/\sigma > 0$. Let

$$\zeta = 1/(1 - 3\alpha\kappa - 3\beta/\sigma).$$

In what follows, we first introduce two key lemmas that connect the convergence behavior of `Network-SVRG` in the network setting to their server/client counterparts (namely, D-SVRG) studied in [CZC+20]. Lemma 3, proved in Appendix A.7, creates the linear system characterizing the iteration dynamics of `Network-SVRG`.

Lemma 3. *Under the assumptions in Theorem 5, `Network-SVRG` satisfies*

$$\mathbb{E}[\boldsymbol{e}^{(t)}] \leq \underbrace{\begin{bmatrix} \left(\nu(1 + 3\alpha\kappa + 4\frac{\beta}{\sigma}) + \frac{\beta}{\sigma}\right)\zeta & 8\frac{\beta}{\sigma}\zeta & \alpha\zeta/\kappa & \zeta/16 \\ 1/2 & 0 & 0 & 0 \\ 8(\frac{\beta}{\sigma})^2 & 64(\frac{\beta}{\sigma})^2 & 4\alpha^2 & \alpha\kappa/2 \\ 64\alpha\kappa & 0 & 0 & 0 \end{bmatrix}}_{:=\boldsymbol{G}_3} \mathbb{E}[\boldsymbol{e}^{(t-1)}], \tag{A.23}$$

where the error vector is defined as

$$\boldsymbol{e}^{(t)} = \begin{bmatrix} \sum_{j=1}^n \left(f(\boldsymbol{x}_j^{(t)}) - f(\boldsymbol{y}^\star)\right) \\ \sum_{j=1}^n \left(f(\boldsymbol{y}_j^{(t)}) - f(\boldsymbol{y}^\star)\right)/2 \\ \|\boldsymbol{s}^{(t)} - \nabla f(\boldsymbol{y}^{(t)})\|_2^2/\sigma \\ 32L\|\boldsymbol{y}^{(t)} - \boldsymbol{1}_n \otimes \overline{\boldsymbol{y}}^{(t)}\|_2^2/\alpha \end{bmatrix}.$$

Here, $\nu \leq \frac{1}{2}\frac{\sigma-2\beta}{\sigma-3\beta}$ is the convergence rate of D-SVRG in the server/client setting under the same assumptions [CZC+20, Theorem 1].

Since every term in the matrices of linear systems of Lemma 3 is non-negative, all eigenvalues of $\boldsymbol{G}_3$ are bounded by the maximum of the sum of rows according to the Gershgorin circle theorem. For

`Network-SVRG`, by setting $\alpha = \frac{1}{70\kappa}$, which needs $K \asymp O(\log_\alpha 1/\kappa) = O\big(\log\kappa/(1-\alpha)\big)$, we can ensure that the sum of the first row is bounded by $5/6$, and the sums of other rows are also bounded by a constant smaller than 1, under the assumption $\beta \le \sigma/200$. Therefore, invoking the Gershgorin circle theorem, the spectral radius is bounded by a constant smaller than 1. To achieve ε-accuracy, the total number of iterations needed is $O(\log(1/\varepsilon))$ and thus the communication complexity is $O\left(\log\kappa \cdot \frac{\log(1/\varepsilon)}{1-\alpha}\right)$.

A.5 Proof of Lemma 1

The proof is divided into several steps. (i) In Appendix A.5.1, we bound the convergence error $\sqrt{n}\|\overline{\boldsymbol{x}}^{(t)} - \boldsymbol{y}^\star\|_2$; (ii) in Appendix A.5.2, we bound the parameter consensus error $\|\boldsymbol{x}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{x}}^{(t)}\|_2$; (iii) in Appendix A.5.3, we bound the gradient estimation error $\|\boldsymbol{s}_j^{(t)} - \nabla f(\boldsymbol{y}^{(t)})\|_2$; (iv) finally, we create induction inequalities of $\|\boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t)}\|_2$, $\sqrt{n}\|\overline{\boldsymbol{y}}^{(t)} - \boldsymbol{y}^\star\|_2$ and $\|\boldsymbol{s}_j^{(t)} - \nabla f(\boldsymbol{y}^{(t)})\|_2$ in Appendix A.5.4 to conclude the proof.

A.5.1 Convergence error

We begin by defining an auxiliary variable $\boldsymbol{x}_j^+$, which can be seen as the result of one local iterate (2.9) of the original DANE algorithm initialized at $\overline{\boldsymbol{y}}^{(t-1)}$:

$$\boldsymbol{x}_j^+ = \operatorname*{argmin}_{\boldsymbol{x}} \left\{ f_j(\boldsymbol{x}) - \left\langle \nabla f_j(\overline{\boldsymbol{y}}^{(t-1)}) - \nabla f(\overline{\boldsymbol{y}}^{(t-1)}), \boldsymbol{x} \right\rangle + \frac{\mu}{2}\|\boldsymbol{x} - \overline{\boldsymbol{y}}^{(t-1)}\|_2^2 \right\}. \tag{A.24}$$

Following the same convention as in previous definitions, we also define

$$\overline{\boldsymbol{x}}^+ = \frac{1}{n}\sum_j \boldsymbol{x}_j^+. \tag{A.25}$$

Given that the function we optimize at each agent is strongly convex, the local optimality conditions of (A.24) and (2.9) are as follows:

$$\nabla f_j(\boldsymbol{x}_j^+) + \mu(\boldsymbol{x}_j^+ - \boldsymbol{y}^\star) = \nabla(f_j - f)(\overline{\boldsymbol{y}}^{(t-1)}) + \mu(\overline{\boldsymbol{y}}^{(t-1)} - \boldsymbol{y}^\star), \tag{A.26a}$$

$$\nabla f_j(\boldsymbol{x}_j^{(t-1)}) + \mu(\boldsymbol{x}_j^{(t-1)} - \boldsymbol{y}^\star) = \nabla f_j(\boldsymbol{y}_j^{(t-1)}) - \boldsymbol{s}_j^{(t-1)} + \mu(\boldsymbol{y}_j^{(t-1)} - \boldsymbol{y}^\star). \tag{A.26b}$$

Taking the average of (A.26) over $j = 1, \ldots, n$, we obtain another set of optimality conditions:

$$\frac{1}{n}\sum_j \nabla f_j(\boldsymbol{x}_j^+) + \mu(\overline{\boldsymbol{x}}^+ - \boldsymbol{y}^\star) = \mu(\overline{\boldsymbol{y}}^{(t-1)} - \boldsymbol{y}^\star), \tag{A.27a}$$

$$\frac{1}{n}\sum_j \nabla f_j(\boldsymbol{x}_j^{(t-1)}) + \mu(\overline{\boldsymbol{x}}^{(t-1)} - \boldsymbol{y}^\star) = \mu(\overline{\boldsymbol{y}}^{(t-1)} - \boldsymbol{y}^\star), \tag{A.27b}$$

where we use the fact $\sum_j \boldsymbol{s}_j^{(t-1)} = \sum_j \nabla f_j(\boldsymbol{y}_j^{(t-1)})$ due to the property of gradient tracking (2.2).

In view of the triangle inequality, the convergence error can be decomposed as

$$\|\overline{\boldsymbol{x}}^{(t-1)} - \boldsymbol{y}^\star\|_2 \leq \|\overline{\boldsymbol{x}}^{(t-1)} - \overline{\boldsymbol{x}}^{+}\|_2 + \|\overline{\boldsymbol{x}}^{+} - \boldsymbol{y}^\star\|_2, \tag{A.28}$$

where the first term is the error caused by inaccurate gradient estimate, and the second term is the progress of DANE initialized at $\overline{\boldsymbol{y}}^{(t-1)}$.

1. For the first term $\|\overline{\boldsymbol{x}}^{(t-1)} - \overline{\boldsymbol{x}}^{+}\|_2$, we first plug in the Hessian of the quadratic losses to solve for $\boldsymbol{x}_j^{(t-1)}$ and $\boldsymbol{x}_j^{+}$ explicitly as

$$\boldsymbol{x}_j^{(t-1)} = \boldsymbol{y}_j^{(t-1)} - (\boldsymbol{H}_j + \mu \boldsymbol{I}_d)^{-1} \boldsymbol{s}_j^{(t-1)}, \tag{A.29a}$$

$$\boldsymbol{x}_j^{+} = \overline{\boldsymbol{y}}^{(t-1)} - (\boldsymbol{H}_j + \mu \boldsymbol{I}_d)^{-1} \nabla f(\overline{\boldsymbol{y}}^{(t-1)}). \tag{A.29b}$$

The first error term $\|\overline{\boldsymbol{x}}^{(t-1)} - \overline{\boldsymbol{x}}^{+}\|_2$ can be written as

$$\begin{aligned}
&\|\overline{\boldsymbol{x}}^{(t-1)} - \overline{\boldsymbol{x}}^{+}\|_2 \\
=& \left\| \left(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\right)(\boldsymbol{x}^{(t-1)} - \boldsymbol{x}^{+}) \right\|_2 \\
=& \left\| \left(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\right)\left(\boldsymbol{y}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)} - (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} \nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)}) + (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} \boldsymbol{s}^{(t-1)}\right) \right\|_2 \\
=& \left\| \left(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\right)(\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1}\left(\boldsymbol{s}^{(t-1)} - \nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)})\right) \right\|_2,
\end{aligned}$$

where the last line follows from the definition of $\overline{\boldsymbol{y}}^{(t-1)}$. Then, we add and subtract $(\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} + \mu \boldsymbol{I}_{nd})^{-1}$ and rearrange terms, obtaining

$$\begin{aligned}
&\|\overline{\boldsymbol{x}}^{(t-1)} - \overline{\boldsymbol{x}}^{+}\|_2 \\
=& \left\| \left(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\right)\left((\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} - (\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} + \mu \boldsymbol{I}_{nd})^{-1}\right)\left(\boldsymbol{s}^{(t-1)} - \nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)})\right) \right. \\
&\left. + \left(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\right)(\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} + \mu \boldsymbol{I}_{nd})^{-1}\left(\boldsymbol{s}^{(t-1)} - \nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)})\right) \right\|_2 \\
=& \left\| \left(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\right)(\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1}(\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} - \boldsymbol{H})(\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} + \mu \boldsymbol{I}_{nd})^{-1}\left(\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)})\right) \right. \\
&+ \left(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\right)(\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1}(\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} - \boldsymbol{H})(\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} + \mu \boldsymbol{I}_{nd})^{-1}\left(\nabla f(\boldsymbol{y}^{(t-1)}) - \nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)})\right) \\
&\left. + \left(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\right)(\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} + \mu \boldsymbol{I}_{nd})^{-1}(\boldsymbol{H} - \boldsymbol{I}_n \otimes \overline{\boldsymbol{H}})(\boldsymbol{y}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)}) \right\|_2 \qquad \text{(A.30)} \\
\leq& \left\| \left(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\right) \right\|_2 \left\| (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1}(\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} - \boldsymbol{H})(\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} + \mu \boldsymbol{I}_{nd})^{-1} \right\|_2 \|\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)})\|_2 \\
&+ \left\| \left(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\right) \right\|_2 \left\| (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1}(\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} - \boldsymbol{H})\left(\boldsymbol{I}_{nd} + \mu \boldsymbol{I}_n \otimes \overline{\boldsymbol{H}}^{-1}\right)^{-1} \right\|_2 \|\boldsymbol{y}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)}\|_2 \\
&+ \left\| \left(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\right) \right\|_2 \left\| (\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} + \mu \boldsymbol{I}_{nd})^{-1}(\boldsymbol{H} - \boldsymbol{I}_n \otimes \overline{\boldsymbol{H}}) \right\|_2 \|\boldsymbol{y}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)}\|_2.
\end{aligned}$$

The last term in (A.30) follows from the identity

$$\left(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\right)(\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} + \mu \boldsymbol{I}_{nd})^{-1}\left(\boldsymbol{s}^{(t-1)} - \nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)})\right)$$

$$
\begin{aligned}
&=(\overline{\boldsymbol{H}}+\mu \boldsymbol{I}_d)^{-1}\Big(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\Big)\big(\boldsymbol{s}^{(t-1)}-\nabla f(\mathbf{1}_n\otimes \overline{\boldsymbol{y}}^{(t-1)})\big)\\
&=(\overline{\boldsymbol{H}}+\mu \boldsymbol{I}_d)^{-1}\Big(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\Big)\big(\boldsymbol{H}\boldsymbol{y}^{(t-1)}-\mathbf{1}_n\otimes \overline{\boldsymbol{H}}\overline{\boldsymbol{y}}^{(t-1)}\big)\\
&=(\overline{\boldsymbol{H}}+\mu \boldsymbol{I}_d)^{-1}\Big(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\Big)\big(\boldsymbol{H}\boldsymbol{y}^{(t-1)}-\mathbf{1}_n\otimes \boldsymbol{H}\overline{\boldsymbol{y}}^{(t-1)}\big)\\
&=\Big(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\Big)(\boldsymbol{I}_n\otimes \overline{\boldsymbol{H}}+\mu \boldsymbol{I}_{nd})^{-1}\boldsymbol{H}(\boldsymbol{y}^{(t-1)}-\mathbf{1}_n\otimes \overline{\boldsymbol{y}}^{(t-1)})\\
&=\Big(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\Big)(\boldsymbol{I}_n\otimes \overline{\boldsymbol{H}}+\mu \boldsymbol{I}_{nd})^{-1}(\boldsymbol{H}-\boldsymbol{I}_n\otimes \overline{\boldsymbol{H}})(\boldsymbol{y}^{(t-1)}-\mathbf{1}_n\otimes \overline{\boldsymbol{y}}^{(t-1)}).
\end{aligned}
$$

Taken together with the identity $\|\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\|_2=\frac{1}{\sqrt{n}}$, the assumption $\|\boldsymbol{H}_j-\overline{\boldsymbol{H}}\|_2\le\beta$, and the bound $\|(\boldsymbol{H}+\mu \boldsymbol{I}_{nd})^{-1}\|_2\le\frac{1}{\sigma+\mu}$ and $\Big\|\big(\boldsymbol{I}_{nd}+\mu \boldsymbol{I}_n\otimes \overline{\boldsymbol{H}}^{-1}\big)^{-1}\Big\|_2\le\frac{L}{L+\mu}$, we can further bound (A.30) by

$$
\begin{aligned}
\sqrt{n}\|\overline{\boldsymbol{x}}^{(t-1)}-\overline{\boldsymbol{x}}^{+}\|_2\le&\frac{1}{\sigma+\mu}\frac{\beta}{\sigma+\mu}\|\boldsymbol{s}^{(t-1)}-\nabla f(\boldsymbol{y}^{(t-1)})\|_2\\
&+\Big(\frac{L}{L+\mu}\frac{\beta}{\sigma+\mu}+\frac{\beta}{\sigma+\mu}\Big)\|\boldsymbol{y}^{(t-1)}-\mathbf{1}_n\otimes \overline{\boldsymbol{y}}^{(t-1)}\|_2. \qquad \text{(A.31)}
\end{aligned}
$$

2. Regarding the second term $\|\overline{\boldsymbol{x}}^{+}-\boldsymbol{y}^\star\|_2$, we provide a slightly improved bound compared to [SSZ14]. In view of (A.29b),

$$
\begin{aligned}
\|\overline{\boldsymbol{x}}^{+}-\boldsymbol{y}^\star\|_2=&\Big\|\overline{\boldsymbol{y}}^{(t-1)}-\boldsymbol{y}^\star-\frac{1}{n}\sum_j(\boldsymbol{H}_j+\mu \boldsymbol{I}_d)^{-1}\nabla f(\overline{\boldsymbol{y}}^{(t-1)})\Big\|_2\\
=&\Big\|\Big(\boldsymbol{I}-\frac{1}{n}\sum_{i=1}^n(\boldsymbol{H}_i+\mu \boldsymbol{I})^{-1}\overline{\boldsymbol{H}}\Big)(\overline{\boldsymbol{y}}^{(t-1)}-\boldsymbol{y}^\star)\Big\|_2\\
\le&\Big\|\boldsymbol{I}-\frac{1}{n}\sum_{i=1}^n(\boldsymbol{H}_i+\mu \boldsymbol{I})^{-1}\overline{\boldsymbol{H}}\Big\|_2\|\overline{\boldsymbol{y}}^{(t-1)}-\boldsymbol{y}^\star\|_2. \qquad \text{(A.32)}
\end{aligned}
$$

Then, we use the triangle inequality to break the convergence rate in (A.32) into two parts:

$$
\begin{aligned}
&\Big\|\boldsymbol{I}-\frac{1}{n}\sum_{i=1}^n(\boldsymbol{H}_i+\mu \boldsymbol{I})^{-1}\overline{\boldsymbol{H}}\Big\|_2\\
\le&\Big\|\boldsymbol{I}-(\overline{\boldsymbol{H}}+\mu \boldsymbol{I})^{-1}\overline{\boldsymbol{H}}\Big\|_2+\Big\|\frac{1}{n}\sum_{i=1}^n\Big((\boldsymbol{H}_i+\mu \boldsymbol{I})^{-1}-(\overline{\boldsymbol{H}}+\mu \boldsymbol{I})^{-1}\Big)\overline{\boldsymbol{H}}\Big\|_2. \qquad \text{(A.33)}
\end{aligned}
$$

When $\overline{\boldsymbol{H}}\succeq\sigma \boldsymbol{I}_d$, it is straightforward to check that the first term of (A.33) is upper bounded by

$$
\Big\|\boldsymbol{I}-(\overline{\boldsymbol{H}}+\mu \boldsymbol{I})^{-1}\overline{\boldsymbol{H}}\Big\|_2\le 1-\frac{\sigma}{\sigma+\mu}.
$$

Regarding the second term of (A.33), let $\boldsymbol{\Delta}_i:=\boldsymbol{H}_i-\overline{\boldsymbol{H}}$ and use the definition of β, one derives

$$
\big\|(\overline{\boldsymbol{H}}+\mu \boldsymbol{I})^{-1}\boldsymbol{\Delta}_i\big\|_2\le\big\|(\overline{\boldsymbol{H}}+\mu \boldsymbol{I})^{-1}\big\|_2\cdot\big\|\boldsymbol{\Delta}_i\big\|_2\le\frac{\beta}{\sigma+\mu}<1 \qquad \text{(A.34)}
$$

under our hypothesis $\beta<\mu+\sigma$. In addition,

$$
\Big\|\frac{1}{n}\sum_{i=1}^n\Big((\boldsymbol{H}_i+\mu \boldsymbol{I})^{-1}-(\overline{\boldsymbol{H}}+\mu \boldsymbol{I})^{-1}\Big)\overline{\boldsymbol{H}}\Big\|_2
$$

$$= \Big\| \frac{1}{n} \sum_{i=1}^{n} \Big(\sum_{m=0}^{\infty} (-1)^m [(\overline{\boldsymbol{H}} + \mu \boldsymbol{I})^{-1} \boldsymbol{\Delta}_i]^m (\overline{\boldsymbol{H}} + \mu \boldsymbol{I})^{-1} - (\overline{\boldsymbol{H}} + \mu \boldsymbol{I})^{-1} \Big) \overline{\boldsymbol{H}} \Big\|_2 \tag{A.35}$$

$$= \Big\| \frac{1}{n} \sum_{i=1}^{n} \Big(\sum_{m=2}^{\infty} (-1)^m [(\overline{\boldsymbol{H}} + \mu \boldsymbol{I})^{-1} \boldsymbol{\Delta}_i]^m (\overline{\boldsymbol{H}} + \mu \boldsymbol{I})^{-1} \Big) \overline{\boldsymbol{H}} \Big\|_2 \tag{A.36}$$

$$\leq \frac{1}{n} \sum_{i=1}^{n} \sum_{m=2}^{\infty} \|(\overline{\boldsymbol{H}} + \mu \boldsymbol{I})^{-1}\|_2^m \cdot \|\boldsymbol{\Delta}_i\|_2^m \cdot \big\|(\boldsymbol{I} + \mu \overline{\boldsymbol{H}}^{-1})^{-1}\big\|_2$$

$$\leq \sum_{m=2}^{\infty} (\sigma + \mu)^{-m} \beta^m \frac{L}{L + \mu} = \frac{L}{L + \mu} \frac{\beta^2}{(\sigma + \mu)(\sigma + \mu - \beta)}.$$

Here, the line (A.35) is an expansion based on the Neumann series (whose convergence is guaranteed by (A.34))

$$(\boldsymbol{H}_i + \mu \boldsymbol{I})^{-1} = (\overline{\boldsymbol{H}} + \mu \boldsymbol{I} + \boldsymbol{\Delta}_i)^{-1} = \big(\boldsymbol{I} + (\overline{\boldsymbol{H}} + \mu \boldsymbol{I})^{-1} \boldsymbol{\Delta}_i\big)^{-1} (\overline{\boldsymbol{H}} + \mu \boldsymbol{I})^{-1}$$
$$= \left\{ \sum_{m=0}^{\infty} (-1)^m \big[(\overline{\boldsymbol{H}} + \mu \boldsymbol{I})^{-1} \boldsymbol{\Delta}_i\big]^m \right\} (\overline{\boldsymbol{H}} + \mu \boldsymbol{I})^{-1}.$$

The identity (A.36) holds since $\sum_{i=1}^{n} \boldsymbol{\Delta}_i = \boldsymbol{0}$, and hence the summation in (A.36) effectively starts at $m = 2$.

Putting the above two bounds together back in (A.33), we arrive at

$$\Big\| \boldsymbol{I} - \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{H}_i + \mu \boldsymbol{I})^{-1} \overline{\boldsymbol{H}} \Big\|_2 \leq \theta_1 = 1 - \frac{\sigma}{\sigma + \mu} + \frac{L}{L + \mu} \frac{\beta^2}{(\sigma + \mu)(\sigma + \mu - \beta)}. \tag{A.37}$$

Putting together (A.31) and (A.37), and plugging back into (A.28), we can bound the convergence error by:

$$\begin{aligned} \sqrt{n} \|\overline{\boldsymbol{y}}^{(t)} - \boldsymbol{y}^\star\|_2 &= \sqrt{n} \|\overline{\boldsymbol{x}}^{(t-1)} - \boldsymbol{y}^\star\|_2 \\ &\leq \theta_1 \sqrt{n} \|\overline{\boldsymbol{y}}^{(t-1)} - \boldsymbol{y}^\star\|_2 + \frac{1}{\sigma + \mu} \frac{\beta}{\sigma + \mu} \|\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)})\|_2 \\ &\quad + \Big(\frac{L}{L + \mu} \frac{\beta}{\sigma + \mu} + \frac{\beta}{\sigma + \mu} \Big) \|\boldsymbol{y}^{(t-1)} - \boldsymbol{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)}\|_2. \end{aligned} \tag{A.38}$$

A.5.2 Consensus error

Using the identity $\overline{\boldsymbol{y}}^{(t)} = \Big(\frac{1}{n} \boldsymbol{1}_n^\top \otimes \boldsymbol{I}_d\Big) \boldsymbol{y}^{(t)}$ and the update rule (A.1c), we can demonstrate that

$$\begin{aligned} &\Big\| \boldsymbol{y}^{(t)} - \boldsymbol{1}_n \otimes \overline{\boldsymbol{y}}^{(t)} \Big\|_2 \\ &= \Big\| \Big(\boldsymbol{I}_{nd} - \frac{1}{n} \boldsymbol{1}_n \boldsymbol{1}_n^\top \otimes \boldsymbol{I}_d \Big) \boldsymbol{y}^{(t)} \Big\|_2 \\ &= \Big\| \Big(\boldsymbol{I}_{nd} - \frac{1}{n} \boldsymbol{1}_n \boldsymbol{1}_n^\top \otimes \boldsymbol{I}_d \Big) (\boldsymbol{W} \otimes \boldsymbol{I}_d) \Big(\boldsymbol{y}^{(t-1)} - (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} \boldsymbol{s}^{(t-1)} \Big) \Big\|_2 \\ &\leq \Big\| \Big(\boldsymbol{W}^K - \frac{1}{n} \boldsymbol{1}_n \boldsymbol{1}_n^\top \Big) \otimes \boldsymbol{I}_d \Big\|_2 \Big\| \boldsymbol{y}^{(t-1)} - \boldsymbol{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)} - \Big(\boldsymbol{I}_{nd} - \frac{1}{n} \boldsymbol{1}_n \boldsymbol{1}_n^\top \otimes \boldsymbol{I}_d \Big) \Big((\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} \boldsymbol{s}^{(t-1)} \Big) \Big\|_2 \end{aligned} \tag{A.39}$$

$$\leq \alpha \|\boldsymbol{y}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)}\|_2 + \alpha \left\| \left(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right) (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} \boldsymbol{s}^{(t-1)} \right\|_2, \tag{A.40}$$

where (A.39) is due to the following equality:

$$\left(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right) (\boldsymbol{W} \otimes \boldsymbol{I}_d) = \left[\left(\boldsymbol{W}^K - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \right) \otimes \boldsymbol{I}_d \right] \left(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right),$$

which holds because the property of the averaging operator $\left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\right)$,

$$\left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right) \left(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right) = \left[\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \left(\boldsymbol{I}_d - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right) \right] \otimes \boldsymbol{I}_n = \mathbf{0},$$

and the fact that $(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) = (\boldsymbol{AC}) \otimes (\boldsymbol{BD})$.

We rearrange the second term in (A.40) as

$$\begin{aligned}
&\left\| \left(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right) (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} \boldsymbol{s}^{(t-1)} \right\|_2 \\
=&\Big\| \left(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right) (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} \left(\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)}) \right) \\
&+ \left(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right) (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} \left(\nabla f(\boldsymbol{y}^{(t-1)}) - \nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)}) \right) \\
&+ \left(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right) (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} \left(\nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)}) - \nabla f(\mathbf{1}_n \otimes \boldsymbol{y}^\star) \right) \Big\|_2 \\
=&\Big\| \left(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right) (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} \left(\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)}) \right) \\
&+ \left(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right) (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} (\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}}) (\boldsymbol{y}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)}) \\
&+ \left(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right) \left((\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} - (\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} + \mu \boldsymbol{I}_{nd})^{-1} \right) (\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}}) (\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)} - \mathbf{1}_n \otimes \boldsymbol{y}^\star) \Big\|_2.
\end{aligned}$$

Using similar trick as in (A.30), the above quantity can be further upper bounded as

$$\begin{aligned}
&\left\| \left(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right) (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} \boldsymbol{s}^{(t-1)} \right\|_2 \\
\leq&\left\| \boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right\|_2 \left\| (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} \right\|_2 \left\| \boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)}) \right\|_2 \\
&+ \left\| \boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right\|_2 \left\| (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} \right\|_2 \left\| \boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} \right\|_2 \left\| \boldsymbol{y}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)} \right\|_2 \\
&+ \sqrt{n} \left\| \boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d \right\|_2 \left\| (\boldsymbol{H} + \mu \boldsymbol{I}_{nd})^{-1} (\boldsymbol{I}_n \otimes \overline{\boldsymbol{H}} - \boldsymbol{H}) (\boldsymbol{I}_{nd} + \mu \boldsymbol{I}_n \otimes \overline{\boldsymbol{H}}^{-1})^{-1} \right\|_2 \left\| \overline{\boldsymbol{y}}^{(t-1)} - \boldsymbol{y}^\star \right\|_2.
\end{aligned} \tag{A.41}$$

Combine (A.40) and (A.41), we conclude that

$$\begin{aligned}
\left\| \boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t)} \right\|_2 \leq& \left(\alpha + \frac{\alpha L}{\sigma + \mu} \right) \left\| \boldsymbol{y}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)} \right\|_2 + \frac{\alpha}{\sigma + \mu} \left\| \boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)}) \right\|_2 \\
&+ \frac{\alpha L}{L + \mu} \frac{\beta}{\sigma + \mu} \sqrt{n} \left\| \overline{\boldsymbol{y}}^{(t-1)} - \boldsymbol{y}^\star \right\|_2.
\end{aligned} \tag{A.42}$$

A.5.3 Gradient estimation error

In view of the fundamental theorem of calculus and the definition of β, it holds that

$$\left\| \nabla(f - f_j)(\boldsymbol{x}) - \nabla(f - f_j)(\boldsymbol{y}) \right\|_2 = \left\| \left[\int_0^1 \nabla^2 (f - f_j)(c\boldsymbol{x} + (1-c)\boldsymbol{y}) dc \right] (\boldsymbol{x} - \boldsymbol{y}) \right\|_2 \leq \beta \|\boldsymbol{x} - \boldsymbol{y}\|_2.$$

To begin, the update formulas (2.7) and (2.8) are equivalent to

$$\boldsymbol{y}^{(t)} = (\boldsymbol{W} \otimes \boldsymbol{I}_d)\boldsymbol{x}^{(t-1)}, \tag{A.43}$$

$$\boldsymbol{s}^{(t)} = (\boldsymbol{W} \otimes \boldsymbol{I}_d)\boldsymbol{s}^{(t-1)} + \nabla F(\boldsymbol{y}^{(t)}) - \nabla F(\boldsymbol{y}^{(t-1)}). \tag{A.44}$$

Note that, since

$$\begin{aligned}\left(\boldsymbol{W} - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right)^K &= \left(\boldsymbol{W} - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right)\left(\boldsymbol{W} - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right)\cdots\left(\boldsymbol{W} - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right) \\ &= \left(\boldsymbol{W}^2 - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right)\cdots\left(\boldsymbol{W} - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right) = \boldsymbol{W}^K - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top,\end{aligned}$$

we have the mixing rate of $\boldsymbol{W}^K$ is

$$\alpha := \|\boldsymbol{W}^K - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\| = \|\boldsymbol{W} - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\|^K = \alpha_0^K.$$

In view of the equivalent update rule (A.44),

$$\begin{aligned}\|\boldsymbol{s}^{(t)} - \nabla f(\boldsymbol{y}^{(t)})\|_2 =& \Big\|(\boldsymbol{W} \otimes \boldsymbol{I}_d)\boldsymbol{s}^{(t-1)} + \nabla F(\boldsymbol{y}^{(t)}) - \nabla F(\boldsymbol{y}^{(t-1)}) - \nabla f(\boldsymbol{y}^{(t)})\Big\|_2 \\ =& \Big\|(\boldsymbol{W} \otimes \boldsymbol{I}_d)\Big(\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)})\Big) + (\boldsymbol{W} \otimes \boldsymbol{I}_d)\nabla f(\boldsymbol{y}^{(t-1)}) \\ &+ \nabla F(\boldsymbol{y}^{(t)}) - \nabla F(\boldsymbol{y}^{(t-1)}) - \nabla f(\boldsymbol{y}^{(t)})\Big\|_2 \\ =& \Big\|(\boldsymbol{W} \otimes \boldsymbol{I}_d)\Big(\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)})\Big) + \nabla(F - f)(\boldsymbol{y}^{(t)}) \\ &+ (\boldsymbol{W} \otimes \boldsymbol{I}_d)\nabla f(\boldsymbol{y}^{(t-1)}) - \nabla F(\boldsymbol{y}^{(t-1)})\Big\|_2\end{aligned}$$

Subtract and add $\Big((\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\Big)\Big(\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)})\Big)$, $\nabla(f - F)(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t)})$ and $\nabla(f - F)(\mathbf{1}_n \otimes \boldsymbol{y}^\star)$ to the previous equation, and rearrange terms,

$$\begin{aligned}\|\boldsymbol{s}^{(t)} - \nabla f(\boldsymbol{y}^{(t)})\|_2 =& \Big\|\Big[(\boldsymbol{W} \otimes \boldsymbol{I}_d) - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\Big]\Big(\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)})\Big) \\ &+ \nabla(F - f)(\boldsymbol{y}^{(t)}) - \nabla(F - f)(\mathbf{1}_n \otimes \boldsymbol{y}^\star) \\ &+ (\boldsymbol{W} \otimes \boldsymbol{I}_d)\Big(\nabla f(\boldsymbol{y}^{(t-1)}) - \nabla f(\mathbf{1}_n \otimes \boldsymbol{y}^\star)\Big) - \Big[\nabla F(\boldsymbol{y}^{(t-1)}) - \nabla F(\mathbf{1}_n \otimes \boldsymbol{y}^\star)\Big] \\ &+ \Big[(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\Big]\Big(\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)})\Big)\Big\|_2 \\ \le& \alpha\|\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)})\|_2 + \beta\|\boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \boldsymbol{y}^\star\|_2 \\ &+ \Big\|(\boldsymbol{W} \otimes \boldsymbol{I}_d)\Big(\nabla f(\boldsymbol{y}^{(t-1)}) - \nabla f(\mathbf{1}_n \otimes \boldsymbol{y}^\star)\Big) - \Big[\nabla F(\boldsymbol{y}^{(t-1)}) - \nabla F(\mathbf{1}_n \otimes \boldsymbol{y}^\star)\Big] \\ &+ \Big[(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\Big]\Big(\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)})\Big)\Big\|_2.\end{aligned} \tag{A.45}$$

Using the facts $\Big[(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\Big]\boldsymbol{s}^{(t-1)} = \Big[(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\Big]\nabla F(\boldsymbol{y}^{(t-1)})$ and $\Big[(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\Big]\nabla(F - f)(\mathbf{1}_n \otimes \boldsymbol{y}^\star) = \mathbf{0}$, the last term of (A.45) becomes

$$\Big\|\Big[(\boldsymbol{W} \otimes \boldsymbol{I}_d) - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\Big]\Big(\nabla(f - F)(\boldsymbol{y}^{(t-1)}) - \nabla(f - F)(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)})\Big)$$

$$
\begin{aligned}
&+\Big(\nabla(f-F)(\mathbf{1}_n\otimes\overline{\boldsymbol{y}}^{(t-1)})-\nabla(f-F)(\mathbf{1}_n\otimes\boldsymbol{y}^\star)\Big)\\
&+\Big[(\boldsymbol{W}\otimes\boldsymbol{I}_d)-\boldsymbol{I}_{nd}\Big]\Big(\nabla F(\boldsymbol{y}^{(t-1)})-\nabla F(\mathbf{1}_n\otimes\overline{\boldsymbol{y}}^{(t-1)})\Big)\Big\|_2\\
\le&\Big\|(\boldsymbol{W}\otimes\boldsymbol{I}_d)-(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\boldsymbol{I}_d\Big\|_2\|\nabla(f-F)(\boldsymbol{y}^{(t-1)})-\nabla(f-F)(\mathbf{1}_n\otimes\overline{\boldsymbol{y}}^{(t-1)})\|_2\\
&+\|\nabla(f-F)(\mathbf{1}_n\otimes\overline{\boldsymbol{y}}^{(t-1)})-\nabla(f-F)(\mathbf{1}_n\otimes\boldsymbol{y}^\star)\|_2\\
&+\Big\|(\boldsymbol{W}\otimes\boldsymbol{I}_d)-\boldsymbol{I}_{nd}\Big\|_2\|\nabla F(\boldsymbol{y}^{(t-1)})-\nabla F(\mathbf{1}_n\otimes\overline{\boldsymbol{y}}^{(t-1)})\|_2\\
\le&\alpha\beta\|\boldsymbol{y}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{y}}^{(t-1)}\|_2+\beta\sqrt{n}\|\overline{\boldsymbol{y}}^{(t-1)}-\boldsymbol{y}^\star\|_2+(\alpha+1)L\|\boldsymbol{y}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{y}}^{(t-1)}\|_2. \qquad \text{(A.46)}
\end{aligned}
$$

We used $\Big\|(\boldsymbol{W}\otimes\boldsymbol{I}_d)-\boldsymbol{I}_{nd}\Big\|_2=\Big\|(\boldsymbol{W}\otimes\boldsymbol{I}_d)-\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes\boldsymbol{I}_d\Big)+\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes\boldsymbol{I}_d\Big)-\boldsymbol{I}_{nd}\Big\|_2\le\Big\|(\boldsymbol{W}\otimes\boldsymbol{I}_d)-\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes\boldsymbol{I}_d\Big)\Big\|_2+\Big\|\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes\boldsymbol{I}_d\Big)-\boldsymbol{I}_{nd}\Big\|_2\le\alpha+1$ to obtain the last inequality.

Combining (A.45) and (A.46), we obtain the bound

$$
\begin{aligned}
\|\boldsymbol{s}^{(t)}-\nabla f(\boldsymbol{y}^{(t)})\|_2\le&\alpha\Big\|\boldsymbol{s}^{(t-1)}-\nabla f(\boldsymbol{y}^{(t-1)})\Big\|_2+\beta\|\boldsymbol{y}^{(t)}-\mathbf{1}_n\otimes\overline{\boldsymbol{y}}^{(t)}\|_2+\beta\sqrt{n}\|\overline{\boldsymbol{y}}^{(t)}-\boldsymbol{y}^\star\|_2\\
&+\big(\alpha\beta+(\alpha+1)L\big)\|\boldsymbol{y}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{y}}^{(t-1)}\|_2+\beta\sqrt{n}\|\overline{\boldsymbol{y}}^{(t-1)}-\boldsymbol{y}^\star\|_2. \qquad \text{(A.47)}
\end{aligned}
$$

A.5.4 Linear system

Recall the definitions $\eta=\frac{1}{\sigma+\mu}$, $\gamma=\frac{L}{L+\sigma}$ and the error vector (2.13). Combining (A.38), (A.42) and (A.47) leads to the matrix $\boldsymbol{G}$ defined in (A.3).

A.6 Proof of Lemma 2

The proof follows the same procedures as the proof of Lemma 1. (i) In Appendix A.6.1, we bound the convergence error $\sqrt{n}\|\overline{\boldsymbol{y}}^{(t)}-\boldsymbol{y}^\star\|_2$; (ii) in Appendix A.6.2, we bound the parameter consensus error $\|\boldsymbol{y}^{(t)}-\mathbf{1}_n\otimes\overline{\boldsymbol{y}}^{(t)}\|_2$; (iii) finally, using the bound we obtained in Appendix A.5.3 of the gradient estimation error, we create induction inequalities of $\|\boldsymbol{y}^{(t)}-\mathbf{1}_n\otimes\overline{\boldsymbol{y}}^{(t)}\|_2$, $\sqrt{n}\|\overline{\boldsymbol{y}}^{(t)}-\boldsymbol{y}^\star\|_2$ and $L^{-1}\|\boldsymbol{s}_j^{(t)}-\nabla f(\boldsymbol{y}^{(t)})\|_2$ in Appendix A.6.3 to conclude the proof. For consistency and simplicity, we use the same definitions of $\boldsymbol{x}^+$ in (A.25), $\eta=\frac{1}{\sigma+\mu}$, and $\gamma=\frac{L}{L+\sigma}$ as in the proof of Lemma 1.

A.6.1 Convergence error

We continue to decompose the convergence error as (A.28), and bound the two terms respectively.

1. For the term $\|\overline{\boldsymbol{x}}^{(t-1)}-\overline{\boldsymbol{x}}^+\|_2$, we first subtract (A.26a) from (A.26b), which gives

$$
\nabla f_j(\boldsymbol{x}_j^{(t-1)})-\nabla f_j(\boldsymbol{x}_j^+)+\mu(\boldsymbol{x}_j^{(t-1)}-\boldsymbol{x}_j^+)=\nabla f(\boldsymbol{y}_j^{(t-1)})-\boldsymbol{s}_j^{(t-1)}
$$

$$+\nabla(f-f_j)(\overline{\boldsymbol{y}}^{(t-1)})-\nabla(f-f_j)(\boldsymbol{y}_j^{(t-1)})+\mu(\boldsymbol{y}_j^{(t-1)}-\overline{\boldsymbol{y}}^{(t-1)}),$$

then use the strong convexity of $f_j(\cdot)$ and the definition of β to bound both sides,

$$\begin{aligned}
&\|\nabla f_j(\boldsymbol{x}_j^{(t-1)})-\nabla f_j(\boldsymbol{x}_j^{+})+\mu(\boldsymbol{x}_j^{(t-1)}-\boldsymbol{x}_j^{+})\|_2\geq(\sigma+\mu)\|\boldsymbol{x}_j^{(t-1)}-\boldsymbol{x}_j^{+}\|_2,\\
&\left\|\nabla f(\boldsymbol{y}_j^{(t-1)})-\boldsymbol{s}_j^{(t-1)}+\nabla(f-f_j)(\overline{\boldsymbol{y}}^{(t-1)})-\nabla(f-f_j)(\boldsymbol{y}_j^{(t-1)})+\mu(\boldsymbol{y}_j^{(t-1)}-\overline{\boldsymbol{y}}^{(t-1)})\right\|_2\\
\leq&(\beta+\mu)\|\boldsymbol{y}_j^{(t-1)}-\overline{\boldsymbol{y}}^{(t-1)}\|_2+\|\nabla f(\boldsymbol{y}_j^{(t-1)})-\boldsymbol{s}_j^{(t-1)}\|_2.
\end{aligned}$$

Therefore, combining the above two inequalities, we have

$$\|\boldsymbol{x}_j^{(t-1)}-\boldsymbol{x}_j^{+}\|_2\leq\frac{1}{\sigma+\mu}\|\nabla f(\boldsymbol{y}_j^{(t-1)})-\boldsymbol{s}_j^{(t-1)}\|_2+\frac{\beta+\mu}{\sigma+\mu}\|\boldsymbol{y}_j^{(t-1)}-\overline{\boldsymbol{y}}^{(t-1)}\|_2. \tag{A.48}$$

Subtracting the optimality conditions in (A.27),

$$\begin{aligned}
\boldsymbol{0}\in&\frac{1}{n}\sum_j\nabla f_j(\boldsymbol{x}_j^{(t-1)})-\frac{1}{n}\sum_j\nabla f_j(\boldsymbol{x}_j^{+})+\mu(\overline{\boldsymbol{x}}^{(t-1)}-\overline{\boldsymbol{x}}^{+})\\
=&\frac{1}{n}\sum_j\left(\nabla f_j(\boldsymbol{x}_j^{(t-1)})-L\boldsymbol{x}_j^{(t-1)}\right)-\frac{1}{n}\sum_j\left(\nabla f_j(\boldsymbol{x}_j^{+})-L\boldsymbol{x}_j^{+}\right)+(L+\mu)(\overline{\boldsymbol{x}}^{(t-1)}-\overline{\boldsymbol{x}}^{+}).
\end{aligned}$$

Note the gradient of the function $L\boldsymbol{x}-\nabla f_j(\boldsymbol{x})$ is a $(L-\sigma)$-Lipschitz function. Taking the ℓ_2 norm and plugging in (A.48), we have

$$\begin{aligned}
\|\overline{\boldsymbol{x}}^{(t-1)}-\overline{\boldsymbol{x}}^{+}\|_2\leq&\frac{1}{L+\mu}\left\|\frac{1}{n}\sum_j\left(\left[L\boldsymbol{x}_j^{(t-1)}-\nabla f_j(\boldsymbol{x}_j^{(t-1)})\right]-\left[L\boldsymbol{x}_j^{+}-\nabla f_j(\boldsymbol{x}_j^{+})\right]\right)\right\|_2\\
\leq&\frac{1}{L+\mu}\frac{1}{n}\sum_j\left\|\left[L\boldsymbol{x}_j^{(t-1)}-\nabla f_j(\boldsymbol{x}_j^{(t-1)})\right]-\left[L\boldsymbol{x}_j^{+}-\nabla f_j(\boldsymbol{x}_j^{+})\right]\right\|_2\\
\leq&\frac{L-\sigma}{L+\mu}\frac{1}{n}\sum_j\|\boldsymbol{x}_j^{(t-1)}-\boldsymbol{x}_j^{+}\|_2\\
\leq&\frac{L-\sigma}{L+\mu}\frac{1}{\sigma+\mu}\frac{1}{n}\sum_j\|\nabla f(\boldsymbol{y}_j^{(t-1)})-\boldsymbol{s}_j^{(t-1)}\|_2+\frac{L-\sigma}{L+\mu}\frac{\beta+\mu}{\sigma+\mu}\frac{1}{n}\sum_j\|\boldsymbol{y}_j^{(t-1)}-\overline{\boldsymbol{y}}^{(t-1)}\|_2,
\end{aligned} \tag{A.49}$$

where the last line follows (A.48).

2. For the second term $\|\overline{\boldsymbol{x}}^{+}-\boldsymbol{y}^{\star}\|_2$, because of the assumption $\left(\frac{\beta}{\sigma+\mu}\right)^2\leq\frac{\sigma}{\sigma+2\mu}$, we can invoke [FGW21, Theorem 3.1], which is a careful analysis of the error of DANE, and bound the error as

$$\|\overline{\boldsymbol{x}}^{+}-\boldsymbol{y}^{\star}\|_2\leq\frac{\frac{\beta}{\sigma+\mu}\sqrt{\sigma^2+2\sigma\mu}+\mu}{\sigma+\mu}\|\overline{\boldsymbol{y}}-\boldsymbol{y}^{\star}\|_2:=\theta_2\|\overline{\boldsymbol{y}}^{(t-1)}-\boldsymbol{y}^{\star}\|_2. \tag{A.50}$$

Putting together (A.49) and (A.50), and plugging back into (A.28), we can bound the convergence error by:

$$\begin{aligned}
\sqrt{n}\|\overline{\boldsymbol{y}}^{(t)}-\boldsymbol{y}^{\star}\|_2=&\sqrt{n}\|\overline{\boldsymbol{x}}^{(t-1)}-\boldsymbol{y}^{\star}\|_2\\
\leq&\theta_2\sqrt{n}\|\overline{\boldsymbol{y}}^{(t-1)}-\boldsymbol{y}^{\star}\|_2+\frac{1}{L+\mu}\frac{L}{\sigma+\mu}\|\nabla f(\boldsymbol{y}^{(t-1)})-\boldsymbol{s}^{(t-1)}\|_2\\
&+\frac{\beta+\mu}{L+\mu}\frac{L}{\sigma+\mu}\|\boldsymbol{y}^{(t-1)}-\boldsymbol{1}_n\otimes\overline{\boldsymbol{y}}^{(t-1)}\|_2.
\end{aligned} \tag{A.51}$$

A.6.2 Consensus error

Let $\boldsymbol{H}_j^{(t)} = \int_0^1 \nabla^2 f_j(c\boldsymbol{x}_j^{(t)} + (1-c)\boldsymbol{y}_j^{(t)})\,\mathrm{d}c$ and $\boldsymbol{H}^{(t)} = \mathrm{diag}(\boldsymbol{H}_1^{(t)}, \boldsymbol{H}_2^{(t)}, \ldots, \boldsymbol{H}_n^{(t)})$. Via the fundamental theorem of calculus, we can solve for $\boldsymbol{x}_j^{(t-1)}$ from the optimality condition (A.26b) as

$$\boldsymbol{x}_j^{(t-1)} = \boldsymbol{y}_j^{(t-1)} - (\boldsymbol{H}_j^{(t-1)} + \mu \boldsymbol{I}_d)^{-1}\boldsymbol{s}_j^{(t-1)}. \tag{A.52}$$

Similar to (A.40), we decompose the consensus error as

$$\|\boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t)}\|_2 \leq \alpha\|\boldsymbol{y}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)}\|_2 + \alpha\left\|\left(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\right)(\boldsymbol{H}^{(t-1)} + \mu \boldsymbol{I}_{nd})^{-1}\boldsymbol{s}^{(t-1)}\right\|_2 \tag{A.53}$$

Then, we bound (A.53). Adding and subtracting terms and using the triangle inequality,

$$\begin{aligned}
&\left\|\left(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\right)(\boldsymbol{H}^{(t-1)} + \mu \boldsymbol{I}_{nd})^{-1}\boldsymbol{s}^{(t-1)}\right\|_2 \\
\leq &\left\|\left(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\right)(\boldsymbol{H}^{(t-1)} + \mu \boldsymbol{I}_{nd})^{-1}\left(\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)}) + \nabla f(\boldsymbol{y}^{(t-1)}) - \nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)})\right)\right\|_2 \\
&+ \left\|\left(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\right)(\boldsymbol{H}^{(t-1)} + \mu \boldsymbol{I}_{nd})^{-1}\nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)})\right\|_2
\end{aligned} \tag{A.54}$$

We can bound the first term in (A.54) as

$$\begin{aligned}
&\left\|\left(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\right)(\boldsymbol{H}^{(t-1)} + \mu \boldsymbol{I}_{nd})^{-1}\left(\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)}) + \nabla f(\boldsymbol{y}^{(t-1)}) - \nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)})\right)\right\|_2 \\
\leq &\left\|\left(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\right)(\boldsymbol{H}^{(t-1)} + \mu \boldsymbol{I}_{nd})^{-1}\right\|_2 \|\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)}) + \nabla f(\boldsymbol{y}^{(t-1)}) - \nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)})\|_2 \\
\leq &\frac{1}{\sigma+\mu}\left(\|\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)})\|_2 + \|\nabla f(\boldsymbol{y}^{(t-1)}) - \nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)})\|_2\right) \\
\leq &\frac{1}{\sigma+\mu}\left(\|\boldsymbol{s}^{(t-1)} - \nabla f(\boldsymbol{y}^{(t-1)})\|_2 + L\|\boldsymbol{y}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)}\|_2\right)
\end{aligned} \tag{A.55}$$

Then, for the second term in (A.54),

$$\begin{aligned}
&\left\|\left(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\right)(\boldsymbol{H}^{(t-1)} + \mu \boldsymbol{I}_{nd})^{-1}\nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)})\right\|_2 \\
= &\left\|\left(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\right)\left((\boldsymbol{H}^{(t-1)} + \mu \boldsymbol{I}_{nd})^{-1} - \left((L+\mu)\boldsymbol{I}_{nd}\right)^{-1}\right)\nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)})\right\|_2 \\
\leq &\left\|(\boldsymbol{H}^{(t-1)} + \mu \boldsymbol{I}_{nd})^{-1}(L\boldsymbol{I}_{nd} - \boldsymbol{H}^{(t-1)})\left((L+\mu)\boldsymbol{I}_{nd}\right)^{-1}\nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)})\right\|_2 \\
\leq &\frac{L-\sigma}{L+\mu}\frac{L}{\sigma+\mu}\sqrt{n}\|\overline{\boldsymbol{y}}^{(t-1)} - \boldsymbol{y}^\star\|_2
\end{aligned} \tag{A.56}$$

Therefore, by combing (A.53), (A.54), (A.55) and (A.56), we can bound the consensus error by:

$$\begin{aligned}
\|\boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t)}\|_2 \leq &\left(\alpha + \frac{\alpha L}{\sigma+\mu}\right)\|\boldsymbol{y}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)}\|_2 \\
&+ \frac{\alpha}{\sigma+\mu}\|\nabla f(\boldsymbol{y}^{(t-1)}) - \boldsymbol{s}^{(t-1)}\|_2 + \frac{\alpha L}{L+\mu}\frac{L}{\sigma+\mu}\sqrt{n}\|\overline{\boldsymbol{y}}^{(t-1)} - \boldsymbol{y}^\star\|_2.
\end{aligned} \tag{A.57}$$

A.6.3 Linear system

Combining (A.47), (A.57), (A.51), we reach the matrix claimed in (A.15).

A.7 Proof of Lemma 3

The proof follows similar procedures as the proof of Lemma 1. (i) In Appendix A.7.1, we bound the expected function value convergence errors $\mathbb{E}\big[\sum_{j=1}^n \big(f(\boldsymbol{x}_j^{(t)}) - f(\boldsymbol{y}^\star)\big)\big]$ and $\mathbb{E}\big[\sum_{j=1}^n \big(f(\boldsymbol{y}_j^{(t)}) - f(\boldsymbol{y}^\star)\big)\big]$; (ii) in Appendix A.7.2, we bound the expected parameter consensus error $\mathbb{E}\|\boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t)}\|_2^2$; (iii) in Appendix A.7.3, we bound the expected parameter consensus error $\mathbb{E}\|\boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t)}\|_2^2$; (iv) finally, we create induction inequalities of $\mathbb{E}\big[\sum_{j=1}^n \big(f(\boldsymbol{x}_j^{(t)}) - f(\boldsymbol{y}^\star)\big)\big]$, $\mathbb{E}\big[\sum_{j=1}^n \big(f(\boldsymbol{y}_j^{(t)}) - f(\boldsymbol{y}^\star)\big)\big]$, $\mathbb{E}\|\boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t)}\|_2^2$ and $\mathbb{E}\|\boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t)}\|_2^2$ to conclude the proof. Expectations in this section are conditioned on $\boldsymbol{x}^{(t-1)}$, $\boldsymbol{y}^{(t-1)}$ and $\boldsymbol{s}^{(t-1)}$, if not specified.

A.7.1 Function value convergence error

First, we bound the function value convergence error of $\boldsymbol{y}^{(t)}$ using the previous estimate $\boldsymbol{x}^{(t-1)}$. By the strong convexity of $f(\cdot)$ and the assumption of $\alpha \le 1/\kappa$,

$$\begin{aligned}
\sum_{j=1}^n f(\boldsymbol{y}_j^{(t)}) \le& n f(\overline{\boldsymbol{y}}^{(t-1)}) + \frac{L}{2}\|\boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t)}\|_2^2 \\
\le& n f(\overline{\boldsymbol{x}}^{(t-1)}) + \frac{\alpha^2 L}{2}\|\boldsymbol{x}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{x}}^{(t)}\|_2^2 \\
\le& n f(\overline{\boldsymbol{x}}^{(t-1)}) + \frac{\sigma}{2}\|\boldsymbol{x}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{x}}^{(t)}\|_2^2 \\
=& \sum_{j=1}^n \Big(f(\overline{\boldsymbol{x}}^{(t-1)}) + \Big\langle \nabla f(\overline{\boldsymbol{x}}^{(t-1)}), \boldsymbol{x}_j^{(t-1)} - \overline{\boldsymbol{x}}^{(t-1)} \Big\rangle + \frac{\sigma}{2}\|\boldsymbol{x}_j^{(t-1)} - \overline{\boldsymbol{x}}^{(t)}\|_2^2\Big) \\
\le& \sum_{j=1}^n f(\boldsymbol{x}_j^{(t-1)}).
\end{aligned} \tag{A.58}$$

Next, we bound the function value convergence error after local update, $\sum_{j=1}^n \big(f(\boldsymbol{y}_j^{(t)}) - f(\boldsymbol{y}^\star)\big)$. By constructing the following helper function, we can connect local updates of `Network-SVRG` to that of D-SVRG [CZC+20], which is the counterpart of SVRG in the server/client setting. For agent j at the tth time, we define the corrected sample loss function as

$$\tilde{\ell}^{(j)}(\boldsymbol{x}; \boldsymbol{z}) = \ell(\boldsymbol{x}; \boldsymbol{z}) + \Big\langle \boldsymbol{s}_j^{(t)} - \nabla f(\boldsymbol{y}_j^{(t)}), \boldsymbol{x} - \boldsymbol{y}_j^{(t)} \Big\rangle.$$

Then, define the corrected local and global loss functions as

$$h_i^{(t,j)}(\boldsymbol{x}) = \frac{1}{m}\sum_{\boldsymbol{z}\in\mathcal{M}_i} \tilde{\ell}^{(j)}(\boldsymbol{x}; \boldsymbol{z}) = f_i(\boldsymbol{x}) + \Big\langle \boldsymbol{s}_j^{(t)} - \nabla f(\boldsymbol{y}_j^{(t)}), \boldsymbol{x} - \boldsymbol{y}_j^{(t)} \Big\rangle,$$

$$h^{(t,j)}(\boldsymbol{x}) = \frac{1}{n}\sum_i h_i^{(t,j)}(\boldsymbol{x}) = f(\boldsymbol{x}) + \left\langle \boldsymbol{s}_j^{(t)} - \nabla f(\boldsymbol{y}_j^{(t)}), \boldsymbol{x} - \boldsymbol{y}_j^{(t)} \right\rangle. \tag{A.59}$$

Here, $h^{(t,j)}(\cdot)$ and $h_i^{(t,j)}(\cdot)$ are σ-strongly convex and L-smooth functions, and $\left\|h_i^{(t,j)}(\boldsymbol{x}) - h^{(t,j)}(\boldsymbol{x})\right\|_2 \le \beta$ by the definition of β. Let $h_*^{(t,j)}$ denote the optimum value of $h^{(t,j)}(\cdot)$.

The key observation is that the local update (2.20) at agent j is the same as the update at agent j when applying D-SVRG to optimize $h^{(t,j)}$ initialized with $\boldsymbol{y}_j^{(t)}$. This is true because $\forall \boldsymbol{z} \in \mathcal{M}_j$, the sample gradient and global gradient used in D-SVRG updates at $\boldsymbol{y}_j^{(t)}$ satisfy

$$\nabla\tilde{\ell}^{(j)}(\boldsymbol{u};\boldsymbol{z}) - \nabla\tilde{\ell}^{(j)}(\boldsymbol{u}';\boldsymbol{z}) = \nabla\ell(\boldsymbol{u}';\boldsymbol{z}) - \nabla\ell(\boldsymbol{u};\boldsymbol{z}), \quad \text{and} \quad \nabla h^{(t,j)}(\boldsymbol{y}_j^{(t)}) = \boldsymbol{s}_j^{(t)},$$

which agree with (2.20). Therefore, we can apply [CZC+20, Theorem 1] to bound the optimization error of optimizing $h^{(t,j)}$

$$\mathbb{E}\left[h^{(t,j)}(\boldsymbol{x}_j^{(t)}) - h_*^{(t,j)}\right] < \nu\left(h^{(t,j)}(\boldsymbol{y}_j^{(t)}) - h_*^{(t)}\right), \tag{A.60}$$

where $\boldsymbol{x}_j^{(t)}$ is the output at agent j produced by running one iteration of Alg. 3, which is also the output of running one iteration of D-SVRG at the same agent, ν is the convergence rate of D-SVRG, which can be bounded by $\nu \le 1 - \frac{1}{2}\frac{\sigma-2\beta}{\sigma-3\beta}$ when choosing step size $\delta = \frac{1}{40L}\left(1 - \frac{4\beta}{\sigma}\right)$ and the number of local updates $S = 160\frac{L}{\sigma}\left(1 - \frac{4\beta}{\sigma}\right)^{-2}$.

Next, we relate function value descent of $h^{(t,j)}$ to the function value descent of f. Plug in (A.59) and rearrange terms,

$$\begin{aligned}
f(\boldsymbol{x}_j^{(t)}) - f(\boldsymbol{y}^\star) =& h^{(t,j)}(\boldsymbol{x}_j^{(t)}) - (1-\nu)f(\boldsymbol{y}^\star) - \nu f(\boldsymbol{y}^\star) - \left\langle \boldsymbol{s}_j^{(t)} - \nabla f(\boldsymbol{y}_j^{(t)}), \boldsymbol{x}_j^{(t)} - \boldsymbol{y}_j^{(t)} \right\rangle \\
=& h^{(t,j)}(\boldsymbol{x}_j^{(t)}) - (1-\nu)h^{(t,j)}(\boldsymbol{y}^{\mathsf{opt}}) - \nu f(\boldsymbol{y}^\star) \\
& - \left\langle \boldsymbol{s}_j^{(t)} - \nabla f(\boldsymbol{y}_j^{(t)}), \boldsymbol{x}_j^{(t)} - \boldsymbol{y}_j^{(t)} - (1-\nu)\left(\boldsymbol{y}^{\mathsf{opt}} - \boldsymbol{y}_j^{(t)}\right) \right\rangle \\
\le& h^{(t,j)}(\boldsymbol{x}_j^{(t)}) - (1-\nu)h_*^{(t,j)} - \nu f(\boldsymbol{y}^\star) \\
& - \left\langle \boldsymbol{s}_j^{(t)} - \nabla f(\boldsymbol{y}_j^{(t)}), \boldsymbol{x}_j^{(t)} - \boldsymbol{y}_j^{(t)} - (1-\nu)\left(\boldsymbol{y}^{\mathsf{opt}} - \boldsymbol{y}_j^{(t)}\right) \right\rangle \\
=& h^{(t,j)}(\boldsymbol{x}_j^{(t)}) - h_*^{(t,j)} + \nu\left(h_*^{(t,j)} - f(\boldsymbol{y}^\star)\right) \\
& - \left\langle \boldsymbol{s}_j^{(t)} - \nabla f(\boldsymbol{y}_j^{(t)}), \boldsymbol{x}_j^{(t)} - \boldsymbol{y}_j^{(t)} - (1-\nu)\left(\boldsymbol{y}^{\mathsf{opt}} - \boldsymbol{y}_j^{(t)}\right) \right\rangle,
\end{aligned}$$

where we used $h^{(t,j)}(\boldsymbol{y}^\star) \ge h_*^{(t,j)}$ and $\nu \le 1$ to reach the last inequality.

Taking expectation on both sides and combining with (A.60), we reach the following function value descent of $f(\cdot)$:

$$\mathbb{E}\left[f(\boldsymbol{x}_j^{(t)}) - f(\boldsymbol{y}^\star)\right] \le \nu\left(h^{(t,j)}(\boldsymbol{y}_j^{(t)}) - h_*^{(t,j)}\right) + \nu\left(h_*^{(t,j)} - f(\boldsymbol{y}^\star)\right)$$

$$
\begin{aligned}
&- \mathbb{E}\Big[\Big\langle \boldsymbol{s}_j^{(t)} - \nabla f(\boldsymbol{y}_j^{(t)}), \boldsymbol{x}_j^{(t)} - \boldsymbol{y}_j^{(t)} - (1-\nu)\left(\boldsymbol{y}^{\mathsf{opt}} - \boldsymbol{y}_j^{(t)}\right)\Big\rangle\Big] \\
=&\nu\left(f(\boldsymbol{y}_j^{(t)}) - f(\boldsymbol{y}^\star)\right) - \mathbb{E}\left[\left\langle \boldsymbol{s}_j^{(t)} - \nabla f(\boldsymbol{y}_j^{(t)}), \boldsymbol{x}_j^{(t)} - \boldsymbol{y}^{\mathsf{opt}} - \nu(\boldsymbol{y}_j^{(t)} - \boldsymbol{y}^{\mathsf{opt}})\right\rangle\right],
\end{aligned}
$$

where the last line follows from (A.59). Summing the previous inequality over all agents and using matrix notations, we obtain the following inequality

$$
\begin{aligned}
\mathbb{E}\left[\sum_{j=1}^n f(\boldsymbol{x}_j^{(t)}) - f(\boldsymbol{y}^\star)\right] \leq& \nu\left[\sum_{j=1}^n f(\boldsymbol{y}_j^{(t)}) - f(\boldsymbol{y}^\star)\right] - \mathbb{E}\left[\left\langle \boldsymbol{s}^{(t)} - \nabla f(\boldsymbol{y}^{(t)}), \boldsymbol{x}^{(t)} - \mathbf{1}_n \otimes \boldsymbol{y}^{\mathsf{opt}}\right\rangle\right] \\
&+ \nu\mathbb{E}\left[\left\langle \boldsymbol{s}^{(t)} - \nabla f(\boldsymbol{y}^{(t)}), \boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \boldsymbol{y}^{\mathsf{opt}}\right\rangle\right]. \qquad \text{(A.61)}
\end{aligned}
$$

Our next step is to carefully bound the last two error terms in (A.61).

$$
\begin{aligned}
&\left|\left\langle \boldsymbol{s}^{(t)} - \nabla f(\boldsymbol{y}^{(t)}), \boldsymbol{x}^{(t)} - \mathbf{1}_n \otimes \boldsymbol{y}^{\mathsf{opt}}\right\rangle\right| \\
\leq& \|\boldsymbol{s}^{(t)} - \nabla f(\boldsymbol{y}^{(t)})\|_2 \|\boldsymbol{x}^{(t)} - \mathbf{1}_n \otimes \boldsymbol{y}^{\mathsf{opt}}\|_2 \\
\leq& \Big(\alpha\|\boldsymbol{s}^{t-1} - \nabla f(\boldsymbol{y}^{(t-1)})\|_2 + 2L\|\boldsymbol{y}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)}\|_2 \\
&+ 2\beta\|\boldsymbol{y}^{(t-1)} - \mathbf{1}_n \otimes \boldsymbol{y}^{\mathsf{opt}}\|_2 + \beta\|\boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \boldsymbol{y}^{\mathsf{opt}}\|_2\Big)\|\boldsymbol{x}^{(t)} - \mathbf{1}_n \otimes \boldsymbol{y}^{\mathsf{opt}}\|_2 \\
\leq& \frac{1}{2}\alpha L^{-1}\|\boldsymbol{s}^{t-1} - \nabla f(\boldsymbol{y}^{(t-1)})\|_2^2 + \alpha^{-1}L\|\boldsymbol{y}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{y}}^{(t-1)}\|_2^2 + \frac{3}{2}\alpha L\|\boldsymbol{x}^{(t)} - \mathbf{1}_n \otimes \boldsymbol{y}^{\mathsf{opt}}\|_2^2 \\
&+ \beta\|\boldsymbol{y}^{(t-1)} - \mathbf{1}_n \otimes \boldsymbol{y}^{\mathsf{opt}}\|_2^2 + \frac{\beta}{2}\|\boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \boldsymbol{y}^{\mathsf{opt}}\|_2^2 + \frac{3\beta}{2}\|\boldsymbol{x}^{(t)} - \mathbf{1}_n \otimes \boldsymbol{y}^{\mathsf{opt}}\|_2^2, \qquad \text{(A.62)}
\end{aligned}
$$

where the first inequality is due to (A.72), and the last inequality is obtained by Cauchy-Schwarz inequality. Similar to (A.61), because of the strong convexity of loss functions, we have

$$
\|\boldsymbol{y}^{(t)} - \mathbf{1}_n \otimes \boldsymbol{y}^\star\|_2^2 \leq \frac{2}{\sigma}\sum_j \left(f(\boldsymbol{y}_j^{(t)}) - f(\boldsymbol{y}^\star)\right).
$$

Then, we can further bound (A.62) as

$$
\begin{aligned}
\left|\left\langle \boldsymbol{s}^{(t)} - \nabla f(\boldsymbol{y}^{(t)}), \boldsymbol{x}^{(t)} - \boldsymbol{y}^\star\right\rangle\right| \leq& \frac{1}{2}\alpha L^{-1}\|\boldsymbol{s}^{t-1} - \nabla f(\boldsymbol{y}^{(t-1)})\|_2^2 + \alpha^{-1}L\|\boldsymbol{y}^{(t-1)} - \overline{\boldsymbol{y}}^{(t-1)}\|_2^2 \\
&+ \frac{2\beta}{\sigma}\sum_{j=1}^n \left(f(\boldsymbol{y}_j^{(t-1)}) - f(\boldsymbol{y}^\star)\right) + \frac{\beta}{\sigma}\sum_{j=1}^n \left(f(\boldsymbol{x}_j^{(t-1)}) - f(\boldsymbol{y}^\star)\right) \\
&+ \left(\frac{3\beta}{\sigma} + 3\kappa\alpha\right)\sum_{j=1}^n \left(f(\boldsymbol{x}_j^{(t)}) - f(\boldsymbol{y}^\star)\right). \qquad \text{(A.63)}
\end{aligned}
$$

Similarly, we have the same bound applicable for the last term of (A.61):

$$
\begin{aligned}
\left|\left\langle \boldsymbol{s}^{(t)} - \nabla f(\boldsymbol{y}^{(t)}), \boldsymbol{y}^{(t)} - \boldsymbol{y}^{\mathsf{opt}}\right\rangle\right| \leq& \frac{1}{2}\alpha L^{-1}\|\boldsymbol{s}^{t-1} - \nabla f(\boldsymbol{y}^{(t-1)})\|_2^2 + \alpha^{-1}L\|\boldsymbol{y}^{(t-1)} - \overline{\boldsymbol{y}}^{(t-1)}\|_2^2 \\
&+ \frac{2\beta}{\sigma}\sum_{j=1}^n \left(f(\boldsymbol{y}_j^{(t-1)}) - f(\boldsymbol{y}^\star)\right) + \frac{\beta}{\sigma}\sum_{j=1}^n \left(f(\boldsymbol{x}_j^{(t-1)}) - f(\boldsymbol{y}^\star)\right) \\
&+ \left(\frac{3\beta}{\sigma} + 3\kappa\alpha\right)\sum_{j=1}^n \left(f(\boldsymbol{x}_j^{(t-1)}) - f(\boldsymbol{y}^\star)\right), \qquad \text{(A.64)}
\end{aligned}
$$

where the last term is due to (A.58).

Put together (A.62), (A.63) and (A.64) and taking expectation, we reach the following bound

$$\begin{aligned}\mathbb{E}\left[\sum_{j=1}^{n}\left(f(\boldsymbol{x}_j^{(t)})-f(\boldsymbol{y}^\star)\right)\right] \leq&\Big(\nu\big(1+3\alpha\kappa+\frac{4\beta}{\sigma}\big)+\frac{\beta}{\sigma}\Big)\sum_{j=1}^{n}\left(f(\boldsymbol{x}_j^{(t-1)})-f(\boldsymbol{y}^\star)\right)\\ &+\alpha L^{-1}\|\boldsymbol{s}^{t-1}-\nabla f(\boldsymbol{y}^{(t-1)})\|_2^2+2\alpha^{-1}L\|\boldsymbol{y}^{(t-1)}-\overline{\boldsymbol{y}}^{(t-1)}\|_2^2\\ &+\frac{4\beta}{\sigma}\sum_{j=1}^{n}\left(f(\boldsymbol{y}_j^{(t-1)})-f(\boldsymbol{y}^\star)\right)+\left(\frac{3\beta}{\sigma}+3\kappa\alpha\right)\mathbb{E}\left[\sum_{j=1}^{n}\left(f(\boldsymbol{x}_j^{(t)})-f(\boldsymbol{y}^\star)\right)\right].\end{aligned} \tag{A.65}$$

Rearranging terms, we proved the advertised bound.

A.7.2 Consensus error

We first bound the consensus error $\|\boldsymbol{y}^{(t)}-\mathbf{1}_n\otimes\overline{\boldsymbol{y}}^{(t)}\|_2^2/(\alpha L)$. Similar to (A.40),

$$\begin{aligned}\|\boldsymbol{y}^{(t)}-\mathbf{1}_n\otimes\overline{\boldsymbol{y}}^{(t)}\|_2^2 \leq&\alpha^2\|\boldsymbol{x}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{x}}^{(t-1)}\|_2^2\\ =&\alpha^2\|\boldsymbol{x}^{(t-1)}-\mathbf{1}_n\otimes\boldsymbol{y}^{\mathsf{opt}}\|_2^2-n\alpha^2\|\boldsymbol{y}^\star-\overline{\boldsymbol{x}}^{(t-1)}\|_2\\ \leq&\alpha^2\|\boldsymbol{x}^{(t-1)}-\mathbf{1}_n\otimes\boldsymbol{y}^{\mathsf{opt}}\|_2^2.\end{aligned} \tag{A.66}$$

Then, using the strong convexity of $f(\cdot)$,

$$\begin{aligned}\|\boldsymbol{y}^{(t)}-\mathbf{1}_n\otimes\overline{\boldsymbol{y}}^{(t)}\|_2^2 \leq&\alpha^2\sum_{j=1}^{n}\|\boldsymbol{x}_j^{(t-1)}-\boldsymbol{y}^{\mathsf{opt}}\|_2^2\\ \leq&\frac{2\alpha^2}{\sigma}\sum_{j=1}^{n}\left(f(\boldsymbol{x}_j^{(t-1)})-f(\boldsymbol{y}^\star)\right).\end{aligned} \tag{A.67}$$

A.7.3 Gradient estimation error

To bound the gradient estimation error, we note that

$$\begin{aligned}\|\boldsymbol{s}^{(t)}-\nabla f(\boldsymbol{y}^{(t)})\|_2 =&\|(\boldsymbol{W}\otimes\boldsymbol{I}_d)\boldsymbol{s}^{t-1}+\nabla F(\boldsymbol{y}^{(t)})-\nabla F(\boldsymbol{y}^{(t-1)})-\nabla f(\boldsymbol{y}^{(t)})\|_2\\ =&\Big\|(\boldsymbol{W}\otimes\boldsymbol{I}_d)\Big(\boldsymbol{s}^{t-1}-\nabla f(\boldsymbol{y}^{(t-1)})\Big)+(\boldsymbol{W}\otimes\boldsymbol{I}_d)\nabla f(\boldsymbol{y}^{(t-1)})-\nabla f(\boldsymbol{y}^{(t-1)})\\ &+\nabla F(\boldsymbol{y}^{(t)})-\nabla F(\boldsymbol{y}^{(t-1)})+\nabla f(\boldsymbol{y}^{(t-1)})-\nabla f(\boldsymbol{y}^{(t)})\Big\|_2\\ \leq&\Big\|(\boldsymbol{W}\otimes\boldsymbol{I}_d)\Big(\boldsymbol{s}^{t-1}-\nabla f(\boldsymbol{y}^{(t-1)})\Big)\Big\|_2+\Big\|(\boldsymbol{W}\otimes\boldsymbol{I}_d)\nabla f(\boldsymbol{y}^{(t-1)})-\nabla f(\boldsymbol{y}^{(t-1)})\Big\|_2\\ &+\|\nabla(F-f)(\boldsymbol{y}^{(t)})+\nabla(F-f)(\boldsymbol{y}^{(t-1)})\|_2.\end{aligned} \tag{A.68}$$

We then bound the three terms in (A.68) respectively.

1. The first term can be bounded as

$$\begin{aligned}
&\|(\boldsymbol{W}\otimes \boldsymbol{I}_d)(\boldsymbol{s}^{t-1}-\nabla f(\boldsymbol{y}^{(t-1)}))\|_2\\
=&\left\|(\boldsymbol{W}\otimes \boldsymbol{I}_d)(\boldsymbol{s}^{t-1}-\nabla f(\boldsymbol{y}^{(t-1)}))-\left((\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes \boldsymbol{I}_d\right)(\boldsymbol{s}^{t-1}-\nabla f(\boldsymbol{y}^{(t-1)}))\right\|_2\\
&+\left\|\left((\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes \boldsymbol{I}_d\right)\left(\boldsymbol{s}^{t-1}-\nabla f(\boldsymbol{y}^{(t-1)})\right)\right\|_2\\
\leq&\alpha\|\boldsymbol{s}^{t-1}-\nabla f(\boldsymbol{y}^{(t-1)})\|_2+\left\|\left((\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes \boldsymbol{I}_d\right)(\boldsymbol{s}^{t-1}-\nabla f(\boldsymbol{y}^{(t-1)}))\right\|_2\\
=&\alpha\|\boldsymbol{s}^{t-1}-\nabla f(\boldsymbol{y}^{(t-1)})\|_2+\left\|\left((\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes \boldsymbol{I}_d\right)\left(\nabla(F-f)(\boldsymbol{y}^{t-1})-\nabla(F-f)(\boldsymbol{y}^{\mathsf{opt}})\right)\right\|_2\\
\leq&\alpha\|\boldsymbol{s}^{t-1}-\nabla f(\boldsymbol{y}^{(t-1)})\|_2+\beta\|\boldsymbol{y}^{(t-1)}-\boldsymbol{y}^{\mathsf{opt}}\|_2,
\end{aligned}\tag{A.69}$$

   where we used the fact $\left\|\left((\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes \boldsymbol{I}_d\right)\right\|_2=1$ and the definition of β to reach the last inequality.

2. As for the second term in (A.68), we have

$$\begin{aligned}
&\left\|(\boldsymbol{W}\otimes \boldsymbol{I}_d)\nabla f(\boldsymbol{y}^{(t-1)})-\nabla f(\boldsymbol{y}^{(t-1)})\right\|_2\\
\leq&\left\|(\boldsymbol{W}\otimes \boldsymbol{I}_d)\nabla f(\boldsymbol{y}^{(t-1)})-\left((\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes \boldsymbol{I}_d\right)\nabla f(\boldsymbol{y}^{(t-1)})\right\|_2\\
&+\left\|\left((\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes \boldsymbol{I}_d\right)\nabla f(\boldsymbol{y}^{(t-1)})-\nabla f(\boldsymbol{y}^{(t-1)})\right\|_2\\
\leq&2\left\|\left((\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes \boldsymbol{I}_d\right)\nabla f(\boldsymbol{y}^{(t-1)})-\nabla f(\boldsymbol{y}^{(t-1)})\right\|_2\\
\leq&2\|\nabla f(\overline{\boldsymbol{y}}^{(t-1)})-\nabla f(\boldsymbol{y}^{(t-1)})\|_2\\
\leq&2L\|\boldsymbol{y}^{(t-1)}-\overline{\boldsymbol{y}}^{(t-1)}\|_2,
\end{aligned}\tag{A.70}$$

   where the third inequality follows from the similar trick we used to obtain (A.66).

3. Using the triangle inequality and the definition of β, the last term in (A.68) can be bounded by

$$\|\nabla(F-f)(\boldsymbol{y}^{(t)})+\nabla(F-f)(\boldsymbol{y}^{(t-1)})\|_2\leq\beta\|\boldsymbol{y}^{(t)}-\boldsymbol{y}^{\mathsf{opt}}\|_2+\beta\|\boldsymbol{y}^{(t-1)}-\boldsymbol{y}^{\mathsf{opt}}\|_2.\tag{A.71}$$

Combining (A.68), (A.69), (A.70) and (A.71), the gradient estimation error can be bounded by

$$\begin{aligned}
\|\boldsymbol{s}^{(t)}-\nabla f(\boldsymbol{y}^{(t)})\|_2\leq&\alpha\|\boldsymbol{s}^{t-1}-\nabla f(\boldsymbol{y}^{(t-1)})\|_2+2\beta\|\boldsymbol{y}^{(t-1)}-\boldsymbol{y}^{\mathsf{opt}}\|_2\\
&+\beta\|\boldsymbol{y}^{(t)}-\boldsymbol{y}^{\mathsf{opt}}\|_2+2L\|\boldsymbol{y}^{(t-1)}-\overline{\boldsymbol{y}}^{(t-1)}\|_2.
\end{aligned}\tag{A.72}$$

Because of the strong convexity, $\|\boldsymbol{y}-\boldsymbol{y}^{\mathsf{opt}}\|_2^2\leq\frac{2}{\sigma}\sum_{j=1}^n\left(f(\boldsymbol{y}_j)-f(\boldsymbol{y}^\star)\right)$. Combining with (A.58), we reached the following bound

$$\begin{aligned}
\|\boldsymbol{s}^{(t)}-\nabla f(\boldsymbol{y}^{(t)})\|_2^2\leq&4\alpha^2\|\boldsymbol{s}^{t-1}-\nabla f(\boldsymbol{y}^{(t-1)})\|_2^2+\frac{32\beta^2}{\sigma}\sum_{j=1}^n\left(f(\boldsymbol{y}_j^{(t-1)})-f(\boldsymbol{y}^\star)\right)\\
&+\frac{8\beta^2}{\sigma}\sum_{j=1}^n\left(f(\boldsymbol{x}_j^{(t-1)})-f(\boldsymbol{y}^\star)\right)+16L^2\|\boldsymbol{y}^{(t-1)}-\overline{\boldsymbol{y}}^{(t-1)}\|_2^2.
\end{aligned}\tag{A.73}$$

A.7.4 Linear System

Combining (A.58), (A.67), (A.65), and (A.73), we obtain the claimed linear system.

Appendix B

Appendix for Chapter 3

B.1 Experiment details

For completeness, we list two baseline algorithms, DSGD [NO09, LZZ+17] (cf. Algorithm 10) and GT-SARAH [XKK22a] (cf. Algorithm 11), which are compared numerically against the proposed DESTRESS algorithm in Section 3.3.

Algorithm 10 Decentralized stochastic gradient descent (DSGD)

input: initial parameter $\overline{x}^{(0)}$, initial step size η_0, number of iterations T.

initialization: set $x_i^{(0)} = \overline{x}^{(0)}$.

for $t = 1, \ldots, T$ **do**

  Each agent i samples a mini-batch $\mathcal{Z}_i^{(t)}$ from $\mathcal{M}_i$ uniformly at random, and then performs the following updates:

$$g_i^{(t)} = \frac{1}{b} \sum_{z_i \in \mathcal{Z}_i^{(t)}} \nabla \ell(u_i^{(t)}; z_i).$$

  Update via local communication: $x^{(t+1)} = (W \otimes I_d)(x^{(t)} - \frac{\eta_0}{\sqrt{t}} g^{(t)})$.

output: $x^{\mathsf{out}} = \overline{x}^{(T)}$.

B.2 Proof of Theorem 6

For notation simplicity, let

$$\alpha_{\mathsf{in}} = \alpha^{K_{\mathsf{in}}}, \qquad \alpha_{\mathsf{out}} = \alpha^{K_{\mathsf{out}}}$$

throughout the proof. We define the global gradient $\nabla f(x) \in \mathbb{R}^{nd}$ of an (nd)-dimensional vector $x = \left[x_1^\top, \cdots, x_n^\top\right]^\top$ analogously to Section 2.1.3.

Algorithm 11 GT-SARAH

input: initial parameter $\overline{\boldsymbol{x}}^{(0)}$, step size η, number of outer loops T, number of inner loops q.

initialization: set $\boldsymbol{v}^{(0)} = \boldsymbol{y}^{(0)} = \nabla F(\boldsymbol{x}^{(0)})$.

for $t = 1, \ldots, T$ **do**

  Update via local communication $\boldsymbol{x}^{(t)} = (\boldsymbol{W} \otimes \boldsymbol{I}_d)\boldsymbol{x}^{(t-1)} - \eta \boldsymbol{y}^{(t-1)}$.

  if $\mod (t, q) = 0$ **then**

    $\boldsymbol{v}^{(t)} = \nabla F(\boldsymbol{x}^{(t)})$.

  else

    Each agent i samples a mini-batch $\mathcal{Z}_i^{(t)}$ from $\mathcal{M}_i$ uniformly at random, and then performs the following updates:

$$\boldsymbol{v}_i^{(t)} = \frac{1}{b} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_i^{(t)}} \left(\nabla \ell(\boldsymbol{x}_i^{(t)}; \boldsymbol{z}_i) - \nabla \ell(\boldsymbol{x}_i^{(t-1)}; \boldsymbol{z}_i) \right) + \boldsymbol{v}_i^{(t-1)}.$$

  Update via local communication $\boldsymbol{y}^{(t)} = (\boldsymbol{W} \otimes \boldsymbol{I}_d)\boldsymbol{y}^{(t-1)} + \boldsymbol{v}^{(t)} - \boldsymbol{v}^{(t-1)}$.

output: $\boldsymbol{x}^{\mathsf{out}} = \overline{\boldsymbol{x}}^{(T)}$.

The following fact is a straightforward consequence of our assumption on the mixing matrix $\boldsymbol{W}$ in Definition 1.

Fact 1. *Let* $\boldsymbol{x} = \left[\boldsymbol{x}_1^\top, \cdots, \boldsymbol{x}_n^\top\right]^\top$, *and* $\overline{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i$, *where* $\boldsymbol{x}_i \in \mathbb{R}^d$. *For a mixing matrix* $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ *satisfying Definition 1, we have*

1. $\left(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\right)(\boldsymbol{W} \otimes \boldsymbol{I}_d)\boldsymbol{x} = \left(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\right)\boldsymbol{x} = \overline{\boldsymbol{x}}$;
2. $\left(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\right)(\boldsymbol{W} \otimes \boldsymbol{I}_d) = (\boldsymbol{W} \otimes \boldsymbol{I}_d - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d)\left(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\right)$.

To begin with, we introduce a key lemma that upper bounds the norm of the gradient of the global loss function evaluated at the average local estimates over n agents, in terms of the function value difference at the beginning and the end of the inner loop, the gradient estimation error, and the norm of gradient estimates.

Lemma 4 (Inner loop induction)**.** *Assume Assumption 2 holds. After* $S \geq 1$ *inner loops, one has*

$$\sum_{s=0}^{S-1} \|\nabla f(\overline{\boldsymbol{u}}^{(t),s})\|_2^2 \leq \frac{2}{\eta}\left(f(\overline{\boldsymbol{u}}^{(t),0}) - f(\overline{\boldsymbol{u}}^{(t),S})\right) + \sum_{s=0}^{S-1} \left\|\nabla f(\overline{\boldsymbol{u}}^{(t),s}) - \overline{\boldsymbol{v}}^{(t),s}\right\|_2^2 - (1 - \eta L)\sum_{s=0}^{S-1} \left\|\overline{\boldsymbol{v}}^{(t),s}\right\|_2^2.$$

Proof of Lemma 4. The local update rule (3.1a), combined with Lemma 1, yields

$$\overline{\boldsymbol{u}}^{(t),s+1} = \overline{\boldsymbol{u}}^{(t),s} - \eta \overline{\boldsymbol{v}}^{(t),s}.$$

By Assumption 2, we have

$$\begin{aligned} f(\overline{\boldsymbol{u}}^{(t),s+1}) &= f(\overline{\boldsymbol{u}}^{(t),s} - \eta\overline{\boldsymbol{v}}^{(t),s}) \\ &\le f(\overline{\boldsymbol{u}}^{(t),s}) - \langle \nabla f(\overline{\boldsymbol{u}}^{(t),s}), \eta\overline{\boldsymbol{v}}^{(t),s}\rangle + \frac{L}{2}\|\eta\overline{\boldsymbol{v}}^{(t),s}\|_2^2 \\ &= f(\overline{\boldsymbol{u}}^{(t),s}) - \frac{\eta}{2}\|\nabla f(\overline{\boldsymbol{u}}^{(t),s})\|_2^2 + \frac{\eta}{2}\|\nabla f(\overline{\boldsymbol{u}}^{(t),s}) - \overline{\boldsymbol{v}}^{(t),s}\|_2^2 - \Big(\frac{\eta}{2} - \frac{\eta^2 L}{2}\Big)\|\overline{\boldsymbol{v}}^{(t),s}\|_2^2, \end{aligned} \tag{B.1}$$

where the last equality is obtained by applying $-\langle \boldsymbol{a}, \boldsymbol{b}\rangle = \frac{1}{2}(\|\boldsymbol{a}-\boldsymbol{b}\|_2^2 - \|\boldsymbol{a}\|_2^2 - \|\boldsymbol{b}\|_2^2)$. Summing over $s = 0, \ldots, S-1$ finishes the proof. □

Because the output $\boldsymbol{x}^{\text{out}}$ is chosen from $\{\boldsymbol{u}_i^{(t),s-1} | i \in [n], t \in [T], s \in [S]\}$ uniformly at random, we can compute the expectation of the output's gradient as follows:

$$\begin{aligned} nTS\mathbb{E}\|\nabla f(\boldsymbol{x}^{\text{out}})\|_2^2 &= \sum_{i=1}^{n}\sum_{t=1}^{T}\sum_{s=0}^{S-1}\mathbb{E}\|\nabla f(\boldsymbol{u}_i^{(t),s})\|_2^2 \\ &\overset{\text{(i)}}{=} \sum_{t=1}^{T}\sum_{s=0}^{S-1}\mathbb{E}\|\nabla f(\boldsymbol{u}^{(t),s})\|_2^2 \\ &= \sum_{t=1}^{T}\sum_{s=0}^{S-1}\mathbb{E}\|\nabla f(\boldsymbol{u}^{(t),s}) - \nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),s}) + \nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),s})\|_2^2 \\ &\overset{\text{(ii)}}{\le} 2\sum_{t=1}^{T}\sum_{s=0}^{S-1}\Big(\mathbb{E}\|\nabla f(\boldsymbol{u}^{(t),s}) - \nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),s})\|_2^2 + \mathbb{E}\|\nabla f(\mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),s})\|_2^2\Big) \\ &\overset{\text{(iii)}}{\le} 2\sum_{t=1}^{T}\sum_{s=0}^{S-1}\Big(L^2\mathbb{E}\|\boldsymbol{u}^{(t),s} - \mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),s}\|_2^2 + n\mathbb{E}\|\nabla f(\overline{\boldsymbol{u}}^{(t),s})\|_2^2\Big), \end{aligned} \tag{B.2}$$

where (i) follows from the change of notation using the stacked gradient, (ii) follows from the Cauchy-Schwartz inequality, and (iii) follows from Assumption 2. Then, in view of Lemma 4, (B.2) can be further bounded by

$$\begin{aligned} nTS\mathbb{E}\|\nabla f(\boldsymbol{x}^{\text{out}})\|_2^2 \le \frac{4n}{\eta}\Big(\mathbb{E}[f(\overline{\boldsymbol{x}}^{(0)})] - f^*\Big) + 2L^2\sum_{t=1}^{T}\sum_{s=0}^{S-1}\mathbb{E}\|\boldsymbol{u}^{(t),s} - \mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),s}\|_2^2 \\ + 2n\sum_{t=1}^{T}\sum_{s=0}^{S-1}\Big(\mathbb{E}\|\nabla f(\overline{\boldsymbol{u}}^{(t),s}) - \overline{\boldsymbol{v}}^{(t),s}\|_2^2 - (1-\eta L)\mathbb{E}\|\overline{\boldsymbol{v}}^{(t),s}\|_2^2\Big), \end{aligned} \tag{B.3}$$

where we use $\overline{\boldsymbol{u}}^{(t),0} = \overline{\boldsymbol{x}}^{(t)}$ and $f(\overline{\boldsymbol{u}}^{(t),S}) \ge f^*$.

Next, we present Lemmas 5 and 6 to bound the double sum in (B.3), whose proofs can be found in Appendix B.4 and Appendix B.5, respectively.

Lemma 5 (Sum of inner loop errors). *Assuming all conditions in Theorem 6 hold. For all $t > 0$, we can bound the summation of inner loop errors as*

$$\begin{aligned} &2L^2\sum_{s=0}^{S-1}\mathbb{E}\|\boldsymbol{u}^{(t),s} - \mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),s}\|_2^2 + 2n\sum_{s=0}^{S-1}\mathbb{E}\|\nabla f(\overline{\boldsymbol{u}}^{(t),s}) - \overline{\boldsymbol{v}}^{(t),s}\|_2^2 \\ &\quad\le \frac{64L^2}{1-\alpha_{\text{in}}}\cdot\Big(\frac{S}{npb}+1\Big)\mathbb{E}\|\boldsymbol{x}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{x}}^{(t-1)}\|_2^2 + 2\alpha_{\text{in}}^2\mathbb{E}\|\boldsymbol{s}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{s}}^{(t-1)}\|_2^2 + \frac{2n}{25}\sum_{s=1}^{S}\mathbb{E}\|\overline{\boldsymbol{v}}^{(t),s-1}\|_2^2. \end{aligned}$$

Lemma 6 (Sum of outer loop gradient estimation error and consensus error). *Assuming all conditions in Theorem 6 hold. We have*

$$\frac{64L^2}{1-\alpha_{\mathsf{in}}}\cdot\Big(\frac{S}{npb}+1\Big)\sum_{t=1}^{T}\mathbb{E}\big\|\boldsymbol{x}^{(t)}-\mathbf{1}_n\otimes\bar{\boldsymbol{x}}^{(t)}\big\|_2^2+2\alpha_{\mathsf{in}}^2\sum_{t=1}^{T}\mathbb{E}\big\|\boldsymbol{s}^{(t)}-\mathbf{1}_n\otimes\bar{\boldsymbol{s}}^{(t)}\big\|_2^2\le\frac{11n}{25}\sum_{t=1}^{T}\sum_{s=0}^{S-1}\mathbb{E}\big\|\bar{\boldsymbol{v}}^{(t),s}\big\|_2^2.$$

Using Lemma 5, (B.3) can be bounded as follows:

$$\begin{aligned}nTS\mathbb{E}\big\|\nabla f(\boldsymbol{x}^{\mathsf{out}})\big\|_2^2<{}&\frac{4n}{\eta}\Big(\mathbb{E}[f(\bar{\boldsymbol{x}}^{(t),0})]-f^*\Big)-2n\Big(\frac{24}{25}-\eta L\Big)\sum_{t=1}^{T}\sum_{s=0}^{S-1}\mathbb{E}\big\|\bar{\boldsymbol{v}}^{(t),s}\big\|_2^2\\&+\frac{64L^2}{1-\alpha_{\mathsf{in}}}\cdot\Big(\frac{S}{npb}+1\Big)\sum_{t=0}^{T-1}\mathbb{E}\big\|\boldsymbol{x}^{(t)}-\mathbf{1}_n\otimes\bar{\boldsymbol{x}}^{(t)}\big\|_2^2+2\alpha_{\mathsf{in}}^2\sum_{t=0}^{T-1}\mathbb{E}\big\|\boldsymbol{s}^{(t)}-\mathbf{1}_n\otimes\bar{\boldsymbol{s}}^{(t)}\big\|_2^2,\end{aligned}\tag{B.4}$$

where we bound the sum of inner loop errors $L^2\sum_{s=0}^{S-1}\mathbb{E}\big\|\boldsymbol{u}^{(t),s}-\mathbf{1}_n\otimes\bar{\boldsymbol{u}}^{(t),s}\big\|_2^2$ and $n\sum_{s=0}^{S-1}\mathbb{E}\big\|\nabla f(\bar{\boldsymbol{u}}^{(t),s})-\bar{\boldsymbol{v}}^{(t),s}\big\|_2^2$ by the initial value of each inner loop $\mathbb{E}\big\|\boldsymbol{x}^{(t)}-\mathbf{1}_n\otimes\bar{\boldsymbol{x}}^{(t)}\big\|_2^2$ and $\mathbb{E}\big\|\boldsymbol{s}^{(t)}-\mathbf{1}_n\otimes\bar{\boldsymbol{s}}^{(t)}\big\|_2^2$, and the summation of the norm of average inner loop gradient estimator $n\sum_{s=1}^{S}\mathbb{E}\big\|\bar{\boldsymbol{v}}^{(t),s-1}\big\|_2^2$.

By Lemma 6, (B.4) can be further bounded as

$$\begin{aligned}nTS\mathbb{E}\big\|\nabla f(\boldsymbol{x}^{\mathsf{out}})\big\|_2^2&\le\frac{4n}{\eta}\Big(\mathbb{E}[f(\bar{\boldsymbol{x}}^{(t),0})]-f^*\Big)-2n\Big(\frac{37}{50}-\eta L\Big)\sum_{t=1}^{T}\sum_{s=0}^{S-1}\mathbb{E}\big\|\bar{\boldsymbol{v}}^{(t),s}\big\|_2^2\\&<\frac{4n}{\eta}\Big(\mathbb{E}[f(\bar{\boldsymbol{x}}^{(t),0})]-f^*\Big),\end{aligned}$$

which concludes the proof.

B.3 Proof of Corollary 3

Without loss of generality, we assume $n\ge 2$. Otherwise, the problem reduces to the centralized setting with a single agent $n=1$, and the bound holds trivially. We will confirm the choice of parameters in Corollary 3 in the following paragraphs, and finally obtain the IFO complexity and communication complexity.

Step size η We first assume $\alpha_{\mathsf{in}}\le\frac{p}{2}\le\frac{1}{2}$ and $\alpha_{\mathsf{out}}\le\frac{1}{\sqrt{npb}+1}\le\frac{1}{2}$, which will be proved to hold shortly, then we can verify the step size choice meets the requirement in (3.3) as:

$$\frac{(1-\alpha_{\mathsf{in}})^3(1-\alpha_{\mathsf{out}})}{1+\alpha^{K_{\mathsf{in}}}\alpha^{K_{\mathsf{out}}}\sqrt{pnb}}\cdot\frac{1}{10L\big(\sqrt{S/(npb)}+1\big)}\ge\frac{(1/2)^4}{2}\cdot\frac{1}{20L}=\frac{1}{640L}.$$

Mixing steps K_{in} and K_{out} Using Chebyshev's acceleration [AS14] to implement the mixing steps, it amounts to an improved mixing rate of $\alpha_{\mathsf{cheb}}\asymp 1-\sqrt{2(1-\alpha)}$, when the original mixing rate α is close to 1. Set $K_{\mathsf{in}}=\Big\lceil\frac{\log(2/p)}{\sqrt{1-\alpha}}\Big\rceil$ and $K_{\mathsf{out}}=\Big\lceil\frac{\log(\sqrt{npb}+1)}{\sqrt{1-\alpha}}\Big\rceil$. We are now positioned to examine the effective mixing rate $\alpha_{\mathsf{in}}=\alpha_{\mathsf{cheb}}^{K_{\mathsf{in}}}$ and $\alpha_{\mathsf{out}}=\alpha_{\mathsf{cheb}}^{K_{\mathsf{out}}}$, as follows

$$\alpha_{\mathsf{out}}=\alpha_{\mathsf{cheb}}^{K_{\mathsf{out}}}\overset{\text{(i)}}{\le}\alpha_{\mathsf{cheb}}^{\frac{\log(\sqrt{npb}+1)}{\sqrt{1-\alpha}}}\asymp\alpha_{\mathsf{cheb}}^{\frac{\sqrt{2}\log(\sqrt{npb}+1)}{1-\alpha_{\mathsf{cheb}}}}\overset{\text{(ii)}}{\le}\alpha_{\mathsf{cheb}}^{\frac{\sqrt{2}\log(\sqrt{npb}+1)}{-\log\alpha_{\mathsf{cheb}}}}<\frac{1}{\sqrt{npb}+1}\overset{\text{(iii)}}{\le}\frac{1}{2},$$

where (i) follows from $K_{\mathsf{out}} = \left\lceil \frac{\log(\sqrt{npb}+1)}{\sqrt{1-\alpha}} \right\rceil$, (ii) follows from $\log x \leq x-1$, $\forall x > 0$, and (iii) follows from $n \geq 1$ and $b \geq 1$. By a similar argument, we have $\alpha_{\mathsf{in}} = \alpha_{\mathsf{cheb}}^{K_{\mathsf{in}}} \leq \frac{p}{2}$.

Complexity Plugging in the selected parameters into (3.4) in Theorem 6, We have

$$\mathbb{E}\left\|\nabla f(\boldsymbol{x}^{\mathsf{out}})\right\|_2^2 \leq \frac{4}{\eta TS}\Big(\mathbb{E}[f(\overline{\boldsymbol{x}}^{(t),0})] - f^*\Big) = O\Big(\frac{L}{T\sqrt{mn}}\Big).$$

Consequently, the outer iteration complexity is $T = O\Big(1 + \frac{L}{(mn)^{1/2}\epsilon^2}\Big)$. With this in place, we summarize the communication and IFO complexities as follows:

- The communication complexity is

$$\begin{aligned} T \cdot (SK_{\mathsf{in}} + K_{\mathsf{out}}) &= O\Big(\frac{(mn)^{1/2}\log\left(2(n/m)^{1/2}+2\right) + \log\left((mn)^{1/4}+1\right)}{\sqrt{1-\alpha}} \cdot \Big(1 + \frac{L}{(mn)^{1/2}\epsilon^2}\Big)\Big) \\ &= O\Big(\frac{\log\left((n/m)^{1/2}+2\right)}{\sqrt{1-\alpha}} \cdot \big((mn)^{1/2} + \frac{L}{\epsilon^2}\big)\Big), \end{aligned}$$

  where we use $2/p = \frac{2\lceil\sqrt{m/n}\rceil}{\sqrt{m/n}} \leq \frac{2(\sqrt{m/n}+1)}{\sqrt{m/n}} = 2(\sqrt{n/m}+1)$ to bound K_{in}.

- The IFO complexity is $T \cdot (Spb + 2m) = O\Big(m + \frac{(m/n)^{1/2}L}{\epsilon^2}\Big)$.

B.4 Proof of Lemma 5

This section proves Lemma 5. Appendices B.4.1 and B.4.2 bounds the expected inner loop gradient estimation error and consensus errors by their previous values and the sum of inner loop gradient estimator's norms, Appendix B.4.3 then creates a linear system to compute the summation of inner loop errors using their initial values of each inner loop, which concludes the proof.

B.4.1 Sum of inner loop gradient estimation errors

To begin with, note that the gradient estimation error at the s-th inner loop iteration can be written as

$$\begin{aligned} &\mathbb{E}\left\|\nabla f(\overline{\boldsymbol{u}}^{(t),s}) - \overline{\boldsymbol{v}}^{(t),s}\right\|_2^2 \\ &= \mathbb{E}\left\|\Big(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\Big)\Big(\nabla F(\mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),s}) - \boldsymbol{v}^{(t),s}\Big)\right\|_2^2 \\ &= \mathbb{E}\left\|\Big(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\Big)\big(\nabla F(\mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),s}) - \nabla F(\boldsymbol{u}^{(t),s})\big) + \Big(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\Big)\big(\nabla F(\boldsymbol{u}^{(t),s}) - \boldsymbol{v}^{(t),s}\big)\right\|_2^2 \\ &\leq 2\mathbb{E}\left\|\Big(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\Big)\big(\nabla F(\mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),s}) - \nabla F(\boldsymbol{u}^{(t),s})\big)\right\|_2^2 + 2\mathbb{E}\left\|\Big(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\Big)\big(\nabla F(\boldsymbol{u}^{(t),s}) - \boldsymbol{v}^{(t),s}\big)\right\|_2^2 \\ &\leq \frac{2L^2}{n}\mathbb{E}\left\|\boldsymbol{u}^{(t),s} - \mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),s}\right\|_2^2 + 2\mathbb{E}\left\|\Big(\frac{1}{n}\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\Big)\big(\nabla F(\boldsymbol{u}^{(t),s}) - \boldsymbol{v}^{(t),s}\big)\right\|_2^2, \end{aligned} \tag{B.5}$$

where the first equality follows from (2.4), and the last inequality is due to Assumption 2. To continue, the expectation of the second term in (B.5) can be bounded as

$$
\begin{aligned}
&\mathbb{E}\Big\|\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes \boldsymbol{I}_d\Big)\Big(\nabla F(\boldsymbol{u}^{(t),s})-\boldsymbol{v}^{(t),s}\Big)\Big\|_2^2\\
&=\mathbb{E}\Big\|\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes \boldsymbol{I}_d\Big)\Big(\big(\nabla F(\boldsymbol{u}^{(t),s})-\boldsymbol{v}^{(t),s}\big)-\big(\nabla F(\boldsymbol{u}^{(t),s-1})-\boldsymbol{v}^{(t),s-1}\big)+\big(\nabla F(\boldsymbol{u}^{(t),s-1})-\boldsymbol{v}^{(t),s-1}\big)\Big)\Big\|_2^2\\
&\overset{\text{(i)}}{=}\mathbb{E}\Big\|\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes \boldsymbol{I}_d\Big)\Big(\big(\nabla F(\boldsymbol{u}^{(t),s})-\boldsymbol{v}^{(t),s}\big)-\big(\nabla F(\boldsymbol{u}^{(t),s-1})-\boldsymbol{v}^{(t),s-1}\big)\Big)\Big\|_2^2\\
&\quad+\mathbb{E}\Big\|\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes \boldsymbol{I}_d\Big)\big(\nabla F(\boldsymbol{u}^{(t),s-1})-\boldsymbol{v}^{(t),s-1}\big)\Big\|_2^2\\
&\overset{\text{(ii)}}{=}\sum_{k=1}^{s}\mathbb{E}\Big\|\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes \boldsymbol{I}_d\Big)\Big(\big(\nabla F(\boldsymbol{u}^{(t),k})-\boldsymbol{v}^{(t),k}\big)-\big(\nabla F(\boldsymbol{u}^{(t),k-1})-\boldsymbol{v}^{(t),k-1}\big)\Big)\Big\|_2^2\\
&\quad+\mathbb{E}\Big\|\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes \boldsymbol{I}_d\Big)\big(\nabla F(\boldsymbol{u}^{(t),0})-\boldsymbol{v}^{(t),0}\big)\Big\|_2^2\\
&\overset{\text{(iii)}}{=}\sum_{k=1}^{s}\mathbb{E}\Big\|\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes \boldsymbol{I}_d\Big)\Big(\big(\nabla F(\boldsymbol{u}^{(t),k})-\boldsymbol{v}^{(t),k}\big)-\big(\nabla F(\boldsymbol{u}^{(t),k-1})-\boldsymbol{v}^{(t),k-1}\big)\Big)\Big\|_2^2.
\end{aligned}
\tag{B.6}
$$

Here, (i) follows from the expectation with respect to the activating indicator $\lambda_i^{(t),s}$ and random samples $\mathcal{Z}^{(t),s}$, conditioned on $\boldsymbol{u}^{(t),s-1}$ and $\boldsymbol{v}^{(t),s-1}$:

$$
\begin{aligned}
&\mathbb{E}\left[\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes \boldsymbol{I}_d\Big)\big(\nabla F(\boldsymbol{u}^{(t),s})-\boldsymbol{v}^{(t),s}\big)\Big|\boldsymbol{u}^{(t),s-1},\boldsymbol{v}^{(t),s-1}\right]\\
&=\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{u}_i^{(t),s})-\overline{\boldsymbol{v}}^{(t),s-1}\\
&\qquad-\mathbb{E}\left[\frac{1}{npb}\sum_i\lambda_i^{(t),s}\sum_{z_i\in\mathcal{Z}_i^{(t),s}}\Big(\nabla\ell(\boldsymbol{u}_i^{(t),s};z_i)-\nabla\ell(\boldsymbol{u}_i^{(t),s-1};z_i)\Big)\Big|\boldsymbol{u}^{(t),s-1},\boldsymbol{v}^{(t),s-1}\right]\\
&=\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{u}_i^{(t),s})-\frac{1}{np}\sum_i\mathbb{E}\big[\lambda_i^{(t),s}\big]\Big(\nabla f_i(\boldsymbol{u}_i^{(t),s})-\nabla f_i(\boldsymbol{u}_i^{(t),s-1})\Big)-\overline{\boldsymbol{v}}^{(t),s-1}\\
&=\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{u}_i^{(t),s-1})-\overline{\boldsymbol{v}}^{(t),s-1}\\
&=\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes \boldsymbol{I}_d\Big)\big(\nabla F(\boldsymbol{u}^{(t),s-1})-\boldsymbol{v}^{(t),s-1}\big),
\end{aligned}
\tag{B.7}
$$

(ii) follows by recursively applying the relation obtained from (i); and (iii) follows from the property of gradient tracking, i.e.

$$
\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes \boldsymbol{I}_d\Big)\nabla F(\boldsymbol{u}^{(t),0})=\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{u}_i^{(t),0})=\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x}_i^{(t-1)})=\overline{\boldsymbol{s}}^{(t-1)}=\overline{\boldsymbol{v}}^{(t),0},
\tag{B.8}
$$

which leads to $\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes \boldsymbol{I}_d\Big)\big(\nabla F(\boldsymbol{u}^{(t),0})-\boldsymbol{v}^{(t),0}\big)=\mathbf{0}$.

We now continue to bound each term in (B.6), which can be viewed as the variance of the stochastic gradient, as

$$
\mathbb{E}\Big\|\Big(\frac{1}{n}\mathbf{1}_n^\top\otimes \boldsymbol{I}_d\Big)\Big(\big(\nabla F(\boldsymbol{u}^{(t),s})-\boldsymbol{v}^{(t),s}\big)-\big(\nabla F(\boldsymbol{u}^{(t),s-1})-\boldsymbol{v}^{(t),s-1}\big)\Big)\Big\|_2^2
$$

$$
\begin{aligned}
&\overset{\text{(i)}}{=} \mathbb{E}\left\| \frac{1}{nb}\sum_{i=1}^{n}\sum_{z_i\in\mathcal{Z}_i^{(t),s}} \Big((\nabla f_i(\boldsymbol{u}_i^{(t),s}) - \nabla f_i(\boldsymbol{u}_i^{(t),s-1})) - \frac{\lambda_i^{(t),s}}{p}(\nabla \ell(\boldsymbol{u}_i^{(t),s};z_i) - \nabla \ell(\boldsymbol{u}_i^{(t),s-1};z_i))\Big)\right\|_2^2 \\
&\overset{\text{(ii)}}{=} \frac{1}{n^2b^2}\sum_{i=1}^{n}\sum_{z_i\in\mathcal{Z}_i^{(t),s}} \mathbb{E}\left\| (\nabla f_i(\boldsymbol{u}_i^{(t),s}) - \nabla f_i(\boldsymbol{u}_i^{(t),s-1})) - \frac{\lambda_i^{(t),s}}{p}(\nabla \ell(\boldsymbol{u}_i^{(t),s};z_i) - \nabla \ell(\boldsymbol{u}_i^{(t),s-1};z_i))\right\|_2^2 \\
&\overset{\text{(iii)}}{=} \frac{1}{n^2p^2b^2}\sum_{i=1}^{n}\sum_{z_i\in\mathcal{Z}_i^{(t),s}} \mathbb{E}\Big[(\lambda_i^{(t),s})^2\Big]\mathbb{E}\|\nabla \ell(\boldsymbol{u}_i^{(t),s};z_i) - \nabla \ell(\boldsymbol{u}_i^{(t),s-1};z_i)\|_2^2 \\
&\quad - \frac{1}{n^2b}\mathbb{E}\|\nabla F(\boldsymbol{u}^{(t),s}) - \nabla F(\boldsymbol{u}^{(t),s-1})\|_2^2 \\
&\le \frac{L^2}{n^2pb}\mathbb{E}\|\boldsymbol{u}^{(t),s} - \boldsymbol{u}^{(t),s-1}\|_2^2,
\end{aligned} \tag{B.9}
$$

where (i) follows from the update rules (3.1b) and (3.1c), (ii) follows from the independence of samples and $\mathbb{E}\big[\lambda_i^{(t),s}\big] = p$, (iii) follows from similar argument with (B.7), and the last inequality follows from Assumption 2 and $\mathbb{E}\Big[(\lambda_i^{(t),s})^2\Big] = p$.

In view of (3.1a), the difference between inner loop variables in (B.9) can be bounded deterministically as

$$
\begin{aligned}
&\|\boldsymbol{u}^{(t),s} - \boldsymbol{u}^{(t),s-1}\|_2^2 \\
&= \|(\boldsymbol{W}_{\mathsf{in}}\otimes \boldsymbol{I}_d)(\boldsymbol{u}^{(t),s-1} - \eta \boldsymbol{v}^{(t),s-1}) - \boldsymbol{u}^{(t),s-1}\|_2^2 \\
&\overset{\text{(i)}}{=} \Big\| \Big((\boldsymbol{W}_{\mathsf{in}}\otimes \boldsymbol{I}_d) - \boldsymbol{I}_{nd}\Big)(\boldsymbol{u}^{(t),s-1} - \mathbf{1}_n\otimes \overline{\boldsymbol{u}}^{(t),s-1}) \\
&\qquad - \eta\Big((\boldsymbol{W}_{\mathsf{in}}\otimes \boldsymbol{I}_d) - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes \boldsymbol{I}_d\Big)(\boldsymbol{v}^{(t),s-1} - \mathbf{1}_n\otimes \overline{\boldsymbol{v}}^{(t),s-1}) - \eta\mathbf{1}_n\otimes \overline{\boldsymbol{v}}^{(t),s-1}\Big\|_2^2 \\
&\overset{\text{(ii)}}{=} \Big\| \Big((\boldsymbol{W}_{\mathsf{in}}\otimes \boldsymbol{I}_d) - \boldsymbol{I}_{nd}\Big)(\boldsymbol{u}^{(t),s-1} - \mathbf{1}_n\otimes \overline{\boldsymbol{u}}^{(t),s-1}) \\
&\qquad - \eta\Big((\boldsymbol{W}_{\mathsf{in}}\otimes \boldsymbol{I}_d) - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes \boldsymbol{I}_d\Big)(\boldsymbol{v}^{(t),s-1} - \mathbf{1}_n\otimes \overline{\boldsymbol{v}}^{(t),s-1})\Big\|_2^2 + \eta^2 n\|\overline{\boldsymbol{v}}^{(t),s-1}\|_2^2 \\
&\le 8\|\boldsymbol{u}^{(t),s-1} - \mathbf{1}_n\otimes \overline{\boldsymbol{u}}^{(t),s-1}\|_2^2 + 2\alpha_{\mathsf{in}}^2\eta^2\|\boldsymbol{v}^{(t),s-1} - \mathbf{1}_n\otimes \overline{\boldsymbol{v}}^{(t),s-1}\|_2^2 + \eta^2 n\|\overline{\boldsymbol{v}}^{(t),s-1}\|_2^2,
\end{aligned} \tag{B.10}
$$

where (i) and (ii) follow from $\big((\boldsymbol{W}_{\mathsf{in}}\otimes \boldsymbol{I}_d) - \boldsymbol{I}_{nd}\big)(\mathbf{1}_n\otimes \overline{\boldsymbol{x}}) = \mathbf{0}$ and $\big((\boldsymbol{W}_{\mathsf{in}}\otimes \boldsymbol{I}_d) - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes \boldsymbol{I}_d\big)(\mathbf{1}_n\otimes \overline{\boldsymbol{x}}) = \mathbf{0}$ for any mean vector $\overline{\boldsymbol{x}}$; and the last inequality follows from the property of the mixing matrix $\big\|(\boldsymbol{W}_{\mathsf{in}}\otimes \boldsymbol{I}_d) - \boldsymbol{I}_{nd}\big\|_{\mathsf{op}} \le 2$ and $\big\|(\boldsymbol{W}_{\mathsf{in}}\otimes \boldsymbol{I}_d) - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes \boldsymbol{I}_d\big\|_{\mathsf{op}} \le \alpha_{\mathsf{in}}$.

Plugging (B.9) and (B.10) into (B.6), we can further obtain

$$
\begin{aligned}
&\mathbb{E}\Big\|\Big((\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes \boldsymbol{I}_d\Big)\Big(\nabla F(\boldsymbol{u}^{(t),s}) - \boldsymbol{v}^{(t),s}\Big)\Big\|_2^2 \\
&\le \frac{L^2}{n^2pb}\sum_{k=1}^{s}\mathbb{E}\|\boldsymbol{u}^{(t),k} - \boldsymbol{u}^{(t),k-1}\|_2^2 \\
&\le \frac{8L^2}{n^2pb}\sum_{k=0}^{s-1}\mathbb{E}\|\boldsymbol{u}^{(t),k} - \mathbf{1}_n\otimes \overline{\boldsymbol{u}}^{(t),k}\|_2^2 + \frac{2\alpha_{\mathsf{in}}^2\eta^2L^2}{n^2pb}\sum_{k=0}^{s-1}\mathbb{E}\|\boldsymbol{v}^{(t),k} - \mathbf{1}_n\otimes \overline{\boldsymbol{v}}^{(t),k}\|_2^2 + \frac{\eta^2L^2}{npb}\sum_{k=0}^{s-1}\mathbb{E}\|\overline{\boldsymbol{v}}^{(t),k}\|_2^2.
\end{aligned}
$$

Using (B.5) and the previous inequality, we can bound the summation of inner loop gradient estimation errors as

$$\begin{aligned}
&\sum_{s=0}^{S-1}\mathbb{E}\left\|\nabla f(\overline{\boldsymbol{u}}^{(t),s})-\overline{\boldsymbol{v}}^{(t),s}\right\|_2^2\\
&\le\frac{2L^2}{n}\sum_{s=0}^{S-1}\mathbb{E}\left\|\boldsymbol{u}^{(t),s}-\mathbf{1}_n\otimes\overline{\boldsymbol{u}}^{(t),s}\right\|_2^2+2\sum_{s=0}^{S-1}\mathbb{E}\left\|\left(\frac{1}{n}\mathbf{1}_n^\top\otimes\boldsymbol{I}_d\right)\left(\nabla F(\boldsymbol{u}^{(t),s})-\boldsymbol{v}^{(t),s}\right)\right\|_2^2\\
&\le\frac{2L^2}{n}\sum_{s=0}^{S-1}\mathbb{E}\left\|\boldsymbol{u}^{(t),s}-\mathbf{1}_n\otimes\overline{\boldsymbol{u}}^{(t),s}\right\|_2^2+\frac{16L^2}{n^2pb}\sum_{s=0}^{S-1}\sum_{k=0}^{s-1}\mathbb{E}\left\|\boldsymbol{u}^{(t),k}-\mathbf{1}_n\otimes\overline{\boldsymbol{u}}^{(t),k}\right\|_2^2\\
&\quad+\frac{4\alpha_{\mathsf{in}}^2\eta^2L^2}{n^2pb}\sum_{s=0}^{S-1}\sum_{k=0}^{s-1}\mathbb{E}\left\|\boldsymbol{v}^{(t),k}-\mathbf{1}_n\otimes\overline{\boldsymbol{v}}^{(t),k}\right\|_2^2+\frac{2\eta^2L^2}{npb}\sum_{s=0}^{S-1}\sum_{k=0}^{s-1}\mathbb{E}\left\|\overline{\boldsymbol{v}}^{(t),k}\right\|_2^2\\
&\le\left(\frac{8S}{npb}+1\right)\cdot\frac{2L^2}{n}\sum_{s=0}^{S-1}\mathbb{E}\left\|\boldsymbol{u}^{(t),s}-\mathbf{1}_n\otimes\overline{\boldsymbol{u}}^{(t),s}\right\|_2^2\\
&\quad+\frac{4S\alpha_{\mathsf{in}}^2\eta^2L^2}{n^2pb}\sum_{s=0}^{S-1}\mathbb{E}\left\|\boldsymbol{v}^{(t),s}-\mathbf{1}_n\otimes\overline{\boldsymbol{v}}^{(t),s}\right\|_2^2+\frac{2S\eta^2L^2}{npb}\sum_{s=0}^{S-1}\mathbb{E}\left\|\overline{\boldsymbol{v}}^{(t),s}\right\|_2^2,
\end{aligned}$$

where the last inequality is obtained by relaxing the upper bound of the summation w.r.t. k from $s-1$ to $S-1$.

The quantity of interest can be now bounded as

$$\begin{aligned}
2L^2\sum_{s=0}^{S-1}\mathbb{E}\left\|\boldsymbol{u}^{(t),s}-\mathbf{1}_n\otimes\overline{\boldsymbol{u}}^{(t),s}\right\|_2^2+2n\sum_{s=0}^{S-1}\mathbb{E}\left\|\nabla f(\overline{\boldsymbol{u}}^{(t),s})-\overline{\boldsymbol{v}}^{(t),s}\right\|_2^2&\\
\le\left(\frac{4S}{npb}+1\right)\cdot 8L^2\sum_{s=0}^{S-1}\mathbb{E}\left\|\boldsymbol{u}^{(t),s}-\mathbf{1}_n\otimes\overline{\boldsymbol{u}}^{(t),s}\right\|_2^2&\\
+\frac{8S\alpha_{\mathsf{in}}^2\eta^2L^2}{npb}\sum_{s=0}^{S-1}\mathbb{E}\left\|\boldsymbol{v}^{(t),s}-\mathbf{1}_n\otimes\overline{\boldsymbol{v}}^{(t),s}\right\|_2^2+\frac{4S\eta^2L^2}{pb}\sum_{s=0}^{S-1}\mathbb{E}\left\|\overline{\boldsymbol{v}}^{(t),s}\right\|_2^2&\qquad\text{(B.11)}
\end{aligned}$$

B.4.2 Sum of inner loop consensus errors

Using the update rule (3.1a), the variable consensus error can be expanded deterministically as follows:

$$\begin{aligned}
\left\|\boldsymbol{u}^{(t),s}-\mathbf{1}_n\otimes\overline{\boldsymbol{u}}^{(t),s}\right\|_2^2&=\left\|\left(\boldsymbol{I}_{nd}-(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\boldsymbol{I}_d\right)(\boldsymbol{W}_{\mathsf{in}}\otimes\boldsymbol{I}_d)(\boldsymbol{u}^{(t),s-1}-\eta\boldsymbol{v}^{(t),s-1})\right\|_2^2\\
&\overset{\text{(i)}}{\le}\alpha_{\mathsf{in}}^2\left\|\left(\boldsymbol{I}_{nd}-(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\boldsymbol{I}_d\right)(\boldsymbol{u}^{(t),s-1}-\eta\boldsymbol{v}^{(t),s-1})\right\|_2^2\\
&\le\frac{2\alpha_{\mathsf{in}}^2}{1+\alpha_{\mathsf{in}}^2}\left\|\boldsymbol{u}^{(t),s-1}-\mathbf{1}_n\otimes\overline{\boldsymbol{u}}^{(t),s-1}\right\|_2^2+\frac{2\alpha_{\mathsf{in}}^2\eta^2}{1-\alpha_{\mathsf{in}}^2}\left\|\boldsymbol{v}^{(t),s-1}-\mathbf{1}_n\otimes\overline{\boldsymbol{v}}^{(t),s-1}\right\|_2^2,
\end{aligned}\qquad\text{(B.12)}$$

where (i) follows from the fact

$$\left(\boldsymbol{I}_{nd}-(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\boldsymbol{I}_d\right)(\boldsymbol{W}\otimes\boldsymbol{I}_d)=\left(\boldsymbol{W}\otimes\boldsymbol{I}_d-(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\boldsymbol{I}_d\right)\left(\boldsymbol{I}_{nd}-(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\boldsymbol{I}_d\right)$$

and the definition of the mixing rate. The last inequality follows from the elementary inequality $2\langle\boldsymbol{a},\boldsymbol{b}\rangle\le\frac{1-\alpha_{\mathsf{in}}^2}{1+\alpha_{\mathsf{in}}^2}\|\boldsymbol{a}\|_2^2+\frac{1+\alpha_{\mathsf{in}}^2}{1-\alpha_{\mathsf{in}}^2}\|\boldsymbol{b}\|_2^2$, so that $\|\boldsymbol{a}+\boldsymbol{b}\|_2^2\le\frac{2}{1+\alpha_{\mathsf{in}}^2}\|\boldsymbol{a}\|_2^2+\frac{2}{1-\alpha_{\mathsf{in}}^2}\|\boldsymbol{b}\|_2^2$.

Furthermore, using the update rules (3.1b) and (3.1c) and defining $\boldsymbol{\Lambda}^{(t),s} = \frac{1}{p}\mathrm{diag}(\lambda_1^{(t),s}, \lambda_2^{(t),s}, \ldots, \lambda_n^{(t),s}) \otimes \boldsymbol{I}_d$, the gradient consensus error can be similarly expanded as follows:

$$
\begin{aligned}
&\big\|\boldsymbol{v}^{(t),s} - \mathbf{1}_n \otimes \overline{\boldsymbol{v}}^{(t),s}\big\|_2^2 \\
&= \Big\|\Big(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\Big)(\boldsymbol{W}_{\mathsf{in}} \otimes \boldsymbol{I}_d)\boldsymbol{g}^{(t),s}\Big\|_2^2 \\
&\le \alpha_{\mathsf{in}}^2\Big\|\Big(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\Big)\boldsymbol{g}^{(t),s}\Big\|_2^2 \\
&\overset{\text{(i)}}{\le} \frac{2\alpha_{\mathsf{in}}^2}{1+\alpha_{\mathsf{in}}^2}\big\|\boldsymbol{v}^{(t),s-1} - \mathbf{1}_n \otimes \overline{\boldsymbol{v}}^{(t),s-1}\big\|_2^2 \\
&\quad + \frac{2\alpha_{\mathsf{in}}^2}{1-\alpha_{\mathsf{in}}^2}\cdot\frac{1}{b}\sum_{z\in\mathcal{Z}^{(t),s}}\Big\|\Big(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\Big)\boldsymbol{\Lambda}^{(t),s}\big(\nabla\ell(\boldsymbol{u}^{(t),s};z) - \nabla\ell(\boldsymbol{u}^{(t),s-1};z)\big)\Big\|_2^2 \\
&\overset{\text{(ii)}}{\le} \frac{2\alpha_{\mathsf{in}}^2}{1+\alpha_{\mathsf{in}}^2}\big\|\boldsymbol{v}^{(t),s-1} - \mathbf{1}_n \otimes \overline{\boldsymbol{v}}^{(t),s-1}\big\|_2^2 + \frac{2\alpha_{\mathsf{in}}^2 L^2}{(1-\alpha_{\mathsf{in}}^2)p^2}\big\|\boldsymbol{u}^{(t),s} - \boldsymbol{u}^{(t),s-1}\big\|_2^2 \\
&\overset{\text{(iii)}}{\le} \frac{2\alpha_{\mathsf{in}}^2}{1+\alpha_{\mathsf{in}}^2}\big\|\boldsymbol{v}^{(t),s-1} - \mathbf{1}_n \otimes \overline{\boldsymbol{v}}^{(t),s-1}\big\|_2^2 + \frac{2\alpha_{\mathsf{in}}^2 L^2}{(1-\alpha_{\mathsf{in}}^2)p^2}\Big(8\big\|\boldsymbol{u}^{(t),s-1} - \mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),s-1}\big\|_2^2 \\
&\quad + 2\alpha_{\mathsf{in}}^2\eta^2\big\|\boldsymbol{v}^{(t),s-1} - \mathbf{1}_n \otimes \overline{\boldsymbol{v}}^{(t),s-1}\big\|_2^2 + \eta^2 n\big\|\overline{\boldsymbol{v}}^{(t),s-1}\big\|_2^2\Big) \\
&= \Big(\frac{2\alpha_{\mathsf{in}}^2}{1+\alpha_{\mathsf{in}}^2} + \frac{4\alpha_{\mathsf{in}}^4\eta^2L^2}{(1-\alpha_{\mathsf{in}}^2)p^2}\Big)\big\|\boldsymbol{v}^{(t),s-1} - \mathbf{1}_n \otimes \overline{\boldsymbol{v}}^{(t),s-1}\big\|_2^2 \\
&\quad + \frac{16\alpha_{\mathsf{in}}^2 L^2}{(1-\alpha_{\mathsf{in}}^2)p^2}\big\|\boldsymbol{u}^{(t),s-1} - \mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),s-1}\big\|_2^2 + \frac{2\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}}^2)p^2}\cdot n\big\|\overline{\boldsymbol{v}}^{(t),s-1}\big\|_2^2,
\end{aligned}
\tag{B.13}
$$

where the second term in (i) is obtained by Jensen's inequality, (ii) follows from Assumption 2 and $\big\|\boldsymbol{\Lambda}^{(t),s}\big\|_{\mathsf{op}} \le \frac{1}{p}$, and (iii) follows from (B.10).

B.4.3 Linear system

Let $\boldsymbol{e}^{(t),s} = \begin{bmatrix} L^2\mathbb{E}\big\|\boldsymbol{u}^{(t),s} - \mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),s}\big\|_2^2 \\ \mathbb{E}\big\|\boldsymbol{v}^{(t),s} - \mathbf{1}_n \otimes \overline{\boldsymbol{v}}^{(t),s}\big\|_2^2 \end{bmatrix}$, and $\boldsymbol{b}^{(t),s} = \frac{2\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}}^2)p^2}\begin{bmatrix} 0 \\ n\mathbb{E}\big\|\overline{\boldsymbol{v}}^{(t),s}\big\|_2^2 \end{bmatrix}$. By taking expectation of (B.12) and (B.13), we can construct the following linear system

$$
\begin{aligned}
\boldsymbol{e}^{(t),s} &\le \begin{bmatrix} \frac{2\alpha_{\mathsf{in}}^2}{1+\alpha_{\mathsf{in}}^2} & \frac{2\alpha_{\mathsf{in}}^2\eta^2L^2}{1-\alpha_{\mathsf{in}}^2} \\ \frac{16\alpha_{\mathsf{in}}^2}{(1-\alpha_{\mathsf{in}}^2)p^2} & \frac{2\alpha_{\mathsf{in}}^2}{1+\alpha_{\mathsf{in}}^2} + \frac{4\alpha_{\mathsf{in}}^4\eta^2L^2}{(1-\alpha_{\mathsf{in}}^2)p^2} \end{bmatrix}\boldsymbol{e}^{(t),s-1} + \boldsymbol{b}^{(t),s-1} \\
&\le \underbrace{\begin{bmatrix} \alpha_{\mathsf{in}} & \frac{2\alpha_{\mathsf{in}}^2\eta^2L^2}{1-\alpha_{\mathsf{in}}} \\ \frac{16\alpha_{\mathsf{in}}^2}{(1-\alpha_{\mathsf{in}})p^2} & \alpha_{\mathsf{in}} + \frac{4\alpha_{\mathsf{in}}^4\eta^2L^2}{(1-\alpha_{\mathsf{in}})p^2} \end{bmatrix}}_{=:\boldsymbol{G}_{\mathsf{in}}}\boldsymbol{e}^{(t),s-1} + \boldsymbol{b}^{(t),s-1} = \boldsymbol{G}_{\mathsf{in}}\boldsymbol{e}^{(t),s-1} + \boldsymbol{b}^{(t),s-1},
\end{aligned}
\tag{B.14}
$$

where the second inequality is due to $2\alpha_{\mathsf{in}} < 1 + \alpha_{\mathsf{in}}^2$ and $1 + \alpha_{\mathsf{in}} \ge 1$. Telescope the above inequality to obtain

$$
\boldsymbol{e}^{(t),s} \le \boldsymbol{G}_{\mathsf{in}}^s\boldsymbol{e}^{(t),0} + \sum_{k=1}^{s}\boldsymbol{G}_{\mathsf{in}}^{s-k}\boldsymbol{b}^{(t),k-1}. \tag{B.15}
$$

Thus, the sum of the consensus errors can be bounded by

$$\begin{aligned}\sum_{s=0}^{S-1}\boldsymbol{e}^{(t),s} &\le \boldsymbol{e}^{(t),0}+\sum_{s=1}^{S-1}\Big(\boldsymbol{G}_{\mathsf{in}}^{s}\boldsymbol{e}^{(t),0}+\sum_{k=1}^{s}\boldsymbol{G}_{\mathsf{in}}^{s-k}\boldsymbol{b}_i^{(t),k-1}\Big)\\ &= \sum_{s=0}^{S-1}\boldsymbol{G}_{\mathsf{in}}^{s}\boldsymbol{e}^{(t),0}+\sum_{s=1}^{S-1}\sum_{k=1}^{s}\boldsymbol{G}_{\mathsf{in}}^{s-k}\boldsymbol{b}^{(t),k-1}\\ &\overset{\text{(i)}}{=} \sum_{s=0}^{S-1}\boldsymbol{G}_{\mathsf{in}}^{s}\boldsymbol{e}^{(t),0}+\sum_{k=1}^{S-1}\sum_{s=0}^{S-1-k}\boldsymbol{G}_{\mathsf{in}}^{s}\boldsymbol{b}^{(t),k-1}\\ &\overset{\text{(ii)}}{\le} \sum_{s=0}^{S-1}\boldsymbol{G}_{\mathsf{in}}^{s}\boldsymbol{e}^{(t),0}+\sum_{k=1}^{S-1}\sum_{s=0}^{S-1}\boldsymbol{G}_{\mathsf{in}}^{s}\boldsymbol{b}^{(t),k-1}\\ &\overset{\text{(iii)}}{\le} \sum_{s=0}^{\infty}\boldsymbol{G}_{\mathsf{in}}^{s}\Big(\boldsymbol{e}^{(t),0}+\sum_{s=0}^{S-1}\boldsymbol{b}^{(t),s}\Big)\end{aligned} \tag{B.16}$$

where (i) follows by changing the order of summation, (ii) and (iii) follows from the nonnegativity of $\boldsymbol{G}_{\mathsf{in}}$ and $\boldsymbol{b}^{(t),s}$ respectively. To continue, we begin with the following claim about $\boldsymbol{G}_{\mathsf{in}}$ which will be proved momentarily.

Claim 1. *Under the choice of η in Theorem 6, the eigenvalues of $\boldsymbol{G}_{\mathsf{in}}$ are in $(-1,1)$, and the Neumann series converges,*

$$\sum_{s=0}^{\infty}\boldsymbol{G}_{\mathsf{in}}^{s}=(\boldsymbol{I}_2-\boldsymbol{G}_{\mathsf{in}})^{-1}\le\begin{bmatrix}\frac{2}{1-\alpha_{\mathsf{in}}} & \frac{4\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})^3}\\ \frac{32\alpha_{\mathsf{in}}^2}{(1-\alpha_{\mathsf{in}})^3p^2} & \frac{2}{1-\alpha_{\mathsf{in}}}\end{bmatrix}. \tag{B.17}$$

Let $\boldsymbol{\varsigma}_{\mathsf{in}}^{\top}=\begin{bmatrix}8\left(\frac{4S}{pnb}+1\right) & \frac{8S\alpha_{\mathsf{in}}^2\eta^2L^2}{pnb}\end{bmatrix}$, in view of Claim 1, the summation of consensus erros in (B.11) can be bounded as

$$\begin{aligned}&\Big(\frac{4S}{npb}+1\Big)\cdot 8L^2\sum_{s=0}^{S-1}\mathbb{E}\big\|\boldsymbol{u}^{(t),s}-\mathbf{1}_n\otimes\overline{\boldsymbol{u}}^{(t),s}\big\|_2^2+\frac{8S\alpha_{\mathsf{in}}^2\eta^2L^2}{npb}\sum_{s=0}^{S-1}\mathbb{E}\big\|\boldsymbol{v}^{(t),s}-\mathbf{1}_n\otimes\overline{\boldsymbol{v}}^{(t),s}\big\|_2^2\\ &=\boldsymbol{\varsigma}_{\mathsf{in}}^{\top}\sum_{s=0}^{S-1}\boldsymbol{e}^{(t),s}\\ &\le\boldsymbol{\varsigma}_{\mathsf{in}}^{\top}\Big(\sum_{s=0}^{\infty}\boldsymbol{G}_{\mathsf{in}}^{s}\Big)\Big(\boldsymbol{e}^{(t),0}+\sum_{k=0}^{S-1}\boldsymbol{b}^{(t),k}\Big)\\ &\le\boldsymbol{\varsigma}_{\mathsf{in}}^{\top}(\boldsymbol{I}_2-\boldsymbol{G}_{\mathsf{in}})^{-1}\Big(\boldsymbol{e}^{(t),0}+\sum_{s=0}^{S-1}\boldsymbol{b}^{(t),s}\Big),\end{aligned}$$

and

$$\begin{aligned}\boldsymbol{\varsigma}_{\mathsf{in}}^{\top}(\boldsymbol{I}_2-\boldsymbol{G}_{\mathsf{in}})^{-1}&\le\begin{bmatrix}\frac{16}{1-\alpha_{\mathsf{in}}}\left(\frac{4S}{npb}+1\right)+\frac{32\alpha_{\mathsf{in}}^2}{(1-\alpha_{\mathsf{in}})^3p^2}\cdot\frac{8S\alpha_{\mathsf{in}}^2\eta^2L^2}{pnb} & \frac{32\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})^3}\cdot\left(\frac{4S}{npb}+1\right)+\frac{2}{1-\alpha_{\mathsf{in}}}\cdot\frac{8S\alpha_{\mathsf{in}}^2\eta^2L^2}{npb}\end{bmatrix}\\ &\le\begin{bmatrix}\frac{16}{1-\alpha_{\mathsf{in}}}\left(\frac{4S}{npb}+1\right)+\frac{3\alpha_{\mathsf{in}}^2}{(1-\alpha_{\mathsf{in}})p^2} & \frac{128\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})^3}\cdot\left(\frac{S}{npb}+1\right)+\frac{16\alpha_{\mathsf{in}}^2(1-\alpha_{\mathsf{in}})}{100}\end{bmatrix}\\ &\le\begin{bmatrix}\frac{64}{1-\alpha_{\mathsf{in}}}\left(\frac{S}{npb}+1\right) & 2\alpha_{\mathsf{in}}^2\end{bmatrix},\end{aligned}$$

where we use (3.3), $\eta L \le \frac{(1-\alpha_{\mathsf{in}})^3(1-\alpha_{\mathsf{out}})}{10\left(1+\alpha_{\mathsf{in}}\alpha_{\mathsf{out}}\sqrt{npb}\right)\left(\sqrt{S/(npb)}+1\right)} \le \frac{(1-\alpha_{\mathsf{in}})^3(1-\alpha_{\mathsf{out}})}{10\left(\sqrt{S/(pnb)}+1\right)}$, to prove the last two inequalities.

Therefore, (B.11) can be bounded as

$$\begin{aligned}
&2L^2\sum_{s=0}^{S-1}\mathbb{E}\left\|\boldsymbol{u}^{(t),s}-\mathbf{1}_n\otimes\overline{\boldsymbol{u}}^{(t),s}\right\|_2^2+2n\sum_{s=0}^{S-1}\mathbb{E}\left\|\nabla f(\overline{\boldsymbol{u}}^{(t),s})-\overline{\boldsymbol{v}}^{(t),s}\right\|_2^2\\
&\qquad\le \boldsymbol{\varsigma}_{\mathsf{in}}^{\top}\left(\boldsymbol{I}_2-\boldsymbol{G}_{\mathsf{in}}\right)^{-1}\left(\boldsymbol{e}^{(t),0}+\sum_{s=0}^{S-1}\boldsymbol{b}^{(t),s}\right)+\frac{4S\eta^2L^2}{pb}\sum_{s=0}^{S-1}\mathbb{E}\left\|\overline{\boldsymbol{v}}^{(t),s}\right\|_2^2\\
&\qquad\le \frac{64L^2}{1-\alpha_{\mathsf{in}}}\cdot\left(\frac{S}{npb}+1\right)\mathbb{E}\left\|\boldsymbol{x}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{x}}^{(t-1)}\right\|_2^2+2\alpha_{\mathsf{in}}^2\mathbb{E}\left\|\boldsymbol{s}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{s}}^{(t-1)}\right\|_2^2\\
&\qquad\quad+\left(\frac{4\alpha_{\mathsf{in}}^4\eta^2L^2}{(1-\alpha_{\mathsf{in}})^2p^2}+\frac{4S\eta^2L^2}{npb}\right)\cdot n\sum_{s=1}^{S}\mathbb{E}\left\|\overline{\boldsymbol{v}}^{(t),s-1}\right\|_2^2\\
&\qquad< \frac{64L^2}{1-\alpha_{\mathsf{in}}}\cdot\left(\frac{S}{npb}+1\right)\mathbb{E}\left\|\boldsymbol{x}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{x}}^{(t-1)}\right\|_2^2+2\alpha_{\mathsf{in}}^2\mathbb{E}\left\|\boldsymbol{s}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{s}}^{(t-1)}\right\|_2^2\\
&\qquad\quad+\frac{2n}{25}\sum_{s=1}^{S}\mathbb{E}\left\|\overline{\boldsymbol{v}}^{(t),s-1}\right\|_{2'}^2
\end{aligned}$$

where the last inequality is proved by incorporating (3.3) as $\frac{4\alpha_{\mathsf{in}}^4\eta^2L^2}{(1-\alpha_{\mathsf{in}})^2p^2}\le\frac{4\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})^2}<\frac{4\alpha_{\mathsf{in}}^2}{(1-\alpha_{\mathsf{in}})^2}\cdot\frac{(1-\alpha_{\mathsf{in}})^6}{100}\le\frac{1}{25}$ and $\frac{4S\eta^2L^2}{npb}\le\frac{S}{npb}\cdot\frac{4}{100\left(\sqrt{S/(npb)}+1\right)^2}<\frac{1}{25}$.

Proof of Claim 1. By the definition of $\boldsymbol{G}_{\mathsf{in}}$ in (B.14), the characteristic polynomial of $\boldsymbol{G}_{\mathsf{in}}$ is

$$f(\lambda)=(\alpha_{\mathsf{in}}-\lambda)\left(\alpha_{\mathsf{in}}+\frac{4\alpha_{\mathsf{in}}^4\eta^2L^2}{(1-\alpha_{\mathsf{in}})p^2}-\lambda\right)-\frac{32\alpha_{\mathsf{in}}^4\eta^2L^2}{(1-\alpha_{\mathsf{in}})^2p^2}.$$

By (3.3), $\eta L\le\frac{(1-\alpha_{\mathsf{in}})^3(1-\alpha_{\mathsf{out}})}{10\left(1+\alpha_{\mathsf{in}}\alpha_{\mathsf{out}}\sqrt{npb}\right)\left(\sqrt{S/(npb)}+1\right)}\le\frac{(1-\alpha_{\mathsf{in}})^3}{10}$ and $\alpha_{\mathsf{in}}\le p$, we have $\frac{32\alpha_{\mathsf{in}}^4\eta^2L^2}{(1-\alpha_{\mathsf{in}})^2p^2}\le\frac{32\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})^2}\le\frac{32}{100}\alpha_{\mathsf{in}}^2(1-\alpha_{\mathsf{in}})^4<1$, so that $f(-1)\ge1-\frac{32\alpha_{\mathsf{in}}^4\eta^2L^2}{(1-\alpha_{\mathsf{in}})^2}>0$, and

$$\begin{aligned}
f(1)&=(1-\alpha_{\mathsf{in}})^2-\frac{4\alpha_{\mathsf{in}}^4\eta^2L^2}{p^2}-\frac{32\alpha_{\mathsf{in}}^4\eta^2L^2}{(1-\alpha_{\mathsf{in}})^2p^2}\\
&\ge(1-\alpha_{\mathsf{in}})^2-\frac{36\alpha_{\mathsf{in}}^4\eta^2L^2}{(1-\alpha_{\mathsf{in}})^2p^2}\\
&>(1-\alpha_{\mathsf{in}})^2-\frac{36}{100}(1-\alpha_{\mathsf{in}})^4>0.
\end{aligned}$$

Because $f(\alpha_{\mathsf{in}})\le0$, all eigenvalues of $\boldsymbol{G}_{\mathsf{in}}$ are in $(-1,1)$, then the Neumann series converges, yielding

$$\begin{aligned}
\sum_{s=0}^{\infty}\boldsymbol{G}_{\mathsf{in}}^s&=\left(\boldsymbol{I}_2-\boldsymbol{G}_{\mathsf{in}}\right)^{-1}\\
&=\frac{1-\alpha_{\mathsf{in}}}{(1-\alpha_{\mathsf{in}})^4p^2-4\left((1-\alpha_{\mathsf{in}})^2+8\right)\alpha_{\mathsf{in}}^4\eta^2L^2}\begin{bmatrix}(1-\alpha_{\mathsf{in}})^2p^2-4\alpha_{\mathsf{in}}^4\eta^2L^2 & 2\alpha_{\mathsf{in}}^2\eta^2L^2p^2\\ 16\alpha_{\mathsf{in}}^2 & (1-\alpha_{\mathsf{in}})^2p^2\end{bmatrix}\\
&\le\frac{1-\alpha_{\mathsf{in}}}{(1-\alpha_{\mathsf{in}})^4p^2-4\left((1-\alpha_{\mathsf{in}})^2+8\right)\alpha_{\mathsf{in}}^4\eta^2L^2}\begin{bmatrix}(1-\alpha_{\mathsf{in}})^2p^2 & 2\alpha_{\mathsf{in}}^2\eta^2L^2p^2\\ 16\alpha_{\mathsf{in}}^2 & (1-\alpha_{\mathsf{in}})^2p^2\end{bmatrix}
\end{aligned}$$

$$\overset{\text{(i)}}{\leq} \frac{1-\alpha_{\text{in}}}{(1-\alpha_{\text{in}})^4 p^2 - 36\alpha_{\text{in}}^4\eta^2L^2} \begin{bmatrix} (1-\alpha_{\text{in}})^2p^2 & 2\alpha_{\text{in}}^2\eta^2L^2p^2 \\ 16\alpha_{\text{in}}^2 & (1-\alpha_{\text{in}})^2p^2 \end{bmatrix}$$

$$\overset{\text{(ii)}}{\leq} \begin{bmatrix} \frac{2}{1-\alpha_{\text{in}}} & \frac{4\alpha_{\text{in}}^2\eta^2L^2}{(1-\alpha_{\text{in}})^3} \\ \frac{32\alpha_{\text{in}}^2}{(1-\alpha_{\text{in}})^3p^2} & \frac{2}{1-\alpha_{\text{in}}} \end{bmatrix},$$

where (i) and (ii) follow the fact $(1-\alpha_{\text{in}})^2 \leq 1$, and $(1-\alpha_{\text{in}})^4p^2 - 36\alpha_{\text{in}}^4\eta^2L^2 \geq (1-\alpha_{\text{in}})^4p^2 - \frac{36}{100}\alpha_{\text{in}}^4(1-\alpha_{\text{in}})^6 \geq (1-\alpha_{\text{in}})^4p^2 - \frac{36}{100}\alpha_{\text{in}}^2(1-\alpha_{\text{in}})^6p^2 > \frac{1}{2}(1-\alpha_{\text{in}})^4p^2$ due to (3.3).

□

B.5 Proof of Lemma 6

This section proves Lemma 6. In the following subsections, Appendices B.5.1 and B.5.2 derive induction inequalities for the consensus errors and Appendix B.5.3 creates a linear system of consensus errors to compute the summation.

B.5.1 Sum of outer loop variable consensus errors

The variable consensus error can be bounded deterministically as following,

$$\begin{aligned}
&\left\|\boldsymbol{x}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{x}}^{(t)}\right\|_2^2 \\
&= \left\|\left(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\right)\boldsymbol{x}^{(t)}\right\|_2^2 \\
&\overset{\text{(i)}}{=} \left\|\left(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\right)\boldsymbol{u}^{(t),S}\right\|_2^2 \\
&\overset{\text{(ii)}}{=} \left\|\left(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\right)(\boldsymbol{W}_{\text{in}} \otimes \boldsymbol{I}_d)\left(\boldsymbol{u}^{(t),S-1} - \eta\boldsymbol{v}^{(t),S-1}\right)\right\|_2^2 \\
&\leq \alpha_{\text{in}}^2\left\|\left(\boldsymbol{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) \otimes \boldsymbol{I}_d\right)\left(\boldsymbol{u}^{(t),S-1} - \eta\boldsymbol{v}^{(t),S-1}\right)\right\|_2^2 \\
&\leq \frac{2\alpha_{\text{in}}^2}{1+\alpha_{\text{in}}^2}\left\|\boldsymbol{u}^{(t),S-1} - \mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),S-1}\right\|_2^2 + \frac{2\alpha_{\text{in}}^2\eta^2}{1-\alpha_{\text{in}}^2}\left\|\boldsymbol{v}^{(t),S-1} - \mathbf{1}_n \otimes \overline{\boldsymbol{v}}^{(t),S-1}\right\|_2^2,
\end{aligned}$$

where (i) uses $\boldsymbol{x}^{(t)} = \boldsymbol{u}^{(t),S}$, (ii) uses the update rule (3.1a), and the last two inequalities follow from similar reasoning as (B.12). Apply the same reasoning to $\frac{2\alpha_{\text{in}}^2}{1+\alpha_{\text{in}}^2}\left\|\boldsymbol{u}^{(t),S-1} - \mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),S-1}\right\|_2^2$ and use $\frac{2\alpha_{\text{in}}^2}{1+\alpha_{\text{in}}^2} \leq 1$, we can prove

$$\begin{aligned}
\left\|\boldsymbol{x}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{x}}^{(t)}\right\|_2^2 &\leq \left(\frac{2\alpha_{\text{in}}^2}{1+\alpha_{\text{in}}^2}\right)^S\left\|\boldsymbol{u}^{(t),0} - \mathbf{1}_n \otimes \overline{\boldsymbol{u}}^{(t),0}\right\|_2^2 + \frac{2\alpha_{\text{in}}^2\eta^2}{1-\alpha_{\text{in}}^2}\sum_{s=0}^{S-1}\left\|\boldsymbol{v}^{(t),s} - \mathbf{1}_n \otimes \overline{\boldsymbol{v}}^{(t),s}\right\|_2^2 \\
&= \left(\frac{2\alpha_{\text{in}}^2}{1+\alpha_{\text{in}}^2}\right)^S\left\|\boldsymbol{x}^{(t-1)} - \mathbf{1}_n \otimes \overline{\boldsymbol{x}}^{(t-1)}\right\|_2^2 + \frac{2\alpha_{\text{in}}^2\eta^2}{1-\alpha_{\text{in}}^2}\sum_{s=0}^{S-1}\left\|\boldsymbol{v}^{(t),s} - \mathbf{1}_n \otimes \overline{\boldsymbol{v}}^{(t),s}\right\|_2^2, \qquad \text{(B.18)}
\end{aligned}$$

where the last equality follows from $\boldsymbol{x}^{(t-1)} = \boldsymbol{u}^{(t),0}$.

Take expectation of the previous inequality, by (B.16), we can further compute the summation in (B.18) as follows

$$\sum_{s=0}^{S-1}\mathbb{E}\left\|\boldsymbol{v}^{(t),s}-\mathbf{1}_n\otimes\overline{\boldsymbol{v}}^{(t),s}\right\|_2^2\leq\frac{32\alpha_{\mathsf{in}}^2L^2}{(1-\alpha_{\mathsf{in}})^3p^2}\mathbb{E}\left\|\boldsymbol{x}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{x}}^{(t-1)}\right\|_2^2 + \frac{2}{1-\alpha_{\mathsf{in}}}\Big(\mathbb{E}\left\|\boldsymbol{s}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{s}}^{(t-1)}\right\|_2^2+\frac{2\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}}^2)p^2}\cdot n\sum_{s=0}^{S-1}\mathbb{E}\left\|\overline{\boldsymbol{v}}^{(t),s}\right\|\Big).$$

Together with $\boldsymbol{x}^{(t)}=\boldsymbol{u}^{(t),0}$ and $\boldsymbol{s}^{(t)}=\boldsymbol{v}^{(t),0}$, (B.18) can be further bounded as

$$\begin{aligned}\mathbb{E}\left\|\boldsymbol{x}^{(t)}-\mathbf{1}_n\otimes\overline{\boldsymbol{x}}^{(t)}\right\|_2^2&\leq\left(\Big(\frac{2\alpha_{\mathsf{in}}^2}{1+\alpha_{\mathsf{in}}^2}\Big)^S+\frac{2\alpha_{\mathsf{in}}^2\eta^2L^2}{1-\alpha_{\mathsf{in}}^2}\cdot\frac{32\alpha_{\mathsf{in}}^2}{(1-\alpha_{\mathsf{in}})^3p^2}\right)\mathbb{E}\left\|\boldsymbol{x}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{x}}^{(t-1)}\right\|_2^2\\&\quad+\frac{2\alpha_{\mathsf{in}}^2\eta^2}{1-\alpha_{\mathsf{in}}^2}\cdot\frac{2}{1-\alpha_{\mathsf{in}}}\Big(\left\|\boldsymbol{s}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{s}}^{(t-1)}\mathbb{E}\right\|_2^2+\frac{2\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}}^2)p^2}\cdot n\sum_{s=0}^{S-1}\mathbb{E}\left\|\overline{\boldsymbol{v}}^{(t),s}\right\|\Big)\\&<\alpha_{\mathsf{in}}\mathbb{E}\left\|\boldsymbol{x}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{x}}^{(t-1)}\right\|_2^2\\&\quad+\frac{4\alpha_{\mathsf{in}}^2\eta^2}{(1-\alpha_{\mathsf{in}})^2}\Big(\mathbb{E}\left\|\boldsymbol{s}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{s}}^{(t-1)}\right\|_2^2+\frac{2\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})p^2}\cdot n\sum_{s=0}^{S-1}\mathbb{E}\left\|\overline{\boldsymbol{v}}^{(t),s}\right\|\Big).\end{aligned}\tag{B.19}$$

The last inequality is obtained by using (3.3) and the fact that $0\leq\alpha_{\mathsf{in}}<1$ as follows

$$\begin{aligned}\Big(\frac{2\alpha_{\mathsf{in}}^2}{1+\alpha_{\mathsf{in}}^2}\Big)^S+\frac{2\alpha_{\mathsf{in}}^2\eta^2L^2}{1-\alpha_{\mathsf{in}}^2}\cdot\frac{32\alpha_{\mathsf{in}}^2}{(1-\alpha_{\mathsf{in}})^3p^2}&=\Big(\frac{2\alpha_{\mathsf{in}}^2}{1+\alpha_{\mathsf{in}}^2}\Big)^S+\frac{\alpha_{\mathsf{in}}^2(1-\alpha_{\mathsf{in}})^2}{1+\alpha_{\mathsf{in}}}\cdot\frac{64\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})^6p^2}\\&<\frac{2\alpha_{\mathsf{in}}^2}{1+\alpha_{\mathsf{in}}^2}+\frac{64}{100}\cdot\frac{\alpha_{\mathsf{in}}^2(1-\alpha_{\mathsf{in}})^2}{1+\alpha_{\mathsf{in}}}\\&\leq\frac{2\alpha_{\mathsf{in}}^2}{1+\alpha_{\mathsf{in}}^2}+\frac{\alpha_{\mathsf{in}}(1-\alpha_{\mathsf{in}})^2}{1+\alpha_{\mathsf{in}}^2}=\alpha_{\mathsf{in}}.\end{aligned}$$

B.5.2 Sum of outer loop gradient estimation consensus errors

In view of the update rule for the gradient tracking term (3.2) and reorganize terms,

$$\begin{aligned}\left\|\boldsymbol{s}^{(t)}-\mathbf{1}_n\otimes\overline{\boldsymbol{s}}^{(t)}\right\|_2^2&=\left\|\Big(\boldsymbol{I}_{nd}-(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\boldsymbol{I}_d\Big)\boldsymbol{s}^{(t)}\right\|_2^2\\&=\left\|\Big(\boldsymbol{I}_{nd}-(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\boldsymbol{I}_d\Big)(\boldsymbol{W}_{\mathsf{out}}\otimes\boldsymbol{I}_d)\Big(\boldsymbol{s}^{(t-1)}+\nabla F(\boldsymbol{x}^{(t)})-\nabla F(\boldsymbol{x}^{(t-1)})\Big)\right\|_2^2\\&\leq\frac{2\alpha_{\mathsf{out}}^2}{1+\alpha_{\mathsf{out}}^2}\left\|\boldsymbol{s}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{s}}^{(t-1)}\right\|_2^2\\&\quad+\frac{2\alpha_{\mathsf{out}}^2}{1-\alpha_{\mathsf{out}}^2}\left\|\Big(\boldsymbol{I}_{nd}-(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\boldsymbol{I}_d\Big)\Big(\nabla F(\boldsymbol{x}^{(t)})-\nabla F(\boldsymbol{x}^{(t-1)})\Big)\right\|_2^2,\end{aligned}\tag{B.20}$$

which follows from similar reasonings as (B.12). The second term can be further decomposed as

$$\begin{aligned}&\left\|\Big(\boldsymbol{I}_{nd}-(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\boldsymbol{I}_d\Big)\Big(\nabla F(\boldsymbol{x}^{(t)})-\nabla F(\boldsymbol{x}^{(t-1)})\Big)\right\|_2^2\\&\leq\left\|\nabla F(\boldsymbol{x}^{(t)})-\nabla F(\boldsymbol{x}^{(t-1)})\right\|_2^2\\&\leq L^2\left\|(\boldsymbol{x}^{(t)}-\mathbf{1}_n\otimes\overline{\boldsymbol{x}}^{(t)})-(\boldsymbol{x}^{(t-1)}-\mathbf{1}_n\otimes\overline{\boldsymbol{x}}^{(t-1)})+(\mathbf{1}_n\otimes\overline{\boldsymbol{x}}^{(t)}-\mathbf{1}_n\otimes\overline{\boldsymbol{x}}^{(t-1)})\right\|_2^2\end{aligned}$$

$$
\begin{aligned}
&= L^2\big\|(\boldsymbol{x}^{(t)} - \mathbf{1}_n \otimes \bar{\boldsymbol{x}}^{(t)}) - (\boldsymbol{x}^{(t-1)} - \mathbf{1}_n \otimes \bar{\boldsymbol{x}}^{(t-1)})\big\|_2^2 + nL^2\big\|\bar{\boldsymbol{x}}^{(t)} - \bar{\boldsymbol{x}}^{(t-1)}\big\|_2^2 \\
&\le 2L^2\big\|\boldsymbol{x}^{(t)} - \mathbf{1}_n \otimes \bar{\boldsymbol{x}}^{(t)}\big\|_2^2 + 2L^2\big\|\boldsymbol{x}^{(t-1)} - \mathbf{1}_n \otimes \bar{\boldsymbol{x}}^{(t-1)}\big\|_2^2 + S\eta^2L^2 \cdot n\sum_{s=0}^{S-1}\big\|\bar{\boldsymbol{v}}^{(t),s}\big\|_2^2, \qquad \text{(B.21)}
\end{aligned}
$$

where the last line follows from the update rule (3.1a) by identifying $\bar{\boldsymbol{x}}^{(t)} - \bar{\boldsymbol{x}}^{(t-1)} = \eta\sum_{s=0}^{S-1}\bar{\boldsymbol{v}}^{(t),s}$ and Cauchy-Schwartz inequality.

With (B.21), (B.20) can be further bounded as follows

$$
\begin{aligned}
\big\|\boldsymbol{s}^{(t)} - \mathbf{1}_n \otimes \bar{\boldsymbol{s}}^{(t)}\big\|_2^2 \le{}& \frac{2\alpha_{\mathsf{out}}^2}{1+\alpha_{\mathsf{out}}^2}\big\|\boldsymbol{s}^{(t-1)} - \mathbf{1}_n \otimes \bar{\boldsymbol{s}}^{(t-1)}\big\|_2^2 + \frac{2\alpha_{\mathsf{out}}^2}{1-\alpha_{\mathsf{out}}^2}\Big(2L^2\big\|\boldsymbol{x}^{(t)} - \mathbf{1}_n \otimes \bar{\boldsymbol{x}}^{(t)}\big\|_2^2 \\
&+ 2L^2\big\|\boldsymbol{x}^{(t-1)} - \mathbf{1}_n \otimes \bar{\boldsymbol{x}}^{(t-1)}\big\|_2^2 + S\eta^2L^2 \cdot n\sum_{s=0}^{S-1}\big\|\bar{\boldsymbol{v}}^{(t),s}\big\|_2^2\Big) \\
\le{}& \alpha_{\mathsf{out}}\big\|\boldsymbol{s}^{(t-1)} - \mathbf{1}_n \otimes \bar{\boldsymbol{s}}^{(t-1)}\big\|_2^2 + \frac{2\alpha_{\mathsf{out}}^2}{1-\alpha_{\mathsf{out}}}\Big(2L^2\big\|\boldsymbol{x}^{(t)} - \mathbf{1}_n \otimes \bar{\boldsymbol{x}}^{(t)}\big\|_2^2 \\
&+ 2L^2\big\|\boldsymbol{x}^{(t-1)} - \mathbf{1}_n \otimes \bar{\boldsymbol{x}}^{(t-1)}\big\|_2^2 + S\eta^2L^2 \cdot n\sum_{s=0}^{S-1}\big\|\bar{\boldsymbol{v}}^{(t),s}\big\|_2^2\Big). \qquad \text{(B.22)}
\end{aligned}
$$

Combine with (B.19), after taking expectations, (B.22) can be further bounded as

$$
\begin{aligned}
\mathbb{E}\big\|\boldsymbol{s}^{(t)} - \mathbf{1}_n \otimes \bar{\boldsymbol{s}}^{(t)}\big\|_2^2 <{}& \alpha_{\mathsf{out}}\mathbb{E}\big\|\boldsymbol{s}^{(t-1)} - \mathbf{1}_n \otimes \bar{\boldsymbol{s}}^{(t-1)}\big\|_2^2 + \frac{4\alpha_{\mathsf{out}}^2L^2}{1-\alpha_{\mathsf{out}}}\mathbb{E}\big\|\boldsymbol{x}^{(t-1)} - \mathbf{1}_n \otimes \bar{\boldsymbol{x}}^{(t-1)}\big\|_2^2 \\
&+ \frac{2\alpha_{\mathsf{out}}^2S\eta^2L^2}{1-\alpha_{\mathsf{out}}} \cdot n\sum_{s=0}^{S-1}\mathbb{E}\big\|\bar{\boldsymbol{v}}^{(t),s}\big\|_2^2 + \frac{4\alpha_{\mathsf{out}}^2L^2}{1-\alpha_{\mathsf{out}}}\Bigg(\alpha_{\mathsf{in}}\mathbb{E}\big\|\boldsymbol{x}^{(t-1)} - \mathbf{1}_n \otimes \bar{\boldsymbol{x}}^{(t-1)}\big\|_2^2 \\
&+ \frac{4\alpha_{\mathsf{in}}^2\eta^2}{(1-\alpha_{\mathsf{in}})^2}\Big(\mathbb{E}\big\|\boldsymbol{s}^{(t-1)} - \mathbf{1}_n \otimes \bar{\boldsymbol{s}}^{(t-1)}\big\|_2^2 + \frac{2\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})p^2} \cdot n\sum_{s=0}^{S-1}\mathbb{E}\big\|\bar{\boldsymbol{v}}^{(t),s}\big\|_2^2\Big)\Bigg) \\
={}& \Big(\alpha_{\mathsf{out}} + \frac{4\alpha_{\mathsf{out}}^2L^2}{1-\alpha_{\mathsf{out}}} \cdot \frac{4\alpha_{\mathsf{in}}^2\eta^2}{(1-\alpha_{\mathsf{in}})^2}\Big)\mathbb{E}\big\|\boldsymbol{s}^{(t-1)} - \mathbf{1}_n \otimes \bar{\boldsymbol{s}}^{(t-1)}\big\|_2^2 \\
&+ \frac{4\alpha_{\mathsf{out}}^2L^2}{1-\alpha_{\mathsf{out}}}(1+\alpha_{\mathsf{in}})\mathbb{E}\big\|\boldsymbol{x}^{(t-1)} - \mathbf{1}_n \otimes \bar{\boldsymbol{x}}^{(t-1)}\big\|_2^2 \\
&+ \Big(\frac{2\alpha_{\mathsf{out}}^2S\eta^2L^2}{1-\alpha_{\mathsf{out}}} + \frac{4\alpha_{\mathsf{out}}^2L^2}{1-\alpha_{\mathsf{out}}} \cdot \frac{4\alpha_{\mathsf{in}}^2\eta^2}{(1-\alpha_{\mathsf{in}})^2} \cdot \frac{2\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})p^2}\Big) \cdot n\sum_{s=0}^{S-1}\mathbb{E}\big\|\bar{\boldsymbol{v}}^{(t),s}\big\|_2^2 \\
\overset{\text{(i)}}{<}{}& \Big(\alpha_{\mathsf{out}} + \frac{4\alpha_{\mathsf{out}}^2L^2}{1-\alpha_{\mathsf{out}}} \cdot \frac{4\alpha_{\mathsf{in}}^2\eta^2}{(1-\alpha_{\mathsf{in}})^2}\Big)\mathbb{E}\big\|\boldsymbol{s}^{(t-1)} - \mathbf{1}_n \otimes \bar{\boldsymbol{s}}^{(t-1)}\big\|_2^2 \\
&+ \frac{4\alpha_{\mathsf{out}}^2L^2}{1-\alpha_{\mathsf{out}}}(1+\alpha_{\mathsf{in}})\mathbb{E}\big\|\boldsymbol{x}^{(t-1)} - \mathbf{1}_n \otimes \bar{\boldsymbol{x}}^{(t-1)}\big\|_2^2 \\
&+ \frac{3\alpha_{\mathsf{out}}^2S\eta^2L^2}{1-\alpha_{\mathsf{out}}} \cdot n\sum_{s=0}^{S-1}\mathbb{E}\big\|\bar{\boldsymbol{v}}^{(t),s}\big\|_2^2, \qquad \text{(B.23)}
\end{aligned}
$$

where (i) is obtained by applying the condition in (3.3) as follows

$$
\begin{aligned}
\frac{4\alpha_{\mathsf{out}}^2L^2}{1-\alpha_{\mathsf{out}}} \cdot \frac{4\alpha_{\mathsf{in}}^2\eta^2}{(1-\alpha_{\mathsf{in}})^2} \cdot \frac{2\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})p^2} &= \frac{\alpha_{\mathsf{out}}^2\eta^2L^2}{1-\alpha_{\mathsf{out}}} \cdot \frac{32\alpha_{\mathsf{in}}^4\eta^2L^2}{(1-\alpha_{\mathsf{in}})^3p^2} \\
&\le \frac{\alpha_{\mathsf{out}}^2S\eta^2L^2}{1-\alpha_{\mathsf{out}}} \cdot \frac{32\alpha_{\mathsf{in}}^2(1-\alpha_{\mathsf{in}})^6}{100(1-\alpha_{\mathsf{in}})^3} \\
&\le \frac{\alpha_{\mathsf{out}}^2S\eta^2L^2}{1-\alpha_{\mathsf{out}}},
\end{aligned}
$$

where the inequalities are obtained by using $S \geq 1$ and $0 \leq \alpha_{\mathsf{in}} < 1$.

B.5.3 Linear system

Defining $\boldsymbol{e}^{(t)} := \boldsymbol{e}^{(t),0} = \begin{bmatrix} L^2\mathbb{E}\left\|\boldsymbol{x}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{x}}^{(t)}\right\|_2^2 \\ \mathbb{E}\left\|\boldsymbol{s}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{s}}^{(t)}\right\|_2^2 \end{bmatrix}$ and $\boldsymbol{b}'^{(t)} = \begin{bmatrix} \frac{8\alpha_{\mathsf{in}}^4\eta^4L^4}{(1-\alpha_{\mathsf{in}})^3p^2} \cdot n\sum_{s=0}^{S-1}\mathbb{E}\left\|\overline{\boldsymbol{v}}^{(t),s}\right\|_2^2 \\ \frac{3\alpha_{\mathsf{out}}^2 S\eta^2L^2}{1-\alpha_{\mathsf{out}}} \cdot n\sum_{s=0}^{S-1}\mathbb{E}\left\|\overline{\boldsymbol{v}}^{(t),s}\right\|_2^2 \end{bmatrix}$, we construct a linear system by putting together (B.19) and (B.23) as

$$\boldsymbol{e}^{(t)} \leq \underbrace{\begin{bmatrix} \alpha_{\mathsf{in}} & \frac{4\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})^2} \\ \frac{4\alpha_{\mathsf{out}}^2}{1-\alpha_{\mathsf{out}}}(1+\alpha_{\mathsf{in}}) & \alpha_{\mathsf{out}} + \frac{4\alpha_{\mathsf{out}}^2}{1-\alpha_{\mathsf{out}}} \cdot \frac{4\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})^2} \end{bmatrix}}_{=:\boldsymbol{G}_{\mathsf{out}}} \boldsymbol{e}^{(t-1)} + \boldsymbol{b}'^{(t)} = \boldsymbol{G}_{\mathsf{out}}\boldsymbol{e}^{(t-1)} + \boldsymbol{b}'^{(t)}. \tag{B.24}$$

Then, following the same argument as (B.16), we obtain

$$\sum_{t=1}^{T}\boldsymbol{e}^{(t)} \leq \sum_{t=1}^{\infty}\boldsymbol{G}_{\mathsf{out}}^t\left(\boldsymbol{e}^{(0)} + \sum_{t=1}^{T}\boldsymbol{b}'^{(t)}\right). \tag{B.25}$$

Before continuing, we state the following claim about $\boldsymbol{G}_{\mathsf{out}}$ which will be proven momentarily.

Claim 2. *Under the choice of η in Theorem 6, the eigenvalues of $\boldsymbol{G}_{\mathsf{out}}$ are in $(-1,1)$, and the Neumann series converges,*

$$\sum_{t=0}^{\infty}\boldsymbol{G}_{\mathsf{out}}^t = (\boldsymbol{I}_2 - \boldsymbol{G}_{\mathsf{out}})^{-1} \leq \begin{bmatrix} \frac{2}{1-\alpha_{\mathsf{in}}} & \frac{8\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})^3(1-\alpha_{\mathsf{out}})} \\ \frac{16\alpha_{\mathsf{out}}^2}{(1-\alpha_{\mathsf{in}})(1-\alpha_{\mathsf{out}})^2} & \frac{2}{1-\alpha_{\mathsf{out}}} \end{bmatrix}.$$

With Claim 2 in hand, and the fact that $\boldsymbol{e}^{(0)} = \mathbf{0}$, we can bound the summation of outer loop consensus errors by

$$\begin{aligned} &\frac{64L^2}{1-\alpha_{\mathsf{in}}} \cdot \left(\frac{S}{npb} + 1\right)\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{x}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{x}}^{(t)}\right\|_2^2 + \frac{2\alpha_{\mathsf{in}}^2}{1-\alpha_{\mathsf{in}}}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{s}^{(t)} - \mathbf{1}_n \otimes \overline{\boldsymbol{s}}^{(t)}\right\|_2^2 \\ &= \boldsymbol{\varsigma}_{\mathsf{out}}^\top\sum_{t=1}^{T}\boldsymbol{e}^{(t)} \\ &\leq \boldsymbol{\varsigma}_{\mathsf{out}}^\top(\boldsymbol{I}_2 - \boldsymbol{G}_{\mathsf{out}})^{-1}\left(\boldsymbol{e}^{(0)} + \sum_{t=1}^{T}\boldsymbol{b}'^{(t)}\right) \\ &= \boldsymbol{\varsigma}_{\mathsf{out}}^\top(\boldsymbol{I}_2 - \boldsymbol{G}_{\mathsf{out}})^{-1}\sum_{t=1}^{T}\boldsymbol{b}'^{(t)}, \end{aligned} \tag{B.26}$$

where $\boldsymbol{\varsigma}_{\mathsf{out}}^\top = \begin{bmatrix} \frac{64}{1-\alpha_{\mathsf{in}}} \cdot \left(\frac{S}{npb} + 1\right) & \frac{2\alpha_{\mathsf{in}}^2}{1-\alpha_{\mathsf{in}}} \end{bmatrix}$.

Note that by elementary calculations,

$$\begin{aligned} &\boldsymbol{\varsigma}_{\mathsf{out}}^\top(\boldsymbol{I}_2 - \boldsymbol{G}_{\mathsf{out}})^{-1} \\ &\leq \begin{bmatrix} \frac{64}{1-\alpha_{\mathsf{in}}}\left(\frac{S}{npb} + 1\right) & 2\alpha_{\mathsf{in}}^2 \end{bmatrix}\begin{bmatrix} \frac{2}{1-\alpha_{\mathsf{in}}} & \frac{8\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})^3(1-\alpha_{\mathsf{out}})} \\ \frac{16\alpha_{\mathsf{out}}^2}{(1-\alpha_{\mathsf{in}})(1-\alpha_{\mathsf{out}})^2} & \frac{2}{1-\alpha_{\mathsf{out}}} \end{bmatrix} \end{aligned}$$

$$
\begin{aligned}
&= \begin{bmatrix} \frac{64}{1-\alpha_{\mathsf{in}}}\left(\frac{S}{npb}+1\right)\cdot\frac{2}{1-\alpha_{\mathsf{in}}} + \frac{32\alpha_{\mathsf{in}}^2\alpha_{\mathsf{out}}^2}{(1-\alpha_{\mathsf{in}})(1-\alpha_{\mathsf{out}})^2} & \frac{64}{1-\alpha_{\mathsf{in}}}\left(\frac{S}{npb}+1\right)\cdot\frac{8\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})^3(1-\alpha_{\mathsf{out}})} + \frac{4\alpha_{\mathsf{in}}^2}{1-\alpha_{\mathsf{out}}} \end{bmatrix} \\
&\overset{\text{(i)}}{<} \begin{bmatrix} \frac{128}{(1-\alpha_{\mathsf{in}})^2}\left(\frac{S}{npb}+1\right) + \frac{32\alpha_{\mathsf{in}}^2\alpha_{\mathsf{out}}^2}{(1-\alpha_{\mathsf{in}})(1-\alpha_{\mathsf{out}})^2} & 6\alpha_{\mathsf{in}}^2 + \frac{4\alpha_{\mathsf{in}}^2}{1-\alpha_{\mathsf{out}}} \end{bmatrix} \\
&\overset{\text{(ii)}}{<} \begin{bmatrix} \frac{128}{(1-\alpha_{\mathsf{in}})^2}\left(\frac{S}{npb}+1\right) + \frac{32\alpha_{\mathsf{in}}^2\alpha_{\mathsf{out}}^2}{(1-\alpha_{\mathsf{in}})(1-\alpha_{\mathsf{out}})^2} & \frac{10\alpha_{\mathsf{in}}^2}{1-\alpha_{\mathsf{out}}} \end{bmatrix},
\end{aligned}
$$

where we use (3.3) to prove (i), and $1/(1-\alpha_{\mathsf{in}}) \geq 1$ and $1/(1-\alpha_{\mathsf{out}}) \geq 1$ to prove (ii).

Thus, (B.26) can be bounded using (3.3) as

$$
\begin{aligned}
&\varsigma_{\mathsf{out}}^\top (\boldsymbol{I}_2 - \boldsymbol{G}_{\mathsf{out}})^{-1} \begin{bmatrix} \frac{8\alpha_{\mathsf{in}}^4\eta^4L^4}{(1-\alpha_{\mathsf{in}})^3p^2} \\ \frac{3\alpha_{\mathsf{out}}^2S\eta^2L^2}{1-\alpha_{\mathsf{out}}} \end{bmatrix} \\
&\leq \left(\frac{128}{(1-\alpha_{\mathsf{in}})^2}\left(\frac{S}{npb}+1\right) + \frac{32\alpha_{\mathsf{in}}^2\alpha_{\mathsf{out}}^2}{(1-\alpha_{\mathsf{in}})(1-\alpha_{\mathsf{out}})^2}\right)\frac{8\alpha_{\mathsf{in}}^4\eta^4L^4}{(1-\alpha_{\mathsf{in}})^3p^2} + \frac{10\alpha_{\mathsf{in}}^2}{1-\alpha_{\mathsf{out}}}\cdot\frac{3\alpha_{\mathsf{out}}^2S\eta^2L^2}{1-\alpha_{\mathsf{out}}} \\
&= \frac{1024\alpha_{\mathsf{in}}^4\eta^4L^4}{(1-\alpha_{\mathsf{in}})^5p^2}\left(\frac{S}{npb}+1\right) + \frac{256\alpha_{\mathsf{in}}^6\alpha_{\mathsf{out}}^2\eta^4L^4}{(1-\alpha_{\mathsf{in}})^4(1-\alpha_{\mathsf{out}})^2p^2} + \frac{30\alpha_{\mathsf{in}}^2\alpha_{\mathsf{out}}^2npb\cdot S/(npb)}{(1-\alpha_{\mathsf{out}})^2}\cdot\eta^2L^2 \\
&\leq 11\alpha_{\mathsf{in}}^4\eta^2L^2 + 3\alpha_{\mathsf{in}}^6\alpha_{\mathsf{out}}^2\eta^2L^2 + \frac{30}{100} \\
&< \frac{11}{25},
\end{aligned}
$$

which concludes the proof.

Proof of Claim 2. For simplicity, denote $c = \frac{4\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})^2}$ and $d = \frac{4\alpha_{\mathsf{out}}^2}{1-\alpha_{\mathsf{out}}}$. Then $\boldsymbol{G}_{\mathsf{out}}$ can be written as

$$
\boldsymbol{G}_{\mathsf{out}} = \begin{bmatrix} \alpha_{\mathsf{in}} & c \\ d(1+\alpha_{\mathsf{in}}) & \alpha_{\mathsf{out}} + cd \end{bmatrix},
$$

whose characteristic polynomial is

$$
f(\lambda) = (\alpha_{\mathsf{in}} - \lambda)(\alpha_{\mathsf{out}} + cd - \lambda) - (1+\alpha_{\mathsf{in}})cd.
$$

First, note that $f(1)$ can be bounded by

$$
\begin{aligned}
f(1) &= (\alpha_{\mathsf{in}} - 1)(\alpha_{\mathsf{out}} + cd - 1) - (1+\alpha_{\mathsf{in}})cd \\
&= (1-\alpha_{\mathsf{in}})(1-\alpha_{\mathsf{out}}) - 2cd > 0,
\end{aligned}
$$

where the last inequality is due to the choice of η, namely,

$$
cd = \frac{4\alpha_{\mathsf{out}}^2}{1-\alpha_{\mathsf{out}}}\cdot\frac{4\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})^2} \leq \frac{1}{6}(1-\alpha_{\mathsf{in}})(1-\alpha_{\mathsf{out}}).
$$

Combined with the trivial fact that $f(-1) > 0$ and $f(\alpha_{\mathsf{in}}) \leq 0$, all eigenvalues of $\boldsymbol{G}_{\mathsf{out}}$ are in $(-1,1)$. Consequently, the Neumann series converges, leading to

$$
\sum_{t=0}^{\infty} \boldsymbol{G}_{\mathsf{out}}^t = (\boldsymbol{I}_2 - \boldsymbol{G}_{\mathsf{out}})^{-1} = \begin{bmatrix} \frac{(1-\alpha_{\mathsf{in}})^2(1-\alpha_{\mathsf{out}})^2 - 16\alpha_{\mathsf{in}}^2\alpha_{\mathsf{out}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})^3(1-\alpha_{\mathsf{out}})^2 - 32\alpha_{\mathsf{in}}^2\alpha_{\mathsf{out}}^2\eta^2L^2} & \frac{4\alpha_{\mathsf{in}}^2(1-\alpha_{\mathsf{out}})\eta^2L^2}{(1-\alpha_{\mathsf{in}})^3(1-\alpha_{\mathsf{out}})^2 - 32\alpha_{\mathsf{in}}^2\alpha_{\mathsf{out}}^2\eta^2L^2} \\ \frac{4(1-\alpha_{\mathsf{in}})^2(1+\alpha_{\mathsf{in}})\alpha_{\mathsf{out}}^2}{(1-\alpha_{\mathsf{in}})^3(1-\alpha_{\mathsf{out}})^2 - 32\alpha_{\mathsf{in}}^2\alpha_{\mathsf{out}}^2\eta^2L^2} & \frac{(1-\alpha_{\mathsf{in}})^3(1-\alpha_{\mathsf{out}})}{(1-\alpha_{\mathsf{in}})^3(1-\alpha_{\mathsf{out}})^2 - 32\alpha_{\mathsf{in}}^2\alpha_{\mathsf{out}}^2\eta^2L^2} \end{bmatrix}
$$

$$\leq \begin{bmatrix} \frac{2}{1-\alpha_{\mathsf{in}}} & \frac{8\alpha_{\mathsf{in}}^2\eta^2L^2}{(1-\alpha_{\mathsf{in}})^3(1-\alpha_{\mathsf{out}})} \\ \frac{16\alpha_{\mathsf{out}}^2}{(1-\alpha_{\mathsf{in}})(1-\alpha_{\mathsf{out}})^2} & \frac{2}{1-\alpha_{\mathsf{out}}} \end{bmatrix},$$

where we use the condition in (3.3) to prove $32\alpha_{\mathsf{in}}^2\alpha_{\mathsf{out}}^2\eta^2L^2 \leq \frac{32}{100}(1-\alpha_{\mathsf{in}})^6(1-\alpha_{\mathsf{out}})^2 < \frac{1}{2}(1-\alpha_{\mathsf{in}})^3(1-\alpha_{\mathsf{out}})^2$ to bound the denominator.

□

Appendix C

Appendix for Chapter 4

C.1 Technical lemmas

We first recall some classical inequalities that helps our derivation.

Proposition 1. *Let $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_\tau\}$ be a set of τ vectors in $\mathbb{R}^d$. Then, $\forall \beta > 0$, we have*

$$\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle \leq \frac{\beta}{2}\|\boldsymbol{u}\|^2 + \frac{1}{2\beta}\|\boldsymbol{v}\|^2, \tag{C.1}$$

$$\|\boldsymbol{v}_i + \boldsymbol{v}_j\|^2 \leq (1+\beta)\|\boldsymbol{v}_i\|^2 + \left(1 + \frac{1}{\beta}\right)\|\boldsymbol{v}_j\|^2, \tag{C.2}$$

$$\left\|\sum_{i=1}^{\tau} \boldsymbol{v}_i\right\|^2 \leq \tau \sum_{i=1}^{\tau} \|\boldsymbol{v}_i\|^2. \tag{C.3}$$

Here, (C.1) is referred as the Cauchy-Schwarz inequality, (C.2) and (C.3) are referred as Young's inequality.

Additional notation The following notation will be used throughout our proof:

$$\nabla F(\overline{\boldsymbol{x}}) := [\nabla f_1(\overline{\boldsymbol{x}}), \nabla f_2(\overline{\boldsymbol{x}}), \ldots, \nabla f_n(\overline{\boldsymbol{x}})], \qquad \widetilde{\nabla}_b F(\overline{\boldsymbol{x}}) := [\widetilde{\nabla}_b f_1(\overline{\boldsymbol{x}}), \widetilde{\nabla}_b f_2(\overline{\boldsymbol{x}}), \ldots, \widetilde{\nabla}_b f_n(\overline{\boldsymbol{x}})].$$

Properties of the mixing matrix We make note of several useful properties of the mixing matrix in the following lemma.

Lemma 7. *Let $\boldsymbol{W}$ be a mixing matrix satisfying Definition 1. For any matrix $\boldsymbol{M} \in \mathbb{R}^{d\times n}$ and $\overline{\boldsymbol{m}} = \frac{1}{n}\boldsymbol{M}\boldsymbol{1}_n$, we have*

$$\left\|\boldsymbol{M}\boldsymbol{W} - \overline{\boldsymbol{m}}\boldsymbol{1}_n^\top\right\|_{\mathsf{F}}^2 = \left\|\boldsymbol{M}\boldsymbol{W} - \overline{\boldsymbol{m}}\boldsymbol{1}_n^\top\boldsymbol{W}\right\|_{\mathsf{F}}^2 \leq \alpha \left\|\boldsymbol{M} - \overline{\boldsymbol{m}}\boldsymbol{1}_n^\top\right\|_{\mathsf{F}}^2. \tag{C.4}$$

In addition, for any $\gamma \in (0,1]$, the matrix $\widetilde{\boldsymbol{W}} = \boldsymbol{I} + \gamma(\boldsymbol{W} - \boldsymbol{I})$ satisfies Definition 1 with a spectral gap at least $\gamma(1-\alpha)$.

Proof. The first claim follows from the spectral decomposition of $\boldsymbol{W}$. Since $\boldsymbol{W}$ is a doubly stochastic matrix, the largest absolute eigenvalue of $\boldsymbol{W}$ is 1 and the corresponding eigenvector is $\mathbf{1}_n$. Let $v_2, \ldots, v_n$ be the eigenvectors of $\boldsymbol{W}$ corresponding to the remaining eigenvalues. Then, we have

$$\left\|\boldsymbol{M}\boldsymbol{W} - \overline{\boldsymbol{m}}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 = \left\|\boldsymbol{M}\boldsymbol{W} - \overline{\boldsymbol{m}}\mathbf{1}_n^\top\boldsymbol{W}\right\|_{\mathsf{F}}^2 = \sum_{i=1}^{r}\|\boldsymbol{W}(\boldsymbol{m}_i - \overline{\boldsymbol{m}}_i\mathbf{1}_n)\|_2^2,$$

where the first equality follows from $\mathbf{1}_n^\top\boldsymbol{W} = \mathbf{1}_n^\top$, $\boldsymbol{m}_i$ denotes the transpose of i-th row of matrix $\boldsymbol{M}$, and $\overline{\boldsymbol{m}}_i$ denotes the average of $\boldsymbol{m}_i$. Now we decompose $\boldsymbol{m}_i - \overline{\boldsymbol{m}}_i\mathbf{1}_n$ using the eigenvectors of $\boldsymbol{W}$. Noting that

$$\mathbf{1}_n^\top(\boldsymbol{m}_i - \overline{\boldsymbol{m}}_i\mathbf{1}_n) = \mathbf{1}_n^\top\boldsymbol{m}_i - \mathbf{1}_n^\top\mathbf{1}_n\frac{1}{n}\mathbf{1}_n^\top\boldsymbol{m}_i = 0,$$

and thus we can write

$$\boldsymbol{m}_i - \overline{\boldsymbol{m}}_i\mathbf{1}_n = \sum_{j=2}^{n} c_j\boldsymbol{v}_j$$

for some $\{c_j\}_{j=2}^n$. Then, we have

$$\|\boldsymbol{W}(\boldsymbol{m}_i - \overline{\boldsymbol{m}}_i\mathbf{1}_n)\|_2^2 = \|\boldsymbol{W}\sum_{j=2}^{n} c_j\boldsymbol{v}_j\|_2^2 \le (1-(1-\alpha))^2\sum_{j=2}^{n} c_j^2 \le (1-(1-\alpha))\sum_{j=2}^{n} c_j^2 = (1-(1-\alpha))\|\boldsymbol{m}_i - \overline{\boldsymbol{m}}_i\mathbf{1}_n\|_2^2,$$

and we conclude the proof of this claim.

For the second claim, recall the fact that if $\boldsymbol{v}$ is an eigenvector of $\boldsymbol{W}$ corresponding to the eigenvalue λ, then $\boldsymbol{v}$ is also an eigenvector of $\widetilde{\boldsymbol{W}}$ with the corresponding eigenvalue $(1-\gamma)+\gamma\lambda$. This claim follows from simple computation based on this relation.

□

A key consequence of gradient tracking Before diving in the proofs of the main theorems, we record a key property of gradient tracking. Specifically, we have the following lemma.

Lemma 8. *If* $\overline{\boldsymbol{v}}^{(0)} = \frac{1}{n}\widetilde{\nabla}_b F(\boldsymbol{X}^{(0)})\mathbf{1}_n$, *then for any* $t \ge 1$, *we have*

$$\overline{\boldsymbol{v}}^{(t)} = \frac{1}{n}\widetilde{\nabla}_b F(\boldsymbol{X}^{(t)})\mathbf{1}_n, \tag{C.5}$$

and

$$\overline{\boldsymbol{x}}^{(t+1)} = \overline{\boldsymbol{x}}^{(t)} - \frac{\eta}{n}\widetilde{\nabla}_b F(\boldsymbol{X}^{(t)})\mathbf{1}_n. \tag{C.6}$$

Proof. We first prove (C.5) by induction. For the base case ($t = 0$), the relation (C.5) is obviously true by the means of initialization. Now suppose that at the t-th iteration, the relation (C.5) is true, i.e.,

$$\overline{\boldsymbol{v}}^{(t)} = \frac{1}{n}\widetilde{\nabla}_b F(\boldsymbol{X}^{(t)})\mathbf{1}_n,$$

then at the $(t+1)$-th iteration, we have

$$\begin{aligned}\overline{\boldsymbol{v}}^{(t+1)} &= \frac{1}{n}\boldsymbol{V}^{(t+1)}\mathbf{1}_n \\ &= \frac{1}{n}\boldsymbol{V}^{(t)}\mathbf{1}_n + \frac{1}{n}\gamma\boldsymbol{G}^{(t)}(\boldsymbol{W}-\boldsymbol{I})\mathbf{1}_n + \frac{1}{n}\left(\widetilde{\nabla}_b F(\boldsymbol{X}^{(t+1)}) - \widetilde{\nabla}_b F(\boldsymbol{X}^{(t)})\right)\mathbf{1}_n \qquad \text{(C.7)} \\ &= \frac{1}{n}\boldsymbol{V}^{(t)}\mathbf{1}_n + \frac{1}{n}\left(\widetilde{\nabla}_b F(\boldsymbol{X}^{(t+1)}) - \widetilde{\nabla}_b F(\boldsymbol{X}^{(t)})\right)\mathbf{1}_n \\ &= \frac{1}{n}\widetilde{\nabla}_b F(\boldsymbol{X}^{(t+1)})\mathbf{1}_n.\end{aligned}$$

where (C.7) follows from the update rule of BEER (cf. Line 6), the penultimate line follows from $\boldsymbol{W}\mathbf{1}_n = \mathbf{1}_n$, and the last line follows from the induction hypothesis at the t-th iteration. Thus the induction hypothesis is also true at the $(t+1)$-th iteration, and we complete the proof of (C.5).

For (C.6), it follows from the update rule of BEER (cf. Line 3) that

$$\begin{aligned}\overline{\boldsymbol{x}}^{(t+1)} &= \overline{\boldsymbol{x}}^{(t)} + \frac{\gamma}{n}\boldsymbol{H}^{(t)}(\boldsymbol{W}-\boldsymbol{I})\mathbf{1} - \frac{\eta}{n}\boldsymbol{V}^{(t)}\mathbf{1} \\ &= \overline{\boldsymbol{x}}^{(t)} - \eta\overline{\boldsymbol{v}}^{(t)} = \overline{\boldsymbol{x}}^{(t)} - \frac{\eta}{n}\widetilde{\nabla}_b F(\boldsymbol{X}^{(t)})\mathbf{1}_n,\end{aligned}$$

where the second line uses $\boldsymbol{W}\mathbf{1}_n = \mathbf{1}_n$ and (C.5). □

C.2 Recursive relations of main errors

For convenience, we repeat the definitions in (4.2) below.

$$\begin{aligned}&\text{(compression approximation error:)} && \Omega_1^{(t)} = \mathbb{E}\left\|\boldsymbol{H}^{(t)} - \boldsymbol{X}^{(t)}\right\|_{\mathsf{F}}^2, && \Omega_2^{(t)} = \mathbb{E}\left\|\boldsymbol{G}^{(t)} - \boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2, \\ &\text{(consensus error:)} && \Omega_3^{(t)} = \mathbb{E}\left\|\boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2, && \Omega_4^{(t)} = \mathbb{E}\left\|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2, \\ &\text{(gradient norm:)} && \Omega_5^{(t)} = \mathbb{E}\|\overline{\boldsymbol{v}}^{(t)}\|_2^2.\end{aligned}$$

Lemma 9 creates induction inequalities for $\{\Omega_i^{(t)}\}$ that captures the approximation errors induced by compression and the consensus errors due to the decentralized setting. Then, we can show that the Lyapunov function defined in (4.1) descends, which leads to the claimed convergence results in Theorem 7 and Theorem 8.

Lemma 9. *Suppose Assumptions 5 and 7 hold, then for any $t \geq 0$, we have*

$$\Omega_1^{(t+1)} \leq \left(1 - \frac{\rho}{2} + \frac{6\gamma^2 C}{\rho}\right)\Omega_1^{(t)} + 0\cdot\Omega_2^{(t)} + \frac{6\gamma^2 C}{\rho}\Omega_3^{(t)} + \frac{6\eta^2}{\rho}\Omega_4^{(t)} + \frac{6n\eta^2}{\rho}\Omega_5^{(t)}, \qquad \text{(C.8a)}$$

$$\begin{aligned}\Omega_2^{(t+1)} \leq{}& \frac{18\gamma^2 C L^2}{\rho}\Omega_1^{(t)} + \left(1 - \frac{\rho}{2} + \frac{6\gamma^2 C}{\rho}\right)\Omega_2^{(t)} + \frac{18\gamma^2 C L^2}{\rho}\Omega_3^{(t)} \\ &+ \frac{6\gamma^2 C + 18L^2\eta^2}{\rho}\Omega_4^{(t)} + \frac{18L^2\eta^2 n}{\rho}\Omega_5^{(t)} + \frac{12n\sigma^2}{b\rho},\end{aligned} \qquad \text{(C.8b)}$$

$$\Omega_3^{(t+1)} \le \frac{6\gamma C}{(1-\alpha)}\Omega_1^{(t)} + 0\cdot\Omega_2^{(t)} + \left(1-\frac{\gamma(1-\alpha)}{2}\right)\Omega_3^{(t)} + \frac{6\eta^2}{\gamma(1-\alpha)}\Omega_4^{(t)} + 0\cdot\Omega_5^{(t)}, \tag{C.8c}$$

$$\Omega_4^{(t+1)} \le \frac{18\gamma CL^2}{(1-\alpha)}\Omega_1^{(t)} + \frac{6\gamma C}{(1-\alpha)}\Omega_2^{(t)} + \frac{18\gamma CL^2}{(1-\alpha)}\Omega_3^{(t)} + \left(1-\frac{\gamma(1-\alpha)}{2}+\frac{18L^2\eta^2}{\gamma(1-\alpha)}\right)\Omega_4^{(t)} + \frac{18n\eta^2L^2}{\gamma(1-\alpha)}\cdot\Omega_5^{(t)} + \frac{12n\sigma^2}{b\gamma(1-\alpha)}, \tag{C.8d}$$

where

$$C = \left\|\boldsymbol{W}-\boldsymbol{I}\right\|_{\mathsf{op}}^2 = \sigma_{\max}(\boldsymbol{W}-\boldsymbol{I})^2 \tag{C.9}$$

is the square of the maximum singular value of the matrix $\boldsymbol{W}-\boldsymbol{I}$.

Note that the eigenvalues of $\boldsymbol{W}$ and $\boldsymbol{I}$ all lies in $[-1,1]$, and thus clearly $C \le 4$.

Proof. We will establish the inequalities in (C.8) one by one.

Bounding $\Omega_1^{(t)}$ in (C.8a) First from the update rule of BEER (cf. Line 5), we have

$$\begin{aligned}\left\|\boldsymbol{H}^{(t+1)}-\boldsymbol{X}^{(t+1)}\right\|_{\mathsf{F}}^2 &= \left\|\boldsymbol{H}^{(t)}+\mathcal{C}(\boldsymbol{X}^{(t+1)}-\boldsymbol{H}^{(t)})-\boldsymbol{X}^{(t+1)}\right\|_{\mathsf{F}}^2\\ &\le (1-\rho)\left\|\boldsymbol{X}^{(t+1)}-\boldsymbol{H}^{(t)}\right\|_{\mathsf{F}}^2\\ &\le \left(1-\frac{\rho}{2}\right)\left\|\boldsymbol{X}^{(t)}-\boldsymbol{H}^{(t)}\right\|_{\mathsf{F}}^2 + \frac{2}{\rho}\left\|\boldsymbol{X}^{(t+1)}-\boldsymbol{X}^{(t)}\right\|_{\mathsf{F}}^2,\end{aligned} \tag{C.10}$$

where the first inequality comes from the definition of compression operators (Definition 5) and the second inequality comes from Young's inequality. It then boils down to bound $\left\|\boldsymbol{X}^{(t+1)}-\boldsymbol{X}^{(t)}\right\|_{\mathsf{F}}^2$, for which we have

$$\left\|\boldsymbol{X}^{(t+1)}-\boldsymbol{X}^{(t)}\right\|_{\mathsf{F}}^2 = \left\|\gamma\boldsymbol{H}^{(t)}(\boldsymbol{W}-\boldsymbol{I})-\eta\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2 \tag{C.11}$$

$$\begin{aligned}&= \left\|\gamma(\boldsymbol{H}^{(t)}-\boldsymbol{X}^{(t)})(\boldsymbol{W}-\boldsymbol{I})+\gamma(\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top)(\boldsymbol{W}-\boldsymbol{I})-\eta\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2\\ &\le 3\gamma^2C\left\|\boldsymbol{X}^{(t)}-\boldsymbol{H}^{(t)}\right\|_{\mathsf{F}} + 3\gamma^2C\left\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 + 3\eta^2\left\|\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2\\ &= 3\gamma^2C\left\|\boldsymbol{X}^{(t)}-\boldsymbol{H}^{(t)}\right\|_{\mathsf{F}} + 3\gamma^2C\left\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 + 3\eta^2\left\|\boldsymbol{V}^{(t)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 + 3\eta^2n\|\overline{\boldsymbol{v}}^{(t)}\|_2^2,\end{aligned} \tag{C.12}$$

where in the first line we use the update rule of BEER (cf. Line 3), in the second line we use the property of the mixing matrix $\mathbf{1}_n^\top\boldsymbol{W}=\mathbf{1}_n^\top$, and in the third line, we apply Young's inequality (cf. (C.3)). In the fourth line, we use $\|\boldsymbol{v}\|_2^2 = \|\boldsymbol{v}-\bar{v}\mathbf{1}_n\|_2^2 + n\bar{v}^2$ for any vector $\boldsymbol{v}$ with an average $\bar{v}$. Plugging this back into (C.10), we get

$$\begin{aligned}\left\|\boldsymbol{H}^{(t+1)}-\boldsymbol{X}^{(t+1)}\right\|_{\mathsf{F}}^2 \le \left(1-\frac{\rho}{2}+\frac{6\gamma^2C}{\rho}\right)\left\|\boldsymbol{X}^{(t)}-\boldsymbol{H}^{(t)}\right\|_{\mathsf{F}}^2 + \frac{6\gamma^2C}{\rho}\left\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2\\ + \frac{6\eta^2}{\rho}\left\|\boldsymbol{V}^{(t)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 + \frac{6n\eta^2}{\rho}\|\overline{\boldsymbol{v}}^{(t)}\|_2^2.\end{aligned}$$

Plugging in the definitions of $\Omega_i^{(t)}$, we obtain (C.8a).

Bounding $\Omega_2^{(t)}$ in (C.8b) Similar to the derivation of (C.8a), by applying the update rule of $\boldsymbol{G}^{(t)}$ in BEER (Line 8), and Young's inequality, we have

$$\begin{aligned}\left\|\boldsymbol{V}^{(t+1)}-\boldsymbol{G}^{(t+1)}\right\|_{\mathsf{F}}^2 &= \left\|\boldsymbol{G}^{(t)}+\mathcal{C}(\boldsymbol{V}^{(t+1)}-\boldsymbol{G}^{(t)})-\boldsymbol{V}^{(t+1)}\right\|_{\mathsf{F}}^2\\ &\le (1-\rho)\left\|\boldsymbol{G}^{(t)}-\boldsymbol{V}^{(t+1)}\right\|_{\mathsf{F}}^2\\ &\le \left(1-\frac{\rho}{2}\right)\left\|\boldsymbol{G}^{(t)}-\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2+\frac{2}{\rho}\left\|\boldsymbol{V}^{(t+1)}-\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2. \end{aligned} \tag{C.13}$$

It then boils down to bound $\left\|\boldsymbol{V}^{(t+1)}-\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2$. By the update rule of BEER (cf. Line 6), we have

$$\begin{aligned}\left\|\boldsymbol{V}^{(t+1)}-\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2 &= \left\|\gamma\boldsymbol{G}^{(t)}(\boldsymbol{W}-\boldsymbol{I})+(\widetilde{\nabla}_b F(\boldsymbol{X}^{(t+1)})-\widetilde{\nabla}_b F(\boldsymbol{X}^{(t)}))\right\|_{\mathsf{F}}^2\\ &= \left\|\gamma(\boldsymbol{G}^{(t)}-\boldsymbol{V}^{(t)})(\boldsymbol{W}-\boldsymbol{I})+\gamma(\boldsymbol{V}^{(t)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top)(\boldsymbol{W}-\boldsymbol{I})+(\widetilde{\nabla}_b F(\boldsymbol{X}^{(t+1)})-\widetilde{\nabla}_b F(\boldsymbol{X}^{(t)})\right\|_{\mathsf{F}}^2\\ &\overset{\text{(i)}}{\le} 3\gamma^2 C\left\|\boldsymbol{G}^{(t)}-\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2+3\gamma^2 C\left\|\boldsymbol{V}^{(t)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2+3\left\|\widetilde{\nabla}_b F(\boldsymbol{X}^{(t+1)})-\widetilde{\nabla}_b F(\boldsymbol{X}^{(t)})\right\|_{\mathsf{F}}^2\\ &\overset{\text{(ii)}}{\le} 3\gamma^2 C\left\|\boldsymbol{G}^{(t)}-\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2+3\gamma^2 C\left\|\boldsymbol{V}^{(t)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2+3\left\|\nabla F(\boldsymbol{X}^{(t+1)})-\nabla F(\boldsymbol{X}^{(t)})\right\|_{\mathsf{F}}^2+\frac{6n\sigma^2}{b}\\ &\overset{\text{(iii)}}{\le} 3\gamma^2 C\left\|\boldsymbol{G}^{(t)}-\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2+3\gamma^2 C\left\|\boldsymbol{V}^{(t)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2+3L^2\left\|\boldsymbol{X}^{(t+1)}-\boldsymbol{X}^{(t)}\right\|_{\mathsf{F}}^2+\frac{6n\sigma^2}{b}\\ &\overset{\text{(iv)}}{\le} 3\gamma^2 C\left\|\boldsymbol{G}^{(t)}-\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2+(3\gamma^2 C+9L^2\eta^2)\left\|\boldsymbol{V}^{(t)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2\\ &\quad+9\gamma^2 CL^2\left\|\boldsymbol{X}^{(t)}-\boldsymbol{H}^{(t)}\right\|_{\mathsf{F}}^2+9\gamma^2 CL^2\left\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2+9L^2\eta^2 n\|\overline{\boldsymbol{v}}^{(t)}\|_2^2+\frac{6n\sigma^2}{b}, \end{aligned}$$

where (i) comes from Young's inequality (cf. (C.3)) and basic facts of matrix norm (cf. (C.9)), (ii) comes from the bounded variance assumption (Assumption 7), (iii) comes from the smoothness assumption (Assumption 5), and (iv) follows from (C.12). Combining the above inequality with (C.13), we have

$$\begin{aligned}\left\|\boldsymbol{V}^{(t+1)}-\boldsymbol{G}^{(t+1)}\right\|_{\mathsf{F}}^2 &\le \left(1-\frac{\rho}{2}\right)\left\|\boldsymbol{G}^{(t)}-\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2+\frac{2}{\rho}\left\|\boldsymbol{V}^{(t+1)}-\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2\\ &\le \left(1-\frac{\rho}{2}+\frac{6\gamma^2 C}{\rho}\right)\left\|\boldsymbol{G}^{(t)}-\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2+\frac{6\gamma^2 C+18L^2\eta^2}{\rho}\left\|\boldsymbol{V}^{(t)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2\\ &\quad+\frac{18\gamma^2 CL^2}{\rho}\left\|\boldsymbol{X}^{(t)}-\boldsymbol{H}^{(t)}\right\|_{\mathsf{F}}^2+\frac{18\gamma^2 CL^2}{\rho}\left\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2+\frac{18L^2\eta^2 n}{\rho}\|\overline{\boldsymbol{v}}^{(t)}\|_2^2+\frac{12n\sigma^2}{b\rho}. \end{aligned}$$

Plugging in the definitions of $\Omega_i^{(t)}$, we obtain (C.8b).

Bounding $\Omega_3^{(t)}$ in (C.8c) To bound the consensus error $\left\|\boldsymbol{X}^{(t+1)}-\overline{\boldsymbol{x}}^{(t+1)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2$, by the update rule of BEER (cf. Line 3), we have

$$\begin{aligned}&\left\|\boldsymbol{X}^{(t+1)}-\overline{\boldsymbol{x}}^{(t+1)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2\\ &=\left\|\boldsymbol{X}^{(t)}+\gamma\boldsymbol{H}^{(t)}(\boldsymbol{W}-\boldsymbol{I})-\eta\boldsymbol{V}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top+\eta\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2\\ &\overset{\text{(i)}}{=}\left\|\boldsymbol{X}^{(t)}\widetilde{\boldsymbol{W}}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top+\gamma(\boldsymbol{H}^{(t)}-\boldsymbol{X}^{(t)})(\boldsymbol{W}-\boldsymbol{I})-\eta\boldsymbol{V}^{(t)}+\eta\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 \end{aligned}$$

$$
\begin{aligned}
&\overset{\text{(ii)}}{\leq} (1+\beta)(1-\gamma(1-\alpha))\left\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2+\left(1+\frac{1}{\beta}\right)\left(2\gamma^2\left\|(\boldsymbol{H}^{(t)}-\boldsymbol{X}^{(t)})(\boldsymbol{W}-\boldsymbol{I})\right\|_{\mathsf{F}}^2+2\eta^2\left\|\boldsymbol{V}^{(t)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2\right)\\
&\overset{\text{(iii)}}{\leq} \left(1-\frac{\gamma(1-\alpha)}{2}\right)\left\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2+\left(1+\frac{2}{\gamma(1-\alpha)}\right)\left(2\gamma^2\left\|(\boldsymbol{H}^{(t)}-\boldsymbol{X}^{(t)})(\boldsymbol{W}-\boldsymbol{I})\right\|_{\mathsf{F}}^2+2\eta^2\left\|\boldsymbol{V}^{(t)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2\right)\\
&\overset{\text{(iv)}}{\leq} \left(1-\frac{\gamma(1-\alpha)}{2}\right)\left\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2+\left(1+\frac{2}{\gamma(1-\alpha)}\right)\left(2\gamma^2C\left\|\boldsymbol{H}^{(t)}-\boldsymbol{X}^{(t)}\right\|_{\mathsf{F}}^2+2\eta^2\left\|\boldsymbol{V}^{(t)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2\right)\\
&\leq \left(1-\frac{\gamma(1-\alpha)}{2}\right)\left\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2+\frac{6\gamma C}{(1-\alpha)}\left\|\boldsymbol{H}^{(t)}-\boldsymbol{X}^{(t)}\right\|_{\mathsf{F}}^2+\frac{6\eta^2}{\gamma(1-\alpha)}\left\|\boldsymbol{V}^{(t)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2,
\end{aligned}
$$

where (i) follows from the definition $\widetilde{\boldsymbol{W}}=\boldsymbol{I}+\gamma(\boldsymbol{W}-\boldsymbol{I})$, (ii) follows from applying Young's inequality twice and Lemma 7, i.e.

$$
\left\|\boldsymbol{X}^{(t)}\widetilde{\boldsymbol{W}}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}\leq(1-\gamma(1-\alpha))\left\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2,
$$

(iii) follows by choosing $\beta=\gamma(1-\alpha)/2$, and (iv) uses the definition of C (cf. (C.9)). Plugging in the definitions of $\Omega_i^{(t)}$, we obtain (C.8c).

Bounding $\Omega_4^{(t)}$ in (C.8d) First, note that

$$
\begin{aligned}
\left\|\boldsymbol{V}^{(t+1)}-\overline{\boldsymbol{v}}^{(t+1)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2&=\left\|\boldsymbol{V}^{(t+1)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top+\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top-\overline{\boldsymbol{v}}^{(t+1)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2\\
&=\left\|\boldsymbol{V}^{(t+1)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2-n\|\overline{\boldsymbol{v}}^{(t+1)}-\overline{\boldsymbol{v}}^{(t)}\|_2^2\\
&\leq\left\|\boldsymbol{V}^{(t+1)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2.
\end{aligned}
$$

Thus by the update rule of BEER (cf. Line 6), we have

$$
\begin{aligned}
&\left\|\boldsymbol{V}^{(t+1)}-\overline{\boldsymbol{v}}^{(t+1)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2\\
&\leq\left\|\boldsymbol{V}^{(t+1)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2\\
&=\left\|\boldsymbol{V}^{(t)}+\gamma\boldsymbol{G}^{(t+1)}(\boldsymbol{W}-\boldsymbol{I})+\widetilde{\nabla}_bF(\boldsymbol{X}^{(t+1)})-\widetilde{\nabla}_bF(\boldsymbol{X}^{(t)})-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2\\
&=\left\|(\boldsymbol{V}^{(t)}\widetilde{\boldsymbol{W}}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top)+\gamma(\boldsymbol{G}^{(t+1)}-\boldsymbol{V}^{(t)})(\boldsymbol{W}-\boldsymbol{I})+(\widetilde{\nabla}_bF(\boldsymbol{X}^{(t+1)})-\widetilde{\nabla}_bF(\boldsymbol{X}^{(t)}))\right\|_{\mathsf{F}}^2\\
&\overset{\text{(i)}}{\leq}\left(1-\frac{\gamma(1-\alpha)}{2}\right)\left\|\boldsymbol{V}^{(t)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2+\left(1+\frac{2}{\gamma(1-\alpha)}\right)\left(2\gamma^2C\left\|\boldsymbol{G}^{(t)}-\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2+2L^2\left\|\boldsymbol{X}^{(t+1)}-\boldsymbol{X}^{(t)}\right\|_{\mathsf{F}}^2+\frac{4n\sigma^2}{b}\right)\\
&\overset{\text{(ii)}}{\leq}\left(1-\frac{\gamma(1-\alpha)}{2}\right)\left\|\boldsymbol{V}^{(t)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2+\frac{6\gamma C}{(1-\alpha)}\left\|\boldsymbol{G}^{(t)}-\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2+\frac{6L^2}{\gamma(1-\alpha)}\left\|\boldsymbol{X}^{(t+1)}-\boldsymbol{X}^{(t)}\right\|_{\mathsf{F}}^2+\frac{12n\sigma^2}{b\gamma(1-\alpha)}\\
&\leq\left(1-\frac{\gamma(1-\alpha)}{2}+\frac{18L^2\eta^2}{\gamma(1-\alpha)}\right)\left\|\boldsymbol{V}^{(t)}-\overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2+\frac{6\gamma C}{(1-\alpha)}\left\|\boldsymbol{G}^{(t)}-\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2\\
&\quad+\frac{18\gamma CL^2}{(1-\alpha)}\left\|\boldsymbol{X}^{(t)}-\boldsymbol{H}^{(t)}\right\|_{\mathsf{F}}^2+\frac{18\gamma CL^2}{(1-\alpha)}\left\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2+\frac{18n\eta^2L^2}{\gamma(1-\alpha)}\|\overline{\boldsymbol{v}}^{(t)}\|_2^2+\frac{12n\sigma^2}{b\gamma(1-\alpha)},
\end{aligned}
$$

where (i) and (ii) are obtained similarly as the derivation of (C.8c), and the last line follows from (C.12). Thus, we can get (C.8d) by plugging in the definitions of $\Omega_i^{(t)}$ and conclude the proof. □

C.3 Proof of Theorem 7

This proof makes use of Lemma 8 and Lemma 9 to construct a Lyapunov function and then demonstrates its descending property using a linear system argument.

Step 1: establishing a descent property of the function value First, we have the following inequality captures the "descent" of the function value.

$$\begin{aligned}
f(\overline{\boldsymbol{x}}^{(t+1)}) &\overset{\text{(i)}}{\leq} f(\overline{\boldsymbol{x}}^{(t)}) - \eta \left\langle \overline{\boldsymbol{v}}^{(t)}, \nabla f(\overline{\boldsymbol{x}}^{(t)}) \right\rangle + \frac{\eta^2 L}{2} \|\overline{\boldsymbol{v}}^{(t)}\|_2^2 \\
&= f(\overline{\boldsymbol{x}}^{(t)}) - \frac{\eta}{2} \|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2^2 - \frac{\eta}{2} \|\overline{\boldsymbol{v}}^{(t)}\|_2^2 + \frac{\eta}{2} \left\|\nabla f(\overline{\boldsymbol{x}}^{(t)}) - \overline{\boldsymbol{v}}^{(t)}\right\|_2^2 + \frac{\eta^2 L}{2} \|\overline{\boldsymbol{v}}^{(t)}\|_2^2 \\
&= f(\overline{\boldsymbol{x}}^{(t)}) - \frac{\eta}{2} \|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2^2 + \frac{\eta}{2} \left\|\nabla f(\overline{\boldsymbol{x}}^{(t)}) - \overline{\boldsymbol{v}}^{(t)}\right\|_2^2 - \left(\frac{\eta}{2} - \frac{\eta^2 L}{2}\right) \|\overline{\boldsymbol{v}}^{(t)}\|_2^2 \\
&\overset{\text{(ii)}}{\leq} f(\overline{\boldsymbol{x}}^{(t)}) - \frac{\eta}{2} \|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2^2 + \frac{\eta}{2n^2} \|\nabla F(\overline{\boldsymbol{x}}^{(t)}) \mathbf{1}_n - \widetilde{\nabla}_b F(\boldsymbol{X}^{(t)}) \mathbf{1}_n\|_2^2 - \left(\frac{\eta}{2} - \frac{\eta^2 L}{2}\right) \|\overline{\boldsymbol{v}}^{(t)}\|_2^2 \\
&= f(\overline{\boldsymbol{x}}^{(t)}) - \frac{\eta}{2} \|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2^2 + \frac{\eta}{2n^2} \|\nabla F(\overline{\boldsymbol{x}}^{(t)}) \mathbf{1}_n - \nabla F(\boldsymbol{X}^{(t)}) \mathbf{1}_n\|_2^2 - \left(\frac{\eta}{2} - \frac{\eta^2 L}{2}\right) \|\overline{\boldsymbol{v}}^{(t)}\|_2^2 \\
&\quad + \frac{\eta}{2n^2} \|\nabla F(\boldsymbol{X}^{(t)}) \mathbf{1}_n - \widetilde{\nabla}_b F(\boldsymbol{X}^{(t)}) \mathbf{1}_n\|_2^2 \\
&\quad + \frac{\eta}{n^2} \left\langle \nabla F(\boldsymbol{X}^{(t)}) \mathbf{1}_n - \widetilde{\nabla}_b F(\boldsymbol{X}^{(t)}) \mathbf{1}_n, \nabla F(\overline{\boldsymbol{x}}^{(t)}) \mathbf{1}_n - \nabla F(\boldsymbol{X}^{(t)}) \mathbf{1}_n \right\rangle,
\end{aligned}$$

where (i) comes from the L-smooth assumption (Assumption 5), (ii) comes from Lemma 8. Take expectation on both sides, and using the bounded variance assumption (Assumption 7) and independence of stochastic samples, we get

$$\begin{aligned}
\mathbb{E} f(\overline{\boldsymbol{x}}^{(t+1)}) &\leq \mathbb{E} f(\overline{\boldsymbol{x}}^{(t)}) - \frac{\eta}{2} \mathbb{E} \|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2^2 + \frac{\eta}{2n^2} \mathbb{E} \|\nabla F(\overline{\boldsymbol{x}}^{(t)}) \mathbf{1}_n - \nabla F(\boldsymbol{X}^{(t)}) \mathbf{1}_n\|_2^2 - \left(\frac{\eta}{2} - \frac{\eta^2 L}{2}\right) \mathbb{E} \|\overline{\boldsymbol{v}}^{(t)}\|_2^2 + \frac{\eta \sigma^2}{2bn} \\
&\overset{\text{(i)}}{\leq} \mathbb{E} f(\overline{\boldsymbol{x}}^{(t)}) - \frac{\eta}{2} \mathbb{E} \|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2^2 + \frac{\eta}{2n} \mathbb{E} \left\|\nabla F(\boldsymbol{X}^{(t)}) - \nabla F(\overline{\boldsymbol{x}}^{(t)})\right\|_{\mathsf{F}}^2 - \left(\frac{\eta}{2} - \frac{\eta^2 L}{2}\right) \mathbb{E} \|\overline{\boldsymbol{v}}^{(t)}\|_2^2 + \frac{\eta \sigma^2}{2bn} \\
&\overset{\text{(ii)}}{\leq} \mathbb{E} f(\overline{\boldsymbol{x}}^{(t)}) - \frac{\eta}{2} \mathbb{E} \|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2^2 + \frac{\eta L^2}{2n} \mathbb{E} \left\|\boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)} \mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 - \left(\frac{\eta}{2} - \frac{\eta^2 L}{2}\right) \mathbb{E} \|\overline{\boldsymbol{v}}^{(t)}\|_2^2 + \frac{\eta \sigma^2}{2bn},
\end{aligned}$$

where (i) comes from Young's inequality, and (ii) comes from the L-smooth assumption (Assumption 5) again. Finally, by substituting definitions of $\Omega_3^{(t)}$ and $\Omega_5^{(t)}$, we reach

$$\mathbb{E} f(\overline{\boldsymbol{x}}^{(t+1)}) \leq \mathbb{E} f(\overline{\boldsymbol{x}}^{(t)}) - \frac{\eta}{2} \mathbb{E} \|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2^2 + \frac{\eta L^2}{2n} \Omega_3^{(t)} - \left(\frac{\eta}{2} - \frac{\eta^2 L}{2}\right) \Omega_5^{(t)} + \frac{\eta \sigma^2}{2bn}. \tag{C.14}$$

Step 2: constructing the Lyapunov function By representing

$$\boldsymbol{\Omega}^{(t)} = [\Omega_1^{(t)} \quad \Omega_2^{(t)} \quad \Omega_3^{(t)} \quad \Omega_4^{(t)}]^\top, \tag{C.15}$$

Lemma 9 can be written more compactly as

$$\boldsymbol{\Omega}^{(t+1)} \leq \underbrace{\begin{bmatrix} 1-\frac{\rho}{2}+\frac{6\gamma^2 C}{\rho} & 0 & \frac{6\gamma^2 C}{\rho} & \frac{6\eta^2}{\rho} \\ \frac{18\gamma^2 CL^2}{\rho} & 1-\frac{\rho}{2}+\frac{6\gamma^2 C}{\rho} & \frac{18\gamma^2 CL^2}{\rho} & \frac{6\gamma^2 C+18L^2\eta^2}{\rho} \\ \frac{6\gamma C}{(1-\alpha)} & 0 & 1-\frac{\gamma(1-\alpha)}{2} & \frac{6\eta^2}{\gamma(1-\alpha)} \\ \frac{18\gamma CL^2}{(1-\alpha)} & \frac{6\gamma C}{(1-\alpha)} & \frac{18\gamma CL^2}{(1-\alpha)} & 1-\frac{\gamma(1-\alpha)}{2}+\frac{18L^2\eta^2}{\gamma(1-\alpha)} \end{bmatrix}}_{=:\boldsymbol{A}} \boldsymbol{\Omega}^{(t)} + \underbrace{\begin{bmatrix} \frac{6n\eta^2}{\rho} \\ \frac{18L^2\eta^2 n}{\rho} \\ 0 \\ \frac{18n\eta^2 L^2}{\gamma(1-\alpha)} \end{bmatrix}}_{=:\boldsymbol{b}_1} \Omega_5^{(t)} + \underbrace{\begin{bmatrix} 0 \\ \frac{12n}{\rho} \\ 0 \\ \frac{12n}{\gamma(1-\alpha)} \end{bmatrix}}_{=:\boldsymbol{b}_2} \frac{\sigma^2}{b}. \tag{C.16}$$

Define the Lyapunov function

$$\begin{aligned} \Phi^{(t)} &= \mathbb{E} f(\bar{\boldsymbol{x}}^{(t)}) - f^\star + \frac{c_1 L}{n} \cdot \Omega_1^{(t)} + \frac{c_2(1-\alpha)^2}{nL} \cdot \Omega_2^{(t)} + \frac{c_3 L}{n} \cdot \Omega_3^{(t)} + \frac{c_4(1-\alpha)^2}{nL} \Omega_4^{(t)} \\ &= \mathbb{E} f(\bar{\boldsymbol{x}}^{(t)}) - f^\star + \boldsymbol{s}^\top \boldsymbol{\Omega}^{(t)}, \end{aligned} \tag{C.17}$$

where

$$\boldsymbol{s} = \begin{bmatrix} \frac{c_1 L}{n} & \frac{c_2(1-\alpha)^2}{nL} & \frac{c_3 L}{n} & \frac{c_4(1-\alpha)^2}{nL} \end{bmatrix}$$

for some constants c_1, c_2, c_3, c_4 that will be specified later.

By (C.16) from Lemma 9 and the descent property (C.14), we have

$$\begin{aligned} \Phi^{(t+1)} &= \mathbb{E} f(\bar{\boldsymbol{x}}^{(t+1)}) - f^\star + \boldsymbol{s}^\top \boldsymbol{\Omega}^{(t+1)} \\ &\leq \mathbb{E} f(\bar{\boldsymbol{x}}^{(t)}) - f^\star - \frac{\eta}{2} \mathbb{E} \|\nabla f(\bar{\boldsymbol{x}}^{(t)})\|_2^2 + \frac{\eta L^2}{2n} \Omega_3^{(t)} - \left(\frac{\eta}{2} - \frac{\eta^2 L}{2} \right) \Omega_5^{(t)} + \frac{\eta \sigma^2}{2bn} \\ &\qquad + \boldsymbol{s}^\top \left(\boldsymbol{A} \boldsymbol{\Omega}^{(t)} + \Omega_5^{(t)} \boldsymbol{b}_1 + \frac{\sigma^2}{b} \boldsymbol{b}_2 \right) \\ &\leq \Phi^{(t)} - \frac{\eta}{2} \mathbb{E} \|\nabla f(\bar{\boldsymbol{x}}^{(t)})\|_2^2 - \left(\frac{\eta}{2} - \frac{\eta^2 L}{2} \right) \Omega_5^{(t)} + \frac{\eta \sigma^2}{2bn} + (\boldsymbol{s}^\top \boldsymbol{A} - \boldsymbol{s}^\top + \boldsymbol{q}^\top) \boldsymbol{\Omega}^{(t)} + \boldsymbol{s}^\top (\Omega_5^{(t)} \boldsymbol{b}_1 + \boldsymbol{b}_2 \frac{\sigma^2}{b}) \\ &= \Phi^{(t)} - \frac{\eta}{2} \mathbb{E} \|\nabla f(\bar{\boldsymbol{x}}^{(t)})\|_2^2 + (\boldsymbol{s}^\top \boldsymbol{A} - \boldsymbol{s}^\top + \boldsymbol{q}^\top) \boldsymbol{\Omega}^{(t)} - \left(\frac{\eta}{2} - \frac{\eta^2 L}{2} - \boldsymbol{s}^\top \boldsymbol{b}_1 \right) \Omega_5^{(t)} + \left(\frac{\eta}{2n} + \boldsymbol{s}^\top \boldsymbol{b}_2 \right) \frac{\sigma^2}{b}, \end{aligned} \tag{C.18}$$

where $\boldsymbol{q} = [0 \quad 0 \quad \frac{\eta L^2}{2n} \quad 0]^\top$. For a moment we assume that there exist some constants $c_1, c_2, c_3, c_4 > 0$ such that

$$\boldsymbol{s}^\top (\boldsymbol{A} - \boldsymbol{I}) + \boldsymbol{q}^\top \leq \boldsymbol{0}, \tag{C.19a}$$

$$\frac{\eta}{2} - \frac{\eta^2 L}{2} - \boldsymbol{s}^\top \boldsymbol{b}_1 \geq 0, \tag{C.19b}$$

leading to

$$\Phi^{(t+1)} \leq \Phi^{(t)} - \frac{\eta}{2} \mathbb{E} \|\nabla f(\bar{\boldsymbol{x}}^{(t)})\|_2^2 + \left(\frac{\eta}{2n} + \boldsymbol{s}^\top \boldsymbol{b}_2 \right) \frac{\sigma^2}{b} \leq \Phi^{(t)} - \frac{\eta}{2} \mathbb{E} \|\nabla f(\bar{\boldsymbol{x}}^{(t)})\|_2^2 + \frac{36 c_4 \sigma^2}{c_\gamma L b \rho}.$$

The proof is thus completed by recursing the above relation over $t = 0, \ldots, T-1$.

Step 3: verifying (C.19) It boils down to verify (C.19) is feasible, and it is equivalent to verify there exist parameters $c_1, c_2, c_3, c_4, \gamma, \eta > 0$ satisfying the following matrix inequality:

$$\begin{bmatrix} \boldsymbol{I} - \boldsymbol{A}^\top \\ -\boldsymbol{b}_1 \end{bmatrix} \operatorname{diag}\left[\frac{L}{n}, \frac{(1-\alpha)^2}{nL}, \frac{L}{n}, \frac{(1-\alpha)^2}{nL}\right] \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} \geq \begin{bmatrix} \boldsymbol{q} \\ \frac{\eta^2 L}{2} - \frac{\eta}{2} \end{bmatrix}.$$

Note that by choosing $\gamma = c_\gamma(1-\alpha)\rho, \eta = c_\eta \gamma (1-\alpha)^2/L$, and setting $c_\gamma \leq \frac{1}{6\sqrt{C}}$ and $c_\eta \leq \frac{1}{9}$, we get

$$1 - \frac{\rho}{2} + \frac{6\gamma^2 C}{\rho} \leq 1 - \frac{\rho}{4}, \quad 1 - \frac{\gamma(1-\alpha)}{2} + \frac{18L^2\eta^2}{\gamma(1-\alpha)} \leq 1 - \frac{\gamma(1-\alpha)}{4}. \tag{C.20}$$

Now, it suffices to show that there exist $c_1, c_2, c_3, c_4, c_\gamma, c_\eta > 0$ such that the following inequalities are satisfied:

$$\begin{bmatrix} \frac{\rho L}{4n} & -\frac{18c_\gamma^2\rho(1-\alpha)^4 L}{n} & -\frac{6c_\gamma \rho L}{n} & -\frac{18Cc_\gamma\rho(1-\alpha)^2 L}{n} \\ 0 & \frac{\rho(1-\alpha)^2}{4nL} & 0 & -\frac{6Cc_\gamma\rho(1-\alpha)^2}{nL} \\ -\frac{6c_\gamma(1-\alpha)\gamma CL}{n} & -\frac{18c_\gamma(1-\alpha)\gamma L}{n} & \frac{\gamma(1-\alpha)L}{2n} & -\frac{18C\gamma(1-\alpha)L}{n} \\ -\frac{6c_\eta^2 c_\gamma \gamma(1-\alpha)^3}{nL} & -\frac{6C\gamma(1-\alpha)^3(1+3c_\eta^2(1-\alpha)^4)}{nL} & -\frac{6c_\eta^2\gamma(1-\alpha)^3}{nL} & \frac{\gamma(1-\alpha)^3}{4nL} \\ -12c_\eta c_\gamma(1-\alpha)^3\frac{\eta}{2} & -36c_\eta c_\gamma(1-\alpha)^5\frac{\eta}{2} & 0 & -36c_\eta(1-\alpha)\frac{\eta}{2} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ \frac{c_\eta\gamma(1-\alpha)L}{2n} \\ 0 \\ \left(c_\eta c_\gamma \rho(1-\alpha)^3 - 1\right)\frac{\eta}{2} \end{bmatrix}.$$

Given $\rho \leq 1, (1-\alpha) \leq 1$, this can be further reduced to show the existence of $c_1, c_2, c_3, c_4, c_\gamma, c_\eta > 0$ such that

$$\begin{bmatrix} 1 & -72Cc_\gamma^2 & -24Cc_\gamma & -72Cc_\gamma \\ 0 & 1 & 0 & -24Cc_\gamma \\ -12Cc_\gamma & -35Cc_\gamma & 1 & -36C \\ -24c_\eta^2 c_\gamma & -24c_\gamma(1+3c_\eta^2) & -24c_\eta^2 & 1 \\ -12c_\eta c_\gamma & -36c_\eta c_\gamma & 0 & -36c_\eta \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ c_\eta \\ 0 \\ -1 + c_\eta c_\gamma \end{bmatrix}.$$

This can be easily verified by noting that as long as c_η and c_γ are set sufficiently small, it is straightforward to find feasible c_1, c_2, c_3, c_4.

C.4 Proof of Theorem 8

The proof strategy of Theorem 8 is similar to that of Theorem 7. However, we need to create a slightly different linear system to improve convergence under the new assumption.

Denote $\kappa := L/\mu$. Taking the same Lyapunov function $\Phi^{(t)}$ in (C.17), by the same argument of Appendix C.3 up to (C.18), we have

$$\Phi^{(t+1)} \leq \mathbb{E} f(\overline{\boldsymbol{x}}^{(t)}) - f^\star - \frac{\eta}{2}\mathbb{E}\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2^2 + \frac{\eta L^2}{2n}\Omega_3^{(t)} - \left(\frac{\eta}{2} - \frac{\eta^2 L}{2}\right)\Omega_5^{(t)} + \frac{\eta\sigma^2}{2bn} + \boldsymbol{s}^\top\left(\boldsymbol{A}\boldsymbol{\Omega}^{(t)} + \Omega_5^{(t)}\boldsymbol{b}_1 + \frac{\sigma^2}{b}\boldsymbol{b}_2\right)$$

$$\leq (1-\eta\mu)(\mathbb{E}f(\bar{\boldsymbol{x}}^{(t)})-f^\star)+(\boldsymbol{s}^\top\boldsymbol{A}+\boldsymbol{q}^\top)\boldsymbol{\Omega}^{(t)}-\left(\frac{\eta}{2}-\frac{\eta^2L}{2}-\boldsymbol{s}^\top\boldsymbol{b}_1\right)\Omega_5^{(t)}+\left(\frac{\eta}{2n}+\boldsymbol{s}^\top\boldsymbol{b}_2\right)\frac{\sigma^2}{b}$$
$$=(1-\eta\mu)\Phi^{(t)}+\left(\boldsymbol{s}^\top\boldsymbol{A}-(1-\eta\mu)\boldsymbol{s}^\top+\boldsymbol{q}^\top\right)\boldsymbol{\Omega}^{(t)}-\left(\frac{\eta}{2}-\frac{\eta^2L}{2}-\boldsymbol{s}^\top\boldsymbol{b}_1\right)\Omega_5^{(t)}+\left(\frac{\eta}{2n}+\boldsymbol{s}^\top\boldsymbol{b}_2\right)\frac{\sigma^2}{b},$$

where $\boldsymbol{q}=[0\quad 0\quad \frac{\eta L^2}{2n}\quad 0]^\top$, and the second inequality follows from the PL condition (Assumption 6). If we can establish that there exist there exist some constants c_1,c_2,c_3,c_4 such that

$$\boldsymbol{s}^\top(\boldsymbol{A}-(1-\eta\mu)\boldsymbol{I})+\boldsymbol{q}^\top\leq\boldsymbol{0}, \tag{C.21a}$$
$$\frac{\eta}{2}-\frac{\eta^2L}{2}-\boldsymbol{s}^\top\boldsymbol{b}_1\geq 0, \tag{C.21b}$$

we arrive at

$$\Phi^{(t+1)}\leq(1-\eta\mu)\Phi^{(t)}+\left(\frac{\eta}{2n}+\boldsymbol{s}^\top\boldsymbol{b}_2\right)\frac{\sigma^2}{b}\leq(1-\eta\mu)\Phi^{(t)}+\frac{36c_4\sigma^2}{c_\gamma Lb\rho}.$$

Recursing the above relation then complete the proof.

It then boils down to establish (C.21). By similar arguments as Appendix C.3, in view of (C.20) and $\rho\leq 1,(1-\alpha)\leq 1$, it is sufficient to show there exist constants $c_1,c_2,c_3,c_4,c_\gamma,c_\eta>0$ such that

$$\begin{bmatrix}1-\frac{4c_\eta c_\gamma}{\kappa} & -72Cc_\gamma^2 & -24Cc_\gamma & -72Cc_\gamma\\ 0 & 1-\frac{4c_\eta c_\gamma}{\kappa} & 0 & -24Cc_\gamma\\ -12Cc_\gamma & -35Cc_\gamma & 1-\frac{2c_\eta}{\kappa} & -36C\\ -24c_\eta^2c_\gamma & -24c_\gamma(1+3c_\eta^2) & -24c_\eta^2 & 1-\frac{4c_\eta}{\kappa}\\ -12c_\eta c_\gamma & -36c_\eta c_\gamma & 0 & -36c_\eta\end{bmatrix}\begin{bmatrix}c_1\\c_2\\c_3\\c_4\end{bmatrix}\geq\begin{bmatrix}0\\0\\c_\eta\\0\\-1+c_\eta c_\gamma\end{bmatrix}.$$

This can be easily verified by noting that as long as c_η and c_γ are set sufficiently small, it is straightforward to find feasible c_1,c_2,c_3,c_4.

Appendix D

Appendix for Chapter 5

D.1 Proof of Theorem 9

This section proves Theorem 9 in the following steps: 1) define privacy loss and moment generating function, 2) define mechanisms and sub-mechanisms, 3) bound overall moment generating function and show the choice of perturbation variance satisfies all conditions.

Moment generating function Let o and aux denote an outcome and an auxiliary input, respectively. Then, we can define the privacy loss of an outcome o on neighboring dataset $\mathbf{Z}$ and $\mathbf{Z}_i$ as

$$c(o; \mathcal{M}, \text{aux}, \mathbf{Z}, \mathbf{Z}_i) = \log \frac{\mathbb{P}\big(\mathcal{M}(\text{aux}, \mathbf{Z}) = o\big)}{\mathbb{P}\big(\mathcal{M}(\text{aux}, \mathbf{Z}_i) = o\big)},$$

and its log moment generating functions as

$$\alpha_i^{\mathcal{M}}(\lambda; \text{aux}, \mathbf{Z}, \mathbf{Z}_i) = \log \mathbb{E}_{o \sim \mathcal{M}(\text{aux}, \mathbf{Z})} \big[\exp\big(\lambda c(o; \mathcal{M}, \text{aux}, \mathbf{Z}, \mathbf{Z}_i)\big)\big].$$

Take maximum over conditions, the unconditioned log moment generating function is

$$\hat{\alpha}_i^{\mathcal{M}}(\lambda) = \max_{\text{aux}, \mathbf{Z}, \mathbf{Z}_i} \alpha_i^{\mathcal{M}}(\lambda; \text{aux}, \mathbf{Z}, \mathbf{Z}_i).$$

Sub-mechanisms Definition 7 defines the LDP mechanism, but it is not enough to model decentralized algorithms. To model the perturbation operation happens on agent i at time t, we define a sub-mechanism as $\mathcal{M}_i^{(t)} : \mathcal{D} \to \mathcal{R}$, where $i \in [n], t \in [T]$, which can be understood as the perturbation added on agent i at time t. In addition, we define another mechanism $\mathcal{C} : \mathcal{R} \to \mathcal{R}$ to model the compression operator and $\mathcal{C} \circ \mathcal{M}_i^{(t)}$ to represent the full update at an agent, and use $\mathcal{M}$ to represent the full algorithm.

Proof of LDP The overall log moment generating function for agent i can be bounded using [LZLC22, Lemma 2] as

$$\hat{\alpha}_i^{\mathcal{M}}(\lambda) \leq \sum_{t=1}^{T} \hat{\alpha}_i^{\mathcal{C}\circ\mathcal{M}_i^{(t)}}(\lambda) \leq \sum_{t=1}^{T} \hat{\alpha}_i^{\mathcal{M}_i^{(t)}}(\lambda).$$

Let $q = \frac{b}{m}$ denote the probability each data sample is chose. For agent i and $\lambda > 0$, assume $q \leq \frac{\tau}{16\sigma_p}$ and $\lambda \leq \frac{\sigma_p^2}{\tau^2}\log\frac{\tau}{q\sigma_p}$. We can apply [ACG+16, Lemma 3] to bound each $\hat{\alpha}_i^{\mathcal{M}_i^{(t)}}(\lambda)$ as

$$\hat{\alpha}^{\mathcal{M}_i^{(t)}}(\lambda) \leq \frac{q^2\lambda(\lambda+1)\tau^2}{(1-q)\sigma_p^2} + O\Big(\frac{q^3\lambda^3\tau^3}{\sigma_p^3}\Big) = O\Big(\frac{q^2\lambda^2\tau^2}{\sigma_p^2}\Big).$$

To conclude the proof, we can verify there exists some λ that satisfies the following inequalities when choosing $\sigma_p = \frac{\tau q\sqrt{T\log(1/\delta)}}{\epsilon}$ and $q = \frac{b}{m}$,

$$\Big(\frac{Tq\tau\lambda}{\sigma_p}\Big)^2 \leq \frac{\lambda\epsilon}{2},$$

$$\exp(-\lambda\epsilon/2) \leq \delta,$$

$$\lambda \leq \frac{\sigma_p^2}{\tau^2}\log\frac{\tau}{q\sigma_p}.$$

D.2 Proof of Theorem 10

This section proves Theorem 10 in the following 4 subsections: Appendix D.2.1 derives the descent inequality, Appendices D.2.2 and D.2.3 create two linear systems to bound the sum of consensus errors in the descent inequality, and finally Appendix D.2.4 specifies hyper parameters to obtain convergence rate.

To reuse this section's results in Appendix D.3, we assume Assumption 10 in deriving descent lemma and linear systems, and lift this assumption when computing convergence rate in Appendix D.2.4 using $\sigma_g \leq 2\tau$.

D.2.1 Function value descent

Using Taylor expansion, and taking expectation conditioned on time t,

$$\begin{aligned}
\mathbb{E}_t\big[f(\overline{\boldsymbol{x}}^{(t+1)}) - f(\overline{\boldsymbol{x}}^{(t)})\big] &\leq \mathbb{E}_t\langle\nabla f(\overline{\boldsymbol{x}}^{(t)}), -\eta\overline{\boldsymbol{v}}^{(t+1)}\rangle + \frac{L}{2}\mathbb{E}_t\big\|\eta\overline{\boldsymbol{v}}^{(t+1)}\big\|_2^2 \\
&= -\eta\langle\nabla f(\overline{\boldsymbol{x}}^{(t)}), \mathbb{E}_t[\overline{\boldsymbol{v}}^{(t+1)}]\rangle + \frac{\eta^2 L}{2}\mathbb{E}_t\big\|\overline{\boldsymbol{v}}^{(t+1)}\big\|_2^2 \\
&= -\eta\langle\nabla f(\overline{\boldsymbol{x}}^{(t)}), \mathbb{E}_t[\overline{\boldsymbol{g}}_\tau^{(t+1)} + \overline{\boldsymbol{e}}^{(t+1)}]\rangle + \frac{\eta^2 L}{2}\mathbb{E}_t\big\|\overline{\boldsymbol{v}}^{(t+1)}\big\|_2^2,
\end{aligned}$$

where the last equality is due to $\overline{\boldsymbol{v}}^{(t)} = \overline{\boldsymbol{g}}_p^{(t)}$ that can be proved by induction.

Because $\mathbb{E}_t[\boldsymbol{e}_i^{(t)}] = \boldsymbol{0}_d$ and stochastic gradients are unbiased,

$$\mathbb{E}_t\big[f(\overline{\boldsymbol{x}}^{(t+1)}) - f(\overline{\boldsymbol{x}}^{(t)})\big]$$

$$
\begin{aligned}
&= -\eta\langle\nabla f(\bar{\boldsymbol{x}}^{(t)}), \nabla F(\boldsymbol{X}^{(t)})(\tfrac{1}{n}\mathbf{1}_n)\rangle + \frac{\eta^2 L}{2}\mathbb{E}_t\big\|\bar{\boldsymbol{v}}^{(t+1)}\big\|_2^2 \\
&= \frac{\eta}{2}\Big(\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)}) - \nabla F(\boldsymbol{X}^{(t)})(\tfrac{1}{n}\mathbf{1}_n)\big\|_2^2 - \big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2^2 - \big\|\nabla F(\boldsymbol{X}^{(t)})(\tfrac{1}{n}\mathbf{1}_n)\big\|_2^2\Big) + \frac{\eta^2 L}{2}\mathbb{E}_t\big\|\bar{\boldsymbol{v}}^{(t+1)}\big\|_2^2 \\
&\le -\frac{\eta}{2}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2^2 + \frac{\eta L^2}{2n}\big\|\boldsymbol{X}^{(t)} - \bar{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}^2 + \frac{\eta^2 L}{2}\mathbb{E}_t\big\|\bar{\boldsymbol{v}}^{(t+1)}\big\|_2^2 - \frac{\eta}{2}\big\|\nabla F(\boldsymbol{X}^{(t)})(\tfrac{1}{n}\mathbf{1}_n)\big\|_2^2, \qquad \text{(D.1)}
\end{aligned}
$$

where the last inequality is due to Assumption 2.

Let $\Delta = \mathbb{E}\big[f(\bar{\boldsymbol{x}}^{(0)})\big] - f^\star$. Take full expectation and average (D.1) over $t = 1, \ldots, T$, the expected utility can be bounded by

$$
\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2^2 \le \frac{2\Delta}{\eta T} + \frac{1}{T}\cdot\frac{L^2}{n}\sum_{t=1}^{T}\mathbb{E}\big\|\boldsymbol{X}^{(t)} - \bar{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}^2 \\
+ \frac{1}{T}\cdot\eta L\sum_{t=1}^{T}\mathbb{E}\big\|\bar{\boldsymbol{v}}^{(t+1)}\big\|_2^2 - \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big\|\nabla F(\boldsymbol{X}^{(t)})(\tfrac{1}{n}\mathbf{1}_n)\big\|_2^2. \qquad \text{(D.2)}
\end{aligned}
$$

D.2.2 Sum of variable consensus errors

This subsection creates a linear system to bound $\sum_{t=1}^{T}\mathbb{E}\big\|\boldsymbol{X}^{(t)} - \bar{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}^2$ by $\sum_{t=1}^{T}\mathbb{E}\big\|\boldsymbol{V}^{(t)} - \bar{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}^2$ and $\sum_{t=1}^{T}\mathbb{E}\big\|\bar{\boldsymbol{v}}^{(t)}\big\|_2^2$. To simplify notations, let $\widehat{\boldsymbol{W}} = \boldsymbol{I}_n + \gamma(\boldsymbol{W} - \boldsymbol{I}_n)$, and denote the mixing rate of $\widehat{\boldsymbol{W}}$ by $\hat{\alpha} = \big\|\widehat{\boldsymbol{W}} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\big\|_{\mathrm{op}}$. Lemma 10 analyzes the mixing rate of the regularized mxing matrix.

Lemma 10 (Mixing rate of regularized mixing matrix)**.** *Assuming $0 < \gamma \le 1$. The mixing rate of $\widehat{\boldsymbol{W}}$ can be bounded as*

$$
\hat{\alpha} \le 1 + \gamma(\alpha - 1). \qquad \text{(D.3)}
$$

Proof of Lemma 10. Let $\lambda_1 = 1 > \lambda_2 \ge \ldots \ge \lambda_n > -1$ denote the eigenvalues of $\boldsymbol{W}$. Corresponding eigenvalues of $\widehat{\boldsymbol{W}}$ are $1 + \gamma(\lambda_i - 1)$, $i = 1, \ldots, n$.

The mixing rate of $\widehat{\boldsymbol{W}}$ is

$$
\begin{aligned}
\hat{\alpha} &= \max\big\{\big|1 + \gamma(\lambda_2 - 1)\big|, \big|1 + \gamma(\lambda_n - 1)\big|\big\} \\
&\le \max\big\{|1 - \gamma| + \gamma|\lambda_2|, |1 - \gamma| + \gamma|\lambda_n|\big\} \\
&= 1 + \gamma(\alpha - 1).
\end{aligned}
$$

□

Variable consensus error

Take expectation conditioned on time t, and use Young's inequality, the variable consensus error can be bounded as

$$
\mathbb{E}_t\big\|\boldsymbol{X}^{(t+1)} - \bar{\boldsymbol{x}}^{(t+1)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}^2
$$

$$
\begin{aligned}
&= \mathbb{E}_t \left\| \left(\boldsymbol{X}^{(t)} + \gamma \boldsymbol{Q}_x^{(t+1)} (\boldsymbol{W} - \boldsymbol{I}_n) - \eta \boldsymbol{V}^{(t+1)} \right) \left(\boldsymbol{I}_n - (\tfrac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top) \right) \right\|_{\mathsf{F}}^2 \\
&= \mathbb{E}_t \left\| (\boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)} \mathbf{1}_n^\top) (\widehat{\boldsymbol{W}} - (\tfrac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top)) + \gamma (\boldsymbol{Q}_x^{(t+1)} - \boldsymbol{X}^{(t)})(\boldsymbol{W} - \boldsymbol{I}_n) - \eta \boldsymbol{V}^{(t+1)} \left(\boldsymbol{I}_n - (\tfrac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top) \right) \right\|_{\mathsf{F}}^2 \\
&\le \frac{2}{1 + \hat{\alpha}^2} \left\| (\boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)} \mathbf{1}_n^\top) (\widehat{\boldsymbol{W}} - (\tfrac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top)) \right\|_{\mathsf{F}}^2 \\
&\quad + \frac{2}{1 - \hat{\alpha}^2} \mathbb{E}_t \left\| \gamma (\boldsymbol{Q}_x^{(t+1)} - \boldsymbol{X}^{(t)})(\boldsymbol{W} - \boldsymbol{I}_n) - \eta \boldsymbol{V}^{(t+1)} \left(\boldsymbol{I}_n - (\tfrac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top) \right) \right\|_{\mathsf{F}}^2 \\
&\le \frac{2}{1 + \hat{\alpha}^2} \left\| (\boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)} \mathbf{1}_n^\top) (\widehat{\boldsymbol{W}} - (\tfrac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top)) \right\|_{\mathsf{F}}^2 \\
&\quad + \frac{4}{1 - \hat{\alpha}^2} \mathbb{E}_t \left\| \gamma (\boldsymbol{Q}_x^{(t+1)} - \boldsymbol{X}^{(t)})(\boldsymbol{W} - \boldsymbol{I}_n) \right\|_{\mathsf{F}}^2 + \frac{4}{1 - \hat{\alpha}^2} \mathbb{E}_t \left\| \eta \boldsymbol{V}^{(t+1)} \left(\boldsymbol{I}_n - (\tfrac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top) \right) \right\|_{\mathsf{F}}^2 \\
&\overset{\text{(i)}}{\le} \frac{2\hat{\alpha}^2}{1 + \hat{\alpha}^2} \left\| \boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)} \mathbf{1}_n^\top \right\|_{\mathsf{F}}^2 + \frac{16\gamma^2}{1 - \hat{\alpha}^2} \mathbb{E}_t \left\| \boldsymbol{Q}_x^{(t+1)} - \boldsymbol{X}^{(t)} \right\|_{\mathsf{F}}^2 + \frac{4\eta^2}{1 - \hat{\alpha}^2} \mathbb{E}_t \left\| \boldsymbol{V}^{(t+1)} - \overline{\boldsymbol{v}}^{(t+1)} \mathbf{1}_n^\top \right\|_{\mathsf{F}}^2 \\
&\overset{\text{(ii)}}{\le} \hat{\alpha} \left\| \boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)} \mathbf{1}_n^\top \right\|_{\mathsf{F}}^2 + \frac{16(1 - \rho)\gamma^2}{1 - \hat{\alpha}} \left\| \boldsymbol{Q}_x^{(t)} - \boldsymbol{X}^{(t)} \right\|_{\mathsf{F}}^2 + \frac{4\eta^2}{1 - \hat{\alpha}} \mathbb{E}_t \left\| \boldsymbol{V}^{(t+1)} - \overline{\boldsymbol{v}}^{(t+1)} \mathbf{1}_n^\top \right\|_{\mathsf{F}}^2,
\end{aligned} \tag{D.4}
$$

where (i) is obtained by $\|\boldsymbol{W} - \boldsymbol{I}_n\|_{\text{op}} \le 2$, (ii) uses $2\hat{\alpha} \le 1 + \hat{\alpha}^2$, $1 - \hat{\alpha} \le 1 - \hat{\alpha}^2$ and Definition 5.

Variable quantization error

Assume γ satisfies the following inequality (which will be verified in Appendix D.2.4)

$$
\gamma^2 \le \frac{\rho^2}{96(1 - \rho)}. \tag{D.5}
$$

Take expectation conditioned on time t, the variable quantization error can be decomposed and bounded as

$$
\begin{aligned}
&\mathbb{E}_t \left\| \boldsymbol{Q}_x^{(t+1)} - \boldsymbol{X}^{(t+1)} \right\|_{\mathsf{F}}^2 \\
&= \mathbb{E}_t \left\| \boldsymbol{Q}_x^{(t)} + \mathcal{C}(\boldsymbol{X}^{(t)} - \boldsymbol{Q}_x^{(t)}) - \boldsymbol{X}^{(t+1)} \right\|_{\mathsf{F}}^2 \\
&= \mathbb{E}_t \left\| \mathcal{C}(\boldsymbol{X}^{(t)} - \boldsymbol{Q}_x^{(t)}) - (\boldsymbol{X}^{(t)} - \boldsymbol{Q}_x^{(t)}) - (\boldsymbol{X}^{(t+1)} - \boldsymbol{X}^{(t)}) \right\|_{\mathsf{F}}^2 \\
&\overset{\text{(i)}}{\le} \frac{2}{1 + (1 - \rho)} \mathbb{E}_t \left\| \mathcal{C}(\boldsymbol{X}^{(t)} - \boldsymbol{Q}_x^{(t)}) - (\boldsymbol{X}^{(t)} - \boldsymbol{Q}_x^{(t)}) \right\|_{\mathsf{F}}^2 + \frac{2}{1 - (1 - \rho)} \mathbb{E}_t \left\| \boldsymbol{X}^{(t+1)} - \boldsymbol{X}^{(t)} \right\|_{\mathsf{F}}^2 \\
&\overset{\text{(ii)}}{\le} \frac{2(1 - \rho)}{1 + (1 - \rho)} \left\| \boldsymbol{X}^{(t)} - \boldsymbol{Q}_x^{(t)} \right\|_{\mathsf{F}}^2 + \frac{2}{\rho} \mathbb{E}_t \left\| \boldsymbol{X}^{(t+1)} - \boldsymbol{X}^{(t)} \right\|_{\mathsf{F}}^2 \\
&= \frac{2(1 - \rho)}{1 + (1 - \rho)} \left\| \boldsymbol{X}^{(t)} - \boldsymbol{Q}_x^{(t)} \right\|_{\mathsf{F}}^2 \\
&\quad + \frac{2}{\rho} \mathbb{E}_t \left\| \gamma (\boldsymbol{Q}_x^{(t+1)} - \boldsymbol{X}^{(t)})(\boldsymbol{W} - \boldsymbol{I}_n) + \gamma (\boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)} \mathbf{1}_n^\top)(\boldsymbol{W} - \boldsymbol{I}_n) - \eta \boldsymbol{V}^{(t+1)} \right\|_{\mathsf{F}}^2 \\
&\overset{\text{(iii)}}{\le} \left(1 - \frac{\rho}{2} \right) \left\| \boldsymbol{X}^{(t)} - \boldsymbol{Q}_x^{(t)} \right\|_{\mathsf{F}}^2 + \frac{24\gamma^2}{\rho} \mathbb{E}_t \left\| \boldsymbol{Q}_x^{(t+1)} - \boldsymbol{X}^{(t)} \right\|_{\mathsf{F}}^2 + \frac{24\gamma^2}{\rho} \left\| \boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)} \mathbf{1}_n^\top \right\|_{\mathsf{F}}^2 + \frac{6\eta^2}{\rho} \mathbb{E}_t \left\| \boldsymbol{V}^{(t+1)} \right\|_{\mathsf{F}}^2 \\
&\le \left(1 - \frac{\rho}{2} + \frac{24(1 - \rho)\gamma^2}{\rho} \right) \left\| \boldsymbol{Q}_x^{(t)} - \boldsymbol{X}^{(t)} \right\|_{\mathsf{F}}^2 + \frac{24\gamma^2}{\rho} \left\| \boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)} \mathbf{1}_n^\top \right\|_{\mathsf{F}}^2 + \frac{6\eta^2}{\rho} \mathbb{E}_t \left\| \boldsymbol{V}^{(t+1)} \right\|_{\mathsf{F}}^2 \\
&\overset{\text{(iv)}}{\le} \left(1 - \frac{\rho}{4} \right) \left\| \boldsymbol{Q}_x^{(t)} - \boldsymbol{X}^{(t)} \right\|_{\mathsf{F}}^2 + \frac{24\gamma^2}{\rho} \left\| \boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)} \mathbf{1}_n^\top \right\|_{\mathsf{F}}^2 + \frac{6\eta^2}{\rho} \mathbb{E}_t \left\| \boldsymbol{V}^{(t+1)} \right\|_{\mathsf{F}}^2,
\end{aligned} \tag{D.6}
$$

where (i) is obtained by applying Young's inequality, (ii) uses Definition 5, (iii) uses the fact $\left\|\boldsymbol{W}-\boldsymbol{I}_n\right\|_{\text{op}} \leq 2$, and (iv) uses (D.5).

Linear system

Let $\boldsymbol{e}_1^{(t)} = \begin{bmatrix} \left\|\boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 \\ \left\|\boldsymbol{Q}_x^{(t)} - \boldsymbol{X}^{(t)}\right\|_{\mathsf{F}}^2 \end{bmatrix}$, we can take full expectation and rewrite (D.4) and (D.6) in matrix form as

$$\begin{aligned}\mathbb{E}[\boldsymbol{e}_1^{(t+1)}] &\leq \begin{bmatrix} \hat{\alpha} & \frac{16(1-\rho)\gamma^2}{1-\hat{\alpha}} \\ \frac{24\gamma^2}{\rho} & 1-\frac{\rho}{4} \end{bmatrix} \mathbb{E}[\boldsymbol{e}_1^{(t)}] + \begin{bmatrix} \frac{4\eta^2}{1-\hat{\alpha}}\mathbb{E}\left\|\boldsymbol{V}^{(t+1)} - \overline{\boldsymbol{v}}^{(t+1)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 \\ \frac{6\eta^2}{\rho}\mathbb{E}\left\|\boldsymbol{V}^{(t+1)}\right\|_{\mathsf{F}}^2 \end{bmatrix} \\ &:= \boldsymbol{G}_1\mathbb{E}[\boldsymbol{e}_1^{(t)}] + \boldsymbol{b}_1^{(t)}. \end{aligned} \tag{D.7}$$

We can compute $(\boldsymbol{I}_n - \boldsymbol{G}_1)^{-1}$ and verify all its entries are positive:

$$\begin{aligned}(\boldsymbol{I}_n - \boldsymbol{G}_1)^{-1} &= \frac{1}{(1-\hat{\alpha})\cdot\frac{\rho}{4} - \frac{16(1-\rho)\gamma^2}{1-\hat{\alpha}}\cdot\frac{24\gamma^2}{\rho}} \begin{bmatrix} \frac{\rho}{4} & \frac{16(1-\rho)\gamma^2}{1-\hat{\alpha}} \\ \frac{24\gamma^2}{\rho} & 1-\hat{\alpha} \end{bmatrix} \\ &\leq \frac{1}{\frac{1}{8}(1-\hat{\alpha})\rho} \begin{bmatrix} \frac{\rho}{4} & \frac{16(1-\rho)\gamma^2}{1-\hat{\alpha}} \\ \frac{24\gamma^2}{\rho} & 1-\hat{\alpha} \end{bmatrix}, \end{aligned} \tag{D.8}$$

where we assume the following inequality to to prove (D.8), which will be validated in Appendix D.2.4:

$$(1-\hat{\alpha})\cdot\frac{\rho}{4} - \frac{16(1-\rho)\gamma^2}{1-\hat{\alpha}}\cdot\frac{24\gamma^2}{\rho} \geq \frac{1}{8}(1-\hat{\alpha})\rho. \tag{D.9}$$

Sum expected error vectors $\mathbb{E}[\boldsymbol{e}_1^{(t)}]$ over $t = 1, \ldots, T$,

$$\begin{aligned}\sum_{t=1}^{T}\mathbb{E}[\boldsymbol{e}_1^{(t)}] &\leq \sum_{t=1}^{T}(\boldsymbol{G}_1\mathbb{E}[\boldsymbol{e}_1^{(t-1)}] + \boldsymbol{b}_1^{(t-1)}) \\ &\leq \boldsymbol{G}_1\sum_{t=1}^{T}\mathbb{E}[\boldsymbol{e}_1^{(t)}] + \boldsymbol{G}_1\mathbb{E}[\boldsymbol{e}_1^{(0)}] + \sum_{t=1}^{T}\boldsymbol{b}_1^{(t-1)}. \end{aligned}$$

Reorganize terms, multiply $(\boldsymbol{I}_n - \boldsymbol{G}_1)^{-1}$ on both sides and use $\boldsymbol{e}_1^{(0)} = \mathbf{0}_2$, the sum of error vectors can be bounded as

$$\sum_{t=1}^{T}\mathbb{E}[\boldsymbol{e}_1^{(t)}] \leq (\boldsymbol{I}_n - \boldsymbol{G}_1)^{-1}\sum_{t=0}^{T-1}\boldsymbol{b}_1^{(t)}. \tag{D.10}$$

The sum of consensus error can be computed as

$$\begin{aligned}&\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 \\ &\leq \begin{bmatrix} 1 & 0 \end{bmatrix}(\boldsymbol{I}_n - \boldsymbol{G})^{-1}\sum_{t=0}^{T-1}\boldsymbol{b}_1^{(t)} \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{\frac{1}{8}(1-\hat{\alpha})\rho}\begin{bmatrix}\frac{\rho}{4} & \frac{16(1-\rho)\gamma^2}{1-\hat{\alpha}}\end{bmatrix}\begin{bmatrix}\frac{4\eta^2}{1-\hat{\alpha}}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 \\ \frac{6\eta^2}{\rho}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2\end{bmatrix} \\
&= \frac{\eta^2}{\frac{1}{8}(1-\hat{\alpha})\rho}\Big(\frac{\rho}{1-\hat{\alpha}}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 + \frac{16(1-\rho)\gamma^2}{1-\hat{\alpha}}\cdot\frac{6}{\rho}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2\Big) \\
&\overset{\text{(i)}}{=} \frac{8\eta^2}{(1-\hat{\alpha})\rho}\Big(\frac{\rho}{1-\hat{\alpha}} + \frac{96(1-\rho)\gamma^2}{(1-\hat{\alpha})\rho}\Big)\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 + \frac{768(1-\rho)\gamma^2\eta^2}{(1-\hat{\alpha})^2\rho^2}\sum_{t=1}^{T}n\mathbb{E}\left\|\overline{\boldsymbol{v}}^{(t)}\right\|_{\mathsf{F}}^2 \\
&\overset{\text{(ii)}}{\leq} \frac{16\eta^2}{(1-\hat{\alpha})^2}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 + \frac{768(1-\rho)\gamma^2\eta^2}{(1-\hat{\alpha})^2\rho^2}\sum_{t=1}^{T}n\mathbb{E}\left\|\overline{\boldsymbol{v}}^{(t)}\right\|_{\mathsf{F}}^2, \qquad \text{(D.11)}
\end{aligned}
$$

where we use the equality $\left\|\boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2 = \left\|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 + n\|\overline{\boldsymbol{v}}^{(t)}\|_2^2$ for (i) and use (D.5) for (ii).

D.2.3 Sum of gradient consensus errors

This section creates a linear system to bound the sum of gradient consensus error $\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2$ by $\sum_{t=1}^{T}\mathbb{E}\left\|\overline{\boldsymbol{v}}^{(t)}\right\|_2^2$ and constant terms.

Gradient consensus error

Take expectation conditioned on time t and reorganize terms, the gradient consensus error can be expanded as

$$
\begin{aligned}
&\mathbb{E}_t\left\|\boldsymbol{V}^{(t+1)} - \overline{\boldsymbol{v}}^{(t+1)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 \\
&= \mathbb{E}_t\left\|\left(\boldsymbol{V}^{(t)} + \gamma\boldsymbol{Q}_v^{(t+1)}(\boldsymbol{W}-\boldsymbol{I}_n) + \boldsymbol{G}_p^{(t+1)} - \boldsymbol{G}_p^{(t)}\right)\left(\boldsymbol{I}_n - (\tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\right)\right\|_{\mathsf{F}}^2 \\
&= \mathbb{E}_t\left\|\left(\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right)\left(\widehat{\boldsymbol{W}} - (\tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\right) + \gamma\left(\boldsymbol{Q}_v^{(t+1)} - \boldsymbol{V}^{(t)}\right)(\boldsymbol{W}-\boldsymbol{I}_n) + \left(\boldsymbol{G}_p^{(t+1)} - \boldsymbol{G}_p^{(t)}\right)\left(\boldsymbol{I}_n - (\tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\right)\right\|_{\mathsf{F}}^2.
\end{aligned}
$$

Then, take full expectation, use the update formula and Young's inequality similarly to (D.4),

$$
\begin{aligned}
&\mathbb{E}\left\|\boldsymbol{V}^{(t+1)} - \overline{\boldsymbol{v}}^{(t+1)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 \\
&\leq \frac{2\hat{\alpha}^2}{1+\hat{\alpha}^2}\left\|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 \\
&+ \frac{2}{1-\hat{\alpha}^2}\mathbb{E}\left\|\gamma\left(\boldsymbol{Q}_v^{(t+1)} - \boldsymbol{V}^{(t)}\right)(\boldsymbol{W}-\boldsymbol{I}_n) + \left(\boldsymbol{G}_p^{(t+1)} - \boldsymbol{G}_p^{(t)}\right)\left(\boldsymbol{I}_n - (\tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\right)\right\|_{\mathsf{F}}^2 \\
&\leq \hat{\alpha}\left\|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 + \frac{4}{1-\hat{\alpha}}\mathbb{E}\left\|\gamma\left(\boldsymbol{Q}_v^{(t+1)} - \boldsymbol{V}^{(t)}\right)(\boldsymbol{W}-\boldsymbol{I}_n)\right\|_{\mathsf{F}}^2 \\
&\quad + \frac{4}{1-\hat{\alpha}}\mathbb{E}\left\|\left(\boldsymbol{G}_p^{(t+1)} - \boldsymbol{G}_p^{(t)}\right)\left(\boldsymbol{I}_n - (\tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\right)\right\|_{\mathsf{F}}^2 \\
&\overset{\text{(i)}}{\leq} \hat{\alpha}\left\|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 + \frac{16\gamma^2}{1-\hat{\alpha}}\mathbb{E}\left\|\boldsymbol{Q}_v^{(t+1)} - \boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2 + \frac{4}{1-\hat{\alpha}}\mathbb{E}\left\|\boldsymbol{G}_p^{(t+1)} - \boldsymbol{G}_p^{(t)}\right\|_{\mathsf{F}}^2 \\
&\overset{\text{(ii)}}{\leq} \hat{\alpha}\left\|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\right\|_{\mathsf{F}}^2 + \frac{16(1-\rho)\gamma^2}{1-\hat{\alpha}}\mathbb{E}\left\|\boldsymbol{Q}_v^{(t)} - \boldsymbol{V}^{(t)}\right\|_{\mathsf{F}}^2 + \frac{16n(\tau^2+\sigma_p^2 d)}{1-\hat{\alpha}}, \qquad \text{(D.12)}
\end{aligned}
$$

where (i) is proved using the facts $\|\boldsymbol{W}-\boldsymbol{I}_n\|_{\mathsf{op}} \leq 2$ and $\|\boldsymbol{I}_n - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\|_{\mathsf{op}} \leq 1$, (ii) is due to Definition 5 and

$$\begin{aligned}\mathbb{E}\|\boldsymbol{G}_p^{(t+1)} - \boldsymbol{G}_p^{(t)}\|_{\mathsf{F}}^2 &\leq 2\mathbb{E}\|\boldsymbol{G}_p^{(t+1)}\|_{\mathsf{F}}^2 + 2\mathbb{E}\|\boldsymbol{G}_p^{(t)}\|_{\mathsf{F}}^2 \\ &= 2\big(\mathbb{E}\|\boldsymbol{G}_\tau^{(t+1)}\|_{\mathsf{F}}^2 + n\sigma_p^2 d\big) + 2\big(\mathbb{E}\|\boldsymbol{G}_\tau^{(t)}\|_{\mathsf{F}}^2 + n\sigma_p^2 d\big) \\ &\leq 4n(\tau^2 + \sigma_p^2 d). \end{aligned} \tag{D.13}$$

Gradient quantization error

$$\begin{aligned}\mathbb{E}_t\|\boldsymbol{Q}_v^{(t+1)} - \boldsymbol{V}^{(t+1)}\|_{\mathsf{F}}^2 &= \mathbb{E}_t\|(\boldsymbol{Q}_v^{(t+1)} - \boldsymbol{V}^{(t)}) - (\boldsymbol{V}^{(t+1)} - \boldsymbol{V}^{(t)})\|_{\mathsf{F}}^2 \\ &\leq \frac{2}{1+(1-\rho)}\mathbb{E}_t\|\boldsymbol{Q}_v^{(t+1)} - \boldsymbol{V}^{(t)}\|_{\mathsf{F}}^2 + \frac{2}{1-(1-\rho)}\mathbb{E}_t\|\boldsymbol{V}^{(t+1)} - \boldsymbol{V}^{(t)}\|_{\mathsf{F}}^2 \\ &\leq \frac{2(1-\rho)}{2-\rho}\|\boldsymbol{Q}_v^{(t)} - \boldsymbol{V}^{(t)}\|_{\mathsf{F}}^2 + \frac{2}{\rho}\mathbb{E}_t\|\gamma\boldsymbol{Q}_v^{(t+1)}(\boldsymbol{W}-\boldsymbol{I}_n) + \boldsymbol{G}_p^{(t+1)} - \boldsymbol{G}_p^{(t)}\|_{\mathsf{F}}^2 \\ &\leq \frac{2(1-\rho)}{2-\rho}\|\boldsymbol{Q}_v^{(t)} - \boldsymbol{V}^{(t)}\|_{\mathsf{F}}^2 + \frac{6\gamma^2}{\rho}\mathbb{E}_t\|(\boldsymbol{Q}_v^{(t+1)} - \boldsymbol{V}^{(t)})(\boldsymbol{W}-\boldsymbol{I}_n)\|_{\mathsf{F}}^2 \\ &\quad + \frac{6\gamma^2}{\rho}\|\boldsymbol{V}^{(t)}(\boldsymbol{W}-\boldsymbol{I}_n)\|_{\mathsf{F}}^2 + \frac{6}{\rho}\mathbb{E}_t\|\boldsymbol{G}_p^{(t+1)} - \boldsymbol{G}_p^{(t)}\|_{\mathsf{F}}^2 \\ &\overset{\text{(i)}}{\leq} \Big(1 - \frac{\rho}{2} + \frac{24\gamma^2(1-\rho)}{\rho}\Big)\|\boldsymbol{Q}_v^{(t)} - \boldsymbol{V}^{(t)}\|_{\mathsf{F}}^2 + \frac{24\gamma^2}{\rho}\|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\|_{\mathsf{F}}^2 + \frac{24n(\tau^2+\sigma_p^2 d)}{\rho} \\ &\overset{\text{(ii)}}{\leq} \Big(1 - \frac{\rho}{4}\Big)\|\boldsymbol{Q}_v^{(t)} - \boldsymbol{V}^{(t)}\|_{\mathsf{F}}^2 + \frac{24\gamma^2}{\rho}\|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\|_{\mathsf{F}}^2 + \frac{24n(\tau^2+\sigma_p^2 d)}{\rho}, \end{aligned} \tag{D.14}$$

where we use (D.13) and the fact $\frac{2(1-\rho)}{2-\rho} = 1 - \frac{\rho}{2-\rho} \geq 1 - \frac{\rho}{2}$ when $\rho \geq 0$ to reach (i) and use (D.5) to reach (ii).

Linear system

Let $\boldsymbol{e}_2^{(t)} = \begin{bmatrix} \|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\|_{\mathsf{F}}^2 \\ \|\boldsymbol{Q}_v^{(t)} - \boldsymbol{V}^{(t)}\|_{\mathsf{F}}^2 \end{bmatrix}$. We can write (D.12) and (D.14) in matrix form as

$$\begin{aligned}\mathbb{E}[\boldsymbol{e}_2^{(t+1)}] &\leq \begin{bmatrix} \hat{\alpha} & \frac{16(1-\rho)\gamma^2}{1-\hat{\alpha}} \\ \frac{24\gamma^2}{\rho} & 1 - \frac{\rho}{4} \end{bmatrix}\mathbb{E}[\boldsymbol{e}_2^{(t)}] + \begin{bmatrix} \frac{16n(\tau^2+\sigma_p^2 d)}{1-\hat{\alpha}} \\ \frac{24n(\tau^2+\sigma_p^2 d)}{\rho} \end{bmatrix} \\ &:= \boldsymbol{G}_2\mathbb{E}[\boldsymbol{e}_2^{(t)}] + \boldsymbol{b}_2^{(t)}. \end{aligned}$$

Because $\boldsymbol{G}_2 = \boldsymbol{G}_1$, we can use the same argument as in Appendix D.2.2, and use (D.8) to prove

$$\begin{aligned}\sum_{t=1}^{T}\mathbb{E}\|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\|_{\mathsf{F}}^2 &\leq \begin{bmatrix} 1 & 0 \end{bmatrix}(\boldsymbol{I}_n - \boldsymbol{G}_2)^{-1}\Big(\mathbb{E}[\boldsymbol{e}_2^{(0)}] + \sum_{t=0}^{T-1}\boldsymbol{b}_2^{(t)}\Big) \\ &\leq \frac{1}{\frac{1}{8}(1-\hat{\alpha})\rho}\begin{bmatrix} \frac{\rho}{4} & \frac{16(1-\rho)\gamma^2}{1-\hat{\alpha}} \end{bmatrix}\begin{bmatrix} \frac{16Tn(\tau^2+\sigma_p^2 d)}{1-\hat{\alpha}} \\ \frac{24Tn(\tau^2+\sigma_p^2 d)}{\rho} \end{bmatrix} \end{aligned}$$

$$
\begin{aligned}
&= \frac{Tn(\tau^2+\sigma_p^2 d)}{\frac{1}{8}(1-\hat{\alpha})\rho} \cdot \Big(\frac{\rho}{4}\cdot\frac{16}{1-\hat{\alpha}} + \frac{16(1-\rho)\gamma^2}{1-\hat{\alpha}}\cdot\frac{24}{\rho}\Big) \\
&\le \frac{Tn(\tau^2+\sigma_p^2 d)}{\frac{1}{8}(1-\hat{\alpha})\rho} \cdot \Big(\frac{4\rho}{1-\hat{\alpha}} + \frac{4\rho}{1-\hat{\alpha}}\Big) \\
&= \frac{64}{(1-\hat{\alpha})^2} Tn(\tau^2+\sigma_p^2 d),
\end{aligned} \tag{D.15}
$$

where we use (D.5) to prove the last inequality.

With (D.15), we can bound (D.11) by

$$
\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}\big\|\boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}^2 \\
&\le \frac{16\eta^2}{(1-\hat{\alpha})^2}\sum_{t=1}^{T}\mathbb{E}\big\|\boldsymbol{V}^{(t)} - \overline{\boldsymbol{v}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}^2 + \frac{768(1-\rho)\gamma^2\eta^2}{(1-\hat{\alpha})^2\rho^2}\sum_{t=1}^{T} n\mathbb{E}\big\|\overline{\boldsymbol{v}}^{(t)}\big\|_{\mathsf{F}}^2 \\
&\le \frac{16\eta^2}{(1-\hat{\alpha})^2}\cdot\frac{64}{(1-\hat{\alpha})^2}Tn(\tau^2+\sigma_p^2 d) + \frac{768(1-\rho)\gamma^2\eta^2}{(1-\hat{\alpha})^2\rho^2}\sum_{t=1}^{T} n\mathbb{E}\big\|\overline{\boldsymbol{v}}^{(t)}\big\|_{\mathsf{F}}^2 \\
&\le \frac{1024\eta^2}{(1-\hat{\alpha})^4}Tn(\tau^2+\sigma_p^2 d) + \frac{8\eta^2}{(1-\hat{\alpha})^2}\sum_{t=1}^{T} n\mathbb{E}\big\|\overline{\boldsymbol{v}}^{(t)}\big\|_{\mathsf{F}}^2,
\end{aligned} \tag{D.16}
$$

where we use (D.5) again to prove the last inequality.

D.2.4 Convergence rate

Note bounded gradient assumption can imply Assumption 10 for some $\sigma_g \le 2\tau$, we can bound the expected norm of average gradient estimate as

$$
\begin{aligned}
\mathbb{E}\big\|\overline{\boldsymbol{v}}^{(t)}\big\|_2^2 &= \mathbb{E}\big\|\overline{\boldsymbol{g}}_p^{(t)}\big\|_2^2 \\
&= \mathbb{E}\big\|\overline{\boldsymbol{g}}_\tau^{(t)}\big\|_2^2 + \frac{\sigma_p^2 d}{n} \\
&\le \mathbb{E}\big\|\nabla F(\boldsymbol{X}^{(t)})(\tfrac{1}{n}\mathbf{1}_n)\big\|_2^2 + \frac{\sigma_g^2}{b} + \frac{\sigma_p^2 d}{n} \\
&\le \mathbb{E}\big\|\nabla F(\boldsymbol{X}^{(t)})(\tfrac{1}{n}\mathbf{1}_n)\big\|_2^2 + \frac{4\tau^2}{b} + \frac{\sigma_p^2 d}{n}.
\end{aligned} \tag{D.17}
$$

We assume

$$
\eta L \le \frac{1}{8}(1-\hat{\alpha})^{\frac{4}{3}}. \tag{D.18}
$$

Using (D.16) (D.17), expected utility (D.2) can be bounded by

$$
\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2^2 \le \frac{2\Delta}{\eta T} + \frac{1}{T}\cdot\frac{L^2}{n}\sum_{t=1}^{T}\mathbb{E}\big\|\boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}^2 \\
+ \frac{1}{T}\sum_{t=1}^{T}\eta L\mathbb{E}\|\overline{\boldsymbol{v}}^{(t)}\|_2^2 - \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big\|\nabla F(\boldsymbol{X}^{(t)})(\tfrac{1}{n}\mathbf{1}_n)\big\|_2^2
\end{aligned}
$$

$$
\begin{aligned}
&\leq \frac{2\Delta}{\eta T} + \frac{1}{T}\cdot\frac{L^2}{n}\Big(\frac{1024\eta^2}{(1-\hat{\alpha})^4}Tn(\tau^2+\sigma_p^2 d) + \frac{8\eta^2}{(1-\hat{\alpha})^2}\sum_{t=1}^{T} n\mathbb{E}\big\|\overline{\boldsymbol{v}}^{(t)}\big\|_2^2\Big) \\
&\quad + \frac{1}{T}\sum_{t=1}^{T}\eta L\mathbb{E}\|\overline{\boldsymbol{v}}^{(t)}\|_2^2 - \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big\|\nabla F(\boldsymbol{X}^{(t)})(\tfrac{1}{n}\mathbf{1}_n)\big\|_2^2 \\
&\overset{\text{(i)}}{\leq} \frac{2\Delta}{\eta T} + \frac{1024\eta^2L^2}{(1-\hat{\alpha})^4}(\tau^2+\sigma_p^2 d) + \frac{2\eta L}{(1-\hat{\alpha})^{\frac{4}{3}}T}\sum_{t=1}^{T}\mathbb{E}\|\overline{\boldsymbol{v}}^{(t)}\|_2^2 - \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big\|\nabla F(\boldsymbol{X}^{(t)})(\tfrac{1}{n}\mathbf{1}_n)\big\|_2^2 \\
&\leq \frac{2\Delta}{\eta T} + \frac{1024\eta^2L^2}{(1-\hat{\alpha})^4}(\tau^2+\sigma_p^2 d) + \frac{2\eta L}{(1-\hat{\alpha})^{\frac{4}{3}}}\Big(\frac{4\tau^2}{b} + \frac{\sigma_p^2 d}{n}\Big) \\
&\overset{\text{(ii)}}{=} \frac{2\Delta}{\eta T} + \frac{1024\eta^2L^2\tau^2}{(1-\hat{\alpha})^4}(1+T\phi_m^2) + \frac{8\eta L\tau^2}{(1-\hat{\alpha})^{\frac{4}{3}}}(1+T\phi_m^2) \\
&\overset{\text{(iii)}}{=} \frac{2\Delta}{\eta T} + \frac{2048\eta^2L^2\tau^2}{(1-\hat{\alpha})^4} + \frac{16\eta L\tau^2}{(1-\hat{\alpha})^{\frac{4}{3}}}
\end{aligned}
\tag{D.19}
$$

where we use (D.18) for (i), substitute $b = 1$ and $\sigma_p^2 d = \Big(\frac{\tau\sqrt{T\log(1/\delta)}}{m\epsilon}\Big)^2 = T\tau^2\phi_m^2$ for (ii), and substitute $T = \phi_m^{-2}$ for (iii).

We set the step size as

$$
\eta = \frac{\gamma^{\frac{4}{3}}(1-\alpha)^{\frac{4}{3}}}{32}\cdot\frac{\phi_m}{L},
$$

(D.19) can be further bounded as

$$
\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2^2 &\leq \frac{64L\Delta\phi_m}{\gamma^{\frac{4}{3}}(1-\alpha)^{\frac{4}{3}}} + \frac{2\tau^2\phi_m^2}{(1-\hat{\alpha})^{\frac{4}{3}}} + \frac{\tau^2\phi_m}{2} \\
&\leq \frac{64L\Delta\phi_m}{\gamma^{\frac{4}{3}}(1-\alpha)^{\frac{4}{3}}} + \frac{3\tau^2\phi_m}{(1-\hat{\alpha})^{\frac{4}{3}}} \\
&\leq \frac{67\phi_m}{\gamma^{\frac{4}{3}}(1-\alpha)^{\frac{4}{3}}}\max\{\tau^2, L\Delta\},
\end{aligned}
$$

where we use the assumption in (5.2) that $\phi_m < 1$ to prove the second inequality, and use Lemma 10 to reach the last inequality.

Lastly, set the hyper parameter γ as

$$
\gamma = \frac{1}{100}(1-\alpha)\rho.
$$

We can now verify conditions (D.5), (D.9) and the condition on η are all met to conclude the proof:

$$
\begin{aligned}
&\gamma^2 \leq \frac{\rho^2}{10000} && \Rightarrow && \text{(D.5)} \\
&\gamma^4 = \gamma^2\cdot\frac{(1-\alpha)^2\rho^2}{10000} \leq \frac{(1-\hat{\alpha})^2\rho^2}{10000} && \Rightarrow && \text{(D.9)} \\
&\eta L \leq \frac{(1-\hat{\alpha})^{\frac{4}{3}}}{32} && \Rightarrow && \text{(D.18)}
\end{aligned}
$$

D.3 Proof of Theorem 11

This section proves Theorem 11 in 2 subsections. Appendix D.3.1 derives the descent inequality using results from Appendices D.2.2 and D.2.3. Appendix D.3.2 first assumes all expected gradient norm $\mathbb{E}\left\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\right\|_2$ are greater than a threshold ν (i.e. $\mathbb{E}\left\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\right\|_2 \geq \nu$ for all $t = 1, \ldots, T$), then specifies parameters and proves the average of expected gradient norm is smaller than that threshold $\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\right\|_2 \leq \nu$, which contradicts the assumption hence proves the algorithm reaches $\mathbb{E}\left\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\right\|_2 \leq \nu$ within T steps.

D.3.1 Function value descent

Let $\delta_i^{(t)} = \frac{\tau}{\tau + \|\boldsymbol{g}_i^{(t)}\|_2}$ and $\delta^{(t)} = \frac{\tau}{\tau + \|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2}$. Similar to Appendix D.2.1, use Taylor expansion and take expectation conditioned on t, we can expand the function value descent as

$$
\begin{aligned}
&\mathbb{E}_t\left[f(\overline{\boldsymbol{x}}^{(t+1)}) - f(\overline{\boldsymbol{x}}^{(t)})\right] \\
&\leq \mathbb{E}_t\langle \nabla f(\overline{\boldsymbol{x}}^{(t)}), -\eta\overline{\boldsymbol{v}}^{(t+1)}\rangle + \frac{L}{2}\mathbb{E}_t\left\|\eta\overline{\boldsymbol{v}}^{(t+1)}\right\|_2^2 \\
&= -\eta\mathbb{E}_t\left\langle \nabla f(\overline{\boldsymbol{x}}^{(t)}), \overline{\boldsymbol{g}}_p^{(t+1)}\right\rangle + \frac{\eta^2 L}{2}\mathbb{E}_t\left\|\overline{\boldsymbol{v}}^{(t+1)}\right\|_2^2 \\
&= -\eta\mathbb{E}_t\left\langle \nabla f(\overline{\boldsymbol{x}}^{(t)}), \overline{\boldsymbol{g}}_\tau^{(t+1)}\right\rangle + \frac{\eta^2 L}{2}\mathbb{E}_t\left\|\overline{\boldsymbol{v}}^{(t+1)}\right\|_2^2 \\
&= -\eta\mathbb{E}_t\left\langle \nabla f(\overline{\boldsymbol{x}}^{(t)}), \mathrm{Clip}_\tau(\nabla f(\overline{\boldsymbol{x}}^{(t)}))\right\rangle + \eta\mathbb{E}_t\left\langle \nabla f(\overline{\boldsymbol{x}}^{(t)}), \mathrm{Clip}_\tau(\nabla f(\overline{\boldsymbol{x}}^{(t)})) - \overline{\boldsymbol{g}}_\tau^{(t+1)}\right\rangle + \frac{\eta^2 L}{2}\mathbb{E}_t\left\|\overline{\boldsymbol{v}}^{(t+1)}\right\|_2^2 \\
&= -\eta\delta^{(t)}\left\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\right\|_2^2 + \eta\mathbb{E}_t\left\langle \nabla f(\overline{\boldsymbol{x}}^{(t)}), \mathrm{Clip}_\tau(\nabla f(\overline{\boldsymbol{x}}^{(t)})) - \overline{\boldsymbol{g}}_\tau^{(t+1)}\right\rangle + \frac{\eta^2 L}{2}\mathbb{E}_t\left\|\overline{\boldsymbol{v}}^{(t+1)}\right\|_2^2 \\
&\leq -\eta\delta^{(t)}\left\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\right\|_2^2 + \eta\left\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\right\|_2\mathbb{E}_t\left\|\mathrm{Clip}_\tau(\nabla f(\overline{\boldsymbol{x}}^{(t)})) - \overline{\boldsymbol{g}}_\tau^{(t+1)}\right\|_2 + \frac{\eta^2 L}{2}\mathbb{E}_t\left\|\overline{\boldsymbol{v}}^{(t+1)}\right\|_2^2. \qquad \text{(D.20)}
\end{aligned}
$$

The $\mathbb{E}_t\left\|\mathrm{Clip}_\tau(\nabla f(\overline{\boldsymbol{x}}^{(t)})) - \overline{\boldsymbol{g}}_\tau^{(t+1)}\right\|_2$ term in (D.20) is the error introduced by gradient clipping, which can be analyzed by splitting it to 4 terms as following

$$
\begin{aligned}
&\mathbb{E}_t\left\|\mathrm{Clip}_\tau(\nabla f(\overline{\boldsymbol{x}}^{(t)})) - \overline{\boldsymbol{g}}_\tau^{(t+1)}\right\|_2 \\
&= \mathbb{E}_t\left\|\frac{1}{n}\sum_{i=1}^{n}\frac{\tau}{\tau + \|\boldsymbol{g}_i^{(t)}\|_2}\boldsymbol{g}_i^{(t)} - \frac{\tau}{\tau + \|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2}\nabla f(\overline{\boldsymbol{x}}^{(t)})\right\|_2 \\
&= \mathbb{E}_t\Bigg\|\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\tau}{\tau + \|\boldsymbol{g}_i^{(t)}\|_2}\boldsymbol{g}_i^{(t)} - \frac{\tau}{\tau + \|\nabla f_i(\boldsymbol{x}_i^{(t)})\|_2}\boldsymbol{g}_i^{(t)}\right) \\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\tau}{\tau + \|\nabla f_i(\boldsymbol{x}_i^{(t)})\|_2}\boldsymbol{g}_i^{(t)} - \frac{\tau}{\tau + \|\nabla f_i(\boldsymbol{x}_i^{(t)})\|_2}\nabla f_i(\boldsymbol{x}_i^{(t)})\right) \\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\tau}{\tau + \|\nabla f_i(\boldsymbol{x}_i^{(t)})\|_2}\nabla f_i(\boldsymbol{x}_i^{(t)}) - \frac{\tau}{\tau + \|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2}\nabla f_i(\boldsymbol{x}_i^{(t)})\right) \\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\tau}{\tau + \|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2}\nabla f_i(\boldsymbol{x}_i^{(t)}) - \frac{\tau}{\tau + \|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2}\nabla f(\overline{\boldsymbol{x}}^{(t)})\right)\Bigg\|_2
\end{aligned}
$$

$$\leq \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_t\left\|\Big(\frac{\tau}{\tau+\|g_i^{(t)}\|_2}-\frac{\tau}{\tau+\|\nabla f_i(x_i^{(t)})\|_2}\Big)g_i^{(t)}\right\|_2 \tag{D.21}$$

$$+\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_t\left\|\frac{\tau}{\tau+\|\nabla f_i(x_i^{(t)})\|_2}(g_i^{(t)}-\nabla f_i(x_i^{(t)}))\right\|_2 \tag{D.22}$$

$$+\frac{1}{n}\sum_{i=1}^{n}\left\|\Big(\frac{\tau}{\tau+\|\nabla f_i(x_i^{(t)})\|_2}-\frac{\tau}{\tau+\|\nabla f(\bar{x}^{(t)})\|_2}\Big)\nabla f_i(x_i^{(t)})\right\|_2 \tag{D.23}$$

$$+\left\|\frac{\tau}{\tau+\|\nabla f(\bar{x}^{(t)})\|_2}\Big(\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_i^{(t)})-\nabla f(\bar{x}^{(t)})\Big)\right\|_2. \tag{D.24}$$

Next, we bound each term separately using triangle inequality, Assumptions 9 and 10.

Bound the first term (D.21) as

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_t\left\|\Big(\frac{\tau}{\tau+\|g_i^{(t)}\|_2}-\frac{\tau}{\tau+\|\nabla f_i(x_i^{(t)})\|_2}\Big)g_i^{(t)}\right\|_2\\
&=\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_t\left\|\frac{\tau(\|g_i^{(t)}\|_2-\|\nabla f_i(x_i^{(t)})\|_2)}{(\tau+\|g_i^{(t)}\|_2)(\tau+\|\nabla f_i(x_i^{(t)})\|_2)}g_i^{(t)}\right\|_2\\
&=\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_t\left(\Big|\|g_i^{(t)}\|_2-\|\nabla f_i(x_i^{(t)})\|_2\Big|\cdot\frac{\tau}{\tau+\|\nabla f_i(x_i^{(t)})\|_2}\cdot\frac{\|g_i^{(t)}\|_2}{\tau+\|g_i^{(t)}\|_2}\right)\\
&\leq\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_t\Big|\|g_i^{(t)}\|_2-\|\nabla f_i(x_i^{(t)})\|_2\Big|\\
&\leq\frac{1}{n}\sum_{i=1}^{n}\sqrt{\mathbb{E}_t(\|g_i^{(t)}\|_2-\|\nabla f_i(x_i^{(t)})\|_2)^2}\\
&=\frac{1}{n}\sum_{i=1}^{n}\sqrt{\mathbb{E}_t(\|g_i^{(t)}\|_2^2+\|\nabla f_i(x_i^{(t)})\|_2^2-2\|g_i^{(t)}\|_2\|\nabla f_i(x_i^{(t)})\|_2)}\\
&\leq\frac{1}{n}\sum_{i=1}^{n}\sqrt{\mathbb{E}_t(\|g_i^{(t)}\|_2^2+\|\nabla f_i(x_i^{(t)})\|_2^2-2\langle g_i^{(t)},\nabla f_i(x_i^{(t)})\rangle)}\\
&=\frac{1}{n}\sum_{i=1}^{n}\sqrt{\mathbb{E}_t\|g_i^{(t)}-\nabla f_i(x_i^{(t)})\|_2^2}\\
&\leq\frac{\sigma_g}{\sqrt{b}}.
\end{aligned}\tag{D.25}$$

Bound the second term (D.22) as

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_t\left\|\frac{\tau}{\tau+\|\nabla f_i(x_i^{(t)})\|_2}(g_i^{(t)}-\nabla f_i(x_i^{(t)}))\right\|_2\leq\frac{1}{n}\sum_{i=1}^{n}\frac{\tau\sigma_g/\sqrt{b}}{\tau+\|\nabla f_i(x_i^{(t)})\|_2}\leq\frac{\sigma_g}{\sqrt{b}}. \tag{D.26}$$

Bound the third term (D.23) as

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\left\|\Big(\frac{\tau}{\tau+\|\nabla f_i(x_i^{(t)})\|_2}-\frac{\tau}{\tau+\|\nabla f(\bar{x}^{(t)})\|_2}\Big)\nabla f_i(x_i^{(t)})\right\|_2\\
&=\frac{1}{n}\sum_{i=1}^{n}\left\|\frac{\tau(\|\nabla f_i(x_i^{(t)})\|_2-\|\nabla f(\bar{x}^{(t)})\|_2)}{(\tau+\|\nabla f_i(x_i^{(t)})\|_2)(\tau+\|\nabla f(\bar{x}^{(t)})\|_2)}\nabla f_i(x_i^{(t)})\right\|_2
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{1}{n}\sum_{i=1}^{n}\frac{\tau\Big|\big\|\nabla f_i(\boldsymbol{x}_i^{(t)})\big\|_2-\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2\Big|}{\tau+\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2}\\
&\leq \frac{1}{n}\sum_{i=1}^{n}\delta^{(t)}\Big|\big\|\nabla f_i(\boldsymbol{x}_i^{(t)})\big\|_2-\big\|\nabla f_i(\overline{\boldsymbol{x}}^{(t)})\big\|_2\Big|+\frac{1}{n}\sum_{i=1}^{n}\delta^{(t)}\Big|\big\|\nabla f_i(\overline{\boldsymbol{x}}^{(t)})\big\|_2-\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2\Big|\\
&\leq \frac{1}{n}\sum_{i=1}^{n}\delta^{(t)}L\big\|\boldsymbol{x}_i^{(t)}-\overline{\boldsymbol{x}}^{(t)}\big\|_2+\frac{1}{n}\sum_{i=1}^{n}\delta^{(t)}\cdot\frac{1}{12}\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2\\
&\leq \frac{\delta^{(t)}L}{\sqrt{n}}\big\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}+\frac{1}{12}\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2, \qquad \text{(D.27)}
\end{aligned}
$$

where we use $\delta^{(t)}\leq 1$ to reach the last inequality.

Bound (D.24) as

$$
\begin{aligned}
&\left\|\frac{\tau}{\tau+\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2}\Big(\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x}_i^{(t)})-\nabla f(\overline{\boldsymbol{x}}^{(t)})\Big)\right\|_2\\
&=\frac{\tau}{\tau+\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2}\big\|\nabla F(\boldsymbol{X}^{(t)})(\tfrac{1}{n}\mathbf{1}_n)-\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2\\
&\leq\frac{\delta^{(t)}L}{\sqrt{n}}\big\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}. \qquad \text{(D.28)}
\end{aligned}
$$

Using (D.25), (D.26), (D.27) and (D.28), the function value descent inequality (D.20) becomes

$$
\begin{aligned}
\mathbb{E}_t\big[f(\overline{\boldsymbol{x}}^{(t+1)})-f(\overline{\boldsymbol{x}}^{(t)})\big]&\leq-\eta\delta^{(t)}\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2^2+\frac{\eta^2L}{2}\mathbb{E}_t\big\|\overline{\boldsymbol{v}}^{(t+1)}\big\|_2^2\\
&\quad+\eta\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2\mathbb{E}_t\big\|\mathrm{Clip}_\tau\big(\nabla f(\overline{\boldsymbol{x}}^{(t)})\big)-\overline{\boldsymbol{g}}_\tau^{(t+1)}\big\|_2\\
&\leq-\eta\delta^{(t)}\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2^2+\frac{\eta^2L}{2}\mathbb{E}_t\big\|\overline{\boldsymbol{v}}^{(t+1)}\big\|_2^2\\
&\quad+\eta\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2\Big(\frac{2\sigma_g}{\sqrt{b}}+\frac{1}{12}\delta^{(t)}\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\|_2+\frac{2\delta^{(t)}L}{\sqrt{n}}\big\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}\Big)\\
&=-\frac{11}{12}\eta\delta^{(t)}\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2^2+\frac{\eta^2L}{2}\mathbb{E}_t\big\|\overline{\boldsymbol{v}}^{(t+1)}\big\|_2^2\\
&\quad+\eta\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2\Big(\frac{2\sigma_g}{\sqrt{b}}+\frac{2\delta^{(t)}L}{\sqrt{n}}\big\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}\Big)\\
&\leq-\frac{5}{12}\eta\delta^{(t)}\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2^2+\frac{\eta^2L}{2}\mathbb{E}_t\big\|\overline{\boldsymbol{v}}^{(t+1)}\big\|_2^2\\
&\quad+\frac{2\eta\sigma_g}{\sqrt{b}}\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2+\frac{2\delta^{(t)}\eta L^2}{n}\big\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}^2, \qquad \text{(D.29)}
\end{aligned}
$$

where the last inequality is due to

$$
\begin{aligned}
&\eta\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2\cdot\frac{2\delta^{(t)}L}{\sqrt{n}}\big\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}\\
&\leq\eta\delta^{(t)}\cdot 2\cdot\sqrt{\frac{1}{2}\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2^2}\cdot\sqrt{\frac{2L^2}{n}\big\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}^2}\\
&\leq\frac{1}{2}\eta\delta^{(t)}\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2^2+\frac{2\delta^{(t)}\eta L^2}{n}\big\|\boldsymbol{X}^{(t)}-\overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}^2.
\end{aligned}
$$

D.3.2 Convergence rate

Different from (D.17), with the use of gradient clipping operator, we can only bound the expected norm of average gradient estimate as

$$\begin{aligned}\mathbb{E}\|\overline{\boldsymbol{v}}^{(t)}\|_2^2 &= \mathbb{E}\|\overline{\boldsymbol{g}}_p^{(t)}\|_2^2 \\ &= \mathbb{E}\|\overline{\boldsymbol{g}}_\tau^{(t)} + \overline{\boldsymbol{e}}^{(t)}\|_2^2 \\ &= \mathbb{E}\|\overline{\boldsymbol{g}}_\tau^{(t)}\|_2^2 + \mathbb{E}\|\overline{\boldsymbol{e}}^{(t)}\|_2^2 \\ &\leq \tau^2 + \frac{\sigma_p^2 d}{n}. \end{aligned} \tag{D.30}$$

Let $\Delta = \mathbb{E}\big[f(\overline{\boldsymbol{x}}^{(0)})\big] - f^\star$. The techniques used is similar to that used in Appendix D.2, so that we can reuse results from Appendices D.2.2 and D.2.3, namely (D.16), in the following proof. Take full expectation and use (D.30), sum (D.29) over $t = 1, \ldots, T$,

$$\begin{aligned}-\Delta &\leq -\frac{5\eta}{12}\sum_{t=1}^T \mathbb{E}\Big(\delta^{(t)}\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2^2\Big) + \frac{2\eta\sigma}{\sqrt{b}}\sum_{t=1}^T \mathbb{E}\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2 \\ &\quad + \frac{2\eta L^2}{n}\sum_{t=1}^T \mathbb{E}\big\|\boldsymbol{X}^{(t)} - \overline{\boldsymbol{x}}^{(t)}\mathbf{1}_n^\top\big\|_{\mathsf{F}}^2 + \frac{\eta^2 L}{2}\sum_{t=1}^T \mathbb{E}\big\|\overline{\boldsymbol{v}}^{(t+1)}\big\|_2^2 \\ &\leq -\frac{5\eta}{12}\sum_{t=1}^T \mathbb{E}\Big(\delta^{(t)}\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2^2\Big) + \frac{2\eta\sigma}{\sqrt{b}}\sum_{t=1}^T \mathbb{E}\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2 \\ &\quad + \frac{2\eta L^2}{n}\Big(\frac{8\eta^2}{(1-\hat{\alpha})^2}\sum_{t=1}^T n\mathbb{E}\big\|\overline{\boldsymbol{v}}^{(t)}\big\|_2^2 + \frac{1024\eta^2}{(1-\hat{\alpha})^4}Tn(\tau^2+\sigma_p^2 d)\Big) + \frac{\eta^2 L}{2}\sum_{t=1}^T \mathbb{E}\big\|\overline{\boldsymbol{v}}^{(t+1)}\big\|_2^2 \\ &= -\frac{5\eta}{12}\sum_{t=1}^T \mathbb{E}\Big(\delta^{(t)}\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2^2\Big) + \frac{2\eta\sigma}{\sqrt{b}}\sum_{t=1}^T \mathbb{E}\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2 \\ &\quad + \frac{16\eta^3 L^2}{(1-\hat{\alpha})^2}\Big(\tau^2 + \frac{\sigma_p^2 d}{n}\Big) + \frac{2048\eta^3 L^2}{(1-\hat{\alpha})^4}T(\tau^2+\sigma_p^2 d) + \frac{\eta^2 LT}{2}\Big(\tau^2 + \frac{\sigma_p^2 d}{n}\Big). \end{aligned} \tag{D.31}$$

Compared to the descent inequality in Appendix D.2.4, (D.31) has a unique terms: $\mathbb{E}\big(\delta^{(t)}\big\|\nabla f(\overline{\boldsymbol{x}}^{(t)})\big\|_2^2\big)$ which is expected norm of clipped gradients. To be able to analyze this terms, we need to use convexity and monotonicity from Lemma 11.

Lemma 11. *Let $g(x) = \frac{x}{c+x}$ and $h(x) = xg(x) = \frac{x^2}{c+x}$. When $x \geq 0$, $g(x)$ and $h(x)$ increase monotonically, while $g(x)$ is concave and $h(x)$ is convex.*

Proof of Lemma 11. It is sufficient to prove Lemma 11 by evaluating the first-order and second-order derivatives of $g(x)$ and $h(x)$.

Because $g'(x) = \frac{(c+x)-x}{(c+x)^2} = \frac{c}{(c+x)^2} > 0$ and $h'(x) = g(x) + xg'(x) \geq 0$, $g(x)$ and $h(x)$ increase monotonically.

$g(x)$ is concave because $g''(x) - \frac{2c(c+x)}{(c+x)^4} = -\frac{2c}{(c+x)^3} < 0$.

$h(x)$ is convex because $h''(x) = 2g'(x) + xg''(x) = \frac{2c}{(c+x)^2} - \frac{2cx}{(c+x)^3} = \frac{2c^2}{(c+x)^3} > 0$. □

Next, we substitute $\tau = \nu$ (cf. Theorem 11), and assume the following inequality

$$\mathbb{E}\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\|_2 \geq \nu. \tag{D.32}$$

By Lemma 11, the expectation of clipped gradients can be bounded as

$$\begin{aligned}
\mathbb{E}\Big(\delta^{(t)}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2^2\Big) &= \mathbb{E}\Big(\frac{\tau\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2^2}{\tau + \big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2}\Big) \\
&\geq \frac{\tau\big(\mathbb{E}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2\big)^2}{\tau + \mathbb{E}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2} \\
&\geq \frac{\tau\nu}{\tau+\nu}\mathbb{E}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2 \\
&= \frac{\nu}{2}\mathbb{E}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2.
\end{aligned} \tag{D.33}$$

Using (D.30), (D.33) and Assumptions 9 and 10, we can further bound the RHS of (D.31) as

$$\begin{aligned}
-\Delta &\leq -\frac{5\eta\nu}{24}\sum_{t=1}^{T}\mathbb{E}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2 + \frac{2\eta\sigma_g}{\sqrt{b}}\sum_{t=1}^{T}\mathbb{E}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2 \\
&\quad + \frac{16\eta^3L^2}{(1-\hat{\alpha})^2}\Big(\tau^2 + \frac{\sigma_p^2 d}{n}\Big) + \frac{2048\eta^3L^2}{(1-\hat{\alpha})^4}T(\tau^2+\sigma_p^2 d) + \frac{\eta^2 LT}{2}\Big(\tau^2 + \frac{\sigma_p^2 d}{n}\Big) \\
&\leq -\frac{\eta\nu}{8}\sum_{t=1}^{T}\mathbb{E}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2 + \frac{2048T\eta^3L^2}{(1-\hat{\alpha})^4}(\tau^2+\sigma_p^2 d) + \frac{3T\eta^2 L}{1-\hat{\alpha}}\Big(\tau^2 + \frac{\sigma_p^2 d}{n}\Big) \\
&= -\frac{\eta\nu}{8}\sum_{t=1}^{T}\mathbb{E}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2 + \frac{2048T\eta^3L^2\tau^2}{(1-\hat{\alpha})^4}(1+Tb^2\phi_m^2) + \frac{3T\eta^2L\tau^2}{1-\hat{\alpha}}(1+Tb^2\phi_m^2),
\end{aligned} \tag{D.34}$$

where we use (D.18) and $b = \big(\frac{24\sigma_g}{\nu}\big)^2$ to prove the last inequality.

Reorganize terms, (D.34) can be further bounded as

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2 &\leq \frac{8\Delta}{\eta\nu T} + \frac{16384\eta^2L^2\nu}{(1-\hat{\alpha})^4}(1+Tb^2\phi_m^2) + \frac{24\eta L\nu}{1-\hat{\alpha}}(1+Tb^2\phi_m^2) \\
&\overset{\text{(i)}}{\leq} \frac{8\Delta b\phi_m}{\eta\nu} + \frac{32768\eta^2L^2\nu}{(1-\hat{\alpha})^4} + \frac{48\eta L\nu}{1-\hat{\alpha}} \\
&\overset{\text{(ii)}}{\leq} \frac{8\Delta b\phi_m}{\eta\nu} + \frac{4096\eta L\nu}{(1-\hat{\alpha})^{\frac{8}{3}}} + \frac{48\eta L\nu}{1-\hat{\alpha}}, \\
&\leq \frac{8\Delta b\phi_m}{\eta\nu} + \frac{4144\eta L\nu}{(1-\hat{\alpha})^{\frac{8}{3}}},
\end{aligned} \tag{D.35}$$

where we substitute $T = (b\phi_m)^{-1}$ to prove (i), use (D.18) for (ii), and use $(1-\hat{\alpha})^{-1} \leq (1-\hat{\alpha})^{-8/3}$ to prove the last inequality.

Set $\eta = \frac{\gamma^{\frac{4}{3}}(1-\alpha)^{\frac{4}{3}}}{24}\cdot\sqrt{\frac{\Delta}{L}}\cdot\frac{\sqrt{b\phi_m}}{\tau}$ and $\gamma = \frac{1}{100}(1-\alpha)\rho$, and use $b = \big(\frac{24\sigma_g}{\nu}\big)^2$, (D.35) can be further bounded as

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2 \leq \frac{192\sqrt{L\Delta b\phi_m}}{(1-\hat{\alpha})^{\frac{4}{3}}} + \frac{173\sqrt{L\Delta b\phi_m}}{(1-\hat{\alpha})^{\frac{4}{3}}}$$

$$= \frac{365\sqrt{L\Delta b\phi_m}}{(1-\hat{\alpha})^{\frac{4}{3}}}$$
$$= \frac{8760\sqrt{L\Delta\phi_m}}{(1-\hat{\alpha})^{\frac{4}{3}}} \cdot \frac{\sigma_g}{\nu}. \tag{D.36}$$

Choosing $\nu = \sqrt{\frac{8760\sigma_g\sqrt{L\Delta\phi_m}}{(1-\hat{\alpha})^{\frac{4}{3}}}}$, (D.36) simplifies to $\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\right\|_2 < \nu$, which further implies $\exists t \in [T]$ such that $\mathbb{E}\left\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\right\|_2 < \nu$. However, this contradicts the assumption (D.32), which leads to the convergence results in the theorem.

Lastly, we can verify conditions (D.5), (D.9) and (D.18) are all met to conclude the proof:

$$\gamma^2 \leq \frac{\rho^2}{10000} \qquad \Rightarrow \qquad \text{(D.5)}$$
$$\gamma^4 = \gamma^2 \cdot \frac{(1-\alpha)^2\rho^2}{10000} \leq \frac{(1-\hat{\alpha})^2\rho^2}{10000} \qquad \Rightarrow \qquad \text{(D.9)}$$
$$\eta L = \frac{\gamma^{\frac{4}{3}}(1-\alpha)^{\frac{4}{3}}}{24} \cdot \frac{\sqrt{bL\Delta\phi_m}}{\tau}$$
$$= \gamma^{\frac{4}{3}}(1-\alpha)^{\frac{4}{3}} \cdot \frac{\sigma_g\sqrt{L\Delta\phi_m}}{\nu^2}$$
$$= \gamma^{\frac{4}{3}}(1-\alpha)^{\frac{4}{3}} \cdot \frac{(1-\hat{\alpha})^{\frac{4}{3}}}{8760} \qquad \Rightarrow \qquad \text{(D.18)}$$

Bibliography

[ABCP13] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 901–914, New York, NY, USA, November 2013. Association for Computing Machinery.

[ACCÖ21] S. Asoodeh, W.-N. Chen, F. P. Calmon, and A. Özgür. Differentially private federated learning: An information-theoretic perspective. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 344–349, July 2021.

[ACG+16] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, October 2016.

[AGL+17] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic. QSGD: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[AHJ+18] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli. The convergence of sparsified gradient methods. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[ALBR19] M. Assran, N. Loizou, N. Ballas, and M. Rabbat. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, pages 344–353. PMLR, 2019.

[AS14] M. Arioli and J. Scott. Chebyshev acceleration of iterative refinement. *Numerical Algorithms*, 66(3):591–608, 2014.

[AZ17] Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, page 1200–1205, New York, NY, USA, 2017. Association for Computing Machinery.

[AZH16] Z. Allen-Zhu and E. Hazan. Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning*, pages 699–707. PMLR, 2016.

[BBKW19] A. S. Berahas, R. Bollapragada, N. S. Keskar, and E. Wei. Balancing communication and computation in distributed optimization. *IEEE Transactions on Automatic Control*, 64(8):3141–3155, 2019.

[BBP13] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu. Advances in optimizing recurrent networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8624–8628, May 2013.

[BBW21] A. S. Berahas, R. Bollapragada, and E. Wei. On the convergence of nested decentralized gradient methods with multiple consensus and gradient steps. *IEEE Transactions on Signal Processing*, 69:4192–4203, 2021.

[BJ13] P. Bianchi and J. Jakubowicz. Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization. *IEEE Transactions on Automatic Control*, 58(2):391–405, February 2013.

[BPC+11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[BT89] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ, 1989.

[BT09] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[CABP13] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi. Broadening the scope of differential privacy using metrics. In E. De Cristofaro and M. Wright, editors, *Privacy Enhancing Technologies*, Lecture Notes in Computer Science, pages 82–102, Berlin, Heidelberg, 2013. Springer.

[CEBM22] E. Cyffers, M. Even, A. Bellet, and L. Massoulié. Muffliato: Peer-to-peer privacy amplification for decentralized optimization and averaging. In *Advances in Neural Information Processing Systems*, 2022.

[CL11] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1–27, May 2011.

[CSU+19] A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev. Distributed differential privacy via shuffling. In Y. Ishai and V. Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, Lecture Notes in Computer Science, pages 375–403, Cham, 2019. Springer International Publishing.

[CWH20] X. Chen, S. Z. Wu, and M. Hong. Understanding gradient clipping in private SGD: A geometric perspective. In *Advances in Neural Information Processing Systems*, volume 33, pages 13773–13782. Curran Associates, Inc., 2020.

[CZC+20] S. Cen, H. Zhang, Y. Chi, W. Chen, and T.-Y. Liu. Convergence of distributed stochastic variance reduced methods without sampling extra data. *IEEE Transactions on Signal Processing*, 68:3976–3989, 2020.

[DBLJ14] A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

[Den12] L. Deng. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[DJW13] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438, October 2013.

[DKX+22] R. Das, S. Kale, Z. Xu, T. Zhang, and S. Sanghavi. Beyond uniform Lipschitz condition in differentially private optimization. *arXiv preprint arXiv:2206.10713*, 2022.

[DLC+22] J. Du, S. Li, X. Chen, S. Chen, and M. Hong. Dynamic differential-privacy preserving sgd. *arXiv preprint arXiv:2111.00173*, 2022.

[DLS16] P. Di Lorenzo and G. Scutari. Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016.

[DMNS06] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi and T. Rabin, editors, *Theory of Cryptography*, Lecture Notes in Computer Science, pages 265–284, Berlin, Heidelberg, 2006. Springer.

[DR14] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, August 2014.

[DSZOR15] C. M. De Sa, C. Zhang, K. Olukotun, and C. Ré. Taming the wild: A unified analysis of Hogwild-style algorithms. *Advances in Neural Information Processing Systems*, 28, 2015.

[Dwo08] C. Dwork. Differential privacy: A survey of results. In M. Agrawal, D. Du, Z. Duan, and A. Li, editors, *Theory and Applications of Models of Computation*, Lecture Notes in Computer Science, pages 1–19, Berlin, Heidelberg, 2008. Springer.

[FGW21] J. Fan, Y. Guo, and K. Wang. Communication-efficient accurate statistical estimation. *Journal of the American Statistical Association*, 0(0):1–11, August 2021.

[FKT20] V. Feldman, T. Koren, and K. Talwar. Private stochastic convex optimization: Optimal rates in linear time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pages 439–449, New York, NY, USA, June 2020. Association for Computing Machinery.

[FLLZ18] C. Fang, C. J. Li, Z. Lin, and T. Zhang. SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pages 687–697, 2018.

[FSG+21] I. Fatkhullin, I. Sokolov, E. Gorbunov, Z. Li, and P. Richtárik. EF21 with bells & whistles: Practical algorithmic extensions of modern error feedback. *arXiv preprint arXiv:2110.03294*, 2021.

[GBLR21] E. Gorbunov, K. P. Burlachenko, Z. Li, and P. Richtarik. MARINA: Faster non-convex distributed learning with compression. In *Proceedings of the 38th International Conference on Machine Learning*, pages 3788–3798. PMLR, July 2021.

[GF20] L. V. Gambuzza and M. Frasca. Distributed control of multiconsensus. *IEEE Transactions on Automatic Control*, 66(5):2032–2044, 2020.

[HAD+21] A. Hashemi, A. Acharya, R. Das, H. Vikalo, S. Sanghavi, and I. S. Dhillon. On the benefits of multiple gossip steps in communication-constrained decentralized federated learning. *IEEE Transactions on Parallel and Distributed Systems*, 2021.

[HDJ+20] X. Huang, Y. Ding, Z. L. Jiang, S. Qi, X. Wang, and Q. Liao. DP-FL: A novel differentially private federated learning framework for the unbalanced data. *World Wide Web*, 23(4):2529–2545, July 2020.

[HSZ+22] Y. Huang, Y. Sun, Z. Zhu, C. Yan, and J. Xu. Tackling data heterogeneity: A new unified framework for decentralized sgd with sample-induced topology. In *Proceedings of the 39th International Conference on Machine Learning*, pages 9310–9345. PMLR, June 2022.

[INS+19] R. Iyengar, J. P. Near, D. Song, O. Thakkar, A. Thakurta, and L. Wang. Towards practical differentially private convex optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 299–316, May 2019.

[IW22] C. Iakovidou and E. Wei. S-NEAR-DGD: A flexible distributed stochastic gradient method for inexact communication. *IEEE Transactions on Automatic Control*, 2022.

[JZ13] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

[KDG03] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 482–491, October 2003.

[KFI17] S. Kanai, Y. Fujiwara, and S. Iwamura. Preventing gradient explosions in gated recurrent units. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[KFJ18] S. Khirirat, H. R. Feyzmahdavian, and M. Johansson. Distributed learning with compressed gradients. *arXiv preprint arXiv:1806.06573*, 2018.

[KKJ+21] D. Kovalev, A. Koloskova, M. Jaggi, P. Richtarik, and S. Stich. A linearly convergent algorithm for decentralized optimization: Sending less bits for free! In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 4087–4095. PMLR, 13–15 Apr 2021.

[KLL16] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016.

[KMR15] J. Konečnỳ, B. McMahan, and D. Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015.

[KMY+16] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[KRSJ19] S. P. Karimireddy, Q. Rebjock, S. Stich, and M. Jaggi. Error feedback fixes signsgd and other gradient compression schemes. In *International Conference on Machine Learning*, pages 3252–3261. PMLR, 2019.

[KSJ19] A. Koloskova, S. Stich, and M. Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3478–3487. PMLR, 09–15 Jun 2019.

[Lam01] L. Lamberg. Confidentiality and privacy of electronic medical records. *JAMA*, 285(24):3075–3076, June 2001.

[LBZR21] Z. Li, H. Bao, X. Zhang, and P. Richtárik. Page: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *International Conference on Machine Learning*, pages 6286–6295. PMLR, 2021.

[LCCC20] B. Li, S. Cen, Y. Chen, and Y. Chi. Communication-efficient distributed optimization in networks with gradient tracking and variance reduction. *Journal of Machine Learning Research*, 21(180):1–51, 2020.

[LDS21] Y. Lu and C. De Sa. Optimal complexity in decentralized training. In *International Conference on Machine Learning*, pages 7111–7123. PMLR, 2021.

[LFYL20] H. Li, C. Fang, W. Yin, and Z. Lin. Decentralized accelerated gradient methods with increasing penalty parameters. *IEEE Transactions on Signal Processing*, 68:4855–4870, 2020.

[Li19] Z. Li. SSRGD: Simple stochastic recursive gradient descent for escaping saddle points. In *Advances in Neural Information Processing Systems*, pages 1523–1533, 2019.

[LJB+95] Y. LeCun, L. D. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, and P. Simard. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261(276):2, 1995.

[LJCJ17] L. Lei, C. Ju, J. Chen, and M. I. Jordan. Non-convex finite-sum optimization via SCSG methods. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[LKQR20] Z. Li, D. Kovalev, X. Qian, and P. Richtarik. Acceleration for compressed gradient descent in distributed and federated optimization. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5895–5904. PMLR, November 2020.

[LL18] Z. Li and J. Li. A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 5569–5579, 2018.

[LLC22] B. Li, Z. Li, and Y. Chi. DESTRESS: Computation-optimal and communication-efficient decentralized nonconvex finite-sum optimization. *SIAM Journal on Mathematics of Data Science*, 4(3):1031–1051, September 2022.

[LLHP22] Y. Liao, Z. Li, K. Huang, and S. Pu. A compressed gradient tracking method for decentralized optimization with linear convergence. *IEEE Transactions on Automatic Control*, 67(10):5622–5629, 2022.

[LLP23] Y. Liao, Z. Li, and S. Pu. A linearly convergent robust compressed Push-Pull method for decentralized optimization. *arXiv preprint arXiv:2303.07091*, 2023.

[LR20] Z. Li and P. Richtárik. A unified analysis of stochastic gradient methods for nonconvex federated optimization. *arXiv preprint arXiv:2006.07013*, 2020.

[LR21a] Z. Li and P. Richtarik. CANITA: Faster rates for distributed convex optimization with communication compression. In *Advances in Neural Information Processing Systems*, volume 34, pages 13770–13781. Curran Associates, Inc., 2021.

[LR21b] Z. Li and P. Richtárik. ZeroSARAH: Efficient nonconvex finite-sum optimization with zero full gradient computation. *arXiv preprint arXiv:2103.01447*, 2021.

[LSY19] Z. Li, W. Shi, and M. Yan. A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Transactions on Signal Processing*, 67(17):4494–4506, 2019.

[LY22] L. Luo and H. Ye. An optimal stochastic algorithm for decentralized nonconvex finite-sum optimization. *arXiv preprint arXiv:2210.13931*, 2022.

[LZLC22] Z. Li, H. Zhao, B. Li, and Y. Chi. SoteriaFL: A unified framework for private federated learning with communication compression. In *Advances in Neural Information Processing Systems*, volume 35, pages 4285–4300. Curran Associates, Inc., 2022.

[LZZ+17] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 5330–5340, 2017.

[MGTR19] K. Mishchenko, E. Gorbunov, M. Takáč, and P. Richtárik. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019.

[MMR+17] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[MS23] T. Murata and T. Suzuki. DIFF2: Differential private optimization via gradient differences for nonconvex distributed learning. *arXiv preprint arXiv:2302.03884*, 2023.

[NBD22] M. Noble, A. Bellet, and A. Dieuleveut. Differentially private federated learning on heterogeneous data. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pages 10110–10145. PMLR, May 2022.

[NLST17] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, pages 2613–2621, 2017.

[NO09] A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[NOP10] A. Nedić, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.

[NOR18] A. Nedić, A. Olshevsky, and M. G. Rabbat. Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5):953–976, 2018.

[NOS17] A. Nedić, A. Olshevsky, and W. Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.

[NvP+22] L. M. Nguyen, M. van Dijk, D. T. Phan, P. H. Nguyen, T.-W. Weng, and J. R. Kalagnanam. Finite-sum smooth optimization with SARAH. *Computational Optimization and Applications*, May 2022.

[PLW20] T. Pan, J. Liu, and J. 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.

[PMB13] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1310–1318. PMLR, May 2013.

[Pol63] B. T. Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963.

[QL18] G. Qu and N. Li. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260, 2018.

[RHS+16] S. J. Reddi, A. Hefny, S. Sra, B. Póczós, and A. Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323, 2016.

[RKR+16] S. J. Reddi, J. Konečnỳ, P. Richtárik, B. Póczós, and A. Smola. AIDE: fast and communication efficient distributed optimization. *arXiv preprint arXiv:1608.06879*, 2016.

[RLDJ23] A. Reisizadeh, H. Li, S. Das, and A. Jadbabaie. Variance-reduced clipping for non-convex optimization. *arXiv preprint arXiv:2303.00883*, 2023.

[RSF21] P. Richtarik, I. Sokolov, and I. Fatkhullin. EF21: A new, simpler, theoretically better, and practically faster error feedback. In *Advances in Neural Information Processing Systems*, volume 34, pages 4384–4396. Curran Associates, Inc., 2021.

[RSPS16] S. J. Reddi, S. Sra, B. Póczos, and A. Smola. Fast incremental method for smooth nonconvex optimization. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1971–1977. IEEE, 2016.

[SBB+17] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *International Conference on Machine Learning*, pages 3027–3036, 2017.

[SBB+18] K. Scaman, F. Bach, S. Bubeck, L. Massoulié, and Y. T. Lee. Optimal algorithms for non-smooth distributed optimization in networks. In *Advances in Neural Information Processing Systems*, pages 2740–2749, 2018.

[SCJ18] S. U. Stich, J.-B. Cordonnier, and M. Jaggi. Sparsified SGD with memory. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[SDGD21] N. Singh, D. Data, J. George, and S. Diggavi. SQuARM-SGD: Communication-Efficient Momentum SGD for Decentralized Optimization. *IEEE Journal on Selected Areas in Information Theory*, 2(3):954–969, September 2021.

[SFD+14] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In *Interspeech 2014*, pages 1058–1062. ISCA, September 2014.

[Sha07] D. Shah. Gossip algorithms. *Foundations and Trends® in Networking*, 3(1):1–125, 2007.

[SK20] S. U. Stich and S. P. Karimireddy. The error-feedback framework: Better rates for sgd with delayed gradients and compressed updates. *J. Mach. Learn. Res.*, 21(1), jan 2020.

[SLH20] H. Sun, S. Lu, and M. Hong. Improving the sample and communication complexity for decentralized non-convex optimization: Joint gradient estimation and tracking. In *International Conference on Machine Learning*, pages 9217–9228. PMLR, 2020.

[SLWY15a] W. Shi, Q. Ling, G. Wu, and W. Yin. EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

[SLWY15b] W. Shi, Q. Ling, G. Wu, and W. Yin. A proximal gradient algorithm for decentralized composite optimization. *IEEE Transactions on Signal Processing*, 63(22):6013–6023, 2015.

[SS19] G. Scutari and Y. Sun. Distributed nonconvex constrained optimization over time-varying digraphs. *Mathematical Programming*, 176(1-2):497–544, 2019.

[SSZ14] O. Shamir, N. Srebro, and T. Zhang. Communication-efficient distributed optimization using an approximate Newton-type method. In *International conference on machine learning*, pages 1000–1008, 2014.

[TGZ+18] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu. Communication compression for decentralized training. *Advances in Neural Information Processing Systems*, 31:7652–7662, 2018.

[TLQ+19] H. Tang, X. Lian, S. Qiu, L. Yuan, C. Zhang, T. Zhang, and J. Liu. Deepsqueeze: Decentralization meets error-compensated compression. *arXiv preprint arXiv:1907.07346*, 2019.

[TLY+18] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu. D^2: Decentralized training over decentralized data. In *International Conference on Machine Learning*, pages 4848–4856. PMLR, 2018.

[TMHP20] H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani. Quantized decentralized stochastic learning over directed graphs. In *Proceedings of the 37th International Conference on Machine Learning*, pages 9324–9333. PMLR, November 2020.

[WJ21] J. Wang and G. Joshi. Cooperative SGD: A unified framework for the design and analysis of local-update SGD algorithms. *The Journal of Machine Learning Research*, 22(1):213:9709–213:9758, January 2021.

[WJEG19] L. Wang, B. Jayaraman, D. Evans, and Q. Gu. Efficient privacy-preserving stochastic nonconvex optimization. *arXiv preprint arXiv:1910.13659*, 2019.

[WJZ+19] Z. Wang, K. Ji, Y. Zhou, Y. Liang, and V. Tarokh. SpiderBoost and momentum: Faster variance reduction algorithms. In *Advances in Neural Information Processing Systems*, pages 2406–2416, 2019.

[WXDX20] D. Wang, H. Xiao, S. Devadas, and J. Xu. On differentially private stochastic convex optimization with heavy-tailed data. In *Proceedings of the 37th International Conference on Machine Learning*, pages 10081–10091. PMLR, November 2020.

[WYWH18] H.-T. Wai, Z. Yang, P. Z. Wang, and M. Hong. Multi-agent reinforcement learning via double averaging primal-dual optimization. In *Advances in Neural Information Processing Systems*, pages 9649–9660, 2018.

[WYX17] D. Wang, M. Ye, and J. Xu. Differentially private empirical risk minimization revisited: Faster and more general. *Advances in Neural Information Processing Systems*, 30, 2017.

[XB04] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, September 2004.

[XKK22a] R. Xin, U. A. Khan, and S. Kar. Fast decentralized nonconvex finite-sum optimization with recursive variance reduction. *SIAM Journal on Optimization*, 32(1):1–28, March 2022.

[XKK22b] R. Xin, U. A. Khan, and S. Kar. A Fast Randomized Incremental Gradient Method for Decentralized Nonconvex Optimization. *IEEE Transactions on Automatic Control*, 67(10):5150–5165, October 2022.

[XSKK19] R. Xin, A. K. Sahu, U. A. Khan, and S. Kar. Distributed stochastic optimization with gradient tracking over strongly-connected networks. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 8353–8358, 2019.

[XXK17] C. Xi, R. Xin, and U. A. Khan. ADD-OPT: Accelerated distributed directed optimization. *IEEE Transactions on Automatic Control*, 63(5):1329–1339, 2017.

[XYD19] H. Xiao, Y. Ye, and S. Devadas. Local differential privacy in decentralized optimization. *arXiv preprint arXiv:1902.06101*, 2019.

[XZ14] L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

[YCC+23] Y. Yan, J. Chen, P.-Y. Chen, X. Cui, S. Lu, and Y. Xu. Compressed decentralized proximal stochastic gradient method for nonconvex composite problems with heterogeneous data. *arXiv preprint arXiv:2302.14252*, 2023.

[YGG17] Y. You, I. Gitman, and B. Ginsburg. Scaling SGD batch size to 32k for ImageNet training. Technical Report UCB/EECS-2017-156, EECS Department, University of California, Berkeley, Sep 2017.

[YYZS18] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed. Exact diffusion for distributed optimization and learning – part I: Algorithm development. *IEEE Transactions on Signal Processing*, 67(3):708–723, 2018.

[YZLZ20] H. Ye, Z. Zhou, L. Luo, and T. Zhang. Decentralized accelerated proximal gradient descent. *Advances in Neural Information Processing Systems*, 33:18308–18317, 2020.

[ZBLR21] H. Zhao, K. Burlachenko, Z. Li, and P. Richtárik. Faster rates for compressed federated learning with client-variance reduction. *arXiv preprint arXiv:2112.13097*, 2021.

[ZCH+22] X. Zhang, X. Chen, M. Hong, S. Wu, and J. Yi. Understanding clipping for federated learning: Convergence and client-level differential privacy. In *Proceedings of the 39th International Conference on Machine Learning*, pages 26048–26067. PMLR, June 2022.

[ZHSJ20] J. Zhang, T. He, S. Sra, and A. Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, March 2020.

[ZJFW20] B. Zhang, J. Jin, C. Fang, and L. Wang. Improved analysis of clipping algorithms for non-convex optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 15511–15521. Curran Associates, Inc., 2020.

[ZKL18] X. Zhang, M. M. Khalili, and M. Liu. Improving the privacy and accuracy of ADMM-based distributed algorithms. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5796–5805. PMLR, July 2018.

[ZKL20] X. Zhang, M. M. Khalili, and M. Liu. Recycled ADMM: Improving the privacy and accuracy of distributed algorithms. *IEEE Transactions on Information Forensics and Security*, 15:1723–1734, 2020.

[ZKV+20] J. Zhang, S. P. Karimireddy, A. Veit, S. Kim, S. J. Reddi, S. Kumar, and S. Sra. Why are adaptive methods good for attention models? *arXiv preprint arXiv:1912.03194*, October 2020.

[ZLL+22] H. Zhao, B. Li, Z. Li, P. Richtárik, and Y. Chi. Beer: Fast O (1/T) rate for decentralized nonconvex optimization with communication compression. *Advances in Neural Information Processing Systems*, 35, 2022.

[ZM10] M. Zhu and S. Martínez. Discrete-time dynamic average consensus. *Automatica*, 46(2):322–329, 2010.

[ZP23] Y. Zhou and S. Pu. Private and accurate decentralized optimization via encrypted and structured functional perturbation. *IEEE Control Systems Letters*, 7:1339–1344, 2023.

[ZXG18] D. Zhou, P. Xu, and Q. Gu. Stochastic nested variance reduction for nonconvex optimization. *Advances in Neural Information Processing Systems*, 31:3921–3932, 2018.

[ZY19] J. Zhang and K. You. Decentralized stochastic gradient tracking for non-convex empirical risk minimization. *arXiv preprint arXiv:1909.02712*, 2019.

[ZZY+21] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K.-Y. Lam. Local differential privacy-based federated learning for Internet of Things. *IEEE Internet of Things Journal*, 8(11):8836–8853, June 2021.