

Distributed Optimization with Byzantine Robustness Guarantees

Présentée le 7 décembre 2023

Faculté informatique et communications
Laboratoire d'apprentissage automatique et d'optimisation
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

Lie HE

Acceptée sur proposition du jury

Prof. N. H. B. Flammarion, président du jury
Prof. M. Jaggi, directeur de thèse
Prof. D. Alistarh, rapporteur
Prof. P. Richtárik, rapporteur
Prof. R. Guerraoui, rapporteur

I would like to dedicate this thesis to my loving parents and wife. . .

Acknowledgements

I am profoundly indebted to my parents, whose bravery and relentless hard work have not only provided me with opportunities but also instilled in me the drive and persistence needed to pursue academic research. I am also immensely grateful to my beloved wife, whose unwavering support, patience, and love has been my cornerstone. I extend my deepest gratitude to Martin for his enduring patience, encouragement, and invaluable supervision over the years. Without his understanding and flexibility, I couldn't have balanced my academic responsibilities with family commitments.

I wish to express my heartfelt appreciation to my extraordinary collaborators: Praneeth, Yatao, Thijs, Anastasia, Tao, and Sebastian. Their diverse expertise and constructive feedback have not only enriched the quality and scope of my research but also made the process thoroughly enjoyable. Additionally, my sincere thanks go to Shiva, Ashish, and Zheng, with whom I had the honor of collaborating during my internships and external engagements.

Lastly, I would like to extend my thanks to my office mates, Aditya and El Mahdi, as well as other labmates and friends whom I have been fortunate enough to encounter. The memories we shared in this incredibly beautiful country are deeply cherished, and I will carry them with me along my path forward.

Abstract

As modern machine learning continues to achieve unprecedented benchmarks, the resource demands to train these advanced models grow drastically. This has led to a paradigm shift towards distributed training. However, the presence of adversaries—whether malicious or unintentional—complicates the training process. These attacks present notable security and performance challenges. This thesis primarily focuses on enhancing the Byzantine robustness in distributed machine learning. More precisely, we seek to enhance Byzantine robustness across varying conditions, including heterogeneous data, decentralized communication, and preserving input privacy. In this thesis, we formalize these problems and provide solutions backed by theoretical guarantees.

Apart from Byzantine robustness, we investigate alternative communication schemes in decentralized learning and methods for improving sample complexities in conditional stochastic optimization (CSO). In decentralized learning, gossip is predominantly the communication technique employed. However, it is susceptible to data heterogeneity and is slow to converge. We introduce a novel relay mechanism implemented over the spanning tree of the communication graph, offering independence of data heterogeneity. Lastly, in addressing the CSO problem, we observe that its stochastic gradient possesses inherent bias stemming from the nested structure of its objective. This bias contributes to an overhead in sample complexity. In this thesis, we enhance the sample complexity by deploying variance reduction and bias correction methods.

Keywords Distributed optimization, Byzantine robustness, decentralized learning, input privacy, bilevel optimization.

Résumé

Alors que l'apprentissage automatique moderne atteint constamment de nouveaux sommets, les ressources nécessaires pour entraîner ces modèles avancés s'accroissent considérablement. Cela a entraîné un changement vers l'entraînement distribué. Toutefois, la présence d'adversaires, intentionnels ou non, complexifie ce processus d'entraînement. Ces menaces posent d'importants défis en matière de sécurité et de performance. Cette thèse se focalise principalement sur l'amélioration de la robustesse face aux attaques byzantines dans le cadre de l'apprentissage automatique distribué. Plus spécifiquement, nous visons à renforcer cette robustesse dans divers contextes, tels que la présence de données hétérogènes, la communication décentralisée, et la protection de la confidentialité des données entrantes. Dans ce travail, nous formalisons ces problématiques et proposons des solutions soutenues par des garanties théoriques.

Par ailleurs, au-delà de la robustesse byzantine, nous explorons des schémas de communication alternatifs pour l'apprentissage décentralisé ainsi que des méthodes visant à optimiser la complexité de l'échantillonnage dans le cadre de l'optimisation stochastique conditionnelle (CSO). En matière d'apprentissage décentralisé, le "gossip" est généralement la méthode de communication privilégiée. Or, elle est sujette à des problématiques d'hétérogénéité des données et présente une convergence lente. Nous proposons donc un mécanisme de relais innovant, basé sur l'arbre couvrant du graphique de communication, qui pallie ces limitations. Enfin, concernant le problème du CSO, nous notons que son gradient stochastique est intrinsèquement biaisé à cause de la structure imbriquée de son objectif. Ce biais entraîne une augmentation de la complexité de l'échantillonnage. Ainsi, nous avons travaillé à améliorer cette complexité en employant des méthodes de réduction de la variance et de correction du biais.

Mots clés Optimisation distribuée, robustesse byzantine, apprentissage décentralisé, confidentialité des entrées, optimisation à deux niveaux.

Table of contents

1	Introduction	1
2	Byzantine-robust Learning on Heterogeneous Dataset via Bucketing	5
2.1	Preface	5
2.2	Introduction	6
2.3	Related work	8
2.4	Attacks against existing aggregation schemes	10
2.4.1	Failure on imbalanced data without Byzantine workers	10
2.4.2	Mimic attack on balanced data	11
2.5	Constructing an agnostic robust aggregator using bucketing	11
2.5.1	Bucketing algorithm	12
2.5.2	Agnostic robust aggregation	13
2.6	Robust non-iid optimization using a robust aggregator	14
2.6.1	Algorithm description	15
2.6.2	Convergence rates	15
2.6.3	Lower bounds and the challenge of heterogeneity	16
2.6.4	Circumventing lower bounds using overparameterization	17
2.7	Experiments	17
2.8	Conclusion	20
3	Byzantine-robust decentralized learning via ClippedGossip	21
3.1	Preface	21
3.2	Introduction	22
3.3	Related work	23
3.4	Setup	24
3.4.1	Decentralized threat model	24
3.4.2	Optimization assumptions	26
3.5	Robust Decentralized Consensus	26
3.5.1	The Clipped Gossip algorithm	26
3.5.2	Lower bounds due to communication constraints	28

3.6	Robust Decentralized Optimization	29
3.7	Experiments	32
3.7.1	Decentralized defenses without attackers	32
3.7.2	Decentralized learning under more attacks and topologies.	34
3.7.3	Lower bound of optimization	35
3.8	Discussion	35
4	Secure Byzantine-Robust Machine Learning	37
4.1	Preface	37
4.2	Introduction	38
4.3	Problem setup, privacy, and robustness	39
4.4	Secure aggregation protocol: two-server model	40
4.4.1	Non-robust secure aggregation	41
4.4.2	Robust secure aggregation	42
4.4.3	Salient features	43
4.5	Theoretical guarantees	44
4.5.1	Exactness	44
4.5.2	Privacy	45
4.5.3	Combining with differential privacy	46
4.6	Empirical analysis of overhead	47
4.7	Literature review	48
4.8	Conclusion	49
5	RelaySum for Decentralized Deep Learning on Heterogeneous Data	51
5.1	Preface	51
5.2	Introduction	52
5.3	Related work	54
5.4	Method	55
5.5	Theoretical analysis	57
5.6	Experimental analysis and practical properties	60
5.6.1	Effect of network topology	60
5.6.2	Spanning trees compared to other topologies	61
5.6.3	Effect of data heterogeneity in decentralized deep learning	61
5.6.4	Robustness to unreliable communication	62
5.7	Conclusion	65
6	Debiasing Conditional Stochastic Optimization	67
6.1	Preface	67
6.2	Introduction	67
6.3	Stochastic Extrapolation as a Tool for Bias Correction	71

6.4	Applying Stochastic Extrapolation in the CSO Problem	74
6.5	Applying Stochastic Extrapolation in the FCCO Problem	77
6.6	Applications	79
6.7	Concluding Remarks	80
7	Conclusion and Future Work	83
Appendix A Byzantine-robust Learning on Heterogeneous Dataset via Bucket-		
	ing	85
A.1	Experiment setup and additional experiments	85
A.1.1	Experiment setup	85
A.1.2	Additional experiments	87
A.2	Implementing the mimic attack	91
A.3	Constructing a robust aggregator using bucketing	92
A.3.1	Supporting lemmas	92
A.3.2	Proofs of robustness	94
A.4	Lower bounds on non-iid data (Proof of Theorem 2.3)	98
A.5	Convergence of robust optimization on non-iid data (Theorems 2.2 and 2.4) . .	99
Appendix B Byzantine-robust decentralized learning via ClippedGossip		107
B.1	Existing robust aggregators	107
B.2	Byzantine attacks in the decentralized environment	108
B.2.1	Existing attacks in federated learning	108
B.2.2	Dissensus attack and other attacks in the decentralized environment . .	108
B.3	Topologies and mixing matrices	110
B.3.1	Constrained topologies	110
B.3.2	Constructing mixing matrices	111
B.4	Experiments	112
B.4.1	Byzantine-robust consensus	112
B.4.2	Byzantine-robust decentralized optimization	113
B.4.3	Experiment: CIFAR-10 task	117
B.4.4	Experiment for “Weaker topology assumption”	117
B.4.5	Experiment: choosing clipping radius	118
B.5	Analysis	120
B.5.1	Definitions, and inequalities	120
B.5.2	Lemmas	122
B.5.3	Proof of the main theorem	132
B.6	Other related works and discussions	137

Appendix C Secure Byzantine-Robust Machine Learning	141
C.1 Proofs	141
C.2 Notes on security	144
C.2.1 Beaver’s MPC Protocol	144
C.2.2 Notes on obtaining a secret share	145
C.2.3 Computational indistinguishability	145
C.2.4 Notes on the security of S2	146
C.3 Data ownership diagram	147
C.4 Example: Two-server protocol with ByzantineSGD oracle	148
C.5 Additional experiments	150
Appendix D RelaySum for Decentralized Deep Learning on Heterogeneous Data	153
D.1 Convergence Analysis of RelaySGD	153
D.1.1 Notation	153
D.1.2 Technical Preliminaries	154
D.1.3 Results of Theorem 5.1	161
D.1.4 Proof of Theorem 5.1 in the convex case	161
D.1.5 Proof of Theorem 5.1 in the strongly convex case	170
D.1.6 Proof of Theorem 5.1 in the non-convex case	171
D.2 Detailed experimental setup	178
D.2.1 Cifar-10	178
D.2.2 ImageNet	178
D.2.3 BERT finetuning	178
D.2.4 Random quadratics	178
D.3 Hyper-parameters and tuning details	180
D.3.1 Cifar-10	180
D.3.2 ImageNet	180
D.3.3 BERT finetuning	181
D.3.4 Random quadratics	181
D.4 Algorithmic details	182
D.4.1 Learning-rate correction for RelaySGD	182
D.4.2 RelaySGD with momentum	183
D.4.3 RelaySGD with Adam	183
D.4.4 D^2 with momentum	183
D.4.5 Gradient Tracking	184
D.4.6 Stochastic Gradient Push with the time-varying exponential topology	184
D.5 Additional experiments on RelaySGD	184
D.5.1 Rings vs double binary trees on Cifar-10	184

D.5.2	Scaling the number of workers on Cifar-10	185
D.5.3	Independence of heterogeneity	186
D.5.4	Star topology	186
D.6	RelaySum for distributed mean estimation	187
D.7	Alternative optimizer based on RelaySum	188
D.7.1	Theoretical analysis of RelaySGD/Grad	189
D.7.2	Empirical analysis of RelaySGD/Grad	194
Appendix E	Debiasing Conditional Stochastic Optimization	197
E.1	Missing Pseudocodes	197
E.2	Missing Details from § 6.2	197
E.2.1	Other Related Work	197
E.3	Missing Details from § 6.3	199
E.4	Stationary Point Convergence Proofs from § 6.4 (CSO)	205
E.4.1	Helpful Lemmas	205
E.4.2	Convergence of BSGD	208
E.4.3	Convergence of E-BSGD	211
E.4.4	Convergence of BSpiderBoost	213
E.4.5	Convergence of E-BSpiderBoost	218
E.5	Stationary Point Convergence Proofs from § 6.5 (FCCO)	220
E.5.1	E-BSpiderBoost for FCCO problem	220
E.5.2	Convergence of NestedVR	222
E.5.3	Convergence of E-NestedVR	235
E.6	Missing Details from Section 2.7	241
E.6.1	Application of First-order MAML	241
E.6.2	Application of Deep Average Precision Maximization	242
E.6.3	Necessity of Additional Smoothness Conditions	243
References		245
Curriculum Vitae		273

Chapter 1

Introduction

With the growing size of data and complexity of model, modern machine learning has gradually reached and surpassed human-level performance across many applications [He et al., 2015]. Training such models requires an enormous amount of time and computational resources. For example, the training of GPT-3, a language model with 175 billion parameters, costs 355 years of GPU time [Brown et al., 2020]. This computational burden calls for distributed training, where workers collaboratively compute and share updates. Crowdsourced datasets, being naturally distributed across many clients, bring their own set of challenges, especially when they contain sensitive information that needs to be kept private. Federated learning [Bonawitz et al., 2019; Kairouz et al., 2019; McMahan et al., 2017a] provides a solution by promoting collaboration while ensuring data remains localized. Nonetheless, in many scenarios, it is not feasible to assume all participants will act honestly or adhere to protocols.

In distributed training, participants can inadvertently or maliciously harm performance. For instance, a malicious worker in federated learning might send a very large gradient to the server, causing the averaged vector to deviate significantly from the optimal and potentially making the model diverge. Beyond malicious intent, hardware failures present challenges too; bits in memory might randomly flip, leading to gradients changing signs. In crowdsourced datasets, even human experts can mislabel data, impacting the quality of the gradients. These adversaries are inherent in the distributed training process, and complicating the issue, one cannot simply exclude these adversaries since their identities remain unknown. A systematic approach is essential to address these challenges.

More precisely, these attacks can be characterized as *Byzantine*, marked by two defining characteristics: the ability to deviate arbitrarily from established protocols and send arbitrary messages [Pease et al., 1980a]. An ideal defense mechanism should be robust enough to counteract any attack fitting this definition. Moreover, it should still benefit from collaborative training. However, defending against Byzantine attacks is intricate and becomes even more so under certain conditions. For example, when regular workers have heterogeneous data, the server finds it challenging to differentiate between a Byzantine worker and a regular worker outlier. In

decentralized training, where a central server is absent and regular workers communicate via a defined communication topology, a Byzantine worker’s influence can markedly affect convergence, with the degree of disruption often being contingent on the chosen topology. Ensuring Byzantine robustness, particularly with theoretical guarantees, presents a substantial challenge.

Another potential risk in federated learning is that participants might access the privacy-sensitive data of regular clients. While clients would like to benefit from collaborative learning, they may not entirely trust the server. This mistrust is not unfounded; servers have the capability to infer data from plaintext gradients [Zhu et al., 2019]. Given this backdrop, the ideal approach would have servers aggregate gradients without directly interfacing with them, a notion in line with the tenets of secure multiparty computation (SMPC). While Bonawitz et al. [2017] have implemented a secure aggregation protocol that bolsters input privacy within federated learning, the protocol’s functionality is primarily confined to computing the gradient’s mean which doesn’t offer robustness. As a result, finding a defense compatible with SMPC protocols continues to be a complex endeavor.

In this thesis, our primary focus is on the Byzantine robustness of distributed machine learning. We endeavor to enhance robustness across various scenarios—ranging from heterogeneous data and decentralized environments to considerations of privacy—all while furnishing theoretical guarantees. In addition to Byzantine robustness, our research extends to two distinct optimization challenges, which we aim to address:

1. **Data Heterogeneity in Decentralized Learning:** In decentralized learning settings, workers exchange model updates solely with neighboring workers, typically using a method called gossip averaging. If the workers do not share same local stationary points, they do not converge even when the starting point is a stationary point of the global objective. This divergence, induced by data heterogeneity, compromises the convergence of gossip averaging. Our focus, therefore, lies in developing alternative communication mechanisms that are robust to the heterogeneity in data distribution across workers.
2. **Bias in Conditional Stochastic Optimization Problems:** In stochastic optimization, the objective may involve two nested layers of randomness. One layer depends conditionally on the other; for instance, in first-order model agnostic meta learning (MAML) [Finn et al., 2017] a random task set is first selected, followed by random samples conditional on the chosen tasks. Identifying stationary points in such landscapes is challenging due to the biased nature of the stochastic gradients. To mitigate this, additional iterations or samples are commonly required to achieve a desired level of precision. Our research aims to identify methods that can effectively reduce this bias and improve sample complexity.

Outline of the thesis

Chapter 2 studies Byzantine robustness of federated learning in the presence of heterogeneous data distribution. To address this setting, we introduce a bucketing scheme that seamlessly adapts existing robust algorithms to heterogeneous datasets with negligible computational overhead. Both theoretical and experimental results demonstrate the effectiveness of coupling our bucketing strategy with established robust algorithms, particularly against challenging attacks. Moreover, our research underscores the advantages of leveraging over-parameterized models in tandem with robust aggregation rules for enhanced heterogeneous Byzantine robust optimization.

Chapter 3 delves into the Byzantine robustness within decentralized learning environments. A primary observation from our studies indicates that poorly connected communication topologies can significantly amplify the detrimental effects of malicious actors. In response to this challenge, we introduce CLIPPEDGOSSIP, an innovative algorithm designed to withstand Byzantine attacks when the communication network maintains a reasonable level of connectivity. Notably, our research establishes that in certain extreme scenarios, it's impossible for any algorithm to guarantee robustness. Additionally, we offer a strategic approach to enhance the robustness of decentralized learning.

Chapter 4 explores defenses against both Byzantine and privacy adversaries. To address this dual challenge, we present a multi-server based secure aggregation framework. This multi-server system can leverage secret-sharing based SMPC protocols to implement robust aggregation functions. It is thus capable of withstanding Byzantine attacks and honest-but-curious privacy attacks. The performance of model remain same as non-private counterpart.

In Chapter 5 and Chapter 6, we pivot away from Byzantine robustness. Chapter 5 addresses the issue of enhancing communication efficiency for decentralized learning, particularly when faced with heterogeneous data. We propose RelaySGD, a novel algorithm that relays models through spanning trees of a network without decaying their magnitude. This algorithm is not only theoretically independent of data heterogeneity, but also high performing in deep learning tasks.

In Chapter 6, we tackle the challenge of improving the sample complexity associated with the conditional stochastic optimization (CSO) problem. The CSO problem is a generalized bilevel optimization problem where the inner random variables conditioned on the outer random variables. The CSO problem covers a wide range of applications, including instrumental variable regression, first order MAML, etc. A unique challenge arises from its nested structure, which results in a biased stochastic gradient, thereby increasing the sample complexities. In this chapter, we first identify the source of the bias and then use variance reduction and bias-correction methods to improve the sample complexity. We also extend our results to address the finite-sum variant of CSO problem.

Chapter 2

Byzantine-robust Learning on Heterogeneous Dataset via Bucketing

2.1 Preface

Contribution and sources. This chapter reproduces [Karimireddy et al., 2020a]. The author conducted most of the experiments and came up with the initial idea for using bucketing. Detailed individual contributions:

- Lie He (author): Conceptualization (50%), Software, Writing (original draft preparation 30 %)
- Sai Praneeth Karimireddy (co-first author): Conceptualization (50%), Methodology, Formal analysis, Writing (original draft preparation 70 %)
- Martin Jaggi: Supervision, Administration, Writing (review and editing).

Summary. Algorithms for Byzantine robust distributed or federated learning typically assume that the workers are identical. In such a case, using worker momentum is sufficient to reduce the variance, and hence the inter-worker heterogeneity. However, in most real world settings the workers data is heterogeneous (non-iid).

In this chapter, we will see how to design new attacks in such settings which circumvent current defenses and lead to significant loss of performance. We then propose a simple bucketing scheme that adapts existing robust algorithms to heterogeneous datasets at a negligible computational cost. We demonstrate (theoretically and experimentally) that combining bucketing with existing robust algorithms is effective against challenging attacks. Our work also shows that having over-parameterized models, when combined with robust aggregation rules, is very beneficial for heterogeneous Byzantine robust optimization. The code is available at <https://github.com/epfml/byzantine-robust-noniid-optimizer>.

2.2 Introduction

Distributed or federated machine learning, where the data is distributed across multiple workers, has become an increasingly important learning paradigm both due to growing sizes of datasets, as well as data privacy concerns. In such a setting, the workers collaborate to train a single model without directly transmitting their training data [Bonawitz et al., 2019; Kairouz et al., 2019; McMahan et al., 2017a]. However, by decentralizing the training across a vast number of workers we potentially open ourselves to new security threats. Due to the presence of agents in the network which are actively malicious, or simply due to system and network failures, some workers may disobey the protocols and send arbitrary messages; such workers are also known as *Byzantine* workers [Lamport et al., 2019]. Byzantine robust optimization algorithms attempt to combine the updates received from the workers using robust aggregation rules and ensure that the training is not impacted by the presence of a small number of malicious workers.

While this problem has received significant recent attention due to its importance, [Alistarh et al., 2018; Blanchard et al., 2017; Karimireddy et al., 2021b; Yin et al., 2018b], most of the current approaches assume that the data present on each different worker has identical distribution. This assumption is very unrealistic in practice and heterogeneity is inherent in distributed and federated learning [Kairouz et al., 2019]. In this work, we show that existing Byzantine aggregation rules catastrophically fail with very simple attacks (or sometimes even with no attacks) in realistic settings. We carefully examine the causes of these failures, and propose a simple solution which provably solves the Byzantine resilient optimization problem under heterogeneous workers.

Concretely, our contributions in this work are summarized below

- We show that when the data across workers is heterogeneous, existing aggregation rules fail to converge, even when no Byzantine adversaries are present. We also propose a simple new attack, *mimic*, which explicitly takes advantage of data heterogeneity and circumvents median-based defenses. Together, these highlight the fragility of existing methods in real world settings.
- We then propose a simple fix — a new bucketing step which can be used before any existing aggregation rule. We introduce a formal notion of a robust aggregator (ARAGG) and prove that existing methods like Krum, coordinate-wise median (CM), and geometric median aka robust federated averaging (RFA)—though insufficient on their own—become provably robust aggregators when augmented with our bucketing.
- We combine our notion of robust aggregator (ARAGG) with worker momentum to obtain optimal rates for Byzantine robust optimization with matching lower bounds. Unfortunately, our lower bounds imply that convergence to an exact optimum may not be possible due to heterogeneity. We then circumvent this lower bound and show that when heterogeneity is

mild (or when the model is overparameterized), we can in fact converge to an exact optimum. This is the first result establishing convergence to the optimum for heterogeneous Byzantine robust optimization.

- Finally, we evaluate the effect of the proposed techniques (bucketing and worker momentum) against known and new attacks showcasing drastic improvement on realistic heterogeneously distributed datasets.

Setup and notations. Consider a system comprising a single server and n workers. In each iteration, every worker retrieves the latest model from the server, computes its local gradients, and sends them back to the server synchronously. Subsequently, the server aggregates these gradients and updates the model.

Threat model. We assume the presence of Byzantine workers within our system, who may deviate from the designated protocol arbitrarily and transmit arbitrary messages [Allen-Zhu et al., 2021b; Chen et al., 2018, 2017a; Guerraoui et al., 2018; Rajput et al., 2019; Xie et al., 2019b; Yin et al., 2018a], aiming to undermine its performance. Although Byzantine workers have the capability to transmit vectors with different shapes or in an asynchronous manner, such vectors can be promptly detected and excluded. Consequently, our focus is directed towards Byzantine workers transmitting vectors identical in shape to regular ones and do so synchronously.

Remark 1. *Byzantine workers, equipped with system knowledge, can access defense strategies, data samples, communications between workers and servers, and observations of current and past random variables on regular workers. However, they cannot directly alter the states on regular workers, nor can they directly access the random seeds or future randomness on the regular workers. The gradients on regular workers are still unbiased.*

The set of good workers is denoted by $\mathcal{V}_R \subseteq \{1, \dots, n\}$. Our objective is to minimize

$$f(\mathbf{x}) := \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \{f_i(\mathbf{x}) := \mathbb{E}_{\xi_i}[F_i(\mathbf{x}; \xi_i)]\} \quad (2.1)$$

where f_i is the loss function on worker i defined over its own (heterogeneous) data distribution ξ_i .

The (stochastic) gradient computed by a good worker $i \in \mathcal{V}_R$ over minibatch ξ_i is given as $\mathbf{g}_i(\mathbf{x}, \xi_i) := \nabla F_i(\mathbf{x}; \xi_i)$. The noise in every stochastic gradient is independent, unbiased with $\mathbb{E}_{\xi_i}[\mathbf{g}_i(\mathbf{x}, \xi_i)] = \nabla f_i(\mathbf{x})$, and has bounded variance $\mathbb{E}_{\xi_i} \|\mathbf{g}_i(\mathbf{x}, \xi_i) - \nabla f_i(\mathbf{x})\|^2 \leq \sigma^2$. Further, we assume that the data heterogeneity across the workers can be bounded as

$$\mathbb{E}_{j \sim \mathcal{V}_R} \|\nabla f_j(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \zeta^2, \quad \forall \mathbf{x}.$$

We write \mathbf{g}_i^t or simply \mathbf{g}_i instead of $\mathbf{g}_i(\mathbf{x}^t, \xi_i^t)$ when there is no ambiguity.

The set of Byzantine workers $\mathcal{V}_B \subset [n]$ is fixed over time, with the remaining workers \mathcal{V}_R being good, i.e. $[n] = \mathcal{V}_B \uplus \mathcal{V}_R$. We write δ for the fraction of Byzantine workers, $|\mathcal{V}_B| =: q \leq \delta n$.

Our modeling assumes that the practitioner picks a value of $\delta \in [0, 0.5]$. This δ reflects the level of robustness required. A choice of a large δ (say near 0.5) would mean that the system is very robust and can tolerate a large fraction of attackers, but the algorithm becomes much more conservative and slow. On the flip side, if the practitioner knows that the number of Byzantine agents are going to be few, they can pick a small δ (say 0.05–0.1) ensuring some robustness with almost no impact on convergence. The choice of δ can also be formulated as how expensive do we want to make an attack? To carry out a successful attack the attacker would need to control δ fraction of all workers. We recommend implementations claiming robustness be transparent about their choice of δ .

2.3 Related work

IID defenses. There has been a significant amount of recent work on the case when all workers have identical data distributions. [Blanchard et al. \[2017\]](#) initiated the study of Byzantine robust learning and proposed a distance-based aggregation approach KRUM and extended to [\[Damaskinos et al., 2019; Mhamdi et al., 2018\]](#). [Yin et al. \[2018b\]](#) propose to use and analyze the coordinate-wise median (CM), and [Pillutla et al. \[2019\]](#) use approximate geometric median. [Bernstein et al. \[2019a\]](#) propose to use the signs of gradients and then aggregate them by majority vote, however, [Karimireddy et al. \[2019\]](#) show that it may fail to converge. Most recently, [Alistarh et al. \[2018\]](#); [Allen-Zhu et al. \[2021a\]](#); [Karimireddy et al. \[2021b\]](#); [Mhamdi et al. \[2021b\]](#) showcase how to use past gradients to more accurately filter iid Byzantine workers and specifically *time-coupled* attacks. In particular, our work builds on top of [\[Karimireddy et al., 2021b\]](#) and non-trivially extends to the non-iid setting.

IID vs. Non-IID attacks. For the iid setting, the state-of-the-art attacks are *time-coupled* attacks [\[Baruch et al., 2019; Xie et al., 2019a\]](#). These attacks introduce a small but consistent bias at every step which is hard to detect in any particular round, but accumulates over time and eventually leads to divergence, breaking most prior robust methods. Our work focuses on developing attacks (and defenses) which specifically take advantages of the non-iid setting. The non-iid setting also enables targeted *backdoor* attacks which are designed to take advantage of heavy-tailed data [\[Bagdasaryan et al., 2020a; Bhagoji et al., 2019\]](#). However, this is a challenging and open problem [\[Sun et al., 2019; Wang et al., 2020\]](#). Our focus is on the overall accuracy of the trained model, not on any subproblem.

Non-IID defenses. The non-iid defenses are relatively under-examined. [Ghosh et al. \[2019\]](#); [Sattler et al. \[2020\]](#) use an outlier-robust clustering method. When the server has the entire training dataset, the non-iid-ness is automatically addressed [\[Chen et al., 2018; Rajput et al., 2019; Xie et al., 2019c\]](#). Typical examples are parallel training of neural networks on public cloud, or volunteer computing [\[Meeds et al., 2015; Miura and Harada, 2015\]](#). Note that [Rajput et al. \[2019\]](#) use hierarchical aggregation over “vote group” which is similar to the bucketing techniques but their results are limited to the iid setting. However, none of these

methods are applicable to the standard federated learning. This is partially tackled in [Data and Diggavi, 2020, 2021b] who analyze spectral methods for robust optimization. However, these methods require $\Omega(d^2)$ time, making them infeasible for large scale optimization. Li et al. [2019] proposes an SGD variant (RSA) with additional ℓ_p penalty which only works for strongly convex objectives. In an independent recent work, Acharya et al. [2021] analyze geometric median (GM) on non-iid data using sparsified gradients. However, they do not defend against time coupled attacks, and their analysis neither proves convergence to the optimum nor recovers the standard rate of SGD when $\delta \rightarrow 0$. In contrast, our analysis of GM addresses both issues and is more general. For decentralized training with non-iid data, a parallel work [El-Mhamdi et al., 2021] considers asynchronous communication and unconstrained topologies and tolerates a maximum number of Byzantine workers in their setting. However, no convergence rate is given. He et al. [2022] consider decentralized training on constrained topologies and establish the consensus and convergence theory for a clipping based algorithm which tolerates a δ -fraction of Byzantine workers, limited by the spectral gap of the topology. Finally, Yang and Li [2021a] propose to use bucketing for asynchronous Byzantine learning which is very similar to the bucketing trick proposed in this paper for non-iid setup.¹

Strong growth condition. The assumption that

$$\mathbb{E}_{j \sim \mathcal{V}_R} \|\nabla f_j(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq B^2 \|\nabla f(\mathbf{x})\|^2$$

for some $B \geq 0$ is also referred to as the strong growth condition [Schmidt and Roux, 2013]. This has been extensively used to analyze and derive optimization algorithms for deep learning [Ma et al., 2018; Meng et al., 2020; Schmidt and Roux, 2013; Vaswani et al., 2019a,b]. This line of work shows that the strong growth assumption is both realistic and (perhaps more importantly) useful in understanding optimization algorithms in deep learning. However, this is stronger than the *weak* growth condition which states that $\mathbb{E}_{j \sim \mathcal{V}_R} \|\nabla f_j(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq B^2(f(\mathbf{x}) - f^*)$ for some $B \geq 0$. For a smooth function f , the strong growth condition always implies the weak growth condition. Further, for smooth convex functions this is equivalent to assuming that all the workers functions $\{f_i\}$ share a common optimum, commonly known as interpolation. Our work uses the stronger version of the growth condition and it remains open to extend our results to the weaker version. This latter condition is strictly necessary for heterogeneous Byzantine optimization [Gupta and Vaidya, 2020].

¹The previous version of this work uses resampling which has identical performance as bucketing. The detailed comparison is listed in § A.1.2.

2.4 Attacks against existing aggregation schemes

In this section we show that when the data across the workers is heterogeneous (non-iid), then we can design simple new attacks which take advantage of the heterogeneity, leading to the failure of existing aggregation schemes. We study three representative and widely used defenses:

Krum. For $i \neq j$, let $i \rightarrow j$ denote that \mathbf{x}_j belongs to the $n - q - 2$ closest vectors to \mathbf{x}_i . Then,

$$\text{KRUM}(\mathbf{x}_1, \dots, \mathbf{x}_n) := \arg \min_i \sum_{i \rightarrow j} \|\mathbf{x}_i - \mathbf{x}_j\|^2.$$

Krum is computationally expensive, requiring $\mathcal{O}(n^2)$ work by the server [Blanchard et al., 2017].

CM. Coordinate-wise median computes for the k th coordinate:

$$[\text{CM}(\mathbf{x}_1, \dots, \mathbf{x}_n)]_k := \text{median}([\mathbf{x}_1]_k, \dots, [\mathbf{x}_n]_k) = \arg \min_i \sum_{j=1}^n |[\mathbf{x}_i]_k - [\mathbf{x}_j]_k|.$$

Coordinate-wise median is fast to implement requiring only $\mathcal{O}(n)$ time [Chen et al., 2017b].

RFA. Robust federated averaging (RFA) computes the geometric median

$$\text{RFA}(\mathbf{x}_1, \dots, \mathbf{x}_n) := \arg \min_{\mathbf{v}} \sum_{i=1}^n \|\mathbf{v} - \mathbf{x}_i\|_2.$$

While the geometric median has no closed form solution, [Pillutla et al., 2019] approximate it using multiple iterations of smoothed Weiszfeld algorithm, each of which requires $\mathcal{O}(n)$ computation.

2.4.1 Failure on imbalanced data without Byzantine workers

We show that when the data amongst the workers is imbalanced, existing aggregation rules *fail* even in the *absence* of any Byzantine workers. Algorithms like KRUM select workers who are *representative* of a majority of the workers by relying on statistics such as pairwise differences between the various worker updates. Our key insight is that when the data across the workers is heterogeneous, there is no single worker who is representative of the whole dataset. This is because each worker computes their local gradient over vastly different local data.

Example. Suppose that there are $2n + 1$ workers with worker i holding $(-1)^i \in \{\pm 1\}$. This means that the true mean is ≈ 0 , but KRUM, CM, and RFA will output ± 1 . This motivates our next attack.

Hence, for convergence it is important to not only select a good (non-Byzantine) worker, but also ensure that each of the good workers is selected with roughly equal frequency. In Table 2.1, we demonstrate failures of such aggregators by training on MNIST with $n=20$ and no attackers ($\delta=0$). We construct an imbalanced dataset where each successive class has only a fraction of samples of the previous class. We defer details of the experiments to § A.1. As we can see, KRUM, CM and RFA match the ideal performance of SGD in the iid case, but only attain less

Table 2.1 Test accuracy (%) with no Byzantine workers ($\delta=0$) on imbalanced data.

Aggr	iid	non-iid
AVG	98.79 \pm 0.10	98.75 \pm 0.02
KRUM	97.95 \pm 0.25	89.90 \pm 4.75
CM	97.72 \pm 0.22	80.36 \pm 0.05
RFA	98.62 \pm 0.08	82.60 \pm 0.84
CCLIP	98.78 \pm 0.10	98.78 \pm 0.06

Table 2.2 Test accuracy (%) under mimic attack with $\delta = 0.2$ fraction of Byzantine workers.

Aggr	iid	non-iid
AVG	93.20 \pm 0.21	92.73 \pm 0.32
KRUM	90.36 \pm 0.25	37.33 \pm 6.78
CM	90.80 \pm 0.12	64.27 \pm 3.70
RFA	92.92 \pm 0.25	78.93 \pm 9.27
CCLIP	93.16 \pm 0.22	91.53 \pm 0.06

than 90% accuracy in the non-iid case. This corresponds to learning only the top 2–3 classes and ignoring the rest.

A similar phenomenon was observed when using batch-size 1 in the iid case by [Karimireddy et al., 2021b]. However, in the iid case this can be easily overcome by increasing the batch-size. In contrast, when the data across the works is non-iid (e.g. split by class), increasing the batch-size does *not* make the worker gradients any more similar and there remains a big drop in performance. Finally, note that in Table 2.1 a hitherto new algorithm (CCLIP) maintains its performance both in the iid and the non-iid setting. We will explore this in more detail in § 2.5.

2.4.2 Mimic attack on balanced data

Motivated by how data imbalance could lead to consistent errors in the aggregation rules and significant loss in accuracy, in this section, we will propose a new attack *mimic* which specifically tries to maximize the perceived data imbalance even if the original data is balanced.

Mimic attack. All Byzantine workers pick a good worker (say i_\star) to mimic and copy its output ($\mathbf{x}_{i_\star}^t$). This inserts a consistent bias towards over-emphasizing worker i_\star and thus under-representing other workers. Since the attacker simply mimics a good worker, it is impossible to distinguish it from a real worker and hence it cannot be filtered out. Indeed, the target i_\star can be any fixed good worker. In § A.2, we present an empirical rule to choose i_\star and include a simple example demonstrating how median based aggregators suffer from the heterogeneity under mimic attack.

Table 2.2 shows the effectiveness of mimic attack even when the fraction of Byzantine nodes is small (i.e. $n = 25$, $|\mathcal{V}_B| = 5$). Note that this attack specifically targets the non-iid nature of the data—all robust aggregators maintain their performance in the iid setting and only suffer in the non-iid setting. Their performance is in fact worse than even simply averaging. As predicted by our example, KRUM and CM have the worst performance and RFA performs slightly better. We will discuss the remarkable performance of CCLIP in the next section.

2.5 Constructing an agnostic robust aggregator using bucketing

In § 2.4 we demonstrated how existing aggregation rules fail in realistic non-iid scenarios, with and without attackers. In this section, we show how using bucketing can provably fix such

Algorithm 1 Robust Aggregation (ARAGG) using bucketing

-
- 1: **input** $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, $s \in \mathbb{N}$, aggregation rule AGGR
 - 2: pick random permutation π of $[n]$
 - 3: compute $\mathbf{y}_i \leftarrow \frac{1}{s} \sum_{k=(i-1) \cdot s + 1}^{\min(n, i \cdot s)} \mathbf{x}_{\pi(k)}$ for $i = \{1, \dots, \lceil n/s \rceil\}$
 - 4: **output** $\hat{\mathbf{x}} \leftarrow \text{AGGR}(\mathbf{y}_1, \dots, \mathbf{y}_{\lceil n/s \rceil})$ // aggregate after bucketing
-

aggregation rules. The underlying reason for this failure, as we saw previously, is that the existing methods fixate on the contribution of only the most likely worker, and ignore the contributions from the rest. To overcome this issue, we propose to use bucketing which ‘mixes’ the data from all the workers thereby reducing the chance of any subset of the data being consistently ignored.

2.5.1 Bucketing algorithm

Given n inputs $\mathbf{x}_1, \dots, \mathbf{x}_n$, we perform s -bucketing which randomly partitions them into $\lceil n/s \rceil$ buckets with each bucket having no more than s elements. Then, the contents of each bucket are averaged to construct $\{\mathbf{y}_1, \dots, \mathbf{y}_{\lceil n/s \rceil}\}$ which are then input to an aggregator AGGR. The details are summarized in Algorithm 1. The key property of our approach is that after bucketing, the resulting set of averaged $\{\mathbf{y}_1, \dots, \mathbf{y}_{\lceil n/s \rceil}\}$ are much more homogeneous (lower variance) than the original inputs. Thus, when fed into existing aggregation schemes, the chance of success increases. We formalize this in the following simple lemma.

Lemma 2.2 (Bucketing reduces variance). *Suppose we are given n independent (but not identical) random vectors $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ such that a good subset $\mathcal{V}_R \subseteq [n]$ of size at least $|\mathcal{V}_R| \geq n(1 - \delta)$ satisfies:*

$$\mathbb{E}\|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq \rho^2, \quad \text{for any fixed } i, j \in \mathcal{V}_R.$$

Define $\bar{\mathbf{x}} := \frac{1}{|\mathcal{V}_R|} \sum_{j \in \mathcal{V}_R} \mathbf{x}_j$. Let the outputs after s -bucketing be $\{\mathbf{y}_1, \dots, \mathbf{y}_{\lceil n/s \rceil}\}$ and denote $\tilde{\mathcal{V}}_R \subseteq \{1, \dots, \lceil n/s \rceil\}$ as a good bucket set where a good bucket contains only elements belonging to \mathcal{V}_R . Then $|\tilde{\mathcal{V}}_R| \geq \lceil n/s \rceil(1 - \delta s)$ satisfies

$$\mathbb{E}[\mathbf{y}_i] = \mathbb{E}[\bar{\mathbf{x}}] \quad \text{and} \quad \mathbb{E}\|\mathbf{y}_i - \mathbf{y}_j\| \leq \rho^2/s \quad \text{for any fixed } i, j \in \tilde{\mathcal{V}}_R.$$

The expectation in the above lemma is taken both over the random vectors as well as over the randomness of the bucketing procedure.

Remark 3. Lemma 2.2 proves that after our bucketing procedure, we are left with outputs \mathbf{y}_i which have i) pairwise variance reduced by s , and ii) potentially s times more fraction of Byzantine vectors. Hence, bucketing trades off increasing influence of Byzantine inputs against having more homogeneous vectors. Using $s = 1$ simply shuffles the inputs and leaves them otherwise unchanged.

2.5.2 Agnostic robust aggregation

We now define what it means for an agnostic robust aggregator to succeed.

Definition 2.1 ((δ_{\max}, c) -ARAGG). *Suppose we are given input $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ of which a subset \mathcal{V}_R of size at least $|\mathcal{V}_R| > (1 - \delta)n$ for $\delta \leq \delta_{\max} < 0.5$ and satisfies $\mathbb{E}\|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq \rho^2$. Then, the output $\hat{\mathbf{x}}$ of a Byzantine robust aggregator satisfies:*

$$\mathbb{E}\|\hat{\mathbf{x}} - \bar{\mathbf{x}}\|^2 \leq c\delta\rho^2 \quad \text{where} \quad \hat{\mathbf{x}} = \text{ARAGG}_\delta(\mathbf{x}_1, \dots, \mathbf{x}_n).$$

Further, ARAGG does not need to know ρ^2 (only δ), and automatically adapts to any value ρ^2 .

Our robust aggregator is parameterized by δ_{\max} , denoting the maximum fraction of Byzantine inputs it can tolerate. This threshold is bounded by the optimal breakdown point of 0.5 [Rousseeuw and Leroy, 2005]. The constant c governs the performance of the aggregator. Systems equipped with such robust aggregator satisfy the Byzantine agreement property [Fischer et al., 1986]: 1) *agreement*: all good workers agree on the aggregated $\hat{\mathbf{x}}$ dictated by the server; 2) *validity*: if all good workers have the same input ($\rho = 0$), then the output $\hat{\mathbf{x}} = \bar{\mathbf{x}}$ is the same as input. Moreover, if $\delta = 0$, i.e. when there are no Byzantine inputs, we are guaranteed to *exactly* recover the true average $\bar{\mathbf{x}}$. When both $\rho > 0$ and $\delta > 0$, we recover the average up to an additive error term. We also require that the robust aggregator is *agnostic* to the value of ρ^2 and automatically adjusts its output to the current ρ during training. The aggregator can take δ as an input though. This property is very useful in the context of Byzantine robust optimization since the variance ρ^2 keeps changing over the training period, whereas the fraction of Byzantine workers δ remains constant. This is a major difference from the definition used in [Karimireddy et al., 2021b]. Note that Definition 2.1 is defined for both homogeneous and heterogeneous data.

We next show that aggregators which we saw were not robust in § 2.4, can be made to satisfy Definition 2.1 by combining with bucketing.

Theorem 2.1. *Suppose we are given n inputs $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ satisfying properties in Lemma 2.2 for some $\delta \leq \delta_{\max}$, with δ_{\max} to be defined. Then, running Algorithm 1 with $s = \lfloor \delta_{\max}/\delta \rfloor$ yields the following:*

- *Krum*: $\mathbb{E}\|\text{KRUM} \circ \text{BUCKETING}(\mathbf{x}_1, \dots, \mathbf{x}_n) - \bar{\mathbf{x}}\|^2 \leq \mathcal{O}(\delta\rho^2)$ with $\delta_{\max} < 1/4$.
- *Geometric median*: $\mathbb{E}\|\text{RFA} \circ \text{BUCKETING}(\mathbf{x}_1, \dots, \mathbf{x}_n) - \bar{\mathbf{x}}\|^2 \leq \mathcal{O}(\delta\rho^2)$ with $\delta_{\max} < 1/2$.
- *Coordinate-wise median*: $\mathbb{E}\|\text{CM} \circ \text{BUCKETING}(\mathbf{x}_1, \dots, \mathbf{x}_n) - \bar{\mathbf{x}}\|^2 \leq \mathcal{O}(d\delta\rho^2)$ with $\delta_{\max} < 1/2$.

Note that all these methods satisfy our notion of an *agnostic* Byzantine robust aggregator (Definition 2.1). This is because both our bucketing procedures as well as the underlying aggregators are independent of ρ^2 . Further, our error is $\mathcal{O}(\delta\rho^2)$ and is information theoretically optimal, unlike previous analyses (e.g. Acharya et al. [2021]) who had an error of $\mathcal{O}(\rho^2)$.

The error of CM depends on the dimension d which is problematic when $d \gg n$. However, we suspect this is because we measure stochasticity using Euclidean norms instead of coordinate-wise. In practice, we found that CM often outperforms KRUM, with RFA outperforming them both. Note that we select $s = \lfloor \delta_{\max}/\delta \rfloor$ to ensure that after bucketing, we have the maximum amount of Byzantine inputs tolerated by the method with $(s\delta) = \delta_{\max}$.

Remark 4 (1-step Centered clipping). *The 1-step centered clipping aggregator (CCLIP) given a clipping radius τ and an initial guess \mathbf{v} of the average $\bar{\mathbf{x}}$ performs*

$$\text{CCLIP}(\mathbf{x}_1, \dots, \mathbf{x}_n) = \mathbf{v} + \frac{1}{n} \sum_{i \in [n]} (\mathbf{x}_i - \mathbf{v}) \min(1, \tau / \|\mathbf{x}_i - \mathbf{v}\|_2).$$

Karimireddy et al. [2021b] prove that CCLIP even without bucketing satisfies Definition 2.1 with $\delta_{\max} = 0.1$, and $c = \mathcal{O}(1)$. This explains its good performance on non-iid data in § 2.4. However, CCLIP is not agnostic since it requires clipping radius τ as an input which in turn depends on ρ^2 . Devising a version of CCLIP which automatically adapts its clipping radius is an important open question. Empirically however, we observe that simple rules for setting τ work quite well—we always use $\tau = \frac{10}{1-\beta}$ in our limited experiments where β is the coefficient of momentum.

While we have shown how to construct a robust aggregator which satisfies some notion of a robustness, we haven't yet seen how this affects the Byzantine robust *optimization* problem. We investigate this question theoretically in the next section and empirically in § 2.7.

2.6 Robust non-iid optimization using a robust aggregator

In this section, we study the problem of optimization in the presence of Byzantine workers and heterogeneity, given access to any robust aggregator satisfying Definition 2.1. We then show that data heterogeneity makes Byzantine robust optimization especially challenging and prove lower bounds for the same. Finally, we see how mild heterogeneity, or sufficient overparameterization can circumvent these lower bounds, obtaining convergence to the optimum.

Algorithm 2 Robust Optimization using any Agnostic Robust Aggregator

Require: ARAGG, η , β

- 1: **for** $t = 1, \dots$ **do**
 - 2: **for** worker $i \in [n]$ **in parallel**
 - 3: $\mathbf{g}_i \leftarrow \nabla F_i(\mathbf{x}, \boldsymbol{\xi}_i)$ and $\mathbf{m}_i \leftarrow (1 - \beta)\mathbf{g}_i + \beta\mathbf{m}_i$ ▷ worker momentum
 - 4: **send** \mathbf{m}_i if $i \in \mathcal{V}_R$, else send $*$ if Byzantine
 - 5: $\hat{\mathbf{m}} = \text{ARAGG}(\mathbf{m}_1, \dots, \mathbf{m}_n)$ and $\mathbf{x} \leftarrow \mathbf{x} - \eta\hat{\mathbf{m}}$. ▷ update params using robust aggregate
-

2.6.1 Algorithm description

In § 2.5 we saw that bucketing could tackle heterogeneity across the workers by reducing ζ^2 . However, there still remains variance σ^2 in the gradients within each worker since each worker uses stochastic gradients. To reduce the effect of this variance, we rely on worker momentum. Each worker sends their local worker momentum vector \mathbf{m}_i to be aggregated by ARAGG instead of \mathbf{g}_i :

$$\begin{aligned}\mathbf{m}_i^t &= \beta \mathbf{m}_i^{t-1} + (1 - \beta) \mathbf{g}_i(\mathbf{x}^{t-1}) \quad \text{for every } i \in \mathcal{V}_R, \\ \mathbf{x}^t &= \mathbf{x}^{t-1} - \eta \text{ARAGG}(\mathbf{m}_1^t, \dots, \mathbf{m}_n^t).\end{aligned}$$

This is equivalent to the usual momentum description up to a rescaling of step-size η . Intuitively, using worker momentum \mathbf{m}_i averages over $1/(1-\beta)$ independent stochastic gradients \mathbf{g}_i and thus reduces the effect of the within-worker-variance σ^2 [Karimireddy et al., 2021b]. Note that the resulting $\{\mathbf{m}_i\}$ are *still heterogeneous* across the workers. This heterogeneity is the key challenge we face.

2.6.2 Convergence rates

We now turn towards proving convergence rates for our bucketing aggregation method Algorithm 1 based on any existing aggregator AGGR. We will assume that for any fixed $i \in \mathcal{V}_R$

$$\mathbb{E}_{\xi_i} \|\mathbf{g}_i(\mathbf{x}) - \nabla f_i(\mathbf{x})\|^2 \leq \sigma^2 \text{ and } \mathbb{E}_{j \sim \mathcal{V}_R} \|\nabla f_j(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \zeta^2, \quad \forall \mathbf{x}. \quad (2.2)$$

This first condition bounds the variance of the stochastic gradient within a worker whereas the latter is a standard measure of inter-client heterogeneity in federated learning [Karimireddy et al., 2020b; Khaled et al., 2020; Yu et al., 2019]. Under these conditions, we can prove the following.

Theorem 2.2. *Suppose we are given a (δ_{\max}, c) -ARAGG satisfying Definition 2.1, and n workers of which a subset \mathcal{V}_R of size at least $|\mathcal{V}_R| \geq n(1 - \delta)$ faithfully follow the algorithm for $\delta \leq \delta_{\max}$. Further, for any good worker $i \in \mathcal{V}_R$ let f_i be a possibly non-convex function with L -Lipschitz gradients, and the stochastic gradients on each worker be independent, unbiased and satisfy (2.2). Then, for $F^0 := f(\mathbf{x}^0) - f^*$, the output of Algorithm 2 satisfies*

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \|\nabla f(\mathbf{x}^{t-1})\|^2 \leq \mathcal{O} \left(c\delta\zeta^2 + \sigma \sqrt{\frac{LF^0}{T}(c\delta + 1/n)} + \frac{LF^0}{T} \right).$$

Remark 5 (Unified proofs). Remark 4 shows that CCLIP is a robust aggregator, and Theorem 2.1 shows KRUM, RFA, and CM on combining with sufficient bucketing are all robust aggregators satisfying Definition 2.1. Most of these methods had no end-to-end convergence guarantees prior

to our results. Thus, Theorem 2.2 gives the first unified analysis in both the iid and non-iid settings.

When $\delta \rightarrow 0$ i.e. as we reduce the number of Byzantine workers, the above rate recovers the optimal $\mathcal{O}(\frac{\sigma}{\sqrt{Tn}})$ rate for non-convex SGD and even has linear speed-up with respect to the n workers. In contrast, all previous algorithms for non-iid data (e.g. [Acharya et al., 2021; Data and Diggavi, 2021b]) do not improve their rates for decreasing values of δ . This is also empirically reflected in § 2.4.1, where these algorithms are shown to fail even in the absence of Byzantine workers ($\delta = 0$).

Further, when $\zeta = 0$ the rate above simplifies to $\mathcal{O}(\frac{\sigma}{\sqrt{T}} \cdot \sqrt{c\delta + 1/n})$ which matches the iid Byzantine robust rates of [Karimireddy et al., 2021b]. In both cases we converge to the optimum and can make the gradient arbitrarily small. However, when $\delta > 0$ and $\zeta > 0$, Theorem 2.2 only shows convergence to a radius of $\mathcal{O}(\sqrt{\delta\zeta})$ and not to the actual optimum. We will next explore this limitation.

2.6.3 Lower bounds and the challenge of heterogeneity

Suppose worker j sends us an update which looks ‘weird’ and is very different from the updates from the rest of the workers. This may be because worker j might be malicious and their update represents an attempted attack. It may also be because worker j has highly *non-representative data*. In the former case the update should be ignored, whereas in the latter the update represents a valuable source of specialized data. However, it is impossible for the server to distinguish between the two situations. The above argument can in fact be formalized to prove the following lower bound.

Theorem 2.3. *Given any optimization algorithm ALG, we can find n functions $\{f_1(x), \dots, f_n(x)\}$ of which at least $(1 - \delta)n$ are good (belong to \mathcal{V}_R), 1-smooth, μ -strongly convex functions, and satisfy $\mathbb{E}_{i \sim \mathcal{V}_R} \|\nabla f_i(x) - \nabla f(x)\|^2 \leq \zeta^2$ such that the output of ALG has an error at least*

$$\mathbb{E}[f(\text{ALG}(f_1, \dots, f_n)) - f^*] \geq \Omega\left(\frac{\delta\zeta^2}{\mu}\right) \quad \text{and} \quad \mathbb{E}\|\nabla f(\text{ALG}(f_1, \dots, f_n))\|^2 \geq \Omega(\delta\zeta^2).$$

The expectation above is over the potential randomness of the algorithm. This theorem unfortunately implies that it is impossible to converge to the true optimum in the presence of Byzantine workers. Note that the above lower bound is information theoretic in nature and is independent of how many gradients are computed or how long the algorithm is run.

Remark 6 (Matches lower bound). *Suppose that we satisfy the heterogeneity condition (2.2) with $\zeta^2 > 0$ and $\sigma = 0$. Then, the rate in Theorem 2.2 can be simplified to $\mathcal{O}(\delta\zeta^2 + 1/T)$. While the second term in this decays to 0 with T , the first term remains, implying that we only converge to a radius of $\sqrt{\delta\zeta}$ around the optimum. However, this matches our lower bound result from Theorem 2.3 and hence is in general unimprovable.*

This is a very strong negative result and seems to indicate that Byzantine robustness might be impossible to achieve in real world federated learning. This would be major stumbling block for deployment since the system would provably be vulnerable to attackers. We will next carefully examine the lower bound and will attempt to circumvent it.

2.6.4 Circumventing lower bounds using overparameterization

We previously saw some strong impossibility results posed by heterogeneity. In this section, we show that while indeed in the worst case being robust under heterogeneity is impossible, we may still converge to the true optimum under more realistic settings. We consider an alternative bound of (2.2):

$$\mathbb{E}_{j \sim \mathcal{V}_R} \|\nabla f_j(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq B^2 \|\nabla f(\mathbf{x})\|^2, \quad \forall \mathbf{x}. \quad (2.3)$$

Note that at the optimum \mathbf{x}^* we have $\nabla f(\mathbf{x}^*) = 0$, and hence this assumption implies that $\nabla f_j(\mathbf{x}^*) = 0$ for all $j \in \mathcal{V}_R$. This is satisfied if the model is *sufficiently over-parameterized* and typically holds in most realistic settings [Vaswani et al., 2019a].

Theorem 2.4. *Suppose we are given a (δ_{\max}, c) -ARAGG and n workers with loss functions $\{f_1, \dots, f_n\}$ satisfying the conditions in Theorem 2.2 with $\delta \leq \delta_{\max}$ and (2.3) for some $B^2 < \frac{1}{60c\delta}$. Then, for $F^0 := f(\mathbf{x}^0) - f^*$, the output of Algorithm 2 satisfies*

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \|\nabla f(\mathbf{x}^{t-1})\|^2 \leq \mathcal{O} \left(\frac{1}{1-60c\delta B^2} \cdot \left(\sigma \sqrt{\frac{L F^0}{T}} (c\delta + 1/n) + \frac{L F^0}{T} \right) \right).$$

Remark 7 (Overparameterization fixes convergence). *The rate in Theorem 2.4 not only goes to 0 with T , but also matches that of the optimal iid rate of $\mathcal{O}(\frac{\sigma}{\sqrt{T}} \cdot \sqrt{c\delta + 1/n})$ [Karimireddy et al., 2021b]. Thus, using a stronger heterogeneity assumption allows us to circumvent lower bounds for the non-iid case and converge to a good solution even in the presence of Byzantine workers. This is the first result of its kind, and takes a major step towards realistic and practical robust algorithms.*

In the overparameterized setting, we can be sure that we will be able to *simultaneously* optimize all worker's losses. Hence, over time the agreement between all worker's gradients increases. This in turn makes any attempts by the attackers to derail training stand out easily, especially towards the end of the training. To take advantage of this increasing closeness, we need an aggregator which automatically adapts the quality of its output as the good workers get closer. Thus, the *agnostic* robust aggregator is crucial to our overparameterized convergence result. We empirically demonstrate the effects of overparameterization in § A.1.2.

2.7 Experiments

In this section, we demonstrate the effects of bucketing on datasets distributed in a non-iid fashion. Throughout the section, we illustrate the tasks, attacks, and defenses by an example of training

Table 2.3 Table 2.1 + Bucketing ($s=2$).

Aggr	iid	non-iid
AVG	98.80 \pm 0.10	98.74 \pm 0.02
KRUM	98.35 \pm 0.20	93.27 \pm 0.10
CM	98.26 \pm 0.22	95.59 \pm 0.89
RFA	98.75 \pm 0.14	97.34 \pm 0.58
CCLIP	98.79 \pm 0.10	98.75 \pm 0.02

Table 2.4 Table 2.2 + Bucketing ($s=2$).

Aggr	iid	non-iid
AVG	93.17 \pm 0.23	92.67 \pm 0.27
KRUM	91.64 \pm 0.30	53.15 \pm 3.96
CM	91.91 \pm 0.24	78.60 \pm 3.15
RFA	93.00 \pm 0.23	91.17 \pm 0.51
CCLIP	93.17 \pm 0.23	92.56 \pm 0.21

an MLP on a heterogeneous version of the MNIST dataset [LeCun et al., 1998]. The dataset is sorted by labels and sequentially divided into equal parts among good workers; Byzantine workers have access to the entire dataset. Implementations are based on PyTorch [Paszke et al., 2019] and will be made publicly available.² We defer details of setup, implementation, and runtime to § A.1.

Bucketing against the attacks on non-iid data. In § 2.4 we have presented how heterogeneous data can lead to failure of existing robust aggregation rules. Here we apply our proposed bucketing with $s=2$ to the same aggregation rules, showing that bucketing overcomes the described failures. Results are presented in Table 2.3. Comparing Table 2.3 with Table 2.1, bucketing improves the aggregators’ top-1 test accuracy on long-tail and non-iid dataset by 4% to 14% and allows them to learn classes at the tail distribution. For non-iid balanced dataset, bucketing also greatly improves the performance of KRUM and CM and makes RFA and CCLIP close to ideal performance. Similarly, combining aggregators with bucketing also performs much better on non-iid dataset under mimic attack. In Table 2.4, RFA and CCLIP recover iid accuracy, and KRUM, and CM are improved by around 15%.

Bucketing against general Byzantine attacks. In Figure 2.1, we present thorough experiments on non-iid data over 25 workers with 5 Byzantine workers, under different attacks. In each subfigure, we compare an aggregation rule with its variant with bucketing. The aggregation rules compared are KRUM, CM, RFA, CCLIP. 5 different kinds of attacks are applied (one per column in the figure): bit flipping (BF), label flipping (LF), *mimic* attack, as well as inner product manipulation (IPM) attack [Xie et al., 2019a] and the “a little is enough” (ALIE) attack [Baruch et al., 2019].

- **Bit flipping:** A Byzantine worker sends $-\nabla f(\mathbf{x})$ instead of $\nabla f(\mathbf{x})$ due to hardware failures etc.
- **Label flipping:** Corrupt MNIST dataset by transforming labels by $\mathcal{T}(y) := 9 - y$.
- **Mimic:** Explained in § 2.4.2.
- **IPM:** The attackers send $-\frac{\epsilon}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \nabla f(\mathbf{x}_i)$ where ϵ controls the strength of the attack.
- **ALIE:** The attackers estimate the mean $\mu_{\mathcal{V}_R}$ and standard deviation $\sigma_{\mathcal{V}_R}$ of the good gradients, and send $\mu_{\mathcal{V}_R} - z\sigma_{\mathcal{V}_R}$ to the server where z is a small constant controlling the strength of the attack.

²The code is available at [this url](#).

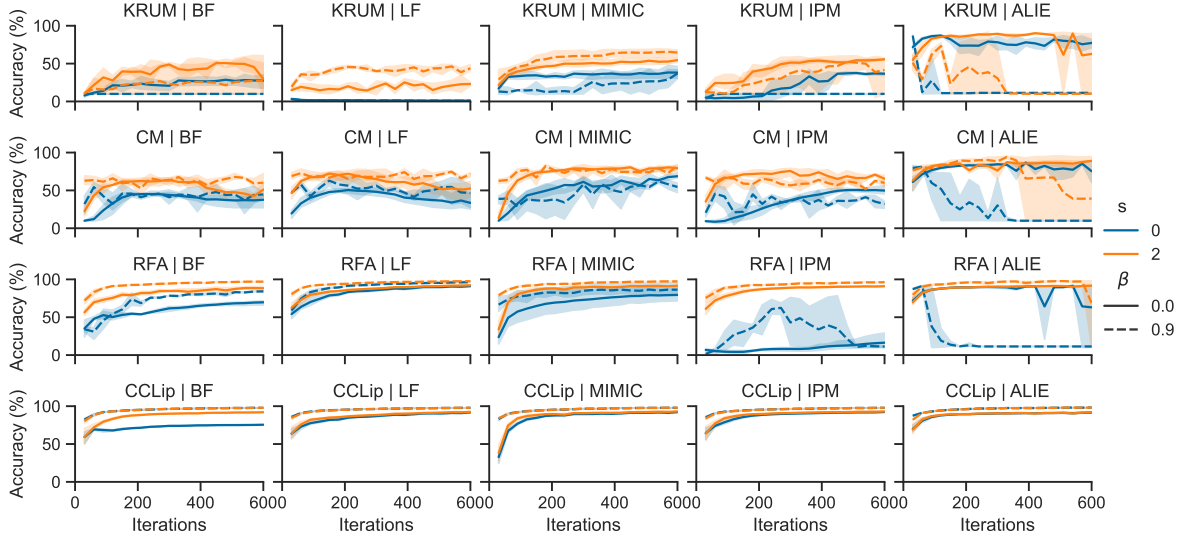


Fig. 2.1 Top-1 test accuracies of KRUM, CM, CCLIP, RFA, under 5 attacks on non-iid datasets.

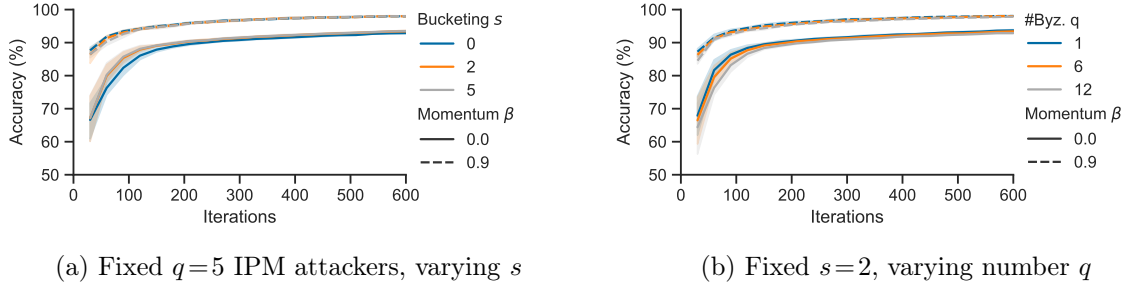


Fig. 2.2 Top-1 accuracies of CCLIP with varying q and s when training on a cluster of $n=53$ nodes.

Both IPM and ALIE are the state-of-the-art attacks in the iid distributed learning setups which takes advantage of the variances among workers. These attacks are much stronger in the non-iid setup. In the last two columns of Figure 2.1 we show that worker momentum and bucketing reduce such variance while momentum alone is not enough. Overall, Figure 2.1 shows that bucketing improves the performances of almost all aggregators under all kinds of attacks. Note that τ of CCLIP is not finetuned for each attack but rather fixed to $\frac{10}{1-\beta}$ for all attacks. This scaling is required because CCLIP is *not agnostic*. We defer the discussion to § A.1.2.

Bucketing hyperparameter. Finally we study the influence of s and q on the heterogeneous MNIST dataset. We use CCLIP as the base aggregator and apply IPM attack. The Figure 2.2a confirms that larger s gives faster convergence but $s=2$ is sufficient. Figure 2.2b shows that $s=2$ still behaves well when increasing q close to 25%. The complete evaluation of the results are deferred to § A.1.

Discussion. In all our experiments, we consistently observe: i) mild bucketing ($s = 2$) improves performance, ii) worker momentum further stabilizes training, and finally iii) CCLIP recovers the ideal performance. Given its ease of implementation, this leads us to strongly recommend using CCLIP in practical federated learning to safeguard against actively malicious agents or passive failures. RFA combined with bucketing and worker momentum also nearly recovers ideal performance and can instead be used when a proper radius τ is hard to find. Designing an *automatic* and *adaptively* clipping radius as well as its large scale empirical study is left for future work.

2.8 Conclusion

Heterogeneity poses unique challenges for Byzantine robust optimization. The first challenge is that existing defenses attempt to pick a “representative” update, which may not exist in the non-iid setting. This, we showed, can be overcome by using bucketing. A second more fundamental challenge is that it is difficult to distinguish between a “weird” but good worker from an actually Byzantine attacker. In fact, we proved strong impossibility results in such a setting. For this we showed how overparameterization (which is prevalent in real world deep learning) provides a solution, ensuring convergence to the optimum even in the presence of attackers. Together, our results yield a practical provably Byzantine robust algorithms for the non-iid setting.

Chapter 3

Byzantine-robust decentralized learning via ClippedGossip

3.1 Preface

Contribution and sources. This chapter reproduces the work presented in [He et al., 2022], which delves into the complexities of Byzantine attacks in communication-constrained graphs in decentralized scenarios. The authors collectively conceptualized the study, conducted the formal analysis, and drafted the manuscript. The individual contributions are as follows:

- Lie He: Conceptualization, Writing, Formal Analysis, Software.
- Sai Praneeth Karimireddy: Conceptualization, Writing, Formal Analysis.
- Martin Jaggi: Supervision, Administration, Writing (review and editing).

Summary. In decentralized environments where direct communication among workers is not feasible, Byzantine attacks present significant challenges in communication-constrained graphs. The convergence rate of decentralized algorithms can be notably influenced by the position and quantity of Byzantine workers in the communication graph. Prior studies have utilized the number of Byzantine workers as a robustness measure, which, however, inadequately characterizes such robustness. In this chapter, we introduce a novel network robustness criterion based on the spectral gap of the topology of regular workers, offering a more accurate characterization. To defend against these attacks, we propose CLIPPEDGOSSIP as a defensive strategy, providing precise rates of robust convergence to a neighborhood of a stationary point for the first time under standard assumptions. Our empirical results underline the superiority of CLIPPEDGOSSIP over previous methodologies across a range of networks. The code is accessible at <https://github.com/epfml/byzantine-robust-decentralized-optimizer>.

3.2 Introduction

“Divide et impera”.

Distributed training arises as an important topic due to privacy constraints of decentralized data storage [Kairouz et al., 2019; McMahan et al., 2017a]. As the server-worker paradigm suffers from a single point of failure, there is a growing amount of works on training in the absence of server [Koloskova et al., 2020b; Lian et al., 2017a; Nedic, 2020]. We are particularly interested in decentralized scenarios where direct communication may be unavailable due to physical constraints. For example, devices in a sensor network can only communicate devices within short physical distances.

Failures—from malfunctioning or even malicious participants—are ubiquitous in all kinds of distributed computing. We use the same Byzantine attacker definition as in Chapter 2, i.e., every *Byzantine* adversarial worker can deviate from the prescribed algorithm and send arbitrary messages [Lamport et al., 2019]. Note that these attackers cannot directly modify the states on regular workers, nor compromise messages sent between two connected regular workers.

Defending Byzantine attacks in a communication-constrained graph is challenging. As secure broadcast protocols are no longer available [Dolev and Strong, 1983; Hirt and Raykov, 2014; Pease et al., 1980b], regular workers can only utilize information from their own neighbors who have heterogeneous data distribution or are malicious, making it very difficult to reach global consensus. While there are some works attempt to solve this problem [Su and Vaidya, 2016a; Sundaram and Gharesifard, 2018], their strategies suffer from serious drawbacks: 1) they require regular workers to be very densely connected; 2) they only show asymptotic convergence or no convergence proof; 3) there is no evidence if their algorithms are better than training alone.

In this work, we study the Byzantine robustness decentralized training in a constrained topology and address the aforementioned issues. The main contributions of our paper are summarized as follows:

- We identify a novel network robustness criterion, characterized in terms of the spectral gap of the topology (γ) and the number of attackers (δ), for consensus and decentralized training, applying to a much broader spectrum of graphs than [Su and Vaidya, 2016a; Sundaram and Gharesifard, 2018].
- We propose CLIPPEDGOSSIP as the defense strategy and provide, for the first time, precise rates of robust convergence to a $\mathcal{O}(\delta_{\max}\zeta^2/\gamma^2)$ neighborhood of a stationary point for stochastic objectives under standard assumptions.¹ We also empirically demonstrate the advantages of CLIPPEDGOSSIP over previous works.
- Along the way, we also obtain the fastest convergence rates for standard non-robust (Byzantine-free) decentralized stochastic non-convex optimization by using local worker momentum.

¹In a previous version, we referred to CLIPPEDGOSSIP as *self-centered clipping*.

3.3 Related work

Recently there have been extensive works on Byzantine-resilient distributed learning with a trustworthy server. The statistics-based robust aggregation methods cover a wide spectrum of works including median [Blanchard et al., 2017; Chen et al., 2017c; Mhamdi et al., 2018; Xie et al., 2018a; Yin et al., 2018b, 2019], geometric median [Pillutla et al., 2019], signSGD [Bernstein et al., 2019b; Li et al., 2019; Sohn et al., 2020], clipping [Karimireddy et al., 2021a,c], and concentration filtering [Alistarh et al., 2018; Allen-Zhu et al., 2021a; Data and Diggavi, 2021a]. Other works explore special settings where the server owns the entire training dataset [Chen et al., 2018; Gupta et al., 2021; Rajput et al., 2019; Regatti et al., 2020; Su and Vaidya, 2016b; Xie et al., 2020a]. The state-of-the-art attacks take advantage of the variance of good gradients and accumulate bias over time [Baruch et al., 2019; Xie et al., 2019a]. A few strategies have been proposed to provably defend against such attacks, including momentum [Karimireddy et al., 2021a; Mhamdi et al., 2021a] and concentration filtering [Allen-Zhu et al., 2021b].

Decentralized machine learning has been extensively studied in the past few years [Koloskova et al., 2020b; Kong et al., 2021; Kovalev et al., 2021; Li et al., 2021; Lian et al., 2017a; Lin et al., 2021a; Ying et al., 2021b; Yuan et al., 2021]. The state-of-the-art convergence rate is established in [Koloskova et al., 2020b] is $\mathcal{O}(\frac{\sigma^2}{n\epsilon^2} + \frac{\sigma}{\sqrt{\gamma}\epsilon^{3/2}})$ where the leading $\frac{\sigma^2}{n\epsilon^2}$ is optimal. In this paper we improve this rate to $\mathcal{O}(\frac{\sigma^2}{n\epsilon^2} + \frac{\sigma^{2/3}}{\gamma^{2/3}\epsilon^{4/3}})$ using local momentum.

Decentralized machine learning with certified Byzantine robustness is less studied. When the communication is unconstrained, there exist secure broadcast protocols that guarantee all regular workers have identical copies of each other’s update [El-Mhamdi et al., 2021; Gorbunov et al., 2021]. We are interested in a more challenging scenario where not all workers have direct communication links. In this case, regular workers may behave very differently depending on their neighbors in the topology. One line of work constructs a Public-Key Infrastructure (PKI) so that the message from each worker can be authenticated using digital signatures. However, this is very inefficient requiring quadratic communication [Abraham et al., 2020]. Further, it also requires every worker to have a globally unique identifier which is known to every other worker. This assumption is rendered impossible on general communication graphs, motivating our work to explicitly address the graph topology in decentralized training. Sybil attacks are an important orthogonal issue where a single Byzantine node can create innumerable “fake nodes” overwhelming the network (cf. recent overview by Ford [2021]). Truly decentralized solutions to this are challenging and sometimes rely on heavy machinery, e.g. blockchains [Poupko et al., 2021] or Proof-of-Personhood [Borge et al., 2017].

More related to the approaches we study, Su and Vaidya [2016a]; Sundaram and Gharesifard [2018]; Yang and Bajwa [2019a,b] use trimmed mean at each worker to aggregate models of its neighbors. This approach only works when all regular workers have an honest majority among their neighbors and are densely connected. Guo et al. [2021] evaluate the incoming models of a good worker with its local samples and only keep those well-perform models for its local

update step. However, this method only works for IID data. Peng and Ling [2020] reformulate the original problem by adding TV-regularization and propose a GossipSGD type algorithm which works for strongly convex and non-IID objectives. However, its convergence guarantees are inferior to non-parallel SGD. In this work, we address all of the above issues and are able to provably relate the communication graph (spectral gap) with the fraction of Byzantine workers. Besides, most works do not consider attacks that exploit communication topology, except [Peng and Ling, 2020] who propose zero-sum attack. We defer detailed comparisons and more related works to § B.6.

3.4 Setup

3.4.1 Decentralized threat model

Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, \dots, n\}$ denotes the set of workers and \mathcal{E} denotes the set of edges. Let $\mathcal{N}_i \subset \mathcal{V}$ be the neighbors of node i and $\bar{\mathcal{N}}_i := \mathcal{N}_i \cup \{i\}$. In addition, we assume there are no self-loops and the system is synchronous. We consider the same notion of Byzantine workers as outlined in Chapter 2, i.e. they can deviate from the designated protocol arbitrarily and transmit arbitrary messages [Allen-Zhu et al., 2021b; Yin et al., 2018a]. Let $\mathcal{V}_B \subset \mathcal{V}$ be the set of Byzantine workers with $b = |\mathcal{V}_B|$ and the set of regular (non-Byzantine) workers is $\mathcal{V}_R := \mathcal{V} \setminus \mathcal{V}_B$. Let \mathcal{G}_R be the subgraph of \mathcal{G} induced by the regular nodes \mathcal{V}_R which means removing all Byzantine nodes and their associated edges. If the reduced graph \mathcal{G}_R is disconnected, then there exist two regular workers who cannot reliably exchange information. In this setting, training on the combined data of all the good workers is impossible. Hence, we make the following necessary assumption.

Assumption A (Connectivity). \mathcal{G}_R is connected.

Remark 1. In contrast, Su and Vaidya [2016a]; Sundaram and Gharesifard [2018] impose a much stronger assumption that the subgraph of \mathcal{G}_R of the regular workers remain connected even after additionally removing any $|\mathcal{V}_B|$ number of edges. For example, the graph in Fig. 3.1 with 1 Byzantine worker V_1 satisfies Assumption A but does not satisfy their assumption as removing an additional edge at A_1 or B_1 may discard the graph cut.

In decentralized learning, each regular worker $i \in \mathcal{V}_R$ locally stores a vector $\{\mathbf{W}_{ij}\}_{j=1}^n$ of mixing weights, for how to aggregate model updates received from neighbors. We make the following assumption on the weight vectors.

Assumption B (Mixing weights). The weight vectors on regular workers satisfy the following properties:

- Each regular worker $i \in \mathcal{V}_R$ stores non-negative $\{\mathbf{W}_{ij}\}_{j=1}^n$ with $\mathbf{W}_{ij} > 0$ iff $j \in \bar{\mathcal{N}}_i$;
- The adjacent weights to each regular worker $i \in \mathcal{V}_R$ sum up to 1, i.e. $\sum_{j=1}^n \mathbf{W}_{ij} = 1$;

- For $i, j \in \mathcal{V}_R$, $\mathbf{W}_{ij} = \mathbf{W}_{ji}$.

We can construct such weights even in the presence of Byzantine workers, using algorithms that only rely on communication with local neighbors, e.g. Metropolis-Hastings [Hastings, 1970]. We defer details of the construction to § B.3.2. Note that the Byzantine workers \mathcal{V}_B might also obtain such weights, however, they can use arbitrary different weights in reality during the training.

We define $\delta_i := \sum_{j \in \mathcal{V}_B} \mathbf{W}_{ij}$ to be the total weight of adjacent Byzantine edges around a regular worker i , and define the maximum Byzantine weight as $\delta_{\max} := \max_{i \in \mathcal{V}_R} \delta_i$.

Remark 2. *In the decentralized setting, the total fraction of Byzantine nodes $|\mathcal{V}_B|/n$ is irrelevant. Instead, what matters is the fraction of the edge weights they control which are adjacent to regular nodes (as defined by δ_i and δ_{\max}). This is because a Byzantine worker can send different messages along each edge. Thus, a single Byzantine worker connected to all other workers with large edge weights can have a large influence on all the other workers. Similarly, a potentially very large number of Byzantine workers may overall have very little effect—if the edges they control towards good nodes have little weight. When we have a uniform fully connected graph (such as in the centralized setting), the two notions of bad nodes & edges become equivalent.*

To facilitate our analysis of convergence rate, we define a *hypothetical* mixing matrix $\widetilde{\mathbf{W}} \in \mathbb{R}^{(n-b) \times (n-b)}$ for the subgraph \mathcal{G}_R of regular workers with entry $i, j \in \mathcal{V}_R$ defined as

$$\widetilde{\mathbf{W}}_{ij} = \begin{cases} \mathbf{W}_{ij} & \text{if } i \neq j \\ \mathbf{W}_{ii} + \delta_i & \text{if } i = j. \end{cases} \quad (3.1)$$

By the construction of this hypothetical matrix $\widetilde{\mathbf{W}}$, the following property directly follows.

Lemma 3.3. *Given Assumption B, then $\widetilde{\mathbf{W}}$ is symmetric and doubly stochastic, i.e.*

$$\widetilde{\mathbf{W}}_{ij} = \widetilde{\mathbf{W}}_{ji}, \quad \sum_{i=1}^n \widetilde{\mathbf{W}}_{ij} = 1, \quad \sum_{j=1}^n \widetilde{\mathbf{W}}_{ij} = 1. \quad \forall i, j \in [n-b]$$

Further, the spectral gap of the matrix $\widetilde{\mathbf{W}}$ is positive.

Lemma 3.4. *By Assumption A and Assumption B, there exists $\gamma \in (0, 1]$ such that $\forall \mathbf{x} \in \mathbb{R}^{n-b}$ and $\bar{\mathbf{x}} = \frac{\mathbf{1}^\top \mathbf{x}}{n-b} \mathbf{1} \in \mathbb{R}^{n-b}$*

$$\|\widetilde{\mathbf{W}}\mathbf{x} - \bar{\mathbf{x}}\|_2 \leq (1 - \gamma)\|\mathbf{x} - \bar{\mathbf{x}}\|_2. \quad (3.2)$$

The $\gamma(\widetilde{\mathbf{W}})$ is the *spectral gap* of the subgraph of regular workers \mathcal{G}_R . We have $\gamma = 0$ if and only if \mathcal{G}_R is disconnected, and $\gamma = 1$ if and only if \mathcal{G}_R is fully connected.

In summary, γ measures the connectivity of the regular subgraph \mathcal{G}_R formed after removing the Byzantine nodes, whereas δ_i and δ_{\max} are a measure of the influence of the Byzantine nodes.

3.4.2 Optimization assumptions

We study the general distributed optimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}) := \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \{f_i(\mathbf{x}) := \mathbb{E}_{\xi_i \sim \mathcal{D}_i} F_i(\mathbf{x}; \xi_i)\} \quad (3.3)$$

on heterogeneous (non-IID) data, where f_i is the local objective on worker i with data distribution \mathcal{D}_i and independent noise ξ_i . We assume that the gradients computed over these data distributions satisfy the following standard properties.

Assumption C (Bounded noise and heterogeneity). *Assume that for all $i \in \mathcal{V}_R$ and $\mathbf{x} \in \mathbb{R}^d$, we have*

$$\mathbb{E}_{\xi \sim \mathcal{D}_i} \|\nabla F_i(\mathbf{x}; \xi) - \nabla f_i(\mathbf{x})\|^2 \leq \sigma^2, \quad \mathbb{E}_{j \sim \mathcal{V}_R} \|\nabla f_j(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \zeta^2. \quad (3.4)$$

Assumption D (L-smoothness). *For $i \in \mathcal{V}_R$, $f_i(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$ is differentiable and there exists a constant $L \geq 0$ such that for each $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$:*

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|. \quad (3.5)$$

We denote $\mathbf{x}_i^t \in \mathbb{R}^d$ as the state of worker $i \in \mathcal{V}_R$ at time t .

3.5 Robust Decentralized Consensus

Agreeing on one value (*consensus*) among regular workers is one of the fundamental questions in distributed computing. *Gossip averaging* is a common consensus algorithm in the Byzantine-free case ($\delta = 0$). Applying gossip averaging steps iteratively to all nodes formally writes as

$$\mathbf{x}_i^{t+1} := \sum_{j=1}^n \mathbf{W}_{ij} \mathbf{x}_j^t, \quad t = 0, 1, \dots \quad (\text{GOSSIP})$$

Suppose each worker $i \in [n]$ initially owns a different \mathbf{x}_i^0 and Assumption A and Assumption B hold true, then each worker's iterate \mathbf{x}_i^t asymptotically converges to $\mathbf{x}_i^\infty = \bar{\mathbf{x}} = \frac{1}{n} \sum_{j=1}^n \mathbf{x}_j^0$, for all $i \in [n]$, which is also known as average consensus [Boyd et al., 2006]. Reaching consensus in the presence of Byzantine workers is more challenging, with a long history of study [LeBlanc et al., 2013; Su and Vaidya, 2016a].

3.5.1 The Clipped Gossip algorithm

We introduce a novel decentralized gossip-based aggregator, termed CLIPPEDGOSSIP, for Byzantine-robust consensus. CLIPPEDGOSSIP uses its local reference model as center and clips all received neighbor model weights. Formally, for $\text{CLIP}(\mathbf{z}, \tau) := \min(1, \tau/\|\mathbf{z}\|) \cdot \mathbf{z}$, we define for

node i

$$\mathbf{x}_i^{t+1} := \sum_{j=1}^n \mathbf{W}_{ij}(\mathbf{x}_i^t + \text{CLIP}(\mathbf{x}_j^t - \mathbf{x}_i^t, \tau_i)), \quad t = 0, 1, \dots \quad (\text{CLIPPEDGOSSIP})$$

Theorem 3.1. *Let $\bar{\mathbf{x}}^t := \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbf{x}_i^t$ be the average iterate over the unknown set of regular nodes. If the initial consensus distance is bounded as $\frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \|\mathbf{x}_i^t - \bar{\mathbf{x}}^t\|^2 \leq \rho^2$, then for all $i \in \mathcal{V}_R$, the output \mathbf{x}_i^{t+1} of CLIPPEDGOSSIP with an appropriate choice of clipping radius satisfies*

$$\frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \|\mathbf{x}_i^{t+1} - \bar{\mathbf{x}}^{t+1}\|^2 \leq (1 - \gamma + c\sqrt{\delta_{\max}})^2 \rho^2 \quad \text{and} \quad \mathbb{E} \|\bar{\mathbf{x}}^{t+1} - \bar{\mathbf{x}}^t\|^2 \leq c^2 \delta_{\max} \rho^2$$

where the expectation is over the random variable $\{\mathbf{x}_i^t\}_{i \in \mathcal{V}_R}$ and $c > 0$ is a constant.

If the information propagation among regular workers is faster than among Byzantine workers ($\gamma > c\sqrt{\delta_{\max}}$), then our algorithm can achieve *approximate Byzantine consensus* [Dolev et al., 1986]. The *agreement* property is upheld as the upper bound of consensus distance diminishes exponentially over time, eventually bringing all regular workers within ϵ of each other. The *validity* condition is met because when regular workers attain consensus prior to aggregation ($\rho = 0$), our algorithm ensures that consensus is maintained.

We inspect Theorem 3.1 on corner cases. In this case, we can use a simple majority, which corresponds to setting clipping threshold $\tau_i = 0$. Further, if there is no Byzantine worker ($\delta_{\max} = 0$), then the robust aggregator must improve the consensus distance by a factor of $(1 - \gamma)^2$ which matches standard gossiping analysis [Boyd et al., 2006]. Finally, for the complete graph ($\gamma = 1$) CLIPPEDGOSSIP satisfies the centralized notion of (δ_{\max}, c^2) -robust aggregator in [Karimireddy et al., 2021a, Definition C]. Thus, CLIPPEDGOSSIP recovers all past optimal aggregation methods as special cases.

Note that if the topology is poorly connected and there are Byzantine attackers with ($\gamma < c\sqrt{\delta_{\max}}$), then Theorem 3.1 gives no guarantee that the consensus distance will reduce after aggregation. This is unfortunately not possible to improve upon, as we will show in the following § 3.5.2—if the connectivity is poor then the effect of Byzantine workers can be significantly amplified.

The conclusion above does not contradict the established impossibility result regarding the attainment of Byzantine consensus with fewer than $3b + 1$ nodes or less than $2b + 1$ connectivity [Fischer et al., 1986]. A distinctive element in our consideration is the inclusion of the mixing matrix among workers, rendering the mere count of nodes and edges insufficient for measuring the influence of Byzantine workers accurately. In scenarios where there are fewer than $3b + 1$ nodes, yet the edge weights linked to Byzantine workers are exceptionally low, the overall Byzantine influence becomes negligible, thereby enabling the achievement of approximate consensus. Conversely, with a connectivity less than $2b + 1$, if the edge weights between regular

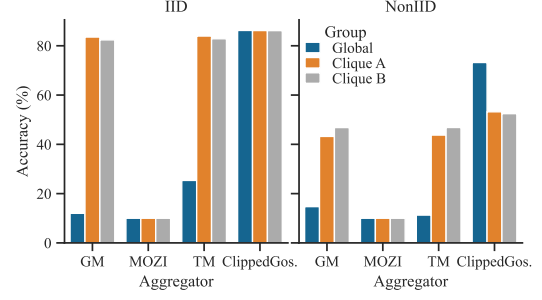
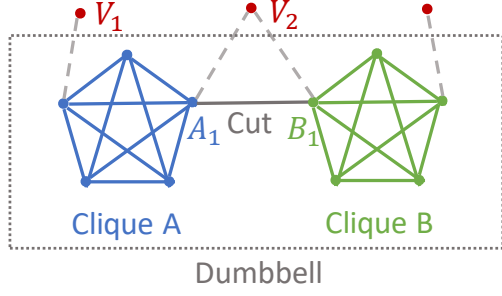


Fig. 3.1 A dumbbell topology of two cliques A and B of regular workers connected by an edge and a Byzantine worker V_2 (red) may attack the graph at different places. Fig. 3.2 Accuracies of models trained with ro- and B of regular workers connected by an edge and a Byzantine worker V_2 (red) may attack the graph at different places. The models are averaged within clique A, B, or all regular workers separately.

workers are relatively high, approximate consensus is still attainable. We encapsulate these dynamics using the spectral gap δ and the maximum weight of Byzantine workers δ_{\max} , which offer more precise measures of the influence exerted by Byzantine workers on the path to achieving approximate consensus.

3.5.2 Lower bounds due to communication constraints

Not all pairs of workers have direct communication links due to constraints such as physical distances in a sensor network. It is common that a subset of sensors are clustered within a small physical space while only few of them have communication links to the rest of the sensors. Such links form a cut-set of the communication topology and are crucial for information diffusion. On the other hand, attackers can increase consensus errors in the presence of these critical links.

Theorem 3.2. *Consider networks satisfying Assumption A of n nodes, each holding a number in $\{0, 1\}$, and only $\mathcal{O}(1/n^2)$ of the edges are adjacent to attackers. For any robust consensus algorithm \mathcal{A} , there exists a network such that the output of \mathcal{A} has an average consensus error of at least $\Omega(1)$.*

Proof. Consider two cliques A and B with n nodes each connected by an edge to each other and to a Byzantine node V_2 , c.f. Fig. 3.1. Suppose that we know all nodes have values in $\{0, 1\}$. Let all nodes in A have value 0. Now consider two settings:

World 1. All B nodes have value 0. However, Byzantine node V_2 pretends to be part of a clique identical to B which it *simulates*, except that all nodes have value 1. The true consensus average is 0.

World 2. All B nodes have value 1. This time the Byzantine node V_2 simulates clique B with value 0. The true consensus average here is 0.5.

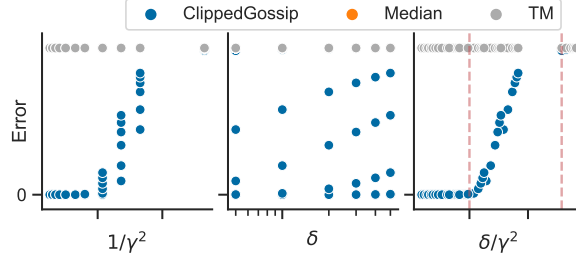


Fig. 3.3 Performance of CLIPPEDGOSSIP and baselines (TM and MEDIAN) under Byzantine attacks with varying γ and δ_{\max} . Each point represents the squared average consensus error of the last iterate of an algorithm. MEDIAN and TM have identical performance and CLIPPEDGOSSIP is consistently better. Further, the performance of CLIPPEDGOSSIP is best explained by the magnitude of (δ/γ^2) – it is excellent when the ratio is less than a threshold and degrades as it increases.

From the perspective of clique A, the two worlds are identical—it seems to be connected to one clique with value 0 and another with value 1. Thus, it must make $\Omega(1)$ error at least in one of the worlds. This proves that consensus is impossible in this setting. \square

While arguments above are similar to classical lower bounds in decentralized consensus which show we need $\delta \leq 1/3$ [Fischer et al., 1986], in our case there is only 1 Byzantine node (out of $2n + 1$ regular nodes) which controls only 2 edges i.e. $\delta = \mathcal{O}(1/n^2)$. This impossibility result thus drives home the additional impact through the restricted communication topology. Further, past impossibility results about robust decentralized consensus such as [Su and Vaidya, 2016a; Sundaram and Gharesifard, 2018] use combinatorial concepts such as the number of node-disjoint paths between the good nodes. However, such notions cannot account for the edge weights easily and cannot give finite-time convergence guarantees. Instead, our theory shows that the ratio of δ_{\max}/γ^2 accurately captures the difficulty of the problem. We next verify this empirically.

In Fig. 3.3, we show the final consensus error of three defenses under Byzantine attacks. TM and MEDIAN have a large error even for small δ_{\max} and large γ . The consensus error of CLIPPEDGOSSIP increases almost linearly with δ_{\max}/γ^2 . However, this phenomenon is not observed by looking at γ^{-2} or δ_{\max} alone, validating our theoretical analysis in Theorem 3.1. Details are deferred to § B.4.1.

3.6 Robust Decentralized Optimization

The general decentralized training algorithm can be formulated as

$$\mathbf{x}_i^{t+1/2} := \begin{cases} \mathbf{x}_i^t - \eta \mathbf{g}_i(\mathbf{x}_i^t) & i \in \mathcal{V}_R \\ * & i \in \mathcal{V}_B \end{cases}, \quad \mathbf{x}_i^{t+1} := \text{AGG}_i(\{\mathbf{x}_k^{t+1/2} : k \in \overline{\mathcal{N}}_i\})$$

Algorithm 3 Byzantine-Resilient Decentralized Optimization with CLIPPEDGOSSIP

Input: $\mathbf{x}^0 \in \mathbb{R}^d$, $\alpha, \eta, \{\tau_i^t\}, \mathbf{m}_i^0 = \mathbf{g}_i(\mathbf{x}^0)$

- 1: **for** $t = 0, 1, \dots$ **do**
- 2: **for** $i = 1, \dots, n$ **in parallel**
- 3: $\mathbf{m}_i^{t+1} = (1 - \alpha)\mathbf{m}_i^t + \alpha\mathbf{g}_i(\mathbf{x}_i^t)$
- 4: $\mathbf{x}_i^{t+1/2} = \mathbf{x}_i^t - \eta\mathbf{m}_i^{t+1}$ if $i \in \mathcal{V}_R$ else *
- 5: Exchange $\mathbf{x}_i^{t+1/2}$ with \mathcal{N}_i
- 6: $\mathbf{x}_i^{t+1} = \text{CLIPPEDGOSSIP}_i(\mathbf{x}_1^{t+1/2}, \dots, \mathbf{x}_n^{t+1/2}; \tau_i^{t+1})$
- 7: **end for**

Table 3.1 Comparison with prior work of convergence rates for non-convex objectives to a $\mathcal{O}(\delta\zeta^2)$ -neighborhood of stationary points. We recover comparable or improved rates as special cases.

	Reference	Setting	Convergence to ϵ -accuracy
Regular ($\delta = 0$)	Koloskova et al. [2020b]	-	$\mathcal{O}(\frac{\sigma^2}{n\epsilon^2} + \frac{\zeta}{\gamma\epsilon^{3/2}} + \frac{\sigma}{\sqrt{\gamma}\epsilon^{3/2}} + \frac{1}{\gamma\epsilon})$
Decentralized	This work	$\delta = 0$	$\mathcal{O}(\frac{\sigma^2}{n\epsilon^2} + \frac{\zeta}{\gamma\epsilon^{3/2}} + \frac{\sigma^{2/3}}{\gamma^{2/3}\epsilon^{4/3}} + \frac{1}{\gamma\epsilon})$
Byzantine-robust	Guo et al. [2021]	-	X
Fully-connected ($\gamma = 1$)	Gorbunov et al. [2021]	δ known	$\mathcal{O}(\frac{\sigma^2}{n\epsilon^2} + \frac{n\delta\sigma^2}{m\epsilon} + \frac{1}{\epsilon})^\dagger$
IID ($\zeta = 0$)	Gorbunov et al. [2021]	δ unknown	$\mathcal{O}(\frac{\sigma^2}{n\epsilon^2} + \frac{n^2\delta\sigma^2}{m\epsilon} + \frac{1}{\epsilon})^\dagger$
	This work	$\gamma = 1, \zeta = 0$	$\mathcal{O}(\frac{\sigma^2}{n\epsilon^2} + \frac{\delta\sigma^2}{\epsilon^2} + \frac{1}{\epsilon})$
Byzantine-robust	Karimireddy et al. [2021c]	-	$\mathcal{O}(\frac{\sigma^2}{\epsilon^2}(\delta + \frac{1}{n}) + \frac{1}{\epsilon})$
Federated Learning	This work	$\gamma = 1$	$\mathcal{O}(\frac{\sigma^2}{\epsilon^2}(\delta + \frac{1}{n}) + \frac{\zeta}{\epsilon^{3/2}} + \frac{\sigma^{2/3}}{\epsilon^{4/3}} + \frac{1}{\epsilon})$

[†] This method does not generalize to constrained communication topologies.

where η is the learning rate, $\mathbf{g}_i(\mathbf{x}) := \nabla F(\mathbf{x}, \xi_i)$ is a stochastic gradient, and $\xi_i^t \sim \mathcal{D}_i$ is the random batch at time t on worker i . The received message $\mathbf{x}_k^{t+1/2}$ can be arbitrary for Byzantine nodes $k \in \mathcal{V}_B$. Replacing AGG with plain gossip averaging (GOSSIP) recovers standard gossip SGD [Koloskova et al., 2019]. Under the presence of Byzantine workers, which is the main interest of our work, we will show that we can replace AGG with CLIPPEDGOSSIP and use local worker momentum to achieve Byzantine robustness [Karimireddy et al., 2021a]. The full procedure is described in Algorithm 3.

Theorem 3.3. Suppose Assumptions A-3.4 hold and $\delta_{\max} = \mathcal{O}(\gamma^2)$. Then for $\alpha := 3\eta L$, Algorithm 3 reaches $\frac{1}{T+1} \sum_{t=0}^T \|\nabla f(\bar{\mathbf{x}}^t)\|_2^2 \leq \frac{\delta_{\max}\zeta^2}{\gamma^2} + \epsilon$ in iteration complexity

$$\mathcal{O}\left(\frac{\sigma^2}{n\epsilon^2}\left(\frac{1}{n} + \delta_{\max}\right) + \frac{\zeta}{\gamma\epsilon^{3/2}} + \frac{\sigma^{2/3}}{\gamma^{2/3}\epsilon^{4/3}} + \frac{1}{\gamma\epsilon}\right).$$

Furthermore, the consensus distance satisfies the upper bound

$$\frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \|\mathbf{x}_i^\top - \bar{\mathbf{x}}^\top\|_2^2 \leq \mathcal{O}(\frac{\zeta^2}{\gamma^2(T+1)}).$$

We compare our analysis with existing works for non-convex objectives in Table 3.1.

Regular decentralized training. Even if there are no Byzantine workers ($\delta_{\max}=0$), our convergence rate is slightly faster than that of standard gossip SGD [Koloskova et al., 2020b]. The difference is that our third term $\mathcal{O}(\frac{\sigma^{2/3}}{\gamma^{2/3}\epsilon^{4/3}})$ is faster than their $\mathcal{O}(\frac{\sigma}{\sqrt{\gamma}\epsilon^{3/2}})$ for large σ and small ϵ . This is because we use local momentum which reduces the effect of variance σ . Thus momentum has a double use in this paper in achieving robustness as well as accelerating optimization.

Byzantine-robust federated learning. Federated learning uses a fully connected graph ($\gamma = 1$). We compare state of the art federated learning method [Karimireddy et al., 2021c] with our rate when $\gamma = 1$. Both algorithms converge to a $\Theta(\delta\zeta^2)$ -neighborhood of a stationary point and share the same leading term. This neighborhood can be circumvented with strong growth condition and over-parameterized models [Karimireddy et al., 2021c, Theorem III]. We incur additional higher-order terms $\mathcal{O}(\frac{\zeta}{\gamma\epsilon^{3/2}} + \frac{\sigma^{2/3}}{\gamma^{2/3}\epsilon^{4/3}})$ as a penalty for the generality of our analysis. This shows that the trusted server in federated learning can be removed without significant slowdowns.

Byzantine-robust decentralized SGD with fully connected topology. If we limit our analysis to a special case of a fully connected graph ($\gamma=1$) and IID data ($\zeta=0$), then our rate has the same leading term as [Gorbunov et al., 2021], which enjoys the scaling of the total number of regular nodes. The second term $\mathcal{O}(\frac{n}{m}\frac{\delta\sigma^2}{\epsilon})$ of [Gorbunov et al., 2021] is better than our $\mathcal{O}(\frac{1}{\epsilon}\frac{\delta\sigma^2}{\epsilon})$ for small ϵ because they additionally validate m random updates in each step. However, [Gorbunov et al., 2021] relies on secure protocols which do not easily generalize to constrained communication.

Byzantine-robust decentralized SGD with constrained communication. MOZI [Guo et al., 2021] does not provide a theoretical analysis on convergence and TM [Su and Vaidya, 2016a; Sundaram and Gharesifard, 2018; Yang and Bajwa, 2019a] only prove the asymptotic convergence of full gradient under a very strong assumption on connectivity and local honest majority.² Peng and Ling [2020] don't prove a rate for non-convex objective; but Gorbunov et al. [2021] which shows convergence of [Peng and Ling, 2020] on strongly convex objectives at a rate inferior to parallel SGD. In contrast, our convergence rate matches the standard stochastic analysis under much weaker assumptions than Su and Vaidya [2016a]; Sundaram and Gharesifard [2018]; Yang and Bajwa [2019a]. Unlike these prior works, our guarantees hold even if some subsets of nodes are surrounded by a majority of Byzantine attackers. This can also be observed in practice, as we show in § B.4.2.

Consensus for Byzantine-robust decentralized optimization. Theorem 3.3 gives a non-trivial result that regular workers reach consensus under the CLIPPEDGOSSIP aggregator. In Fig. 3.2 we demonstrate the consensus behavior of robust aggregators on the CIFAR-10 dataset on a dumbbell topology, without attackers ($\delta=0$). We compare the accuracies of models

²MOZI is renamed to UBAR in the latest version.

averaged within cliques A and B with model averaged over all workers. In the IID setting, the clique-averaged models of GM and TM are over 80% accuracy but the globally-averaged models are less than 30% accuracy. It means clique A and clique B are converging to two different critical points and GM and TM fail to reach consensus within the entire network! In contrast, the globally-averaged model of CLIPPEDGOSSIP is as good as or better than the clique-averaged models, both in the IID and non-IID setting.

Finally, we point out some avenues for further improvement: our results depend on the worst-case δ_{\max} . We believe it is possible to replace it with a (weighted) average of the $\{\delta_i\}$ instead. Also, extending our protocols to time-varying topologies would greatly increase their practicality.

Remark 5 (Adaptive choice of clipping radius τ_i^t). *In § B.4.5, we give an adaptive rule to choose the clipping radius τ_i^t for all $i \in \mathcal{V}_R$ and times t , based on the top percentile of close neighbors. This adaptive rule results in a value τ_i^t slightly smaller than the required theoretical value to preserve Byzantine robustness. In experiments, we found that the performance of optimization is robust to small perturbations of the clipping radius and that the adaptive rule performs well in all cases.*

3.7 Experiments

In this section, we empirically demonstrate successes and failures of decentralized training in the presence of Byzantine workers, and compare the performance of CLIPPEDGOSSIP with existing robust aggregators: 1) geometric median GM [Pillutla et al., 2019]; 2) coordinate-wise trimmed mean TM [Yang and Bajwa, 2019a]; 3) MOZI [Guo et al., 2020]. Coordinate-wise median [Yin et al., 2018b] and Krum [Blanchard et al., 2017] usually perform worse than GM so we exclude them in the experiments. All implementations are based on PyTorch [Paszke et al., 2019] and evaluated on different graph topologies, with a distributed MNIST dataset [LeCun et al., 1998]. We defer the experiments on CIFAR10 [Krizhevsky, 2012] to § B.4.3.³

We defer details of robust aggregators to § B.1, attacks to § B.2, topologies and mixing matrix to § B.3 and experiment setups and additional experiments to § B.4.

3.7.1 Decentralized defenses without attackers

Challenging topologies and data distribution may prevent existing robust aggregators from reaching consensus even when there is no Byzantine worker ($\delta = 0$). In this part, we consider the “dumbbell” topology c.f. Fig. 3.1. As non-IID data distribution, we split the training dataset by labels such that workers in clique A are training on digits 0 to 4 while workers in clique B are training on digits 5 to 9. This entanglement of topology and data distribution is motivated by realistic geographic constraints such as continents with dense intra-connectivity but sparse

³The code is available at [here](#).

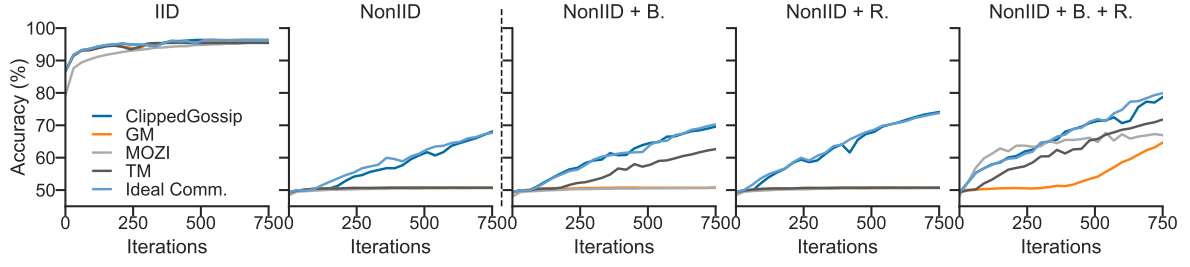


Fig. 3.4 Accuracy of the averaged model in clique A for the dumbbell topology. In the plot title “B.” stands for the bucketing (aggregating means of bucketed values) and “R.” stands for adding 1 additional random edge between two cliques. We see that i) CLIPPEDGOSSIP is consistently the best matching ideal averaging performance, ii) performance mildly improves by using bucketing, and iii) significantly improves when adding a single random edge (thereby improving connectivity).

inter-connection links e.g. through an undersea cable. In Fig. 3.4 we compare CLIPPEDGOSSIP with existing robust aggregators GM, TM, MOZI in terms of their accuracies of averaged model in clique A. The ideal communication refers to aggregation with gossip averaging.

Existing robust aggregators impede information diffusion. When cliques A and B have distinct data distribution (non-IID), workers in clique A rely on the graph cut to access the full spectrum of data and attain good performance. However, existing robust aggregators in clique A completely discard information from clique B because: 1) clique B model updates are outliers to clique A due to data heterogeneity; 2) clique B updates are outnumbered by clique A updates — clique A can only observe 1 update from B due to constrained communication. The 2nd plot in Fig. 3.4 shows that GM, TM, and MOZI only reach 50% accuracy in the non-IID setting, supporting that they impede information diffusion. This is in contrast to the 1st plot where cliques A and B have identical data distribution (IID) and information on clique A alone is enough to attain good performance. However, reaching local models does not imply reaching consensus, c.f. Fig. 3.2. On the other hand, CLIPPEDGOSSIP is the only robust aggregator that preserves the information diffusion rate as the ideal gossip averaging.

Techniques that improve information diffusion. To address these issues, we locally employ the *bucketing* technique of [Karimireddy et al., 2021c] for the non-IID case in the 3rd subplot. Plots 4 and 5 demonstrate the impact of one additional edge between the cliques to improve the spectral gap.

- The bucketing technique randomly inputs received vectors into buckets of equal size, averages the vectors in each bucket, and finally feeds the averaged vectors to the aggregator. While bucketing helps TM to overcome 50% accuracy, TM is still behind CLIPPEDGOSSIP. GM only improves by 1% while MOZI remains at almost the same accuracy.

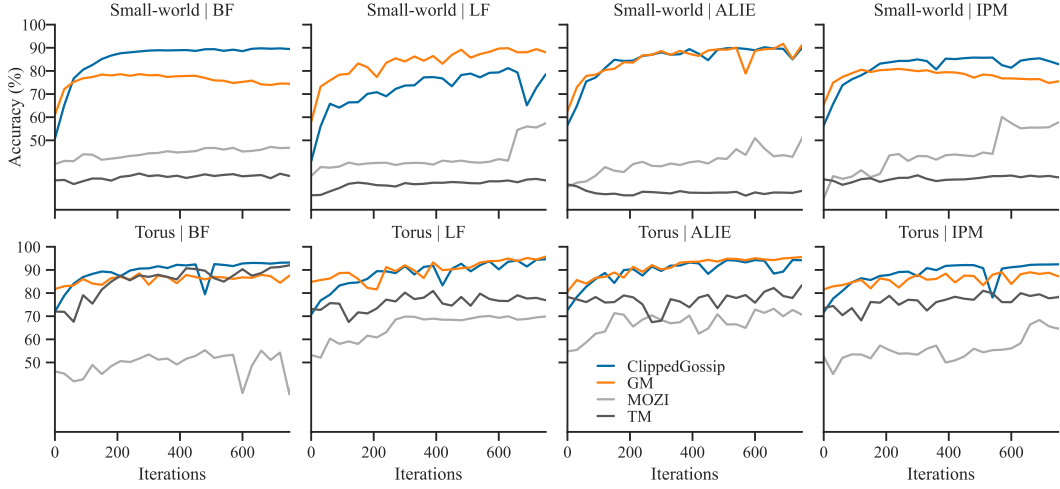


Fig. 3.5 Robust aggregators on randomized small-world (10 regular nodes) and torus topology (9 regular nodes) under Byzantine attacks (2 attackers). We observe that across all attacks and networks, clipped gossip has excellent performance, with the geometric median (GM) coming second.

- Adding one more random edge between two cliques improves the spectral gap γ from 0.0154 to 0.0286. CLIPPEDGOSSIP and gossip averaging converge faster as the theory predicts. However, TM, GM, and MOZI are still stuck at 50% for the same heterogeneity reason.
- Bucketing and adding a random edge help all aggregators exceed 50% accuracy.

3.7.2 Decentralized learning under more attacks and topologies.

In this section, we compare robust aggregators over more topologies and Byzantine attacks in the non-IID setting. We consider two topologies: randomized small world ($\gamma=0.084$) and torus ($\gamma=0.131$). They are much less restrictive than the dumbbell topology ($\gamma=0.043$) where all existing aggregators fail to reach consensus even $\delta=0$. For attacks, we implement state of the art federated attacks Inner product manipulation (IPM) [Xie et al., 2019a] and A little is enough (ALIE) [Baruch et al., 2019] and label-flipping (LF) and bit-flipping (BF). Details about topologies and the adaptation of FL attacks to the decentralized setup are provided in § B.3.1 and § B.2.

The results in Fig. 3.5 show that CLIPPEDGOSSIP has consistently superior performance under all topologies and attacks. All robust aggregators are generally performing better on easier topology (large γ). The GM has a very good performance on these two topologies but, as we have demonstrated in the dumbbell topology, GM does not work in more challenging topologies. Therefore, CLIPPEDGOSSIP is recommended for a general constrained topology.

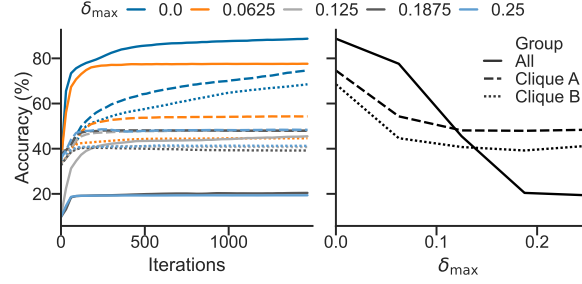


Fig. 3.6 Effect of the number of attackers on the accuracy of CLIPPEDGOSSIP under dissensus attack with varying δ_{\max} and fixed γ, ζ^2 . The solid (resp. dashed) lines denote models averaged over all (resp. clique A or B) regular workers. The right figure shows the performance of the last iterates of curves in the left figure.

3.7.3 Lower bound of optimization

We empirically investigate the lower bound of optimization $O(\delta_{\max}\zeta^2\gamma^{-2})$ in Theorem 3.3. In this experiment, we fix spectral gap γ , heterogeneity ζ^2 and use different δ_{\max} fractions of Byzantine edges in the dumbbell topology. The Byzantine workers are added to V_1 in clique A and its mirror node in clique B. We define the following *dissensus* attack for decentralized optimization

Definition 3.5 (DISSENSUS attack). *For $i \in \mathcal{V}_R$ and $\epsilon_i > 0$, a dissensus attacker $j \in \mathcal{N}_i \cap \mathcal{V}_B$ sends*

$$\mathbf{x}_j := \mathbf{x}_i - \epsilon_i \frac{\sum_{k \in \mathcal{N}_i \cap \mathcal{V}_R} \mathbf{W}_{ik}(\mathbf{x}_k - \mathbf{x}_i)}{\sum_{j \in \mathcal{N}_i \cap \mathcal{V}_B} \mathbf{W}_{ij}}. \quad (3.6)$$

The resulting Figure 3.6 shows that with increasing δ_{\max} the model quality drops significantly. This is in line with our proven robust convergence rate in terms of δ_{\max} . Notice that for large δ_{\max} , the model averaged over all workers performs even worse than those averaged within cliques. It means the models in two cliques are essentially disconnected and are converging to different local minima or stationary points of a non-convex landscape. See § B.4.2 for details.

3.8 Discussion

The main takeaway from our work is that ill-connected communication topologies can vastly magnify the effect of bad actors. As long as the communication topology is reasonably well connected (say $\gamma = 0.35$) and the fraction of attackers is mild (say $\delta = 10\%$), clipped gossip provably ensures robustness. Under more extreme conditions, however, *no algorithm* can guarantee robust convergence. Given that decentralized consensus has been proposed as a backbone for digital democracy [Bulteau et al., 2021], and that decentralized learning is touted to be an alternative to current centralized training paradigms, our findings are significant. A simple strategy we recommend (along with using CLIPPEDGOSSIP) is adding random edges to improve the connectivity and robustify the network.

Acknowledgements. This project was supported by SNSF grant 200020_200342. SPK is supported by an SNSF postdoc mobility fellowship. This project was initiated in the master thesis of [Cappelletti \[20c\]](#) who analyzed the Byzantine-free setting. We also thank Anastasiia Koloskova and Lê Nguyễn Hoàng for fruitful discussions on optimization and authentication.

Chapter 4

Secure Byzantine-Robust Machine Learning

4.1 Preface

Contribution and sources. This chapter reproduces [He et al., 2020b], proposing a novel distributed training framework to tackle data privacy and robustness in machine learning applications. The authors had shared responsibility in conceptualizing the ideas and the writing process. In detail, the individual contributions are:

- Lie He: Conceptualization, Writing, Formal Analysis.
- Sai Praneeth Karimireddy: Conceptualization, Writing.
- Martin Jaggi: Supervision, Administration, Writing (review and editing), Conceptualization.

Summary. Privacy and robustness are two important factors in distributed machine learning applications. Regular participants would like to benefit from collaborative training and at the same time want to keep their data private during the multiparty computation (MPC). The service provider would like to protect the training from malicious participants. However, these two goals are often conflicting as typical robust aggregators (e.g. median) are not MPC friendly.

This chapter introduces a multi-server based secure aggregation framework capable of withstanding Byzantine attacks and server-worker collusion, offering a solution to a challenge previously thought to be intractable. Our focus is to integrate current and future distance-based robust aggregation rules with secure aggregation, thus improving privacy without compromising the accuracy of machine learning models.

4.2 Introduction

Recent years have witnessed fast growth of successful machine learning applications based on data collected from decentralized user devices. Unfortunately, however, currently most of the important machine learning models on a societal level do not have their utility, control, and privacy aligned with the data ownership of the participants. This issue can be partially attributed to a fundamental conflict between the two leading paradigms of traditional centralized training of models on one hand, and decentralized/collaborative training schemes on the other hand. While centralized training violates the privacy rights of participating users, existing alternative training schemes are typically not robust. Malicious participants can sabotage the training system by feeding it wrong data intentionally, known as *data poisoning*. In this paper, we tackle this problem and propose a novel distributed training framework which offers both *privacy* and *robustness*.

When applied to datasets containing personal data, the use of privacy-preserving techniques is currently required under regulations such as the *General Data Protection Regulation* (GDPR) or *Health Insurance Portability and Accountability Act* (HIPAA). The idea of training models on decentralized datasets and incrementally aggregating model updates via a central server motivates the federated learning paradigm [McMahan et al., 2017a]. However, the averaging in federated learning, when viewed as a *multi-party computation* (MPC), does not preserve the *input privacy* because the server observes the models directly. The *input privacy* requires each party learns nothing more than the output of computation which in this paradigm means the aggregated model updates. To solve this problem, *secure* aggregation rules as proposed in [Bonawitz et al., 2017] achieve guaranteed input privacy. Such secure aggregation rules have found wider industry adoption recently e.g. by Google on Android phones [Bonawitz et al., 2019; Ramage and Mazzocchi, 2020] where input privacy guarantees can offer e.g. efficiency and exactness benefits compared to differential privacy (both can also be combined).

The concept of Byzantine robustness has received considerable attention in the past few years for practical applications, as a way to make the training process robust to malicious actors. A Byzantine participant or worker can behave arbitrarily malicious, e.g. sending arbitrary updates to the server. This poses great challenge to the most widely used aggregation rules, e.g. simple average, since a single Byzantine worker can compromise the results of aggregation. A number of Byzantine-robust aggregation rules have been proposed recently [Alistarh et al., 2018; Blanchard et al., 2017; Mhamdi et al., 2018; Muñoz-González et al., 2017, 2019; Yin et al., 2018b] and can be used as a building block for our proposed technique.

Achieving both input privacy and Byzantine robustness however remained elusive so far, with Bagdasaryan et al. [2020b] stating that robust rules “...are incompatible with secure aggregation”. We here prove that this is not the case. Closest to our approach is [Pillutla et al., 2019] which tolerates data poisoning but does not offer Byzantine robustness. Prio [Corrigan-Gibbs and Boneh, 2017] is a private and robust aggregation system relying on secret-shared non-interactive

proofs (SNIP). While their setting is similar to ours, the robustness they offer is limited to check the range of the input. Besides, the encoding for SNIP has to be affine-aggregable and is expensive for clients to compute.

In this paper, we propose a secure aggregation framework with the help of two non-colluding honest-but-curious servers. This framework also tolerates server-worker collusion. In addition, we combine robustness and privacy at the cost of leaking only worker similarity information which is marginal for high-dimensional neural networks. Note that our focus is not to develop new defenses against state-of-the-art attacks, e.g. [Baruch et al., 2019; Xie et al., 2019a]. Instead, we focus on making *arbitrary* current and future distance-based robust aggregation rules (e.g. Krum by Mhamdi et al. [2018], RFA by Pillutla et al. [2019]) compatible with secure aggregation.

Main contributions. We propose a novel distributed training framework which is

- **Privacy-preserving:** our method keeps the input data of each user secure against any other user, and against our honest-but-curious servers.
- **Byzantine robust:** our method offers Byzantine robustness and allows to incorporate existing robust aggregation rules, e.g. [Alistarh et al., 2018; Blanchard et al., 2017]. The results are exact, i.e. identical to the non-private robust methods.
- **Fault tolerant and easy to use:** our method natively supports workers dropping out or newly joining the training process. It is also easy to implement and to understand for users.
- **Efficient and scalable:** the computation and communication overhead of our method is negligible (less than a factor of 2) compared to non-private methods. Scalability in terms of cost including setup and communication is linear in the number of workers.

4.3 Problem setup, privacy, and robustness

We consider the distributed setup of n user devices, which we call workers, with the help of two additional servers. Each worker i has its own private part of the training dataset. The workers want to collaboratively train a public model benefitting from the joint training data of all participants.

In every training step, each worker computes its own private model update (e.g. a gradient based on its own data) denoted by the vector \mathbf{x}_i . Then workers synchronously send their gradients to the servers. The aggregation protocol aims to compute the sum $\mathbf{z} = \sum_{i=1}^n \mathbf{x}_i$ (or a robust version of this aggregation), which is then used to update a public model. While the result \mathbf{z} is public in all cases, the protocol must keep each \mathbf{x}_i private from any adversary or other workers.

We posit the simultaneous existence of two distinct types of adversaries: Byzantine attackers and privacy attackers. A worker can embody at most one type of attacker and these two forms of attackers do not collude. Byzantine attackers are defined the same as those in Chapter 2,

capable of deviating from the prescribed protocols to transmit arbitrary adversarial messages aimed at undermining the training. Both servers and workers can potentially assume a role of a privacy attacker. We assume honest-but-curious servers, which, while not colluding amongst themselves, may potentially collude with malicious workers. Such a server follows the protocol but may inspect all transmitted messages. Additionally, we presume all communication channels are secure. Our framework ensures *input privacy*, implying that servers and workers ascertain nothing beyond what can be deduced from the public output of the aggregation \mathbf{z} .

Additive secret sharing. Secret sharing is a way to split any secret into multiple parts such that no part leaks the secret. Formally, suppose a scalar a is a *secret* and the secret holder shares it with k parties through *secret-shared values* $\langle a \rangle$. In this paper, we only consider additive secret-sharing where $\langle a \rangle$ is a notation for the set $\{a_i\}_{i=1}^k$ which satisfy $a = \sum_{p=1}^k a_p$, with a_p held by party p . Crucially, it must not be possible to reconstruct a from any a_p . For vectors like \mathbf{x} , their secret-shared values $\langle \mathbf{x} \rangle$ are simply component-wise scalar secret-shared values.

Two-server setting. We assume there are two non-colluding servers: model server (S1) and worker server (S2). S1 holds the output of each aggregation and thus also the machine learning model which is public to all workers. S2 holds intermediate values to perform Byzantine aggregation. Another key assumption is that the servers have no incentive to collude with workers, perhaps enforced via a potential huge penalty if exposed. It is realistic to assume that the communication link between the two servers S1 and S2 is faster than the individual links to the workers. To perform robust aggregation, the servers will need access to a sufficient number of *Beaver's triples*. These are data-independent values required to implement secure multiplication in MPC on both servers, and can be precomputed beforehand. For completeness, the classic algorithm for multiplication is given in Appendix C.2.1.

Byzantine-robust aggregation oracles. Most of existing robust aggregation algorithms rely on distance measures to identify potential adversarial behaviors [Blanchard et al., 2017; Ghosh et al., 2019; Li et al., 2019; Mhamdi et al., 2018; Yin et al., 2018b]. All such distance-based aggregation rules can be directly incorporated into our proposed scheme, making them secure. While many aforementioned papers assume that the workers have i.i.d datasets, our protocol is oblivious to the distribution of the data across the workers. In particular, our protocol also works with schemes such as [Ghosh et al., 2019; He et al., 2020a; Li et al., 2019] designed for non-iid data.

4.4 Secure aggregation protocol: two-server model

Each worker first splits its private vector \mathbf{x}_i into two additive secret shares, and transmits those to each corresponding server, ensuring that neither server can reconstruct the original vector

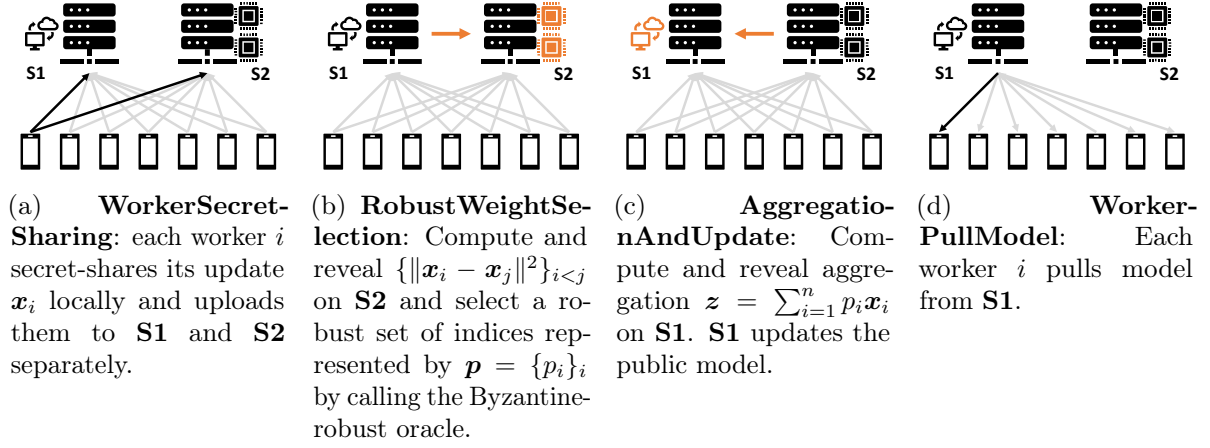


Fig. 4.1 Illustration of Algorithm 5. The orange components on servers represent the computation-intensive operations at low communication cost between servers.

on its own. The two servers then execute our secure aggregation protocol. On the level of servers, the protocol is a two-party computation (2PC). In the case of non-robust aggregation, servers simply add all shares (we present this case in detail in Algorithm 4). In the robust case which is of our main interest here, the two servers exactly emulate an existing Byzantine robust aggregation rule, at the cost of revealing only distances of worker gradients on the server (the robust algorithm is presented in Algorithm 5). Finally, the resulting aggregated output vector \mathbf{z} is sent back to all workers and applied as the update to the public machine learning model.

4.4.1 Non-robust secure aggregation

In each round, Algorithm 4 consists of two stages:

- **WorkerSecretSharing** (Figure 4.1a): each worker i randomly splits its private input \mathbf{x}_i into two additive secret shares $\mathbf{x}_i = \mathbf{x}_i^{(1)} + \mathbf{x}_i^{(2)}$. This can be done e.g. by sampling a large noise value ξ_i and then using $(\mathbf{x}_i \pm \xi_i)/2$ as the shares. Worker i sends $\mathbf{x}_i^{(1)}$ to **S1** and $\mathbf{x}_i^{(2)}$ to **S2**. We write $\langle \mathbf{x}_i \rangle$ for the two secret-shared values distributed over the two servers.
- **AggregationAndUpdate** (Figure 4.1c): Given binary weights $\{p_i\}_{i=1}^n$, each server locally computes $\langle \sum_{i=1}^n p_i \mathbf{x}_i \rangle$. Then **S2** sends its share $\langle \sum_{i=1}^n p_i \mathbf{x}_i \rangle^{(2)}$ to **S1** so that **S1** can then compute $\mathbf{z} = \sum_{i=1}^n p_i \mathbf{x}_i$. **S1** updates the public model with \mathbf{z} .

Our secure aggregation protocol is extremely simple, and as we will discuss later, has very low communication overhead, does not require heavy cryptographic primitives, gives strong input privacy and is compatible with differential privacy, and is robust to worker dropouts and failures. We believe this makes our protocol especially attractive for federated learning applications.

We now argue about correctness and privacy. It is clear that the output \mathbf{z} of the above protocol satisfies $\mathbf{z} = \sum_{i=1}^n p_i \mathbf{x}_i$, ensuring that all workers compute the right update. Now we

argue about the privacy guarantees. We track the values stored by each of the servers and workers:

- **S1**: Its own secret shares $\{\mathbf{x}_i^{(1)}\}_{i=1}^n$ and the sum of the other shares $\langle \sum_{i=1}^n p_i \mathbf{x}_i \rangle^{(2)}$.
- **S2**: Its own secret shares $\{\mathbf{x}_i^{(2)}\}_{i=1}^n$.
- Worker i : \mathbf{x}_i and $\mathbf{z} = \sum_{i=1}^n p_i \mathbf{x}_i$.

Clearly, the workers have no information other than the aggregate \mathbf{z} and their own data. **S2** only has the secret share which on their own leak no information about any data. Hence surprisingly, **S2** does not learn anything in this process. **S1** has its own secret share and also the sum of the other shares. If $n = 1$, then $\mathbf{z} = \mathbf{x}_i$ and hence **S1** is allowed to learn everything. If $n > 1$, then **S1** cannot recover information about any individual secret share $\mathbf{x}_i^{(2)}$ from the sum. Thus, **S1** learns \mathbf{z} and nothing else.

4.4.2 Robust secure aggregation

We now describe how Algorithm 5 replaces the simple aggregation with any distance-based robust aggregation rule **Aggr**, e.g. Multi-Krum [Blanchard et al., 2017]. The key idea is to use two-party MPC to securely compute multiplication.

- **WorkerSecretSharing** (Figure 4.1a): As before, each worker i secret shares $\langle \mathbf{x}_i \rangle$ distributed over the two servers **S1** and **S2**.
- **RobustWeightSelection** (Figure 4.1b): After collecting all secret-shared values $\{\langle \mathbf{x}_i \rangle\}_i$, the servers compute pairwise difference $\{\langle \mathbf{x}_i - \mathbf{x}_j \rangle\}_{i < j}$ locally. **S2** then reveals—to itself exclusively—in plain text all of the pairwise Euclidean distances between workers $\{\|\mathbf{x}_i - \mathbf{x}_j\|^2\}_{i < j}$ with the help of precomputed Beaver’s triples and Algorithm 8. The distances are kept private from **S1** and workers. **S2** then feeds these distances to the distance-based robust aggregation rule **Aggr**, returning (on **S2**) a binary weight vector $\mathbf{p} = \{p_i\}_{i=1}^n \in \{0,1\}^n$, representing the indices of the robust subset selected by **Aggr**.
- **AggregationAndUpdate** (Figure 4.1c): Given weight vector \mathbf{p} from previous step, we would like **S1** to compute $\sum_{i=1}^n p_i \mathbf{x}_i$. To do so, **S2** secret shares with **S1** the values of $\{\langle p_i \rangle\}$ instead of sending in plain-text since they may be private. Then, **S1** reveals to itself, but not to **S2**, in plain text the value of $\mathbf{z} = \sum_{i=1}^n p_i \mathbf{x}_i$ using secret-shared multiplication and updates the public model.
- **WorkerPullModel** (Figure 4.1d): Workers pull the latest public model on **S1** and update it locally.

The key difference between the robust and the non-robust aggregation scheme is the weight selection phase where **S2** computes all pairwise distances and uses this to run a robust-aggregation rule in a black-box manner. **S2** computes these distances i) without leaking any information to **S1**, and ii) without itself learning anything other than the pair-wise distances (and in particular none of the actual values of \mathbf{x}_i). To perform such a computation, **S1** and **S2** use precomputed

Beaver's triplets (Algorithm 8 in the Appendix), which can be made available in a scalable way [Smart and Tanguy, 2019].

4.4.3 Salient features

Overall, our protocols are very resource-light and straightforward from the perspective of the workers. Further, since we use Byzantine-robust aggregation, our protocols are provably fault-tolerant even if a large fraction of workers misbehave. This further lowers the requirements of a worker. We elaborate the features as follows.

Communication overhead. In applications, individual uplink speed from worker and servers is typically the main bottleneck, as it is typically much slower than downlink, and the bandwidth between servers can be very large. For our protocols, the time spent on the uplink is within a factor of 2 of the non-secure variants. Besides, our protocol only requires one round of communication, which is an advantage over interactive proofs.

Fault tolerance. The workers in Algorithm 4 and Algorithm 5 are completely stateless across multiple rounds and there is no *offline* phase required. This means that workers can start participating in the protocols simply by pulling the latest public model. Further, our protocols are unaffected if some workers drop out in the middle of a round. Unlike in [Bonawitz et al., 2017], there is no entanglement between workers and we don't have unbounded recovery issues.

Compatibility with local differential privacy. One byproduct of our protocol can be used to convert differentially private mechanisms, such as [Abadi et al., 2016] which only guarantees the privacy of the aggregated model, into the stronger *locally* differentially private mechanisms which guarantee user-level privacy.

Other Byzantine-robust oracles. We can also use some robust-aggregation rules which are not based on pair-wise distances such as Byzantine SGD [Alistarh et al., 2018]. Since the basic structures are very similar to Algorithm 5, we put Algorithm 10 in the appendix.

Security. The security of Algorithm 4 is straightforward as we previously discussed. The security of Algorithm 5 again relies on the separation of information between **S1** and **S2** with neither the workers nor **S1** learning anything other than the aggregate \mathbf{z} . We will next formally prove that this is true even in the presence of malicious workers.

Remark 1. *Our proposed scheme leverages classic 2-party secret-sharing for addition and multiplication. These building blocks however are originally proposed for integers and quantized values, not real values. For floating point operations as used in machine learning, one can use the secure counterparts [Aliasgari et al., 2013] of the two operations. This is facilitated by deep learning training being robust to limited precision training [Gupta et al., 2015] and additional*

Algorithm 4 Two-Server Secure Aggregation (Non-robust variant)

Setup: n workers (non-Byzantine) with private vectors \mathbf{x}_i . Two non-colluding servers **S1** and **S2**.

Workers: (**WorkerSecretSharing**)

1. split private \mathbf{x}_i into additive secret shares $\langle \mathbf{x}_i \rangle = \{\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}\}$ (such that $\mathbf{x}_i = \mathbf{x}_i^{(1)} + \mathbf{x}_i^{(2)}$)
2. send $\mathbf{x}_i^{(1)}$ to **S1** and $\mathbf{x}_i^{(2)}$ to **S2**

Servers:

1. $\forall i$, **S1** collects $\mathbf{x}_i^{(1)}$ and **S2** collects $\mathbf{x}_i^{(2)}$
 2. (**AggregationAndUpdate**):
 - (a) On **S1** and **S2**, compute $\langle \sum_{i=1}^n \mathbf{x}_i \rangle$ locally
 - (b) **S2** sends its share of $\langle \sum_{i=1}^n \mathbf{x}_i \rangle$ to **S1**
 - (c) **S1** reveals $\mathbf{z} = \sum_{i=1}^n \mathbf{x}_i$ to everyone
-

noise [Neelakantan et al., 2016], with current models routinely trained in 16 bit precision. In contrast to [Bonawitz et al., 2017] which relies on advanced cryptographic primitives such as Diffie-Hellman’s key agreement which must remain exact and discrete, our protocols only use much simpler secure arithmetic operations—only addition and multiplication—which are tolerant to rounding errors. For the privacy implications of secret sharing when using floating point, which go beyond the scope of our work, we refer the reader to the information theoretic analysis of Aliasgari et al. [2013].

4.5 Theoretical guarantees

4.5.1 Exactness

In the following lemma we show that Algorithm 5 gives the exact same result as non-privacy-preserving version.

Lemma 4.2 (Exactness of Algorithm 5). *The resulting \mathbf{z} in Algorithm 5 is identical to the output of the non-privacy-preserving version of the used robust aggregation rule.*

Proof. After secret-sharing \mathbf{x}_i to $\langle \mathbf{x}_i \rangle$ to two servers, Algorithm 5 performs local differences $\{\langle \mathbf{x}_i - \mathbf{x}_j \rangle\}_{i < j}$. Using shared-values multiplication via Beaver’s triple, **S2** obtains the list of true Euclidean distances $\{\|\mathbf{x}_i - \mathbf{x}_j\|^2\}_{i < j}$. The result is fed to a *distance-based* robust aggregation rule oracle, all solely on **S2**. Therefore, the resulting indices $\{p_i\}_i$ as used in $\mathbf{z} := \sum_{i=1}^n p_i \mathbf{x}_i$ are identical to the aggregation of non-privacy-preserving robust aggregation. \square

With the exactness of the protocol established, we next focus on the privacy guarantee.

Algorithm 5 Two-Server Secure Robust Aggregation (Distance-Based)

Setup: n workers, αn of which are Byzantine. Two non-colluding servers **S1** and **S2**.

Workers: (**WorkerSecretSharing**)

1. split private \mathbf{x}_i into additive secret shares $\langle \mathbf{x}_i \rangle = \{\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}\}$ (such that $\mathbf{x}_i = \mathbf{x}_i^{(1)} + \mathbf{x}_i^{(2)}$)
2. send $\mathbf{x}_i^{(1)}$ to **S1** and $\mathbf{x}_i^{(2)}$ to **S2**

Servers:

1. $\forall i$, **S1** collects gradient $\mathbf{x}_i^{(1)}$ and **S2** collects $\mathbf{x}_i^{(2)}$
2. (**RobustWeightSelection**):
 - (a) For each pair $(\mathbf{x}_i, \mathbf{x}_j)$ compute their Euclidean distance ($i < j$):
 - On **S1** and **S2**, compute $\langle \mathbf{x}_i - \mathbf{x}_j \rangle = \langle \mathbf{x}_i \rangle - \langle \mathbf{x}_j \rangle$ locally
 - Use precomputed Beaver's triples (see Algorithm 8) to compute the distance $\|\mathbf{x}_i - \mathbf{x}_j\|^2$
 - (b) **S2** perform robust aggregation rule $\mathbf{p} = \text{Aggr}(\{\|\mathbf{x}_i - \mathbf{x}_j\|^2\}_{i < j})$
 - (c) **S2** secret-shares $\langle \mathbf{p} \rangle$ with **S1**
3. (**AggregationAndUpdate**):
 - (a) On **S1** and **S2**, use MPC multiplication to compute $\langle \sum_{i=1}^n p_i \mathbf{x}_i \rangle$ locally
 - (b) **S2** sends its share of $\langle \sum_{i=1}^n p_i \mathbf{x}_i \rangle^{(2)}$ to **S1**
 - (c) **S1** reveals $\mathbf{z} = \sum_{i=1}^n p_i \mathbf{x}_i$ to all workers.

Workers:

1. (**WorkerPullModel**): Collect \mathbf{z} and update model locally

4.5.2 Privacy

We prove probabilistic (information-theoretic) notion of privacy which gives the strongest guarantee possible. Formally, we will show that the distribution of the secret does not change even after being conditioned on all observations made by all participants, i.e. each worker i , **S1** and **S2**. This implies that the observations carry absolutely no information about the secret. Our results rely on the existence of simple additive secret-sharing protocols as discussed in the Appendix.

Each worker i only receives the final aggregated \mathbf{z} at the end of the protocol and is not involved in any other manner. Hence no information can be leaked to them. We will now examine **S1**. The proofs below rely on Beaver's triples which we summarize in the following lemma.

Lemma 4.3 (Beaver's triples). *Suppose we secret share $\langle x \rangle$ and $\langle y \rangle$ between **S1** and **S2** and want to compute xy on **S2**. There exists a protocol which enables such computation which uses precomputed shares $BV = (\langle a \rangle, \langle b \rangle, \langle c \rangle)$ such that **S1** does not learn anything and **S2** only learns xy .*

Due to the page limit, we put the details about Beaver's triples, multiplying secret shares, as well as the proofs for the next two theorems to the Appendix.

Theorem 4.1 (Privacy for **S1**). *Let $\mathbf{z} = \sum_{i=1}^n p_i \mathbf{x}_i$ where $\{p_i\}_{i=1}^n$ is the output of byzantine oracle or a vector of 1s (non-private). Let $BV_{ij} = \langle \mathbf{a}_{ij}, \mathbf{b}_{ij}, \mathbf{c}_{ij} \rangle$ and $BV_{p_i} = \langle \mathbf{a}_i^p, \mathbf{b}_i^p, \mathbf{c}_i^p \rangle$ be the*

Beaver's triple used in the multiplications. Let $\langle \cdot \rangle^{(1)}$ be the share of the secret-shared values $\langle \cdot \rangle$ on **S1**. Then for all workers i

$$\mathbb{P}(\mathbf{x}_i = x_i \mid \{\langle \mathbf{x}_i \rangle^{(1)}, \langle p_i \rangle^{(1)}\}_{i=1}^n, \{BV_{i,j}^{(1)}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_{ij}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{b}_{ij}\}_{i < j}, \\ \{\langle \|\mathbf{x}_i - \mathbf{x}_j\|^2 \rangle^{(1)}\}_{i < j}, \{BVp_i^{(1)}, p_i - \mathbf{a}_i^p, p_i - \mathbf{b}_i^p\}_{i=1}^n, \mathbf{z}) = \mathbb{P}(\mathbf{x}_i = x_i \mid \mathbf{z})$$

Note that the conditioned values are what **S1** observes throughout the algorithm. $\{BV_{i,j}^{(1)}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_{ij}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{b}_{ij}\}_{i < j}$ and $\{BVp_i^{(1)}, p_i - \mathbf{a}_i^p, p_i - \mathbf{b}_i^p\}_{i=1}^n$ are intermediate values during shared values multiplication.

For **S2**, the theorem to prove is a bit different because in this case **S2** doesn't know the output of aggregation \mathbf{z} . In fact, this is more similar to an independent system which knows little about the underlying tasks, model weights, etc. We show that while **S2** has observed many intermediate values, it can only learn no more than what can be inferred from model distances.

Theorem 4.2 (Privacy for **S2**). Let $\{p_i\}_{i=1}^n$ is the output of byzantine oracle or a vector of 1s (non-private). Let $BV_{ij} = \langle \mathbf{a}_{ij}, \mathbf{b}_{ij}, \mathbf{c}_{ij} \rangle$ and $BVp_i = \langle \mathbf{a}_i^p, \mathbf{b}_i^p, \mathbf{c}_i^p \rangle$ be the Beaver's triple used in the multiplications. Let $\langle \cdot \rangle^{(2)}$ be the share of the secret-shared values $\langle \cdot \rangle$ on **S2**. Then for all workers i

$$\mathbb{P}(\mathbf{x}_i = x_i \mid \{\langle \mathbf{x}_i \rangle^{(2)}, \langle p_i \rangle^{(2)}, p_i\}_{i=1}^n, \{BV_{i,j}^{(2)}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_{ij}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{b}_{ij}\}_{i < j}, \\ \{\langle \|\mathbf{x}_i - \mathbf{x}_j\|^2 \rangle^{(2)}, \|\mathbf{x}_i - \mathbf{x}_j\|^2\}_{i < j}, \{BVp_i^{(2)}, p_i - \mathbf{a}_i^p, p_i - \mathbf{b}_i^p\}_{i=1}^n) \\ = \mathbb{P}(\mathbf{x}_i = x_i \mid \{\|\mathbf{x}_i - \mathbf{x}_j\|^2\}_{i < j}) \quad (4.1)$$

Note that the conditioned values are what **S2** observed throughout the algorithm. $\{BV_{i,j}^{(2)}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_{ij}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{b}_{ij}\}_{i < j}$ and $\{BVp_i^{(2)}, p_i - \mathbf{a}_i^p, p_i - \mathbf{b}_i^p\}_{i=1}^n$ are intermediate values during shared values multiplication.

The model distances indeed only leaks similarity among the workers. Such similarity, however, does not tell **S2** information about the parameters; in [Mhamdi et al., 2018] the *leeway attack* attacks distance based-rules because they don't distinguish two gradients with evenly distributed noise and two different gradients very different in one parameter. This means the leaked information has low impact to the privacy.

It is also worth noting that curious workers can only inspect others' values by learning from the public model/update. This is because in our scheme, workers don't interact directly and there is only one round of communication between servers and workers. So the only message a worker receives is the public model update.

4.5.3 Combining with differential privacy

While input privacy is our main goal, our approach is naturally compatible with other orthogonal notions of privacy. Global differential privacy (DP) [Abadi et al., 2016; Chase et al., 2017;

[Shokri and Shmatikov, 2015] is mainly concerned about the privacy of the *aggregated* model, and whether it leaks information about the training data. On the other hand, local differential privacy (LDP) [Evfimievski et al., 2003; Kasiviswanathan et al., 2011] is stronger notions which is also concerned with the training process itself. It requires that every communication transmitted by the worker does not leak information about their data. In general, it is hard to learn deep learning models satisfying LDP using iterate perturbation (which is the standard mechanism for DP) [Bonawitz et al., 2017].

Our non-robust protocol *is naturally compatible* with local differential privacy. Consider the usual iterative optimization algorithm which in each round t performs

$$\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \eta(\mathbf{x}_t + \nu_t), \text{ where } \mathbf{x}_t = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_{t,i}. \quad (4.2)$$

Here \mathbf{x}_t is the aggregate update, \mathbf{w}_t is the model parameters, and ν_t is the noise added for DP [Abadi et al., 2016].

Theorem 4.3 (from DP to LDP). *Suppose that the noise ν_t in (4.2) is sufficient to ensure that the set of model parameters $\{\mathbf{w}_t\}_{t \in [T]}$ satisfy (ϵ, δ) -DP for $\epsilon \geq 1$. Then, running (4.2) with using Alg. 4 to compute $(\mathbf{x}_t + \eta_t)$ by securely aggregating $\{\mathbf{x}_{1,t} + n\eta_t, \mathbf{x}_{2,t}, \dots, \mathbf{x}_{n,t}\}$ satisfies (ϵ, δ) -LDP.*

Unlike existing approaches, we do not face a tension between differential privacy which relies on real-valued vectors and cryptographic tools which operate solely on discrete/quantized objects. This is because our protocols do not rely on cryptographic primitives like Diffie-Hellman key agreement, in contrast to e.g. [Bonawitz et al., 2017]. In particular, the vectors \mathbf{x}_i can be full-precision (real-valued) at the cost of adding marginal rounding error which can be tolerated by robust aggregation rule and stochastic gradient descent algorithms. Thus, our secure aggregation protocol can be integrated with a mechanism which has global DP properties e.g. [Abadi et al., 2016], and prove *local* DP guarantees for the resulting mechanism.

4.6 Empirical analysis of overhead

We present an illustrative simulation on a local machine (i7-8565U) to demonstrate the overhead of our scheme. We use PyTorch with MPI to train a neural network of 1.2 million parameters on the MNIST dataset. We compare the following three settings: simple aggregation with 1 server, secure aggregation with 2 servers, robust secure aggregation with 2 servers (with Krum [Blanchard et al., 2017]). The number of workers is always 5.

Figure 4.2 shows the time spent on all parts of training for one aggregation step. T_{grad} is the time spent on batch gradient computation; T_{w2s} refers to the time spend on uploading and downloading gradients; T_{s2s} is the time spend on communication between servers. Note that the server-to-server communication could be further reduced by employing more efficient

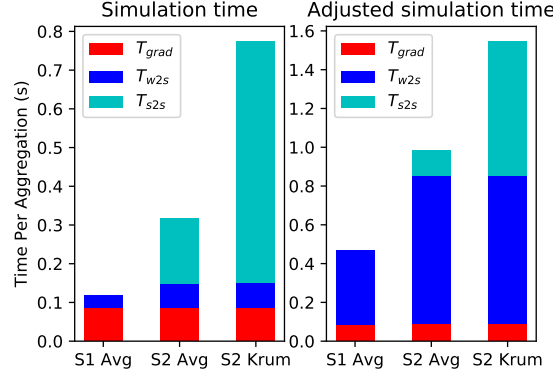


Fig. 4.2 Left: Actual time spent; Right: Time adjusted for network bandwidth.

aggregation rules. Since the simulation is run on a local machine, time spent on communication is underestimated. In the right hand side figure, we adjust time by assuming the worker-to-server link has 100Mbps bandwidth and 1Gbps respectively for the server-to-server link. Even in this scenario, we can see that the overhead from private aggregation is small. Furthermore, the additional overhead by the robustness module is moderate comparing to the standard training, even for realistic deep-learning settings. For comparison, a zero-knowledge-proof-based approach need to spend 0.03 seconds to encode a submission of 100 integers [Corrigan-Gibbs and Boneh, 2017].

4.7 Literature review

Secure Aggregation. In the standard distributed setting with 1 server, Bonawitz et al. [2017] proposes a secure aggregation rule which is also fault tolerant. They generate a shared secret key for each pair of users. The secret keys are used to construct masks to the input gradients so that masks cancel each other after aggregation. To achieve fault tolerance, they employ Shamir’s secret sharing. To deal with active adversaries, they use a public key infrastructure (PKI) as well as a second mask applied to the input. A followup work [Mandal et al., 2018] minimizes the pairwise communication by outsourcing the key generation to two non-colluding cryptographic secret providers. However, both protocols are still not scalable because each worker needs to compute a shared-secret key and a noise mask for every other client. When recovering from failures, all live clients are notified and send their masks to the server, which introduces significant communication overhead. In contrast, workers in our scheme are freed from coordinating with other workers, which leads to a more scalable system.

Byzantine-Robust Aggregation/SGD. Blanchard et al. [2017] first proposes Krum and Multi-Krum for training machine learning models in the presence of Byzantine workers. Mhamdi et al. [2018] proposes a general enhancement recipe termed *Bulyan*. Alistarh et al. [2018]

proves a robust SGD training scheme with optimal sample complexity and the number of SGD computations. [Muñoz-González et al. \[2019\]](#) uses HMM to detect and exclude Byzantine workers for federated learning. [Yin et al. \[2018b\]](#) proposes median and trimmed-mean based robust algorithms which achieve optimal statistical performance. For robust learning on non-i.i.d dataset only appear recently [[Ghosh et al., 2019](#); [He et al., 2020a](#); [Li et al., 2019](#)]. Further, [Xie et al. \[2018b\]](#) generalizes the Byzantine attacks to manipulate data transfer between workers and server and [Xie et al. \[2019c\]](#) extends it to tolerate an arbitrary number of Byzantine workers.

[Pillutla et al. \[2019\]](#) proposes a robust aggregation rule RFA which is also privacy preserving. However, it is only robust to data poisoning attack as it requires workers to compute aggregation weights according to the protocol. [Corrigan-Gibbs and Boneh \[2017\]](#) proposes a private and robust aggregation system based on secret-shared non-interactive proof (SNIP). Despite the similarities between our setups, the generation of a SNIP proof on client is expansive and grows with the dimensions. Besides, this paper offers limited robustness as it only validates the range of the data.

Inference As A Service. An orthogonal line of work is inference as a service or oblivious inference. A user encrypts its own data and uploads it to the server for inference. [[Chou et al., 2018](#); [Gilad-Bachrach et al., 2016](#); [Hesamifard et al., 2017](#); [Juvekar et al., 2018](#); [Liu et al., 2017](#); [Mohassel and Zhang, 2017](#); [Riazi et al., 2019](#); [Rouhani et al., 2017](#)] falls into a general category of 2-party computation (2PC). A number of issues have to be taken into account: the non-linear activations should be replaced with MPC-friendly activations, represent the floating number as integers. [Ryffel et al. \[2019\]](#) uses functional encryption on polynomial networks. [Gilad-Bachrach et al. \[2016\]](#) also have to adapt activations to polynomial activations and max pooling to scaled mean pooling.

Server-Aided MPC. One common setting for training machine learning model with MPC is the server-aided case [[Chen et al., 2019](#); [Mohassel and Zhang, 2017](#)]. In previous works, both the model weights and the data are stored in shared values, which in turn makes the inference process computationally very costly. Another issue is that only a limited number of operations (function evaluations) are supported by shared values. Therefore, approximating non-linear activation functions again introduces significant overhead. In our paper, the computation of gradients are local to the workers, only output gradients are sent to the servers. Thus no adaptations of the worker’s neural network architectures for MPC are required.

4.8 Conclusion

In this paper, we propose a novel secure and Byzantine-robust aggregation framework. To our knowledge, this is the first work to address these two key properties jointly. Our algorithm is simple and fault tolerant and scales well with the number of workers. In addition, our framework

holds for any existing distance-based robust rule. Besides, the communication overhead of our algorithm is roughly bounded by a factor of 2 and the computation overhead, as shown in Algorithm 8, is marginal and can even be computed prior to training.

Chapter 5

RelaySum for Decentralized Deep Learning on Heterogeneous Data

5.1 Preface

Contribution and sources. This chapter reproduces [Vogels et al., 2021] with minor edits. Most of the methodology and writing were done by the author and Thijs Vogels. The author carried out most of formal analysis. The experiments and visualization were conducted mostly by Thijs Vogels. Detailed individual contributions:

- Lie He (author): Formal analysis (70%), Methodology (40%), Writing (50%).
- Thijs Vogels (co-first author): Methodology (60%), Software (80%), Visualization, Writing (50%).
- Anastasia Koloskova: Formal analysis.
- Tao Lin: Software.
- Sai Praneeth Karimireddy: Formal analysis
- Sebastian U. Stich: Formal analysis, Writing–review and editing.
- Martin Jaggi: Writing, Review and editing, Project administration, Supervision.

Summary. Decentralized machine learning involves individual workers interleaving model updates on their local data and communicating with neighboring nodes. The gossip averaging mechanism is commonly used to exchange information through weighted average. However, gossip averaging is slow to distribute information across the network and is sensitive to data heterogeneity. In this paper, we propose RelaySum, a novel mechanism for information propagation in decentralized learning. RelaySum utilizes spanning trees to ensure precise and uniform distribution of information to all workers, with finite delays based on inter-node distances. We show that RelaySum can be implemented on trees with the same communication volume per step as gossip averaging, using additional memory linear in the number of neighbors. We use

RelaySum in the RelaySGD learning algorithm, which is independent of data heterogeneity and scalable for scenarios with numerous workers. We demonstrate the effectiveness of RelaySGD on image- and text classification tasks, where it outperforms state-of-the-art decentralized learning algorithms. The code for RelaySum can be found at <http://github.com/epfml/relaysgd>.

5.2 Introduction

Ever-growing datasets lay at the foundation of the recent breakthroughs in machine learning. Learning algorithms therefore must be able to leverage data distributed over multiple devices, in particular for reasons of efficiency and data privacy. There are various paradigms for distributed learning, and they differ mainly in how the devices collaborate in communicating model updates with each other. In the *all-reduce* paradigm, workers average model updates with all other workers at every training step. In *federated learning* [McMahan et al., 2017b], workers perform local updates before sending them to a central server that returns their global average to the workers. Finally, *decentralized learning* significantly generalizes the two previous scenarios. Here, workers communicate their updates with only few directly-connected neighbors in a network, without the help of a server.

Decentralized learning offers strong promise for new applications, allowing any group of agents to collaboratively train a model while respecting the data locality and privacy of each contributor [Nedic, 2020]. At the same time, it removes the single point of failure in centralized systems such as in federated learning [Kairouz et al., 2019], improving robustness, security, and privacy. Even from a pure efficiency standpoint, decentralized communication patterns can speed up training in data centers [Assran et al., 2019a].

In decentralized learning, workers share their local stochastic gradient updates with the others through *gossip* communication [Xiao and Boyd, 2004]. They send their updates to their neighbors, which iteratively propagate the updates further into the network. The workers typically use iterative *gossip averaging* of their models with their neighbors, using averaging weights chosen to ensure asymptotic uniform distribution of each update across the network. It will take τ rounds of communication for an update from worker i to reach a worker j that is τ hops away, and when it first arrives, the update is exponentially weakened by repeated averaging with weights < 1 . In general networks, worker j will never exactly, but only asymptotically receive its uniform share of the update. The slow distribution of updates not only slows down training, but also makes decentralized learning sensitive to heterogeneity in workers' data distributions.

We study an alternative mechanism to gossip averaging, which we call RelaySum. RelaySum operates on spanning trees of the network, and distributes information exactly uniformly within a finite number of gossip steps equal to the diameter of the network. Rather than iteratively averaging models, each node acts as a 'router' that *relays* messages through the whole network without decaying their weight at every hop. While naive all-to-all routing requires n^2 messages

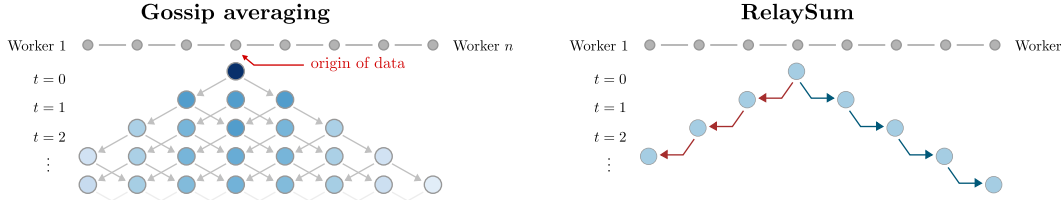


Fig. 5.1 To spread information across a decentralized network, classical gossip averaging diffuses information slowly through the network. The left figure illustrates the spread of information originating from the fourth worker in a chain network. In RelaySum, the messages are *relayed* without reweighting, resulting in uniform delivery of the information to every worker. When multiple workers broadcast simultaneously (not pictured), RelaySum can *sum* their messages and use the same bandwidth as gossip averaging.

to be transmitted at each step, we show that on trees, only n messages (one per edge) are sufficient. This is enabled by the key observation that the routers can *merge* messages by *summation* to avoid any extra communication compared to gossip averaging. RelaySum achieves this using additional memory linear in the number of edges, and by tailoring the messages sent to different neighbors. At each time step, RelaySum workers receive a uniform average of exactly one message from each worker. Those messages just originate from different time delays depending on how many hops they travelled. The difference between gossip averaging and RelaySum is illustrated in Figure 5.1.

The RelaySum mechanism is structurally similar to Belief Propagation algorithms for inference in graphical models. This link was made by Zhang et al. [2019], who used the same mechanism for decentralized weighted average consensus in control.

We use RelaySum in the RelaySGD learning algorithm. We theoretically show that this algorithm is not affected by differences in workers' data distributions. Compared to other algorithms that have this property [Pu and Nedic, 2018; Tang et al., 2018], RelaySGD does not require the selection of averaging weights, and its convergence does not depend on the spectral gap of the averaging matrix, but instead on the network diameter.

While RelaySum is formulated for trees, it can be used in any decentralized network. We use the Spanning Tree Protocol [Perlman, 1985] to construct spanning trees of any network in a decentralized fashion. RelaySGD often performs better on any such spanning tree than gossip-based methods on the original graph. When the communication network can be chosen freely, the algorithm can use double binary trees [Sanders et al., 2009]. While these trees have logarithmic diameter and scale to many workers, RelaySGD in this setup uses only constant memory equivalent to two extra copies of the model parameters and sends and receives only two models per iteration.

Surprisingly, in deep learning with highly heterogeneous data, prior methods that are theoretically independent of data heterogeneity [Pu and Nedic, 2018; Tang et al., 2018], perform worse than heuristic methods that do not have this property, but use cleverly designed time-

varying communication topologies [Assran et al., 2019a]. In extensive tests on image- and text classification, RelaySGD performs better than both kinds of baselines at equal communication budget.

5.3 Related work

Out of the multitude of decentralized optimization methods, first-order algorithms that interleave local gradient updates with a form of gossip averaging Johansson et al. [2009]; Nedic et al. [2017] show most promise for deep learning. Such algorithms are theoretically analyzed for convex and non-convex objectives in Johansson et al. [2009]; Nedic and Ozdaglar [2009]; Nedic et al. [2017], and [Assran et al., 2019a; Lian et al., 2017b; Lin et al., 2021b; Tang et al., 2018] demonstrate that gossip-based methods can perform well in deep learning.

In a gossip averaging step, workers average their local models with the models of their direct neighbors. The corresponding ‘mixing matrix’ is a central object of study. The matrix can be doubly-stochastic Koloskova et al. [2020b]; Lian et al. [2017b]; Nedic et al. [2017], column-stochastic Assran et al. [2019a]; Nedic and Olshevsky [2016]; Tsianos et al. [2012]; Xi and Khan [2017], row-stochastic Xi et al. [2018]; Xin et al. [2019], or a combination Pu et al. [2021]; Xin and Khan [2018, 2020]. Column-stochastic methods use the *push-sum* consensus mechanism [Kempe et al., 2003] and can be used on directed graphs. Our analysis borrows from the theory developed for those methods.

While gossip averages in general requires an infinite number of steps to reach exact consensus, another line of work identifies mixing schemes that yield exact consensus in finite steps. For some graphs, this is possible with time-independent averaging weights Georgopoulos [2011]; Ko [2010]. One can also achieve finite-time consensus with time-varying mixing matrices. On trees, for instance, exact consensus can be achieved by routing updates to a root node and back, in exactly diameter number of steps Georgopoulos [2011]; Ko [2010]. On some graphs, tighter bounds can be established Hendrickx et al. [2014]. For fully-connected networks with n workers, Assran et al. [2019a] design a sparse time-varying communication scheme that yields exact consensus in a cycle of $\log n$ averaging steps and performs well in deep learning.

The ‘relay’ mechanism of RelaySGD was previously used by Zhang et al. [2019] in the control community for the decentralized weighted average consensus problem, but they do not use it in the context of optimization. Zhang et al. also introduce a modified algorithm for loopy graphs, but this modification makes the achieved consensus inexact. The ‘relay’ mechanism effectively turns a sparse graph into a fully-connected graph with communication delays. Work on delayed consensus Nedić and Ozdaglar [2010] and optimization Agarwal and Duchi [2011]; Tsianos and Rabbat [2011] analyzes such schemes for centralized distributed algorithms. Those consensus schemes are, however, not directly applicable to decentralized optimization.

A fundamental challenge in decentralized learning is dealing with data that is not identically distributed among workers. Because, in this case, workers pursue different optima, workers

may drift [Nedic et al. \[2017\]](#) and this can harm convergence. There is a large family of algorithms that introduce update corrections that provably mitigate such data heterogeneity. Examples applicable to non-convex problems are exact diffusion [\[Yuan et al., 2019\]](#), Gradient Tracking [\[Lorenzo and Scutari, 2016; Pu and Nedic, 2018; Zhang and You, 2020\]](#), D^2 [\[Tang et al., 2018\]](#), PushPull [\[Pu et al., 2021\]](#). To tackle the same challenge, [Lin et al. \[2021b\]; Yuan et al. \[2021\]](#) propose modifications to local momentum to empirically improve performance in deep learning, but without provable guarantees. [Lu and Sa \[2021\]](#) propose DeTAG which overlaps multiple consecutive gossip steps and gradient computations to accelerate information diffusion. This technique could be applied to the RelaySum mechanism, too.

5.4 Method

Setup We consider standard decentralized optimization with data distributed over $n \geq 1$ nodes:

$$f^* := \min_{\mathbf{x} \in \mathbb{R}^d} [f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n [f_i(\mathbf{x}) := \mathbb{E}_{\xi \sim \mathcal{D}_i} F_i(\mathbf{x}, \xi_i)]] .$$

Here \mathcal{D}_i denotes the distribution of the data on node i and $f_i: \mathbb{R}^d \rightarrow \mathbb{R}$ the local optimization objectives. Workers are connected by a network respecting a graph topology $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, n\}$ denotes the set of workers, and \mathcal{E} the set of undirected communication links between them (without self loops). Each worker i can only directly communicate with its neighbors $\mathcal{N}_i \subset \mathcal{V}$.

Decentralized learning with gossip We consider synchronous first-order algorithms that interleave local gradient-based updates

$$\mathbf{x}_i^{(t+1/2)} = \mathbf{x}_i^{(t)} + \mathbf{u}_i^{(t)}$$

with message exchange between connected workers. For SGD with typical gossip averaging (DP-SGD [\[Lian et al., 2017b\]](#)), the local updates can be written as $\mathbf{u}_i^{(t)} = -\gamma \nabla f_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)})$, and the messages exchanged between pairs of connected workers (i, j) are $\mathbf{m}_{i \rightarrow j}^{(t)} = \mathbf{x}_i^{(t+1/2)} \in \mathbb{R}^d$. Each timestep, the workers average their model with received messages,

$$\mathbf{x}_i^{(t+1)} = \mathbf{W}_{ii} \mathbf{x}_i^{(t+1/2)} + \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij} \mathbf{m}_{j \rightarrow i}^{(t)}, \quad (\text{DP-SGD})$$

using averaging weights defined by a *gossip matrix* $\mathbf{W} \in \mathbb{R}^{n \times n}$.

In this scheme, an update $\mathbf{u}_i^{(t_1)}$ from any worker i will be linearly incorporated into the model $\mathbf{x}_j^{(t_2)}$ at a later timestep t_2 with weight $(\mathbf{W}^{t_2-t_1})_{ij}$. The gossip matrix must be chosen such that these weights asymptotically converge to $\frac{1}{n}$, distributing all updates uniformly over the workers. This setup appears in, for example, [\[Koloskova et al., 2020b; Lian et al., 2017b\]](#).

Uniform model averaging If the graph topology is fully-connected, any worker can communicate with any other worker, and it is ideal to use ‘all-reduce averaging’,

$$\mathbf{x}_i^{(t+1)} = \frac{1}{n} \sum_{j=1}^n \mathbf{x}_j^{(t+1/2)}.$$

Contrary to the decentralized scheme (DP-SGD), this algorithm does not degrade in performance if data is distributed heterogeneously across workers. In sparsely connected networks, however, all-reduce averaging requires routing messages through the network. On arbitrary networks, such a routing protocol requires at least a number of communication steps equal to the network diameter τ_{\max} —the minimum number of hops some messages have to travel.

RelaySGD In this paper, we approximate the all-reduce averaging update as

$$\mathbf{x}_i^{(t+1)} = \frac{1}{n} \sum_{j=1}^n \mathbf{x}_j^{(t-\tau_{ij}+1/2)}, \quad (\text{RelaySGD})$$

where τ_{ij} is minimum number of network hops between workers i and j (and $\tau_{ii} = 0$). Since it takes τ_{ij} steps to route a message from worker i to j , this scheme could be implemented using a peer-to-peer routing protocol like Ethernet. Of course, this naive implementation drastically increases the bandwidth used compared to gossip averaging. The key insight of this paper is that, on tree networks, the RelaySGD update rule can be implemented while using the same communication volume per step as gossip averaging, using additional memory linear in the number of a worker’s direct neighbors.

RelaySum To implement RelaySGD, we require a communication mechanism that delivers sums of delayed ‘parcels’ $s_w^{(t)} = \sum_{j=1}^n p_j^{(t-\tau_{wj})}$ to each worker w in a tree network, where the parcel $p_j^{(t)}$ is created by worker j at time t . To simplify the exposition, let us first consider the simplest type of tree network: a chain. In a chain, a worker w is connected to workers $w - 1$ and $w + 1$, if those exist, and the delays are $\tau_{ij} = |i - j|$. We can then decompose

$$s_w^{(t)} = \sum_{j=1}^n p_j^{(t-\tau_{wj})} = p_w^{(t)} + \underbrace{\sum_{j=1}^{w-1} p_j^{(t-\tau_{wj})}}_{\text{parcels from the ‘left’}} + \underbrace{\sum_{j=w+1}^n p_j^{(t-\tau_{wj})}}_{\text{parcels from the ‘right’}}.$$

The sum of parcels from the ‘left’ will be sent as one message $m_{(w-1) \rightarrow w}$ from worker $w - 1$ to w , and the sum of data from the ‘right’ will be sent as one message $m_{(w+1) \rightarrow w}$ from $w + 1$ to w . Neighboring workers can compute these messages from the messages they received from their neighbors in the previous timestep. Compared to typical gossip averaging, RelaySum requires additional memory linear in the number of neighbors, but it uses the same volume of communication.

Algorithm 6 RelaySGD

Input: $\forall i, \mathbf{x}_i^{(0)} = \mathbf{x}^{(0)}; \forall i, j, \mathbf{m}_{i \rightarrow j}^{(-1)} = \mathbf{0}$, counts $c_{i \rightarrow j}^{(-1)} = 0$, learning rate γ , tree network

- 1: **for** $t = 0, 1, \dots$ **do**
- 2: **for** node i **in parallel**
- 3: $\mathbf{x}_i^{(t+1/2)} = \mathbf{x}_i^{(t)} - \gamma \nabla f_i(\mathbf{x}_i^{(t)})$ (or Adam/momentum)
- 4: **for** each neighbor $j \in \mathcal{N}_i$ **do**
- 5: Send $\mathbf{m}_{i \rightarrow j}^{(t)} = \mathbf{x}_i^{(t+1/2)} + \sum_{k \in \mathcal{N}_i \setminus j} \mathbf{m}_{k \rightarrow i}^{(t-1)}$ (relay messages from other neighbors)
- 6: Send corresponding counters $c_{i \rightarrow j}^{(t)} = 1 + \sum_{k \in \mathcal{N}_i \setminus j} c_{k \rightarrow i}^{(t-1)}$
- 7: Receive $(\mathbf{m}_{j \rightarrow i}^{(t)}, c_{j \rightarrow i}^{(t)})$ from node j
- 8: $\bar{n}_i^{(t+1)} = 1 + \sum_{j \in \mathcal{N}_i} c_{j \rightarrow i}^{(t)}$ (\bar{n} converges to the total number of workers)
- 9: $\mathbf{x}_i^{t+1} = \frac{1}{\bar{n}_i^{(t+1)}} \left(\mathbf{x}_i^{(t+1/2)} + \sum_{j \in \mathcal{N}_i} \mathbf{m}_{j \rightarrow i}^{(t)} \right)$ ($= \frac{1}{n} \sum_{j=1}^n \mathbf{x}_j^{(t-\tau_{ij}+1/2)}$)
- 10: **end for**

Algorithm 6 shows how this scheme is generalized to general tree networks and incorporated into RelaySGD. Along with the model parameters, we send scalar counters that are used in the first few iterations of the algorithm $t \leq \tau_{\max}$ to correct for messages that have not yet arrived.

Spanning trees RelaySGD is formulated on tree networks, but it can be used on any communication graph by constructing a spanning tree. In a truly decentralized setting, we can use the Spanning Tree Protocol [Perlman, 1985] used in Ethernet to find such trees in a decentralized fashion. The protocol elects a leader as the root of the tree, after which every other node finds the fastest path to this leader.

On the other hand, when the decentralized paradigm is used in a data center to reduce communication, RelaySGD can run on double binary trees [Sanders et al., 2009] used in MPI and NCCL [Jeaugey, 2019]. The key idea of double binary trees is to use two different communication topologies for different parts of the model. We communicate odd coordinates using a balanced binary tree A , and communicate the even coordinates with a complimentary tree B . The trees A and B are chosen such that internal nodes (with 3 edges) in one tree are leaves (with only 1 edge) in the other. Using the combination of two trees, RelaySGD requires only constant extra memory equivalent to at most 2 model copies (just like the Adam optimizer [Kingma and Ba, 2015]), and it sends and receives the equivalent of 2 models (just like on a ring).

5.5 Theoretical analysis

Since RelaySGD updates worker's models at time step $t + 1$ using models from (at most) the past τ_{\max} steps, we conveniently reformulate RelaySGD in the following way: Let $\mathbf{Y}^{(t)}, \mathbf{G}^{(t)} \in \mathbb{R}^{n(\tau_{\max}+1) \times d}$ denote stacked worker models and gradients whose row vectors at index $n \cdot \tau + i$

represent

$$\left[\mathbf{Y}^{(t)}\right]_{n\tau+i}^\top = \begin{cases} \mathbf{x}_i^{(t-\tau)} & t \geq \tau \\ \mathbf{x}^{(0)} & \text{otherwise} \end{cases}, \quad \left[\mathbf{G}^{(t)}\right]_{n\tau+i}^\top = \begin{cases} \nabla F_i(\mathbf{x}_i^{(t-\tau)}; \xi_i^{(t-\tau)}) & t \geq \tau \\ \mathbf{x}^{(0)} & \text{otherwise} \end{cases}$$

for all times $t \geq 0$, delay $\tau \in [0, \tau_{\max}]$ and worker $i \in [n]$. Then (RelaySGD) can be written as

$$\mathbf{Y}^{(t+1)} = \mathbf{W}\mathbf{Y}^{(t)} - \gamma\tilde{\mathbf{W}}\mathbf{G}^{(t)}$$

where $\mathbf{W}, \tilde{\mathbf{W}} \in \mathbb{R}^{n(\tau_{\max}+1) \times n(\tau_{\max}+1)}$ are non-negative matrices whose elements are

$$[\mathbf{W}]_{n\tau+i, n\tau'+j} = \begin{cases} \frac{1}{n} & \tau = 0 \text{ and } \tau' = \tau_{ij} \\ 1 & i = j \text{ and } \tau = \tau' + 1, \\ 0 & \text{otherwise} \end{cases}, \quad [\tilde{\mathbf{W}}]_{n\tau+i, n\tau'+j} = \begin{cases} \frac{1}{n} & \tau = 0 \text{ and } \tau' = \tau_{ij} \\ 0 & \text{otherwise} \end{cases}$$

for all $\tau, \tau' \in [0, \tau_{\max}]$ and $i, j \in [n]$. The matrix \mathbf{W} can be interpreted as the mixing matrix of an ‘augmented graph’ [Nedić and Ozdaglar, 2010] with additional virtual ‘forwarding nodes’. \mathbf{W} is row stochastic and its largest eigenvalue is 1. The vector of all ones $\mathbf{1}_{n(\tau_{\max}+1)} \in \mathbb{R}^{n(\tau_{\max}+1)}$ is a right eigenvector of \mathbf{W} and let $\boldsymbol{\pi} \in \mathbb{R}^{n(\tau_{\max}+1)}$ be the left eigenvector such that $\boldsymbol{\pi}^\top \mathbf{1}_{n(\tau_{\max}+1)} = 1$.

We characterize the convergence rate of the consensus distance in the following key lemma:

Lemma 5.1 (Key lemma). *There exists an integer $m = m(\mathbf{W}) > 0$ such that for any $\mathbf{X} \in \mathbb{R}^{n(\tau_{\max}+1) \times d}$ we have*

$$\|\mathbf{W}^m \mathbf{X} - \mathbf{1}\boldsymbol{\pi}^\top \mathbf{X}\|^2 \leq (1-p)^{2m} \|\mathbf{X} - \mathbf{1}\boldsymbol{\pi}^\top \mathbf{X}\|^2,$$

where $p = \frac{1}{2}(1 - |\lambda_2(\mathbf{W})|)$ is a constant.

All the following optimization convergence results will only depend on the *effective spectral gap* $\rho := \frac{p}{m}$ of \mathbf{W} . We empirically observe that $\rho = \Theta(1/n)$ for a variety of network topologies (see Figure D.1 in Appendix D.1).

Remark 2. *The above key lemma is similar to [Koloskova et al., 2020b, Assumption 4] for gossip-type averaging with symmetric matrices. However, in our case \mathbf{W} is just a row stochastic matrix, and its spectral norm $\|\mathbf{W}\|_2 > 1$. In general, the consensus distance can increase after just one single communication step (multiplication by \mathbf{W}). That is why we need $m > 1$. The proof of the Lemma relies on a Perron-Frobenius type theorem, and holds over several steps m instead of a single iteration. It means RelaySum defines a consensus algorithm with linear convergence rate which pulls models closer.*

Our main convergence results hold under the following common assumptions, as e.g. Koloskova et al. [2020b].

Assumption A (L-smoothness). For each $i \in [n]$, $F_i(\mathbf{x}, \xi) : \mathbb{R}^D \times \Omega_i \rightarrow \mathbb{R}$ is differentiable for each $\xi \in \text{supp}(\mathcal{D}_i)$ and there exists a constant $L \geq 0$ such that for each $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, $\xi \in \text{supp}(\mathcal{D}_i)$:

$$\|\nabla F_i(\mathbf{x}, \xi) - \nabla F_i(\mathbf{y}, \xi)\| \leq L\|\mathbf{x} - \mathbf{y}\|.$$

Assumption B (Uniform bounded noise). There exists constant $\bar{\sigma}$, such that for all $\mathbf{x} \in \mathbb{R}^d$, $i \in [n]$,

$$\mathbb{E}_\xi \|\nabla F_i(\mathbf{x}, \xi) - \nabla f_i(\mathbf{x})\|^2 \leq \bar{\sigma}^2.$$

Assumption C (μ -convexity). For $i \in [n]$, each function $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ is μ -(strongly) convex for constant $\mu \geq 0$. That is, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$

$$f_i(\mathbf{x}) - f_i(\mathbf{y}) + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 \leq \nabla f_i(\mathbf{x})^\top (\mathbf{x} - \mathbf{y}).$$

Theorem 5.1 (RelaySGD). For any target accuracy $\epsilon > 0$ and an optimal solution \mathbf{x}^* , (**Convex:**) under Assumptions A, B and C with $\mu \geq 0$, it holds that

$$\frac{1}{T+1} \sum_{t=0}^T (f(\bar{\mathbf{x}}^{(t)}) - f(\mathbf{x}^*)) \leq \epsilon \text{ after } \mathcal{O}\left(\frac{\bar{\sigma}^2}{n\epsilon^2} + \frac{C\sqrt{L}\bar{\sigma}}{\epsilon^{3/2}} + \frac{CL}{\epsilon}\right) R_0^2 \text{ iterations.}$$

Here $\bar{\mathbf{x}}^{(t)} := \boldsymbol{\pi}^\top \mathbf{Y}^{(t)}$ averages past models, $R_0^2 = \|\mathbf{x}^0 - \mathbf{x}^*\|^2$, and $C = \mathcal{O}(\frac{1}{\rho} \tau_{\max}^{3/2})$.

(**Non-convex:**) under Assumptions A and B, it holds that

$$\frac{1}{T+1} \sum_{t=0}^T \|\nabla f(\bar{\mathbf{x}}^{(t)})\|^2 \leq \epsilon \text{ after } \mathcal{O}\left(\frac{\bar{\sigma}^2}{n\epsilon^2} + \frac{C\bar{\sigma}}{\epsilon^{3/2}} + \frac{C}{\epsilon}\right) LF_0 \text{ iterations,}$$

where $F_0 := f(\bar{\mathbf{x}}^{(0)}) - f(\mathbf{x}^*)$.

The dominant term in our convergence result, $\mathcal{O}(\frac{\bar{\sigma}^2}{n\epsilon^2})$ matches with the dominant term in the convergence rate of centralized ('all-reduce') mini-batch SGD, and thus can not be improved.

In contrast to other methods, the presented convergence result of RelaySGD is independent of the data heterogeneity ζ^2 in [Koloskova et al., 2020b, Assumption 3b].

Definition 5.4 (Data heterogeneity). There exists a constant ζ^2 such that $\forall i \in [n], \mathbf{x} \in \mathbb{R}^d$

$$\|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\|_2^2 \leq \zeta^2.$$

Remark 3. For convex objectives, Assumptions B and 5.4 can be relaxed to only hold at the optimum \mathbf{x}^* . A weaker variant of Assumption A only uses L-smoothness of f_i [Koloskova et al., 2020b, Assumption 1b].

Comparing to gossip averaging for convex f_i which has complexity $\mathcal{O}(\frac{\bar{\sigma}^2}{n\epsilon^2} + (\frac{\zeta}{\rho} + \frac{\bar{\sigma}}{\sqrt{\rho}}) \frac{\sqrt{L}}{\epsilon^{3/2}} + \frac{L}{\rho\epsilon}) R_0^2$, our rate for RelaySGD does not depend on ζ^2 and has same leading term $\mathcal{O}(\frac{\bar{\sigma}^2}{n\epsilon^2})$ as D^2 .

5.6 Experimental analysis and practical properties

5.6.1 Effect of network topology

Random quadratics To efficiently investigate the scalability of RelaySGD with respect to the number of workers, and to study the benefits of binary tree topologies over chains, we introduce a family of synthetic functions. We study *random quadratics* with local cost functions $f_i(\mathbf{x}) = \|\mathbf{A}_i \mathbf{x} - \mathbf{b}_i\|^2$ to precisely control all constants that appear in our theoretical analysis. The Hessians \mathbf{A}_i are initialized randomly, and their spectrum is scaled to achieve a desired smoothness L and strong convexity μ . The offsets \mathbf{b}_i ensure a desired level of heterogeneity ζ^2 and distance between optimum and initialization r_0 . Appendix D.2.4 describes the generation of these quadratics in detail.

Scalability on rings and trees Using these quadratics, Figure 5.2 studies the number of steps required to reach a suboptimality $f(\bar{\mathbf{x}}) - f(\mathbf{x}^*) \leq \epsilon$ with tuned constant learning rates. On *ring* topologies with uniform (1/3) gossip weights (and chains for RelaySum), all compared methods require steps at least linear in the number of workers to reach the target quality. RelaySGD and D^2 empirically scale significantly better than Gradient Tracking, these methods are all independent of data heterogeneity. On a *balanced binary tree network* with Metropolis-Hastings weights [Xiao and Boyd, 2004], both D^2 and Gradient Tracking notably do not scale better than on a ring, while RelaySGD on these trees requires only a number of steps logarithmic in the number of workers. SGP with their time-varying exponential topology scales well, too, but it requires more steps on more heterogeneously distributed data.

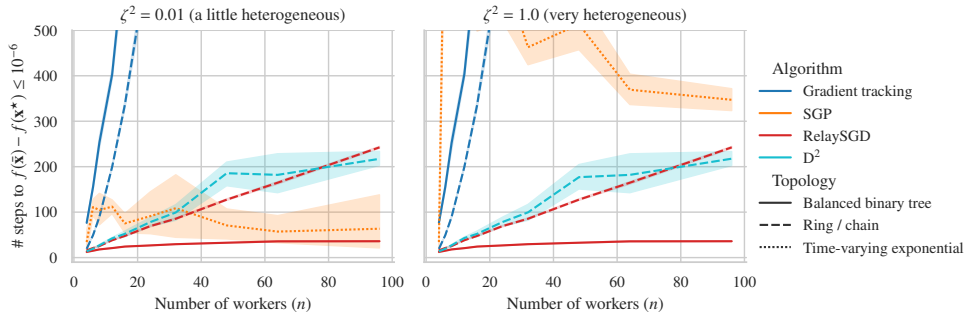


Fig. 5.2 Time required to optimize random quadratics ($\sigma^2 = 0, r_0 = 10, L = 1, \mu = 0.5$) to suboptimality $\leq 10^{-6}$ with varying numbers of workers with tuned constant learning rates. On a ring (---), \blacksquare D^2 and \blacksquare RelaySGD require steps linear in the number of workers, and this number is *independent of the data heterogeneity*. RelaySGD reduces this to $\log n$ on a balanced tree topology (—), but trees do not improve \blacksquare D^2 or \blacksquare Gradient Tracking. For \blacksquare SGP with time-varying exponential topology (.....), the number of steps does not consistently grow with more workers, but this number becomes higher with more heterogeneity (left v.s. right plot).

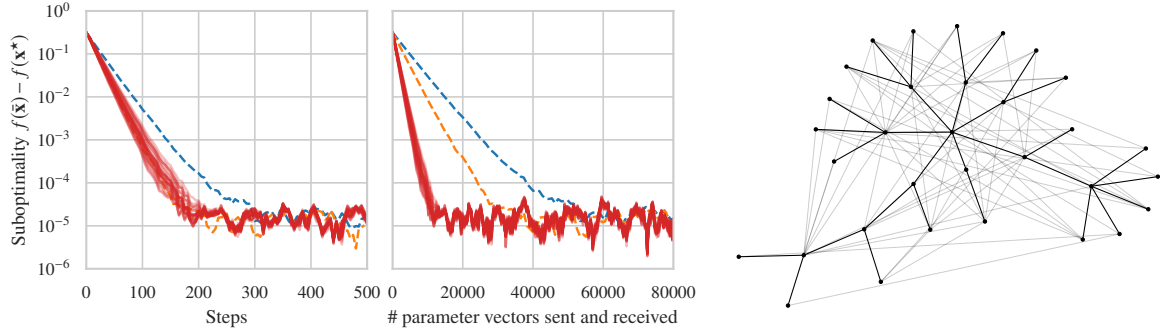


Fig. 5.3 Performance of ■ RelaySGD on spanning trees of the Social Network graph (32 nodes) found using Spanning Tree Protocol, compared to ■ DP-SGD and ■ D^2 on the full network. Solid lines (—) indicate spanning trees while dashed lines (---) indicate the full graph. The figure on the right shows one spanning tree on top of the original network. Learning rates are tuned to reach suboptimality $\leq 10^{-5}$ on random quadratics ($\zeta^2 = 0.1, \sigma^2 = 0.1, r_0 = 1, L = 1, \mu = 0.5$). ■ RelaySGD on spanning trees converges as fast as ■ D^2 on the full network, while the total communication on spanning trees is smaller than on the full graph.

5.6.2 Spanning trees compared to other topologies

RelaySGD cannot utilize all available edges in arbitrary networks to communicate, but is restricted to a spanning tree of the graph. We empirically find that this restriction is not limiting. In Figure 5.3, we take an organic social network topology based on the Davis Southern Women graph [Davis et al., 1930] from NetworkX [Hagberg et al., 2008b], and construct random spanning trees found by the Spanning Tree Protocol [Perlman, 1985]. On any such spanning tree, RelaySGD optimizes random heterogeneous quadratics as fast as D^2 on the full graph with Metropolis-Hastings weights [Xiao and Boyd, 2004], significantly faster than DP-SGD.

For decentralized learning used in a fully-connected data center for communication efficiency, the deep learning experiments below show that RelaySGD on double binary trees outperforms the most popular non-tree-based communication scheme used in decentralized deep learning [Assran et al., 2019a].

5.6.3 Effect of data heterogeneity in decentralized deep learning

We study the performance of RelaySGD in deep-learning based image- and text classification. While the algorithm is theoretically independent of dissimilarities in training data, other methods (D^2 , RelaySGD/Grad) that have the same property often lose accuracy in the presence of high data heterogeneity [Lin et al., 2021b]. To study the dependence of RelaySGD in practical deep learning, we partition training data strictly across 16 workers and distribute the classes using a Dirichlet process [Lin et al., 2021b; Yurochkin et al., 2019]. The Dirichlet parameter α controls the heterogeneity of the data across workers.

We compare RelaySGD against a variety of other algorithms. DP-SGD [Lian et al., 2017b] is the most natural combination of SGD with gossip averaging, and we chose D^2 [Tang et al., 2018] to represent the class of previous work that is theoretically robust to heterogeneity. We extend D^2 to allow varying step sizes and local momentum, according to Appendix D.4.4, and make it suitable for practical deep learning. Although Stochastic Gradient Push [Assran et al., 2019a] is not theoretically independent of data heterogeneity, it is a popular choice in the data center setting, where they use a time-varying exponential scheme on 2^d workers that mixes exactly uniformly in d rounds (Appendix D.4.6). We also compare to DP-SGD with quasi-global momentum [Lin et al., 2021b], a practical method recently introduced to increase robustness to heterogeneous data.

Table 5.1 evaluates RelaySGD in the fully-connected data center setting where we limit the communication budget per iteration to two models. We use 16-workers on Cifar-10, following the experimental details outlined in Appendix D.2 and hyper-parameter tuning procedure from Appendix D.3. For this experiment, we consider three topologies: (1) double binary trees as described in § 5.4, (2) rings, and (3) the time-varying exponential scheme of Stochastic Gradient Push (SGP) [Assran et al., 2019a]. Because SGP normally sends/receives only one model per communication round, we execute two synchronous communication steps per gradient update, increasing its latency. The various algorithms compared have different optimal topology choices. In Table 5.1 we only include the optimal choice for each algorithm. Table 5.2 qualitatively compares the possible combinations. We opt for the VGG-11 architecture because it does not feature BatchNorm Ioffe and Szegedy [2015]. BatchNorm poses particular challenges to data heterogeneity, and the search for alternatives is an active, and orthogonal, area of research [Liu et al., 2020].

Even though RelaySGD does not use a time-varying topology, it performs as well as or better than SGP, and RelaySGD with momentum suffers minimal accuracy loss up to heterogeneity $\alpha = 0.01$, a level higher than considered in previous work [Lin et al., 2021b]. While D^2 is theoretically independent of data heterogeneity, and while some of its random repetitions yield good results, it is unstable in the very heterogeneous setting. Moreover, Figure 5.4 shows that workers with RelaySGD achieve high test accuracies quicker during training than with other algorithms.

These findings are confirmed on ImageNet Deng et al. [2009] with the ResNet-20-EvoNorm architecture [Liu et al., 2020] in Table 5.3. On the BERT fine-tuning task from [Lin et al., 2021b], Table 5.4 demonstrates that RelaySGD with the Adam optimizer, customary for such NLP tasks, outperforms all compared algorithms.

5.6.4 Robustness to unreliable communication

Peer-to-peer applications are a central use case for decentralized learning. Decentralized learning algorithms must therefore be robust to workers joining and leaving, and to unreliable

Table 5.1 Cifar-10 [Krizhevsky \[2012\]](#) test accuracy with the VGG-11 architecture. We vary the data heterogeneity α [[Lin et al., 2021b](#)] between 16 workers. Each method sends/receives 2 models per iteration. We use a ring topology for DP-SGD and D² because they perform better on rings than on trees. RelaySum with momentum achieves the best results across all levels of data heterogeneity.

Algorithm	Topology (optimal c.f. Table 5.2)	$\alpha = 1.00$ (most homogeneous)	$\alpha = 0.1$	$\alpha = .01$ (most heterogeneous)
All-reduce (baseline) +momentum	fully connected	87.0% \rightarrow 90.2% \rightarrow	87.0% \rightarrow 90.2% \rightarrow	87.0% \rightarrow 90.2% \rightarrow
RelaySGD +local momentum	binary trees	87.4% \rightarrow 90.2% \rightarrow	86.9% \rightarrow 89.5% \rightarrow	84.6% \rightarrow 89.1% \rightarrow
DP-SGD * +quasi-global mom. [†]	ring	87.4% \rightarrow 89.5% \rightarrow	79.9% \rightarrow 84.8% \rightarrow	53.9% \rightarrow 63.3% \rightarrow
D ² ‡ +local momentum	ring	87.2% \rightarrow 88.2% \rightarrow	84.0% \rightarrow 88.5% \rightarrow	38.2% \rightarrow 61.0% \rightarrow
Stochastic gradient push ¶ +local momentum	time-varying exponential ¶	87.4% \rightarrow 89.5% \rightarrow	86.7% \rightarrow 89.2% \rightarrow	86.7% \rightarrow 87.5% \rightarrow

* DP-SGD [[Lian et al., 2017b](#)]

† DP-SGD +quasi-global mom. [[Lin et al., 2021b](#)]

‡ D² [[Tang et al., 2018](#)]

¶ Stochastic gradient push [[Assran et al., 2019a](#)]

Table 5.2 Motivation of topology choices. For each algorithm, we compare 4 topologies configured to send/receive 2 models at each SGD iteration. The algorithms have different optimal topologies.

Algorithm	Ring	Chain (= spanning tree of ring)	Double binary trees	Time-varying exponential ¶
RelaySGD	Unsupported	inferior (D.5.1)	Best result	Unsupported
DP-SGD	Best result	inferior	inferior (D.5.1)	Unsupported
D ²	Best result	inferior	inferior (D.5.1)	Unsupported
SGP	≈DP-SGD	≈DP-SGD	≈DP-SGD	Best result

¶ Stochastic gradient push [[Assran et al., 2019a](#)]

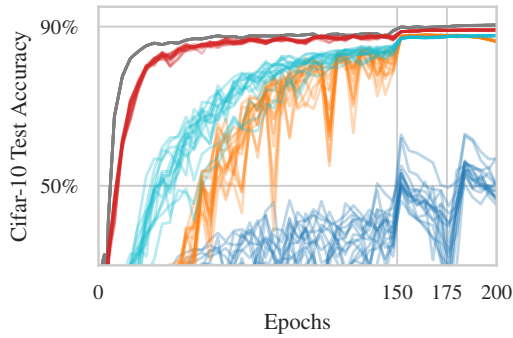


Fig. 5.4 Test accuracy during training of 16 workers with heterogeneous data ($\alpha = 0.01$) on Cifar-10. Like, with the \blacksquare all-reduce baseline, all workers in \blacksquare RelaySGD on double binary trees quickly reach good accuracy, while this takes longer for \blacksquare SGP with time-varying exponential topology and \blacksquare D² on a ring. \blacksquare DP-SGD does not reach good accuracy with such heterogeneous data.

Table 5.3 Test accuracies on ImageNet, using 16 workers with heterogeneous data ($\alpha = 0.1$). Even when communicating over a simple chain network, RelaySGD performs similarly to SGP with their time-varying exponential communicating scheme. Methods use default learning rates (Appendix D.3.2).

Algorithm	Topology	Top-1 Accuracy
Centralized (baseline)	fully-connected	69.7%
RelaySGD w/ momentum	double binary trees	60.0%
DP-SGD * w/ quasi-global momentum [†]	ring	55.8%
D ² ‡ w/ momentum	ring	diverged at epoch 65, at 49.5%
SGP ¶ w/ momentum	time-varying exponential ¶	58.5%

* DP-SGD [Lian et al., 2017b]

† DP-SGD +quasi-global mom. [Lin et al., 2021b]

‡ D² [Tang et al., 2018]

¶ Stochastic gradient push [Assran et al., 2019a]

Algorithm	Topology	Top-1 Accuracy
Centralized Adam	fully-connected	94.2% \pm 0.1%
Relay-Adam	double b. trees	93.2% \pm 0.6%
DP-SGD Adam	ring	87.3% \pm 0.6%
Quasi-global Adam [†]	ring	88.3% \pm 0.7%
SGP ¶ Adam	time-varying exp.	88.3% \pm 0.3%

† DP-SGD +quasi-global mom. [Lin et al., 2021b]

¶ Stochastic gradient push [Assran et al., 2019a]

Table 5.4 DistilBERT [Sanh et al., 2019] fine-tuning on AG news data [Zhang et al., 2015] using 16 nodes with heterogeneous data ($\alpha = 0.1$). Transformers are usually trained with Adam, and RelaySGD naturally supports Adam updates. (Appendix D.2.3).

Table 5.5 Robustness to unreliable networks. On Cifar-10/VGG-11 with 16 workers and heterogeneous data ($\alpha = 0.01$), we compare momentum versions of the best-performing algorithms from Table 5.1. RelaySGD with the robust update rule 5.1 can tolerate up to 10% dropped messages and converge to full test accuracy. Only SGP with the time-varying exponential scheme shares this property.

Algorithm	Topology	Reliable network	1% dropped msgs	10% dropped msgs
RelaySGD w/ momentum	trees	89.2%	89.3%	89.3%
DP-SGD * w/ quasi-global m. [†]	ring	69.3%	diverges	diverges
D ² ‡ w/ momentum	ring	87.4%	diverges	diverges
SGP ¶ w/ momentum	time-varying	88.5%	88.6%	88.1%

* DP-SGD [Lian et al., 2017b]

† DP-SGD +quasi-global mom. [Lin et al., 2021b]

‡ D² [Tang et al., 2018]

¶ Stochastic gradient push [Assran et al., 2019a]

communication between workers. Gossip averaging naturally features such robustness, but for methods like D^2 , that correct for local data biases, achieving such robustness is non-trivial. As a proxy for these challenges, in Table 5.5, we verify that RelaySGD can tolerate randomly dropped messages. The algorithm achieves this by reliably counting the number of models summed up in each message. For this experiment, we use an extended version of Algorithm 6, where line 10 is replaced by

$$\mathbf{x}_i^{(t+1)} = \frac{1}{n} \left(\mathbf{x}_i^{(t+1/2)} + \sum_{j \in \mathcal{N}_i} \mathbf{m}_{j \rightarrow i}^{(t)} + (n - \bar{n}_i^{(t+1)}) \mathbf{x}_i^{(t)} \right). \quad (5.1)$$

We count the number of models received as \bar{n} , and substitute any missing models ($< n$) by the previous state $\mathbf{x}_i^{(t)}$. RelaySGD trains reliably to good test accuracy with up to 10% deleted messages. This behavior is on par with a similarly modified SGP [Assran et al., 2019a] that corrects for missing energy. In contrast, D^2 and DP-SGD with quasi-global momentum are unstable with undelivered messages.

5.7 Conclusion

Decentralized learning has great promise as a building block in the democratization of deep learning. Deep learning relies on large datasets, and while large companies can afford those, many individuals together can, too. Of course, their data does not follow the exact same distribution, calling for robustness of decentralized learning algorithms to data heterogeneity. Algorithms with this property have been proposed and analyzed theoretically, but they do not always perform well in deep learning.

In this paper, we propose RelaySGD for distributed optimization over decentralized networks with heterogeneous data. Unlike algorithms based on gossip averaging, RelaySGD *relays* models through spanning trees of a network without decaying their magnitude. This yields an algorithm that is both theoretically independent of data heterogeneity, but also high performing in actual deep learning tasks. With its demonstrated robustness to unreliable communication, RelaySGD makes an attractive choice for peer-to-peer deep learning and applications in large-scale data centers.

Chapter 6

Debiasing Conditional Stochastic Optimization

6.1 Preface

Contribution and sources. This chapter reproduces [He and Kasiviswanathan \[2023\]](#). In this work, the central ideas and experimental frameworks were developed primarily by the author, with input and guidance from Shiva Prasad Kasiviswanathan. Detailed individual contributions:

- Lie He (author): Conceptualization, Writing (original draft preparation), Formal Analysis, Software.
- Shiva Prasad Kasiviswanathan: Conceptualization, Writing (original draft preparation), Formal Analysis, Supervision, Administration.

Summary. Conditional Stochastic Optimization (CSO) problem covers a wide range of bilevel optimization problems, including first order MAML, instrumental variable regression, etc. However, stochastic gradients of CSO problems are typically biased, which leads to much larger sample complexity than standard stochastic optimization to reach stationary point.

In this paper, we propose a novel extrapolation-based scheme to mitigate the bias in gradient estimations and propose new algorithms that incorporate this scheme, offering improved sample complexity for CSO problems. The theoretical foundation and practical applications of these methods are demonstrated with comprehensive data and experimental results.

6.2 Introduction

In this paper, we investigate the *conditional stochastic optimization* (CSO) problem as presented by [Hu et al. \[2020b\]](#), which is formulated as follows:

$$\min_{\mathbf{x} \in \mathbb{R}^d} F(\mathbf{x}) = \mathbb{E}_{\xi}[f_{\xi}(\mathbb{E}_{\eta|\xi}[g_{\eta}(\mathbf{x}; \xi)])], \quad (\text{CSO})$$

where ξ and η represent two random variables, with η conditioned on ξ . The $f_\xi : \mathbb{R}^p \rightarrow \mathbb{R}$ and $g_\eta : \mathbb{R}^d \rightarrow \mathbb{R}^p$ denote a stochastic function and a mapping respectively. The inner expectation is calculated with respect to the conditional distribution of $\eta|\xi$. In line with the established CSO framework [Hu et al., 2020a,b], throughout this paper, we assume access to samples from the distribution $\mathbb{P}(\xi)$ and the conditional distribution $\mathbb{P}(\eta|\xi)$.

Many machine learning tasks can be formulated as a CSO problem, such as policy evaluation and control in reinforcement learning [Dai et al., 2018; Nachum and Dai, 2020], and linearly-solvable Markov decision process [Dai et al., 2017]. Other examples of the CSO problem include instrumental variable regression [Muandet et al., 2020] and invariant learning [Hu et al., 2020b]. Moreover, the widely-used Model-Agnostic Meta-Learning (MAML) framework, which seeks to determine a meta-initialization parameter using metadata for related learning tasks that are trained through gradient-based algorithms, is another example of a CSO problem. In this context, tasks ξ are drawn randomly, followed by the drawing of samples $\eta|\xi$ from the specified task [Finn et al., 2017]. It is noteworthy that the standard stochastic optimization problem $\min_{\mathbf{x}} \mathbb{E}_\xi[f_\xi(\mathbf{x})]$ represents a degenerate case of the CSO problem, achieved by setting g_η as an identity function.

In numerous prevalent CSO problems, such as first-order MAML (FO-MAML) [Finn et al., 2017], the outer random variable ξ only takes value in a finite set (say in $\{1, \dots, n\}$). These problems can be reformulated to have a finite-sum structure in the outer loop and referred to as *Finite-sum Coupled Compositional Optimization (FCCO)* problem in [Jiang et al., 2022; Wang and Yang, 2022]. In this paper, we also study this problem, formulated as:

$$\min_{\mathbf{x} \in \mathbb{R}^d} F_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n f_i(\mathbb{E}_{\eta|i}[g_\eta(\mathbf{x}; i)]). \quad (\text{FCCO})$$

The FCCO problem also has broad applications in machine learning for optimizing average precision, listwise ranking losses, neighborhood component analysis, deep survival analysis, deep latent variable models [Jiang et al., 2022; Wang and Yang, 2022].

Although the CSO and FCCO problems are widespread, they present challenges for optimization algorithms. Based on the special composition structure of CSO, using chain rule, under mild conditions, the full gradient of CSO is given by

$$\nabla F(\mathbf{x}) = \mathbb{E}_\xi \left[\left(\mathbb{E}_{\eta|\xi}[\nabla g_\eta(\mathbf{x}; \xi)] \right)^\top \nabla f_\xi(\mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}; \xi)]) \right].$$

Constructing an unbiased stochastic estimator for the gradient is generally computationally expensive (and even impossible). A straightforward estimation of $\nabla F(\mathbf{x})$ is to estimate \mathbb{E}_ξ with 1 sample of ξ , estimate $\mathbb{E}_{\eta|\xi}[g_\eta(\cdot)]$ with a set H_ξ of m independent and identically distributed (i.i.d.) samples drawn from the conditional distribution $\mathbb{P}(\eta|\xi)$, and $\mathbb{E}_{\eta|\xi}[\nabla g_\eta(\cdot)]$ with a different

set \tilde{H}_ξ of m i.i.d. samples drawn from the same conditional distribution, i.e.,

$$\nabla \hat{F}_m(\mathbf{x}) := \left(\frac{1}{m} \sum_{\tilde{\eta} \in \tilde{H}_\xi} \nabla g_{\tilde{\eta}}(\mathbf{x}; \xi) \right)^\top \nabla f_\xi \left(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}; \xi) \right). \quad (6.1)$$

Note that $\nabla \hat{F}_m(\mathbf{x})$ consists of two terms. The first term, $(1/m) \sum_{\tilde{\eta} \in \tilde{H}_\xi} \nabla g_{\tilde{\eta}}(\mathbf{x}; \xi)$, is an unbiased estimate of $\mathbb{E}_{\eta|\xi}[\nabla g_\eta(\mathbf{x}; \xi)]$. However, the second term is generally biased, i.e.,

$$\mathbb{E}_{\eta|\xi}[\nabla f_\xi(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}; \xi))] \neq \nabla f_\xi(\mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}; \xi)]).$$

Consequently, $\nabla \hat{F}_m(\mathbf{x})$ is a biased estimator of $\nabla F(\mathbf{x})$. To reach the ϵ -stationary point of $F(\mathbf{x})$ (Definition 6.1), the bias has to be sufficiently small.

Optimization with biased gradients converges only to a neighborhood of the stationary point. While the bias diminishes with increasing batch size, it also introduces additional sample complexity. For nonconvex objectives, Biased Stochastic Gradient Descent (BSGD) requires a total sample complexity of $\mathcal{O}(\epsilon^{-6})$ to reach an ϵ -stationary point [Hu et al., 2020b]. This contrasts with standard stochastic optimization, where sample-averaged gradients are unbiased with a sample complexity of $\mathcal{O}(\epsilon^{-4})$ [Arjevani et al., 2022; Ghadimi and Lan, 2013]. This discrepancy has spurred a multitude of proposals aimed at reducing the sample complexities of both CSO and FCCO problems. Hu et al. [2020b] introduced Biased SpiderBoost (BSpiderBoost), which, based on the variance reduction technique SpiderBoost from Wang et al. [2019], reduces the variance of ξ to achieve a sample complexity of $\mathcal{O}(\epsilon^{-5})$ for the CSO problem. Hu et al. [2021] proposed multi-level Monte Carlo (MLMC) gradient methods V-MLMC and RT-MLMC to further enhance the sample complexity to $\mathcal{O}(\epsilon^{-4})$. The SOX [Wang and Yang, 2022] and MSVR-V2 [Jiang et al., 2022] algorithms concentrated on the FCCO problem and improved the sample complexity to $\mathcal{O}(n\epsilon^{-4})$ and $\mathcal{O}(n\epsilon^{-3})$, respectively.

Our Contributions. In this paper, we improve the sample complexities for both the CSO and FCCO problems (see Table 6.1). To facilitate a clear and concise presentation, we will suppress the dependence on specific problem parameters throughout the ensuing discussion.

- (a) Our main technical tool in this paper is an extrapolation-based scheme that mitigates bias in gradient estimations. Considering a suitably differentiable function $q(\cdot)$ and a random variable $\delta \sim \mathcal{D}$, we show that we can approximate the value of $q(\mathbb{E}[\delta])$ via extrapolation from a limited number of evaluations of $q(\delta)$, while maintaining a minimal bias. In the context of CSO and FCCO problems, this scheme is used in gradient estimation, where the function q corresponds to ∇f_ξ and the random variable δ corresponds to g_η .
- (b) For the CSO problem, we present novel algorithms that integrate the above extrapolation-based scheme with BSGD and BSpiderBoost algorithms of Hu et al. [2020b]. Our algorithms, referred to as E-BSGD and E-BSpiderBoost, achieve a sample complexity of $\mathcal{O}(\epsilon^{-4.5})$ and $\mathcal{O}(\epsilon^{-3.5})$ respectively, in order to attain an ϵ -stationary point for nonconvex smooth objec-

Problem	Old Bounds		Our Bounds	
	Algorithm	Bound	Algorithm	Bound
CSO	BSGD [Hu et al., 2020b]	$\mathcal{O}(\epsilon^{-6})$	E-BSGD	$\mathcal{O}(\epsilon^{-4.5})$
CSO	BSpiderBoost [Hu et al., 2020b]	$\mathcal{O}(\epsilon^{-5})$	E-BSpiderBoost	$\mathcal{O}(\epsilon^{-3.5})$
CSO	RT-MLMC [Hu et al., 2021]	$\mathcal{O}(\epsilon^{-4})$		
FCCO	MSVR-V2 [Jiang et al., 2022]	$\mathcal{O}(n\epsilon^{-3})$	E-NestedVR	$\begin{cases} \mathcal{O}(n\epsilon^{-3}) & \text{if } n \leq \epsilon^{-2/3} \\ \mathcal{O}(\max\{\frac{\sqrt{n}}{\epsilon^{2.5}}, \frac{1}{\sqrt{n\epsilon^4}}\}) & \text{if } n > \epsilon^{-2/3} \end{cases}$

Table 6.1 Sample complexities needed to reach ϵ -stationary point for FCCO and CSO problems with nonconvex smooth objectives. Assumptions are comparable, but our results require an additional mild regularity on f_ξ and g_η . For FCCO also see Footnote 1. Note that $\Omega(\epsilon^{-3})$ is a sample complexity lower bound for standard stochastic nonconvex optimization [Arjevani et al., 2022], and hence, also for the problems considered in this paper.

tives. Notably, the sample complexity of E-BSpiderBoost improves the best-known sample complexity of $\mathcal{O}(n\epsilon^{-4})$ for the CSO problem from Hu et al. [2021].

- (c) For the FCCO problem¹ we propose a new algorithm that again combines the extrapolation-based scheme with a multi-level variance reduction applied to both inner and outer parts of the problem. Our algorithm, referred to as E-NestedVR, achieves a sample complexity of $\mathcal{O}(n\epsilon^{-3})$ if $n \leq \epsilon^{-2/3}$ and $\mathcal{O}(\max\{\sqrt{n}\epsilon^{-2.5}, \epsilon^{-4}/\sqrt{n}\})$ if $n > \epsilon^{-2/3}$ for nonconvex smooth objectives and second-order extrapolation scheme. Our bound is never worse than the $\mathcal{O}(n\epsilon^{-3})$ bound of MSVR-V2 algorithm of Jiang et al. [2022] and is in fact better if $n = \Omega(\epsilon^{-2/3})$. As an illustration, when $n = \Theta(\epsilon^{-1.5})$, our bound of $\mathcal{O}(\epsilon^{-3.25})$ is significantly better than the MSVR-V2 bound of $\mathcal{O}(\epsilon^{-4.5})$.

In terms of proof techniques, our approach diverges from conventional analyses for the CSO and FCCO problems in that we focus on explicitly bounding the bias and variance terms of the gradient estimator to establish the convergence guarantee. Compared to previous results, our improvements do require an additional mild regularity assumption on f_ξ and g_η mainly that ∇f_ξ is 4th order differentiable. Firstly, as we discuss in Remark 2 most common instantiations of CSO/FCCO framework such as: 1) invariant logistic regression Hu et al. [2020b], 2) instrumental variable regression [Muandet et al., 2020], 3) first-order MAML for sine-wave few shot regression [Finn et al., 2017] and other problems, 4) deep average precision maximization [Qi et al., 2021a; Wang et al., 2022a], tend to satisfy this assumption. Secondly, we highlight that the bounds derived from previous studies do not improve when incorporating this additional regularity assumption. Thirdly, $\Omega(\epsilon^{-3})$ remains the lower bound for stochastic optimization even under the arbitrary smoothness constraint [Arjevani et al., 2020], demonstrating that our improvement is non-trivial. Our results show that, this regularity assumption, which seems to

¹ For the FCCO problem we focus on $n = \mathcal{O}(\epsilon^{-2})$ case, for $n = \Omega(\epsilon^{-2})$ we can just treat the FCCO problem as a CSO problem and get an $\mathcal{O}(\epsilon^{-3.5})$ sample complexity bound via our E-BSpiderBoost algorithm.

practically valid, can be exploited through a novel extrapolation-based bias reduction technique to provide substantial improvements in sample complexity.²

We defer some additional related work to Appendix E.2 and conclude with some preliminaries.

Notation. Vectors are denoted by boldface letters. For a vector \mathbf{x} , $\|\mathbf{x}\|_2$ denotes its ℓ_2 -norm. A function with k continuous derivatives is called a \mathcal{C}^k function. We use $a \lesssim b$ to denote that $a \leq Cb$ for some constant $C > 0$. We consider expectation over various randomness: $\mathbb{E}_\xi[\cdot]$ denotes expectation over the random variable ξ , $\mathbb{E}_{\eta|\xi}[\cdot]$ denotes expectation over the conditional distribution of $\eta|\xi$. Unless otherwise specified, for a random variable X , $\mathbb{E}[X]$ denotes expectation over the randomness in X . We focus on nonconvex objectives in this paper and use the following standard convergence criterion for nonconvex optimization [Jain et al., 2017].

Definition 6.1 (ϵ -stationary point). *For a differentiable function $F(\cdot)$, we say that \mathbf{x} is a first-order ϵ -stationary point if $\|\nabla F(\mathbf{x})\|^2 \leq \epsilon^2$.*

For notational convenience, in the rest of this paper, we omit the dependence on ξ (or i in the FCCO context) in the function g and use $g_\eta(\mathbf{x})$ to represent $g_\eta(\mathbf{x}; \xi)$.

6.3 Stochastic Extrapolation as a Tool for Bias Correction

In this section, we present an approach for tackling the bias problem as appears in optimization procedures such as BSGD, BSpiderBoost, etc. Importantly, our approach addresses a general problem appearing in optimization settings and could be of independent interest. All missing details from this section are presented in § E.3.

For ease of presentation, we start by considering the 1-dimensional case and assume a function $q : \mathbb{R} \rightarrow \mathbb{R}$, a constant $s \in \mathbb{R}$. Let δ be a random variable drawn from an arbitrary distribution \mathcal{D} over \mathbb{R} . In Sections 6.4 and 6.5, we apply these ideas to the CSO and FCCO problems where the random variable δ is played by $g_\eta(\cdot)$ and function q is played by ∇f_ξ . Informally stated, our goal in this section will be to

Efficiently approximate $q(s + \mathbb{E}[\delta])$ with few evaluations of $\{q(s + \delta)\}_{\delta \sim \mathcal{D}}$.

An interesting case is when $s = 0$, where we are approximating $q(\mathbb{E}[\delta])$ with evaluations of $\{q(\delta)\}_{\delta \sim \mathcal{D}}$. Now, if q is an affine function, then $q(s + \mathbb{E}[\delta]) = \mathbb{E}[q(s + \delta)]$. However, the equality does not hold true for general q , and there exists a bias, i.e., $|q(s + \mathbb{E}[\delta]) - \mathbb{E}[q(s + \delta)]| > 0$. In this section, we introduce a stochastic *extrapolation*-based method, where we use an affine combination of biased stochastic estimates, to achieve better approximation.

Suppose $q \in \mathcal{C}^{2k}$ is a continuous differentiable up to $2k$ -th derivative and let $h = \mathbb{E}[\delta]$. We expand $q(s + \delta)$, the most straightforward approximation of $q(s + \mathbb{E}[\delta])$, using Taylor series at

²Higher-order smoothness conditions have also been exploited in standard stochastic optimization for performance gains [Bubeck et al., 2019].

$s + h$, and take expectation,

$$\begin{aligned} \mathbb{E}[q(s + \delta)] = & q(s + h) + q'(s + h) \mathbb{E}[\delta - h] + \frac{q''(s+h)}{2} \mathbb{E}[(\delta - h)^2] + \frac{q^{(3)}(s+h)}{6} \mathbb{E}[(\delta - h)^3] \\ & + \dots + \frac{q^{(2k-1)}(s+h)}{(2k-1)!} \mathbb{E}[(\delta - h)^{(2k-1)}] + \frac{1}{(2k)!} \mathbb{E}[q^{(2k)}(\phi_\delta)(\delta - h)^{2k}], \end{aligned} \quad (6.2)$$

where ϕ_δ between $s + \delta$ and $s + h$. While $\mathbb{E}[q(s + \delta)]$ matches $q(s + h)$ in the first 2 terms, the third term is no longer zero. The approximation error (bias) is

$$|\mathbb{E}[q(s + \delta)] - q(s + h)| = \left| \frac{q''(s+h)}{2} \mathbb{E}[(\delta - h)^2] + \dots + \frac{1}{(2k)!} \mathbb{E}[q^{(2k)}(\phi_\delta)(\delta - h)^{2k}] \right|.$$

In order to analyze the upper bound, we make the following assumption on \mathcal{D} and q .

Assumption B (Bounded moments). *For all $\delta \sim \mathcal{D}$ has bounded higher-order moments: $\sigma_l := |\mathbb{E}[(\delta - \mathbb{E}[\delta])^l]| < \infty$ for $l = 2, 3, \dots, 2k$.*

Assumption C (Bounded derivatives). *The $q \in \mathcal{C}^{2k}$ and has bounded derivatives, i.e., $a_l := \sup_{s \in \text{dom}(q)} |q^{(l)}(s)| < \infty$ for $l = 1, 2, \dots, 2k$.*

In addition, we consider a sample averaged distribution \mathcal{D}_m derived from \mathcal{D} as follows.

Definition 6.4. *Given a distribution \mathcal{D} satisfying Assumption B and $m \in \mathbb{N}^+$, we define the distribution \mathcal{D}_m that outputs δ where $\delta = \frac{1}{m} \sum_{i=1}^m \delta_i$ with $\delta_i \stackrel{i.i.d.}{\sim} \mathcal{D}$.*

The moments of such distribution \mathcal{D}_m decrease with batch size m as $k \geq 2$, $|\mathbb{E}[(\delta - \mathbb{E}[\delta])^k]| = \mathcal{O}(m^{-\lceil k/2 \rceil})$ (see Lemma E.1). Our desiderata would be to construct a scheme that uses some samples from the distribution \mathcal{D}_m to construct an approximation of $q(s + \mathbb{E}[\delta])$ that satisfies the following requirement.

Definition 6.5 (*k*th-order Extrapolation Operator). *Given a function $q : \mathbb{R} \rightarrow \mathbb{R}$ satisfying Assumption C and distribution \mathcal{D}_m satisfying Assumption B, we define a *k*th-order extrapolation operator $\mathcal{T}_{\mathcal{D}_m}^{(k)}$ as an operator from $\mathcal{C}^{2k} \rightarrow \mathcal{C}^{2k}$ that given $N = N(k)$ i.i.d. samples $\delta_1, \dots, \delta_N$ from \mathcal{D}_m satisfies $\forall s \in \mathbb{R}: |\mathbb{E}[\mathcal{T}_{\mathcal{D}_m}^{(k)} q(s)] - q(s + \mathbb{E}[\delta])| = \mathcal{O}(m^{-k})$.*

We now propose a sequence of operators $\mathcal{L}_{\mathcal{D}_m}^{(1)}, \mathcal{L}_{\mathcal{D}_m}^{(2)}, \mathcal{L}_{\mathcal{D}_m}^{(3)}, \dots$ that satisfy the above definition. The $\mathcal{L}_{\mathcal{D}_m}^{(k)} q(s)$ is designed to ensure its Taylor expansion at $s + h$ has a form of $q(s + h) + \mathcal{O}(\mathbb{E}[(\delta - h)^{2k}])$. The remainder $\mathcal{O}(\mathbb{E}[(\delta - h)^{2k}])$ is bounded by $\mathcal{O}(m^{-k})$ due to Lemma E.1.

A First-order Extrapolation Operator. We define the simplest operator

$$\mathcal{L}_{\mathcal{D}_m}^{(1)} q : s \mapsto [q(s + \delta)] \quad \text{where } \delta \stackrel{i.i.d.}{\sim} \mathcal{D}_m.$$

In Proposition E.1 (Appendix E.3), we show that $\mathcal{L}_{\mathcal{D}_m}^{(1)}$ is a first-order extrapolation operator.³

³Note that if the function q is only L_q -Lipschitz continuous, then $|\mathbb{E}[q(s + \delta)] - q(s + \mathbb{E}[\delta])| \leq \sqrt{L_q^2 \mathbb{E}[(\delta - \mathbb{E}[\delta])^2]} \leq \frac{L_q \sqrt{\sigma_2}}{m^{1/2}}$. Therefore, in this case, $q(s + \delta)$ does not satisfy the first-order guarantee.

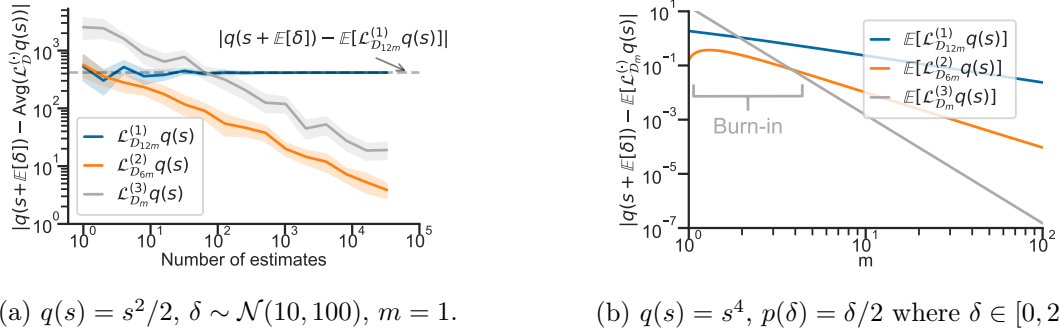


Fig. 6.1 The Fig. 6.1a investigates the estimation errors of $\mathcal{L}^{(\cdot)}q(s)$ with their number of observations. The Fig. 6.1b compares the biases of $\mathbb{E}[\mathcal{L}^{(\cdot)}q(s)]$ with increasing inner batch size m .

A Second-order Extrapolation Operator. We define the following linear operator $\mathcal{L}_{\mathcal{D}_m}^{(2)}$ which transforms $q \in \mathcal{C}^4$ into $\mathcal{L}_{\mathcal{D}_m}^{(2)}q$ which has lesser bias (but similar variance, as shown later).

Definition 6.6 ($\mathcal{L}_{\mathcal{D}_m}^{(2)}$ Operator). *Given \mathcal{D}_m and q , define the following operator,*

$$\mathcal{L}_{\mathcal{D}_m}^{(2)}q : s \mapsto \left[2 \cdot q\left(s + \frac{\delta_1 + \delta_2}{2}\right) - \frac{q(s + \delta_1) + q(s + \delta_2)}{2} \right] \quad \text{where } \delta_1, \delta_2 \stackrel{i.i.d.}{\sim} \mathcal{D}_m.$$

Note that $\frac{\delta_1 + \delta_2}{2}$ is same as sampling from \mathcal{D}_{2m} . The absolute difference in the Taylor expansion of $\mathcal{L}_{\mathcal{D}_m}^{(2)}q$ at $s + h$ differs from $q(s + h)$ as,

$$\mathcal{O}\left(\left|\mathbb{E}\left[2\left(\frac{\delta_1 + \delta_2}{2} - h\right)^3 - \frac{1}{2}((\delta_1 - h)^3 + (\delta_2 - h)^3)\right]\right|\right) = \mathcal{O}(|\mathbb{E}[(\delta - h)^3]|) \text{ for } \delta \stackrel{i.i.d.}{\sim} \mathcal{D}_m. \quad (6.3)$$

The bias error of this scheme can be bounded through the following proposition.

Proposition 6.1 (Second-order Guarantee). *Assume that distribution \mathcal{D}_m and $q(\cdot)$ satisfies Assumption B and C respectively with $k = 2$. Then, for all $s \in \mathbb{R}$, $\left|\mathbb{E}\left[\mathcal{L}_{\mathcal{D}_m}^{(2)}q(s)\right] - q(s + \mathbb{E}[\delta])\right| \leq \frac{4a_3\sigma_3 + 9a_4\sigma_2^2}{48m^2} + \frac{5a_4}{96} \frac{\sigma_4 - 3\sigma_2^2}{m^3}$.*

Remark 1. While extrapolation is motivated by Taylor expansion which requires smoothness, higher order derivatives are not explicitly computed. Appendix E.6.3 empirically shows that applying extrapolation to non-smooth functions achieves similar bias correction. Relaxing the smoothness conditions is a direction for future work.

The above proposition shows that $\mathcal{L}_{\mathcal{D}_m}^{(2)}$ is in fact a second-order extrapolation operator with $k = 2$ under Definition 6.5. We will use this operator when we consider the CSO and FCCO problems later. Now, focusing on variance, we can relate the variance of $\mathcal{L}_{\mathcal{D}_m}^{(2)}q(s)$ in terms of the variance of $q(s + \delta)$. In particular, a consequence of Lemma E.2 is that

$$\mathbb{E}\left[\left(\mathcal{L}_{\mathcal{D}_m}^{(2)}q(s) - \mathbb{E}[\mathcal{L}_{\mathcal{D}_m}^{(2)}q(s)]\right)^2\right] = \mathcal{O}(\mathbb{E}[(q(s + \delta) - \mathbb{E}[q(s + \delta)])^2]).$$

Extension of $\mathcal{L}_{\mathcal{D}_m}^{(2)}$ to Higher-dimensional Case. If $q : \mathbb{R}^p \rightarrow \mathbb{R}^\ell$ is a vector-valued function, then there is a straightforward extension of Definition 6.6. Now, for distribution \mathcal{D} over \mathbb{R}^p and corresponding sampled averaged distribution \mathcal{D}_m , and $\mathbf{s} \in \mathbb{R}^p$

$$\mathcal{L}_{\mathcal{D}_m}^{(2)} q : \mathbf{s} \mapsto \left[2 \cdot q\left(\mathbf{s} + \frac{\boldsymbol{\delta}_1 + \boldsymbol{\delta}_2}{2}\right) - \frac{q(\mathbf{s} + \boldsymbol{\delta}_1) + q(\mathbf{s} + \boldsymbol{\delta}_2)}{2} \right] \quad \text{where } \boldsymbol{\delta}_1, \boldsymbol{\delta}_2 \stackrel{\text{i.i.d.}}{\sim} \mathcal{D}_m. \quad (6.4)$$

Higher-order Extrapolation Operators. The idea behind the construction of $\mathcal{L}_{\mathcal{D}_m}^{(2)}$ can be generalized to higher k 's. For example, in Proposition E.2, we construct a third-order extrapolation operator $\mathcal{L}_{\mathcal{D}_m}^{(3)}$ through higher degree Taylor series approximation

$$\mathcal{L}_{\mathcal{D}_m}^{(3)} q : \mathbf{s} \mapsto \left(-\frac{1}{36} \mathcal{L}_{\mathcal{D}_m}^{(2)} + \frac{5}{9} \mathcal{L}_{\mathcal{D}_{2m}}^{(2)} - \frac{3}{4} \mathcal{L}_{\mathcal{D}_{3m}}^{(2)} - \frac{16}{9} \mathcal{L}_{\mathcal{D}_{4m}}^{(2)} + 3 \mathcal{L}_{\mathcal{D}_{6m}}^{(2)} \right) q(\mathbf{s}).$$

While this idea of expressing the k -th order operator as an affine combination of lower-order operators works for every k , explicit constructions soon become tedious.

In Fig. 6.1, we empirically demonstrate the effectiveness of extrapolation in stochastic estimation.⁴ In Fig. 6.1a, we choose $q(s) = s^2/2$, $\delta \sim \mathcal{N}(10, 100)$. For both $\mathcal{L}_{\mathcal{D}_{6m}}^{(2)} q(s)$ and $\mathcal{L}_{\mathcal{D}_m}^{(3)} q(s)$, their estimation errors converge to 0 with increasing number of estimates. This coincides with Proposition 6.1 as $a_3 = 0$ and $a_4 = 0$ for quadratic q . In contrast, biased first order method only converges to a neighborhood. In Fig. 6.1b, we consider $q(s) = s^4$ and $p(\delta) = \delta/2$ where $\delta \in [0, 2]$. All three methods are biased and their biases decrease with m , i.e. $\mathcal{O}(m^{-k})$ for k th order method. Depending on the constants (e.g. a_i, σ_i), a higher-order extrapolation method may need decently large m (burn-in phase) to outperform lower-order methods.

6.4 Applying Stochastic Extrapolation in the CSO Problem

In this section, we apply the extrapolation-based scheme from the previous section to reduce the bias in the CSO problem. We focus on variants of BSGD and their accelerated version BSpiderBoost based on our second-order approximation operator (Definition 6.6). Let H_ξ, \tilde{H}_ξ , and H'_ξ indicate different sets, each of which contains m i.i.d. random variables/samples drawn from the conditional distribution $\mathbb{P}(\eta|\xi)$. Remember that, as mentioned earlier, we use $g_\eta(\mathbf{x})$ to represent $g_\eta(\mathbf{x}; \xi)$.

Extrapolated BSGD. At time t , BSGD constructs a biased estimator of $\nabla F(\mathbf{x}^t)$ using one sample ξ and $2m$ i.i.d. samples from the conditional distribution as in (6.1)

$$G_{\text{BSGD}}^{t+1} = \left(\frac{1}{m} \sum_{\tilde{\eta} \in \tilde{H}_\xi} \nabla g_{\tilde{\eta}}(\mathbf{x}^t) \right)^\top \nabla f_\xi \left(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t) \right). \quad (6.5)$$

⁴We use $\mathcal{L}_{\mathcal{D}_{12m}}^{(1)}, \mathcal{L}_{\mathcal{D}_{6m}}^{(2)}, \mathcal{L}_{\mathcal{D}_m}^{(3)}$ to ensure that each estimate uses same amount of samples (12m).

To reduce this bias, we apply the second-order extrapolation operator from (6.4). At time t , we define $\mathcal{D}_{\mathbf{g},\xi}^{t+1}$ to be the distribution of the random variable $\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t)$. Then we apply $\mathcal{L}_{\mathcal{D}_{\mathbf{g},\xi}^{t+1}}^{(2)}$ by setting q to ∇f_ξ and $\mathbf{s} = 0$, i.e.

$$\begin{aligned} \mathcal{L}_{\mathcal{D}_{\mathbf{g},\xi}^{t+1}}^{(2)} \nabla f_\xi(0) &:= 2\nabla f_\xi \left(\frac{1}{2m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t) + \frac{1}{2m} \sum_{\eta' \in H'_\xi} g_{\eta'}(\mathbf{x}^t) \right) \\ &\quad - \frac{1}{2} \left(\nabla f_\xi \left(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t) \right) + \nabla f_\xi \left(\frac{1}{m} \sum_{\eta' \in H'_\xi} g_{\eta'}(\mathbf{x}^t) \right) \right), \end{aligned} \quad (6.6)$$

where $\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t)$ and $\frac{1}{m} \sum_{\eta' \in H'_\xi} g_{\eta'}(\mathbf{x}^t)$ are i.i.d. drawn from $\mathcal{D}_{\mathbf{g},\xi}^{t+1}$. In Algorithm 16 (Appendix E.1), we present our extrapolated BSGD (E-BSGD) scheme, where we replace $\nabla f_\xi(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t))$ in (6.5) by $\mathcal{L}_{\mathcal{D}_{\mathbf{g},\xi}^{t+1}}^{(2)} \nabla f_\xi(0)$ resulting in this following gradient estimate:

$$G_{\text{E-BSGD}}^{t+1} = \left(\frac{1}{m} \sum_{\tilde{\eta} \in \tilde{H}_\xi} \nabla g_{\tilde{\eta}}(\mathbf{x}^t) \right)^\top \mathcal{L}_{\mathcal{D}_{\mathbf{g},\xi}^{t+1}}^{(2)} \nabla f_\xi(0). \quad (6.7)$$

Extrapolated BSpiderBoost. BSpiderBoost, proposed by Hu et al. [2020b], uses the variance reduction methods for nonconvex smooth stochastic optimization developed by Fang et al. [2018]; Wang et al. [2019]. BSpiderBoost builds upon BSGD and has two kinds of updates: a large batch and a small batch update. In each step, it decides which update to apply based on a random coin. With probability p_{out} , it selects a large batch update with B_1 outer samples of ξ . With remaining probability $1 - p_{\text{out}}$, it selects a small batch update where the gradient estimator will be updated with gradient information in the current iteration generated with B_2 outer samples of ξ and the information from the last iteration. Formally, it constructs a gradient estimate as follows,

$$G_{\text{BSB}}^{t+1} = \begin{cases} G_{\text{BSB}}^t + \frac{1}{B_2} \sum_{\xi \in \mathcal{B}_2, |\mathcal{B}_2|=B_2} (G_{\text{BSGD}}^{t+1} - G_{\text{BSGD}}^t) & \text{with prob. } 1 - p_{\text{out}} \\ \frac{1}{B_1} \sum_{\xi \in \mathcal{B}_1, |\mathcal{B}_1|=B_1} G_{\text{BSGD}}^{t+1} & \text{with prob. } p_{\text{out}}. \end{cases} \quad (6.8)$$

We propose our extrapolated BSpiderBoost scheme (formally defined in Algorithm 17, Appendix E.1) by replacing the BSGD gradient estimates in (6.8) with E-BSGD.

$$G_{\text{E-BSB}}^{t+1} = \begin{cases} G_{\text{E-BSB}}^t + \frac{1}{B_2} \sum_{\xi \in \mathcal{B}_2, |\mathcal{B}_2|=B_2} (G_{\text{E-BSGD}}^{t+1} - G_{\text{E-BSGD}}^t) & \text{with prob. } 1 - p_{\text{out}} \\ \frac{1}{B_1} \sum_{\xi \in \mathcal{B}_1, |\mathcal{B}_1|=B_1} G_{\text{E-BSGD}}^{t+1} & \text{with prob. } p_{\text{out}}. \end{cases} \quad (6.9)$$

Sample Complexity Analyses of E-BSGD and E-BSpiderBoost. We adopt the standard assumptions used in the literature [Qi et al., 2021b; Wang and Yang, 2022; Wang et al., 2022b; Zhang and Xiao, 2021]. All proofs are deferred to § E.4.

Assumption G (Lower bound). F is lower bounded by F^* .

Assumption H (Bounded variance). Assume that g_η and ∇g_η have bounded variances, i.e., for all ξ in the support of $\mathbb{P}(\xi)$ and $\mathbf{x} \in \mathbb{R}^p$, $\sigma_g^2 := \mathbb{E}_{\eta|\xi}[\|g_\eta(\mathbf{x}; \xi) - \mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}; \xi)]\|_2^2] < \infty$ and $\zeta_g^2 := \mathbb{E}_{\eta|\xi}[\|\nabla g_\eta(\mathbf{x}; \xi) - \mathbb{E}_{\eta|\xi}[\nabla g_\eta(\mathbf{x}; \xi)]\|_2^2] < \infty$.

Assumption I (Lipschitz continuity/smoothness of f_ξ and g_η). For all ξ in the support of $\mathbb{P}(\xi)$, $f_\xi(\cdot)$ is C_f -Lipschitz continuous (i.e., $\|f_\xi(\mathbf{x}) - f_\xi(\mathbf{x}')\|_2 \leq C_f \|\mathbf{x} - \mathbf{x}'\|_2 \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^p$) and L_f -Lipschitz smooth (i.e., $\|\nabla f_\xi(\mathbf{x}) - \nabla f_\xi(\mathbf{x}')\|_2 \leq L_f \|\mathbf{x} - \mathbf{x}'\|_2, \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^p$) for any ξ . Similarly, for all ξ in the support of $\mathbb{P}(\xi)$ and η in the support of $\mathbb{P}(\eta|\xi)$, $g_\eta(\cdot; \xi)$ is C_g -Lipschitz continuous and L_g -Lipschitz smooth.

The smoothness of f_ξ and g_η naturally implies the smoothness of F . Zhang and Xiao [2021, Lemma 4.2] show that Assumption I ensures F is: 1) C_F -Lipschitz continuous with $C_F = C_f C_g$; and 2) L_F -Lipschitz smooth with $L_F = L_g C_f + C_g^2 L_f$. We denote $\tilde{L}_F = \zeta_g C_f + \sigma_g C_g L_f$. Moreover, Assumption I also guarantees that f_ξ and g_η have bounded gradients. In addition, f_ξ and g_η are assumed to satisfy the following regularity condition in order to apply our extrapolation-based scheme from § 6.3.

Assumption J (Regularity). For all ξ in the support of $\mathbb{P}(\xi)$, ∇f_ξ is 4th-order differentiable with bounded derivatives (i.e., $a_l := \sup_{\mathbf{g} \in \mathbb{R}^p} \|\nabla^{(l)} f_\xi(\mathbf{g})\|_2 < \infty$ for $l = 1, 2, 3, 4, \forall \mathbf{x} \in \mathbb{R}^p$) and g_η has bounded moments upto 4th-order (i.e., $\sigma_k = \sup_{\mathbf{x} \in \mathbb{R}^d} \sup_{\xi} \mathbb{E}_{\eta|\xi} \left[\sum_{i=1}^p [g_\eta(\mathbf{x}) - \mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x})]]_i^k \right] < \infty, k = 1, 2, 3, 4$).

Remark 2. The core piece of Assumption J is the 4th order differentiability of ∇f_ξ as other parts can be easily satisfied through appropriate boundedness assumptions. This condition though is satisfied by common instantiations of CSO/FCCO. We discuss some examples including invariant logistic regression, instrumental variable regression, first-order MAML for sine-wave few-shot regression task, deep average precision maximization in § 2.7. Therefore, our improvements in sample complexity apply to all these problems.

Consider some time $t > 0$. Let G^{t+1} be a stochastic estimate of $\nabla F(\mathbf{x}^t)$ where \mathbf{x}^t is the current iterate. The next iterate $\mathbf{x}^{t+1} := \mathbf{x}^t - \gamma G^t$. Let $\mathbb{E}[\cdot]$ denote the conditional expectation, where we condition on all the randomness until time t . We consider the bias and variance terms coming from our gradient estimate. Formally, we define the following two quantities

$$\mathcal{E}_{\text{bias}}^{t+1} = \|\nabla F(\mathbf{x}^t) - \mathbb{E}[G^{t+1}]\|_2^2, \quad \mathcal{E}_{\text{var}}^{t+1} = \mathbb{E}[\|G^{t+1} - \mathbb{E}[G^{t+1}]\|_2^2].$$

Our idea of getting to an ϵ -stationary point (Definition 6.1) will be to ensure that $\mathcal{E}_{\text{bias}}^{t+1}$ and $\mathcal{E}_{\text{var}}^{t+1}$ are bounded. The main technical component of our analyses is in fact analyzing these bias and variance terms for the various gradient estimates considered. For this purpose, we first analyze the bias and variance terms for the (original) BSGD (Lemma E.5) and BSpiderBoost (Lemma E.7) algorithms, which are then used to get the corresponding bounds for our E-BSGD (Lemma E.6) and E-BSpiderBoost (Lemma E.8) algorithms. Through these bias and variance bounds, we establish the following main results of this section.

Theorem 6.2. *[E-BSGD Convergence] Consider the (CSO) problem. Suppose Assumptions G, H, I, J hold true and $L_F, C_F, \tilde{L}_F, C_g, F^*$ are constants and $C_e(f; g) := \frac{8a_3\sigma_3 + 18a_4\sigma_3^2 + 5a_4\sigma_4}{96}$ defined in § E.4.1 are associated with second order extrapolation in the CSO problem. Let step size $\gamma \leq 1/(2L_F)$. Then the output \mathbf{x}^s of E-BSGD (Algorithm 16) satisfies: $\mathbb{E}[\|\nabla F(\mathbf{x}^s)\|_2^2] \leq \epsilon^2$, for nonconvex F , if the inner batch size $m = \Omega(C_e C_g \epsilon^{-1/2})$, and the number of iterations*

$$T = \Omega(L_F(F(\mathbf{x}^0) - F^*)(\tilde{L}_F^2/m + C_F^2)\epsilon^{-4}).$$

The E-BSGD takes $\mathcal{O}(\epsilon^{-4})$ iterations to converge and compute $\mathcal{O}(\epsilon^{-0.5})$ gradients per iteration. Therefore, its resulting sample complexity is $\mathcal{O}(\epsilon^{-4.5})$ which is more efficient than $\mathcal{O}(\epsilon^{-6})$ of BSGD. Similar improvements can be observed for E-BSpiderBoost in Theorem 6.3.

Theorem 6.3. *[E-BSpiderBoost Convergence] Consider the (CSO) problem under the same assumptions as Theorem 6.2. Let step size $\gamma \leq 1/(13L_F)$. Then the output \mathbf{x}^s of E-BSpiderBoost (Algorithm 17) satisfies: $\mathbb{E}[\|\nabla F(\mathbf{x}^s)\|_2^2] \leq \epsilon^2$, for nonconvex F , if the inner batch size $m = \mathcal{O}(C_e C_g \epsilon^{-0.5})$, the hyperparameters of the outer loop of E-BSpiderBoost $B_1 = (\tilde{L}_F^2/m + C_F^2)\epsilon^{-2}$, $B_2 = \sqrt{B_1}$, $p_{out} = 1/B_2$, and the number of iterations*

$$T = \Omega(L_F(F(\mathbf{x}^0) - F^*)\epsilon^{-2}).$$

The resulting sample complexity of E-BSpiderBoost is $\mathcal{O}(\epsilon^{-3.5})$, which improves $\mathcal{O}(\epsilon^{-5})$ bound of BSpiderBoost [Hu et al., 2020b] and $\mathcal{O}(\epsilon^{-4})$ bound of V-MLMC/RT-MLMC [Hu et al., 2021].

6.5 Applying Stochastic Extrapolation in the FCCO Problem

In this section, we apply the extrapolation-based scheme from § 6.3 to the FCCO problem. We focus on case where $n = \mathcal{O}(\epsilon^{-2})$. For larger n , we can treat the FCCO problem as a CSO problem and get an $\mathcal{O}(\epsilon^{-3.5})$ bound from Theorem 6.3. All missing details are presented in Appendix E.5.

Now, a straightforward algorithm for FCCO is to use the finite-sum variant of SpiderBoost (or SPIDER) [Fang et al., 2018; Wang et al., 2019] in Algorithm 17. In this case, if we choose the outer batch sizes to be $B_1 = n$, $B_2 = \sqrt{n}$ and the inner batch size to be $m = \max\{\epsilon^{-2}/n, \epsilon^{-1/2}\}$. The resulting sample complexity of E-BSpiderBoost now becomes, $\mathcal{O}(\max\{\sqrt{n}/\epsilon^{2.5}, 1/\sqrt{n}\epsilon^4\})$, which recovers $\mathcal{O}(\epsilon^{-3.5})$ bound as in Theorem 6.3 for $n = \Theta(\epsilon^{-2})$. However, when n is small, such as $n = \mathcal{O}(1)$, the sample complexity degenerates to $\mathcal{O}(\epsilon^{-4})$ which is worse than the $\Omega(\epsilon^{-3})$ lower bound of stochastic optimization [Arjevani et al., 2022]. We leave the details to Theorem E.5. We still use Assumptions G, H, I, J for the analysis of FCCO problem, replacing the role of ξ with i .

Algorithm 7 E-NestedVR

```

1: Input:  $\mathbf{x}^0 \in \mathbb{R}^d$ , step-size  $\gamma$ , batch sizes  $S_1, S_2, B_1, B_2$ , Probability  $p_{\text{in}}, p_{\text{out}}$ 
2: for  $t = 0, 1, \dots, T - 1$  do
3:   if  $(t = 0)$  or (with prob.  $p_{\text{out}}$ ) then ▷ Large outer batch
4:     for  $i \in \mathcal{B}_1 \sim [n]$  with  $|\mathcal{B}_1| = B_1$  do
5:       draw  $\mathbf{y}_i^{t+1}$  from distribution  $\mathcal{D}_{\mathbf{y},i}^{t+1}$  defined in (6.10)
6:       compute  $\mathbf{z}_i^{t+1}$  using (6.11) and define  $\phi_i^t = \mathbf{x}^t$ 
7:        $G_{\text{E-NVR}}^{t+1} = \frac{1}{B_1} \sum_{i \in \mathcal{B}_1} (\mathbf{z}_i^{t+1})^\top \mathcal{L}_{\mathcal{D}_{\mathbf{y},i}^{t+1}}^{(2)} \nabla f_i(0)$ 
8:   else ▷ Small outer batch
9:     for  $i \in \mathcal{B}_2$  with  $|\mathcal{B}_2| = B_2$  do
10:      draw  $\mathbf{y}_i^{t+1}$  and  $\mathbf{y}_i^t$  from distribution  $\mathcal{D}_{\mathbf{y},i}^{t+1}$  and  $\mathcal{D}_{\mathbf{y},i}^t$  defined in (6.10)
11:      compute  $\mathbf{z}_i^{t+1}$  using (6.11) and define  $\phi_i^t = \mathbf{x}^t$ 
12:       $G_{\text{E-NVR}}^{t+1} = G_{\text{E-NVR}}^t + \frac{1}{B_2} \sum_{i \in \mathcal{B}_2} (\mathbf{z}_i^{t+1})^\top (\mathcal{L}_{\mathcal{D}_{\mathbf{y},i}^{t+1}}^{(2)} \nabla f_i(0) - \mathcal{L}_{\mathcal{D}_{\mathbf{y},i}^t}^{(2)} \nabla f_i(0))$ 
13:    $\mathbf{x}^{t+1} = \mathbf{x}^t - \gamma G_{\text{E-NVR}}^{t+1}$ 
14: Output:  $\mathbf{x}^s$  picked uniformly at random from  $\{\mathbf{x}^t\}_{t=0}^{T-1}$ 

```

Extrapolated NestedVR. We now introduce a nested variance reduction algorithm E-NestedVR which reaches low sample complexity for all choices of n . Missing proofs from this section are presented in § E.5. For the stochasticities in the FCCO problem, our idea is to use two nested SpiderBoost variance reduction components: one for the outer random variable i and the other for the inner random variable $\eta|i$. In each outer (resp. inner) SpiderBoost step, we choose large batch B_1 (resp. S_1) with probability p_{out} (resp. p_{in}); otherwise we choose small batch. Let H_i denote a set of m i.i.d. samples drawn from the conditional distribution $\mathbb{P}(\eta|i)$. Similarly, let \tilde{H}_i denote another set of m i.i.d. samples drawn from the same conditional distribution. For each given i , we approximate $\mathbb{E}_{\eta|i}[g_\eta(\mathbf{x}^t)]$ with \mathbf{y}_i^{t+1} from distribution $\mathcal{D}_{\mathbf{y},i}^{t+1}$ where,

$$\mathbf{y}_i^{t+1} = \begin{cases} \frac{1}{S_1} \sum_{\eta \in H_i} g_\eta(\mathbf{x}^t) & \text{with prob. } p_{\text{in}} \text{ or } t = 0 \\ \mathbf{y}_i^t + \frac{1}{S_2} \sum_{\eta \in H_i} (g_\eta(\mathbf{x}^t) - g_\eta(\phi_i^t)) & \text{with prob. } 1 - p_{\text{in}}. \end{cases} \quad (6.10)$$

Similarly, we approximate $\mathbb{E}_{\tilde{\eta}|i}[\nabla g_{\tilde{\eta}}(\mathbf{x}^t)]$ with \mathbf{z}_i^{t+1} defined as follows

$$\mathbf{z}_i^{t+1} = \begin{cases} \frac{1}{S_1} \sum_{\tilde{\eta} \in \tilde{H}_i} \nabla g_{\tilde{\eta}}(\mathbf{x}^t) & \text{with prob. } p_{\text{in}} \text{ or } t = 0 \\ \mathbf{z}_i^t + \frac{1}{S_2} \sum_{\tilde{\eta} \in \tilde{H}_i} (\nabla g_{\tilde{\eta}}(\mathbf{x}^t) - \nabla g_{\tilde{\eta}}(\phi_i^t)) & \text{with prob. } 1 - p_{\text{in}}, \end{cases} \quad (6.11)$$

where ϕ_i^t is the last time i is visited before time t . If i is not selected at time t , then $\mathbf{y}_i^{t+1} = \mathbf{y}_i^t$ and $\mathbf{z}_i^{t+1} = \mathbf{z}_i^t$. Note that we use independent samples for \mathbf{y}_i^{t+1} and \mathbf{z}_i^{t+1} .

Finally, we present E-NestedVR in Algorithm 7 where second-order extrapolation operator $\mathcal{L}^{(2)}$ is applied to each occurrence of ∇f_i . We now analyze its convergence guarantee. Our analysis works by first looking at the effect of multi-level variance reduction without the

extrapolation (that we refer to as NestedVR, Theorem E.6, Appendix E.5.2), and then showing how extrapolation could further help to drive down the sample complexity.

Theorem 6.4. *[E-NestedVR Convergence] Consider the (FCCO) problem. Under the same assumptions as Theorem 6.2.*

- If $n = \mathcal{O}(\epsilon^{-2/3})$, then we choose the hyperparameters of E-NestedVR (Algorithm 7) as $B_1 = B_2 = n, p_{out} = 1, S_1 = \tilde{L}_F^2 \epsilon^{-2}, S_2 = \tilde{L}_F \epsilon^{-1}, p_{in} = \tilde{L}_F^{-1} \epsilon, \gamma = \mathcal{O}(\frac{1}{\tilde{L}_F})$.

- If $n = \Omega(\epsilon^{-2/3})$, then we choose the hyperparameters of E-NestedVR as $B_1 = n, B_2 = \sqrt{n}, p_{out} = 1/\sqrt{n}, S_1 = S_2 = \max \left\{ C_e C_g \epsilon^{-1/2}, \tilde{L}_F^2 / (n \epsilon^2) \right\}, p_{in} = 1, \gamma = \mathcal{O}(\frac{1}{\tilde{L}_F})$.

Then the output \mathbf{x}^s of E-NestedVR satisfies: $\mathbb{E}[\|\nabla F(\mathbf{x}^s)\|_2^2] \leq \epsilon^2$, for nonconvex F with iterations

$$T = \Omega \left(L_F (F(\mathbf{x}^0) - F^*) \epsilon^{-2} \right).$$

From Theorem 6.4, E-NestedVR has a sample complexity of $\mathcal{O}(n \epsilon^{-3})$ in the small n regime ($n = \mathcal{O}(\epsilon^{-2/3})$) and $\mathcal{O}(\max\{\sqrt{n}/\epsilon^{2.5}, 1/\sqrt{n}\epsilon^4\})$ in the large n regime ($n = \Omega(\epsilon^{-2/3})$). Therefore, in the large n regime, this improves the $\mathcal{O}(n \epsilon^{-3})$ sample complexity of MSVR-V2 [Jiang et al., 2022].

6.6 Applications

In this section, we demonstrate the numerical performance of our proposed algorithms. We focus on the application of invariant logistic regression here. In Appendix E.6, we discuss performance of our proposed algorithms on other common CSO/FCCO applications, including instrumental variable regression and first-order model-agnostic meta-learning.

Invariant Risk Minimization. Invariant learning has wide applications in machine learning and related areas [Anselmi et al., 2016; Mroueh et al., 2015]. Invariant logistic regression [Hu et al., 2020b] is formulated as follows:

$$\min_{\mathbf{x}} \mathbb{E}_{\xi=(\mathbf{a},b)} [\log(1 + \exp(-b \mathbb{E}_{\eta|\xi}[\eta]^\top \mathbf{x}))],$$

where \mathbf{a} and b represent a sample and its corresponding label, and η is a noisy observation of the sample \mathbf{a} . This first part can be considered as a CSO objective, with $f_\xi(y) := \log(1 + \exp(-by))$ and $g_\eta(\mathbf{x}; \xi) := \eta^\top \mathbf{x}$. As the loss $f_\xi \in \mathcal{C}^\infty$ is smooth, our results from Sections 6.4 and 6.5 are applicable.

An ℓ_2 -regularizer is added to ensure the existence of a unique minimizer. Since the gradient of the penalization term is unbiased, we only have to consider the biasness of the data-dependent term. We generate a synthetic dataset with $d = 10$ dimensions. The minimizer is drawn from Gaussian distribution $\mathbf{x}^* \sim \mathcal{N}(0, 1) \in \mathbb{R}^d$. We draw invariant samples $\{(\mathbf{a}_i, b_i)\}_i$ where $\mathbf{a}_i \sim \mathcal{N}(0, 1) \in \mathbb{R}^d$ and compute $b_i = \text{sgn}(\mathbf{a}_i^\top \mathbf{x}^*)$. Given each $\xi = (\mathbf{a}_i, b_i)$, we draw perturbed observations $\eta \sim \mathcal{N}(\mathbf{a}_i, 100) \in \mathbb{R}^d$.

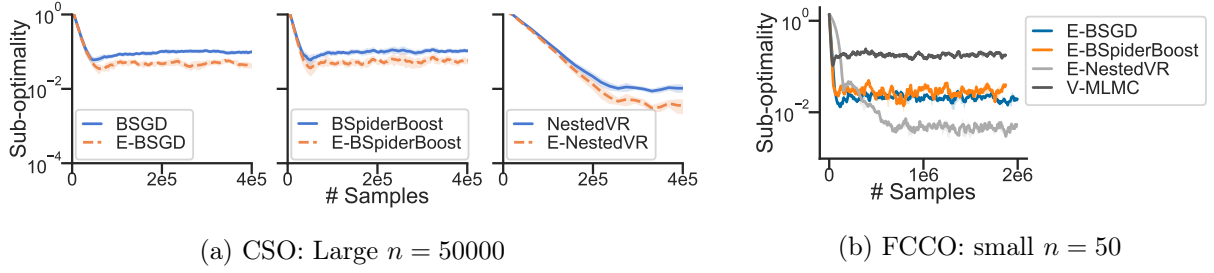


Fig. 6.2 Performances of algorithms and their extrapolated versions on the invariant logistic regression task. Algorithms in each subplot use the same amount of inner batch size in each iteration. The shaded region represents the 95%-confidence interval computed over 10 runs.

We consider drawing ξ from a large set ($n = 50000$) and a small set ($n = 50$) as CSO and FCCO problems respectively. As baselines, we implemented the BSGD and BSpiderBoost methods from [Hu et al., 2020b], V-MLMC approach from [Hu et al., 2021], and NestedVR approach from Appendix E.5.2 which achieves the same complexity as MSVR-V2 [Jiang et al., 2022] for the FCCO problem. The results are shown in Fig. 6.2. In the CSO setting, we compare biased gradient methods with their extrapolated variants (BSGD vs. E-BSGD, BSpiderBoost vs. E-BSpiderBoost, and NestedVR vs. E-NestedVR). The extrapolated versions of BSGD, BSpiderBoost, and NestedVR consistently reach lower error than their non-extrapolated counterparts, as is evident in Figure 6.2a. In this case, the performance of BSpiderBoost is similar to BSGD as also noted by the authors of these techniques [Hu et al., 2020b], and a drawback of BSpiderBoost seems to be that it is much harder to tune in practice. However, it is clear that E-BSGD outperforms BSGD, and E-BSpiderBoost outperforms BSpiderBoost, respectively. In the FCCO setting, we compare extrapolation based methods and MLMC based methods. Figure 6.2a, shows that E-NestedVR outperforms all other extrapolated algorithms, including the V-MLMC approach of [Hu et al., 2021], matching our theoretical findings.

6.7 Concluding Remarks

In this paper, we consider the conditional stochastic optimization CSO problem and its finite-sum variant FCCO. Due to the interplay between nested structure and stochasticity, most of the existing gradient estimates suffer from large biases and have large sample complexity of $\mathcal{O}(\epsilon^{-5})$. We propose stochastic extrapolation-based algorithms that tackle this bias problem and improve the sample complexities for both these problems. While we focus on nonconvex objectives, our proposed algorithms can also be beneficial when used with strongly convex, convex objectives. We also believe that similar ideas could also prove helpful for multi-level stochastic optimization problems [Zhang and Xiao, 2021] with nested dependency.

Acknowledgements

We would like to thank Caner Turkmen, Sai Praneeth Karimireddy, and Martin Jaggi for helpful initial discussions surrounding this project.

Chapter 7

Conclusion and Future Work

Summary of Contributions

Machine learning, particularly deep learning, has become an indispensable tool for addressing a broad spectrum of challenges. The growing need for distributed training allows models to leverage collaborative data and computational resources, yielding better outcomes compared to isolated training. However, the distributed paradigm introduces unique hurdles, primarily concerning participant honesty and protocol compliance. Without adequate safeguards, Byzantine adversaries can degrade model quality, while privacy adversaries might infer sensitive data from inter-participant message exchanges. Such actors severely compromise the utility of collaborative learning. In addition to utility, the expanding sizes of machine learning models and datasets place a substantial burden on computational resources, making optimization a highly debated subject. This thesis aims to enhance both the *utility* and *efficiency* of distributed training.

For utility, we develop Byzantine-robust optimizers and extend them to be compatible with secure multiparty computation (MPC) protocols. Our dual strategy for Byzantine robustness involves clipping-based aggregation at the receiver’s end and variance reduction at the sender’s end. Employing these techniques, we achieve Byzantine tolerance while preserving scalability. Further, we amalgamate Byzantine robustness and input privacy by using secure MPC protocols on multiple non-colluding servers.

On the efficiency front, we introduce a relay mechanism to decentralize communication, mitigating slowdowns caused by data heterogeneity. We also address bias in conditional stochastic optimization problems by applying extrapolation and variance reduction techniques, thereby reducing sample complexity.

Despite our contributions, significant work remains. Certain limitations and assumptions warrant further investigation:

- Improved Privacy: Our model assumes non-colluding servers to combine input privacy and robustness but this assumption is not satisfied in typical federated learning setups.

Other single-server solutions are computationally intensive [Burkhalter et al., 2021]. An efficient single-server solution remains a challenge.

- **Output Privacy:** We focus on input privacy through secure MPC protocols; however, output privacy, as a separate concern, has not been considered. Future work could incorporate differential privacy primitives to preserve output privacy.
- **Efficiency:** Our current solutions for conditional stochastic optimization rely on higher-order regularity conditions for the objective function. Although these assumptions serve to derive our extrapolation scheme, they are not explicitly required by the algorithm. We aim to relax these assumptions in future work.

In summary, this thesis contributes to the advancement of the utility and efficiency of distributed machine learning. Nonetheless, it uncovers myriad avenues for future research, inviting further study to fully harness the potential of this emerging field.

Appendix A

Byzantine-robust Learning on Heterogeneous Dataset via Bucketing

A.1 Experiment setup and additional experiments

A.1.1 Experiment setup

General setup

The default experiment setup is listed in Table A.1. We use number of iterations $T = 8$ for

Table A.1 Default experimental settings for MNIST

Dataset	MNIST
Architecture	CONV-CONV-DROPOUT-FC-DROPOUT-FC
Training objective	Negative log likelihood loss
Evaluation objective	Top-1 accuracy
Batch size	$32 \times$ number of workers
Momentum	0 or 0.9
Learning rate	0.01
LR decay	No
LR warmup	No
# Iterations	600 or 4500
Weight decay	No
Repetitions	3, with varying seeds
Reported metric	Mean test accuracy over the last 150 iterations

RFA, $b = q$ for TM, and $\tau = \frac{10}{1-\beta}$ for CCLIP.

Constructing datasets

The MNIST dataset has 10 classes each with similar amount of samples. In this part, we discuss how to process and distribute MNIST to each workers in order to achieve long-tailness and heterogeneity.

Long-tailness. The long-tailness (*-LT) is achieved by sampling class with exponentially decreasing portions $\gamma \in (0, 1]$. That is, for class $i \in [10]$, we only randomly sample γ^i portion of all samples in class i . We define α as the ratio of the largest class over the smallest class, which can be written as $\alpha = \frac{1}{\gamma^9}$. For example, if $\gamma = 1$, then all classes have same amount of samples and thus $\alpha = 1$; if $\gamma = 0.5$ then $\alpha = 2^9 = 512$. Note that the same procedure has to be applied to the test dataset.

Heterogeneity. Steps to construct IID/non-iid dataset from MNIST dataset

1. Sort the training dataset by its labels.
2. Evenly divide the sorted training dataset into chunks of same size. The number of chunks equals the number of good workers. If the last chunk has fewer samples, we augment it with samples from itself.
3. Shuffle the samples within the same worker.

Heterogeneity + Long-tailness. First transform the training dataset into long-tail dataset, then feed it to the previous procedure to introduce heterogeneity.

About dataset on Byzantine workers. The training set is divided by the number of good workers. So the good workers has to full information of training dataset. The Byzantine worker has access to the whole training dataset.

Setup for each experiment

In Table A.2, we list the hyperparameters for the experiments. In Figure 2.1 and Figure 2.2, we use IPM Attack with $\epsilon = 0.1$. In Figure 2.1, we use ALIE attack with hyperparameter z computed according to [Baruch et al., 2019]

$$z = \max_z \left(\phi(z) < \frac{n - q - s}{n - q} \right)$$

where $s = \lfloor \frac{n}{2} + 1 \rfloor - q$ and ϕ is the cumulative standard normal function. In our setup, the $z \approx 0.25$.

Running environment

We summarize the running environment of this paper as in Table A.3.

Table A.2 Setups for each experiment.

	n	q	momentum	Iters	LT	NonIID
Table 2.1	24	0	0	4500	$\alpha = 1, \alpha = 500$	iid/ non-iid
Table 2.2	25	5	0	600	$\alpha = 1$ (balanced)	iid/ non-iid
Table 2.3	24	0	0	4500	$\alpha = 1, \alpha = 500$	iid/ non-iid
Table 2.4	25	5	0	600	$\alpha = 1$ (balanced)	iid/ non-iid
Figure 2.1	25	5	0 / 0.9	600	$\alpha = 1$ (balanced)	non-iid
Figure 2.2	53	5	0 / 0.9	600	$\alpha = 1$ (balanced)	non-iid
Figure A.1	25	5	0 / 0.5 / 0.9 / 0.99	600	$\alpha = 1$ (balanced)	non-iid
Figure A.2	25	5	0 / 0.5 / 0.9 / 0.99	1200	$\alpha = 1$ (balanced)	non-iid
Figure A.3	20	3	0	1200	$\alpha = 1$ (balanced)	non-iid
Figure A.4	20	3	0	3000	$\alpha = 1$ (balanced)	non-iid
Figure A.6	24	3	0	1200	$\alpha = 1$ (balanced)	non-iid

Table A.3 Runtime hardwares and softwares.

CPU	
Model name	Intel (R) Xeon (R) Gold 6132 CPU @ 2.60 GHz
# CPU(s)	56
NUMA node(s)	2
GPU	
Product Name	Tesla V100-SXM2-32GB
CUDA Version	11.0
PyTorch	
Version	1.7.1

A.1.2 Additional experiments

Clipping radius scaling

The radius τ of CCLIP depends on the norm of good gradients. However, PyTorch implements SGD with momentum using the following formula

$$\mathbf{m}_i^t = \beta \mathbf{m}_i^{t-1} + \mathbf{g}_i(\mathbf{x}^{t-1}) \quad \text{for every } i \in \mathcal{V}_R$$

which may leads to the increase in the gradient norm.

Gradient norms. In Figure A.1 we present the averaged gradient norm from all good workers. Here we use CCLIP as the aggregator and $\tau = \frac{10}{1-\beta}$. The norm of gradients are computed before aggregation. Even though the dataset on workers are non-iid, the gradient norms are roughly of same order. The gradient dissimilarity ζ^2 also increases accordingly.

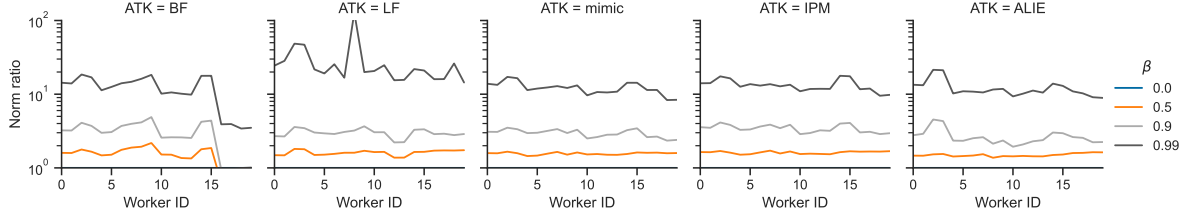


Fig. A.1 The ratio of norm of good gradients with momentum β over no momentum under different attacks.

Scaled clipping radius. As the gradient norm increases with momentum β , the clipping radius should increase accordingly. In Figure A.2 we compare 3 schemes: 1) no scaling ($\tau = 10$, $\beta = 0$); 2) *linear* scaling $\frac{10}{1-\beta}$; 3) *sqrt* scaling $\frac{10}{\sqrt{1-\beta}}$. The no scaling scheme converges but slower while with momentum. The linear scaling is usually better than *sqrt* scaling and with bucketing it becomes more stable. However, The scaled clipping radius fails for $\beta = 0.99$ under label flipping attack. This is because the gradient can be very large and ζ^2 dominates. So in general, a linear scaling of clipping radius with momentum $\beta = 0.9$ would be a good choice.

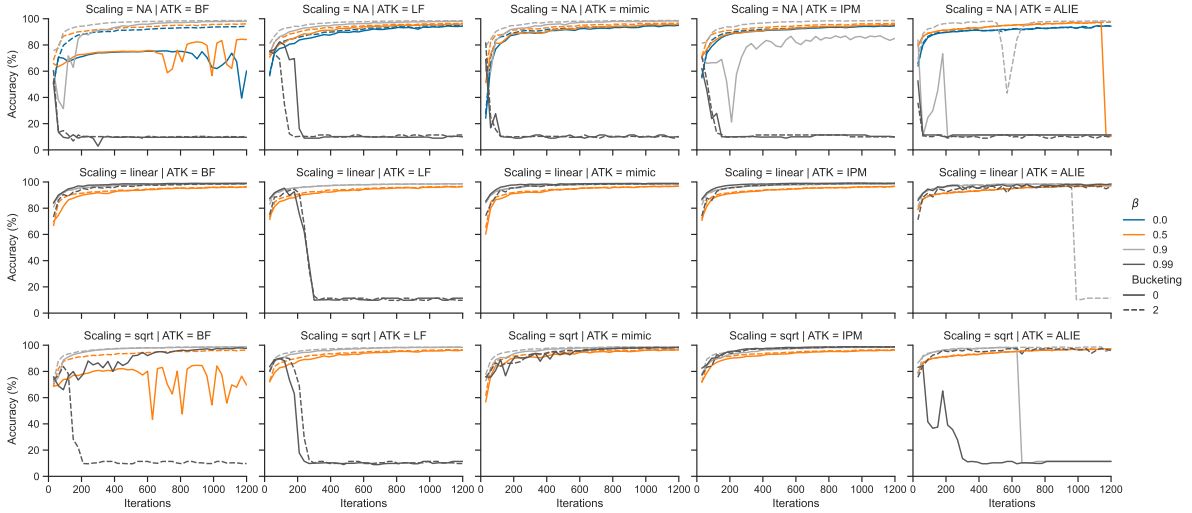


Fig. A.2 Convergence of CCLIP with $\tau = 10$, $\frac{10}{1-\beta}$, $\frac{10}{\sqrt{1-\beta}}$ for $\beta = 0, 0.5, 0.9, 0.99$. The s is the bucketing hyperparameter.

Demonstration of effects of bucketing through the selections of KRUM

In the main text we have theoretically show that bucketing helps aggregators alleviate the impact of non-iid. In this section we empirically show that after bucketing aggregators can incorporate updates more evenly from good workers and therefore the problem of non-iid among good workers is less significant. Since KRUM outputs the id of the selected device, it is very convenient to record the frequency of each worker being selected. Since bucketing replicates each worker for s times, we divide their frequencies by s for normalization. From Figure A.3, we

can see that without bucketing KRUM basically almost always selects updates from Byzantine workers while with larger s , the selection becomes more evenly distributed.

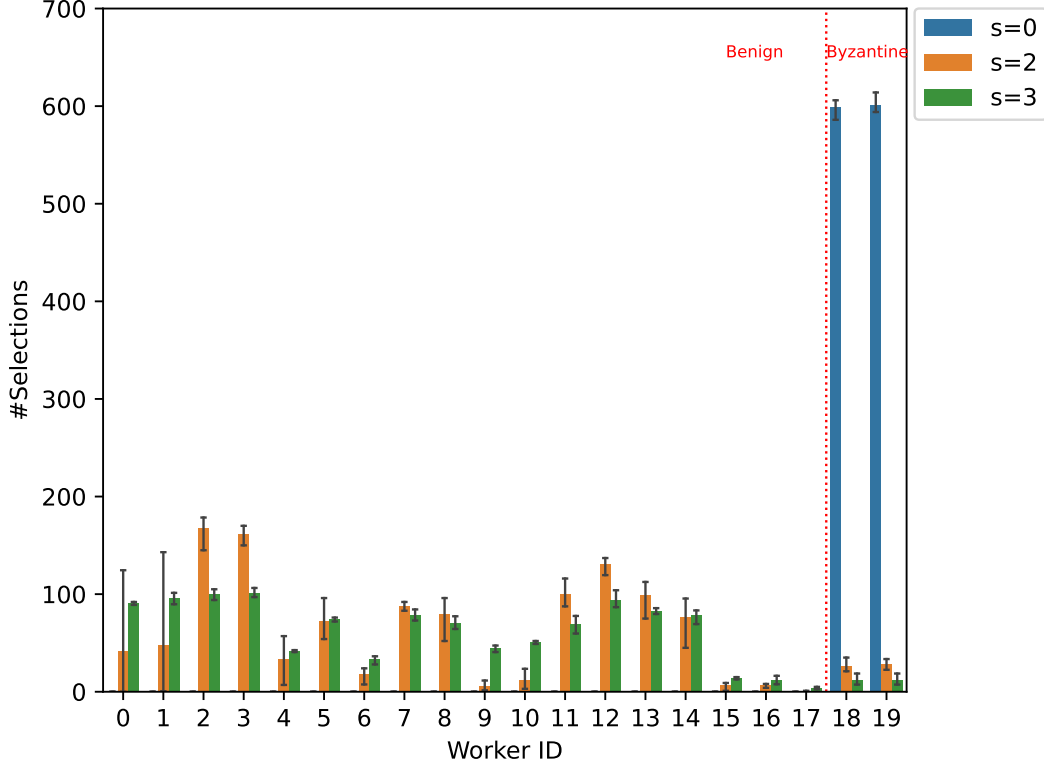


Fig. A.3 The selected workers of KRUM for bucketing coefficient $s = 0, 2, 3$. There are 20 workers and the last 2 workers (worker id=18,19) are Byzantine with label-flipping attack.

Overparameterization

The architecture of the neural net used in the experiments can be scaled to make it overparameterized. We add more parameters to the model by multiplying the channels of 2D Conv layer and fully connected layer by a factor of ‘scale’. So the original model has a scale of 1. We show the training losses decrease faster for overparameterized models in Figure A.4. As we can see, the convergence behaviors are similar for different model scales with overparameterized models having smaller training loss despite the existence of Byzantine workers.

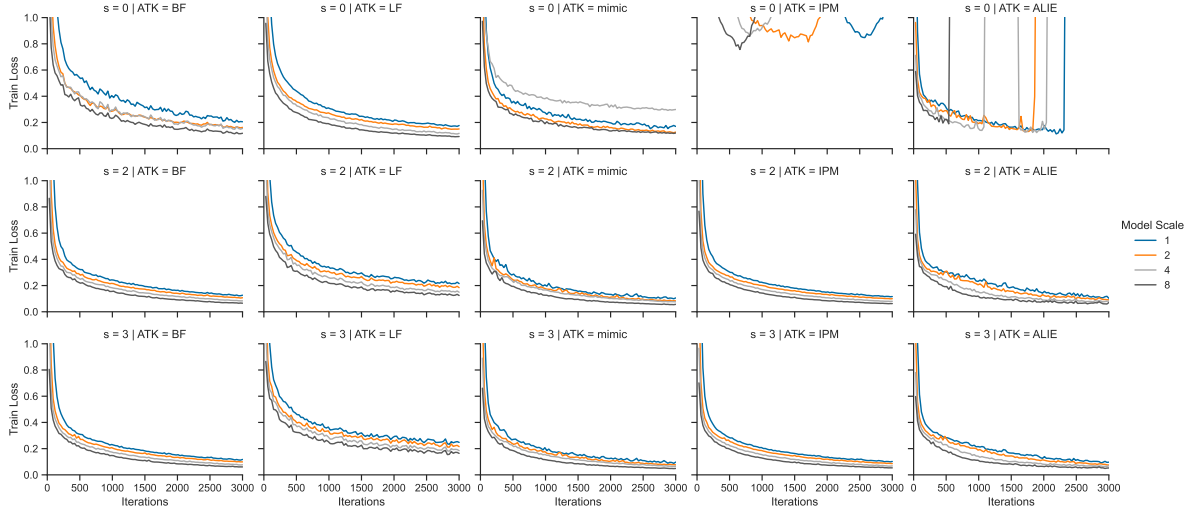


Fig. A.4 The training loss of models of different levels of overparameterization.

In Figure A.5, we explicitly investigate the influence of overparameterization on B^2 defined in (2.3). As we can see, heterogeneity bound B^2 decreases with increasing level of overparameterization, showcasing how overparameterization minimizes the local objectives in the presence of Byzantine workers. It supports our theory in § 2.6.4 that overparameterization can fix the convergence, making it possible to achieve practical Byzantine-robust learning. The underlying base aggregator is RFA.

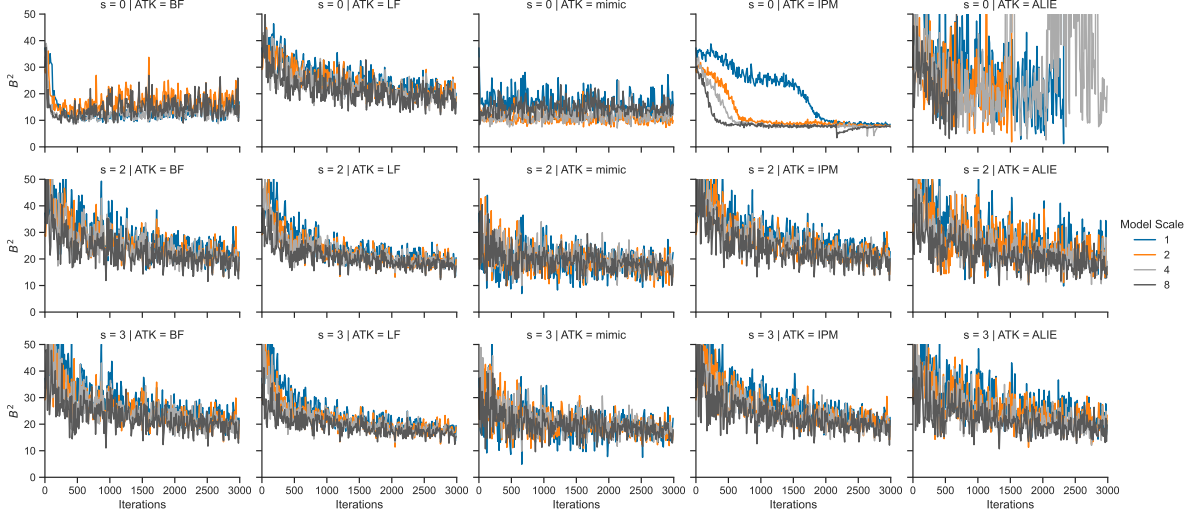


Fig. A.5 The B^2 in (2.3) for different levels of overparameterization.

Resampling - variant of bucketing

In the previous version of this work we repeat the gradients for s times and then put sn gradients into n buckets. The results in Figure A.6 suggest that the convergence rate of bucketing and

resampling is almost the same. So aggregators can benefit more from bucketing as it reduces the number of input gradients and therefore reduce the complexity.

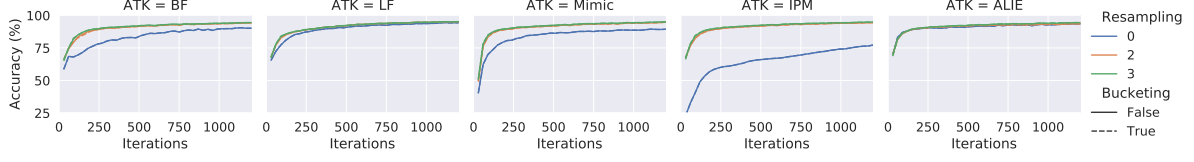


Fig. A.6 The convergence SGD with bucketing and resampling under different attacks. The underlying aggregator is RFA.

A.2 Implementing the mimic attack

The § 2.4.2 describes the idea and formulation of the mimic attack. In this section, we discuss how to pick i_* and implement the mimic attack efficiently. To pick i_* , we use an initial phase ($\mathcal{I}^0 \approx 1$ epoch) to compute a direction \mathbf{z} of maximum variance of the outputs of the good workers:

$$\mathbf{z} = \arg \max_{\|\mathbf{z}\|=1} \mathbf{z}^\top \left(\sum_{t \in \mathcal{I}_0} \sum_{i \in \mathcal{V}_R} (\mathbf{x}_i^t - \boldsymbol{\mu})(\mathbf{x}_i^t - \boldsymbol{\mu})^\top \right) \mathbf{z} \quad \text{where} \quad \boldsymbol{\mu} = \frac{1}{|\mathcal{V}_R| |\mathcal{I}_0|} \sum_{i \in \mathcal{V}_R, t \in \mathcal{I}_0} \mathbf{x}_i^t.$$

Then we pick a worker i^* to mimic by computing

$$i_* = \arg \max_{i \in \mathcal{V}_R} \left| \sum_{t \in \mathcal{I}_0} \mathbf{z}^\top \mathbf{x}_i^t \right|.$$

In the following steps, we show how to solve the optimization problem.

First, rewrite the mimic attack in its online version at time $t \in \mathcal{I}_0$

$$\mathbf{z}^t = \arg \max_{\|\mathbf{z}\|=1} h^t(\mathbf{z})$$

where $\boldsymbol{\mu}^t = \frac{1}{|\mathcal{V}_R|t} \sum_{\tau \leq t} \sum_{i \in \mathcal{V}_R} \mathbf{x}_i^\tau$ and

$$h^t(\mathbf{z}) = \mathbf{z}^\top \left(\sum_{\tau \leq t} \sum_{i \in \mathcal{V}_R} (\mathbf{x}_i^\tau - \boldsymbol{\mu}^t)(\mathbf{x}_i^\tau - \boldsymbol{\mu}^t)^\top \right) \mathbf{z}.$$

Thus we can iteratively update $\boldsymbol{\mu}^t$ by

$$\boldsymbol{\mu}^{t+1} = \frac{t}{1+t} \boldsymbol{\mu}^t + \frac{1}{1+t} \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbf{x}_i^{t+1},$$

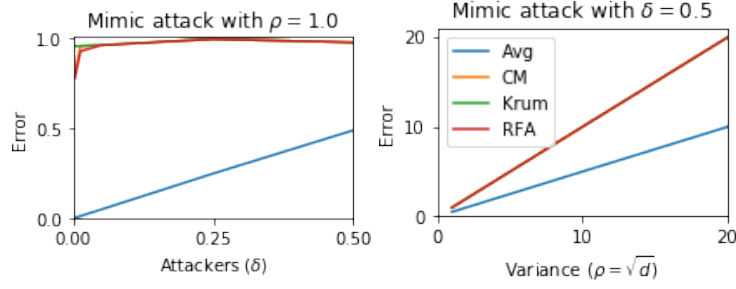


Fig. A.7 Error with random vectors with variance $\rho^2 = d$ and δ fraction of Byzantine workers imitating a fixed good worker (say worker $1 \in \mathcal{V}_R$). RFA performs slightly better than CM and Krum, but all have *higher* error than simply averaging across various settings of δ and ρ .

and then

$$\begin{aligned} \arg \max_{\|z\|=1} h^{t+1}(z) &\approx \frac{t}{1+t} z^t + \frac{1}{1+t} \arg \max_{\|z\|=1} z^\top \left(\sum_{i \in \mathcal{V}_R} (x_i^{t+1} - \mu^{t+1})(x_i^{t+1} - \mu^{t+1})^\top \right) z \\ &\approx \frac{t}{1+t} z^t + \frac{1}{1+t} \left(\sum_{i \in \mathcal{V}_R} (x_i^{t+1} - \mu^{t+1})(x_i^{t+1} - \mu^{t+1})^\top \right) z^t. \end{aligned}$$

The above algorithm corresponds to Oja's method for computing the top eigenvector in a streaming fashion [Oja, 1982]. Then, in each subsequent iteration t , we pick

$$i_\star^t = \arg \max_{i \in \mathcal{V}_R} z^\top x_i^t.$$

Example. Each of the good workers $i \in \mathcal{V}_R \subseteq [n]$ has an input a $x_i \in \{\pm 1\}^d$ where each coordinate is an independent Rademacher random variable. The inputs then have mean $\mathbf{0}$ and variance $\mathbb{E}\|x_i\|^2 = \rho^2 = d$. Now, the Byzantine attackers $j \in \mathcal{V}_B$ have dual goals: i) escape detection, and ii) increase data imbalance. For this, we propose the following simple passive attack: pick some fixed worker $i_\star \in \mathcal{V}_R$ (say 1) and every Byzantine worker $j \in \mathcal{V}_B$ outputs $x_j = x_1$. The attackers cannot be filtered as they imitate an existing good worker, but still can cause imbalance in the data distribution. This serves as the intuition for our attack.

A.3 Constructing a robust aggregator using bucketing

A.3.1 Supporting lemmas

We first start with proving the main bucketing Lemma 2.2 restated below.

Lemma' 2.2. *Suppose we are given n independent (but not identical) random vectors $\{x_1, \dots, x_n\}$ such that a good subset $\mathcal{V}_R \subseteq [n]$ of size at least $|\mathcal{V}_R| \geq n(1 - \delta)$ satisfies:*

$$\mathbb{E}\|x_i - x_j\|^2 \leq \rho^2, \quad \text{for any fixed } i, j \in \mathcal{V}_R.$$

Define $\bar{\mathbf{x}} := \frac{1}{|\mathcal{V}_R|} \sum_{j \in \mathcal{V}_R} \mathbf{x}_j$ and $m = \lceil n/s \rceil$. Let the outputs after s -bucketing be $\{\mathbf{y}_1, \dots, \mathbf{y}_m\}$. Then, a subset of the outputs $\tilde{\mathcal{V}}_R \subseteq \{1, \dots, m\}$ of size at least $|\tilde{\mathcal{V}}_R| \geq m(1 - \delta s)$ satisfies

$$\mathbb{E}[\mathbf{y}_i] = \mathbb{E}[\bar{\mathbf{x}}] \quad \text{and} \quad \mathbb{E}\|\mathbf{y}_i - \mathbf{y}_j\| \leq \rho^2/s \quad \text{for any fixed } i, j \in \tilde{\mathcal{V}}_R.$$

Proof. Let us define the buckets used to compute \mathbf{y}_i as

$$B_i := \{\pi(s(i-1) + 1), \dots, \pi(\min\{s \cdot i, n\})\}.$$

Recall that for some permutation π over $[n]$ and for every $i = \{1, \dots, m\}$, we defined $m = \lceil n/s \rceil$ and

$$\mathbf{y}_i \leftarrow \frac{1}{|B_i|} \sum_{k=(i-1) \cdot s + 1}^{\min(n, i \cdot s)} \mathbf{x}_{\pi(k)}.$$

Then, define the *new* good set

$$\tilde{\mathcal{V}}_R = \{i \in [m] \mid B_i \subseteq \mathcal{V}_R\}$$

$\tilde{\mathcal{V}}_R$ contains the set of all the resampled vectors which are made up of only good vectors i.e. are uninfluenced by any Byzantine vector. Since $|\mathcal{V}_B| \leq \delta n$ and each can belong to only 1 bucket, we have that $|\tilde{\mathcal{V}}_R| \geq (1 - \delta s)m$. Now, for any fixed $i \in \tilde{\mathcal{V}}_R$, let us look at the conditional expectation over the random permutation π we have

$$\mathbb{E}_\pi[\mathbf{y}_i \mid i \in \tilde{\mathcal{V}}_R] = \frac{1}{|B_i|} \sum_{k=(i-1) \cdot s + 1}^{\min(n, i \cdot s)} \mathbb{E}_\pi[\mathbf{x}_{\pi(k)} \mid \pi(k) \in \mathcal{V}_R] = \frac{1}{|\mathcal{V}_R|} \sum_{j \in \mathcal{V}_R} \mathbf{x}_j = \bar{\mathbf{x}}.$$

This yields the first part of the lemma. Now we analyze the variance. Thus, we can write $\mathbf{y}_i = \frac{1}{s} \sum_{k \in B_i} \mathbf{x}_k$. Further, $|B_i| = s$ for any i , and $B_i \subseteq \mathcal{V}_R$ if $i \in \tilde{\mathcal{V}}_R$. With this, for any fixed $i, j \in \tilde{\mathcal{V}}_R$ the variance can be written as

$$\begin{aligned} \mathbb{E}\|\mathbf{y}_i - \mathbf{y}_j\|^2 &= \mathbb{E}\left\| \frac{1}{s} \sum_{k \in B_i} \mathbf{x}_k - \frac{1}{s} \sum_{l \in B_j} \mathbf{x}_l \right\|^2 \\ &= \frac{\rho^2}{s}. \end{aligned}$$

□

This additional lemma about the maximum expected distance between good workers will also be useful later.

Lemma A.1 (maximum good distance). *Suppose we are given the output of bucketing $\mathbf{y}_1, \dots, \mathbf{y}_m$ which for $m = \lceil n/s \rceil$ satisfy for any fixed $i \in \tilde{\mathcal{V}}_R$, $\mathbb{E}[\mathbf{y}_i] = \boldsymbol{\mu}$ and $\mathbb{E}\|\mathbf{y}_i - \boldsymbol{\mu}\|^2 \leq \rho^2/s$. Then, we*

have

$$\mathbb{E} \left[\max_{i \in \tilde{\mathcal{V}}_R} \|\mathbf{y}_i - \boldsymbol{\mu}\|^2 \right] \leq n\rho^2/s^2.$$

Further, there exist instances where

$$\mathbb{E} \left[\max_{i \in \tilde{\mathcal{V}}_R} \|\mathbf{y}_i - \boldsymbol{\mu}\|^2 \right] \geq \Omega(n\rho^2/s^2).$$

Proof. For the upper bound, we simply use

$$\mathbb{E} \left[\max_{i \in \tilde{\mathcal{V}}_R} \|\mathbf{y}_i - \boldsymbol{\mu}\|^2 \right] \leq \sum_{i \in \tilde{\mathcal{V}}_R} \mathbb{E} \|\mathbf{y}_i - \boldsymbol{\mu}\|^2 \leq m\rho^2/s.$$

For the lower bound, let $\tilde{\mathcal{V}}_R = [m]$ and consider $\mathbf{y}_i \sim \tilde{\rho}\sqrt{m}\text{Bern}(p = \frac{1}{m})$. This means \mathbf{y}_i is either 0 or $\tilde{\rho}\sqrt{m}$. Further, its variance is clearly bounded by $\tilde{\rho}^2$. Upon drawing m samples, the probability of seeing at least 1 $\mathbf{y}_j = \tilde{\rho}\sqrt{m}$ is

$$1 - \Pr(\mathbf{y}_i = 0 \ \forall i \in [m]) = 1 - (1 - \frac{1}{m})^m \geq 1 - 1/e \geq 0.5.$$

Thus, with probability at least 0.5 we have

$$\max_{i \in [n]} \|\mathbf{y}_i - \boldsymbol{\mu}\|^2 \geq m\tilde{\rho}^2/2.$$

This directly proves our lower bound by defining $\tilde{\rho}^2 := \rho^2/s$ and recalling that $m = \lceil n/s \rceil$. Note that this lemma can be tightened if we make stronger assumptions on the noise such as $\mathbb{E} \|\mathbf{y}_i - \boldsymbol{\mu}\|^r \leq (\rho/\sqrt{s})^r$ for some large $r \geq 2$. However, we focus on using standard stochastic assumptions ($r = 2$) in this work. \square

A.3.2 Proofs of robustness

Let $\{\mathbf{y}_1, \dots, \mathbf{y}_m\}$ be the resampled vectors with bucketing using $s = \frac{\delta_{\max}}{\delta}$. By Lemma 2.2, we have that there is a $\tilde{\mathcal{V}}_R \subseteq [m]$ of size $|\tilde{\mathcal{V}}_R| > m(1 - \delta_{\max})$ which satisfies for any fixed $i, j \in \tilde{\mathcal{V}}_R$

$$\mathbb{E} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \leq \frac{\delta\rho^2}{\delta_{\max}} =: \tilde{\rho}^2.$$

This observation will be combined with each of the algorithms to obtain robustness guarantees.

Robustness of Krum. We now prove that Krum when combined with bucketing is a robust aggregator. We can rewrite the output of Krum as the following for $\delta_{\max} = 1/4 - \nu$ for some

arbitrarily small positive number $\nu \in (0, 1/4)$:

$$\text{KRUM}(\mathbf{y}_1, \dots, \mathbf{y}_m) = \arg \min_{\mathbf{y}_i} \min_{|S|=3m/4} \sum_{j \in S} \|\mathbf{y}_i - \mathbf{y}_j\|^2.$$

Let \mathcal{S}^* and k^* be the quantities which minimize the optimization problem solved by KRUM.

The main difficulty of analyzing KRUM is that even if we succeed in selecting a $k^* \in \tilde{\mathcal{V}}_R$, k^* depends on the sampling. Hence, we **cannot** claim that the error is bounded by $\tilde{\rho}^2$ i.e.¹

$$\mathbb{E} \|\mathbf{y}_{k^*} - \mathbf{y}_j\|^2 \not\leq \tilde{\rho}^2 \text{ for some fixed } j \in \tilde{\mathcal{V}}_R.$$

This is because the variance is bounded by $\tilde{\rho}^2$ only for a *fixed* i , and not a data dependent k^* . Instead, we will have to rely on Lemma A.1 that

$$\mathbb{E} \|\mathbf{y}_{k^*} - \mathbf{y}_j\|^2 \leq \mathbb{E} \max_{i \in \tilde{\mathcal{V}}_R} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \leq m \tilde{\rho}^2.$$

Lemma A.1 shows that this inequality is essentially tight and hence relying on it necessarily incurs an extra factor of m which can be very large. Instead, we show an alternate analysis which works for a smaller breakdown point of $\delta_{\max} = 1/4$, but *does not* incur the extra m factor.

For any good input $i \in \tilde{\mathcal{V}}_R$, we have

$$\begin{aligned} \|\mathbf{y}_{k^*} - \bar{\mathbf{x}}\|^2 &\leq 2\|\mathbf{y}_{k^*} - \mathbf{y}_i\|^2 + 2\|\mathbf{y}_i - \bar{\mathbf{x}}\|^2 \\ \Rightarrow 2\|\mathbf{y}_{k^*} - \mathbf{y}_i\|^2 &\geq \|\mathbf{y}_{k^*} - \bar{\mathbf{x}}\|^2 - 2\|\mathbf{y}_i - \bar{\mathbf{x}}\|^2. \end{aligned}$$

Further, for a bad worker $j \in \tilde{\mathcal{V}}_B$ we can write

$$2\|\mathbf{y}_{k^*} - \mathbf{y}_j\|^2 \geq \|\mathbf{y}_j - \bar{\mathbf{x}}\|^2 - 2\|\mathbf{y}_{k^*} - \bar{\mathbf{x}}\|^2.$$

Combining both and summing over \mathcal{S}^* ,

$$\begin{aligned} \sum_{i \in \mathcal{S}^*} 2\|\mathbf{y}_{k^*} - \mathbf{y}_i\|^2 &= \sum_{i \in \tilde{\mathcal{V}}_R \cap \mathcal{S}^*} 2\|\mathbf{y}_{k^*} - \mathbf{y}_i\|^2 + \sum_{j \in \tilde{\mathcal{V}}_B \cap \mathcal{S}^*} 2\|\mathbf{y}_{k^*} - \mathbf{y}_j\|^2 \\ &\geq \sum_{j \in \tilde{\mathcal{V}}_B \cap \mathcal{S}^*} \|\mathbf{y}_j - \bar{\mathbf{x}}\|^2 - 2 \sum_{i \in \tilde{\mathcal{V}}_R \cap \mathcal{S}^*} \|\mathbf{y}_i - \bar{\mathbf{x}}\|^2 \\ &\quad + (|\tilde{\mathcal{V}}_R \cap \mathcal{S}^*| - 2|\tilde{\mathcal{V}}_B \cap \mathcal{S}^*|) \|\mathbf{y}_{k^*} - \bar{\mathbf{x}}\|^2. \end{aligned}$$

¹This issue was incorrectly overlooked in the original analysis of KRUM [Blanchard et al., 2017]

We can rearrange the above equation as

$$\begin{aligned}
\|\mathbf{y}_{k^*} - \bar{\mathbf{x}}\|^2 &\leq \frac{1}{(|\tilde{\mathcal{V}}_R \cap \mathcal{S}^*| - 2|\tilde{\mathcal{V}}_B \cap \mathcal{S}^*|)} \left(\sum_{i \in \mathcal{S}^*} 2\|\mathbf{y}_{k^*} - \mathbf{y}_i\|^2 + \sum_{i \in \tilde{\mathcal{V}}_R \cap \mathcal{S}^*} 2\|\mathbf{y}_i - \bar{\mathbf{x}}\|^2 \right) \\
&\leq \frac{1}{(|\mathcal{S}^*| - 3|\tilde{\mathcal{V}}_B|)} \left(\sum_{i \in \mathcal{S}^*} 2\|\mathbf{y}_{k^*} - \mathbf{y}_i\|^2 + \sum_{i \in \tilde{\mathcal{V}}_R \cap \mathcal{S}^*} 2\|\mathbf{y}_i - \bar{\mathbf{x}}\|^2 \right) \\
&\leq \frac{1}{(|\mathcal{S}^*| - 3|\tilde{\mathcal{V}}_B|)} \left(2 \min_{k, |\mathcal{S}|=3m/4} \sum_{i \in \mathcal{S}} \|\mathbf{y}_k - \mathbf{y}_i\|^2 + \sum_{i \in \tilde{\mathcal{V}}_R} 2\|\mathbf{y}_i - \bar{\mathbf{x}}\|^2 \right).
\end{aligned}$$

Taking expectation now on both sides yields

$$\mathbb{E}\|\mathbf{y}_{k^*} - \bar{\mathbf{x}}\|^2 \leq \frac{4m\tilde{\rho}^2}{|\mathcal{S}^*| - 3|\tilde{\mathcal{V}}_B|}.$$

Now, recall that we used a bucketing value of $s = \delta_{\max}/\delta$ where for KRUM we have $\delta_{\max} = 1/4 - \nu$. Then, the number of Byzantine workers can be bounded as $|\tilde{\mathcal{V}}_B| \leq m(1/4 - \nu)$. This gives the final result that

$$\mathbb{E}\|\mathbf{y}_{k^*} - \bar{\mathbf{x}}\|^2 \leq \frac{4m\tilde{\rho}^2}{3m/4 - 3(m/4 - \nu m)} = \frac{4\tilde{\rho}^2}{3\nu} \leq \frac{4}{3\nu(1/4 - \nu)}\delta\rho^2.$$

Thus, KRUM with bucketing indeed satisfies Definition 2.1 with $\delta_{\max} = (1/4 - \nu)$ and $c = 4/(3\nu(1/4 - \nu))$.

Robustness of Geometric median. Geometric median computes the minimum of the following optimization problem

$$\mathbf{y}^* = \arg \min_{\mathbf{y}} \sum_{i \in [m]} \|\mathbf{y} - \mathbf{y}_i\|_2.$$

We will adapt Lemma 24 of Cohen et al. [2016], which itself is based on [Minsker et al., 2015]. For a good bucket $i \in \tilde{\mathcal{V}}_R$ and bad bucket $j \in \tilde{\mathcal{V}}_B$:

$$\begin{aligned}
\|\mathbf{y}^* - \mathbf{y}_i\|_2 &\geq \|\mathbf{y}^* - \bar{\mathbf{x}}\|_2 - \|\mathbf{y}_i - \bar{\mathbf{x}}\|_2 \text{ for } i \in \tilde{\mathcal{V}}_R, \text{ and} \\
\|\mathbf{y}^* - \mathbf{y}_j\|_2 &\geq \|\mathbf{y}_j - \bar{\mathbf{x}}\|_2 - \|\mathbf{y}^* - \bar{\mathbf{x}}\|_2 \text{ for } j \in \tilde{\mathcal{V}}_B.
\end{aligned}$$

Summing this over all buckets we have

$$\begin{aligned}
\sum_{i \in [n]} \|\mathbf{y}^* - \mathbf{y}_i\| &\geq (|\tilde{\mathcal{V}}_R| - |\tilde{\mathcal{V}}_B|) \|\mathbf{y}^* - \bar{\mathbf{x}}\| + \sum_{j \in \tilde{\mathcal{V}}_B} \|\mathbf{y}_j - \bar{\mathbf{x}}\| - \sum_{i \in \tilde{\mathcal{V}}_R} \|\mathbf{y}_i - \bar{\mathbf{x}}\| \\
\Rightarrow \|\mathbf{y}^* - \bar{\mathbf{x}}\| &\leq \frac{1}{(|\tilde{\mathcal{V}}_R| - |\tilde{\mathcal{V}}_B|)} \left(\sum_{i \in [n]} \|\mathbf{y}^* - \mathbf{y}_i\| - \sum_{j \in \tilde{\mathcal{V}}_B} \|\mathbf{y}_j - \bar{\mathbf{x}}\| + \sum_{i \in \tilde{\mathcal{V}}_R} \|\mathbf{y}_i - \bar{\mathbf{x}}\| \right) \\
&= \frac{1}{(|\tilde{\mathcal{V}}_R| - |\tilde{\mathcal{V}}_B|)} \left(\min_{\mathbf{y}} \sum_{i \in [n]} \|\mathbf{y} - \mathbf{y}_i\| - \sum_{j \in \tilde{\mathcal{V}}_B} \|\mathbf{y}_j - \bar{\mathbf{x}}\| + \sum_{i \in \tilde{\mathcal{V}}_R} \|\mathbf{y}_i - \bar{\mathbf{x}}\| \right) \\
&\leq \frac{2}{(|\tilde{\mathcal{V}}_R| - |\tilde{\mathcal{V}}_B|)} \left(\sum_{i \in \tilde{\mathcal{V}}_R} \|\mathbf{y}_i - \bar{\mathbf{x}}\| \right).
\end{aligned}$$

The last step we substituted $\mathbf{y} = \bar{\mathbf{x}}$. Squaring both sides, expanding, and then taking expectation gives

$$\begin{aligned}
\mathbb{E} \|\mathbf{y}^* - \bar{\mathbf{x}}\|^2 &\leq \frac{4}{(|\tilde{\mathcal{V}}_R| - |\tilde{\mathcal{V}}_B|)^2} \mathbb{E} \left(\sum_{i \in \tilde{\mathcal{V}}_R} \|\mathbf{y}_i - \bar{\mathbf{x}}\| \right)^2 \\
&\leq \frac{4}{(|\tilde{\mathcal{V}}_R| - |\tilde{\mathcal{V}}_B|)^2} \left(|\tilde{\mathcal{V}}_R| \sum_{i \in \tilde{\mathcal{V}}_R} \mathbb{E} \|\mathbf{y}_i - \bar{\mathbf{x}}\|^2 \right) \\
&\leq \frac{4|\tilde{\mathcal{V}}_R|^2}{(n - 2|\tilde{\mathcal{V}}_B|)^2} \tilde{\rho}^2.
\end{aligned}$$

Now, recall that we used a bucketing value of $s = \delta_{\max}/\delta$ where for KRUM we have $\delta_{\max} = 1/2 - \nu$. Then, the number of Byzantine workers can be bounded as $|\tilde{\mathcal{V}}_B| \leq n(1/2 - \nu)$. This gives the final result that

$$\mathbb{E} \|\mathbf{y}^* - \bar{\mathbf{x}}\|^2 \leq \frac{4n^2}{4n^2\nu^2} \tilde{\rho}^2 \leq \frac{\tilde{\rho}^2}{\nu^2} \leq \frac{1}{\nu(1/2 - \nu)} \delta \rho^2.$$

Thus, geometric median with bucketing indeed satisfies Definition 2.1 with $\delta_{\max} = (1/2 - \nu)$ and $c = 1/(\nu(1/2 - \nu))$. Note that geometric median has better theoretical performance than KRUM.

Robustness of Coordinate-wise median. The proof of coordinate-wise median largely follows that of the geometric median. First, we note that we can separate out the objective by coordinates

$$\mathbb{E} \|\text{CM}(\mathbf{y}_1, \dots, \mathbf{y}_m) - \bar{\mathbf{x}}\|^2 = \sum_{l=1}^d \mathbb{E} (\text{CM}([\mathbf{y}_1]_l, \dots, [\mathbf{y}_m]_l) - [\bar{\mathbf{x}}]_l)^2.$$

Then note that, for any fixed coordinate $l \in [d]$ and fixed good worker $i \in \mathcal{V}_R$, we have $\mathbb{E} ([\mathbf{y}_i]_l - [\bar{\mathbf{x}}]_l)^2 \leq \mathbb{E} \|\mathbf{y}_i - \bar{\mathbf{x}}\|^2 \leq \tilde{\rho}^2$. Thus, we can simply analyze coordinate-wise median as d

separate (geometric) median problems on scalars. Thus for any fixed coordinate $l \in [d]$, we have

$$\begin{aligned} \mathbb{E}(\text{CM}([y_1]_l, \dots, [y_m]_l) - [\bar{x}]_l)^2 &\leq \frac{\tilde{\rho}^2}{\nu^2} \\ \Rightarrow \mathbb{E}\|\text{CM}(\mathbf{y}_1, \dots, \mathbf{y}_m) - \bar{\mathbf{x}}\|^2 &\leq \frac{d\tilde{\rho}^2}{\nu^2} \leq \frac{d}{\nu(1/2 - \nu)}\delta\rho^2. \end{aligned}$$

Thus, coordinate-wise median with bucketing indeed satisfies Definition 2.1 with $\delta_{\max} = (1/2 - \nu)$ and $c = d/(\nu(1/2 - \nu))$.

A.4 Lower bounds on non-iid data (Proof of Theorem 2.3)

Our proof builds two sets of functions $\{f_i^1(\mathbf{x}) \mid i \in \mathcal{V}_R^1\}$ and $\{f_i^2(\mathbf{x}) \mid i \in \mathcal{V}_R^2\}$ and will show that in the presence of δ -fraction of Byzantine workers, no algorithm can distinguish between them. Since the problems have different optima, this means that the algorithm necessarily has an error on at least one of them.

For the first set of functions, let there be *no* bad workers and hence $\mathcal{V}_R^1 = [n]$. Then, we define the following functions for any $i \in [n]$:

$$f_i^1(x) = \begin{cases} \frac{\mu}{2}x^2 - \zeta\delta^{-1/2}x & \text{for } i \in \{1, \dots, \delta n\} \\ \frac{\mu}{2}x^2 & \text{for } i \in \{\delta n + 1, \dots, n\}. \end{cases}$$

Defining $G := \zeta\delta^{1/2}$, the average function which is our objective is

$$f^1(x) = \frac{1}{n} \sum_{i=1}^n f_i^1(x) = \frac{\mu}{2}x^2 - Gx.$$

The optimum of our $f^1(x)$ is at $x = \frac{G}{\mu}$. Note that the gradient heterogeneity amongst these workers is bounded as

$$\begin{aligned} \mathbb{E}_{i \sim [n]} \|\nabla f_i^1(x) - \nabla f^1(x)\|^2 &= \delta(\zeta\delta^{-1/2} - \zeta\delta^{1/2})^2 + (1 - \delta)(\zeta\delta^{1/2})^2 \\ &= \zeta^2(1 - \delta)^2 + \zeta^2(1 - \delta)\delta = \zeta^2(1 - \delta) \leq \zeta^2. \end{aligned}$$

Now, we define the second set of functions. Here, suppose that we have δn Byzantine attackers with $\mathcal{V}_B^2 = \{1, \dots, \delta n\}$. Then, the functions of the good workers are defined as

$$f_i^2(x) = \frac{\mu}{2}x^2 \text{ for } i \in \mathcal{V}_R^2 = \{\delta n + 1, \dots, n\}.$$

We then have that the second average objective is

$$f^2(x) = \frac{1}{|\mathcal{V}_R^2|} \sum_{i \in \mathcal{V}_R^2} f_i^2(x) = \frac{\mu}{2} x^2.$$

Here, we have gradient heterogeneity of 0 and hence is smaller than ζ^2 . The optimum of $f^2(x)$ is at $x = 0$. The Byzantine attackers simply imitate as if they have the functions:

$$f_j^2(x) = \frac{\mu}{2} x^2 - \zeta \delta^{-1/2} x \text{ for } j \in \mathcal{V}_B^2 = \{1, \dots, \delta n\}.$$

Note that the set of functions, $\{f_1^1(\mathbf{x}), \dots, f_n^1(\mathbf{x})\}$ is exactly identical to the set $\{f_1^2(\mathbf{x}), \dots, f_n^2(\mathbf{x})\}$. Only the identity of the good workers \mathcal{V}_R^1 and \mathcal{V}_R^2 are different, leading to different objective functions $f^1(x)$ and $f^2(x)$. However, since the algorithm does not have access to \mathcal{V}_R , its output on each of them is identical i.e.

$$x^{\text{out}} = \text{ALG}(f_1^1(\mathbf{x}), \dots, f_n^1(\mathbf{x})) = \text{ALG}(f_1^2(\mathbf{x}), \dots, f_n^2(\mathbf{x})).$$

However, this leads to making a large error in least one of f^1 and f^2 since they have different optimum. This proves a lower bound error of

$$\max_{k \in \{1, 2\}} f^k(x^{\text{out}}) - f^k(x^*) \geq \mu \left(\frac{G}{2\mu} \right)^2 = \frac{\delta \zeta^2}{4\mu}.$$

Similarly, we can also bound the gradient norm error bound as

$$\max_{k \in \{1, 2\}} \|\nabla f^k(x^{\text{out}})\|^2 \geq \mu^2 \left(\frac{G}{2\mu} \right)^2 = \frac{\delta \zeta^2}{4}.$$

□

A.5 Convergence of robust optimization on non-iid data (Theorems 2.2 and 2.4)

We will prove a more general convergence theorem which generalizes Theorems 2.2 and 2.4.

Theorem A.1. *Suppose we are given a (δ_{\max}, c) -ARAGG satisfying Definition 2.1, and n workers of which a subset \mathcal{V}_R of size at least $|\mathcal{V}_R| \geq n(1 - \delta)$ faithfully follow the algorithm for $\delta \leq \delta_{\max}$. Further, for any good worker $i \in \mathcal{V}_R$ let f_i be a possibly non-convex function with L -Lipschitz gradients, and the stochastic gradients on each worker be independent, unbiased and satisfy*

$$\mathbb{E}_{\xi_i} \|\mathbf{g}_i(\mathbf{x}) - \nabla f_i(\mathbf{x})\|^2 \leq \sigma^2 \text{ and } \mathbb{E}_{j \sim \mathcal{V}_R} \|\nabla f_j(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \zeta^2 + B^2 \|\nabla f(\mathbf{x})\|^2, \quad \forall \mathbf{x},$$

where $\delta \leq 1/(60cB^2)$. Then, for $F^0 := f(\mathbf{x}^0) - f^*$, the output of Algorithm 2 with step-size $\eta = \min\left(\mathcal{O}\left(\sqrt{\frac{LF^0 + c\delta(\zeta^2 + \sigma^2)}{TL^2\sigma^2(n^{-1} + c\delta)}}\right), \frac{1}{8L}\right)$ and momentum parameter $\beta = (1 - 8L\eta)$ satisfies

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \|\nabla f(\mathbf{x}^{t-1})\|^2 \leq \mathcal{O}\left(\frac{1}{1-60c\delta B^2} \cdot \left(c\delta\zeta^2 + \sigma\sqrt{\frac{LF^0}{T}(c\delta + 1/n)} + \frac{LF^0}{T}\right)\right).$$

Notes on $\delta \leq 1/(60cB^2)$. In practice the upper bound $\delta \leq 1/(60cB^2)$ does not put an extra strict constraint on δ . This is because one can always decrease B^2 and increase ζ^2 such that $\mathbb{E}_{j \sim \mathcal{V}_R} \|\nabla f_j(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \zeta^2 + B^2 \|\nabla f(\mathbf{x})\|^2$ holds for a sufficiently large domain of \mathbf{x} .

Definitions. Recall our algorithm which performs for $t \geq 2$ the following update with $(1 - \beta) = \alpha$:

$$\begin{aligned} \mathbf{m}_i^t &= (1 - \alpha)\mathbf{m}_i^{t-1} + \alpha \mathbf{g}_i(\mathbf{x}^{t-1}) \quad \text{for every } i \in \mathcal{V}_R, \\ \mathbf{x}^t &= \mathbf{x}^{t-1} - \eta \text{ARAGG}(\mathbf{m}_1^t, \dots, \mathbf{m}_n^t). \end{aligned}$$

For $t = 1$, we use $\alpha = 0$ i.e. $\mathbf{m}_i^1 = \mathbf{g}_i(\mathbf{x}^0)$. Let us also define the actual and ideal momentum aggregates as

$$\mathbf{m}^t := \text{ARAGG}(\mathbf{m}_1^t, \dots, \mathbf{m}_n^t) \quad \text{and} \quad \bar{\mathbf{m}}^t := \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbf{m}_i^t.$$

We state several supporting lemmas before proving our main Theorem A.1. We will loosely follow the proof of Byzantine robustness in the iid case by Karimireddy et al. [2021b], with the key difference of Lemma A.2 which accounts for the non-iid error.

Lemma A.2 (Aggregation error). *Given that ARAGG satisfies Definition 2.1 holds, the error between the ideal average momentum $\bar{\mathbf{m}}^t$ and the output of the robust aggregation rule \mathbf{m}^t for any $t \geq 2$ can be bounded as*

$$\mathbb{E}\|\mathbf{m}^t - \bar{\mathbf{m}}^t\|^2 \leq c\delta\rho_t^2,$$

where we define for $t \geq 2$

$$\rho_t^2 := 4(6\alpha\sigma^2 + 3\zeta^2) + 4(6\sigma^2 - 3\zeta^2)(1 - \alpha)^t + 12 \sum_{k=1}^t (1 - \alpha)^{t-k} \alpha B^2 \mathbb{E}\|\nabla f(\mathbf{x}^{k-1})\|^2.$$

For $t = 1$ we can simplify the bound as $\rho_1^2 := 24c\delta\sigma^2 + 12c\delta\zeta^2 + 12c\delta B^2\|\nabla f(\mathbf{x}^0)\|^2$.

Proof. Let $\mathbb{E}_{\xi^t} := \mathbb{E}_{\{\xi_i^\tau\}_{i \in \mathcal{V}_R, \tau=0,1,\dots,t}}$ be the expectation with respect to all of the randomness of good workers until time t and let $\mathbb{E}_i := \mathbb{E}_{i \in \mathcal{V}_R}$ and $\mathbb{E} = \mathbb{E}_{\xi^t} \mathbb{E}_i$. Expanding the definition of the worker momentum for a fixed good worker $i \in \mathcal{V}_R$,

$$\begin{aligned} \mathbb{E}_{\xi_i^t} [\|\mathbf{m}_i^t - \mathbb{E}_{\xi^t}[\mathbf{m}_i^t]\|^2 | \mathbf{x}^{t-1}] &= \mathbb{E}_{\xi^t} [\|\alpha(\mathbf{g}_i(\mathbf{x}^{t-1}) - \nabla f_i(\mathbf{x}^{t-1})) + (1 - \alpha)(\mathbf{m}_i^{t-1} - \mathbb{E}_{\xi^t}[\mathbf{m}_i^{t-1}])\|^2 | \mathbf{x}^{t-1}] \\ &\leq \mathbb{E}_{\xi^{t-1}} \|(1 - \alpha)(\mathbf{m}_i^{t-1} - \mathbb{E}[\mathbf{m}_i^{t-1}])\|^2 + \alpha^2 \sigma^2 \\ &\leq (1 - \alpha) \mathbb{E}_{\xi^{t-1}} \|\mathbf{m}_i^{t-1} - \mathbb{E}[\mathbf{m}_i^{t-1}]\|^2 + \alpha^2 \sigma^2. \end{aligned}$$

Unrolling the recursion above yields

$$\mathbb{E}_{\xi^t} \|\mathbf{m}_i^t - \mathbb{E}_{\xi^t}[\mathbf{m}_i^t]\|^2 \leq \left(\sum_{\ell=2}^t (1 - \alpha)^{t-\ell} \right) \alpha^2 \sigma^2 + (1 - \alpha)^{t-1} \sigma^2 \leq \sigma^2 (\alpha + (1 - \alpha)^{t-1}).$$

Similar computation also shows

$$\mathbb{E}_{\xi^t} \|\bar{\mathbf{m}}^t - \mathbb{E}_{\xi^t}[\bar{\mathbf{m}}^t]\|^2 \leq \frac{\sigma^2}{n} (\alpha + (1 - \alpha)^{t-1}).$$

So far, the expectation was only over the stochasticity of the gradients of worker i . Note that we have $\mathbb{E}_{\xi^t}[\mathbf{m}_i^t] = \mathbb{E}_{\xi^{t-1}}[\alpha \nabla f_i(\mathbf{x}^{t-1}) + (1 - \alpha)\mathbf{m}_i^{t-1}]$. Now, suppose we sample i uniformly at random from \mathcal{V}_R . Then,

$$\begin{aligned} &\mathbb{E}_i [\|\mathbb{E}_{\xi^t}[\mathbf{m}_i^t] - \mathbb{E}_{\xi^t}[\bar{\mathbf{m}}^t]\|^2] \\ &= \mathbb{E}_i [\|\alpha \mathbb{E}_{\xi^{t-1}}[\nabla f_i(\mathbf{x}^{t-1}) - \nabla f(\mathbf{x}^{t-1})] + (1 - \alpha)(\mathbb{E}_{\xi^{t-1}}[\mathbf{m}_i^{t-1}] - \mathbb{E}_{\xi^{t-1}}[\bar{\mathbf{m}}^{t-1}])\|^2] \\ &\leq (1 - \alpha) \mathbb{E}_i [\|\mathbb{E}_{\xi^{t-1}}[\mathbf{m}_i^{t-1}] - \mathbb{E}_{\xi^{t-1}}[\bar{\mathbf{m}}^{t-1}]\|^2] + \alpha \mathbb{E}_i [\|\mathbb{E}_{\xi^{t-1}}[\nabla f_i(\mathbf{x}^{t-1}) - \nabla f(\mathbf{x}^{t-1})]\|^2] \\ &\leq (1 - \alpha) \mathbb{E}_i [\|\mathbb{E}_{\xi^{t-1}}[\mathbf{m}_i^{t-1}] - \mathbb{E}_{\xi^{t-1}}[\bar{\mathbf{m}}^{t-1}]\|^2] + \alpha \mathbb{E}_i \mathbb{E}_{\xi^{t-1}} \|\nabla f_i(\mathbf{x}^{t-1}) - \nabla f(\mathbf{x}^{t-1})\|^2 \\ &\leq (1 - \alpha) \mathbb{E}_i [\|\mathbb{E}_{\xi^{t-1}}[\mathbf{m}_i^{t-1}] - \mathbb{E}_{\xi^{t-1}}[\bar{\mathbf{m}}^{t-1}]\|^2] + \alpha \zeta^2 + \alpha B^2 \mathbb{E} \|\nabla f(\mathbf{x}^{t-1})\|^2 \end{aligned}$$

where the second inequality uses the probabilistic Jensen's inequality. Note that here we only get α instead of α^2 as before. This is because the randomness in the sampling i of $\nabla f_i(\mathbf{x}^{t-1})$ is

not independent of the second term $\mathbb{E}_{\xi^{t-1}}[\mathbf{m}_i^{t-1}] - \mathbb{E}_{\xi^{t-1}}[\bar{\mathbf{m}}^{t-1}]$. Expanding this we get,

$$\mathbb{E}_i \|\mathbb{E}_{\xi^t}[\mathbf{m}_i^t] - \mathbb{E}_{\xi^t}[\bar{\mathbf{m}}^t]\|^2 \leq \zeta^2(1 - (1 - \alpha)^t) + \sum_{k=1}^t (1 - \alpha)^{t-k} \alpha B^2 \mathbb{E} \|\nabla f(\mathbf{x}^{k-1})\|^2.$$

We can combine all three bounds above as

$$\begin{aligned} & \mathbb{E} \|\mathbf{m}_i^t - \bar{\mathbf{m}}^t\|^2 \\ & \leq 3 \mathbb{E} \|\mathbf{m}_i^t - \mathbb{E}_{\xi^t}[\mathbf{m}_i^t]\|^2 + 3 \mathbb{E} \|\bar{\mathbf{m}}^t - \mathbb{E}_{\xi^t}[\bar{\mathbf{m}}^t]\|^2 + 3 \mathbb{E}_i \|\mathbb{E}_{\xi^t}[\mathbf{m}_i^t] - \mathbb{E}_{\xi^t}[\bar{\mathbf{m}}^t]\|^2 \\ & = 3 \mathbb{E}_i \mathbb{E}_{\xi^t} \|\mathbf{m}_i^t - \mathbb{E}_{\xi^t}[\mathbf{m}_i^t]\|^2 + 3 \mathbb{E}_{\xi^t} \|\bar{\mathbf{m}}^t - \mathbb{E}_{\xi^t}[\bar{\mathbf{m}}^t]\|^2 + 3 \mathbb{E}_i \|\mathbb{E}_{\xi^t}[\mathbf{m}_i^t] - \mathbb{E}_{\xi^t}[\bar{\mathbf{m}}^t]\|^2 \\ & \leq (6\alpha\sigma^2 + 3\zeta^2) + (6\sigma^2 - 3\zeta^2)(1 - \alpha)^t + 3 \sum_{k=1}^t (1 - \alpha)^{t-k} \alpha B^2 \mathbb{E} \|\nabla f(\mathbf{x}^{k-1})\|^2. \end{aligned}$$

Therefore for $i, j \in \mathcal{V}_R$

$$\begin{aligned} \mathbb{E} \|\mathbf{m}_i^t - \mathbf{m}_j^t\|^2 & \leq 2 \mathbb{E} \|\mathbf{m}_i^t - \bar{\mathbf{m}}^t\|^2 + 2 \mathbb{E} \|\mathbf{m}_j^t - \bar{\mathbf{m}}^t\|^2 \\ & \leq 4(6\alpha\sigma^2 + 3\zeta^2) + 4(6\sigma^2 - 3\zeta^2)(1 - \alpha)^t \\ & \quad + 12 \sum_{k=1}^t (1 - \alpha)^{t-k} \alpha B^2 \mathbb{E} \|\nabla f(\mathbf{x}^{k-1})\|^2. \end{aligned}$$

Recall that the right hand side was defined to be ρ_t^2 . Using Definition 2.1, we can show that the output of the aggregation rule ARAGG satisfies the condition in the lemma. \square

One major caveat in the above lemma is that here ρ^2 cannot be known to the robust aggregation since it involves $\mathbb{E} \|\nabla f(\mathbf{x}^{k-1})\|^2$ whose value we do not have access to. However, this does not present a hurdle to *agnostic* aggregation rules which are automatically adaptive to the value of ρ^2 . Deriving a similarly provable adaptive clipping method is a very important open problem.

Lemma A.3 (Descent bound). *For any $\alpha \in [0, 1]$ for $t \geq 2$, $\eta \leq \frac{1}{L}$, and an L -smooth function f we have for any $t \geq 1$*

$$\mathbb{E}[f(\mathbf{x}^t)] \leq f(\mathbf{x}^{t-1}) - \frac{\eta}{2} \|\nabla f(\mathbf{x}^{t-1})\|^2 + \eta \mathbb{E} \|\bar{\mathbf{e}}^t\|^2 + \eta \mathbb{E} \|\mathbf{m}^t - \bar{\mathbf{m}}^t\|^2.$$

where $\bar{\mathbf{e}}^t := \bar{\mathbf{m}}^t - \nabla f(\mathbf{x}^{t-1})$.

Proof. By the smoothness of the function f and the server update,

$$\begin{aligned}
f(\mathbf{x}^t) &\leq f(\mathbf{x}^{t-1}) - \eta \langle \nabla f(\mathbf{x}^{t-1}), \mathbf{m}^t \rangle + \frac{L\eta^2}{2} \|\mathbf{m}^t\|^2 \\
&\leq f(\mathbf{x}^{t-1}) - \eta \langle \nabla f(\mathbf{x}^{t-1}), \mathbf{m}^t \rangle + \frac{\eta}{2} \|\mathbf{m}^t\|^2 \\
&= f(\mathbf{x}^{t-1}) + \frac{\eta}{2} \|\mathbf{m}^t - \nabla f(\mathbf{x}^{t-1})\|^2 - \frac{\eta}{2} \|\nabla f(\mathbf{x}^{t-1})\|^2 \\
&= f(\mathbf{x}^{t-1}) + \frac{\eta}{2} \|\mathbf{m}^t \pm \bar{\mathbf{m}}^t - \nabla f(\mathbf{x}^{t-1})\|^2 - \frac{\eta}{2} \|\nabla f(\mathbf{x}^{t-1})\|^2 \\
&\leq f(\mathbf{x}^{t-1}) + \eta \|\bar{\mathbf{e}}^t\|^2 + \eta \|\mathbf{m}^t - \bar{\mathbf{m}}^t\|^2 - \frac{\eta}{2} \|\nabla f(\mathbf{x}^{t-1})\|^2.
\end{aligned}$$

Here we used the identities that $-2ab = (a - b)^2 - a^2 - b^2$, and Young's inequality that $(a + b)^2 \leq (1 + \gamma)a^2 + (1 + \frac{1}{\gamma})b^2$ for any positive constant $\gamma \geq 0$ (here we used $\gamma = 1$). Taking conditional expectation on both sides yields the lemma. \square

Lemma A.4 (Error bound). *Using any constant momentum and step-sizes such that $1 \geq \alpha \geq 8L\eta$ for $t \geq 2$, we have for an L -smooth function f that $\mathbb{E}\|\bar{\mathbf{e}}^1\|^2 \leq \frac{2\sigma^2}{n}$ and for $t \geq 2$*

$$\mathbb{E}\|\bar{\mathbf{e}}^t\|^2 \leq (1 - \frac{2\alpha}{5}) \mathbb{E}\|\bar{\mathbf{e}}^{t-1}\|^2 + \frac{\alpha}{10} \mathbb{E}\|\nabla f(\mathbf{x}^{t-2})\|^2 + \frac{\alpha}{10} \mathbb{E}\|\mathbf{m}^{t-1} - \bar{\mathbf{m}}^{t-1}\|^2 + \alpha^2 \frac{2\sigma^2}{n}.$$

Proof. Let us define $\bar{\mathbf{g}}(\mathbf{x}) := \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbf{g}_i(\mathbf{x})$. This implies that

$$\mathbb{E}\|\bar{\mathbf{g}}(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \frac{\sigma^2}{|\mathcal{V}_R|} \leq \frac{2\sigma^2}{n}.$$

Then by definition of $\bar{\mathbf{m}}$, we can expand the error as:

$$\begin{aligned}
\mathbb{E}\|\bar{\mathbf{e}}^t\|^2 &= \mathbb{E}\|\bar{\mathbf{m}}^t - \nabla f(\mathbf{x}^{t-1})\|^2 \\
&= \mathbb{E}\|\alpha \bar{\mathbf{g}}(\mathbf{x}^{t-1}) + (1 - \alpha) \bar{\mathbf{m}}^{t-1} - \nabla f(\mathbf{x}^{t-1})\|^2 \\
&\leq \mathbb{E}\|\alpha \nabla f(\mathbf{x}^{t-1}) + (1 - \alpha) \bar{\mathbf{m}}^{t-1} - \nabla f(\mathbf{x}^{t-1})\|^2 + \frac{2\alpha^2\sigma^2}{n} \\
&= (1 - \alpha)^2 \mathbb{E}\|(\bar{\mathbf{m}}^{t-1} - \nabla f(\mathbf{x}^{t-2})) + (\nabla f(\mathbf{x}^{t-2}) - \nabla f(\mathbf{x}^{t-1}))\|^2 + \frac{2\alpha^2\sigma^2}{n} \\
&\leq (1 - \alpha)(1 + \frac{\alpha}{2}) \mathbb{E}\|(\bar{\mathbf{m}}^{t-1} - \nabla f(\mathbf{x}^{t-2}))\|^2 \\
&\quad + (1 - \alpha)(1 + \frac{2}{\alpha}) \mathbb{E}\|\nabla f(\mathbf{x}^{t-2}) - \nabla f(\mathbf{x}^{t-1})\|^2 + \frac{2\alpha^2\sigma^2}{n} \\
&\leq (1 - \frac{\alpha}{2}) \mathbb{E}\|\bar{\mathbf{e}}^{t-1}\|^2 + \frac{2L^2}{\alpha} \mathbb{E}\|\mathbf{x}^{t-2} - \mathbf{x}^{t-1}\|^2 + \frac{2\alpha^2\sigma^2}{n} \\
&= (1 - \frac{\alpha}{2}) \mathbb{E}\|\bar{\mathbf{e}}^{t-1}\|^2 + \frac{2L^2\eta^2}{\alpha} \mathbb{E}\|\mathbf{m}^{t-1}\|^2 + \frac{2\alpha^2\sigma^2}{n} \\
&\leq (1 - \frac{\alpha}{2}) \mathbb{E}\|\bar{\mathbf{e}}^{t-1}\|^2 + \frac{6L^2\eta^2}{\alpha} \mathbb{E}\|\bar{\mathbf{e}}^{t-1}\|^2 \\
&\quad + \frac{6L^2\eta^2}{\alpha} \mathbb{E}\|\mathbf{m}^{t-1} - \bar{\mathbf{m}}^{t-1}\|^2 + \frac{6L^2\eta^2}{\alpha} \mathbb{E}\|\nabla f(\mathbf{x}^{t-2})\|^2 + \frac{2\alpha^2\sigma^2}{n}.
\end{aligned}$$

Our choice of the momentum parameter α implies $64L^2\eta^2 \leq \alpha^2$ and yields the lemma statement. \square

Proof of Theorem A.1. Scale the error bound Lemma A.4 by $\frac{5\eta}{2\alpha}$ and add it to the descent bound Lemma A.3 taking expectations on both sides to get for $t \geq 2$

$$\begin{aligned} \mathbb{E}[f(\mathbf{x}^t)] + \frac{5\eta}{2\alpha} \mathbb{E}\|\bar{\mathbf{e}}^t\|^2 &\leq \mathbb{E}[f(\mathbf{x}^{t-1})] - \frac{\eta}{2} \mathbb{E}\|\nabla f(\mathbf{x}^{t-1})\|^2 + \eta \mathbb{E}\|\bar{\mathbf{e}}^t\|^2 + \eta \mathbb{E}\|\mathbf{m}^t - \bar{\mathbf{m}}^t\|^2 + \\ &\quad \frac{5\eta}{2\alpha} \mathbb{E}\|\bar{\mathbf{e}}^{t-1}\|^2 - \eta \mathbb{E}\|\bar{\mathbf{e}}^{t-1}\|^2 + \frac{\eta}{4} \mathbb{E}\|\nabla f(\mathbf{x}^{t-2})\|^2 \\ &\quad + \frac{\eta}{4} \mathbb{E}\|\mathbf{m}^{t-1} - \bar{\mathbf{m}}^{t-1}\|^2 + 5\eta\alpha \frac{\sigma^2}{n}. \end{aligned}$$

Now, let use the aggregation error Lemma A.2 to bound $\mathbb{E}\|\mathbf{m}^{t-1} - \bar{\mathbf{m}}^{t-1}\|^2$ and $\mathbb{E}\|\mathbf{m}^t - \bar{\mathbf{m}}^t\|^2$ in the above expression to get

$$\begin{aligned} \mathbb{E}[f(\mathbf{x}^t)] + \frac{5\eta}{2\alpha} \mathbb{E}\|\bar{\mathbf{e}}^t\|^2 &\leq \mathbb{E}[f(\mathbf{x}^{t-1})] - \frac{\eta}{2} \mathbb{E}\|\nabla f(\mathbf{x}^{t-1})\|^2 + \eta \mathbb{E}\|\bar{\mathbf{e}}^t\|^2 \\ &\quad + \frac{5\eta}{2\alpha} \mathbb{E}\|\bar{\mathbf{e}}^{t-1}\|^2 - \eta \mathbb{E}\|\bar{\mathbf{e}}^{t-1}\|^2 + \frac{\eta}{4} \mathbb{E}\|\nabla f(\mathbf{x}^{t-2})\|^2 + 5\eta\alpha \frac{\sigma^2}{n} \\ &\quad + 5\eta c\delta((6\alpha\sigma^2 + 3\zeta^2) + 6\sigma^2(1-\alpha)^{t-2}) \\ &\quad + \eta c\delta\left(3 \sum_{k=1}^{t-1} (1-\alpha)^{t-1-k} \alpha B^2 \mathbb{E}\|\nabla f(\mathbf{x}^{k-1})\|^2\right) \\ &\quad + 4\eta c\delta\left(3 \sum_{k=1}^t (1-\alpha)^{t-k} \alpha B^2 \mathbb{E}\|\nabla f(\mathbf{x}^{k-1})\|^2\right). \end{aligned}$$

Let us define $S_t := \sum_{k=1}^t (1-\alpha)^{t-k} \alpha B^2 \mathbb{E}\|\nabla f(\mathbf{x}^{k-1})\|^2$. Then, S_t satisfies the recursion:

$$\frac{1}{\alpha} S_t = \left(\frac{1}{\alpha} - 1\right) S_{t-1} + B^2 \mathbb{E}\|\nabla f(\mathbf{x}^{t-1})\|^2.$$

Adding $\frac{3\eta c\delta(\frac{5}{\alpha}-4)}{\alpha} S_t$ on both sides to the bound above and rearranging gives the following for $t \geq 2$

$$\begin{aligned}
& \underbrace{\mathbb{E} f(\mathbf{x}^t) - f^\star + \left(\frac{5\eta}{2\alpha} - \eta\right) \mathbb{E} \|\bar{\mathbf{e}}^t\|^2 + \frac{\eta}{4} \mathbb{E} \|\nabla f(\mathbf{x}^{t-1})\|^2 + \frac{3\eta c \delta (\frac{5}{\alpha} - 4)}{\alpha} S_t}_{=:\mathcal{E}_t} \\
& \leq \underbrace{\mathbb{E} f(\mathbf{x}^{t-1}) - f^\star + \left(\frac{5\eta}{2\alpha} - \eta\right) \mathbb{E} \|\bar{\mathbf{e}}^{t-1}\|^2 + \frac{\eta}{4} \mathbb{E} \|\nabla f(\mathbf{x}^{t-2})\|^2 + \frac{3\eta c \delta (\frac{5}{\alpha} - 4)}{\alpha} S_{t-1}}_{=:\mathcal{E}_{t-1}} \\
& \quad \left(-\frac{\eta}{4} + 15\eta c \delta B^2\right) \mathbb{E} \|\nabla f(\mathbf{x}^{t-1})\|^2 \\
& \quad + \frac{5\eta\alpha}{n} \sigma^2 + 5\eta c \delta ((6\alpha\sigma^2 + 3\zeta^2) + 6\sigma^2(1-\alpha)^{t-2}) \\
& \leq \mathcal{E}_{t-1} - \frac{\eta}{4} (1 - 60c\delta B^2) \mathbb{E} \|\nabla f(\mathbf{x}^{t-1})\|^2 \\
& \quad + \underbrace{5\eta\alpha\sigma^2 \left(\frac{1}{n} + 6c\delta(1 + \frac{1}{\alpha}(1-\alpha)^{t-2})\right) + 15\eta c \delta \zeta^2}_{=:\eta\xi_{t-1}^2}.
\end{aligned}$$

Further, specializing the descent bound Lemma A.3 and error bound Lemma A.4 for $t = 1$ we have

$$\begin{aligned}
\mathcal{E}_1 &= \mathbb{E} f(\mathbf{x}^1) - f^\star + \frac{3\eta}{2} \mathbb{E} \|\bar{\mathbf{e}}^1\|^2 + \frac{\eta}{4} \mathbb{E} \|\nabla f(\mathbf{x}^0)\|^2 + 3\eta c \delta B^2 \left(\frac{5}{\alpha} - 4\right) \|\nabla f(\mathbf{x}^0)\|^2 \\
&\leq f(\mathbf{x}^0) - f^\star + \frac{5\eta}{2} \mathbb{E} \|\bar{\mathbf{e}}^1\|^2 - \frac{\eta}{4} (1 - 60c\delta B^2) \mathbb{E} \|\nabla f(\mathbf{x}^0)\|^2 + \eta \mathbb{E} \|\mathbf{m}_1 - \bar{\mathbf{m}}_1\|^2 \\
&\leq f(\mathbf{x}^0) - f^\star - \frac{\eta}{4} (1 - 60c\delta B^2) \mathbb{E} \|\nabla f(\mathbf{x}^0)\|^2 + \frac{5\eta\sigma^2}{n} + 12c\delta\eta(2\sigma^2 + \zeta^2 + B^2 \|\nabla f(\mathbf{x}^0)\|^2) \\
&= f(\mathbf{x}^0) - f^\star - \frac{\eta}{4} (1 - 60c\delta B^2) \mathbb{E} \|\nabla f(\mathbf{x}^0)\|^2 + \eta\xi_0^2.
\end{aligned}$$

Above, we defined $\xi_0^2 := \frac{5\sigma^2}{n} + 12c\delta(2\sigma^2 + \zeta^2 + B^2\|\nabla f(\mathbf{x}^0)\|^2)$. Summing over t from 2 until T , again rearranging our recursion for \mathcal{E}_t , and adding $(1 - 3c\delta B^2) \mathbb{E}\|\nabla f(\mathbf{x}^0)\|^2$ on both sides gives

$$\begin{aligned}
(1 - 60c\delta B^2) \frac{1}{T} \sum_{t=1}^T \mathbb{E}\|\nabla f(\mathbf{x}^{t-1})\|^2 &\leq \frac{4(f(\mathbf{x}^0) - f^*)}{\eta T} + \frac{1}{T} \sum_{t=1}^T 4\xi_{t-1}^2 \\
&= \frac{4(f(\mathbf{x}^0) - f^*)}{\eta T} + \frac{4\xi_0^2}{T} \\
&\quad + \frac{1}{T} \sum_{t=2}^T 20\alpha\sigma^2 \left(\frac{1}{n} + 6c\delta(1 + \frac{1}{\alpha}(1 - \alpha)^{t-2})\right) \\
&\quad + \frac{1}{T} \sum_{t=2}^T 60c\delta\zeta^2 \\
&\leq \frac{4(f(\mathbf{x}^0) - f^*)}{\eta T} + \frac{4\xi_0^2}{T} + 60c\delta\zeta^2 \\
&\quad + 20\alpha\sigma^2 \left(\frac{1}{n} + 6c\delta\right) + \frac{120c\delta\sigma^2}{\alpha T} \\
&= \frac{4(f(\mathbf{x}^0) - f^*)}{\eta T} + \frac{120c\delta\sigma^2}{\eta 8LT} + \eta 160L\sigma^2 \left(\frac{1}{n} + 6c\delta\right) \\
&\quad + \frac{4\xi_0^2}{T} + 60c\delta\zeta^2.
\end{aligned}$$

The last equality substituted the value of $\alpha = 8L\eta$. Next, let us use the appropriate step-size of

$$\eta = \min \left(\sqrt{\frac{4(f(\mathbf{x}_0) - f^*) + \frac{15c\delta}{L}(\zeta^2 + 2\sigma^2)}{T(160L\sigma^2)(\frac{1}{n} + 6c\delta)}}, \frac{1}{8L} \right).$$

This gives the following final rate of convergence:

$$\begin{aligned}
&\frac{1}{T} \sum_{t=1}^T \mathbb{E}\|\nabla f(\mathbf{x}^{t-1})\|^2 \\
&\leq \frac{1}{1 - 60c\delta B^2} \cdot \left(60c\delta\zeta^2 + \sqrt{\frac{160L\sigma^2(\frac{1}{n} + 6c\delta)}{T}} \cdot \sqrt{4(f(\mathbf{x}_0) - f^*) + \frac{15c\delta}{L}(\zeta^2 + 2\sigma^2)} \right. \\
&\quad + \frac{32L(f(\mathbf{x}^0) - f^*)}{T} + \frac{15c\delta\sigma^2}{T} \\
&\quad \left. + \frac{\frac{20\sigma^2}{n} + 12c\delta(2\sigma^2 + \zeta^2 + B^2\|\nabla f(\mathbf{x}^0)\|^2)}{T} \right).
\end{aligned}$$

□

Appendix B

Byzantine-robust decentralized learning via ClippedGossip

B.1 Existing robust aggregators

In this section, we describe existing robust aggregators mentioned in this paper. Regular nodes can replace gossip averaging ([GOSSIP](#)) with robust aggregators in the federated learning. Let's take geometric median and trimmed mean for example.

- **Geometric median (GM).** [Pillutla et al. \[2019\]](#) implements the geometric median

$$\text{GM}(\mathbf{x}_1, \dots, \mathbf{x}_n) := \arg \min_{\mathbf{v}} \sum_{i=1}^n \|\mathbf{v} - \mathbf{x}_i\|_2.$$

- **Coordinate-wise trimmed mean (TM).** [Yang and Bajwa \[2019a\]](#); [Yin et al. \[2018b\]](#) computes the k -th coordinate of TM as

$$[\text{TM}(\mathbf{x}_1, \dots, \mathbf{x}_n)]_k := \frac{1}{(1-2\beta)n} \sum_{i \in U_k} [\mathbf{x}_i]_k$$

where U_k is a subset of $[n]$ obtained by removing the largest and smallest β -fraction of its elements.

These aggregators don't take advantage of the trusted local information and treat all models equally.

The MOZI algorithm [[Guo et al., 2021](#)] leverages local information to filter outliers.

- **Mozi.** [Guo et al. \[2021\]](#) applies two screening steps on worker $i \in \mathcal{V}_R$

$$\begin{aligned} \mathcal{N}_i^s &:= \arg \min_{\substack{\mathcal{N}^* \subseteq \mathcal{N}_i \\ |\mathcal{N}^*| = \delta_i |\mathcal{N}_i|}} \sum_{j \in \mathcal{N}^*} \|\mathbf{x}_i - \mathbf{x}_j\|, \\ \mathcal{N}_i^r &:= \mathcal{N}_i^s \cap \{j \in [n] : \ell(\mathbf{x}_j, \xi_i) \leq \ell(\mathbf{x}_i, \xi_i)\} \end{aligned}$$

where $\xi_i \sim \mathcal{D}_i$ is a random sample. If $\mathcal{N}_i^r = \emptyset$, then redefine $\mathcal{N}_i^r := \{\arg \min_j \ell(\mathbf{x}_j, \xi_i)\}$. Then they update the model with

$$\mathbf{x}_i^{t+1} := \alpha \mathbf{x}_i^t + \frac{1-\alpha}{|\mathcal{N}_i^r|} \sum_{j \in \mathcal{N}_i^r} \mathbf{x}_j^t - \eta \nabla F_i(\mathbf{x}_i^t, \xi_i^t)$$

where $\alpha \in [0, 1]$ is an hyperparameter.

B.2 Byzantine attacks in the decentralized environment

In this section, we first describe how to transform attacks from the federated learning to the decentralized environment. Then we introduce the *dissensus* attack for decentralized environment.

B.2.1 Existing attacks in federated learning

A little is enough (ALIE). The attackers estimate the mean $\mu_{\mathcal{N}_i}$ and standard deviation $\sigma_{\mathcal{N}_i}$ of the regular models, and send $\mu_{\mathcal{N}_i} - z\sigma_{\mathcal{N}_i}$ to regular worker i where z is a small constant controlling the strength of the attack [Baruch et al., 2019]. The hyperparameter z for ALIE is computed according to [Baruch et al., 2019]

$$z = \max_z \left(\phi(z) < \frac{n-b-s}{n-b} \right) \quad (\text{B.1})$$

where $s = \lfloor \frac{n}{2} + 1 \rfloor - b$ and ϕ is the cumulative standard normal function.

Inner product manipulation attack (IPM). The inner product manipulation attack is proposed in [Xie et al., 2019a] which lets all attackers send same corrupted gradient \mathbf{u} based on the good gradients

$$\mathbf{u}_j = -\epsilon \text{AVG}(\{\mathbf{v}_i : i \in \mathcal{V}_R\}) \quad \forall j \in \mathcal{V}_B.$$

If ϵ is small enough, then \mathbf{u}_j can be detected as **good** by the defense, circumventing the defense. There are 3 main differences where IPM need to adapt to the decentralized environment:

1. Byzantine workers may not connected to the same good worker.
2. The model vectors are transmitted instead of gradients.
3. The AVG should be replaced by its equivalent gossip form.

This motivates our *dissensus* attack in the next section.

B.2.2 Dissensus attack and other attacks in the decentralized environment

In this section, we introduce a novel *dissensus* attack inspired by our impossibility construction in Theorem 3.2 and the IPM attack described above. The dissensus attack aims to prevent regular worker models from reaching consensus. Roughly speaking, dissensus attackers around worker i send its model weights that are symmetric to the weighted average of regular neighbors around i . Then after gossip averaging step, the consensus distance drops slower or even grows which motivates the name “dissensus”.

We can parameterize the attack through hyperparameter ϵ_i and summarize the attack in Definition 3.5

$$\mathbf{x}_j := \mathbf{x}_i - \epsilon_i \frac{\sum_{k \in \mathcal{N}_i \cap \mathcal{V}_R} \mathbf{W}_{ik}(\mathbf{x}_k - \mathbf{x}_i)}{\sum_{j \in \mathcal{N}_i \cap \mathcal{V}_B} \mathbf{W}_{ij}}. \quad (\text{B.2})$$

The ϵ_i determines the behavior of the attack. By taking smaller ϵ_i , Byzantine model weights are closer to the target updates i and difficult to be detected. On the other hand, a larger ϵ_i pulls the model away from the consensus.

Note that this attack requires omniscience since it exploits model information from across the network. If the attackers in addition can choose which node to attack, then they can choose either to spread about the attack across the network or focus on the targeting graph cut, that is min-cut of the graph.

Effect of the dissensus attack. The dissensus attack enjoy the following properties.

Proposition B.1. (i) For all $i \in \mathcal{V}_R$, under the dissensus attack with $\epsilon_i = 1$, the gossip averaging step (GOSSIP) is equivalent to no communication on i , $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t$. Secondly, (ii) If the graph is fully connected, gossip averaging recovers the correct consensus even in the presence of dissensus attack.

The above proposition illustrates two interesting aspects of the attack. Firstly, dissensus works by negating the progress that would be made by gossip. The attack in [Peng and Ling, 2020] also satisfies this property (see Appendix for additional discussion). Secondly, it is a uniquely decentralized attack and has no effect in the centralized setting. Hence, its effect can be used to measure the additional difficulty posed due to the restricted communication topology.

Proof. For the first part, by definition (GOSSIP) we know that

$$\mathbf{x}_i^{t+1} = \sum_{j=1}^n \mathbf{W}_{ij} \mathbf{x}_j^t = \mathbf{x}_i^t + \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij} (\mathbf{x}_j^t - \mathbf{x}_i^t)$$

By setting $\epsilon_i = 1$ in the attack (3.6), the second term 0 and therefore $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t$. For part (ii), note that in a fully connected graph the gossip average is the same as standard average.

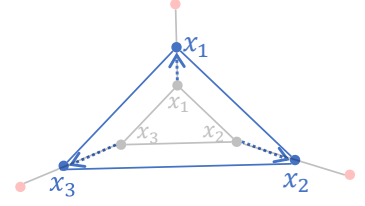


Fig. B.1 Example of the DISSENSUS attack. The gray (resp. red) denotes regular (resp. Byzantine) nodes. The blue dots represents the parameters of regular nodes after gossip steps.

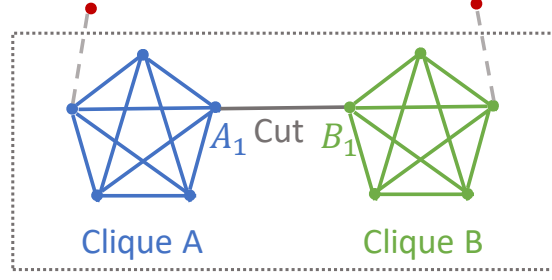


Fig. B.2 Example topology that does not satisfy the robust network assumptions in [Su and Vaidya, 2016a; Sundaram and Gharesifard, 2018].

Averaging all the perturbations introduced by the dissensus attack gives

$$-\epsilon \sum_{i,j \in \mathcal{V}_R} W_{i,j} (\mathbf{x}_j^t - \mathbf{x}_i^t) = 0.$$

All terms cancel and sum to 0 by symmetry. Thus, in a fully connected graph the dissensus perturbations cancel out and the gossip average returns the correct consensus. \square

Relation with zero-sum attack and dissensus. Peng and Ling [2020] propose the “zero-sum” attack which achieves similar effects as Proposition B.1 part (i). This attack is defined for $j \in \mathcal{V}_B$

$$\mathbf{x}_j := -\frac{\sum_{k \in \mathcal{N}_i \cap \mathcal{V}_R} \mathbf{x}_k}{|\mathcal{N}_i \cap \mathcal{V}_B|}.$$

The key difference between zero-sum attack and our proposed attack is three-fold. First, zero-sum attack ensures $\sum_{j \in \mathcal{N}_i} \mathbf{x}_j = 0$ which means the Byzantine models have to be far away from \mathbf{x}_i^t and therefore easy to detect. This attack pull the aggregated model to $\mathbf{0}$. On the other hand, our attack ensures

$$\frac{1}{\sum_{j \in \mathcal{N}_i} W_{ij}} \sum_{j \in \mathcal{N}_i} W_{ij} \mathbf{x}_j^t = \mathbf{x}_i^t$$

and the Byzantine updates can be very close to \mathbf{x}_i^t and it is more difficult to be detected. Second, our proposed attack considers the gossip averaging which is prevalent in decentralized training [Koloskova et al., 2020b] while the zero-sum attack only targets simple average. Third, our attack has an additional parameter ϵ controlling the strength of the attack with $\epsilon > 1$ further compromise the model quality while zero-sum attack is fixed to training alone.

B.3 Topologies and mixing matrices

B.3.1 Constrained topologies

Topologies that do not satisfy the robust network assumption in [LeBlanc et al., 2013; Su and Vaidya, 2016a; Sundaram and Gharesifard, 2018]. The robust network

assumption requires there to be at least $b + 1$ paths between any two regular workers when there are b Byzantine workers in the network [LeBlanc et al., 2013; Su and Vaidya, 2016a; Sundaram and Gharesifard, 2018]. The topology in Figure B.2 only has 1 path between regular workers in two cliques while having 2 Byzantine workers in the network. Therefore this topology does not satisfy the robust network assumption. But the graph cut is not adjacent to the Byzantine workers and, intuitively, it would be possible for an ideal robust aggregator to help reach consensus. The experimental results are given in § B.4.4.

(Randomized) Small-world topology. The small-world topology is a random graph generated with Watts-Strogatz model [Watts and Strogatz, 1998]. The topology is created using NetworkX package [Hagberg et al., 2008a] with 10 regular workers each connected to 2 nearest neighbors and probability of rewiring each edge as 0.15. Two additional Byzantine workers are linked to 2 random regular workers. There are 12 workers in total.

Torus topology. The regular workers form a torus grid $T_{3,3}$ and two additional Byzantine workers are linked to 2 random regular workers. There are 11 workers in total.

The mixing matrix for these topologies are constructed with Metropolis-Hastings algorithm introduced in the previous section. The spectral gap for small-world topology and torus topology are 0.084 and 0.131 respectively. In contrast, the dumbbell topology in Figure B.10 is more challenging with a spectral gap of 0.043. The data distribution is non-IID.

B.3.2 Constructing mixing matrices

In this section, we introduce a few possible ways to construct the mixing weight vectors in the presence of Byzantine workers. The constructed weight vectors satisfy Assumption B in § 3.4.

- **Metropolis-Hastings weight [Hastings, 1970].** The Metropolis-Hastings algorithm locally constructs the mixing weights by exchanging degree information (d_i and d_j) between two nodes i and j . The mixing weight vector on regular worker $i \in \mathcal{V}_R$ is computed as follows

$$\mathbf{W}_{ij} = \begin{cases} \frac{1}{\max\{d_i, d_j\} + 1} & j \in \mathcal{N}_i, \\ 1 - \sum_{l \in \mathcal{N}_i} \mathbf{W}_{il} & j = i, \\ 0 & \text{Otherwise.} \end{cases}$$

If worker $j \in \mathcal{V}_B$ is Byzantine, then the only way for j to maximize its weight \mathbf{W}_{ij} to regular worker i is to report a smaller degree d_j . However, such Byzantine behavior of node j has limited influence on worker i 's weight \mathbf{W}_{ij} because it can not be greater than $\frac{1}{d_i + 1}$.

- **Equal-weight.** Let d_{\max} be the maximum degree of nodes in a graph. Such upper bound d_{\max} can be a public information, for example, a bluetooth device can at most connect to d_{\max} other devices due to physical constraints. The Byzantine worker cannot change the

value of d_{\max} . Then we use the following naive construction

$$\mathbf{W}_{ij} = \begin{cases} \frac{1}{d_{\max}+1} & j \in \mathcal{N}_i, \\ 1 - \frac{|\mathcal{N}_i|}{d_{\max}+1} & j = i, \\ 0 & \text{Otherwise.} \end{cases} \quad (\text{B.3})$$

Note that these construction schemes are not proved to be the optimal. In this work, we focus on the Byzantine attacks given a topology and associated mixing weights. We leave it as future work to explore the best strategy to construct mixing weights.

B.4 Experiments

We summarize the hardware and software for experiments in Table B.1. We list the setups and

Table B.1 Runtime hardwares and softwares.

CPU	
Model name	Intel (R) Xeon (R) Gold 6132 CPU @ 2.60 GHz
# CPU(s)	56
NUMA node(s)	2
GPU	
Product Name	Tesla V100-SXM2-32GB
CUDA Version	11.0
PyTorch	
Version	1.7.1

results of experiments for consensus in § B.4.1 and optimization in § B.4.2.

B.4.1 Byzantine-robust consensus

In this section, we provide detailed setups for Figure 3.3. The Figure B.3 demonstrates the topology for the experiment. The 4 regular workers are connected with two of them holding value 0 and the others holding 200. Then the average consensus is 100 with initial mean square error equals 10000. Two Byzantine workers are connected to two regular workers in the middle. We can tune the weights of each edge to change the mixing matrix and γ . Then we can decide the weight δ on the Byzantine edge. The γ and δ used in the experiments are

- $p := 1 - (1 - \gamma)^2 \in [0.06, 0.05, 0.04, 0.03, 0.02, 0.01, 0.005, 0.0014, 3.7\text{e-}4, 1\text{e-}4, 1\text{e-}5]$
- $\delta \in [0.05, 0.1, 0.2, 0.3, 0.4, 0.5]$

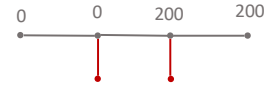


Fig. B.3 The topology for the attacks on consensus. The grey and red nodes denote regular and Byzantine workers respectively.

Table B.2 Default experimental settings for MNIST

Dataset	MNIST
Architecture	CONV-CONV-DROPOUT-FC-DROPOUT-FC
Training objective	Negative log likelihood loss
Evaluation objective	Top-1 accuracy
Batch size per worker	32
Momentum	0.9
Learning rate	0.01
LR decay	No
LR warmup	No
Weight decay	No
Repetitions	1
Reported metric	Mean test accuracy over the last 150 iterations

where non-compatible combination of γ and δ are ignored in the Figure 3.3. The dissensus attack is applied with $\epsilon = 0.05$. The hyperparameter β of trimmed mean (TM) is set to the actual number of Byzantine workers around the regular worker. The clipping radius of CLIPPEDGOSSIP is chosen according to (B.21).

In Figure B.4, we show the iteration-to-error curves for all possible combinations of γ and δ . In addition, we provide a version of TM and MEDIAN which takes the mixing weight into account. As we can see, the naive TM, MEDIAN, and MEDIAN* cannot bring workers closer because of the data distribution we constructed. The TM* is performing better than the other baselines but worse than CLIPPEDGOSSIP especially on the challenging cases where γ is small and δ is large. For CLIPPEDGOSSIP, it matches with our intuition that for a fixed γ the convergences is worse with increasing δ while for a fixed δ the convergence is worse with decreasing γ .

B.4.2 Byzantine-robust decentralized optimization

In this section, we provide detailed hyperparameters and setups for experiments in the main text and then provide additional experiments. For all MNIST tasks, we use the default setup listed in Table B.2 unless specifically stated. The default hyperparameters of the robust aggregators: 1) For GM, we choose number of iterations $T = 8$; 2) The TM drops top and bottom $\beta = \delta_{\max}n$ percent of values in each coordinate; 3) The clipping radius of CLIPPEDGOSSIP is $\tau = 1$; 4) The model averaging hyperparameter of MOZI is $\alpha = 1$.

Setup for “Decentralized defenses without attackers”

The Fig. 3.4 uses the dumbbell topology in Fig. 3.1 with 10 regular workers in each clique. There is no Byzantine workers. The experiments run for 900 iterations. MOZI uses $\alpha = 0.5$ and $\rho_i = 0.99$ in this setting. For bucketing experiment, we choose bucket size of $s = 2$. It means we

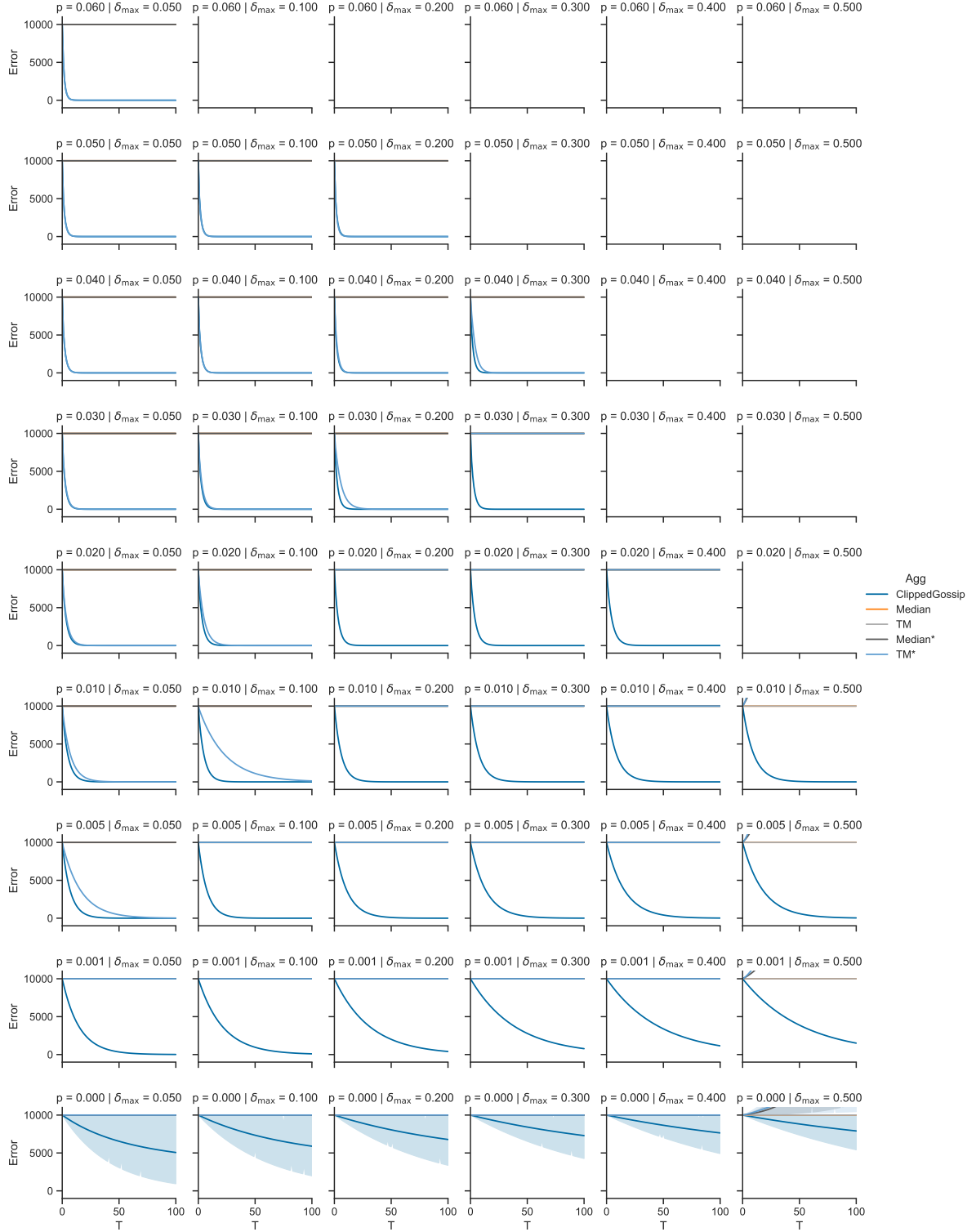


Fig. B.4 The iteration-to-error curves for defenses under dissensus attack. The TM* and MEDIAN* refer to the version of TM and MEDIAN which considers mixing weight.

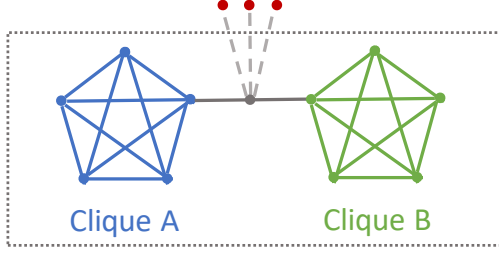


Fig. B.5 Dumbbell variant where Byzantine workers maybe added to the central worker.

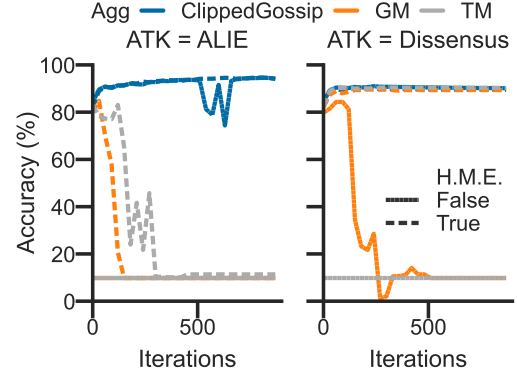


Fig. B.6 Accuracy of aggregators with or without the honest majority everywhere (H.M.E.) assumption. Regular workers are connected through a ring and have IID data.

randomly put at most two updates into one bucket and average within each bucket and then apply robust aggregators to the averaged updates.

Setup for “Effects of the number of Byzantine workers”

The Fig. 3.6 uses a dumbbell topology variant in Fig. B.5. The experiments run for 1500 iterations. In this experiment we choose $n - b = 11$ and $b = 0, 1, 2, 3$. We choose the edge weight of Byzantine workers such that the $\widetilde{\mathbf{W}}$ and p remain the same for all these b . Then we can easily investigate the relation between $\delta_{\max} \in [0, \frac{b}{b+3}]$ and p by varying b . The hyperparameter of dissensus attack is set to $\epsilon_i = 1.5$ for all workers and all experiments.

Setup for “Defense without honest majority”

The Fig. B.6 uses the ring topology of 5 regular workers in Fig. B.7. 11 Byzantine workers are added to the ring so that 1 regular worker do not have honest majority. The experiments run for 900 iterations. We use $\epsilon_i = 1.5$ for dissensus attacks. We use clipping radius $\tau = 0.1$ for CLIPPEDGOSSIP.

In the decentralized environment, the common *honest majority* assumption in the federated learning setup can be strengthened to *honest majority everywhere*, meaning all regular workers have an honest majority of neighbors [Su and Vaidya, 2016b; Yang and Bajwa, 2019a,b]. Considering a ring of 5 regular workers with IID data, and adding 2 Byzantine workers to each node will still satisfy the honest majority assumption everywhere. Now adding one more Byzantine worker to a node will break the assumption.

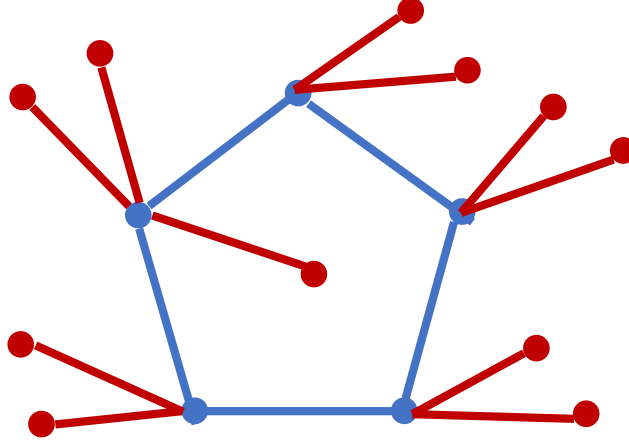


Fig. B.7 Ring topology without honest majority.

Figure B.6 shows that while TM and GM can sometimes counter the attack under the honest majority assumption, adding one more Byzantine worker always corrupts the entire training. The CLIPPEDGOSSIP defend attacks successfully even beyond the assumption, because they leverage the fact that local updates are trustworthy. This suggest that existing statistics-based aggregators which take no advantage of local information are vulnerable under this realistic decentralized threat model.

Setup for “More topologies and attacks.”

In Figure 3.5, we use the small-world and torus topologies described in § B.3.1. More specifically, we created a randomized small-world topology using NetworkX package [Hagberg et al., 2008a] with 10 regular workers each connected to 2 nearest neighbors and probability of rewiring each edge as 0.15. Two additional Byzantine workers are linked to 2 random regular workers. There are 12 workers in total. For the torus topology, we let regular workers form a torus grid $T_{3,3}$ where all 9 regular workers are connected to 3 other workers. Two additional Byzantine workers are linked to 2 random regular workers. There are 11 workers in total.

The mixing matrix for these topologies are constructed with Metropolis-Hastings algorithm in § B.3.2. The spectral gap for small-world topology and torus topology are 0.084 and 0.131 respectively. In contrast, the dumbbell topology in Figure B.10 is more challenging with a spectral gap of 0.043. The data distribution is non-IID.

Table B.3 Default experimental settings for CIFAR-10

Dataset	CIFAR-10
Architecture	VGG-11[Simonyan and Zisserman, 2015]
Training objective	Cross entropy loss
Evaluation objective	Top-1 accuracy
Batch size per worker	64
Momentum	0.9
Learning rate	0.1
LR decay	0.1 at epoch 80 and 120
LR warmup	No
Weight decay	No
Repetitions	1 ¹
Reported metric	Mean test accuracy over the last 150 iterations

B.4.3 Experiment: CIFAR-10 task

In this section, we conduct experiments on CIFAR-10 dataset Krizhevsky [2012]. The running environment of this experiment is the same as MNIST experiment Table B.1. The default setup for CIFAR-10 experiment is summarized in Table B.3.

We compare performances of 5 aggregators on dumbbell topology with 10 nodes in each clique (no attackers). The results of experiments are shown in Figure B.8. In order to investigate if consensus has reached among the workers, we average the worker nodes in 3 different categories (“Global”, Clique A, and Clique B) and compare their performances on IID and NonIID datasets. The “IID-Global” result show that GM and TM is much worse than CLIPPEDGOSSIP and Gossip, in contrast to the MNIST experiment Figure 3.4 where they have matching result. This is because the workers with in each clique are converging to different stationary point — “IID-Clique A” and “IID-Clique B” show GM and TM in each clique can reach over 80% accuracy which is close to Gossip. It demonstrates that GM and TM fail to reach consensus even in this Byzantine-free case and therefore vulnerable to attacks.

The NonIID experiment also support that CLIPPEDGOSSIP perform much better than all other robust aggregators. Notice that CLIPPEDGOSSIP’s “NonIID-Global” performance is better than “NonIID-Clique A” and “NonIID-Clique B” while GM and TM’s result are opposite. This is because CLIPPEDGOSSIP allows effective communication in this topology and therefore clique models are close to each other in the same local minima basin such that their average (global model) is better than both of them. The GM’s and TM’s clique models converge to different local minima, making their averaged model underperform.

B.4.4 Experiment for “Weaker topology assumption”

As is mentioned in Remark 1 and § B.3.1, the topology assumption in this work is weaker than the robust network assumption in Su and Vaidya [2016a]; Sundaram and Gharesifard [2018].

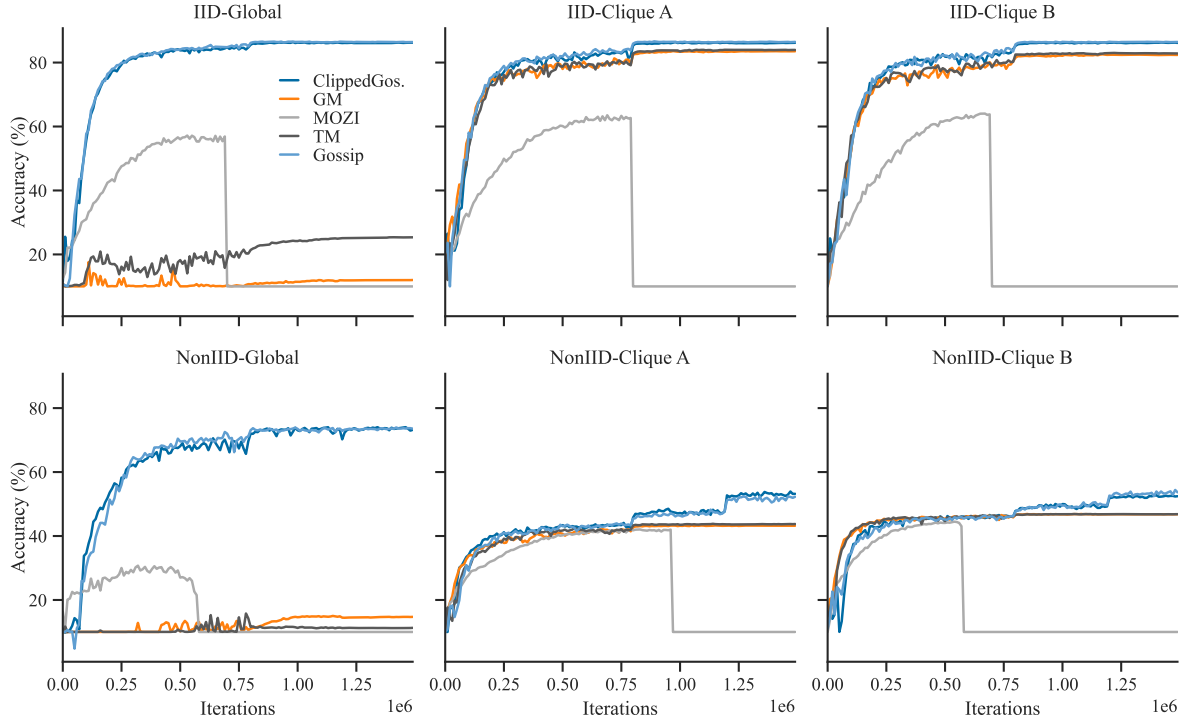


Fig. B.8 Train models on dumbbell topology with IID and NonIID datasets. The three figures in each row correspond to the same experiment with “Global”, “Clique A”, “Clique B” denoting the performances of globally averaged model, within-Clique A averaged model, and within-Clique B averaged model.

We use the topology in Figure B.2 which consists of 10 regular workers and 2 dissensus attack workers. While this topology does not satisfy the robust network assumption, it intuitively should allow communication between two cliques as no Byzantine workers are attached to the cut. However, both GM and TM will discard the graph cut due to data heterogeneity. This shows that GM and TM impede information diffusion. On the other hand, CLIPPEDGOSSIP is the only robust aggregator which help two cliques reaching consensus in the NonIID case. The CLIPPEDGOSSIP theoretically applies to more topologies and empirically perform better.

B.4.5 Experiment: choosing clipping radius

In Figure B.10 we show the sensitive of tuning clipping radius. We use dumbbell topology with 5 regular workers in each clique and add 1 more Byzantine worker to each clique. The clipping radius is searched over a grid of $[0.1, 0.5, 1, 2, 10]$. The Byzantine workers are chosen to be Bit-Flipping, Label-Flipping, and ALIE.

We also give an adaptive clipping strategy for different $i \in \mathcal{V}_R$ and time t . After communication step at time t , the value of $\mathbf{x}_i^{t+1/2}$ is available. Therefore we can sort the values of $\|\mathbf{x}_i^{t+1/2} - \mathbf{x}_j^{t+1/2}\|_2^2$ for all $j \in \mathcal{N}_i$. We denote the set of indices set \mathcal{S}_i^t as the indices of workers

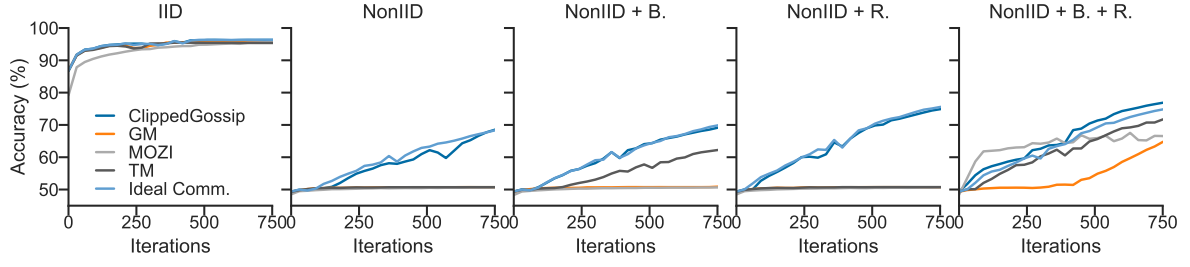


Fig. B.9 Compare robust aggregators under dissensus attacks over dumbbell topology Figure 3.5.

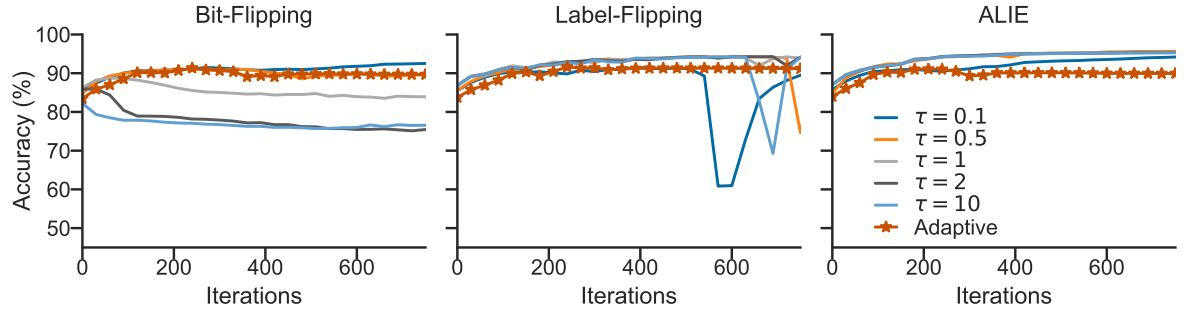


Fig. B.10 Tuning clipping radius on the dumbbell topology against Byzantine attacks. The y-axis is the averaged test accuracy over all of the regular workers.

that have the smallest distances to worker i

$$\mathcal{S}_i^t = \arg \min_{\mathcal{S}: \sum_{j \in \mathcal{S}} \mathbf{W}_{ij} \leq 1 - \delta_{\max}} \sum_{j \in \mathcal{S}} \left\| \mathbf{x}_i^{t+1/2} - \mathbf{x}_j^{t+1/2} \right\|_2^2.$$

Then the adaptive strategy picks clipping radius as follows

$$\tau_i^{t+1} = \sqrt{\sum_{j \in \mathcal{S}_i^t} \mathbf{W}_{ij} \left\| \mathbf{x}_i^{t+1/2} - \mathbf{x}_j^{t+1/2} \right\|_2^2}. \quad (\text{B.4})$$

Note that this adaptive choice of clipping radius is generally a bit smaller than the theoretical value (B.21). It guarantees that the Byzantine workers have limited influences at cost of small slow down on the convergence.

As we can see from Figure B.10, the performances of CLIPPEDGOSSIP are similar with different constant choices of τ which shows that the choice of τ is not very sensitive. The adaptive algorithms perform well in all cases. Therefore, the adaptive choice of τ will be recommended in general.

B.5 Analysis

We restate the core equations in Algorithm 3 at time t on worker i as follows

$$\mathbf{m}_i^{t+1} = (1 - \alpha)\mathbf{m}_i^t + \alpha\mathbf{g}_i(\mathbf{x}_i^t) \quad (\text{B.5})$$

$$\mathbf{x}_i^{t+1/2} = \mathbf{x}_i^t - \eta\mathbf{m}_i^{t+1} \quad (\text{B.6})$$

$$\mathbf{z}_{j \rightarrow i}^{t+1} = \mathbf{x}_i^{t+1/2} + \text{CLIP}(\mathbf{x}_j^{t+1/2} - \mathbf{x}_i^{t+1/2}, \tau_i^t) \quad (\text{B.7})$$

$$\mathbf{x}_i^{t+1} = \sum_{j=1}^n \mathbf{W}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} \quad (\text{B.8})$$

In addition, we define the following virtual iterates on the set of good nodes \mathcal{V}_R

- $\bar{\mathbf{x}}^t = \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbf{x}_i^t$ the average (over time) of good iterates.
- $\bar{\mathbf{m}}^t = \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbf{m}_i^t$ the average (over time) of momentum iterates.

In this proof, we define $p := 1 - (1 - \gamma)^2 \in (0, 1]$ for convenience.

In this section, we show that the convergence behavior of the virtual iterates $\bar{\mathbf{x}}^t$. The structure of this section is as follows:

- In § B.5.1, we give common quantities, simplified notations and list common equalities/inequalities used in the proof.
- In § B.5.2, we provide all auxiliary lemmas necessary for the proof. Among these lemmas, Lemma B.3 is the key sufficient descent lemma.
- In § B.5.3, we provide the proof of the main theorem.

B.5.1 Definitions, and inequalities

Notations for the proof. We use the following variables to simplify the notation

- Optimization sub-optimality:

$$r^t := f(\bar{\mathbf{x}}^t) - f^*$$

- Consensus distance:

$$\Xi^t := \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \|\mathbf{x}_i^t - \bar{\mathbf{x}}^t\|_2^2$$

- The distance between the ideal gradient and actual averaged momentum

$$e_1^{t+1} := \mathbb{E} \|\nabla f(\bar{\mathbf{x}}^t) - \bar{\mathbf{m}}^{t+1}\|_2^2$$

- Similarly, the distance between the ideal gradient and individual momentums

$$\tilde{e}_1^{t+1} := \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \|\nabla f(\bar{\mathbf{x}}^t) - \mathbf{m}_i^{t+1}\|_2^2$$

- Similar, distance between individual ideal gradients and individual momentums which is weighted by the mixing matrix

$$\bar{e}_1^{t+1} := \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij} (\nabla f_j(\bar{\mathbf{x}}^t) - \mathbf{m}_j^{t+1}) \right\|_2^2$$

- Similar we have distance between individual ideal gradients and individual momentums

$$e_I^{t+1} := \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \|\mathbf{m}_i^{t+1} - \nabla f_i(\bar{\mathbf{x}}^t)\|_2^2,$$

- Let e_2^{t+1} be the averaged squared error introduced by clipping and Byzantine workers

$$e_2^{t+1} := \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j \in \mathcal{V}_R} \mathbf{W}_{ij} (\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_j^{t+1/2}) + \sum_{j \in \mathcal{V}_B} \mathbf{W}_{ij} (\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_i^{t+1/2}) \right\|_2^2.$$

Lemma B.1 (Common equalities and inequalities). *We use the following equalities and inequalities*

- *The cosine theorem:* $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$

$$\langle \mathbf{x}, \mathbf{y} \rangle = -\frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 + \frac{1}{2} \|\mathbf{x}\|_2^2 + \frac{1}{2} \|\mathbf{y}\|_2^2 \quad (\text{B.9})$$

- *Young's inequality:* For $\epsilon > 0$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$

$$\|\mathbf{x} + \mathbf{y}\|_2^2 \leq (1 + \epsilon) \|\mathbf{x}\|_2^2 + (1 + \epsilon^{-1}) \|\mathbf{y}\|_2^2 \quad (\text{B.10})$$

- *If f is convex, then for $\alpha \in [0, 1]$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$*

$$f(\alpha \mathbf{x} + (1 - \alpha) \mathbf{y}) \leq \alpha f(\mathbf{x}) + (1 - \alpha) f(\mathbf{y}) \quad (\text{B.11})$$

- *Cauchy-Schwarz inequality*

$$\langle \mathbf{x}, \mathbf{y} \rangle \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \quad (\text{B.12})$$

- *Let $\{\mathbf{x}_i : i \in [m]\}$ be independent random variables and $\mathbb{E} \mathbf{x}_i = \mathbf{0}$ and $\mathbb{E} \|\mathbf{x}_i\|^2 = \sigma^2$ then*

$$\mathbb{E} \left\| \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i \right\|_2^2 = \frac{\sigma^2}{m} \quad (\text{B.13})$$

B.5.2 Lemmas

The following lemma establish the update rule for $\bar{\mathbf{x}}^t$.

Lemma B.2. *Assume Lemma 3.3. Let Δ^{t+1} be the error incurred by clipping and \mathcal{V}_B*

$$\Delta^{t+1} := \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left(\sum_{j \in \mathcal{V}_R} \mathbf{w}_{ij} (\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_j^{t+1/2}) + \sum_{j \in \mathcal{V}_B} \mathbf{w}_{ij} (\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_i^{t+1/2}) \right). \quad (\text{B.14})$$

Then the virtual iterate updates

$$\bar{\mathbf{x}}^{t+1} = \bar{\mathbf{x}}^t - \eta \bar{\mathbf{m}}^{t+1} + \Delta^{t+1}. \quad (\text{B.15})$$

Proof. Expand $\bar{\mathbf{x}}^{t+1}$ with the definition of \mathbf{x}_i^{t+1} in (B.8) yields

$$\begin{aligned} \bar{\mathbf{x}}^{t+1} &= \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbf{x}_i^{t+1} = \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left(\sum_{j \in \mathcal{V}_R} \mathbf{w}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} + \sum_{j \in \mathcal{V}_B} \mathbf{w}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} \right) \\ &= \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left(\sum_{j \in \mathcal{V}_R} \mathbf{w}_{ij} (\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_j^{t+1/2}) + \sum_{j \in \mathcal{V}_R} \mathbf{w}_{ij} \mathbf{x}_j^{t+1/2} \right) \\ &\quad + \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left(\sum_{j \in \mathcal{V}_B} \mathbf{w}_{ij} (\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_i^{t+1/2}) + \sum_{j \in \mathcal{V}_B} \mathbf{w}_{ij} \mathbf{x}_i^{t+1/2} \right). \end{aligned}$$

Reorganize the terms to form Δ^{t+1}

$$\begin{aligned} \bar{\mathbf{x}}^{t+1} &= \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left(\sum_{j \in \mathcal{V}_R} \mathbf{w}_{ij} \mathbf{x}_j^{t+1/2} + \sum_{j \in \mathcal{V}_B} \mathbf{w}_{ij} \mathbf{x}_i^{t+1/2} \right) + \Delta^{t+1} \\ &= \frac{1}{|\mathcal{V}_R|} \sum_{j \in \mathcal{V}_R} (1 - \delta_j) \mathbf{x}_j^{t+1/2} + \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \delta_i \mathbf{x}_i^{t+1/2} + \Delta^{t+1} \\ &= \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbf{x}_i^{t+1/2} + \Delta^{t+1} = \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} (\mathbf{x}_i^t - \eta \mathbf{m}_i^{t+1}) + \Delta^{t+1} \\ &= \bar{\mathbf{x}}_i^t - \eta \bar{\mathbf{m}}^{t+1} + \Delta^{t+1}. \end{aligned} \quad \square$$

Note that the Δ^{t+1} can be written as the follows

$$\Delta^{t+1} = \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left(\mathbf{x}_i^{t+1} - \sum_{j \in \mathcal{V}_R} \tilde{\mathbf{w}}_{ij} \mathbf{x}_j^{t+1/2} \right) = \bar{\mathbf{x}}^{t+1} - \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbf{x}_i^{t+1/2}.$$

where measures the error introduced to $\bar{\mathbf{x}}^{t+1}$ considering the impact of Byzantine workers and clipping. Therefore when $\mathcal{V}_B = \emptyset$ and τ is sufficiently large, $\Delta^{t+1} = 0$ and $\bar{\mathbf{x}}^{t+1}$ converge at the same rate as the centralized SGD with momentum.

Recall that $e_1^{t+1} := \mathbb{E}\|\nabla f(\bar{\mathbf{x}}^t) - \bar{\mathbf{m}}^{t+1}\|_2^2$. The key descent lemma is stated as follow

Lemma B.3 (Sufficient decrease). *Assume Assumption D and $\eta \leq \frac{1}{2L}$, then*

$$\mathbb{E} f(\bar{\mathbf{x}}^{t+1}) \leq f(\bar{\mathbf{x}}^t) - \frac{\eta}{2} \|\nabla f(\bar{\mathbf{x}}^t)\|_2^2 - \frac{\eta}{4} \mathbb{E} \|\bar{\mathbf{m}}^{t+1} - \frac{1}{\eta} \Delta^{t+1}\|_2^2 + \eta e_1^{t+1} + \frac{1}{\eta} e_2^{t+1}.$$

Proof. Use smoothness Assumption D and expand it with (B.15)

$$f(\bar{\mathbf{x}}^{t+1}) \leq f(\bar{\mathbf{x}}^t) - \langle \nabla f(\bar{\mathbf{x}}^t), \eta \bar{\mathbf{m}}^{t+1} - \Delta^{t+1} \rangle + \frac{L}{2} \|\eta \bar{\mathbf{m}}^{t+1} - \Delta^{t+1}\|_2^2$$

Apply cosine theorem (B.9) to the inner product $\eta \langle \nabla f(\bar{\mathbf{x}}^t), \bar{\mathbf{m}}^{t+1} - \frac{1}{\eta} \Delta^{t+1} \rangle$ yields

$$\begin{aligned} \mathbb{E} f(\bar{\mathbf{x}}^{t+1}) &\leq f(\bar{\mathbf{x}}^t) - \frac{\eta}{2} \|\nabla f(\bar{\mathbf{x}}^t)\|_2^2 - \left(\frac{\eta - L\eta^2}{2} \right) \mathbb{E} \|\bar{\mathbf{m}}^{t+1} - \frac{1}{\eta} \Delta^{t+1}\|_2^2 \\ &\quad + \frac{\eta}{2} \mathbb{E} \|\nabla f(\bar{\mathbf{x}}^t) - \bar{\mathbf{m}}^{t+1} + \frac{1}{\eta} \Delta^{t+1}\|_2^2. \end{aligned}$$

If step size $\eta \leq \frac{1}{2L}$, then $-\frac{\eta - L\eta^2}{2} \leq -\frac{\eta}{4}$. Applying inequality (B.10) to the last term

$$\frac{\eta}{2} \mathbb{E} \|\nabla f(\bar{\mathbf{x}}^t) - \bar{\mathbf{m}}^{t+1} + \frac{1}{\eta} \Delta^{t+1}\|_2^2 \leq \eta \mathbb{E} \|\nabla f(\bar{\mathbf{x}}^t) - \bar{\mathbf{m}}^{t+1}\|_2^2 + \frac{1}{\eta} \mathbb{E} \|\Delta^{t+1}\|_2^2.$$

Since $e_1^{t+1} := \mathbb{E} \|\nabla f(\bar{\mathbf{x}}^t) - \bar{\mathbf{m}}^{t+1}\|_2^2$ and $\mathbb{E} \|\Delta^{t+1}\|_2^2 \leq e_2^{t+1}$, then we have

$$\mathbb{E} f(\bar{\mathbf{x}}^{t+1}) \leq f(\bar{\mathbf{x}}^t) - \frac{\eta}{2} \|\nabla f(\bar{\mathbf{x}}^t)\|_2^2 - \frac{\eta}{4} \mathbb{E} \|\bar{\mathbf{m}}^{t+1} - \frac{1}{\eta} \Delta^{t+1}\|_2^2 + \eta e_1^{t+1} + \frac{1}{\eta} e_2^{t+1}. \quad \square$$

In the next lemma, we establish the recursion for the distance between momentums and gradients

Lemma B.4. *Assume Assumptions C and D and lemma 3.3, For any doubly stochastic mixing matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$*

$$e_A^{t+1} = \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j \in \mathcal{V}_R} \mathbf{A}_{ij} (\mathbf{m}_j^{t+1} - \nabla f_j(\bar{\mathbf{x}}^t)) \right\|_2^2,$$

then we have the following recursion

$$e_A^{t+1} \leq (1 - \alpha) e_A^t + \frac{\alpha^2 \sigma^2}{|\mathcal{V}_R|} \|\mathbf{A}\|_{F, \mathcal{V}_R}^2 + 2\alpha L^2 \Xi^t + \frac{2L^2 \eta^2}{\alpha} \|\bar{\mathbf{m}}^t - \frac{1}{\eta} \Delta^t\|_2^2. \quad (\text{B.16})$$

where we define $\|\mathbf{A}\|_{F, \mathcal{V}_R}^2 := \sum_{i \in \mathcal{V}_R} \sum_{j \in \mathcal{V}_R} \mathbf{A}_{ij}^2$ Therefore,

- If $\mathbf{A}_{ij} = \frac{1}{|\mathcal{V}_R|}$ for all $i, j \in \mathcal{V}_R$, then $e_A^{t+1} = e_1^{t+1}$ and $\|\mathbf{A}\|_{F, \mathcal{V}_R}^2 = 1$.

- If $\mathbf{A} = \widetilde{\mathbf{W}}$, then $e_A^{t+1} = \bar{e}_1^{t+1}$ and $\|\mathbf{A}\|_{F, \mathcal{V}_R}^2 = \sum_{i \in \mathcal{V}_R} \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij}^2 \leq |\mathcal{V}_R|$.
- If $\mathbf{A} = \mathbf{I}$, then $\|\mathbf{A}\|_{F, \mathcal{V}_R}^2 = |\mathcal{V}_R|$. In addition,

$$\bar{e}_1^{t+1} \leq 2e_I^{t+1} + 2\zeta^2$$

where $\mathbf{A} = \mathbf{I}$.

Proof. We can expand e_A^{t+1} by expanding \mathbf{m}_j^{t+1}

$$\begin{aligned} e_A^{t+1} &\stackrel{\text{(B.5)}}{=} \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j \in \mathcal{V}_R} \mathbf{A}_{ij} ((1-\alpha)\mathbf{m}_j^t + \alpha \mathbf{g}_j(\mathbf{x}_j^t) - \nabla f_j(\bar{\mathbf{x}}^t)) \right\|_2^2 \\ &= \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j \in \mathcal{V}_R} \mathbf{A}_{ij} ((1-\alpha)\mathbf{m}_j^t + \alpha(\mathbf{g}_j(\mathbf{x}_j^t) \pm \nabla f_j(\mathbf{x}_j^t)) - \nabla f_j(\bar{\mathbf{x}}^t)) \right\|_2^2 \end{aligned}$$

Extract the stochastic term $\mathbf{g}_j(\mathbf{x}_j^t) - \nabla f_j(\mathbf{x}_j^t)$ inside the norm and use that $\mathbb{E} \mathbf{g}_j(\mathbf{x}_j^t) = \nabla f_j(\mathbf{x}_j^t)$,

$$\begin{aligned} e_A^{t+1} &= \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left\| \sum_{j \in \mathcal{V}_R} \mathbf{A}_{ij} ((1-\alpha)\mathbf{m}_j^t + \alpha \nabla f_j(\mathbf{x}_j^t) - \nabla f_j(\bar{\mathbf{x}}^t)) \right\|_2^2 \\ &\quad + \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j \in \mathcal{V}_R} \mathbf{A}_{ij} \alpha (\mathbf{g}_j(\mathbf{x}_j^t) - \nabla f_j(\mathbf{x}_j^t)) \right\|_2^2 \\ &\leq \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left\| \sum_{j \in \mathcal{V}_R} \mathbf{A}_{ij} ((1-\alpha)\mathbf{m}_j^t + \alpha \nabla f_j(\mathbf{x}_j^t) - \nabla f_j(\bar{\mathbf{x}}^t)) \right\|_2^2 \\ &\quad + \frac{\alpha^2}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \sum_{j \in \mathcal{V}_R} \mathbf{A}_{ij}^2 \mathbb{E} \|\mathbf{g}_j(\mathbf{x}_j^t) - \nabla f_j(\mathbf{x}_j^t)\|_2^2. \end{aligned}$$

Then we can use Assumption C for the last term to get

$$e_A^{t+1} = \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left\| \sum_{j \in \mathcal{V}_R} \mathbf{A}_{ij} ((1-\alpha)\mathbf{m}_j^t + \alpha \nabla f_j(\mathbf{x}_j^t) - \nabla f_j(\bar{\mathbf{x}}^t)) \right\|_2^2 + \frac{\alpha^2 \sigma^2}{|\mathcal{V}_R|} \|\mathbf{A}\|_{F, \mathcal{V}_R}^2.$$

Then we insert $\pm(1-\alpha)\nabla f_j(\bar{\mathbf{x}}^{t-1})$ inside the first norm and expand using (B.11)

$$\begin{aligned} e_A^{t+1} &\leq \frac{1-\alpha}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left\| \sum_{j \in \mathcal{V}_R} \mathbf{A}_{ij} (\mathbf{m}_j^t - \nabla f_j(\bar{\mathbf{x}}^{t-1})) \right\|_2^2 + \frac{\alpha^2 \sigma^2}{|\mathcal{V}_R|} \|\mathbf{A}\|_{F, \mathcal{V}_R}^2 \\ &\quad + \frac{\alpha}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left\| \sum_{j \in \mathcal{V}_R} \mathbf{A}_{ij} (\nabla f_j(\mathbf{x}_j^t) - \nabla f_j(\bar{\mathbf{x}}^t) + \frac{1-\alpha}{\alpha} (\nabla f_j(\bar{\mathbf{x}}^{t-1}) - \nabla f_j(\bar{\mathbf{x}}^t))) \right\|_2^2. \end{aligned}$$

Note that the first term is e_A^t and by the convexity of $\|\cdot\|$ for the last term we have

$$\begin{aligned} e_A^{t+1} &\leq (1-\alpha)e_A^t + \frac{\alpha^2\sigma^2}{|\mathcal{V}_R|} \|\mathbf{A}\|_{F,\mathcal{V}_R}^2 \\ &\quad + \frac{\alpha}{|\mathcal{V}_R|} \sum_{j \in \mathcal{V}_R} \|\nabla f_j(\mathbf{x}_j^t) - \nabla f_j(\bar{\mathbf{x}}^t) + \frac{1-\alpha}{\alpha}(\nabla f_j(\bar{\mathbf{x}}^{t-1}) - \nabla f_j(\bar{\mathbf{x}}^t))\|_2^2. \end{aligned}$$

Then we can further expand the last term

$$\begin{aligned} e_A^{t+1} &\leq (1-\alpha)e_A^t + \frac{\alpha^2\sigma^2}{|\mathcal{V}_R|} \|\mathbf{A}\|_{F,\mathcal{V}_R}^2 \\ &\quad + \frac{2\alpha}{|\mathcal{V}_R|} \sum_{j \in \mathcal{V}_R} \|\nabla f_j(\mathbf{x}_j^t) - \nabla f_j(\bar{\mathbf{x}}^t)\|_2^2 + \frac{2(1-\alpha)^2}{\alpha|\mathcal{V}_R|} \sum_{j \in \mathcal{V}_R} \|\nabla f_j(\bar{\mathbf{x}}^{t-1}) - \nabla f_j(\bar{\mathbf{x}}^t)\|_2^2. \end{aligned}$$

Then we can apply smoothness Assumption D and use $(1-\alpha)^2 \leq 1$

$$e_A^{t+1} \leq (1-\alpha)e_A^t + \frac{\alpha^2\sigma^2}{|\mathcal{V}_R|} \|\mathbf{A}\|_{F,\mathcal{V}_R}^2 + 2\alpha L^2 \Xi^t + \frac{2L^2\eta^2}{\alpha} \|\bar{\mathbf{m}}^t - \frac{1}{\eta} \Delta^t\|_2^2.$$

Besides, consider \tilde{e}_1^{t+1}

$$\begin{aligned} \tilde{e}_1^{t+1} &= \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \|\mathbf{m}_i^{t+1} - \nabla f(\bar{\mathbf{x}}^t)\|_2^2 = \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \|\mathbf{m}_i^{t+1} \pm \nabla f_i(\bar{\mathbf{x}}^t) - \nabla f(\bar{\mathbf{x}}^t)\|_2^2 \\ &\leq 2 \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \|\mathbf{m}_i^{t+1} - \nabla f_i(\bar{\mathbf{x}}^t)\|_2^2 + 2 \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \|\nabla f_i(\bar{\mathbf{x}}^t) - \nabla f(\bar{\mathbf{x}}^t)\|_2^2 \\ &= 2e_I^{t+1} + 2\zeta^2. \end{aligned}$$

□

As we know that $\|\Delta^{t+1}\|_2^2 \leq e_2^{t+1}$, then we need to finally bound e_2^{t+1}

Lemma B.5 (Bound on e_2^{t+1}). *For $\delta_{\max} := \max_{i \in \mathcal{V}_R} \delta_i$, if*

$$\tau_i^{t+1} = \sqrt{\frac{1}{\delta_i} \sum_{j \in \mathcal{V}_R} \mathbf{W}_{ij} \mathbb{E} \left\| \mathbf{x}_i^{t+1/2} - \mathbf{x}_j^{t+1/2} \right\|_2^2},$$

then we have

$$e_2^{t+1} \leq c_1 \delta_{\max} (2\eta^2 (e_I^{t+1} + \zeta^2) + \Xi^t).$$

where constant $c_1 = 32$.

Proof. Use Young's inequality (B.10) to bound e_2^{t+1} by two parts

$$\begin{aligned} e_2^{t+1} &= \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j \in \mathcal{V}_R} \mathbf{w}_{ij} (\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_j^{t+1/2}) + \sum_{j \in \mathcal{V}_B} \mathbf{w}_{ij} (\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_i^{t+1/2}) \right\|_2^2 \\ &\leq \underbrace{\frac{2}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j \in \mathcal{V}_R} \mathbf{w}_{ij} (\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_j^{t+1/2}) \right\|_2^2}_{=: A_1} + \underbrace{\frac{2}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j \in \mathcal{V}_B} \mathbf{w}_{ij} (\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_i^{t+1/2}) \right\|_2^2}_{=: A_2}. \end{aligned}$$

Look at the first term use triangular inequality of $\|\cdot\|$ and the definition of τ_i^{t+1}

$$\begin{aligned} A_1 &\leq \frac{2}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left(\sum_{j \in \mathcal{V}_R} \mathbf{w}_{ij} \left\| \mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_j^{t+1/2} \right\|_2 \right)^2 \\ &\leq \frac{2}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left(\frac{1}{\tau_i^{t+1}} \sum_{j \in \mathcal{V}_R} \mathbf{w}_{ij} \left\| \mathbf{x}_i^{t+1/2} - \mathbf{x}_j^{t+1/2} \right\|_2^2 \right)^2. \end{aligned}$$

The second inequality holds true because we can consider two cases of $\mathbf{z}_{j \rightarrow i}^{t+1}$ for all $j \in \mathcal{V}_R$

- If $\|\mathbf{x}_i^{t+1/2} - \mathbf{x}_j^{t+1/2}\|_2^2 \leq \tau_i^{t+1}$, then CLIP has no effect and therefore $\mathbf{z}_{j \rightarrow i}^{t+1} = \mathbf{x}_j^{t+1/2}$

$$0 = \|\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_j^{t+1/2}\|_2 \leq \frac{1}{\tau_i^{t+1}} \|\mathbf{x}_i^{t+1/2} - \mathbf{x}_j^{t+1/2}\|_2^2.$$

- If $\|\mathbf{x}_i^{t+1/2} - \mathbf{x}_j^{t+1/2}\|_2^2 > \tau_i^{t+1}$, then $\mathbf{z}_{j \rightarrow i}^{t+1}$ sits between $\mathbf{x}_j^{t+1/2}$ and $\mathbf{x}_i^{t+1/2}$ with

$$\|\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_j^{t+1/2}\|_2 + \tau_i^{t+1} = \|\mathbf{x}_i^{t+1/2} - \mathbf{x}_j^{t+1/2}\|_2.$$

Therefore, using the inequality $a - \tau \leq \frac{a^2}{\tau}$ for $a > 0$ we have that

$$\|\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_j^{t+1/2}\|_2 = \|\mathbf{x}_i^{t+1/2} - \mathbf{x}_j^{t+1/2}\|_2 - \tau_i^{t+1} \leq \frac{1}{\tau_i^{t+1}} \|\mathbf{x}_i^{t+1/2} - \mathbf{x}_j^{t+1/2}\|_2^2.$$

Therefore we justify the second inequality.

On the other hand,

$$\begin{aligned} A_2 &\leq \frac{2}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left(\sum_{j \in \mathcal{V}_B} \mathbf{w}_{ij} \left\| \mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_i^{t+1/2} \right\|_2 \right)^2 \leq \frac{2}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left(\sum_{j \in \mathcal{V}_B} \mathbf{w}_{ij} (\tau_i^{t+1}) \right)^2 \\ &= \frac{2}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \delta_i^2 (\tau_i^{t+1})^2. \end{aligned}$$

Then minimizing the RHS of e_2^{t+1} by tuning radius for clipping

$$\tau_i^{t+1} = \sqrt{\mathbb{E} \left(\frac{1}{\delta_i} \sum_{j \in \mathcal{V}_R} \mathbf{W}_{ij} \left\| \mathbf{x}_i^{t+1/2} - \mathbf{x}_j^{t+1/2} \right\|_2^2 \right)^2}$$

Then we come to the following bound

$$e_2^{t+1} \leq \frac{4}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \delta_i \sum_{j \in \mathcal{V}_R} \mathbf{W}_{ij} \left\| \mathbf{x}_i^{t+1/2} - \mathbf{x}_j^{t+1/2} \right\|_2^2.$$

Then we expand the norm as follows

$$\begin{aligned} \mathbb{E} \left\| \mathbf{x}_i^{t+1/2} - \mathbf{x}_j^{t+1/2} \right\|_2^2 &= \mathbb{E} \left\| \mathbf{x}_i^t - \eta \mathbf{m}_i^{t+1} - \mathbf{x}_j^t + \eta \mathbf{m}_j^{t+1} \right\|_2^2 \\ &= \mathbb{E} \left\| \mathbf{x}_i^t \pm \bar{\mathbf{x}}^t - \mathbf{x}_j^t + \eta \mathbf{m}_j^{t+1} \pm \eta \nabla f(\bar{\mathbf{x}}^t) - \eta \mathbf{m}_i^{t+1} \right\|_2^2 \\ &\leq 4\eta^2 \mathbb{E} \left\| \mathbf{m}_i^{t+1} - \nabla f(\bar{\mathbf{x}}^t) \right\|_2^2 + 4\eta^2 \mathbb{E} \left\| \mathbf{m}_j^{t+1} - \nabla f(\bar{\mathbf{x}}^t) \right\|_2^2 \\ &\quad + 4 \left\| \mathbf{x}_i^t - \bar{\mathbf{x}}^t \right\|_2^2 + 4 \left\| \mathbf{x}_j^t - \bar{\mathbf{x}}^t \right\|_2^2 \end{aligned} \tag{B.17}$$

Use the fact that $\sum_{j \in \mathcal{V}_R} \mathbf{W}_{ij} = 1 - \delta_i$ we have

$$\begin{aligned} e_2^{t+1} &\leq \frac{16\eta^2}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \delta_i (1 - \delta_i) \mathbb{E} \left\| \mathbf{m}_i^{t+1} - \nabla f(\bar{\mathbf{x}}^t) \right\|_2^2 + \frac{16\eta^2}{|\mathcal{V}_R|} \sum_{j \in \mathcal{V}_R} \sum_{i \in \mathcal{V}_R} \delta_i \mathbf{W}_{ij} \mathbb{E} \left\| \mathbf{m}_j^{t+1} - \nabla f(\bar{\mathbf{x}}^t) \right\|_2^2 \\ &\quad + \frac{16}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \delta_i (1 - \delta_i) \left\| \mathbf{x}_i^t - \bar{\mathbf{x}}^t \right\|_2^2 + \frac{16}{|\mathcal{V}_R|} \sum_{j \in \mathcal{V}_R} \sum_{i \in \mathcal{V}_R} \delta_i \mathbf{W}_{ij} \left\| \mathbf{x}_j^t - \bar{\mathbf{x}}^t \right\|_2^2 \end{aligned}$$

Use the fact that $\delta_i \leq \delta_{\max}$ and $1 - \delta_i \leq 1$ for all $i \in \mathcal{V}_R$,

$$e_2^{t+1} \leq 32\delta_{\max}(2\eta^2(e_I^{t+1} + \zeta^2) + \Xi^t).$$

□

Theorem 3.1'. Let $\bar{\mathbf{x}} := \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbf{x}_i$ be the average iterate over the unknown set of regular nodes with

$$\tau_i = \sqrt{\frac{1}{\delta_i} \sum_{j \in \mathcal{V}_R} \mathbf{W}_{ij} \mathbb{E} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_2^2}. \tag{B.18}$$

If the initial consensus distance is bounded as $\frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \mathbf{x}_i - \bar{\mathbf{x}} \right\|^2 \leq \rho^2$, then for all $i \in \mathcal{V}_R$, the output $\hat{\mathbf{x}}_i$ of CLIPPEDGOSSIP satisfies

$$\frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \hat{\mathbf{x}}_i - \bar{\mathbf{x}} \right\|^2 \leq (1 - \gamma + c\sqrt{\delta_{\max}})^2 \rho^2$$

where the expectation is over the random variable $\{\mathbf{x}_i\}_{i \in \mathcal{V}_R}$ and $c > 0$ is a constant.

Proof. We can consider the 1-step consensus problem as 1-step of optimization problem with $\rho^2 = \Xi^t$ and $\eta = 0$. Then we look for the upper bound of $\frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \|\mathbf{x}_i^{t+1} - \bar{\mathbf{x}}^t\|_2^2$ in terms of ρ^2 , p , and δ_{\max} .

$$\begin{aligned} \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \|\mathbf{x}_i^{t+1} - \bar{\mathbf{x}}^t\|_2^2 &= \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j=1}^n \mathbf{w}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \bar{\mathbf{x}}^t \right\|_2^2 \\ &= \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \left(\sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{w}}_{ij} \mathbf{x}_j^t - \bar{\mathbf{x}}^t \right) + \left(\sum_{j=1}^n \mathbf{w}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{w}}_{ij} \mathbf{x}_j^t \right) \right\|_2^2. \end{aligned}$$

Apply (B.10) with $\epsilon > 0$ and use the expected improvement Lemma 3.4

$$\begin{aligned} &\frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \|\mathbf{x}_i^{t+1} - \bar{\mathbf{x}}^t\|_2^2 \\ &\leq \frac{1+\epsilon}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left\| \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{w}}_{ij} \mathbf{x}_j^t - \bar{\mathbf{x}}^t \right\|_2^2 + \frac{1+\frac{1}{\epsilon}}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j=1}^n \mathbf{w}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{w}}_{ij} \mathbf{x}_j^t \right\|_2^2 \\ &\leq \frac{(1+\epsilon)(1-p)}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \|\mathbf{x}_i^t - \bar{\mathbf{x}}^t\|_2^2 + \frac{1+\frac{1}{\epsilon}}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j=1}^n \mathbf{w}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{w}}_{ij} \mathbf{x}_j^t \right\|_2^2 \\ &\leq (1+\epsilon)(1-p)\Xi^t + \frac{1+\frac{1}{\epsilon}}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j=1}^n \mathbf{w}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{w}}_{ij} \mathbf{x}_j^t \right\|_2^2 \end{aligned}$$

Replace $\mathbf{x}_j^t = \mathbf{x}_j^{t+1/2} + \eta \mathbf{m}_j^{t+1}$ using (B.6), then apply (B.12) and $\eta = 0$

$$\frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \|\mathbf{x}_i^{t+1} - \bar{\mathbf{x}}^t\|_2^2 \leq (1+\epsilon)(1-p)\Xi^t + \frac{1+\frac{1}{\epsilon}}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j=1}^n \mathbf{w}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{w}}_{ij} \mathbf{x}_j^{t+1/2} \right\|_2^2.$$

Recall the definition of e_2^{t+1}

$$e_2^{t+1} := \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j=1}^n \mathbf{w}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{w}}_{ij} \mathbf{x}_j^{t+1/2} \right\|_2^2.$$

Then use Lemma B.4 with the case $\mathbf{A} = \widetilde{\mathbf{W}}$ and apply Lemma B.5 with $\eta = 0$

$$\begin{aligned} \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \|\mathbf{x}_i^{t+1} - \bar{\mathbf{x}}^t\|_2^2 &\leq (1+\epsilon)(1-p)\Xi^t + \left(1 + \frac{1}{\epsilon}\right) e_2^{t+1} \\ &\leq (1+\epsilon)(1-p)\Xi^t + \left(1 + \frac{1}{\epsilon}\right) 32\delta_{\max} \Xi^t. \end{aligned}$$

Let's minimize the right hand side of the above inequality by taking ϵ such that $\epsilon(1-p) = \frac{32\delta_{\max}}{\epsilon}$ which leads to $\epsilon = \sqrt{\frac{32\delta_{\max}}{1-p}}$, then the above inequality becomes

$$\begin{aligned} \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \|\mathbf{x}_i^{t+1} - \bar{\mathbf{x}}^t\|_2^2 &\leq (1-p + 32\delta_{\max} + 2\sqrt{32\delta_{\max}(1-p)})\Xi^t \\ &= (\sqrt{1-p} + \sqrt{32\delta_{\max}})^2 \Xi^t. \end{aligned}$$

The consensus distance to the average consensus is only guaranteed to reduce if $\sqrt{1-p} + \sqrt{32\delta_{\max}} < 1$ which is

$$\delta_{\max} < \frac{1}{32}(1 - \sqrt{1-p})^2.$$

Finally, we complete the proof by simplifying the notation to spectral gap $\gamma := 1 - \sqrt{1-p}$. \square

Recall that

$$e_2^{t+1} := \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left\| \sum_{j \in \mathcal{V}_R} \mathbf{W}_{ij}(\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_j^{t+1/2}) + \sum_{j \in \mathcal{V}_B} \mathbf{W}_{ij}(\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_i^{t+1/2}) \right\|_2^2. \quad (\text{B.19})$$

Next we consider the bound on consensus distance Ξ^t .

Lemma B.6 (Bound consensus distance Ξ^t). *Assume Lemma 3.4, then Ξ^t has the following iteration*

$$\begin{aligned} \Xi^{t+1} &\leq (1+\epsilon)(1-p)\Xi^t \\ &\quad + c_2(1 + \frac{1}{\epsilon}) \left(e_2^{t+1} + \eta^2 \bar{e}_1^{t+1} + \eta^2 \zeta^2 + \eta^2 \|\nabla f(\bar{\mathbf{x}}^t)\|_2^2 + \eta^2 \mathbb{E} \|\bar{\mathbf{m}}^{t+1} - \frac{1}{\eta} \Delta^{t+1}\|_2^2 \right). \end{aligned}$$

where $\epsilon > 0$ is determined later such that $(1+\epsilon)(1-p) < 1$ and $c_2 = 5$.

Proof. Expand the consensus distance at time $t+1$

$$\begin{aligned} \Xi^{t+1} &= \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \|\mathbf{x}_i^{t+1} - \bar{\mathbf{x}}^{t+1}\|_2^2 = \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j=1}^n \mathbf{W}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \bar{\mathbf{x}}^{t+1} \right\|_2^2 \\ &= \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j=1}^n \mathbf{W}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \bar{\mathbf{x}}^t + \bar{\mathbf{x}}^t - \bar{\mathbf{x}}^{t+1} \right\|_2^2 \\ &= \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \left(\sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij} \mathbf{x}_j^t - \bar{\mathbf{x}}^t \right) + \left(\sum_{j=1}^n \mathbf{W}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij} \mathbf{x}_j^t \right) + \bar{\mathbf{x}}^t - \bar{\mathbf{x}}^{t+1} \right\|_2^2. \end{aligned}$$

Apply Young's inequality (B.10) with coefficient ϵ , like the proof of Theorem 3.1, and use the expected improvement Lemma 3.4

$$\begin{aligned}
\Xi^{t+1} &\leq \frac{1+\epsilon}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left\| \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij} \mathbf{x}_j^t - \bar{\mathbf{x}}^t \right\|_2^2 \\
&\quad + \frac{1+\epsilon}{\epsilon |\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j=1}^n \mathbf{W}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij} \mathbf{x}_j^t + \bar{\mathbf{x}}^t - \bar{\mathbf{x}}^{t+1} \right\|_2^2 \\
&\leq \frac{(1+\epsilon)(1-p)}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left\| \mathbf{x}_i^t - \bar{\mathbf{x}}^t \right\|_2^2 \\
&\quad + \frac{1+\epsilon}{\epsilon |\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j=1}^n \mathbf{W}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij} \mathbf{x}_j^t + \bar{\mathbf{x}}^t - \bar{\mathbf{x}}^{t+1} \right\|_2^2 \\
&\leq (1+\epsilon)(1-p) \Xi^t + \underbrace{\frac{1+\epsilon}{\epsilon |\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \left(\sum_{j=1}^n \mathbf{W}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij} \mathbf{x}_j^t \right) + \bar{\mathbf{x}}^t - \bar{\mathbf{x}}^{t+1} \right\|_2^2}_{=: T_1}
\end{aligned}$$

Replace $\mathbf{x}_j^t = \mathbf{x}_j^{t+1/2} + \eta \mathbf{m}_j^{t+1}$ using (B.6), then apply (B.12)

$$\begin{aligned}
T_1 &= \frac{1+\epsilon}{\epsilon |\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j=1}^n \mathbf{W}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij} \mathbf{x}_j^{t+1/2} - \eta \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij} \mathbf{m}_j^{t+1} + \bar{\mathbf{x}}^t - \bar{\mathbf{x}}^{t+1} \right\|_2^2 \\
&\leq 5 \frac{1+\epsilon}{\epsilon} \left(\frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j=1}^n \mathbf{W}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij} \mathbf{x}_j^{t+1/2} \right\|_2^2 \right. \\
&\quad \left. + \frac{\eta^2}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij} (\mathbf{m}_j^{t+1} - \nabla f_j(\bar{\mathbf{x}}^t)) \right\|_2^2 \right. \\
&\quad \left. + \frac{\eta^2}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left\| \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij} \nabla f_j(\bar{\mathbf{x}}^t) - \nabla f(\bar{\mathbf{x}}^t) \right\|_2^2 + \eta^2 \left\| \nabla f(\bar{\mathbf{x}}^t) \right\|_2^2 + \mathbb{E} \left\| \bar{\mathbf{x}}^t - \bar{\mathbf{x}}^{t+1} \right\|_2^2 \right). \tag{B.20}
\end{aligned}$$

Recall the definition of e_2^{t+1}

$$\begin{aligned}
e_2^{t+1} &:= \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j \in \mathcal{V}_R} \mathbf{W}_{ij} (\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_j^{t+1/2}) + \sum_{j \in \mathcal{V}_B} \mathbf{W}_{ij} (\mathbf{z}_{j \rightarrow i}^{t+1} - \mathbf{x}_i^{t+1/2}) \right\|_2^2 \\
&= \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \mathbb{E} \left\| \sum_{j=1}^n \mathbf{W}_{ij} \mathbf{z}_{j \rightarrow i}^{t+1} - \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij} \mathbf{x}_j^{t+1/2} \right\|_2^2
\end{aligned}$$

Then use Lemma B.4 with the case $\mathbf{A} = \widetilde{\mathbf{W}}$,

$$T_1 \leq 5(1 + \frac{1}{\epsilon}) \left(e_2^{t+1} + \eta^2 \bar{e}_1^{t+1} + \frac{\eta^2}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \left\| \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij} \nabla f_j(\bar{\mathbf{x}}^t) - \nabla f(\bar{\mathbf{x}}^t) \right\|_2^2 + \eta^2 \|\nabla f(\bar{\mathbf{x}}^t)\|_2^2 + \mathbb{E} \|\bar{\mathbf{x}}^t - \bar{\mathbf{x}}^{t+1}\|_2^2 \right).$$

Use convexity of $\|\cdot\|_2^2$ and Assumption C we have

$$T_1 \leq 5(1 + \frac{1}{\epsilon}) (e_2^{t+1} + \eta^2 \bar{e}_1^{t+1} + \eta^2 \zeta^2 + \eta^2 \|\nabla f(\bar{\mathbf{x}}^t)\|_2^2 + \mathbb{E} \|\bar{\mathbf{x}}^t - \bar{\mathbf{x}}^{t+1}\|_2^2).$$

Use (B.15) for the last term

$$T_1 \leq 5(1 + \frac{1}{\epsilon}) \left(e_2^{t+1} + \eta^2 \bar{e}_1^{t+1} + \eta^2 \zeta^2 + \eta^2 \|\nabla f(\bar{\mathbf{x}}^t)\|_2^2 + \eta^2 \mathbb{E} \|\bar{\mathbf{m}}^{t+1} - \frac{1}{\eta} \Delta^{t+1}\|_2^2 \right).$$

Finally, by the definition of \bar{e}_1^{t+1} , we have

$$\Xi^{t+1} \leq (1 + \epsilon)(1 - p)\Xi^t + 5(1 + \frac{1}{\epsilon}) \left(e_2^{t+1} + \eta^2 \bar{e}_1^{t+1} + \eta^2 \zeta^2 + \eta^2 \|\nabla f(\bar{\mathbf{x}}^t)\|_2^2 + \eta^2 \mathbb{E} \|\bar{\mathbf{m}}^{t+1} - \frac{1}{\eta} \Delta^{t+1}\|_2^2 \right).$$

□

Lemma B.7 (Tuning stepsize.). *Suppose the following holds for any step size $\eta \leq d$:*

$$\Psi_T \leq \frac{r_0}{\eta(T+1)} + b\eta + e\eta^2 + f\eta^3.$$

Then, there exists a step-size $\eta \leq d$ such that

$$\Psi_T \leq 2\left(\frac{br_0}{T+1}\right)^{\frac{1}{2}} + 2e^{\frac{1}{3}}\left(\frac{r_0}{T+1}\right)^{\frac{2}{3}} + 2f^{\frac{1}{4}}\left(\frac{r_0}{T+1}\right)^{\frac{3}{4}} + \frac{dr_0}{T+1}.$$

Proof. Choosing $\eta = \min \left\{ \left(\frac{r_0}{b(T+1)} \right)^{\frac{1}{2}}, \left(\frac{r_0}{e(T+1)} \right)^{\frac{1}{3}}, \left(\frac{r_0}{f(T+1)} \right)^{\frac{1}{4}}, \frac{1}{d} \right\} \leq \frac{1}{d}$ we have four cases

- $\eta = \frac{1}{d}$ and is smaller than $\left(\frac{r_0}{b(T+1)} \right)^{\frac{1}{2}}, \left(\frac{r_0}{e(T+1)} \right)^{\frac{1}{3}}, \left(\frac{r_0}{f(T+1)} \right)^{\frac{1}{4}}$, then

$$\Psi_T \leq \frac{dr_0}{T+1} + \frac{b}{d} + \frac{e}{d^2} + \frac{f}{d^3} \leq \frac{dr_0}{T+1} + \left(\frac{br_0}{T+1} \right)^{\frac{1}{2}} + e^{1/3} \left(\frac{r_0}{T+1} \right)^{\frac{2}{3}} + f^{1/4} \left(\frac{r_0}{T+1} \right)^{\frac{3}{4}}.$$

- $\eta = \left(\frac{r_0}{b(T+1)} \right)^{\frac{1}{2}} < \min \left\{ \left(\frac{r_0}{e(T+1)} \right)^{\frac{1}{3}}, \left(\frac{r_0}{f(T+1)} \right)^{\frac{1}{4}} \right\}$, then

$$\Psi_T \leq 2 \left(\frac{br_0}{T+1} \right)^{\frac{1}{2}} + \frac{er_0}{b(T+1)} + f \left(\frac{r_0}{b(T+1)} \right)^{\frac{3}{2}} \leq 2 \left(\frac{br_0}{bT+1} \right)^{\frac{1}{2}} + e^{1/3} \left(\frac{r_0}{T+1} \right)^{\frac{2}{3}} + f^{1/4} \left(\frac{r_0}{T+1} \right)^{\frac{3}{4}}.$$

- $\eta = \left(\frac{r_0}{e(T+1)}\right)^{\frac{1}{3}} < \min\left\{\left(\frac{r_0}{b(T+1)}\right)^{\frac{1}{2}}, \left(\frac{r_0}{f(T+1)}\right)^{\frac{1}{4}}\right\}$, then

$$\Psi_T \leq 2e^{1/3} \left(\frac{r_0}{T+1}\right)^{\frac{2}{3}} + b \left(\frac{r_0}{e(T+1)}\right)^{\frac{1}{3}} + \frac{fr_0}{e(T+1)} \leq \left(\frac{br_0}{T+1}\right)^{\frac{1}{2}} + 2e^{1/3} \left(\frac{r_0}{T+1}\right)^{\frac{2}{3}} + f^{1/4} \left(\frac{r_0}{T+1}\right)^{\frac{3}{4}}.$$

- $\eta = \left(\frac{r_0}{f(T+1)}\right)^{\frac{1}{4}} < \min\left\{\left(\frac{r_0}{b(T+1)}\right)^{\frac{1}{2}}, \left(\frac{r_0}{e(T+1)}\right)^{\frac{1}{3}}\right\}$, then

$$\Psi_T \leq 2f^{1/4} \left(\frac{r_0}{T+1}\right)^{\frac{3}{4}} + b \left(\frac{r_0}{f(T+1)}\right)^{\frac{1}{4}} + e \left(\frac{r_0}{f(T+1)}\right)^{\frac{1}{2}} \leq \left(\frac{br_0}{T+1}\right)^{\frac{1}{2}} + e^{1/3} \left(\frac{r_0}{T+1}\right)^{\frac{2}{3}} + 2f^{1/4} \left(\frac{r_0}{T+1}\right)^{\frac{3}{4}}.$$

Then, take the uniform upper bound of the upper bound gives the result. \square

B.5.3 Proof of the main theorem

Theorem 3.3'. Suppose Assumptions A-3.4 hold and $\delta_{\max} = \mathcal{O}(\gamma^2)$. Define the clipping radius as

$$\tau_i^{t+1} = \sqrt{\frac{1}{\delta_i} \sum_{j \in \mathcal{V}_R} \mathbf{W}_{ij} \mathbb{E} \left\| \mathbf{x}_i^{t+1/2} - \mathbf{x}_j^{t+1/2} \right\|_2^2}. \quad (\text{B.21})$$

Then for $\alpha := 3\eta L$, the iterates of Algorithm 3 satisfy

$$\begin{aligned} \frac{1}{T+1} \sum_{t=0}^T \|\nabla f(\bar{\mathbf{x}}^t)\|_2^2 &\leq \frac{200c_1c_2}{\gamma^2} \delta_{\max} \zeta^2 + 2 \left(\frac{3^2}{|\mathcal{V}_R|} + \frac{320c_1c_2}{\gamma^2} \delta_{\max} \right)^{1/2} \left(\frac{3L\sigma^2 r_0}{T+1} \right)^{1/2} \\ &\quad + 2 \left(\frac{48c_2}{\gamma^2} \zeta^2 \right)^{1/3} \left(\frac{r_0 L}{T+1} \right)^{2/3} + 2 \left(\frac{144c_2}{\gamma^2} \sigma^2 \right)^{1/4} \left(\frac{r_0 L}{T+1} \right)^{3/4} + \frac{d_0 r_0}{T+1}. \end{aligned}$$

where $r_0 := f(\mathbf{x}^0) - f^*$ and $c_1 = 32$ and $c_2 = 5$. Furthermore, the consensus distance has an upper bound

$$\frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \|\mathbf{x}_i^t - \bar{\mathbf{x}}^t\|_2^2 = \mathcal{O}\left(\frac{\zeta^2}{\gamma^2(T+1)}\right).$$

Remark 8. The requirement $\delta_{\max} = \mathcal{O}(\gamma^2)$ suggest that δ_{\max} and γ^2 are of same order. The exact constant are determined in the proof and can be tighten simply through better constants in equalities like (B.17), (B.20). In practice CLIPPEDGOSSIP allow high number of attackers. For example in Figure B.9, 1/6 of workers are Byzantine and CLIPPEDGOSSIP still perform well in the non-IID setting.

Proof. Denote the terms of average t from 0 to T as follows

$$\begin{aligned} C_1 &:= \frac{1}{1+T} \sum_{t=0}^T \|\nabla f(\bar{\mathbf{x}}^t)\|_2^2, C_2 := \frac{1}{1+T} \sum_{t=0}^T \left\| \bar{\mathbf{m}}^{t+1} - \frac{1}{\eta} \Delta^{t+1} \right\|_2^2, D_1 := \frac{1}{1+T} \sum_{t=0}^T \Xi^{t+1} \\ E_1 &:= \frac{1}{1+T} \sum_{t=0}^T e_1^{t+1}, \bar{E}_1 := \frac{1}{1+T} \sum_{t=0}^T \bar{e}_1^{t+1}, E_I := \frac{1}{1+T} \sum_{t=0}^T e_I^{t+1}, E_2 := \frac{1}{1+T} \sum_{t=0}^T e_2^{t+1} \end{aligned}$$

First we apply average to Lemma B.5

$$E_2 \leq c_2 \delta_{\max}(2\eta^2(E_I + \zeta^2) + D_1). \quad (\text{B.22})$$

Then we rewrite key Lemma B.3 as

$$\|\nabla f(\bar{\mathbf{x}}^t)\|_2^2 + \frac{1}{2} \mathbb{E} \|\bar{\mathbf{m}}^{t+1} - \frac{1}{\eta} \Delta^{t+1}\|_2^2 \leq \frac{2}{\eta}(r^t - r^{t+1}) + 2e_1^{t+1} + \frac{2}{\eta^2} e_2^{t+1},$$

and further average over time t

$$C_1 + \frac{1}{2} C_2 \leq \frac{2r_0}{\eta(T+1)} + 2E_1 + \frac{2}{\eta^2} E_2$$

where we use $-f(\mathbf{x}^{T+1}) \leq -f^*$. Combined with (B.22) gives

$$C_1 + \frac{1}{2} C_2 \leq \frac{2r_0}{\eta(T+1)} + 2E_1 + 4c_2 \delta_{\max} E_I + 4c_2 \delta_{\max} \zeta^2 + \frac{2c_2 \delta_{\max}}{\eta^2} D_1 \quad (\text{B.23})$$

Now we also average Lemma B.4 for e_1^{t+1} over t gives

$$\begin{aligned} \frac{1}{1+T} \sum_{t=0}^T e_1^{t+1} &\leq \frac{1-\alpha}{1+T} \sum_{t=0}^T e_1^t + 2\alpha L^2 D_1 + \frac{\alpha^2 \sigma^2}{|\mathcal{V}_R|} + \frac{2L^2 \eta^2}{\alpha} \frac{1}{1+T} \sum_{t=0}^T \|\bar{\mathbf{m}}^t - \frac{1}{\eta} \Delta^t\|_2^2 \\ &\leq \frac{1-\alpha}{1+T} \sum_{t=0}^T e_1^{t+1} + 2\alpha L^2 D_1 + \frac{\alpha^2 \sigma^2}{|\mathcal{V}_R|} + \frac{2L^2 \eta^2}{\alpha} C_2 \end{aligned}$$

where we use $\Xi^0 = e_1^0 = 0$ and $\bar{\mathbf{m}}^0 = \Delta^0 = \mathbf{0}$. Then let $\beta_1 := \frac{2L^2 \eta^2}{\alpha^2}$

$$E_1 \leq 2L^2 D_1 + \frac{\alpha \sigma^2}{|\mathcal{V}_R|} + \beta_1 C_2. \quad (\text{B.24})$$

Similarly, Lemma B.4 for e_I^{t+1} the only difference is that we don't have $\frac{1}{n}$ for σ^2

$$E_I \leq 2L^2 D_1 + \alpha \sigma^2 + \beta_1 C_2. \quad (\text{B.25})$$

Similarly, let's call $\beta_2 := \frac{1}{|\mathcal{V}_R|} \sum_{i \in \mathcal{V}_R} \sum_{j \in \mathcal{V}_R} \widetilde{\mathbf{W}}_{ij}^2 \leq 1$

$$\bar{E}_1 \leq 2L^2 D_1 + \beta_2 \alpha \sigma^2 + \beta_1 C_2. \quad (\text{B.26})$$

The consensus distance Lemma B.6 has

$$\begin{aligned} D_1 &\leq \frac{(1+\epsilon)(1-p)}{1+T} \sum_{t=0}^T \Xi^t + c_2(1+\frac{1}{\epsilon})E_2 + c_2(1+\frac{1}{\epsilon})\eta^2(\bar{E}_1^{t+1} + \zeta^2 + C_1 + C_2) \\ &\leq (1+\epsilon)(1-p)D_1 + c_2(1+\frac{1}{\epsilon})E_2 + c_2(1+\frac{1}{\epsilon})\eta^2(\bar{E}_1^{t+1} + \zeta^2 + C_1 + C_2). \end{aligned}$$

Replace E_2 using (B.22) gives

$$\begin{aligned} D_1 &\leq (1+\epsilon)(1-p)D_1 + c_2(1+\frac{1}{\epsilon})(c_1\delta_{\max}(2\eta^2(E_I^{t+1} + \zeta^2) + D_1)) + c_2(1+\frac{1}{\epsilon})\eta^2(\bar{E}_1^{t+1} + \zeta^2 + C_1 + C_2) \\ &\leq ((1+\epsilon)(1-p) + c_1c_2(1+\frac{1}{\epsilon})\delta_{\max})D_1 + c_2(1+\frac{1}{\epsilon})\eta^2(2c_1\delta_{\max}E_I^{t+1} + \bar{E}_1^{t+1} + (1+2c_1\delta_{\max})\zeta^2 + C_1 + C_2). \end{aligned}$$

Now replace \bar{E}_1 , E_I with (B.26), (B.25), then

$$\begin{aligned} D_1 &\leq ((1+\epsilon)(1-p) + c_2(1+\frac{1}{\epsilon})(c_1\delta_{\max}(1+4L^2\eta^2) + 2L^2\eta^2))D_1 \\ &\quad + c_2(1+\frac{1}{\epsilon})\eta^2((2c_1\delta_{\max} + \beta_2)\alpha\sigma^2 + (2c_1\delta_{\max} + 1)\zeta^2 + ((2c_1\delta_{\max} + 1)\beta_1 + 1)C_2 + C_1). \end{aligned}$$

By enforcing $\eta \leq \frac{\gamma}{9L}$ and $\delta_{\max} \leq \frac{\gamma^2}{10c_1c_2}$ we have

$$\begin{aligned} 2c_2L^2\eta^2 &\leq \gamma^2/8 \\ c_1c_2\delta_{\max}(1+4L^2\eta^2) &\leq \gamma^2/8 \end{aligned}$$

we can achieve

$$\sqrt{c_1c_2\delta_{\max}(1+4L^2\eta^2) + 2c_2L^2\eta^2} \leq \frac{\gamma}{2}.$$

Then

$$\begin{aligned} D_1 &\leq \underbrace{((1+\epsilon)(1-p) + (1+\frac{1}{\epsilon})\frac{\gamma^2}{4})}_{=:T_2} D_1 \\ &\quad + c_2(1+\frac{1}{\epsilon})\eta^2((2c_1\delta_{\max} + \beta_2)\alpha\sigma^2 + (2c_1\delta_{\max} + 1)\zeta^2 + ((2c_1\delta_{\max} + 1)\beta_1 + 1)C_2 + C_1). \end{aligned}$$

Let us minimize the the coefficients of D_1 on the right hand side of inequality by having

$$\epsilon(1-p) = \frac{1}{\epsilon} \frac{\gamma^2}{4},$$

that is $\epsilon = \sqrt{\frac{\gamma^2}{4(1-p)}}$. Then the coefficient becomes

$$\begin{aligned} T_2 &= (1 + \epsilon)(1 - p) + (1 + \frac{1}{\epsilon})\frac{\gamma^2}{4} \\ &= (\sqrt{1 - p} + \frac{\gamma}{2})^2 \\ &= (1 - \frac{\gamma}{2})^2. \end{aligned}$$

Then we use $\frac{1}{\epsilon} = \sqrt{\frac{4(1-p)}{\gamma^2}} \leq \frac{2}{\gamma}$ and $1 + \frac{1}{\epsilon} \leq \frac{3}{\gamma}$

$$D_1 \leq \frac{4c_2\eta^2}{\gamma^2}((2c_1\delta_{\max} + \beta_2)\alpha\sigma^2 + (2c_1\delta_{\max} + 1)\zeta^2 + ((2c_1\delta_{\max} + 1)\beta_1 + 1)C_2 + C_1).$$

This leads to $2c_1\delta_{\max} \leq \frac{\gamma^2}{5c_2} \leq 1$ and $\beta_2 \leq 1$, then we know

$$D_1 \leq \frac{4c_2\eta^2}{\gamma^2}(2\alpha\sigma^2 + 2\zeta^2 + C_1 + (1 + 2\beta_1)C_2) \quad (\text{B.27})$$

Finally, we combine (B.23), (B.24), (B.26)

$$\begin{aligned} C_1 + \frac{1}{2}C_2 &\leq \frac{2r_0}{\eta(T+1)} + 2E_1 + 4c_1\delta_{\max}E_I + 4c_1\delta_{\max}\zeta^2 + \frac{2c_1\delta_{\max}}{\eta^2}D_1 \\ &\leq \frac{2r_0}{\eta(T+1)} + (4L^2D_1 + \frac{2\alpha\sigma^2}{|\mathcal{V}_R|} + 2\beta_1C_2) + 2c_1\delta_{\max}(4L^2D_1 + 2\beta_2\alpha\sigma^2 + 2\beta_1C_2) \\ &\quad + 4c_1\delta_{\max}\zeta^2 + \frac{2c_1\delta_{\max}}{\eta^2}D_1 \\ &\leq \frac{2r_0}{\eta(T+1)} + (4L^2 + 8c_1\delta_{\max}L^2 + \frac{2c_1\delta_{\max}}{\eta^2})D_1 + (\frac{1}{|\mathcal{V}_R|} + 2c_1\delta_{\max})2\alpha\sigma^2 \\ &\quad + 4\beta_1C_2 + 4c_1\delta_{\max}\zeta^2 \end{aligned}$$

Then we replace D_1 with (B.27)

$$\begin{aligned} C_1 + \frac{1}{2}C_2 &\leq \frac{2r_0}{\eta(T+1)} + (\frac{1}{|\mathcal{V}_R|} + 2c_1\delta_{\max})2\alpha\sigma^2 + 4\beta_1C_2 + 4c_1\delta_{\max}\zeta^2 \\ &\quad + (4L^2\eta^2 + 8c_1\delta_{\max}L^2\eta^2 + 2c_1\delta_{\max})\frac{4c_2}{\gamma^2}(2\alpha\sigma^2 + 2\zeta^2 + C_1 + (1 + 2\beta_1)C_2) \end{aligned} \quad (\text{B.28})$$

To have a valid bound on C_1 , there are two constraints on the coefficient of the RHS C_1 and C_2 .

$$\begin{aligned} (4L^2\eta^2 + 8c_1\delta_{\max}L^2\eta^2 + 2c_1\delta_{\max})\frac{4c_2}{\gamma^2} &< 1 \\ (4L^2\eta^2 + 8c_1\delta_{\max}L^2\eta^2 + 2c_1\delta_{\max})\frac{4c_2}{\gamma^2}(1 + 2\beta_1) + 4\beta_1 &\leq \frac{1}{2}. \end{aligned}$$

We can strength the first requirement to

$$(4L^2\eta^2 + 8c_1\delta_{\max}L^2\eta^2 + 2c_1\delta_{\max})\frac{4c_2}{\gamma^2} \leq \frac{1}{4}. \quad (\text{B.29})$$

Then, apply this inequality to the second inequality gives

$$\frac{1}{4} + \frac{1}{2}\beta_1 + 4\beta_1 \leq \frac{1}{2}$$

which requires $\eta \leq \frac{\alpha}{3L}$. Next (B.29) can be achieved by requiring $\delta_{\max} \leq \frac{\gamma^2}{64c_1c_2}$

$$(4 + 8c_1\delta_{\max})L^2\eta^2 + 2c_1\delta_{\max} \leq 8L^2\eta^2 + 2c_1\delta_{\max} \leq \frac{\gamma^2}{16c_2}$$

which requires $8\eta^2L^2 \leq \frac{\gamma^2}{32c_2}$, and we can simplify it to $\eta \leq \frac{\gamma}{40L}$. Now we can simplify (B.28) with (B.29)

$$\begin{aligned} \frac{3}{4}C_1 &\leq \frac{2r_0}{\eta(T+1)} + \left(\frac{1}{|\mathcal{V}_R|} + 2c_1\delta_{\max}\right)2\alpha\sigma^2 + 4c_1\delta_{\max}\zeta^2 \\ &\quad + (4L^2\eta^2 + 8c_1\delta_{\max}L^2\eta^2 + 2c_1\delta_{\max})\frac{4c_2}{\gamma^2}(2\alpha\sigma^2 + 2\zeta^2) \end{aligned}$$

Multiply both sides with $\frac{4}{3}$ and relax constant $\frac{4}{3} \cdot 2 \leq 3$. Then by taking $\eta \leq \frac{1}{2L}$ we have that

$$C_1 \leq \frac{3r_0}{\eta(T+1)} + \left(\frac{1}{|\mathcal{V}_R|} + \frac{151}{\gamma^2}2c_1\delta_{\max}\right)3\alpha\sigma^2 + \frac{200c_1c_2}{\gamma^2}\delta_{\max}\zeta^2 + \frac{48c_2}{\gamma^2}(\alpha\sigma^2 + \zeta^2)L^2\eta^2$$

By taking $\alpha := 3\eta L$ and relax the constants we have

$$C_1 \leq \frac{3r_0}{\eta(T+1)} + \left(\frac{3^2}{|\mathcal{V}_R|} + \frac{320c_1}{\gamma^2}\delta_{\max}\right)L\sigma^2\eta + \frac{48c_2}{\gamma^2}(\alpha\sigma^2 + \zeta^2)L^2\eta^2 + \frac{200c_1c_2}{\gamma^2}\delta_{\max}\zeta^2.$$

Minimize the the right hand side by tuning step size Lemma B.7 we have

$$\begin{aligned} \frac{1}{T+1} \sum_{t=0}^T \|\nabla f(\bar{\mathbf{x}}^t)\|_2^2 &\leq \frac{200c_1c_2}{\gamma^2}\delta_{\max}\zeta^2 + 2 \left(\frac{\left(\frac{3^2}{|\mathcal{V}_R|} + \frac{320c_1}{\gamma^2}\delta_{\max}\right)3L\sigma^2r_0}{T+1} \right)^{\frac{1}{2}} \\ &\quad + 2 \left(\frac{48c_2}{\gamma^2}\zeta^2 \right)^{\frac{1}{3}} \left(\frac{r_0L}{T+1} \right)^{\frac{2}{3}} + 2 \left(\frac{144c_2}{\gamma^2}\sigma^2 \right)^{\frac{1}{4}} \left(\frac{r_0L}{T+1} \right)^{\frac{3}{4}} + \frac{d_0r_0}{T+1} \end{aligned}$$

where $\frac{1}{d_0} := \min\{\frac{1}{2L}, \frac{\gamma}{9L}, \frac{\gamma}{40L}\} = \frac{\gamma}{40L}$ and

$$\eta = \min \left\{ \left(\frac{2r_0}{\left(\frac{9}{|\mathcal{V}_R|} + \frac{320c_1}{\gamma^2}\delta_{\max}\right)L\sigma^2(T+1)} \right)^{1/2}, \left(\frac{2r_0\gamma^2}{48c_2\zeta^2L^2(T+1)} \right)^{1/3}, \left(\frac{2r_0\gamma^2}{L^3\sigma^2(T+1)} \right)^{1/4}, \frac{1}{d_0} \right\}.$$

Bound on the consensus distance D_1 . Since $\beta_1 = \frac{2L^2\eta^2}{\alpha^2} = \frac{2}{9}$, we can relax (B.27) to

$$\begin{aligned} D_1 &\leq \frac{4c_2\eta^2}{\gamma^2}(2\alpha\sigma^2 + 2\zeta^2 + 2(1 + 2\beta_1)(C_1 + \tfrac{1}{2}C_2)) \\ &\leq \frac{4c_2\eta^2}{\gamma^2}(2\alpha\sigma^2 + 2\zeta^2 + 3(C_1 + \tfrac{1}{2}C_2)). \end{aligned}$$

For significantly large T , we know that $\eta = \alpha = \mathcal{O}(\frac{1}{\sqrt{T+1}})$ and find the upper bound of $2\alpha\sigma^2 + 2\zeta^2 + C_1 + \frac{1}{2}C_2$ with $\mathcal{O}(\zeta^2)$ where higher order terms of $1/T$ are dropped. Therefore, the upper bound on the consensus distance D_1 is $\mathcal{O}(\frac{\zeta^2}{\gamma^2(T+1)})$. \square

B.6 Other related works and discussions

In this section, we add more related works and discussions.

Byzantine resilient learning with constraints Byzantine robustness is challenging when the training is combined with other constraints, such as asynchrony [Damaskinos et al., 2018; Xie et al., 2020b; Yang and Li, 2021b], data heterogeneity [Data and Diggavi, 2021a; Karimireddy et al., 2021c; Li et al., 2019; Peng and Ling, 2020], privacy [Burkhalter et al., 2021; He et al., 2020b]. These works all assume the existence of a central server which can communicate with all regular workers. In this paper, we consider the decentralized setting and focus on the constraint that not all regular workers can communicate with each other.

More works on decentralized learning. Many works focus on compression techniques [Koloskova et al., 2019, 2020a; Vogels et al., 2020], data heterogeneity [Koloskova et al., 2021; Tang et al., 2018; Vogels et al., 2021], and communication topology [Assran et al., 2019b; Ying et al., 2021a].

Detailed comparison with one line of work. Among all the works on robust decentralized training, Sundaram et al. Sundaram and Gharesifard [2018] and Su et al. Su and Vaidya [2016a] and their followup works Yang and Bajwa [2019a,b] have the most similar setup with ours. They are all using the trimmed mean as the aggregator assumptions on the graph. We illustrate our advantages over these methods as follows

1. Their methods (TM) make unrealistic assumptions about the graph while our method is much more relaxed. Their main assumption on the graph has 2 parts: 1) each good node should have at least $2b + 1$ neighbors where b is the maximum number of Byzantine workers in the *whole* network; 2) by removing any b edges the good nodes should be connected. This assumption essentially requires the good workers have honest majority *everywhere* and additionally they have to be well connected. This can be hardly enforced in the decentralized environment. In contrast, our method has a weaker condition relating the spectral gap and

- δ . Our method also works without a honest majority Figure B.6. The second part of their assumption exclude common topologies like Dumbbell.
2. TM **fails** to reach consensus even in some **Byzantine-free** graphs (e.g. Dumbbell) while SSClip converges as fast as gossip. For example, TM fails to reach consensus in NonIID setting for MNIST dataset (Figure 3.4) and even fails in IID setting for CIFAR-10 dataset (Figure B.8).
3. We have a clear convergence rate for SGD while they only show asymptotic convergence for GD. In fact, we even improve the state-of-art decentralized SGD analysis [Koloskova et al., 2020b].
4. Our work reveals how the quantitative relation between percentage of Byzantine workers (δ) and information bottleneck (γ) influence the consensus (see Figure 3.3 and Theorem 3.1).
5. We propose a novel dissensus attacks that utilize topology information.
6. Impossibility results. Sundaram et al. Sundaram and Ghahsifard [2018] and Su et al. Su and Vaidya [2016a] give impossibility results in terms of number of nodes while we give a novel results in terms of spectral gap (γ).

Other related works and discussions. Zhao et al. Zhao et al. [2019] make assumption that some users are *trusted* and then adopt trimmed mean as robust aggregator. But this assumption is incompatible with our setting where every node only trusts itself. Peng et al. Peng and Ling [2020] propose a “zero-sum” attack which exploits the topology where Byzantine worker j construct

$$\mathbf{x}_j := -\frac{\sum_{k \in \mathcal{N}_i \cap \mathcal{V}_B} \mathbf{x}_k}{|\mathcal{N}_i \cap \mathcal{V}_B|}.$$

They aim to manipulate the good worker i ’s model to 0, but it also makes the constructed Byzantine model very far away from the good worker models, making it easy to detect. In contrast, our dissensus attack (3.6) simply amplifies the existing disagreement amongst the good workers, which keeps the attack much less undetectable. In addition, we take mixing matrix into consideration and use ϵ_i to parameterize the attack which makes it more flexible.

Clarifications about our method. We make the following clarifications regarding our method:

- Ideally we would like to replace the $\delta_{\max} = \max_j \delta_j$ with an average $\bar{\delta} = \frac{1}{n} \sum_j \delta_j$. However, the requirement that δ_{\max} be small may be achieved by the good workers increasing its weight on itself. Note that Byzantine workers cannot alter good workers local behavior.
- Theorem 3.3 does not tell us what happens if the percentage of Byzantine workers δ is relatively larger than spectral gap (γ), but it does not necessarily mean that CLIPPEDGOSSIP

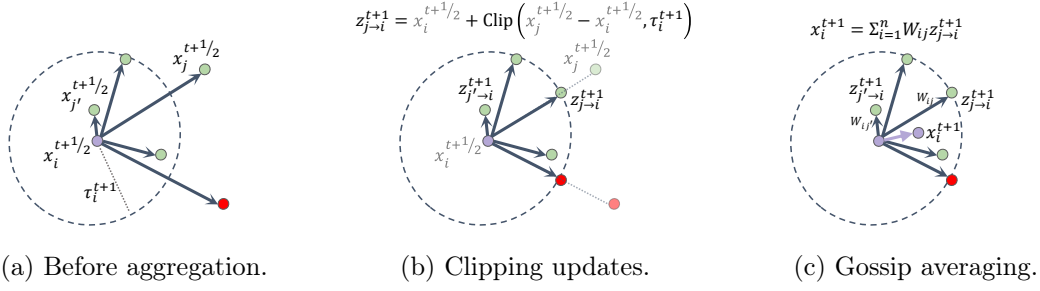


Fig. B.11 Diagram of ClippedGossip at time t on worker i . Let purple node be the model of worker i and green nodes be models of worker i 's regular neighbors and red nodes be models of worker i 's Byzantine neighbors. The figure (a), (b), and (c) demonstrate the 3 stages of ClippedGossip. First, in the left figure (a) worker i collects models $\{\mathbf{x}_j^{t+1/2} : j \in \mathcal{N}_i\}$ from its neighbors. Then in the middle figure (b) worker i clips neighbor models to ensure the clipped models are no farther than τ_i^{t+1} from node i . Nodes outside the circle (e.g. $\mathbf{x}_j^{t+1/2}$) clipped to the circle (e.g. $\mathbf{z}_{j \rightarrow i}^{t+1}$) while nodes inside the circle (e.g. \mathbf{x}_j^{t+1}) remain the same after clipping (e.g. $\mathbf{z}_{j \rightarrow i}^{t+1}$). In the right figure (c) worker i update its model to \mathbf{x}_i^{t+1} using gossip averaging over clipped models.

diverges. Instead, it means reaching global consensus is not possible as Byzantine workers effectively block the information bottleneck. We conjecture that within each connected good component not blocked by the byzantine workers, the good workers still reach component-level consensus by applying the analysis of Theorem 3.3 to only this component. We leave such a component-wise analysis for future work.

Appendix C

Secure Byzantine-Robust Machine Learning

C.1 Proofs

Theorem 4.1 (Privacy for **S1**). *Let $\mathbf{z} = \sum_{i=1}^n p_i \mathbf{x}_i$ where $\{p_i\}_{i=1}^n$ is the output of byzantine oracle or a vector of 1s (non-private). Let $BV_{ij} = \langle \mathbf{a}_{ij}, \mathbf{b}_{ij}, \mathbf{c}_{ij} \rangle$ and $BVp_i = \langle \mathbf{a}_i^p, \mathbf{b}_i^p, \mathbf{c}_i^p \rangle$ be the Beaver's triple used in the multiplications. Let $\langle \cdot \rangle^{(1)}$ be the share of the secret-shared values $\langle \cdot \rangle$ on **S1**. Then for all workers i*

$$\mathbb{P}(\mathbf{x}_i = x_i \mid \{\langle \mathbf{x}_i \rangle^{(1)}, \langle p_i \rangle^{(1)}\}_{i=1}^n, \{BV_{i,j}^{(1)}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_{ij}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{b}_{ij}\}_{i < j}, \{\langle \|\mathbf{x}_i - \mathbf{x}_j\|^2 \rangle^{(1)}\}_{i < j}, \{BVp_i^{(1)}, p_i - \mathbf{a}_i^p, p_i - \mathbf{b}_i^p\}_{i=1}^n, \mathbf{z}) = \mathbb{P}(\mathbf{x}_i = x_i \mid \mathbf{z})$$

Note that the conditioned values are what **S1** observes throughout the algorithm. $\{BV_{ij}^{(1)}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_{ij}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{b}_{ij}\}_{i < j}$ and $\{BVp_i^{(1)}, p_i - \mathbf{a}_i^p, p_i - \mathbf{b}_i^p\}_{i=1}^n$ are intermediate values during shared values multiplication.

Proof. First, we use the independence of Beaver's triple to simplify the conditioned term.

- The Beaver's triples are data-independent. Since $\langle \mathbf{a}_i^p \rangle^{(2)}$ and $\langle \mathbf{b}_i^p \rangle^{(2)}$ only exist in $\{p_i - \mathbf{a}_i^p, p_i - \mathbf{b}_i^p\}_i$ and they are independent of all other variables, we can remove $\{p_i - \mathbf{a}_i^p, p_i - \mathbf{b}_i^p\}_i$ from conditioned terms.
- For the same reason $\{BVp_i^{(1)}\}_{i=1}^n$ are independent of all other variables and can be removed.
- The secret shares of aggregation weights $\langle p_i \rangle^{(1)} := (p_i + \eta_i)/2$ and $\langle p_i \rangle^{(2)} := (p_i - \eta_i)/2$ where η_i is random noise. Then $\{\langle p_i \rangle^{(1)}\}_i$ are independent of all other variables. Thus it can be removed.

Now the left hand side (LHS) can be simplified as

$$\begin{aligned} LHS = & \mathbb{P}(\mathbf{x}_i = x_i | \{\langle \mathbf{x}_i \rangle^{(1)}\}_{i=1}^n, \\ & \{BV_{i,j}^{(1)}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_{ij}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{b}_{ij}, \\ & \langle \|\mathbf{x}_i - \mathbf{x}_j\|^2 \rangle^{(1)}\}_{i < j}, \mathbf{z}) \end{aligned} \quad (\text{C.1})$$

There are other independence properties:

- The secret shares of the input $\langle \mathbf{x}_i \rangle$ can be seen as generated by random noise ξ_i . Thus $\langle \mathbf{x}_i \rangle^{(1)} := (\xi_i + \mathbf{x}_i)/2$ and $\langle \mathbf{x}_i \rangle^{(2)} := (-\xi_i + \mathbf{x}_i)/2$ are independent of others like \mathbf{x}_i . Besides, for all $j \neq i$, $\langle \mathbf{x}_i \rangle^{(\cdot)}$ and $\langle \mathbf{x}_j \rangle^{(\cdot)}$ are independent.
- Beaver's triple $\{BV_{i,j}^{(1)}\}_{i < j}$ and $\{BV_{i,j}^{(2)}\}_{i < j}$ are clearly independent. Since they are generated before the existence of data, they are always independent of $\{\mathbf{x}_j^{(\cdot)}\}_j$.

Next, according to Beaver's multiplication Algorithm 8,

$$\langle \|\mathbf{x}_i - \mathbf{x}_j\|^2 \rangle^{(1)} = \mathbf{c}_{ij}^{(1)} + (\mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_{ij})\mathbf{b}_{ij}^{(1)} + (\mathbf{x}_i - \mathbf{x}_j - \mathbf{b}_{ij})\mathbf{a}_{ij}^{(1)}$$

we can remove this term from condition:

$$\begin{aligned} LHS = & \mathbb{P}(\mathbf{x}_i = x_i | \{\langle \mathbf{x}_i \rangle^{(1)}\}_{i=1}^n, \mathbf{z}, \\ & \{BV_{i,j}^{(1)}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_{ij}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{b}_{ij}\}_{i < j}) \end{aligned} \quad (\text{C.2})$$

By the independence between $\langle \mathbf{x}_i \rangle^{(\cdot)}$ and $BV_{ij}^{(\cdot)}$, we can further simplify the conditioned term

$$\begin{aligned} LHS = & \mathbb{P}(\mathbf{x}_i = x_i | \{\langle \mathbf{x}_i \rangle^{(1)}\}_{i=1}^n, \mathbf{z}, \\ & \{BV_{i,j}^{(1)}, \langle \mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_{ij} \rangle^{(2)}, \langle \mathbf{x}_i - \mathbf{x}_j - \mathbf{b}_{ij} \rangle^{(2)}\}_{i < j}) \end{aligned} \quad (\text{C.3})$$

Since $BV_{ij}^{(1)}$ and $BV_{ij}^{(2)}$ are always independent of all other variables, we know that

$$LHS = \mathbb{P}(\mathbf{x}_i = x_i | \{\langle \mathbf{x}_i \rangle^{(1)}\}_{i=1}^n, \mathbf{z}) \quad (\text{C.4})$$

For worker i , $\forall j \neq i$, $\langle \mathbf{x}_i \rangle^{(\cdot)}$ and $\langle \mathbf{x}_j \rangle^{(1)}$ are independent

$$LHS = \mathbb{P}(\mathbf{x}_i = x_i | \mathbf{z}).$$

□

Theorem 4.2 (Privacy for **S2**). *Let $\{p_i\}_{i=1}^n$ is the output of byzantine oracle or a vector of 1s (non-private). Let $BV_{ij} = \langle \mathbf{a}_{ij}, \mathbf{b}_{ij}, \mathbf{c}_{ij} \rangle$ and $BVp_i = \langle \mathbf{a}_i^p, \mathbf{b}_i^p, \mathbf{c}_i^p \rangle$ be the Beaver's triple used in the multiplications. Let $\langle \cdot \rangle^{(2)}$ be the share of the secret-shared values $\langle \cdot \rangle$ on **S2**. Then for all*

workers i

$$\begin{aligned} & \mathbb{P}(\mathbf{x}_i = x_i \mid \{\langle \mathbf{x}_i \rangle^{(2)}, \langle p_i \rangle^{(2)}, p_i\}_{i=1}^n, \{BV_{ij}^{(2)}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_{ij}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{b}_{ij}\}_{i < j}, \\ & \quad \{\langle \|\mathbf{x}_i - \mathbf{x}_j\|^2 \rangle^{(2)}, \|\mathbf{x}_i - \mathbf{x}_j\|^2\}_{i < j}, \{BVp_i^{(2)}, p_i - \mathbf{a}_i^p, p_i - \mathbf{b}_i^p\}_{i=1}^n) \\ & = \mathbb{P}(\mathbf{x}_i = x_i \mid \{\|\mathbf{x}_i - \mathbf{x}_j\|^2\}_{i < j}) \end{aligned} \quad (4.1)$$

Note that the conditioned values are what **S2** observed throughout the algorithm. $\{BV_{ij}^{(2)}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_{ij}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{b}_{ij}\}_{i < j}$ and $\{BVp_i^{(2)}, p_i - \mathbf{a}_i^p, p_i - \mathbf{b}_i^p\}_{i=1}^n$ are intermediate values during shared values multiplication.

Proof. Similar to the proof of Theorem 4.1, we can first conclude

- $\{p_i - \mathbf{a}_i^p, p_i - \mathbf{b}_i^p\}_i$ and $\{BVp_i^{(2)}\}_{i=1}^n$ could be dropped because these they are data independent and no other terms depend on them.
- $\{\langle p_i \rangle^{(2)}\}_{i=1}^n$ is independent of the others so it can be dropped.
- $\{p_i\}_{i=1}^n$ can be inferred from $\{\|\mathbf{x}_i - \mathbf{x}_j\|^2\}_{ij}$ so it can also be dropped.
- By the definition of $\{\langle \|\mathbf{x}_i - \mathbf{x}_j\|^2 \rangle^{(2)}\}_{ij}$, it can be represented by $\{\mathbf{x}_i\}^{(2)}$ and $\{BV_{ij}^{(2)}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_{ij}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{b}_{ij}\}_{i < j}$.

Now the left hand side (LHS) can be simplified as

$$\begin{aligned} LHS = & \mathbb{P}(\mathbf{x}_i = x_i \mid \{\langle \mathbf{x}_i \rangle^{(2)}\}_{i=1}^n, \\ & \{BV_{ij}^{(2)}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_{ij}, \mathbf{x}_i - \mathbf{x}_j - \mathbf{b}_{ij}, \\ & \|\mathbf{x}_i - \mathbf{x}_j\|^2\}_{i < j}) \end{aligned} \quad (C.5)$$

Because \mathbf{x}_i is independent of $\{\langle \mathbf{x}_i \rangle^{(2)}\}_{i=1}^n$ as well as data independent terms like $\{BV_{ij}^{(2)}, \mathbf{a}_{ij}^{(1)}, \mathbf{b}_{ij}^{(1)}\}_{i < j}$, we have

$$LHS = \mathbb{P}(\mathbf{x}_i = x_i \mid \|\mathbf{x}_i - \mathbf{x}_j\|^2\}_{i < j})$$

□

Theorem 4.3 (from DP to LDP). Suppose that the noise ν_t in (4.2) is sufficient to ensure that the set of model parameters $\{\mathbf{w}_t\}_{t \in [T]}$ satisfy (ϵ, δ) -DP for $\epsilon \geq 1$. Then, running (4.2) with using Alg. 4 to compute $(\mathbf{x}_t + \eta_t)$ by securely aggregating $\{\mathbf{x}_{1,t} + n\eta_t, \mathbf{x}_{2,t}, \dots, \mathbf{x}_{n,t}\}$ satisfies (ϵ, δ) -LDP.

Proof. Suppose that worker $i \in [n]$ computes its gradient \mathbf{x}_i based on data $d_i \in \mathcal{D}$. For the sake of simplicity, let us assume that the aggregate model satisfies ϵ -DP. The proof is identical for the more relaxed notion of (ϵ, δ) -DP for $\epsilon \geq 1$. This implies that for any $j \in [n]$ and $d_j, \tilde{d}_j \in \mathcal{D}$,

$$\frac{\Pr\left[\frac{1}{n}(\sum_{i=1}^n \mathbf{x}_i(d_i)) + \nu = \mathbf{y}\right]}{\Pr\left[\frac{1}{n}(\sum_{i \neq j} \mathbf{x}_i(d_i)) + \frac{1}{n}\mathbf{x}_j(\tilde{d}_j) + \nu = \mathbf{y}\right]} \leq \epsilon, \forall \mathbf{y}. \quad (C.6)$$

Now, we examine the communication received by each server and measure how much information is revealed about any given worker $j \in [n]$. The values stored and seen are:

- **S1:** The secret share $(\mathbf{x}_1 + n\nu)^{(1)}$, $\{\mathbf{x}_i(d_i)^{(1)}\}_{i=2}^n$ and the sum of other shares $(\mathbf{x}_1 + n\nu)^{(2)} + \sum_{i=2}^n \mathbf{x}_i(d_i)^{(2)} = ((\sum_{i=1}^n \mathbf{x}_i(d_i)) + n\nu)^{(2)}$.
- **S2:** The secret share $(\mathbf{x}_1 + n\nu)^{(2)}$, $\{\mathbf{x}_i(d_i)^{(2)}\}_{i=2}^n$.
- Worker i : $\mathbf{z} = (\sum_{i=1}^n \mathbf{x}_i(d_i)) + n\nu$.

The equality above is because our secret shares are *linear*. Now, the values seen by any worker satisfy ϵ -LDP directly by (C.6). For the server, note that by the definition of our secret shares, we have for any worker j ,

$$\begin{aligned} & \mathbf{x}_j(d_j)^{(1)} \text{ is independent of } \mathbf{x}_j(d_j) \\ \Rightarrow & \Pr[\mathbf{x}_j(d_j)^{(1)} = y] = \Pr[\mathbf{x}_j(d_j)^{(1)} = \tilde{y}], \forall y, \tilde{y} \\ \Rightarrow & \Pr[\mathbf{x}_j(d_j)^{(1)} = y] = \Pr[\mathbf{x}_j(\tilde{d}_j)^{(1)} = y], \forall d_j, \tilde{d}_j \in \mathcal{D}. \end{aligned}$$

A similar statement holds for the second share. This proves that the values computed/seen by the workers or servers satisfy ϵ -LDP. \square

C.2 Notes on security

C.2.1 Beaver's MPC Protocol

Algorithm 8 Beaver [1991]'s MPC Protocol

input: $\langle x \rangle$; $\langle y \rangle$; Beaver's triple $(\langle a \rangle, \langle b \rangle, \langle c \rangle)$ s.t. $c = ab$

output: $\langle z \rangle$ s.t. $z = xy$

for party i **do**

 locally compute $x_i - a_i$ and $y_i - b_i$ and then broadcast them to all parties

 collect all shares and reveal $x - a = \sum_i (x_i - a_i)$, $y - b = \sum_i (y_i - b_i)$

 compute $z_i := c_i + (x - a)b_i + (y - b)a_i$

The first party 1 updates $z_1 := z_1 + (x - a)(y - b)$

In this section, we briefly introduce Beaver [1991]'s classic implementations of addition $\langle x + y \rangle$ and multiplication $\langle xy \rangle$ given additive secret-shared values $\langle x \rangle$ and $\langle y \rangle$ where each party i holding x_i and y_i . The algorithm for multiplication is given in Algorithm 8.

Addition. The secret-shared values form of sum, $\langle x + y \rangle$, is obtained by simply each party i locally compute $x_i + y_i$.

Multiplication. Assume we already have three secret-shared values called a triple, $\langle a \rangle$, $\langle b \rangle$, and $\langle c \rangle$ such that $c = ab$.

Then note that if each party broadcasts $x_i - a_i$ and $y_i - b_i$, then each party i can compute $x - a$ and $y - b$ (so these values are publicly known), and hence compute

$$z_i := c_i + (x - a)b_i + (y - b)a_i$$

Additionally, one party (chosen arbitrarily) adds on the public value $(x - a)(y - b)$ to their share so that summing all the shares up, the parties get

$$\sum_i z_i = c + (x - a)b + (y - b)a + (x - a)(y - b) = xy$$

and so they have a secret sharing $\langle z \rangle$ of xy .

The generation of Beaver's triples. There are many different implementations of the offline phase of the MPC multiplication. For example, semi-homomorphic encryption based implementations [Keller et al., 2018] or oblivious transfer-based implementations [Keller et al., 2016]. Since their security and performance have been demonstrated, we may assume the Beaver's triples are ready for use at the initial step of our protocol.

C.2.2 Notes on obtaining a secret share

Suppose that we want to secret share a bounded real vector $\mathbf{x} \in (-B, B]^d$ for some $B \geq 0$. Then, we sample a random vector ξ uniformly from $(-B, B]^d$. This is easily done by sampling each coordinate independently from $(-B, B]$. Then the secret shares become $(\xi, \mathbf{x} - \xi)$. Since ξ is drawn from a uniform distribution from $(-B, B]^d$, the distribution of $\mathbf{x} - \xi$ conditioned on \mathbf{x} is still uniform over $(-B, B]^d$ and (importantly) independent of \mathbf{x} . All arithmetic operations are then carried out modulo $[-B, B]$ i.e. $B + 1 \equiv -B + 1$ and $-B - 1 \equiv B - 1$. This simple scheme ensures information theoretic input-privacy for continuous vectors.

The scheme described above requires access to true randomness i.e. the ability to sample uniformly from $(-B, B]$. We make this assumption to simplify the proofs and the presentation. We note that differential privacy techniques such as [Abadi et al., 2016] also assume access to a similar source of true randomness. In practice, however, this would be replaced with a pseudo-random-generator (PRG) [Blum and Micali, 1984; Yao, 1982].

C.2.3 Computational indistinguishability

Let $\{X_n\}, \{Y_n\}$ be sequences of distributions indexed by a security parameter n (like the length of the input). $\{X_n\}$ and $\{Y_n\}$ are *computationally indistinguishable* if for every polynomial-time A and polynomially-bounded ε , and sufficiently large n

$$|\Pr[A(X_n) = 1] - \Pr[A(Y_n) = 1]| \leq \varepsilon(n) \tag{C.7}$$

If a pseudorandom generator, instead of true randomness, is used in § C.2.2, then the shares are indistinguishable from a uniform distribution over a field of same length. Thus in Theorem 4.1 and Theorem 4.2, the secret shares can be replaced by an independent random variable of uniform distribution with negligible change in probability.

C.2.4 Notes on the security of **S2**

Theorem 4.2 proves that **S2** does not learn anything besides the pairwise distances between the various models. While this does leak some information about the models, **S2** cannot use this information to reconstruct any \mathbf{x}_i . This is because the pair-wise distances are invariant to translations, rotations, and shuffling of the coordinates of $\{\mathbf{x}_i\}$.

This remains true even if **S2** additionally learns the global model too.

C.3 Data ownership diagram

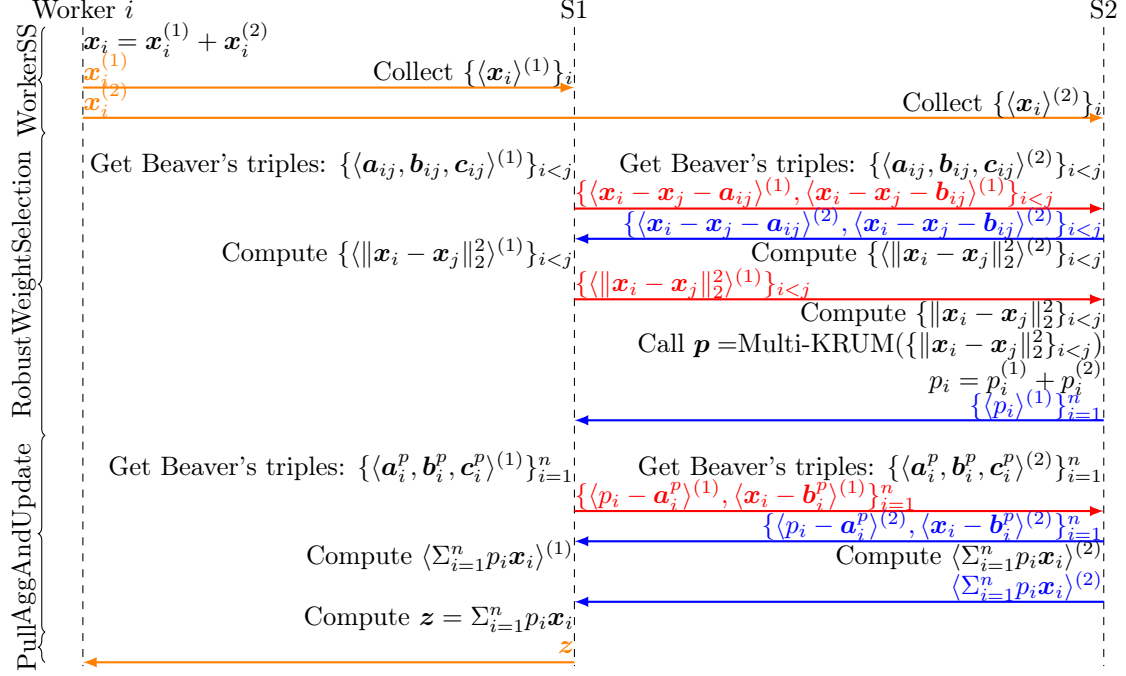


Fig. C.1 Overview of data ownership and Algorithm 4. The underlying Byzantine-robust oracle is Multi-Krum.

In Figure C.1, we show a diagram of data ownership to demonstrate of the data transmitted among workers and servers. Note that the Beaver's triples are already local to each server so that no extra communication is needed.

C.4 Example: Two-server protocol with ByzantineSGD oracle

We can replace MultiKrum with ByzantineSGD in [Alistarh et al., 2018]. To fit into our protocol, we make some minor modifications but still guarantee that output is same. The core part of [Alistarh et al., 2018] is listed in Algorithm 9.

Algorithm 9 ByzantineSGD [Alistarh et al., 2018]

input: \mathcal{I} is the set of good workers, $\{A_i\}_{i \in [m]}$, $\{\|B_i - B_j\|\}_{i < j}$, $\{\|\nabla_{k,i} - \nabla_{k,j}\|\}_{i < j}$ ($i, j \in [m]$), thresholds $\mathfrak{T}_A, \mathfrak{T}_B > 0$

output: Subset good workers \mathcal{S}

$A_{\text{med}} := \text{median}\{A_1, \dots, A_m\};$

$B_{\text{med}} \leftarrow B_i$ where $i \in [m]$ is any machine s.t. $|\{j \in [m] : \|B_j - B_i\| \leq \mathfrak{T}_B\}| > m/2;$

$\nabla_{\text{med}} \leftarrow \nabla_{k,i}$ where $i \in [m]$ is any machine s.t. $|\{j \in [m] : \|\nabla_{k,j} - \nabla_{k,i}\| \leq 2\nu\}| > m/2;$

$\mathcal{S} \leftarrow \{i \in \mathcal{I} : |A_i - A_{\text{med}}| \leq \mathfrak{T}_A \wedge \|B_i - B_{\text{med}}\| \leq \mathfrak{T}_B \wedge \|\nabla_{k,j} - \nabla_{k,i}\| \leq 4\nu\};$

The main algorithm can be summarized in Algorithm 10, the red lines highlights the changes. Different from Multi-Krum [Blanchard et al., 2017], Alistarh et al. [2018] uses states in their algorithm. As a result, the servers need to keep track of such states.

Algorithm 10 Two-Server Secure ByzantineSGD

Setup:

- n workers, at most α percent of which are Byzantine.
- Two non-colluding servers **S1** and **S2**
- **ByzantineSGD Oracle**: returns an indices set \mathcal{S} .
 - With thresholds \mathfrak{T}_A and \mathfrak{T}_B
 - Oracle state $A_i^{\text{old}}, \langle B_i^{\text{old}} \rangle$ for each worker i

Workers:

1. (**WorkerSecretSharing**):
 - (a) randomly split private \mathbf{x}_i into additive secret shares $\langle \mathbf{x}_i \rangle = \{\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}\}$ (such that $\mathbf{x}_i = \mathbf{x}_i^{(1)} + \mathbf{x}_i^{(2)}$)
 - (b) sends $\mathbf{x}_i^{(1)}$ to **S1** and $\mathbf{x}_i^{(2)}$ to **S2**

Servers:

1. $\forall i$, **S1** collects gradient $\mathbf{x}_i^{(1)}$ and **S2** collects $\mathbf{x}_i^{(2)}$.
 - (a) Use Beaver's triple to compute $A_i := \langle \langle \mathbf{x}_i \rangle, \langle \mathbf{w} - \mathbf{w}_0 \rangle \rangle_{\text{inner}} + A_i^{\text{old}}$
 - (b) $\langle B_i \rangle := \langle \mathbf{x}_i \rangle + \langle B_i^{\text{old}} \rangle$
2. (**RobustSubsetSelection**):
 - (a) For each pair (i, j) of gradients computes their distance ($i < j$):
 - On **S1** and **S2**, compute $\langle B_i - B_j \rangle = \langle B_i \rangle - \langle B_j \rangle$ locally
 - Use precomputed Beaver's triple and Algorithm 8 to compute the distance $\|B_i - B_j\|^2$
 - On **S1** and **S2**, compute $\langle \mathbf{x}_i - \mathbf{x}_j \rangle = \langle \mathbf{x}_i \rangle - \langle \mathbf{x}_j \rangle$ locally
 - Use precomputed Beaver's triple and Algorithm 8 to compute the distance $\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$
 - (b) **S2** perform Byzantine SGD $\mathcal{S} = \text{ByzantineSGD}(\{A_i\}_i, \{\|B_i - B_j\|\}_{i < j}, \{\|\mathbf{x}_i - \mathbf{x}_j\|\}_{i < j}, \mathfrak{T}_A, \mathfrak{T}_B)$; if $|\mathcal{S}| < 2$, exit; Convert \mathcal{S} to a weight vector \mathbf{p} of length n
 - (c) **S2** secret-shares $\langle \mathbf{p} \rangle$ with **S1**
3. (**AggregationAndUpdate**):
 - (a) On **S1** and **S2**, use MPC multiplication to compute $\langle \sum_{i=1}^n p_i \mathbf{x}_i \rangle$ locally
 - (b) **S2** sends its share of $\langle \sum_{i=1}^n p_i \mathbf{x}_i \rangle^{(2)}$ to **S1**
 - (c) **S1** reveals $\mathbf{z} = \sum_{i=1}^n p_i \mathbf{x}_i$ to all workers.
 - (d) **S2** updates $A_i^{\text{old}} \leftarrow A_i, \langle B_i^{\text{old}} \rangle \leftarrow \langle B_i \rangle$

Workers:

1. (**WorkerPullModel**): Collect \mathbf{z} and update model $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{z}$ locally
-

C.5 Additional experiments

We benchmark the performance of our two-server protocol with one-server protocol on the google kubernetes engine. We create a cluster of 8 nodes (machine-type=e2-standard-2) where 2 servers are deployed on different nodes and the workers are deployed evenly onto the rest 6 nodes. We run the experiments with 5, 10, 20, 50 workers and a large model of 25.6 million parameters (similar to ResNet-56) and a small model of 1.2 million parameters. We only record the time spent on communication and aggregation (krum). We benchmark each experiment for three times and take their average. The results are shown in Figure C.2.

Scaling with dimensions. In Figure C.2a, we compute the ratio of time spent on large model and small model. We can see that the ratio of two-server model is very close to the ideal ratio which suggests it scales linearly with dimensions. This is expected because krum scales linearly with dimension. For aggregation rules based on high-dimensional robust mean estimation, we can remove the dependence on d . We leave it as a future work to incorporate more efficient robust aggregation functions.

Scaling with number of workers. In Figure C.2b, we can see that the time spent on both one-server and two-server model grow with $O(n^2)$. However, we notated that this complexity comes from the aggregation rule we use, which is krum, not from our core protocol. For other aggregation rules like ByzantineSGD Alistarh et al. [2018], the complexity of aggregation rule is $O(n)$ and we can observe better scaling effects. We leave it as a future work to incorporate and benchmark more efficient robust aggregation rules.

Setups. Note that in our experiments, the worker-to-server communication and server-to-server communication has same bandwidth of 1Gb/s. In the realistic application, the link between servers can be infiniband and the bandwidth between worker and server are typically smaller. Thus, this protocol will be more efficient than we have observed here.

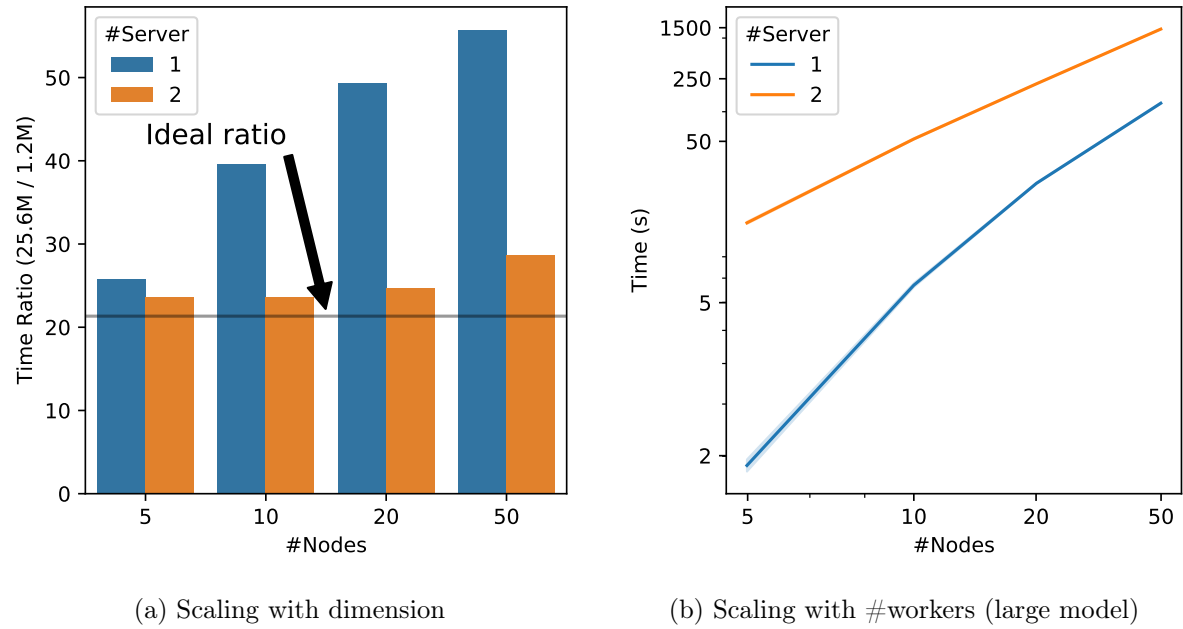


Fig. C.2 Scaling two-server model and one-server model to 5, 10, 20, 50 nodes.

Appendix D

RelaySum for Decentralized Deep Learning on Heterogeneous Data

D.1 Convergence Analysis of RelaySGD

The structure of this section is as follows: § D.1.1 describes the notations used in the proof; § D.1.2 introduces the properties of mixing matrix \mathbf{W} and useful inequalities and lemmas; § D.1.3 elaborates the results of Theorem 5.1 for non-convex, convex, and strongly convex objectives, all of the technical details are deferred to § D.1.4, § D.1.5 and § D.1.6.

D.1.1 Notation

We use upper case, bold letters for matrices and lower case, bold letters for vectors. By default, let $\|\cdot\|$ and $\|\cdot\|_F$ be the spectral norm and Frobenius norm for matrices and 2-norm $\|\cdot\|_2$ be the Euclidean norm for vectors.

Let τ_{ij} be the delay between node i and node j and let $\tau_{\max} = \max_{ij} \tau_{ij}$. Let

$$\mathbf{Z}^{(t)} = [\mathbf{x}_1^{(t)}, \dots, \mathbf{x}_n^{(t)}]^\top \in \mathbb{R}^{n \times d}$$

be the state at time t and let

$$\nabla \mathbf{F}^{(t)} = [\nabla F_1(\mathbf{x}_1^{(t)}; \xi_1^{(t)}), \dots, \nabla F_n(\mathbf{x}_n^{(t)}; \xi_n^{(t)})]^\top \in \mathbb{R}^{n \times d}$$

be the worker gradients at time t . Denote $\mathbf{Y}^{(t)}$ and $\mathbf{G}^{(t)}$ as the state (models) and gradients respectively, of all nodes, from time $t - \tau_{\max}$ to t .

$$\mathbf{Y}^{(t)} = \begin{bmatrix} \mathbf{Z}^{(t)} \\ \mathbf{Z}^{t-1} \\ \vdots \\ \mathbf{Z}^{t-\tau_{\max}} \end{bmatrix} \in \mathbb{R}^{n(\tau_{\max}+1) \times d}, \quad \mathbf{G}^{(t)} = \begin{bmatrix} \nabla \mathbf{F}^{(t)} \\ \nabla \mathbf{F}^{t-1} \\ \vdots \\ \nabla \mathbf{F}^{t-\tau_{\max}} \end{bmatrix} \in \mathbb{R}^{n(\tau_{\max}+1) \times d}.$$

The mixing matrix \mathbf{W} can be alternatively defined as follows

Definition D.1 (Mixing matrix \mathbf{W}). Define $\mathbf{W}, \tilde{\mathbf{W}} \in \mathbb{R}^{n(\tau_{\max}+1) \times n(\tau_{\max}+1)}$ such that RelaySGD can be reformulated as

$$\mathbf{Y}^{(t+1)} = \underbrace{\begin{bmatrix} \mathbf{W}_0 & \mathbf{W}_1 & \dots & \mathbf{W}_{\tau_{\max}-1} & \mathbf{W}_{\tau_{\max}} \\ \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \dots & \mathbf{I} & \mathbf{0} \end{bmatrix}}_{\mathbf{W}} \mathbf{Y}^{(t)} - \gamma \underbrace{\begin{bmatrix} \mathbf{W}_0 & \mathbf{W}_1 & \dots & \mathbf{W}_{\tau_{\max}-1} & \mathbf{W}_{\tau_{\max}} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\tilde{\mathbf{W}}} \mathbf{G}^{(t)}$$

where $\sum_{i=1}^n \mathbf{W}_i = \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top$.

D.1.2 Technical Preliminaries

Properties of \mathbf{W} .

In this part, we show that \mathbf{W} enjoys similar properties as Perron-Frobenius Theorem in Theorem D.1 and its left dominant eigenvector $\boldsymbol{\pi}$ has specific structure in Lemma D.1. Then we use the established tools to prove the key Lemma 5.1. Finally, we define constants C and C_1 in Definition D.3 which are used to simplify the convergence results in § D.1.3.

Definition D.2 (Spectral radius.). Let $\lambda_1, \dots, \lambda_n$ be the eigenvalues of a matrix $\mathbf{A} \in \mathbb{C}^{n \times n}$. Then its spectral radius $\rho(\mathbf{A})$ is defined as:

$$\rho(\mathbf{A}) = \max\{|\lambda_1|, \dots, |\lambda_n|\}.$$

Lemma D.1. The \mathbf{W} in Definition D.1 satisfies

1. The spectral radius $\rho(\mathbf{W}) = 1$ and 1 is an eigenvalue of \mathbf{W} and $\mathbf{1}_{n(\tau_{\max}+1)} \in \mathbb{R}^{n(\tau_{\max}+1)}$ is its right eigenvector.
2. The left eigenvector $\boldsymbol{\pi} \in \mathbb{R}^{n(\tau_{\max}+1)}$ of eigenvalue 1 is nonnegative and $[\boldsymbol{\pi}]_i = \pi_0 > 0, \forall i \in [n]$ and $\boldsymbol{\pi}^\top \mathbf{1}_{n(\tau_{\max}+1)} = 1$.

Proof. Since \mathbf{W} is a row stochastic matrix, the Gershgorin Circle Theorem asserts the spectral radius

$$\rho(\mathbf{W}) = |\lambda_1(\mathbf{W})| \leq 1.$$

It is clear that 1 is an eigenvalue of \mathbf{W} and $\mathbf{1}_{n(\tau_{\max}+1)}$ is its right eigenvector, we have $\rho(\mathbf{W}) = 1$.

Let $\boldsymbol{\pi} \in \mathbb{R}^{n(\tau_{\max}+1)}$ be the left eigenvector corresponding to 1 and denote it as

$$\boldsymbol{\pi} = \begin{bmatrix} \boldsymbol{\pi}_0 \\ \boldsymbol{\pi}_1 \\ \vdots \\ \boldsymbol{\pi}_{\tau_{\max}} \end{bmatrix} \in \mathbb{R}^{n(\tau_{\max}+1)}$$

where $\boldsymbol{\pi}_i \in \mathbb{R}^n, \forall i = 0, 1, \dots, \tau_{\max}$. Since $\boldsymbol{\pi} = \mathbf{W}^\top \boldsymbol{\pi}$, we have

$$\begin{bmatrix} \boldsymbol{\pi}_0 \\ \boldsymbol{\pi}_1 \\ \vdots \\ \boldsymbol{\pi}_{\tau_{\max}} \end{bmatrix} = \boldsymbol{\pi} = \mathbf{W}^\top \boldsymbol{\pi} = \begin{bmatrix} \mathbf{W}_0^\top \boldsymbol{\pi}_0 + \boldsymbol{\pi}_1 \\ \mathbf{W}_1^\top \boldsymbol{\pi}_0 + \boldsymbol{\pi}_2 \\ \vdots \\ \mathbf{W}_{\tau_{\max}-1}^\top \boldsymbol{\pi}_0 + \boldsymbol{\pi}_{\tau_{\max}} \\ \mathbf{W}_{\tau_{\max}}^\top \boldsymbol{\pi}_0 \end{bmatrix}$$

which holds true in each block. Then summing up all blocks yields

$$\sum_{i=0}^{\tau_{\max}} \boldsymbol{\pi}_i = \left(\sum_{i=0}^{\tau_{\max}} \mathbf{W}_i^\top \right) \boldsymbol{\pi}_0 + \sum_{i=1}^{\tau_{\max}} \boldsymbol{\pi}_i = \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top \boldsymbol{\pi}_0 + \sum_{i=1}^{\tau_{\max}} \boldsymbol{\pi}_i$$

which means $\boldsymbol{\pi}_0 = \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top \boldsymbol{\pi}_0$ and therefore $\boldsymbol{\pi}_0 = \pi_0 \mathbf{1}_n$ is a vector of same value.

Other coordinate blocks of $\boldsymbol{\pi}$ can be derived as

$$\boldsymbol{\pi}_i = \left(\sum_{k=i}^{\tau_{\max}} \mathbf{W}_k^\top \right) \boldsymbol{\pi}_0 \quad \forall i = 1, \dots, \tau_{\max}.$$

Since \mathbf{W}_i are nonnegative matrices, we can scale $\boldsymbol{\pi}$ such that $\pi_0 > 0$ and $\mathbf{1}^\top \boldsymbol{\pi} = 1$. Therefore $\boldsymbol{\pi}$ is a nonnegative vector. \square

Lemma D.2. *If $\lambda \in \mathbf{C}$ is an eigenvalue of \mathbf{W} and $|\lambda| = \rho(\mathbf{W}) = 1$, then $\lambda = 1$ and its geometric multiplicity is 1.*

Proof. Let $\mathbf{v} \in \mathbf{C}^{n(\tau_{\max}+1)}$ be a right eigenvector corresponding to eigenvalue $\lambda \in \mathbf{C}$ which $|\lambda| = 1$.

Denote \mathbf{v} as

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_{\tau_{\max}} \end{bmatrix} \in \mathbf{C}^{n(\tau_{\max}+1)}.$$

where $\mathbf{v}_i \in \mathbf{C}^n, \forall i = 0, \dots, \tau_{\max}$. Then $\mathbf{W}\mathbf{v} = \lambda\mathbf{v}$ implies

$$\mathbf{W}\mathbf{v} = \begin{bmatrix} \sum_{i=0}^{\tau_{\max}} \mathbf{W}_i \mathbf{v}_i \\ \mathbf{v}_0 \\ \vdots \\ \mathbf{v}_{\tau_{\max}-2} \\ \mathbf{v}_{\tau_{\max}-1} \end{bmatrix} = \lambda\mathbf{v} = \begin{bmatrix} \lambda\mathbf{v}_0 \\ \lambda\mathbf{v}_1 \\ \vdots \\ \lambda\mathbf{v}_{\tau_{\max}} \end{bmatrix}.$$

The last τ equations ensures $\mathbf{v}_i = \lambda^{-i}\mathbf{v}_0$ and thus the first equality becomes

$$\left(\sum_{i=0}^{\tau_{\max}} \mathbf{W}_i \lambda^{-i} \right) \mathbf{v}_0 = \lambda \mathbf{v}_0$$

Denote $\mathbf{v}_0 = [x_1, x_2, \dots, x_n]^\top \in \mathbf{C}^n$, then $\forall i = 1, \dots, n$

$$\sum_{j=1}^n \frac{1}{n} \lambda^{-\tau_{ij}} x_j = \lambda x_i. \quad (\text{D.1})$$

Pick i such that $|\lambda x_i| = \max_j |\lambda x_j|$, then

$$|\lambda x_i| = \left| \sum_{j=1}^n \frac{1}{n} \lambda^{-\tau_{ij}} x_j \right| \leq \frac{1}{n} \sum_{j=1}^n |\lambda^{-\tau_{ij}} x_j| = \frac{1}{n} \sum_{j=1}^n |\lambda^{-\tau_{ij}}| |x_j| = \frac{1}{n} \sum_{j=1}^n |x_j| \leq |x_i|$$

where we use the triangular inequality $|a+b| \leq |a|+|b|$ and $|ab| = |a||b|$ for all $a, b \in \mathbf{C}$.

Note that as $|\lambda x_i| = |\lambda||x_i| = |x_i|$, the triangular inequality is in fact an equality which means $\lambda^{-\tau_{ij}} x_j$ could be written as

$$\lambda^{-\tau_{ij}} x_j = a_{ij} \xi \quad \forall j \in [n].$$

where $a_{ij} \geq 0$ and $\xi \in \mathbf{C}$. Here $\xi \neq 0$, otherwise $\mathbf{v} = \mathbf{0}$ which contradicts to \mathbf{v} is an eigenvector. Then (D.1) becomes

$$\frac{1}{n} \sum_{j=1}^n a_{ij} \xi = \lambda a_{ii} \xi.$$

which implies $|\frac{1}{n} \sum_{j=1}^n a_{ij}| = |a_{ii}|$. As $|\lambda x_i| = \max_j |\lambda x_j|$, we know $a_{ii} \geq a_{ij}$ for all j , thus

$$a_{i1} = \dots = a_{in} = a \geq 0,$$

moreover, $a > 0$ as $a = 0$ again leads to $\mathbf{v} = \mathbf{0}$. Then (D.1) becomes

$$\lambda a \xi = \lambda x_i = \frac{1}{n} \sum_{j=1}^n \lambda^{-\tau_{ij}} x_j = \frac{1}{n} \sum_{j=1}^n a \xi = a \xi$$

which shows $\lambda = 1$ as $a > 0$ and $\xi \neq 0$.

Therefore, $\mathbf{v}_0 = a \mathbf{1}_n \in \mathbb{R}^n$ and $\mathbf{v} = a \mathbf{1}_{n(\tau_{\max}+1)} \in \mathbb{R}^{n(\tau_{\max}+1)}$. It mean the eigenspace of 1 is one-dimensional and thus its geometric multiplicity is 1. \square

Lemma D.3. *The algebraic multiplicity of eigenvalue 1 of \mathbf{W} is 1.*

Proof. Proof by contradiction. Let $\mathbf{P} \in \mathbb{R}^{n(\tau_{\max}+1) \times n(\tau_{\max}+1)}$ be the invertible matrix which transform \mathbf{W} to its Jordan normal form \mathbf{J} by

$$\mathbf{P}^{-1} \mathbf{W} \mathbf{P} = \mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & & \\ & \ddots & \\ & & \mathbf{J}_p \end{bmatrix}$$

where \mathbf{J}_1 is the block for eigenvalue 1. If we assume the algebraic multiplicity of 1 greater equal than 2, and use the Lemma D.2 that its geometric multiplicity is 1, then \mathbf{J}_1 should look like

$$\mathbf{J}_1 = \begin{bmatrix} 1 & 1 & & \\ & 1 & \ddots & \\ & & \ddots & 1 \\ & & & 1 \end{bmatrix}$$

which is a square matrix of at least 2 columns. Denote the first two columns of \mathbf{P} as \mathbf{p}_1 and \mathbf{p}_2 . We can see that $\mathbf{p}_1 = \mathbf{1}_{n(\tau_{\max}+1)}$. Then inspecting $\mathbf{P}^{-1} \mathbf{W} \mathbf{P} = \mathbf{J}$ for \mathbf{p}_2 yields

$$\mathbf{W} \mathbf{p}_2 = \mathbf{p}_1 + \mathbf{p}_2 = \mathbf{1}_{n(\tau_{\max}+1)} + \mathbf{p}_2.$$

Multiply both sides by $\boldsymbol{\pi}^\top$ gives

$$\begin{aligned} \boldsymbol{\pi}^\top \mathbf{W} \mathbf{p}_2 &= \boldsymbol{\pi}^\top \mathbf{1}_{n(\tau_{\max}+1)} + \boldsymbol{\pi}^\top \mathbf{p}_2 \\ \boldsymbol{\pi}^\top \mathbf{p}_2 &= \boldsymbol{\pi}^\top \mathbf{1}_{n(\tau_{\max}+1)} + \boldsymbol{\pi}^\top \mathbf{p}_2 \\ 0 &= \boldsymbol{\pi}^\top \mathbf{1}_{n(\tau_{\max}+1)} \end{aligned}$$

which contradicts Lemma D.1 that $\boldsymbol{\pi}^\top \mathbf{1}_{n(\tau_{\max}+1)} = 1$. Thus the algebraic multiplicity of 1 is 1. \square

Theorem D.1 (Perron-Frobenius Theorem for \mathbf{W}). *The mixing \mathbf{W} of RelaySGD satisfies*

1. (Positivity) $\rho(\mathbf{W}) = 1$ is an eigenvalue of \mathbf{W} .
2. (Simplicity) The algebraic multiplicity of 1 is 1.
3. (Dominance) $\rho(\mathbf{W}) = |\lambda_1(\mathbf{W})| > |\lambda_2(\mathbf{W})| \geq \dots \geq |\lambda_{n(\tau_{\max}+1)}(\mathbf{W})|$.
4. (Nonnegativity) The \mathbf{W} has a nonnegative left eigenvector $\boldsymbol{\pi}$ and right eigenvector $\mathbf{1}_{n(\tau_{\max}+1)}$.

Proof. Statements 1 and 4 follow from Lemma D.1. Statement 2 follows from Lemma D.3. Statement 3 follows from Lemma D.2 and Lemma D.3. \square

Lemma D.4 (Gelfand's formula). *For any matrix norm $\|\cdot\|$, we have*

$$\rho(\mathbf{A}) = \lim_{k \rightarrow \infty} \|\mathbf{A}^k\|^{\frac{1}{k}}.$$

We characterize the convergence rate of the consensus distance in the following key lemma:

Lemma' 5.1 (Key lemma). *Given \mathbf{W} and $\boldsymbol{\pi}$ as before. There exists an integer $m = m(\mathbf{W}) > 0$ such that for any $\mathbf{X} \in \mathbb{R}^{n(\tau_{\max}+1) \times d}$ we have*

$$\|\mathbf{W}^m \mathbf{X} - \mathbf{1}\boldsymbol{\pi}^\top \mathbf{X}\|^2 \leq (1-p)^{2m} \|\mathbf{X} - \mathbf{1}\boldsymbol{\pi}^\top \mathbf{X}\|^2,$$

where $p = \frac{1}{2}(1 - |\lambda_2(\mathbf{W})|)$ is a constant.

All the following optimization convergence results will only depend on the *effective spectral gap* $\rho := \frac{p}{m}$ of \mathbf{W} . We empirically observe that $\rho = \Theta(1/n)$ for a variety of network topologies, as shown in Figure D.1.

Proof of key lemma 5.1. First, let $\{\lambda_i\}$ and $\{\mathbf{v}_i\}$ be the eigenvalues and right eigenvectors of \mathbf{W} where $\lambda_1 = 1$ and $\mathbf{v}_1 = \mathbf{1}_{n(\tau_{\max}+1)}$, then

$$\begin{aligned} (\mathbf{W} - \mathbf{1}\boldsymbol{\pi}^\top) \mathbf{v}_1 &= (\mathbf{W} - \mathbf{1}\boldsymbol{\pi}^\top) \mathbf{1} = \mathbf{0} \\ (\mathbf{W} - \mathbf{1}\boldsymbol{\pi}^\top) \mathbf{v}_i &= \mathbf{W} \mathbf{v}_i - \mathbf{1}\boldsymbol{\pi}^\top \mathbf{v}_i = \mathbf{W} \mathbf{v}_i = \lambda_i \mathbf{v}_i \quad \forall i > 1 \end{aligned}$$

where $\boldsymbol{\pi}^\top \mathbf{v}_i = 0$ because

$$(1 - \lambda_i) \boldsymbol{\pi}^\top \mathbf{v}_i = \boldsymbol{\pi}^\top \mathbf{v}_i - \lambda_i \boldsymbol{\pi}^\top \mathbf{v}_i = (\boldsymbol{\pi}^\top \mathbf{W}) \mathbf{v}_i - \boldsymbol{\pi}^\top (\mathbf{W} \mathbf{v}_i) = 0.$$

The spectrum of $\mathbf{W} - \mathbf{1}\boldsymbol{\pi}^\top$ are

$$\{0, \lambda_2, \dots, \lambda_{n(\tau_{\max}+1)}\},$$

and thus the spectral radius of $\mathbf{W} - \mathbf{1}\boldsymbol{\pi}^\top$ is $|\lambda_2| < 1$. Since

$$\mathbf{W}^m - \mathbf{1}\boldsymbol{\pi}^\top = (\mathbf{W} - \mathbf{1}\boldsymbol{\pi}^\top)^m,$$



Fig. D.1 Optimal ratios for $\rho = p/m$ for Lemma 5.1 computed empirically for three common types of graph topologies.

then $\mathbf{W}^m - \mathbf{1}\pi^\top$ has a spectral radius of $|\lambda_2|^m < 1$.

Then, we apply Gelfand's formula (Lemma D.4) with $\mathbf{A} = \mathbf{W} - \mathbf{1}\pi^\top$ and can conclude that for a given $\epsilon \in (0, 1 - |\lambda_2|)$, there exists a large enough integer $m > 0$ such that

$$\|\mathbf{W}^m - \mathbf{1}\pi^\top\| = \|(\mathbf{W} - \mathbf{1}\pi^\top)^m\| \leq (\rho(\mathbf{W} - \mathbf{1}\pi^\top) + \epsilon)^m = (|\lambda_2| + \epsilon)^m < 1.$$

Thus

$$\|\mathbf{W}^m \mathbf{X} - \mathbf{1}\pi^\top \mathbf{X}\|^2 \leq \|\mathbf{W}^m - \mathbf{1}\pi^\top\|^2 \|\mathbf{X} - \mathbf{1}\pi^\top \mathbf{X}\|^2 \leq (1 - p)^{2m} \|\mathbf{X} - \mathbf{1}\pi^\top \mathbf{X}\|^2$$

where $p \in (0, 1 - |\lambda_2|)$. □

Definition D.3. Given \mathbf{W} and m , and $\tilde{\mathbf{I}} \in \mathbb{R}^{n(\tau_{\max}+1) \times n(\tau_{\max}+1)}$ is a matrix which satisfies

$$[\tilde{\mathbf{I}}]_{ij} = \begin{cases} 1 & i = j \leq n \\ 0 & \text{Otherwise.} \end{cases}$$

We define constants $C_1^2 := \max_{i=0, \dots, m-1} \|\mathbf{W}^i \tilde{\mathbf{I}}\|^2$ and $C = C(\mathbf{W})$ such that

$$C^2 := \frac{C_1^2}{\|\mathbf{W}^\infty \tilde{\mathbf{I}}\|^2}.$$

where $\mathbf{W}^\infty := \mathbf{1}\pi^\top$.

In addition, the $\|\mathbf{1}\pi^\top \tilde{\mathbf{I}}\|^2$ can be computed as follows.

Lemma D.5. *Given $\tilde{\mathbf{I}}$ in Definition D.3, we have the following estimate*

$$\|\mathbf{1}\boldsymbol{\pi}^\top \tilde{\mathbf{I}}\|^2 = n^2(\tau_{\max} + 1)\pi_0^2 \leq n^3\pi_0^2.$$

Proof. For rank r matrix $\|A\|^2 \leq \|A\|_F^2 \leq r\|A\|^2$. Since $\mathbf{1}\boldsymbol{\pi}^\top \tilde{\mathbf{I}}$ is a rank 1 matrix, we know that

$$\|\mathbf{1}\boldsymbol{\pi}^\top \tilde{\mathbf{I}}\|^2 = \|\mathbf{1}\boldsymbol{\pi}^\top \tilde{\mathbf{I}}\|_F^2.$$

As the first n entries of $\boldsymbol{\pi}$ are π_0 , we can compute that

$$\|\mathbf{1}\boldsymbol{\pi}^\top \tilde{\mathbf{I}}\|_F^2 = n^2(\tau_{\max} + 1)\pi_0^2.$$

□

Useful inequalities and lemmas

For convex objective, the noise in Assumption B can be defined only at the minimizer \mathbf{x}^* which leads to Assumption D. This assumption is used in the proof of Proposition D.2.

Assumption D (Bounded noise at the optimum). *Let $\mathbf{x}^* = \arg \min f(\mathbf{x})$ and define*

$$\zeta_i^2 := \|\nabla f_i(\mathbf{x}^*)\|^2, \quad \bar{\zeta}^2 := \frac{1}{n} \sum_{i=1}^n \zeta_i^2. \quad (\text{D.2})$$

Further, define

$$\sigma_i^2 := \mathbb{E}_{\xi_i} \|\nabla F_i(\mathbf{x}^*, \xi_i) - \nabla f_i(\mathbf{x}^*)\|^2$$

and similarly as above, $\bar{\sigma}^2 := \frac{1}{n} \sum_{i=1}^n \sigma_i^2$. We assume that $\bar{\sigma}^2$ and $\bar{\zeta}^2$ are bounded.

Lemma D.6 (Cauchy-Schwartz inequality). *For arbitrary set of n vectors $\{\mathbf{a}_i\}_{i=1}^n$, $\mathbf{a}_i \in \mathbb{R}^d$*

$$\left\| \sum_{i=1}^n \mathbf{a}_i \right\|^2 \leq n \sum_{i=1}^n \|\mathbf{a}_i\|^2. \quad (\text{D.3})$$

Lemma D.7. *If function $g(\mathbf{x})$ is L -smooth, then*

$$\|\nabla g(\mathbf{x}) - \nabla g(\mathbf{y})\|^2 \leq 2L(g(\mathbf{x}) - g(\mathbf{y}) - \langle \mathbf{x} - \mathbf{y}, \nabla g(\mathbf{y}) \rangle), \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d. \quad (\text{D.4})$$

Lemma D.8. *Let \mathbf{A} be a matrix with $\{\mathbf{a}_i\}_{i=1}^n$ as its columns and $\bar{\mathbf{a}} = \frac{1}{n} \sum_{i=1}^n \mathbf{a}_i$, $\bar{\mathbf{A}} = \bar{\mathbf{a}}\mathbf{1}^\top$ then*

$$\|\mathbf{A} - \bar{\mathbf{A}}\|_F^2 = \sum_{i=1}^n \|\mathbf{a}_i - \bar{\mathbf{a}}\|^2 \leq \sum_{i=1}^n \|\mathbf{a}_i\|^2 = \|\mathbf{A}\|_F^2. \quad (\text{D.5})$$

Lemma D.9. *Let \mathbf{A}, \mathbf{B} be two matrices*

$$\|\mathbf{AB}\|_F^2 \leq \|\mathbf{A}\|_F^2 \|\mathbf{B}\|^2. \quad (\text{D.6})$$

D.1.3 Results of Theorem 5.1

In this subsection, we summarize the precise results of Theorem 5.1 for convex, strongly convex and non-convex cases. The complete proofs for each case are then given in the following § D.1.4, § D.1.5 and § D.1.6.

Theorem' 5.1. *Given mixing matrix \mathbf{W} and $\tilde{\mathbf{W}}$, constant m, p defined in Lemma 5.1, C, C_1 defined in Definition D.3. Under Assumption A and B, then for any target accuracy $\epsilon > 0$,*

Non-convex: *if the objective is non-convex, then $\frac{1}{T+1} \sum_{t=0}^T \|\nabla f(\bar{\mathbf{x}}^{(t)})\|^2 \leq \epsilon$ after*

$$\mathcal{O} \left(\frac{\bar{\sigma}^2}{n\epsilon^2} + \frac{Cm\bar{\sigma}}{\sqrt{p}\epsilon^{3/2}} + \frac{C_1m}{p\epsilon} \right) Lr_0$$

iterations, where $r_0 = f(\mathbf{x}^{(0)}) - f^$.*

Convex: *if the objective is convex and \mathbf{x}^* is the minimizer, then $\frac{1}{T+1} \sum_{t=0}^T (f(\bar{\mathbf{x}}^{(t)}) - f(\mathbf{x}^*)) \leq \epsilon$ after*

$$\mathcal{O} \left(\frac{\bar{\sigma}^2}{n\epsilon^2} + \frac{Cm\sqrt{L}\bar{\sigma}}{\sqrt{p}\epsilon^{3/2}} + \frac{Lm\sqrt{n}C}{p\epsilon} \right) r_0$$

iterations, where $r_0 = \|\mathbf{x}^0 - \mathbf{x}^\|^2$.*

Strongly-convex: *if the objective is μ strongly convex and \mathbf{x}^* is the minimizer, then $\frac{1}{W_T} \sum_{t=0}^T w_t (\mathbb{E} f(\bar{\mathbf{x}}^{(t)}) - f^*) + \mu \mathbb{E} \|\bar{\mathbf{x}}^{(T+1)} - \mathbf{x}^*\|^2 \leq \epsilon$ after*

$$\tilde{\mathcal{O}} \left(\frac{\bar{\sigma}^2}{\mu n \epsilon^2} + \frac{Lm^2 C^2 \bar{\sigma}^2}{\mu n p^2 \epsilon} + \frac{s}{a} \log \frac{bsr_0}{\epsilon} \right)$$

iterations, where $r_0 = \|\mathbf{x}^0 - \mathbf{x}^\|^2$, $w_t = (1 - \frac{\mu\gamma n\pi_0}{2})^{-(t+1)}$ and $W_T = \sum_{t=0}^T w_t$ and $a = \frac{\mu n \pi_0}{2}$, $b = \frac{2}{n\pi_0}$, $s = \frac{aT}{\ln \max\{\frac{ba^2 T^2 r_0}{\pi_0 \bar{\sigma}^2}, 2\}}$.*

In all three cases, the convergence rate is independent of the heterogeneity ζ^2 .

D.1.4 Proof of Theorem 5.1 in the convex case

Let $\bar{\mathbf{x}}^{(t)} := (\boldsymbol{\pi}^\top \mathbf{Y}^{(t)})^\top$ and $\bar{\mathbf{Y}}^{(t)} := \mathbf{1} \boldsymbol{\pi}^\top \mathbf{Y}^{(t)}$. Let \mathbf{x}^* be the minimizer of f and define the following iterates

- $r_t := \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}^*\|^2$,
- $e_t := f(\bar{\mathbf{x}}^{(t)}) - f(\mathbf{x}^*)$,
- $\Xi_t := \frac{1}{n} \|\bar{\mathbf{Y}}^{(t)} - \mathbf{Y}^{(t)}\|_F^2$.

The consensus distance Ξ_t can be written as follows

$$\Xi_t = \frac{1}{n} \sum_{i=1}^n \sum_{\tau=0}^{\tau_{\max}} \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t-\tau)}\|^2. \quad (\text{D.7})$$

There is a related term $\sum_{i=1}^n \sum_{j=1}^n \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t-\tau_{ij})}\|^2$ which will be used frequently in the proof. The next lemma explains their relations.

Lemma D.10. *For all $t \geq 0$*

$$\sum_{i=1}^n \sum_{j=1}^n \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t-\tau_{ij})}\|^2 \leq n^2 \Xi_t.$$

where $\mathbf{x}^{(0)} = \mathbf{x}^{(-1)} = \dots = \mathbf{x}^{(-\tau_{\max})}$.

Proof. Rewrite the τ_{ij} as an indicator function

$$\sum_{i=1}^n \sum_{j=1}^n \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t-\tau_{ij})}\|^2 = \sum_{i=1}^n \sum_{j=1}^n \sum_{\tau=0}^{\tau_{\max}} \mathbf{1}_{\{\tau=\tau_{ij}\}} \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t-\tau)}\|^2.$$

This term can be relaxed by removing the indicator function

$$\sum_{i=1}^n \sum_{j=1}^n \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t-\tau_{ij})}\|^2 \leq n \sum_{i=1}^n \sum_{\tau=0}^{\tau_{\max}} \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t-\tau)}\|^2.$$

Then applying (D.7) for the consensus distance in vector form completes the proof. \square

The next two propositions upper bound the difference between stochastic gradients and full gradients.

Proposition D.2. *Under Assumption A and B. Then for $t \geq 0$,*

$$\mathbb{E} \left\| \boldsymbol{\pi}^\top \tilde{\mathbf{W}} (\mathbb{E} \mathbf{G}^{(t)} - \mathbf{G}^{(t)}) \right\|^2 \leq 3n\pi_0^2 (L^2 \Xi_t + 2Le_t + \bar{\sigma}^2).$$

Proof. Use T_0 to denote the left hand side quantity

$$\begin{aligned} T_0 &:= \mathbb{E} \left\| \boldsymbol{\pi}^\top \tilde{\mathbf{W}} (\mathbb{E} \mathbf{G}^{(t)} - \mathbf{G}^{(t)}) \right\|^2 \\ &= \mathbb{E} \left\| \frac{\pi_0}{n} \sum_{i=1}^n \sum_{j=1}^n (\nabla f_j(\mathbf{x}_j^{(t-\tau_{ij})}) - \nabla F_j(\mathbf{x}_j^{(t-\tau_{ij})}; \xi_j^{(t-\tau_{ij})})) \right\|^2 \\ &\stackrel{\text{Cauchy-Schwartz (D.3)}}{\leq} \frac{\pi_0^2}{n} \sum_{i=1}^n \mathbb{E} \left\| \sum_{j=1}^n (\nabla f_j(\mathbf{x}_j^{(t-\tau_{ij})}) - \nabla F_j(\mathbf{x}_j^{(t-\tau_{ij})}; \xi_j^{(t-\tau_{ij})})) \right\|^2. \end{aligned}$$

Since the randomness inside the norm are independent, we have

$$T_0 \leq \frac{\pi_0^2}{n} \sum_{i=1}^n \sum_{j=1}^n \mathbb{E} \left\| \nabla f_j(\mathbf{x}_j^{(t-\tau_{ij})}) - \nabla F_j(\mathbf{x}_j^{(t-\tau_{ij})}; \xi_j^{(t-\tau_{ij})}) \right\|^2.$$

Inside the vector norm, we can add and subtract terms the same terms and apply Cauchy-Schwartz (D.3)

$$\begin{aligned} T_0 &\leq \frac{3\pi_0^2}{n} \sum_{i=1}^n \sum_{j=1}^n \mathbb{E} \left\| \nabla F_j(\mathbf{x}_j^{(t-\tau_{ij})}; \xi_j^{(t-\tau_{ij})}) - \nabla F_j(\bar{\mathbf{x}}^{(t)}; \xi_j^{(t-\tau_{ij})}) + \nabla f_j(\mathbf{x}_j^{(t-\tau_{ij})}) - \nabla f_j(\bar{\mathbf{x}}^{(t)}) \right\|^2 \\ &\quad + \frac{3\pi_0^2}{n} \sum_{i=1}^n \sum_{j=1}^n \mathbb{E} \left\| \nabla F_j(\bar{\mathbf{x}}^{(t)}; \xi_j^{(t-\tau_{ij})}) - \nabla F_j(\mathbf{x}^*; \xi_j^{(t-\tau_{ij})}) + \nabla f_j(\bar{\mathbf{x}}^{(t)}) - \nabla f_j(\mathbf{x}^*) \right\|^2 \\ &\quad + \frac{3\pi_0^2}{n} \sum_{i=1}^n \sum_{j=1}^n \mathbb{E} \left\| \nabla F_j(\mathbf{x}^*; \xi_j^{(t-\tau_{ij})}) - \nabla f_j(\mathbf{x}^*) \right\|^2. \end{aligned}$$

Use the inequality that for $a = \mathbb{E} Y$, $\mathbb{E} \|Y - a\|^2 = \mathbb{E} \|Y\|^2 - \|a\|^2 \leq \mathbb{E} \|Y\|^2$, then we have

$$\begin{aligned} T_0 &\leq \frac{3\pi_0^2}{n} \sum_{i=1}^n \sum_{j=1}^n \mathbb{E} \left\| \nabla F_j(\mathbf{x}_j^{(t-\tau_{ij})}; \xi_j^{(t-\tau_{ij})}) - \nabla F_j(\bar{\mathbf{x}}^{(t)}; \xi_j^{(t-\tau_{ij})}) \right\|^2 \\ &\quad + \frac{3\pi_0^2}{n} \sum_{i=1}^n \sum_{j=1}^n \mathbb{E} \left\| \nabla F_j(\bar{\mathbf{x}}^{(t)}; \xi_j^{(t-\tau_{ij})}) - \nabla F_j(\mathbf{x}^*; \xi_j^{(t-\tau_{ij})}) \right\|^2 \\ &\quad + \frac{3\pi_0^2}{n} \sum_{i=1}^n \sum_{j=1}^n \mathbb{E} \left\| \nabla F_j(\mathbf{x}^*; \xi_j^{(t-\tau_{ij})}) - \nabla f_j(\mathbf{x}^*) \right\|^2 \end{aligned}$$

Applying Assumption A, Smoothness (D.4), and Assumption B (or Assumption D) to the three terms gives

$$\begin{aligned} T_0 &\leq \frac{3L^2\pi_0^2}{n} \sum_{i=1}^n \sum_{j=1}^n \left\| \mathbf{x}_j^{(t-\tau_{ij})} - \bar{\mathbf{x}}^{(t)} \right\|^2 + 6Ln\pi_0^2(f(\bar{\mathbf{x}}^{(t)}) - f(\mathbf{x}^*)) + 3\pi_0^2n\bar{\sigma}^2 \\ &\stackrel{\text{Lemma D.10}}{\leq} 3n\pi_0^2(L^2\Xi_t + 2Le_t + \bar{\sigma}^2). \end{aligned}$$

where in the last line we have used our previous Lemma D.10. \square

The next proposition is very similar to the Proposition D.2 except that it considers the matrix form instead of the projection onto π .

Proposition D.3. *Under Assumption A and B. Then for $t \geq 0$,*

$$\mathbb{E} \left\| \tilde{\mathbf{W}}(\mathbb{E} \mathbf{G}^{(t)} - \mathbf{G}^{(t)}) \right\|_F^2 \leq 3(L^2\Xi_t + 2Le_t + \bar{\sigma}^2).$$

Proof.

$$\begin{aligned}
& \mathbb{E} \left\| \tilde{\mathbf{W}} (\mathbb{E} \mathbf{G}^{(t)} - \mathbf{G}^{(t)}) \right\|_F^2 \\
&= \sum_{i=1}^n \mathbb{E} \left\| \frac{1}{n} \sum_{j=1}^n (\nabla F(\mathbf{x}_j^{(t-\tau_{ij})}; \xi_j^{(t-\tau_{ij})}) - \nabla f_j(\mathbf{x}_j^{(t-\tau_{ij})})) \right\|^2 \\
&\leq \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \mathbb{E} \left\| \nabla F(\mathbf{x}_j^{(t-\tau_{ij})}; \xi_j^{(t-\tau_{ij})}) - \nabla f_j(\mathbf{x}_j^{(t-\tau_{ij})}) \right\|^2
\end{aligned}$$

The rest of the proof is identical to the one of Proposition D.2. \square

Lemma D.11. (*Descent lemma for convex objective.*) If $\gamma \leq \frac{1}{10Ln\pi_0}$, then

$$r_{t+1} \leq (1 - \frac{\gamma\mu n\pi_0}{2})r_t - \gamma n\pi_0 e_t + 4\gamma Ln\pi_0 \Xi_t + 3\gamma^2 n\pi_0^2 \bar{\sigma}^2.$$

Proof. Expand $r_{t+1} = \mathbb{E} \|\bar{\mathbf{x}}^{(t+1)} - \mathbf{x}^*\|^2$ as follows

$$\begin{aligned}
\mathbb{E} \|\bar{\mathbf{x}}^{(t+1)} - \mathbf{x}^*\|^2 &= \mathbb{E} \|\bar{\mathbf{x}}^{(t)} - \gamma \boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbf{G}^{(t)} - \mathbf{x}^*\|^2 \\
&= \mathbb{E} \|\bar{\mathbf{x}}^{(t)} - \gamma \boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(t)} - \mathbf{x}^* + \gamma \boldsymbol{\pi}^\top \tilde{\mathbf{W}} (\mathbb{E} \mathbf{G}^{(t)} - \mathbf{G}^{(t)})\|^2
\end{aligned}$$

Directly expand it into three terms

$$\begin{aligned}
\mathbb{E} \|\bar{\mathbf{x}}^{(t+1)} - \mathbf{x}^*\|^2 &= \mathbb{E} \left(\|\bar{\mathbf{x}}^{(t)} - \gamma \boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(t)} - \mathbf{x}^*\|^2 + \gamma^2 \|\boldsymbol{\pi}^\top \tilde{\mathbf{W}} (\mathbb{E} \mathbf{G}^{(t)} - \mathbf{G}^{(t)})\|^2 \right. \\
&\quad \left. + \left\langle \bar{\mathbf{x}}^{(t)} - \gamma \boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(t)} - \mathbf{x}^*, \gamma \boldsymbol{\pi}^\top \tilde{\mathbf{W}} (\mathbb{E} \mathbf{G}^{(t)} - \mathbf{G}^{(t)}) \right\rangle \right)
\end{aligned}$$

where the 3rd term is 0 and the second term is bounded in Proposition D.2. The first term is independent of the randomness

$$\begin{aligned}
& \|\bar{\mathbf{x}}^{(t)} - \gamma \boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(t)} - \mathbf{x}^*\|^2 \\
&= \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}^*\|^2 + \gamma^2 \underbrace{\|\boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(t)}\|^2}_{=:T_1} - 2\gamma \underbrace{\langle \boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(t)}, \bar{\mathbf{x}}^{(t)} - \mathbf{x}^* \rangle}_{=:T_2}.
\end{aligned}$$

Since $\boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(t)} = \frac{\pi_0}{n} \sum_{i=1}^n \sum_{j=1}^n \nabla f_i(\mathbf{x}_i^{(t-\tau_{ij})})$, first bound T_1

$$\begin{aligned}
T_1 &= \pi_0^2 \left\| \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n \nabla f_i(\mathbf{x}_i^{(t-\tau_{ij})}) \right\|^2 \\
&= \pi_0^2 \left\| \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n (\nabla f_i(\mathbf{x}_i^{(t-\tau_{ij})}) - \nabla f_i(\bar{\mathbf{x}}^{(t)}) + \nabla f_i(\bar{\mathbf{x}}^{(t)}) - \nabla f_i(\mathbf{x}^*)) \right\|^2 \\
&\leq 2\pi_0^2 \left(\left\| \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n (\nabla f_i(\mathbf{x}_i^{(t-\tau_{ij})}) - \nabla f_i(\bar{\mathbf{x}}^{(t)})) \right\|^2 + \left\| \sum_{i=1}^n (\nabla f_i(\bar{\mathbf{x}}^{(t)}) - \nabla f_i(\mathbf{x}^*)) \right\|^2 \right) \\
&\leq 2\pi_0^2 L^2 \sum_{i=1}^n \sum_{j=1}^n \left\| \mathbf{x}_i^{(t-\tau_{ij})} - \bar{\mathbf{x}}^{(t)} \right\|^2 + 2n\pi_0^2 \sum_{i=1}^n \left\| \nabla f_i(\bar{\mathbf{x}}^{(t)}) - \nabla f_i(\mathbf{x}^*) \right\|^2 \\
&\stackrel{\text{Smoothness (D.4)}}{\leq} 2\pi_0^2 L^2 \sum_{i=1}^n \sum_{j=1}^n \left\| \mathbf{x}_i^{(t-\tau_{ij})} - \bar{\mathbf{x}}^{(t)} \right\|^2 + 4Ln^2\pi_0^2(f(\bar{\mathbf{x}}^{(t)}) - f(\mathbf{x}^*)),
\end{aligned}$$

Using again Lemma D.10 we have

$$T_1 \leq 2L^2n^2\pi_0^2\Xi_t + 4Ln^2\pi_0^2e_t.$$

Then bound T_2

$$\begin{aligned}
T_2 &= \frac{\pi_0}{n} \sum_{i=1}^n \sum_{j=1}^n \langle \nabla f_i(\mathbf{x}_i^{(t-\tau_{ij})}), \bar{\mathbf{x}}^{(t)} - \mathbf{x}^* \rangle \\
&= \frac{\pi_0}{n} \sum_{i=1}^n \sum_{j=1}^n (\langle \nabla f_i(\mathbf{x}_i^{(t-\tau_{ij})}), \bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t-\tau_{ij})} \rangle + \langle \nabla f_i(\mathbf{x}_i^{(t-\tau_{ij})}), \mathbf{x}_i^{(t-\tau_{ij})} - \mathbf{x}^* \rangle) \\
&\geq \frac{\pi_0}{n} \sum_{i=1}^n \sum_{j=1}^n (f_i(\bar{\mathbf{x}}^{(t)}) - f_i(\mathbf{x}_i^{(t-\tau_{ij})}) - \frac{L}{2} \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t-\tau_{ij})}\|^2 \\
&\quad + f_i(\mathbf{x}_i^{(t-\tau_{ij})}) - f_i(\mathbf{x}^*) + \frac{\mu}{2} \|\mathbf{x}_i^{(t-\tau_{ij})} - \mathbf{x}^*\|^2) \\
&= n\pi_0(f(\bar{\mathbf{x}}^{(t)}) - f(\mathbf{x}^*)) + \frac{\pi_0}{n} \sum_{i=1}^n \sum_{j=1}^n (\frac{\mu}{2} \|\mathbf{x}_i^{(t-\tau_{ij})} - \mathbf{x}^*\|^2 - \frac{L}{2} \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t-\tau_{ij})}\|^2) \\
&\geq n\pi_0(f(\bar{\mathbf{x}}^{(t)}) - f(\mathbf{x}^*)) + \frac{\pi_0}{n} \sum_{i=1}^n \sum_{j=1}^n (\frac{\mu}{4} \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}^*\|^2 - \frac{\mu+L}{2} \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t-\tau_{ij})}\|^2) \\
&\stackrel{\text{Lemma D.10}}{\geq} n\pi_0e_t + \frac{n\mu\pi_0}{4}r_t - nL\pi_0\Xi_t
\end{aligned}$$

where the first inequality and the second inequality uses the L -smoothness and μ -convexity of f_i .

Combine both T_1 , T_2 and Proposition D.2 we have

$$\begin{aligned}
r_{t+1} &\leq r_t + \gamma^2 n^2 \pi_0^2 (2L^2 \Xi_t + 4Le_t) - 2\gamma n \pi_0 (e_t + \frac{\mu}{4} r_t - L \Xi_t) \\
&\quad + \gamma^2 n (3L^2 \pi_0^2 \Xi_t + 6L \pi_0^2 e_t + 3\pi_0^2 \bar{\sigma}^2) \\
&= (1 - \frac{\gamma \mu n \pi_0}{2}) r_t - (2\gamma n \pi_0 - 4L \gamma^2 n^2 \pi_0^2 - 6L \gamma^2 n \pi_0^2) e_t \\
&\quad + (2\gamma^2 L^2 n^2 \pi_0^2 + 2\gamma L n \pi_0 + 3L^2 \gamma^2 n \pi_0^2) \Xi_t + 3\gamma^2 n \pi_0^2 \bar{\sigma}^2
\end{aligned}$$

In addition if $\gamma \leq \frac{1}{10Ln\pi_0}$, then we can simplify the coefficient of e_t and Ξ_t

$$\begin{aligned}
4L\gamma^2 n^2 \pi_0^2 + 6L\gamma^2 n \pi_0^2 &\leq \gamma n \pi_0 \\
2\gamma^2 L^2 n^2 \pi_0^2 + 2\gamma L n \pi_0 + 3L^2 \gamma^2 n \pi_0^2 &\leq 4\gamma L n \pi_0
\end{aligned}$$

Then

$$r_{t+1} \leq (1 - \frac{\gamma \mu n \pi_0}{2}) r_t - \gamma n \pi_0 e_t + 4\gamma L n \pi_0 \Xi_t + 3\gamma^2 n \pi_0^2 \bar{\sigma}^2$$

Lemma D.12. For $\gamma \leq \frac{p}{10LmC_1}$ we have

$$\frac{1}{T+1} \sum_{t=0}^T \Xi_t \leq C_1^2 \gamma^2 m^2 \frac{24}{p} \frac{\bar{\sigma}^2}{n} + \frac{80Lm^2}{p^2} C_1^2 \gamma^2 \frac{1}{T+1} \sum_{t=0}^T e_t$$

where C_1 is defined in Definition D.3.

Proof. First bound the consensus distance as follows:

$$\begin{aligned}
n\Xi_t &= \mathbb{E} \|\mathbf{Y}^{(t)} - \bar{\mathbf{Y}}^{(t)}\|_F^2 \leq \mathbb{E} \|(\mathbf{Y}^{(t)} - \bar{\mathbf{Y}}^{(t-m)}) - (\bar{\mathbf{Y}}^{(t)} - \bar{\mathbf{Y}}^{(t-m)})\|_F^2 \\
&\leq \mathbb{E} \|\mathbf{Y}^{(t)} - \bar{\mathbf{Y}}^{(t-m)}\|_F^2
\end{aligned}$$

where the last inequality we use the simple matrix inequality (D.5). For $t \geq m$ unroll to $t-m$.

$$n\Xi_t \leq \mathbb{E} \left\| \mathbf{W}^m \mathbf{Y}^{(t-m)} - \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbf{G}^{(k)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2$$

Separate the stochastic part and deterministic part.

$$\begin{aligned}
n\Xi_t &\leq \left\| \mathbf{W}^m \mathbf{Y}^{(t-m)} - \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2 \\
&\quad + \mathbb{E} \left\| \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} (\mathbb{E} \mathbf{G}^{(k)} - \mathbf{G}^{(k)}) \right\|_F^2 \\
&\leq \left\| \mathbf{W}^m \mathbf{Y}^{(t-m)} - \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2 \\
&\quad + \gamma^2 m \sum_{k=t-m}^{t-1} \mathbb{E} \left\| \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} (\mathbb{E} \mathbf{G}^{(k)} - \mathbf{G}^{(k)}) \right\|_F^2
\end{aligned}$$

Given $\tilde{\mathbf{I}}$ and C_1 in defined in Definition D.3, we know that $\tilde{\mathbf{W}} = \tilde{\mathbf{I}}\tilde{\mathbf{W}}$. Then use (D.6) and Proposition D.3

$$\begin{aligned}
n\Xi_t &\leq \left\| \mathbf{W}^m \mathbf{Y}^{(t-m)} - \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2 \\
&\quad + C_1^2 \gamma^2 m \sum_{k=t-m}^{t-1} \mathbb{E} \left\| \tilde{\mathbf{W}} (\mathbb{E} \mathbf{G}^{(k)} - \mathbf{G}^{(k)}) \right\|_F^2 \\
&\leq \left\| \mathbf{W}^m \mathbf{Y}^{(t-m)} - \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2 \\
&\quad + C_1^2 \gamma^2 m \sum_{k=t-m}^{t-1} 3(L^2 \Xi_k + 2Le_k + \bar{\sigma}^2)
\end{aligned}$$

Separate the first term as

$$\begin{aligned}
n\Xi_t &\leq (1 + \alpha) \left\| \mathbf{W}^m \mathbf{Y}^{(t-m)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2 + (1 + \frac{1}{\alpha}) \left\| \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)} \right\|_F^2 \\
&\quad + C_1^2 \gamma^2 m \sum_{k=t-m}^{t-1} 3(L^2 \Xi_k + 2Le_k + \bar{\sigma}^2) \\
&\leq (1 + \alpha)(1 - p)^{2m} \left\| \mathbf{Y}^{(t-m)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2 + (1 + \frac{1}{\alpha}) \left\| \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)} \right\|_F^2 \\
&\quad + C_1^2 \gamma^2 m \sum_{k=t-m}^{t-1} 3(L^2 \Xi_k + 2Le_k + \bar{\sigma}^2)
\end{aligned}$$

where the first inequality uses $(a + b)^2 \leq (1 + \epsilon)a^2 + (1 + \frac{1}{\epsilon})b^2$ and take $\epsilon = (\frac{2-p}{2-2p})^{2m} - 1$.

$$1 + \frac{1}{\epsilon} \leq 1 + \frac{1-p}{mp} \leq 1 + \frac{1}{mp} \leq \frac{2}{p}.$$

Then by applying our key lemma (Lemma 5.1) we have

$$\begin{aligned} n\Xi_t &\leq \left(1 - \frac{p}{2}\right)^{2m} \left\| \mathbf{Y}^{(t-m)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2 + \frac{2m}{p} C_1^2 \gamma^2 \sum_{k=t-m}^{t-1} \left\| \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)} \right\|_F^2 \\ &\quad + C_1^2 \gamma^2 m \sum_{k=t-m}^{t-1} 3(L^2 \Xi_k + 2Le_k + \bar{\sigma}^2) \end{aligned}$$

Next we bound $\mathbb{E} \left\| \tilde{\mathbf{W}} \mathbf{G}^{(t')} \right\|_F^2$,

$$\begin{aligned} \mathbb{E} \left\| \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)} \right\|_F^2 &= \sum_{i=1}^n \mathbb{E} \left\| \frac{1}{n} \sum_{j=1}^n \nabla f_j(\mathbf{x}_j^{(k-\tau_{ij})}) \right\|^2 \\ &= \sum_{i=1}^n \mathbb{E} \left\| \frac{1}{n} \sum_{j=1}^n (\nabla f_j(\mathbf{x}_j^{(k-\tau_{ij})}) - \nabla f_j(\bar{\mathbf{x}}^{(k)}) + \nabla f_j(\bar{\mathbf{x}}^{(k)}) - \nabla f_j(\mathbf{x}^*)) \right\|^2 \\ &\leq \frac{2}{n} \sum_{i=1}^n \sum_{j=1}^n (\left\| \nabla f_j(\mathbf{x}_j^{(k-\tau_{ij})}) - \nabla f_j(\bar{\mathbf{x}}^{(k)}) \right\|^2 + \left\| \nabla f_j(\bar{\mathbf{x}}^{(k)}) - \nabla f_j(\mathbf{x}^*) \right\|^2) \\ &\leq \frac{2}{n} \sum_{i=1}^n \sum_{j=1}^n (L^2 \left\| \mathbf{x}_j^{(k-\tau_{ij})} - \bar{\mathbf{x}}^{(k)} \right\|^2 + \left\| \nabla f_j(\bar{\mathbf{x}}^{(k)}) - \nabla f_j(\mathbf{x}^*) \right\|^2) \\ &\stackrel{\text{Lemma D.10}}{\leq} 2L^2 n \Xi_k + 2 \sum_{j=1}^n \left\| \nabla f_j(\bar{\mathbf{x}}^{(k)}) - \nabla f_j(\mathbf{x}^*) \right\|^2 \\ &\stackrel{\text{Smoothness (D.4)}}{\leq} 2L^2 n \Xi_k + 4nLe_k. \end{aligned}$$

Then

$$n\Xi_t \leq \left(1 - \frac{p}{2}\right)^{2m} n\Xi_{t-m} + \frac{2m}{p} C_1^2 \gamma^2 \sum_{k=t-m}^{t-1} (2L^2 n \Xi_k + 4nLe_k) + C_1^2 \gamma^2 m \sum_{k=t-m}^{t-1} 3(L^2 \Xi_k + 2Le_k + \bar{\sigma}^2)$$

Then

$$\Xi_t \leq \left(1 - \frac{p}{2}\right)^{2m} \Xi_{t-m} + \frac{2m}{p} C_1^2 \gamma^2 \sum_{k=t-m}^{t-1} (5L^2 \Xi_k + 10Le_k) + 3C_1^2 \gamma^2 m^2 \frac{\bar{\sigma}^2}{n}.$$

Unroll for $t < m$. We can apply similar steps

$$\begin{aligned} n\Xi_t &\leq \mathbb{E} \left\| \mathbf{W}^{(t)} \mathbf{Y}^{(0)} - \gamma \sum_{k=0}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbf{G}^{(k)} - \bar{\mathbf{Y}}^{(0)} \right\|_F^2 = \mathbb{E} \left\| \gamma \sum_{k=0}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbf{G}^{(k)} \right\|_F^2 \\ &\leq C_1^2 \gamma^2 m \sum_{k=0}^{t-1} \mathbb{E} \left\| \tilde{\mathbf{W}} \mathbf{G}^{(k)} \right\|_F^2 \leq 2C_1^2 \gamma^2 m \sum_{k=0}^{t-1} (5L^2 n \Xi_k + 10nLe_k + 3\bar{\sigma}^2) \end{aligned}$$

Merge two parts together and sum over t .

$$\begin{aligned}
\frac{1}{T+1} \sum_{t=0}^T \Xi_t &\leq \left(1 - \frac{p}{2}\right)^{2m} \frac{1}{T+1} \sum_{t=m}^T \Xi_{t-m} + 6C_1^2 \gamma^2 m^2 \frac{\bar{\sigma}^2}{n} \\
&\quad + \frac{2m}{p} C_1^2 \gamma^2 \frac{1}{T+1} \left(\sum_{t=m}^T \sum_{k=t-m}^{t-1} (5L^2 \Xi_k + 10Le_k) + \sum_{t=0}^{m-1} \sum_{k=t-m}^{t-1} (5L^2 \Xi_k + 10Le_k) \right) \\
&\leq \left(1 - \frac{p}{2}\right)^{2m} \frac{1}{T+1} \sum_{t=0}^T \Xi_t + 6C_1^2 \gamma^2 m^2 \frac{\bar{\sigma}^2}{n} + \frac{2m^2}{p} C_1^2 \gamma^2 \frac{1}{T+1} \sum_{t=0}^T (5L^2 \Xi_t + 10Le_t)
\end{aligned}$$

By taking $\gamma \leq \frac{p}{10C_1 m}$, then $\frac{10L^2 m^2}{p} C_1^2 \gamma^2 \leq \frac{p}{4}$.

$$\frac{1}{T+1} \sum_{t=0}^T \Xi_t \leq C_1^2 \gamma^2 m^2 \frac{24}{p} \frac{\bar{\sigma}^2}{n} + \frac{80Lm^2}{p^2} C_1^2 \gamma^2 \frac{1}{T+1} \sum_{t=0}^T e_t. \quad \square$$

Lemma D.13 (Identical to [Koloskova et al., 2020b, Lemma 15]). *For any parameters $r_0 \geq 0, a \geq 0, b \geq 0, c \geq 0$ there exists constant stepsizes $\gamma \leq \frac{1}{c}$ such that*

$$\Psi_T := \frac{r_0}{\gamma(T+1)} + a\gamma + b\gamma^2 \leq 2 \left(\frac{ar_0}{T+1} \right)^{\frac{1}{2}} + 2b^{\frac{1}{3}} \left(\frac{r_0}{T+1} \right)^{\frac{2}{3}} + \frac{cr_0}{T+1}.$$

Theorem D.4. *If $\gamma \leq \frac{p}{30LmC_1}$, then*

$$\frac{1}{T+1} \sum_{t=0}^T (f(\mathbf{x}^{(t)}) - f(\mathbf{x}^*)) \leq 8 \left(\frac{\bar{\sigma}^2 r_0}{n(T+1)} \right)^{\frac{1}{2}} + 2 \left(\frac{16Cm\sqrt{L}\bar{\sigma}r_0}{\sqrt{p}(T+1)} \right)^{\frac{2}{3}} + \frac{30Lm\sqrt{n}Cr_0}{p(T+1)}.$$

where $r_0 = \|\mathbf{x}^0 - \mathbf{x}^*\|^2$ and $C = C(\mathbf{W})$ is defined in Definition D.3.

Proof. Reorganize Lemma D.11 and average over time

$$\frac{1}{T+1} \sum_{t=0}^T e_t \leq \frac{1}{T+1} \sum_{t=0}^T \left(\frac{r_t}{\gamma n \pi_0} - \frac{r_{t+1}}{\gamma n \pi_0} \right) + \frac{4L}{T+1} \sum_{t=0}^T \Xi_t + 3\gamma \pi_0 \bar{\sigma}^2.$$

Combining with Lemma D.12 gives

$$\frac{1}{T+1} \sum_{t=0}^T e_t \leq \frac{1}{T+1} \frac{r_0}{\gamma n \pi_0} + 4L \left(C_1^2 \gamma^2 m^2 \frac{24}{p} \frac{\bar{\sigma}^2}{n} + \frac{80Lm^2}{p^2} C_1^2 \gamma^2 \frac{1}{T+1} \sum_{t=0}^T e_t \right) + 3\gamma \pi_0 \bar{\sigma}^2$$

Select $\gamma \leq \frac{p}{30LmC_1}$ such that $\frac{320L^2}{p^2} \gamma^2 m^2 C_1^2 \leq \frac{1}{2}$

$$\frac{1}{T+1} \sum_{t=0}^T e_t \leq \frac{2}{T+1} \frac{r_0}{\gamma n \pi_0} + 6\gamma \pi_0 \bar{\sigma}^2 + \frac{96L}{p} \gamma^2 m^2 C_1^2 \frac{\bar{\sigma}^2}{n}.$$

Applying Lemma D.13 gives

$$\frac{1}{T+1} \sum_{t=0}^T e_t \leq 40 \left(\frac{\bar{\sigma}^2 r_0}{n(T+1)} \right)^{\frac{1}{2}} + 2 \left(\frac{\sqrt{mL} \bar{\sigma} r_0}{\sqrt{p}(T+1)} \frac{16C_1 \sqrt{m}}{n\pi_0 \sqrt{n}} \right)^{\frac{2}{3}} + \frac{dr_0}{n\pi_0(T+1)}$$

where $d = \max\{\frac{30LmC_1}{p}, 10Ln\pi_0\} = \frac{30LmC_1}{p}$. As in Lemma D.5,

$$C_1 = C \|\mathbf{1}\pi^\top \tilde{\mathbf{I}}\| = Cn\sqrt{\tau_{\max} + 1}\pi_0 \leq Cn\sqrt{n}\pi_0.$$

We can further simplify it as

$$\frac{1}{T+1} \sum_{t=0}^T e_t \leq 40 \left(\frac{\bar{\sigma}^2 r_0}{n(T+1)} \right)^{\frac{1}{2}} + 2 \left(\frac{16Cm\sqrt{L}\bar{\sigma}r_0}{\sqrt{p}(T+1)} \right)^{\frac{2}{3}} + \frac{30Lm\sqrt{n}Cr_0}{p(T+1)} \square$$

D.1.5 Proof of Theorem 5.1 in the strongly convex case

The proof for strongly convex objective follows similar lines as Stich [2019]:

Theorem D.5. Let $a = \frac{\mu n \pi_0}{2}$, $b = \frac{2}{n\pi_0}$, $c = 6\pi_0 \bar{\sigma}^2$, $A = \frac{400L}{p^2} m^2 C_1^2 \bar{\sigma}^2$, and let $\gamma = \frac{1}{s} \leq \frac{1}{aT} \ln \max\{\frac{ba^2 T^2 r_0}{c}, 2\}$, then

$$\frac{1}{W_T} \sum_{t=0}^T w_t e_t + \mu r_{T+1} \leq \tilde{\mathcal{O}} \left(bsr_0 \exp \left[-\frac{a(T+1)}{s} \right] + \frac{c}{a(T+1)} + \frac{A}{a^2(T+1)^2} \right)$$

where $w_t = (1 - \frac{\mu \gamma n \pi_0}{2})^{-(t+1)}$.

Proof. From Lemma D.11 we know that if $\gamma \leq \frac{1}{10Ln\pi_0}$, then

$$r_{t+1} \leq (1 - \frac{\gamma \mu n \pi_0}{2}) r_t - \gamma n \pi_0 e_t + 4\gamma L n \pi_0 \Xi_t + 3\gamma^2 n \pi_0^2 \bar{\sigma}^2.$$

Then

$$e_t \leq \frac{1}{\gamma n \pi_0} (1 - \frac{\mu \gamma n \pi_0}{2}) r_t - \frac{1}{\gamma n \pi_0} r_{t+1} + 4L \Xi_t + 3\gamma \pi_0 \bar{\sigma}^2.$$

Multiply w_t and sum over $t = 0$ to T and divided by W_T

$$\frac{1}{W_T} \sum_{t=0}^T w_t e_t \leq \frac{1}{W_T} \sum_{t=0}^T \left(\frac{1 - \frac{\mu \gamma n \pi_0}{2}}{\gamma n \pi_0} w_t r_t - \frac{w_t}{\gamma n \pi_0} r_{t+1} \right) + \frac{4L}{W_T} \sum_{t=0}^T w_t \Xi_t + 3\gamma \pi_0 \bar{\sigma}^2.$$

Set $(1 - \frac{\mu\gamma n\pi_0}{2})w_{t+1} = w_t$, then

$$\frac{1}{W_T} \sum_{t=0}^T w_t e_t \leq \frac{1}{W_T} \left(\frac{1 - \frac{\mu\gamma n\pi_0}{2}}{\gamma n\pi_0} w_0 r_0 - \frac{1 - \frac{\mu\gamma n\pi_0}{2}}{\gamma n\pi_0} w_{T+1} r_{T+1} \right) + \frac{4L}{W_T} \sum_{t=0}^T w_t \Xi_t + 3\gamma\pi_0 \bar{\sigma}^2.$$

Then using Lemma D.12 we have

$$\begin{aligned} & \frac{1}{W_T} \sum_{t=0}^T w_t e_t + \frac{1 - \frac{\mu\gamma n\pi_0}{2}}{\gamma n\pi_0 W_T} w_{T+1} r_{T+1} \\ & \leq \frac{1}{W_T} \frac{1 - \frac{\mu\gamma n\pi_0}{2}}{\gamma n\pi_0} w_0 r_0 + 4L \left(\frac{80C_1^2 L m^2}{p^2} \gamma^2 \frac{1}{W_T} \sum_{t'=0}^T w_t e_{t'} + \frac{24}{p} \gamma^2 m^2 C_1^2 \frac{\bar{\sigma}^2}{n} \right) + 3\gamma\pi_0 \bar{\sigma}^2 \end{aligned}$$

By taking $\gamma \leq \frac{p}{30LmC_1}$ we have $\frac{320L^2 m^2 C_1^2 \gamma^2}{p^2} \leq \frac{1}{2}$, then

$$\frac{1}{W_T} \sum_{t=0}^T w_t e_t + \frac{1 - \frac{\mu\gamma n\pi_0}{2}}{\gamma n\pi_0 W_T} 2w_{T+1} r_{T+1} \leq \frac{1}{W_T} \frac{1 - \frac{\mu\gamma n\pi_0}{2}}{\gamma n\pi_0} 2w_0 r_0 + 6\gamma\pi_0 \bar{\sigma}^2 + \frac{400L}{p^2} \gamma^2 m^2 C_1^2 \bar{\sigma}^2$$

Since $W_T \geq w_T = (1 - \frac{\mu\gamma n\pi_0}{2})^{-(T+1)}$ and $W_T \leq \frac{2w_T}{\mu\gamma n\pi_0}$

$$\begin{aligned} \frac{1}{W_T} \sum_{t=0}^T w_t e_t + \mu r_{T+1} & \leq \frac{(1 - \frac{\mu\gamma n\pi_0}{2})^{T+1}}{\gamma n\pi_0} 2w_0 r_0 + 6\gamma\pi_0 \bar{\sigma}^2 + \frac{400L}{p^2} \gamma^2 m^2 C_1^2 \bar{\sigma}^2 \\ & \leq \frac{e^{-\frac{\mu\gamma n\pi_0}{2}(T+1)}}{\gamma n\pi_0} 2w_0 r_0 + 6\gamma\pi_0 \bar{\sigma}^2 + \frac{400L}{p^2} \gamma^2 m^2 C_1^2 \bar{\sigma}^2 \end{aligned}$$

Let $a = \frac{\mu n\pi_0}{2}$, $b = \frac{2}{n\pi_0}$, $c = 6\pi_0 \bar{\sigma}^2$, $A = \frac{400L}{p^2} m^2 C_1^2 \bar{\sigma}^2$, then

$$\frac{1}{W_T} \sum_{t=0}^T w_t e_t + \mu r_{T+1} \leq \frac{br_0}{\gamma} \exp[-a\gamma(T+1)] + c\gamma + A\gamma^2$$

Tuning stepsize. Let $\gamma = \frac{1}{d} \leq \frac{1}{dT} \ln \max\{\frac{ba^2 T^2 r_0}{c}, 2\}$, then

$$\frac{1}{W_T} \sum_{t=0}^T w_t e_t + \mu r_{T+1} \leq \tilde{O} \left(bsr_0 \exp\left[-\frac{a(T+1)}{s}\right] + \frac{c}{a(T+1)} + \frac{A}{a^2(T+1)^2} \right). \quad \square$$

D.1.6 Proof of Theorem 5.1 in the non-convex case

Let $\bar{\mathbf{x}}^{(t)} := (\boldsymbol{\pi}^\top \mathbf{Y}^{(t)})^\top$ and $\bar{\mathbf{Y}}^{(t)} := \mathbf{1}\boldsymbol{\pi}^\top \mathbf{Y}^{(t)}$. Let f^* be the optimal objective value at critical points. We can define the following iterates

1. $r_t := \mathbb{E} f(\bar{\mathbf{x}}^{(t)}) - f^*$ is the *expected function suboptimality*.
2. $e_t := \|\nabla f(\bar{\mathbf{x}}^{(t)})\|^2$
3. $\Xi_t := \frac{1}{n} \|\bar{\mathbf{Y}}^{(t)} - \mathbf{Y}^{(t)}\|_F^2$ is the *consensus distance*.

where the expectation is taken with respect to $\boldsymbol{\xi}^{(t)} \in \mathbb{R}^n$ the randomness across all workers at time t . Note that Lemma D.10 still holds.

Proposition D.6 and Proposition D.7 bound the stochastic noise of the gradient.

Proposition D.6. *Under Assumption B, we have*

$$\mathbb{E} \|\boldsymbol{\pi}^\top \tilde{\mathbf{W}}(\mathbf{G}^{(t)} - \mathbb{E} \mathbf{G}^{(t)})\|^2 \leq n\pi_0^2 \bar{\sigma}^2. \quad (\text{D.8})$$

Proof. Denote $\mathbb{E} = \mathbb{E}_{\boldsymbol{\xi}}$. Use Cauchy-Schwartz inequality Equation (D.3)

$$\begin{aligned} \mathbb{E} \|\boldsymbol{\pi}^\top \tilde{\mathbf{W}}(\mathbf{G}^{(t)} - \mathbb{E} \mathbf{G}^{(t)})\|^2 &= \mathbb{E} \left\| \frac{\pi_0}{n} \sum_{i=1}^n \sum_{j=1}^n (\nabla F_j(\mathbf{x}_j^{(t-\tau_{ij})}; \xi_j^{(t-\tau_{ij})}) - \nabla f_j(\mathbf{x}_j^{(t-\tau_{ij})})) \right\|^2 \\ &\leq \frac{\pi_0^2}{n} \sum_{i=1}^n \mathbb{E} \left\| \sum_{j=1}^n \nabla F_j(\mathbf{x}_j^{(t-\tau_{ij})}; \xi_j^{(t-\tau_{ij})}) - \nabla f_j(\mathbf{x}_j^{(t-\tau_{ij})}) \right\|^2 \end{aligned}$$

Now the randomness inside the norm are independent

$$\mathbb{E} \|\boldsymbol{\pi}^\top \tilde{\mathbf{W}}(\mathbf{G}^{(t)} - \mathbb{E} \mathbf{G}^{(t)})\|^2 \mathbb{E} \|\boldsymbol{\pi}^\top \tilde{\mathbf{W}}(\mathbf{G}^{(t)} - \mathbb{E} \mathbf{G}^{(t)})\|^2 \leq n\pi_0^2 \bar{\sigma}^2. \quad \square$$

Proposition D.7. *Under Assumption B, we have*

$$\mathbb{E} \|\tilde{\mathbf{W}}(\mathbf{G}^{(t)} - \mathbb{E} \mathbf{G}^{(t)})\|_F^2 \leq \bar{\sigma}^2. \quad (\text{D.9})$$

Next we establish the recursion of r_t

Lemma D.14 (Descent lemma for non-convex case). *Under Assumption A and B. Let $\gamma \leq \frac{1}{8Ln\pi_0}$, then*

$$r_{t+1} \leq r_t - \frac{\gamma n \pi_0}{4} e_t + 2\gamma L^2 n \pi_0 \Xi_t + 2\gamma^2 L n \pi_0^2 \bar{\sigma}^2.$$

Proof. Since f is L -smooth,

$$\begin{aligned} \mathbb{E} f(\bar{\mathbf{x}}^{(t+1)}) &= \mathbb{E} f(\bar{\mathbf{x}}^{(t)} - \gamma \boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbf{G}^{(t)}) \\ &\leq f(\bar{\mathbf{x}}^{(t)}) - \underbrace{\gamma \langle \nabla f(\bar{\mathbf{x}}^{(t)}), \boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(t)} \rangle}_{:=T_1} + \frac{\gamma^2 L}{2} \underbrace{\mathbb{E} \|\boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbf{G}^{(t)}\|^2}_{:=T_2} \end{aligned}$$

The first-order term T_1 has a lower bound

$$\begin{aligned}
T_1 &= n\pi_0 \langle \nabla f(\bar{\mathbf{x}}^{(t)}), \frac{1}{n\pi_0} \boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(t)} \rangle \\
&= n\pi_0 \left(\|\nabla f(\bar{\mathbf{x}}^{(t)})\|^2 + \langle \nabla f(\bar{\mathbf{x}}^{(t)}), \frac{1}{n\pi_0} \boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(t)} - \nabla f(\bar{\mathbf{x}}^{(t)}) \rangle \right) \\
&\geq n\pi_0 \left(\frac{1}{2} \|\nabla f(\bar{\mathbf{x}}^{(t)})\|^2 - \frac{1}{2} \left\| \frac{1}{n\pi_0} \boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(t)} - \nabla f(\bar{\mathbf{x}}^{(t)}) \right\|^2 \right) \\
&= n\pi_0 \left(\frac{1}{2} e_t - \frac{1}{2n^4} \left\| \sum_{i=1}^n \sum_{j=1}^n (\nabla f_j(\mathbf{x}_j^{(t-\tau_{ij})}) - \nabla f_j(\bar{\mathbf{x}}^{(t)})) \right\|^2 \right) \\
&\geq n\pi_0 \left(\frac{1}{2} e_t - \frac{L^2}{2n^2} \sum_{i=1}^n \sum_{j=1}^n \|\mathbf{x}_j^{(t-\tau_{ij})} - \bar{\mathbf{x}}^{(t)}\|^2 \right) \\
&\geq n\pi_0 \left(\frac{1}{2} e_t - \frac{L^2}{2} \Xi_t \right)
\end{aligned}$$

as $a^2 - \langle a, b \rangle \geq \frac{a^2}{2} - \frac{b^2}{2}$ for $a, b \geq 0$.

On the other hand, separate the stochastic part and deterministic part of T_2 we have

$$T_2 \leq 2\mathbb{E} \|\boldsymbol{\pi}^\top \tilde{\mathbf{W}} (\mathbf{G}^{(t)} - \mathbb{E} \mathbf{G}^{(t)})\|^2 + 2\|\boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(t)}\|^2.$$

Under Assumption B and Proposition D.6, we know the first term

$$\mathbb{E} \|\boldsymbol{\pi}^\top \tilde{\mathbf{W}} (\mathbf{G}^{(t)} - \mathbb{E} \mathbf{G}^{(t)})\|^2 \leq n\pi_0^2 \bar{\sigma}^2.$$

Consider the second term

$$\begin{aligned}
\|\boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(t)}\|^2 &= \left\| \frac{\pi_0}{n} \sum_{i=1}^n \sum_{j=1}^n \nabla f_j(\mathbf{x}_j^{(t-\tau_{ij})}) \right\|^2 \\
&= n^2 \pi_0^2 \left\| \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \nabla f_j(\mathbf{x}_j^{(t-\tau_{ij})}) - \nabla f(\bar{\mathbf{x}}^{(t)}) + \nabla f(\bar{\mathbf{x}}^{(t)}) \right\|^2 \\
&\leq 2n^2 \pi_0^2 \left\| \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (\nabla f_j(\mathbf{x}_j^{(t-\tau_{ij})}) - \nabla f_j(\bar{\mathbf{x}}^{(t)})) \right\|^2 + 2n^2 \pi_0^2 \|\nabla f(\bar{\mathbf{x}}^{(t)})\|^2 \\
&\leq 2\pi_0^2 \sum_{i=1}^n \sum_{j=1}^n \left\| \nabla f_j(\mathbf{x}_j^{(t-\tau_{ij})}) - \nabla f_j(\bar{\mathbf{x}}^{(t)}) \right\|^2 + 2n^2 \pi_0^2 \|\nabla f(\bar{\mathbf{x}}^{(t)})\|^2
\end{aligned}$$

Combine Assumption B we have

$$\|\boldsymbol{\pi}^\top \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(t)}\|^2 \leq 2n^2 \pi_0^2 (L^2 \Xi_t + e_t).$$

Therefore, the T_2 can be bounded as follows

$$T_2 \leq 4n^2 \pi_0^2 \left(\frac{\bar{\sigma}^2}{n} + L^2 \Xi_t + e_t \right). \quad (\text{D.10})$$

Gathering everything together

$$\begin{aligned} r_{t+1} &\leq r_t - \frac{\gamma n \pi_0}{2} (e_t - L^2 \Xi_t) + 2\gamma^2 L n^2 \pi_0^2 \left(\frac{\bar{\sigma}^2}{n} + L^2 \Xi_t + e_t \right) \\ &\leq r_t - \frac{\gamma n \pi_0}{2} (1 - 4\gamma L n \pi_0) e_t + \gamma L^2 n \pi_0 (1 + 2\gamma L n \pi_0) \Xi_t + 2\gamma^2 L n \pi_0^2 \bar{\sigma}^2 \end{aligned}$$

Let $\gamma \leq \frac{1}{8Ln\pi_0}$, then

$$r_{t+1} \leq r_t - \frac{\gamma n \pi_0}{4} e_t + 2\gamma L^2 n \pi_0 \Xi_t + 2\gamma^2 L n \pi_0^2 \bar{\sigma}^2. \quad \square$$

Next we bound the consensus distance

Lemma D.15 (Bounded consensus distance). *Under Assumption B,*

$$\frac{1}{T+1} \sum_{t=0}^T \Xi_t \leq \frac{16C^2 m^2}{p^2} \gamma^2 \bar{\sigma}^2 + \frac{16C^2 m^2}{p^2} \gamma^2 \frac{1}{T+1} \sum_{t=0}^T e_k.$$

Proof. First bound the consensus distance by inserting $\bar{\mathbf{Y}}^{(t-m)}$

$$\begin{aligned} n\Xi_t &= \mathbb{E} \|\bar{\mathbf{Y}}^{(t)} - \mathbf{Y}^{(t)}\|_F^2 \leq \mathbb{E} \|(\bar{\mathbf{Y}}^{(t)} - \bar{\mathbf{Y}}^{(t-m)}) - (\mathbf{Y}^{(t)} - \bar{\mathbf{Y}}^{(t-m)})\|_F^2 \\ &\leq \mathbb{E} \|\mathbf{Y}^{(t)} - \bar{\mathbf{Y}}^{(t-m)}\|_F^2 \end{aligned}$$

where we used $\|A - \bar{A}\|_F^2 = \sum_{i=1}^n \|\mathbf{a}_i - \bar{\mathbf{a}}\|^2 \leq \sum_{i=1}^n \|\mathbf{a}_i\|^2 = \|A\|_F^2$.

For $t \geq m$ unroll $\mathbf{Y}^{(t)}$ until $t - m$.

$$n\Xi_t \leq \mathbb{E} \left\| \mathbf{W}^m \mathbf{Y}^{(t-m)} - \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbf{G}^{(k)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2$$

Separate stochastic part and deterministic part

$$\begin{aligned} n\Xi_t &\leq \left\| \mathbf{W}^m \mathbf{Y}^{(t-m)} - \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2 \\ &\quad + \mathbb{E} \left\| \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} (\mathbb{E} \mathbf{G}^{(k)} - \mathbf{G}^{(k)}) \right\|_F^2 \end{aligned}$$

then let C_1^2 defined in Definition D.3 and use $\|AB\|_F^2 \leq \|A\|_F^2 \|B\|^2$ and (D.9)

$$\begin{aligned} n\Xi_t &\leq \left\| \mathbf{W}^m \mathbf{Y}^{(t-m)} - \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2 \\ &\quad + C_1^2 \gamma^2 m \sum_{k=t-m}^{t-1} \mathbb{E} \left\| \tilde{\mathbf{W}} (\mathbb{E} \mathbf{G}^{(k)} - \mathbf{G}^{(k)}) \right\|_F^2 \\ &\leq \left\| \mathbf{W}^m \mathbf{Y}^{(t-m)} - \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2 + C_1^2 \gamma^2 m^2 \bar{\sigma}^2 \end{aligned}$$

Apply Cauchy-Schwartz inequality with $\alpha > 0$

$$n\Xi_t \leq (1 + \alpha) \left\| \mathbf{W}^m \mathbf{Y}^{(t-m)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2 + (1 + \frac{1}{\alpha}) \left\| \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)} \right\|_F^2 + C_1^2 \gamma^2 m^2 \bar{\sigma}^2$$

Applying Lemma 5.1 to the first term

$$n\Xi_t \leq (1 + \alpha)(1 - p)^{2m} \left\| \mathbf{Y}^{(t-m)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2 + (1 + \frac{1}{\alpha}) \left\| \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)} \right\|_F^2 + C_1^2 \gamma^2 m^2 \bar{\sigma}^2$$

Take $\alpha = (\frac{2-p}{2-2p})^{2m} - 1 = (1 + \frac{p}{2-2p})^{2m} - 1 \geq \frac{mp}{1-p}$ and use

$$1 + \frac{1}{\alpha} \leq 1 + \frac{1-p}{mp} \leq 1 + \frac{1}{mp} \leq \frac{2}{p},$$

then use $\|AB\|_F^2 \leq \|A\|_F^2 \|B\|^2$

$$\begin{aligned} n\Xi_t &\leq \left(1 - \frac{p}{2}\right)^{2m} \left\| \mathbf{Y}^{(t-m)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2 + \frac{2}{p} \left\| \gamma \sum_{k=t-m}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)} \right\|_F^2 + C_1^2 \gamma^2 m^2 \bar{\sigma}^2 \\ &\leq \left(1 - \frac{p}{2}\right)^{2m} \left\| \mathbf{Y}^{(t-m)} - \bar{\mathbf{Y}}^{(t-m)} \right\|_F^2 + \frac{2C_1^2 m}{p} \gamma^2 \sum_{k=t-m}^{t-1} \left\| \tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)} \right\|_F^2 + C_1^2 \gamma^2 m^2 \bar{\sigma}^2. \end{aligned}$$

where the second term can be expanded by

$$\begin{aligned}
\|\tilde{\mathbf{W}} \mathbb{E} \mathbf{G}^{(k)}\|_F^2 &= \sum_{i=1}^n \left\| \frac{1}{n} \sum_{j=1}^n \nabla f_j(\mathbf{x}_j^{(k-\tau_{ij})}) \right\|^2 \\
&= \sum_{i=1}^n \left\| \frac{1}{n} \sum_{j=1}^n \nabla f_j(\mathbf{x}_j^{(k-\tau_{ij})}) - \nabla f(\bar{\mathbf{x}}^{(k)}) + \nabla f(\bar{\mathbf{x}}^{(k)}) \right\|^2 \\
&\leq 2 \sum_{i=1}^n \left\| \frac{1}{n} \sum_{j=1}^n (\nabla f_j(\mathbf{x}_j^{(k-\tau_{ij})}) - \nabla f_j(\bar{\mathbf{x}}^{(k)})) \right\|^2 + 2n \left\| \nabla f(\bar{\mathbf{x}}^{(k)}) \right\|^2 \\
&\leq \frac{2}{n} \sum_{i=1}^n \sum_{j=1}^n \left\| \nabla f_j(\mathbf{x}_j^{(k-\tau_{ij})}) - \nabla f_j(\bar{\mathbf{x}}^{(k)}) \right\|^2 + 2n \left\| \nabla f(\bar{\mathbf{x}}^{(k)}) \right\|^2 \\
&\leq 2nL^2\Xi_k + 2ne_k
\end{aligned}$$

Combine and reduce the n on both sides

$$\Xi_t \leq \left(1 - \frac{p}{2}\right)^{2m} \Xi_{t-m} + 2C_1^2 m^2 \gamma^2 \frac{\bar{\sigma}^2}{n} + \frac{4C_1^2 m}{p} \gamma^2 \sum_{k=t-m}^{t-1} (L^2 \Xi_k + e_k).$$

Unroll for $t < m$. For $t < m$, we can apply similar steps

$$\begin{aligned}
n\Xi_t &\leq \mathbb{E} \left\| \mathbf{W}^{(t)} \mathbf{Y}^{(0)} - \gamma \sum_{k=0}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbf{G}^{(k)} - \bar{\mathbf{Y}}^{(0)} \right\|_F^2 = \mathbb{E} \left\| \gamma \sum_{k=0}^{t-1} \mathbf{W}^{t-1-k} \tilde{\mathbf{W}} \mathbf{G}^{(k)} \right\|_F^2 \\
&\leq C_1^2 \gamma^2 m \sum_{k=0}^{t-1} \mathbb{E} \left\| \tilde{\mathbf{W}} \mathbf{G}^{(k)} \right\|_F^2 \leq 2C_1^2 m \gamma^2 \sum_{k=0}^{t-1} (\bar{\sigma}^2 + nL^2\Xi_k + ne_k).
\end{aligned}$$

Finally, sum over t

$$\begin{aligned}
\frac{1}{T+1} \sum_{t=0}^T \Xi_t &\leq \left(1 - \frac{p}{2}\right)^{2m} \frac{1}{T+1} \sum_{t=m}^T \Xi_{t-m} + 2C_1^2 m^2 \gamma^2 \frac{\bar{\sigma}^2}{n} \\
&\quad + \frac{4C_1^2 m}{p} \gamma^2 \frac{1}{T+1} \left(\sum_{t=m}^T \sum_{k=t-m}^{t-1} (L^2 \Xi_k + e_k) + \sum_{t=0}^{m-1} \sum_{k=0}^{t-1} (L^2 \Xi_k + e_k) \right) \\
&\leq \left(1 - \frac{p}{2}\right)^{2m} \frac{1}{T+1} \sum_{t=0}^T \Xi_t + 2C_1^2 m^2 \gamma^2 \frac{\bar{\sigma}^2}{n} + \frac{4C_1^2 m^2}{p} \frac{\gamma^2}{T+1} \sum_{t=0}^T (L^2 \Xi_k + e_k).
\end{aligned}$$

by taking $\gamma \leq \frac{p}{4C_1Lm}$ we have $\frac{4C_1^2m^2}{p}\gamma^2L^2 \leq \frac{p}{4}$, then rearrange the all of the Ξ terms

$$\frac{1}{T+1} \sum_{t=0}^T \Xi_t \leq \frac{16C_1^2m^2}{p} \frac{\bar{\sigma}^2}{n} \gamma^2 + \frac{16C_1^2m^2}{p^2} \gamma^2 \frac{1}{T+1} \sum_{t=0}^T e_k \quad \square$$

We can use the lemmas for recursion and the descent in the consensus distance to conclude the following theorem.

Theorem D.8. *Under Assumption A and Assumption B. For $\gamma \leq \frac{p}{16C_1Lm}$*

$$\frac{1}{T+1} \sum_{t=0}^T \|\nabla f(\bar{\mathbf{x}}^{(t)})\|^2 \leq 16 \left(\frac{2L\bar{\sigma}^2r_0}{n(T+1)} \right)^{\frac{1}{2}} + 2 \left(\frac{16CLm\bar{\sigma}}{\sqrt{p}} \frac{8r_0}{T+1} \right)^{\frac{2}{3}} + \frac{16C_1Lm}{p} \frac{r_0}{T+1}$$

where $C = C(\mathbf{W})$ is defined in Definition D.3 and $r_0 = f(\mathbf{x}^{(0)}) - f^*$. Alternatively, for any target accuracy ϵ , $\frac{1}{T+1} \sum_{t=0}^T \|\nabla f(\bar{\mathbf{x}}^{(t)})\|^2 \leq \epsilon$ after

$$\mathcal{O} \left(\frac{\bar{\sigma}^2}{n\epsilon^2} + \frac{Cm\bar{\sigma}}{\sqrt{p}\epsilon^{3/2}} + \frac{C_1m}{p\epsilon} \right) Lr_0$$

iterations.

Remark 16. For gossip averaging [Koloskova et al. \[2020b\]](#), the rate with $\zeta^2 = 0$ is

$$\mathcal{O} \left(\frac{\bar{\sigma}^2}{n\epsilon^2} + \frac{\sqrt{m}\bar{\sigma}}{\sqrt{p}\epsilon^{3/2}} + \frac{m}{p\epsilon} \right) Lr_0.$$

Proof. From Lemma D.14 we know that for $\gamma \leq \frac{1}{8Ln\pi_0}$

$$r_{t+1} \leq r_t - \frac{\gamma n \pi_0}{4} e_t + 2\gamma L^2 n \pi_0 \Xi_t + 2\gamma^2 L n \pi_0^2 \bar{\sigma}^2.$$

Rearrange the terms and average over t

$$\begin{aligned} \frac{1}{T+1} \sum_{t=0}^T e_t &\leq \frac{1}{T+1} \sum_{t=0}^T \left(\frac{4r_t}{\gamma n \pi_0} - \frac{4r_{t+1}}{\gamma n \pi_0} \right) + \frac{8L^2}{T+1} \sum_{t=0}^T \Xi_t + 8L\pi_0\gamma\bar{\sigma}^2 \\ &\leq \frac{1}{T+1} \frac{4r_0}{\gamma n \pi_0} + \frac{8L^2}{T+1} \sum_{t=0}^T \Xi_t + 8L\pi_0\gamma\bar{\sigma}^2 \end{aligned}$$

On the other hand, from Lemma D.15 for $\gamma \leq \frac{p}{4C_1Lm}$ we have

$$\frac{1}{T+1} \sum_{t=0}^T \Xi_t \leq \frac{16C_1^2m^2}{p} \frac{\bar{\sigma}^2}{n} \gamma^2 + \frac{16C_1^2m^2}{p^2} \gamma^2 \frac{1}{T+1} \sum_{t=0}^T e_k.$$

Then

$$\frac{1}{T+1} \sum_{t=0}^T e_t \leq \frac{1}{T+1} \frac{4r_0}{\gamma n \pi_0} + 8L^2 \frac{16C_1^2 m^2}{p^2} \gamma^2 \left(\frac{p\bar{\sigma}^2}{n} + \frac{1}{T+1} \sum_{t=0}^T e_k \right) + 8L\pi_0 \gamma \bar{\sigma}^2$$

By taking $\gamma \leq \frac{p}{16C_1 L m}$ such that $8L^2 \frac{16C_1^2 m^2}{p^2} \gamma^2 \leq \frac{1}{2}$, then

$$\frac{1}{T+1} \sum_{t=0}^T e_t \leq \frac{1}{T+1} \frac{8r_0}{\gamma n \pi_0} + 16L\pi_0 \gamma \bar{\sigma}^2 + \frac{16^2 L^2 C_1^2 m^2}{np} \gamma^2 \bar{\sigma}^2$$

Then applying Lemma D.13 we have

$$\frac{1}{T+1} \sum_{t=0}^T e_t \leq 32 \left(\frac{L\bar{\sigma}^2 r_0}{n(T+1)} \right)^{\frac{1}{2}} + 2 \left(\frac{16C_1 L m \bar{\sigma}}{\sqrt{np}} \frac{8r_0}{n\pi_0(T+1)} \right)^{\frac{2}{3}} + \frac{dr_0}{T+1}$$

where $d = \max\{\frac{16C_1 L m}{p}, 8L n \pi_0\} = \frac{16C_1 L m}{p}$. As in Lemma D.5,

$$C_1 = C \|\mathbf{1} \boldsymbol{\pi}^\top \tilde{\mathbf{I}}\| = C n \sqrt{\tau_{\max} + 1} \pi_0 \leq C n \sqrt{n} \pi_0.$$

We can further simplify it as

$$\frac{1}{T+1} \sum_{t=0}^T e_t \leq 32 \left(\frac{L\bar{\sigma}^2 r_0}{n(T+1)} \right)^{\frac{1}{2}} + 2 \left(\frac{16C L m \bar{\sigma}}{\sqrt{p}} \frac{8r_0}{T+1} \right)^{\frac{2}{3}} + \frac{dr_0}{T+1}. \quad \square$$

D.2 Detailed experimental setup

D.2.1 Cifar-10

Table D.1

D.2.2 ImageNet

Table D.2

D.2.3 BERT finetuning

Table D.3

D.2.4 Random quadratics

We generate quadratics $\frac{1}{n} \sum_{i=1}^n f_i(\mathbf{x})$ of $\mathbf{x} \in \mathbb{R}^d$ where

$$f_i(\mathbf{x}) = \|\mathbf{A}_i \mathbf{x} + \mathbf{b}_i\|_2^2.$$

Table D.1 Default experimental settings for Cifar-10/VGG-11

Dataset	Cifar-10 [Krizhevsky et al.]
Data augmentation	random horizontal flip and random 32×32 cropping
Architecture	VGG-11 [Krizhevsky, 2012]
Training objective	cross entropy
Evaluation objective	top-1 accuracy
Number of workers	16
Topology	SGP: time-varying exponential, RelaySGD: double binary trees, baselines: best of ring or double binary trees
Gossip weights	Metropolis-Hastings (1/3 for ring)
Data distribution	Heterogeneous, not shuffled, according to Dirichlet sampling procedure from Lin et al. [2021b]
Batch size	32 patches per worker
Momentum	0.9 (Nesterov)
Learning rate	Tuned c.f. § D.3.1
LR decay	/10 at epoch 150 and 180
LR warmup	Step-wise linearly within 5 epochs, starting from 0
# Epochs	200
Weight decay	10^{-4}
Normalization scheme	no normalization layer
Repetitions	3, with varying seeds
Reported metric	Worst result of any worker of the worker's mean test accuracy over the last 5 epochs

Here the local Hessian $\mathbf{A}_i \in \mathbb{R}^{d \times d}$ control the shape of worker i 's local objective functions and the offset $\mathbf{b}_i \in \mathbb{R}^d$ allows for shifting the worker's optimum. The generation procedure is as follows:

1. Sample $\mathbf{A}_i \in \mathbb{R}^{d \times d}$ from an i.i.d. element-wise standard normal distribution, independently for each worker.
2. Control the smoothness L and strong-convexity constant μ . Decompose $\mathbf{A}_i = \mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^\top$ using Singular Value Decomposition, and replace \mathbf{A}_i with $\mathbf{A}_i \leftarrow \mathbf{U}_i \tilde{\mathbf{S}}_i \mathbf{V}_i^\top$, where $\tilde{\mathbf{S}}_i \in \mathbb{R}^{d \times d}$ is a diagonal matrix with diagonal entries $[\mu, \frac{d-2}{d-1}\mu + \frac{1}{d-1}L, \dots, L]$.
3. Control the heterogeneity ζ_2 by shifting worker's optima into random directions.
 - (a) Sample random directions $\mathbf{d}_i \in \mathbb{R}^d$ from an i.i.d. element-wise standard normal distributions, independently for each worker.
 - (b) Instantiate a scalar $s \leftarrow 1$ and optimize it using binary search:
 - (c) Move local optima by $s\mathbf{d}_i$ by setting $\mathbf{b}_i \leftarrow \mathbf{A}_i s \mathbf{d}_i$.
 - (d) Move all optima $\mathbf{b}_i \leftarrow \mathbf{b}_i - \mathbf{A}_i \mathbf{x}^*$ such that the global optimum \mathbf{x}^* remains at zero.
 - (e) Evaluate $\zeta^2 = \frac{1}{n} \sum_{i=1}^n \|\nabla f_i(\mathbf{x}^*)\|_2^2$ and adjust the scale factor s until ζ^2 is as desired. Repeat from step (c).
4. Control the initial distance to the optimum r_0 . Sample a random vector for the optimum \mathbf{x}^* from an i.i.d. element-wise normal distribution and scale it to have norm r_0 . Shift all worker's optima in this direction by updating $\mathbf{b}_i \leftarrow \mathbf{b}_i + \mathbf{A}_i \mathbf{x}^*$.

Table D.2 Default experimental settings for ImageNet

Dataset	ImageNet [Deng et al., 2009]
Data augmentation	random resized crop (224×224), random horizontal flip
Architecture	ResNet-20-EvoNorm [Lin et al., 2021b; Liu et al., 2020]
Training objective	cross entropy
Evaluation objective	top-1 accuracy
Number of workers	16
Topology	SGP: time-varying exponential, RelaySGD: double binary trees, baselines: best of ring or double binary trees
Gossip weights	Metropolis-Hastings (1/3 for ring)
Data distribution	Heterogeneous, not shuffled, according to Dirichlet sampling procedure from Lin et al. [2021b]
Batch size	32 patches per worker
Momentum	0.9 (Nesterov)
Learning rate	based on centralized training (scaled to $0.1 \times \frac{32 \cdot 16}{256}$)
LR decay	/10 at epoch 30, 60, 80
LR warmup	Step-wise linearly within 5 epochs, starting from 0.1
# Epochs	90
Weight decay	10^{-4}
Normalization layer	EvoNorm [Liu et al., 2020]
Repetitions	Just one
Reported metric	Mean of all worker’s test accuracies over the last 5 epochs

D.3 Hyper-parameters and tuning details

D.3.1 Cifar-10

For our image classification experiments on Cifar-10, we have independently tuned learning rates for each algorithm, at each data heterogeneity level α , and separately for SGD with and without momentum. We followed the following procedure:

1. We found an appropriate learning rate for centralized (all-reduce) training (by using the procedure below)
2. Start the search from this learning rate. For RelaySGD, we apply a correction computed as in § D.4.1.
3. Grid-search the learning rate by multiplying and dividing by powers of two. Try larger and smaller learning rates, until the best result found so far is sandwiched between two learning rates that gave worse results.
4. Repeat the experiment with 3 random seeds.
5. If any of those replicas diverged, reduce the learning rate by a factor two until it does.

For the experiments in Table 5.1, we used the learning rates listed in Table D.4.

D.3.2 ImageNet

Due to the high resource requirements, we did not tune the learning rate for our ImageNet experiments. We identified a suitable learning rate based on prior work, and used this for all

Table D.3 Default experimental settings for BERT finetuning

Dataset	AG News [Zhang et al., 2015]
Data augmentation	none
Architecture	DistilBERT [Sanh et al., 2019]
Training objective	cross entropy
Evaluation objective	top-1 accuracy
Number of workers	16
Topology	restricted to a ring (chain for RelaySGD)
Gossip weights	Metropolis-Hastings (1/3 for ring)
Data distribution	Heterogeneous, not shuffled, according to Dirichlet sampling procedure from Lin et al. [2021b]
Batch size	32 patches per worker
Adam β_1	0.9
Adam β_2	0.999
Adam ϵ	10^{-8}
Learning rate	Tuned c.f. § D.3.3
LR decay	constant learning rate
LR warmup	no warmup
# Epochs	5
Weight decay	0
Normalization layer	LayerNorm [Ba et al., 2016]
Repetitions	3, with varying seeds
Reported metric	Mean of all worker's test accuracies over the last 5 epochs

experiments. For RelaySGD, we used the analytically computed learning rate correction from § D.4.1.

D.3.3 BERT finetuning

For DistilBERT fine-tuning experiments on AG News, we have independently tuned learning rate for each algorithm. We search the learning rate in the grid of $\{1e-5, 3e-5, 5e-5, 7e-5, 9e-5\}$ and we extend the grid to ensure that the best hyper-parameter lies in the middle of our search grids, otherwise we extend our search grid.

For the experiments in Table 5.4, we used the learning rates listed in Table D.5.

D.3.4 Random quadratics

For Figures 5.2 and 5.3, we tuned the learning rate for each compared method to reach a desired quality level as quickly as possible, using binary search. We made a distinction between methods that are expected to converge linearly, and methods that are expected to reach a plateau. For experiments with stochastic noise, we tuned a learning rate without noise first, and then lowered the learning rate if needed to reach a desirable plateau. Please see the supplied code for implementation details.

Table D.4 Learning rates used for Cifar-10/ VGG-11. Numbers between parentheses indicate the number of converged replications with this learning rate.

Algorithm	Topology	$\alpha = 1.00$ (most homogeneous)	$\alpha = 0.1$	$\alpha = .01$ (most heterogeneous)
All-reduce	fully connected	0.100 (3)	0.100 (3)	0.100 (3)
+momentum		0.100 (3)	0.100 (3)	0.100 (3)
RelaySGD	binary trees	1.200 (3)	0.600 (3)	0.300 (3)
+local momentum		0.600 (3)	0.300 (3)	0.150 (3)
DP-SGD	ring	0.400 (3)	0.100 (3)	0.200 (3)
+quasi-global mom.		0.100 (3)	0.025 (3)	0.050 (3)
D ²	ring	0.200 (3)	0.200 (3)	0.100 (3)
+local momentum		0.050 (3)	0.050 (3)	0.013 (3)
Stochastic gradient push	time-varying exponential	0.400 (3)	0.200 (3)	0.200 (3)
+local momentum		0.100 (3)	0.100 (3)	0.025 (3)

Table D.5 Tuned learning rates used for AG News / DistilBERT (Table 5.4)

Algorithm	Topology	Learning rate
Centralized Adam	fully-connected	3e-5
Relay-Adam	chain	9e-4
DP-SGD Adam	ring	1e-6
Quasi-global Adam [Lin et al., 2021b]	ring	1e-6

D.4 Algorithmic details

D.4.1 Learning-rate correction for RelaySGD

In DP-SGD as well as all other algorithms we compared to, a gradient-based update $\mathbf{u}_i^{(t)}$ from worker i at time t will eventually, as $t \rightarrow \infty$ distribute uniformly with weights $\frac{1}{n}$ over all workers. In RelaySGD, the update also distributes uniformly (typically much quicker), but it will converge to a weight $\alpha \leq \frac{1}{n}$. The constant α is fixed throughout training and depends only on the network topology used. To correct for this loss in energy, you can scale the learning rate by a factor $\frac{1}{\alpha n}$.

Experimentally, we pre-compute α for each architecture by initialing a *scalar* model for each worker to zero, updating the models to 1, and running RelaySGD until convergence with no further model updates. The worker will converge to the value α . The correction factors that result from this procedure are illustrated in Figure D.2.

In our deep learning experiments, we find that for each learning rate were centralized SGD converges, RelaySGD with the corrected learning rate converges too. Note that this learning rate correction is only useful if you already have a tuned learning rate from centralized experiments, or experiments with algorithms such as DP-SGD. If you start from scratch, tuning the learning rate for RelaySGD is no different from tuning the learning rate for any of the other algorithms.

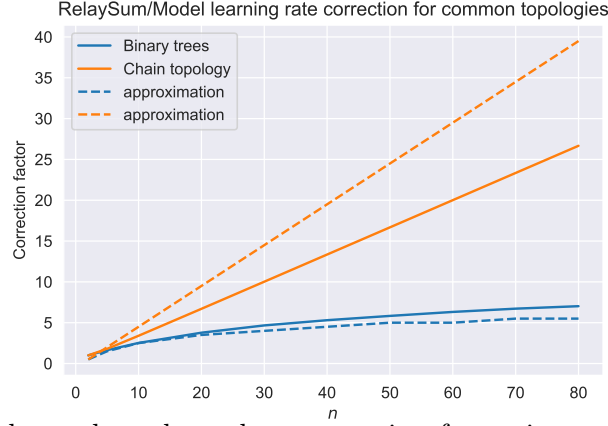


Fig. D.2 This network-topology-dependent correction factor is computed as follows: Each worker initializes a scalar model to 0 and sends a single fixed value 1 as gradient update through the RelaySGD algorithm. For DP-SGD and all-reduce, workers would converge to 1, but for RelaySGD, we lose some of this energy. If the workers converge to a value α , we will scale the learning rate with $1/\alpha$ for RelaySGD compared to all-reduce.

D.4.2 RelaySGD with momentum

RelaySGD follows Algorithm 6, but replaces the local update in line 3 with a local momentum. For Nesterov momentum with momentum-parameter α , this is:

$$\begin{aligned} \mathbf{m}_i^{(t)} &= \alpha \mathbf{m}_i^{(t-1)} + \nabla f_i(\mathbf{x}_i^{(t)}) \quad (\text{initialize } \mathbf{m}_i^0 = 0) \\ \mathbf{x}_i^{(t+1/2)} &= \mathbf{x}_i^{(t)} - \gamma \left(\nabla f_i(\mathbf{x}_i^{(t)}) + \alpha \mathbf{m}_i^{(t)} \right). \end{aligned}$$

D.4.3 RelaySGD with Adam

Modifying RelaySGD (Algorithm 6) to use Adam is analogous to RelaySGD with momentum (§ D.4.2). All Adam state is updated locally. We use the standard Adam implementation of PyTorch 1.18.

D.4.4 D^2 with momentum

We made slight modifications to the D^2 algorithm from Tang et al. [2018] to allow time-varying learning rates and local momentum. The version we use is listed as Algorithm 11. Note that D^2 requires the smallest eigenvalue of the gossip matrix \mathbf{W} to be $\geq -1/3$. This property is satisfied for Metropolis-Hasting matrices used on rings and double binary trees, but it was not in our Social Network Graph experiment (Figure 5.3). For this reason, we used the gossip matrix $(\mathbf{W} + \mathbf{I})/2$, from the otherwise-equivalent Exact Diffusion algorithm [Yuan et al., 2019] on the social network graph.

Algorithm 11 D² [Tang et al., 2018] with momentum

Input: $\forall i, \mathbf{x}_i^{(0)} = \mathbf{x}^{(0)}$, learning rate γ , momentum α , gossip matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$, $\mathbf{c}_i^{(0)} = \mathbf{0} \in \mathbb{R}^d$.

- 1: **for** $t = 0, 1, \dots$ **do**
- 2: **for** node i **in parallel**
- 3: Update the local momentum buffer $\mathbf{m}_i^{(t)} = \alpha \mathbf{m}_i^{(t-1)} + \nabla f_i(\mathbf{x}_i^{(t)})$.
- 4: Compute a local update $\mathbf{u}_i^{(t)} = -\gamma(\nabla f_i(\mathbf{x}_i^{(t)}) + \alpha \mathbf{m}_i^{(t)})$.
- 5: Update the local model $\mathbf{x}_i^{(t+1/2)} = \mathbf{x}_i^{(t)} + \mathbf{u}_i^{(t)} + \mathbf{c}_i^{(t)}$.
- 6: Average with neighbors: $\mathbf{x}_i^{(t+1)} = \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij} \mathbf{x}_j^{(t+1/2)}$.
- 7: Update the local correction $\mathbf{c}_i^{(t+1)} = \mathbf{x}_i^{(t+1)} - \mathbf{x}_i^{(t)} - \mathbf{u}_i^{(t)}$.
- 8: **end for**

D.4.5 Gradient Tracking

Algorithm 12 lists our implementation of Gradient Tracking from Lorenzo and Scutari [2016].

Algorithm 12 Gradient Tracking [Lorenzo and Scutari, 2016]

Input: $\forall i, \mathbf{x}_i^{(0)} = \mathbf{x}^{(0)}$, learning rate γ , gossip matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$, $\mathbf{c}_i^{(0)} = \mathbf{0} \in \mathbb{R}^d$.

- 1: **for** $t = 0, 1, \dots$ **do**
- 2: **for** node i **in parallel**
- 3: Compute a local update $\mathbf{u}_i^{(t)} = -\gamma \nabla f_i(\mathbf{x}_i^{(t)})$.
- 4: Update the local model $\mathbf{x}_i^{(t+1/2)} = \mathbf{x}_i^{(t)} + \mathbf{u}_i^{(t)} + \mathbf{c}_i^{(t)}$.
- 5: Average with neighbors: $\mathbf{x}_i^{(t+1)} = \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij} \mathbf{x}_j^{(t+1/2)}$.
- 6: Update the correction and average: $\mathbf{c}_i^{(t+1)} = \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij} (\mathbf{c}_i^{(t)} - \mathbf{u}_i^{(t)})$.
- 7: **end for**

D.4.6 Stochastic Gradient Push with the time-varying exponential topology

Stochastic Gradient Push with the time-varying exponential topology from Assran et al. [2019a] demonstrates that decentralized learning algorithms can reduce communication in a data center setting where each node could talk to each other node. Algorithm 13 lists our implementation of this algorithm.

D.5 Additional experiments on RelaySGD

D.5.1 Rings vs double binary trees on Cifar-10

In our experiments that target data-center inspired scenarios where the network topology is arbitrarily selected by the user to save bandwidth, RelaySGD uses double binary trees to communicate. They use the same memory and bandwidth as rings (2 models sent/received per iteration) but they delays only scale with $\log n$, enabling RelaySGD, in theory, to run with very











Algorithm 13 Stochastic Gradient Push with time-varying exponential topology [Assran et al., 2019a]

Input: $\forall i, \mathbf{x}_i^{(0)} = \mathbf{x}^{(0)}$, learning rate γ , $n = 2^k$ workers, $t' = 0$.

- 1: **for** $t = 0, 1, \dots$ **do**
- 2: **for** node i **in parallel**
- 3: $\mathbf{x}_i^{(t+1/2)} = \mathbf{x}_i^{(t)} + \mathbf{u}_i^{(t)} - \gamma \nabla f_i(\mathbf{x}_i^{(t)})$. (or momentum/Adam, like RelaySGD)
- 4: **for** 2 communication steps to equalize bandwidth with RelaySGD **do**
- 5: Compute an offset $o = 2^{t' \bmod k}$.
- 6: Send $\mathbf{x}_i^{(t+1/2)}$ to worker $i - o$.
- 7: Receive and overwrite $\mathbf{x}_i^{(t+1/2)} \leftarrow \frac{1}{2} \left(\mathbf{x}_i^{(t+1/2)} + \mathbf{x}_{i+o}^{(t+1/2)} \right)$.
- 8: $t' \leftarrow t' + 1$.
- 9: Set $\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t+1/2)}$.
- 10: **end for**

large numbers of workers n . Table D.6 shows that in our Cifar-10 experiments with 16 there are minor improvements from using double binary trees over rings. Our baselines DP-SGD and D², however, perform significantly better on rings than on trees, so we use those results in the main paper.

Table D.6 Comparing the performance of the algorithms in Table 5.1 on rings and double binary trees in the high-heterogeneity setting $\alpha = 0.01$. In both topologies, workers send and receive two full models per update step. With 16 workers, RelaySGD with momentum seems to benefit from double binary trees, RelaySGD has more consistently good results on a chain. We still opt for double binary trees based on their promise to scale to many workers. Other methods do not benefit from double binary trees over rings.

Algorithm	Ring (Chain for RelaySGD)	Double binary trees
RelaySGD	86.5% 	84.6% 
+local momentum	88.4% 	89.1% 
DP-SGD	53.9% 	36.0% 
+quasi-global mom.	63.3% 	57.5% 
D ²	38.2% 	did not converge
+local momentum	61.0% 	did not converge

D.5.2 Scaling the number of workers on Cifar-10

In this experiment (Table D.7), use momentum-SGD on 16, 32 and 64 workers compare the scaling of RelaySGD to SGP [Assran et al., 2019a]. We fix the parameter α that determines the level of data heterogeneity to $\alpha = 0.01$. Note that this level of α could lead to more challenging heterogeneity when there are many workers (and hence many smaller local subsets of the data), compared to when there are few workers.

Table D.7 Scaling the number of workers in heterogeneous Cifar-10. The heterogeneity level $\alpha = 0.01$ is kept constant, although it does change its meaning when the number of workers changes. RelaySGD scales at least well as Stochastic Gradient Push [Assran et al., 2019a] (with equal communication budget). It is surprising that RelaySGD with 64 workers performs significantly better on a chain topology than on the double binary trees. This behavior does not match what our observations on quadratic toy-problems.

Algorithm	Topology	16 workers	32 workers	64 workers
All-reduce (baseline)	fully connected	89.5%	88.9%	87.2%
RelaySGD	binary trees	89.3%	86.1%	63.7%
	chain	88.4%	86.6%	83.1%
Stochastic gradient push	time-varying exponential	87.0%	68.9%	62.4%

Table D.8 Tuned learning rates for Table D.7. We tuned the learning rate for each setting on a multiplicative grid with spacing $\sqrt{2}$, and then repeated each experiment 3 times. If both repetitions diverged, we would change to a smaller learning rate in the grid. Numbers in parentheses are the ‘effective’ learning rates corrected according to § D.4.1.

Algorithm	Topology	16 workers	32 workers	64 workers
All-reduce (baseline)	fully connected	0.1 (0.100)	0.05 (0.050)	0.05 (0.050)
RelaySGD	binary trees	0.282 (0.066)	0.2 (0.035)	0.2 (0.027)
	chain	0.2 (0.047)	0.4 (0.070)	0.8 (0.108)
Stochastic gradient push	time-varying exp.	0.025 (0.025)	0.025 (0.025)	0.0125 (0.013)

D.5.3 Independence of heterogeneity

The benefits of RelaySGD over some other methods shows most when workers have heterogeneous training objectives. Figure D.3 compares several algorithms with varying levels of data heterogeneity on synthetic quadratics on a ring topology with 32 workers. Like D^2 , RelaySGD converges linearly, and does not require more steps when the data becomes more heterogeneous. Note that, even though RelaySGD operates on a chain network instead of a ring, it is as fast as D^2 . On other topologies, such as a star topology, or on trees, RelaySGD can even be faster than D^2 (see Appendix D.5.4), while maintaining the same independence of heterogeneity.

D.5.4 Star topology

On star-topologies, the set of neighbors of worker 0 is $\{1, 2, \dots, n\}$ and the set of neighbors for every other worker is just $\{0\}$. While D^2 and RelaySGD are equally fast in the synthetic experiments on *ring* topologies in § D.5.3, RelaySGD is significantly faster on *star* topologies as illustrates by Figure D.4.

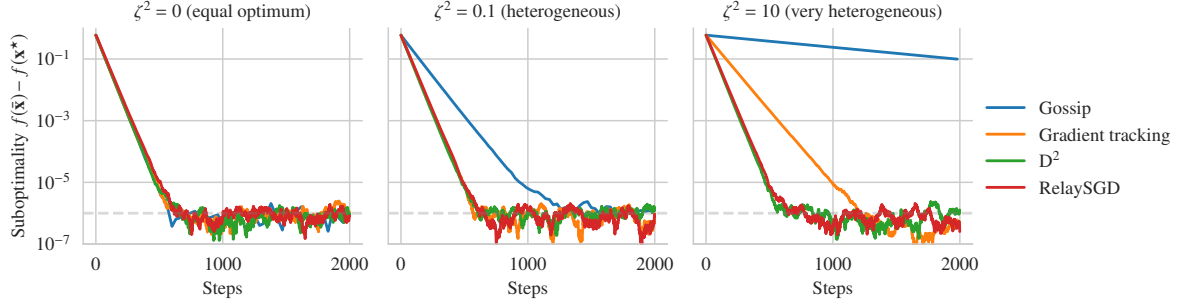


Fig. D.3 Random quadratics on *ring* networks of size 32 with varying data heterogeneity ζ^2 and all other theoretical quantities fixed. To simulate stochastic noise, we add random normal noise to each gradient update. For each method, the learning rate is tuned to reach suboptimality $\leq 10^{-6}$ the fastest. RelaySGD operates on a chain network instead of a ring. Like D^2 , it does not require more steps when the worker’s objectives are more heterogeneous.

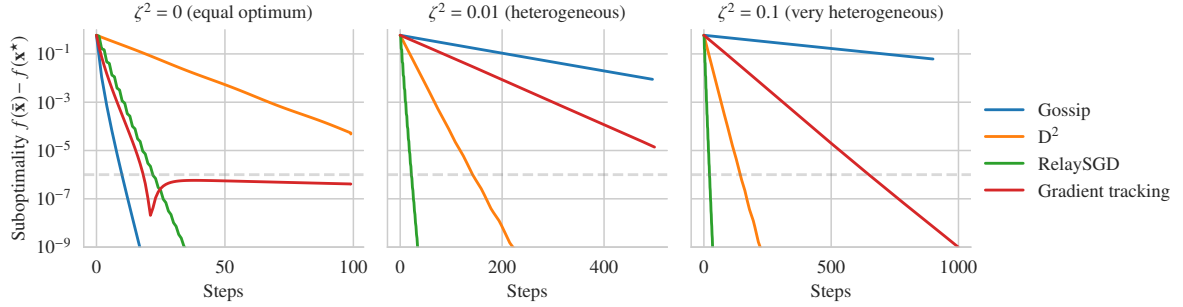


Fig. D.4 Random quadratics on *star* networks of size 32 with varying data heterogeneity ζ^2 and all other theoretical quantities fixed. For each method, the learning rate is tuned to reach suboptimality $\leq 10^{-6}$ the fastest. Like D^2 , RelaySGD does not require more steps when the worker’s objectives are more heterogeneous. Note that for $\zeta^2 = 0$ (left figure), our tuning procedure found a learning rate where Gradient Tracking does converge to $\leq 10^{-6}$, but does not converge linearly. It would with a lower learning rate.

D.6 RelaySum for distributed mean estimation

We conceptually separate the optimization algorithm RelaySGD from the communication mechanism RelaySum that uniformly distributes updates across a peer-to-peer network. We made this choice because we envision other applications of the RelaySum mechanism outside of optimization for machine learning. To illustrate this point, this section introduces RelaySum for Distributed Mean Estimation (Algorithm 14).

In distributed mean estimation, workers are connected in a network just as in our optimization setup, but instead of models gradients, they receive samples $\hat{\mathbf{d}}^{(t)} \sim \mathcal{D}$ of the distribution \mathcal{D} at timestep t . The workers estimate the mean $\bar{\mathbf{d}}$ the mean of \mathcal{D} , and we measure their average squared error to the true mean.

Algorithm 14 RelaySum for Distributed Mean Estimation

Input: $\forall i, \mathbf{x}_i^{(0)} = \mathbf{0}, \mathbf{y}_i^{(0)} = \mathbf{0}, s_i^{(0)} = 0; \forall i, j, \mathbf{m}_{i \rightarrow j}^{(-1)} = \mathbf{0}$, tree network

- 1: **for** $t = 0, 1, \dots$ **do**
- 2: **for** node i **in parallel**
- 3: **for** each neighbor $j \in \mathcal{N}_i$ **do**
- 4: Get a sample $\hat{\mathbf{d}}_i^{(t)} \sim \mathcal{D}$.
- 5: Send $\mathbf{m}_{i \rightarrow j}^{(t)} = \hat{\mathbf{d}}_i^{(t)} + \sum_{k \in \mathcal{N}_i \setminus j} \mathbf{m}_{k \rightarrow i}^{(t-1)}$.
- 6: Send $c_{i \rightarrow j}^{(t)} = 1 + \sum_{k \in \mathcal{N}_i \setminus j} c_{k \rightarrow i}^{(t-1)}$.
- 7: Receive $\mathbf{m}_{j \rightarrow i}^{(t)}$ and $c_{j \rightarrow i}^{(t)}$ from node j .
- 8: Update the sum of samples $\mathbf{y}_i^{(t+1)} = \mathbf{y}_i^{(t)} + \hat{\mathbf{d}}_i^{(t)} + \sum_{j \in \mathcal{N}_i} \mathbf{m}_{j \rightarrow i}^{(t)}$.
- 9: Update the sum of counts $s_i^{(t+1)} = s_i^{(t)} + 1 + \sum_{j \in \mathcal{N}_i} c_{j \rightarrow i}^{(t)}$.
- 10: Output average estimate $\mathbf{x}_i^{(t)} = \mathbf{y}_i^{(t)} / s_i^{(t)}$
- 11: **end for**

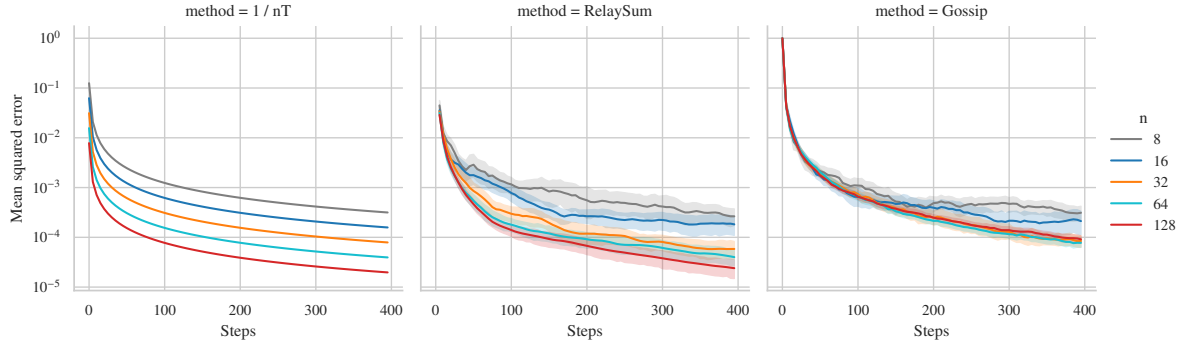


Fig. D.5 RelaySum for Distributed Mean Estimation compared to a gossip-based baseline, on a ring topology (chain for RelaySGD). Workers receive samples from a normal distribution $\mathcal{N}(1, 1)$ with mean 1. RelaySum, using Algorithm 14 achieves a variance reduction of $\mathcal{O}(\frac{1}{nT})$.

In algorithm 14, the output estimates $\mathbf{x}_i^{(t)}$ of a worker i is a uniform average of all samples that can reach a worker i at that timestep. This algorithm enjoys variance reduction of $\mathcal{O}(\frac{1}{nT})$, a desirable property that is in general not shared by gossip-averaging-based algorithms on arbitrary graphs.

In Figure D.5, we compare this algorithm to a simple gossip-based baseline.

D.7 Alternative optimizer based on RelaySum

Apart from RelaySGD presented in the main paper, there are other ways to build optimization algorithms based on the RelaySum communication mechanism. In this section, we describe RelaySGD/Grad (Algorithm 15), an alternative to RelaySGD that does uses the RelaySum mechanism on *gradient updates* rather than on *models*.

RelaySGD/Grad distributes each update uniformly over all workers in a finite number of steps. This means that worker's models differ by only a finite number of $\mathcal{O}(\tau_{\max} \max n)$ that are scaled as $\frac{1}{n}$. With this property, it achieves tighter consensus than typical gossip averaging, and it also works well in deep learning. Contrary to RelaySGD, however, this algorithm is not fully independent of data heterogeneity, due to the delay in the updates. When the data heterogeneity $\zeta^2 > 0$, RelaySGD/Grad does not converge linearly, but its suboptimality saturates at a level that depends on ζ^2 .

The sections below study this alternative algorithm in detail, both theoretically and experimentally. The key differences between RelaySGD and RelaySGD/Grad are:

	RelaySGD	RelaySGD/Grad
Provably independent of data heterogeneity ζ^2	yes	no
Distributes updates exactly uniform in finite steps	no	yes
Loses energy of gradient updates (§ D.4.1)	yes	no
Works experimentally with momentum / Adam	yes	no
Robust to lost messages + can support workers joining/leaving	yes	no

Algorithm 15 RelaySGD/Grad

Input: $\forall i, \mathbf{x}_i^{(0)} = \mathbf{x}^{(0)}; \forall i, j, \mathbf{m}_{i \rightarrow j}^{(-1)} = \mathbf{0}$, learning rate γ , tree network

```

1: for  $t = 0, 1, \dots$  do
2:   for node  $i$  in parallel
3:      $\mathbf{u}_i^{(t)} = -\gamma \nabla f_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)})$ 
4:     for each neighbor  $j \in \mathcal{N}_i$  do
5:       Send  $\mathbf{m}_{i \rightarrow j}^{(t)} = \mathbf{u}_i^{(t)} + \sum_{k \in \mathcal{N}_i \setminus j} \mathbf{m}_{k \rightarrow i}^{(t-1)}$ .
6:       Receive  $\mathbf{m}_{j \rightarrow i}^{(t)}$  from node  $j$ .
7:        $\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \frac{1}{n} \left( \mathbf{u}_i^{(t)} + \sum_{j \in \mathcal{N}_i} \mathbf{m}_{j \rightarrow i}^{(t)} \right)$ 
8:   end for
```

D.7.1 Theoretical analysis of RelaySGD/Grad

In this section we provide the theoretical analysis for RelaySGD/Grad. As the proof and analysis is very similar to [Koloskova et al. \[2020b\]](#), we only provide the case for the convex objective.

Proof of RelaySGD/Grad for the convex case

Let \mathbf{x}^* be the minimizer of f and define the following iterates

- $r_t := \mathbb{E} \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}^*\|^2$,
- $e_t := f(\bar{\mathbf{x}}^{(t)}) - f(\mathbf{x}^*)$,
- $\Xi_t := \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}^{(t)}\|^2$.

Proposition D.9. *Let function $F_i(\mathbf{x}, \xi)$, $i \in [n]$ be L -smooth (Assumption A) with bounded noise at the optimum (Assumption D). Then for any $\mathbf{x}_i \in \mathbb{R}^d$,*

$$\mathbb{E}_{\xi_1^t, \dots, \xi_n^t} \left\| \frac{1}{n} \sum_{i=1}^n (\nabla f_i(\mathbf{x}_i^{(t)}) - \nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)})) \right\|^2 \leq \frac{3}{n} (L^2 \Xi_t + 2Le_t + \bar{\sigma}^2)$$

Proof. In this proof we ignore the superscript t as it does not raise ambiguity.

$$\begin{aligned} & \mathbb{E}_{\xi_1, \dots, \xi_n} \left\| \frac{1}{n} \sum_{i=1}^n (\nabla f_i(\mathbf{x}_i) - \nabla F_i(\mathbf{x}_i, \xi_i)) \right\|^2 \leq \frac{1}{n^2} \sum_{i=1}^n \mathbb{E}_{\xi_i} \|\nabla f_i(\mathbf{x}_i) - \nabla F_i(\mathbf{x}_i, \xi_i)\|^2 \\ &= \frac{1}{n^2} \sum_{i=1}^n \mathbb{E}_{\xi_i} \|\nabla f_i(\mathbf{x}_i) - \nabla F_i(\mathbf{x}_i, \xi_i) \pm \nabla F_i(\bar{\mathbf{x}}, \xi_i) \pm \nabla f_i(\bar{\mathbf{x}}) \pm \nabla F_i(\mathbf{x}^*, \xi_i) \pm \nabla f_i(\mathbf{x}^*)\|^2 \\ &\leq \frac{3}{n^2} \sum_{i=1}^n \mathbb{E}_{\xi_i} (\|\nabla f_i(\mathbf{x}_i) - \nabla f_i(\bar{\mathbf{x}}) + \nabla F_i(\bar{\mathbf{x}}, \xi_i) - \nabla F_i(\mathbf{x}_i, \xi_i)\|^2 \\ &\quad + \|\nabla f_i(\bar{\mathbf{x}}) - \nabla f_i(\mathbf{x}^*) + \nabla F_i(\mathbf{x}^*, \xi_i) - \nabla F_i(\bar{\mathbf{x}}, \xi_i)\|^2 + \|\nabla f_i(\mathbf{x}^*) - \nabla F_i(\mathbf{x}^*, \mathbf{x}_i)\|^2) \\ &\leq \frac{3}{n^2} \sum_{i=1}^n \mathbb{E}_{\xi_i} (\|\nabla F_i(\mathbf{x}_i, \xi_i) - \nabla F_i(\bar{\mathbf{x}}, \xi_i)\|^2 + \|\nabla F_i(\bar{\mathbf{x}}, \xi_i) - \nabla F_i(\mathbf{x}^*, \xi_i)\|^2 + \|\nabla F_i(\mathbf{x}^*, \mathbf{x}_i) - \nabla f_i(\mathbf{x}^*)\|^2) \\ &\leq \frac{3}{n^2} \sum_{i=1}^n (L^2 \|\mathbf{x}_i - \bar{\mathbf{x}}\|^2 + 2L(f_i(\bar{\mathbf{x}}) - f_i(\mathbf{x}^*)) + \sigma_i^2) \end{aligned}$$

□

Lemma D.17. (Descent lemma for convex objective.) *If $\gamma \leq \frac{1}{10L}$, then*

$$r_{t+1} \leq (1 - \frac{\gamma\mu}{2})r_t - \gamma e_t + 3\gamma L\Xi_t + \frac{3}{n}\gamma^2\bar{\sigma}^2.$$

Proof. Throughout this proof we use $\mathbb{E} = \mathbb{E}_{\xi_1^t, \dots, \xi_n^t}$. Expand iterate $r_{t+1} = \mathbb{E} \|\bar{\mathbf{x}}^{(t+1)} - \mathbf{x}^*\|^2$

$$\begin{aligned} & \mathbb{E} \|\bar{\mathbf{x}}^{(t+1)} - \mathbf{x}^*\|^2 \\ &= \mathbb{E} \|\bar{\mathbf{x}}^{(t)} - \frac{\gamma}{n} \sum_{i=1}^n \nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)}) \pm \frac{\gamma}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t)}) - \mathbf{x}^*\|^2 \\ &= \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}^* - \frac{\gamma}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t)})\|^2 + \mathbb{E} \|\frac{\gamma}{n} \sum_{i=1}^n \nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)}) - \frac{\gamma}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t)})\|^2 \\ &\quad + 2\mathbb{E} \langle \bar{\mathbf{x}}^{(t)} - \mathbf{x}^* - \frac{\gamma}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t)}), \frac{\gamma}{n} \sum_{i=1}^n \nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)}) - \frac{\gamma}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t)}) \rangle \\ &= \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}^* - \frac{\gamma}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t)})\|^2 + \mathbb{E} \|\frac{\gamma}{n} \sum_{i=1}^n \nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)}) - \frac{\gamma}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t)})\|^2 \end{aligned}$$

The second term is bounded by Proposition D.9. Consider the first term

$$\begin{aligned} & \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}^* - \frac{\gamma}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t)})\|^2 \\ & \leq \underbrace{\|\bar{\mathbf{x}}^{(t)} - \mathbf{x}^*\|^2}_{=:T_1} + \underbrace{\gamma^2 \left\| \frac{1}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t)}) \right\|^2}_{=:T_2} - 2\gamma \underbrace{\langle \bar{\mathbf{x}}^{(t)} - \mathbf{x}^*, \frac{1}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t)}) \rangle}_{=:T_2}. \end{aligned}$$

First consider T_1 ,

$$\begin{aligned} T_1 &= \left\| \frac{1}{n} \sum_{i=1}^n (\nabla f_i(\mathbf{x}_i^{(t)}) - \nabla f_i(\bar{\mathbf{x}}^{(t)}) + \nabla f_i(\bar{\mathbf{x}}^{(t)}) - \nabla f_i(\mathbf{x}^*)) \right\|^2 \\ &\leq \frac{2L^2}{n} \sum_{i=1}^n \|\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}^{(t)}\|^2 + \frac{2}{n} \sum_{i=1}^n \|\nabla f_i(\bar{\mathbf{x}}^{(t)}) - \nabla f_i(\mathbf{x}^*)\|^2 \\ &\stackrel{(\text{D.4})}{\leq} \frac{2L^2}{n} \sum_{i=1}^n \|\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}^{(t)}\|^2 + \frac{4L}{n} \sum_{i=1}^n (f_i(\bar{\mathbf{x}}^{(t)}) - f_i(\mathbf{x}^*) - \langle \bar{\mathbf{x}}^{(t)} - \mathbf{x}^*, \nabla f_i(\mathbf{x}^*) \rangle) \\ &= \frac{2L^2}{n} \sum_{i=1}^n \|\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}^{(t)}\|^2 + 4L(f(\bar{\mathbf{x}}^{(t)}) - f(\mathbf{x}^*)) \\ &= 2L^2\Xi_t + 4Le_t. \end{aligned}$$

Consider T_2 ,

$$\begin{aligned} T_2 &= \frac{1}{n} \sum_{i=1}^n (\langle \bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t)}, \nabla f_i(\mathbf{x}_i^{(t)}) \rangle + \langle \mathbf{x}_i^{(t)} - \mathbf{x}^*, \nabla f_i(\mathbf{x}_i^{(t)}) \rangle) \\ &\geq \frac{1}{n} \sum_{i=1}^n \left(f_i(\bar{\mathbf{x}}^{(t)}) - f_i(\mathbf{x}_i^{(t)}) - \frac{L}{2} \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t)}\|^2 + \langle \mathbf{x}_i^{(t)} - \mathbf{x}^*, \nabla f_i(\mathbf{x}_i^{(t)}) \rangle \right) \\ &\geq \frac{1}{n} \sum_{i=1}^n \left(f_i(\bar{\mathbf{x}}^{(t)}) - f_i(\mathbf{x}_i^{(t)}) - \frac{L}{2} \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t)}\|^2 + f_i(\mathbf{x}_i^{(t)}) - f_i(\mathbf{x}^*) + \frac{\mu}{2} \|\mathbf{x}_i^{(t)} - \mathbf{x}^*\|^2 \right) \\ &= f(\bar{\mathbf{x}}^{(t)}) - f(\mathbf{x}^*) + \frac{1}{n} \sum_{i=1}^n \left(\frac{\mu}{2} \|\mathbf{x}_i^{(t)} - \mathbf{x}^*\|^2 - \frac{L}{2} \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t)}\|^2 \right) \\ &\geq f(\bar{\mathbf{x}}^{(t)}) - f(\mathbf{x}^*) + \frac{1}{n} \sum_{i=1}^n \left(\frac{\mu}{4} \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}^*\|^2 - \frac{\mu+L}{2} \|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_i^{(t)}\|^2 \right) \\ &\geq e_t + \frac{\mu}{4} r_t - L\Xi_t \end{aligned}$$

where the first inequality and the second inequality uses the L -smoothness and μ -convexity of f_i .

Combine both T_1 , T_2 and Proposition D.9 we have

$$\begin{aligned} r_{t+1} &\leq r_t + \gamma^2(2L^2\Xi_t + 4Le_t) - 2\gamma(e_t + \frac{\mu}{4}r_t - L\Xi_t) + \frac{3}{n}\gamma^2(L^2\Xi_t + 2Le_t + \bar{\sigma}^2) \\ &= (1 - \frac{\gamma\mu}{2})r_t - 2\gamma(1 - 5L\gamma)e_t + \gamma L(5\gamma L + 2)\Xi_t + \frac{3}{n}\gamma^2\bar{\sigma}^2. \end{aligned}$$

In addition if $\gamma \leq \frac{1}{10L}$, then

$$r_{t+1} \leq (1 - \frac{\gamma\mu}{2})r_t - \gamma e_t + 3\gamma L\Xi_t + \frac{3}{n}\gamma^2\bar{\sigma}^2.$$

□

Lemma D.18. *Bound the consensus distance as follows*

$$\Xi_t \leq 3\gamma^2\tau_{\max} \sum_{t'=[t-\tau_{\max}]_+}^{t-1} (2L^2\Xi_{t'} + 4Le_{t'} + (\bar{\sigma}^2 + \bar{\zeta}^2)).$$

Furthermore, multiply with a non-negative sequence $\{w_t\}_{t \geq 0}$ and average over time gives

$$\frac{1}{W_T} \sum_{t=0}^T w_t \Xi_t \leq \frac{1}{6LW_T} \sum_{t=0}^T w_t e_t + 6\gamma^2\tau_{\max}^2 (\bar{\sigma}^2 + \bar{\zeta}^2)$$

where $W_T := \sum_{t=0}^T w_t$ and $\gamma \leq \frac{1}{10L\tau_{\max}}$.

Proof. Throughout this proof we use $\mathbb{E} = \mathbb{E}_{\xi_1^t, \dots, \xi_n^t}$. Denote $[x]^+ := \max\{x, 0\}$. For all $i \in [n]$,

$$\begin{aligned} \mathbb{E}\|e_i^t\|^2 &= \mathbb{E}\left\|\frac{\gamma}{n} \sum_{j=1}^n \sum_{t'=[t-\tau_{\max}i_j]_+}^{t-1} \nabla F_j(\mathbf{x}_j^{(t')}, \xi_j^{(t')}) \pm \nabla f_j(\mathbf{x}_j^{(t')})\right\|^2 \\ &\leq \frac{\gamma^2}{n} \sum_{j=1}^n \mathbb{E}\left\|\sum_{t'=[t-\tau_{\max}i_j]_+}^{t-1} \nabla F_j(\mathbf{x}_j^{(t')}, \xi_j^{(t')}) \pm \nabla f_j(\mathbf{x}_j^{(t')})\right\|^2 \\ &\leq \frac{\gamma^2\tau_{\max}}{n} \sum_{j=1}^n \sum_{t'=[t-\tau_{\max}]_+}^{t-1} \mathbb{E}\|\nabla F_j(\mathbf{x}_j^{(t')}, \xi_j^{(t')}) \pm \nabla f_j(\mathbf{x}_j^{(t')})\|^2 \\ &= \frac{\gamma^2\tau_{\max}}{n} \sum_{j=1}^n \sum_{t'=[t-\tau_{\max}]_+}^{t-1} \mathbb{E}\|\nabla F_j(\mathbf{x}_j^{(t')}, \xi_j^{(t')}) - \nabla f_j(\mathbf{x}_j^{(t')})\|^2 \\ &\quad + \underbrace{\frac{\gamma^2\tau_{\max}}{n} \sum_{j=1}^n \sum_{t'=[t-\tau_{\max}]_+}^{t-1} \|\nabla f_j(\mathbf{x}_j^{(t')})\|^2}_{=: T_3} \end{aligned}$$

We can apply Proposition D.9 to the first term

$$\frac{\gamma^2\tau_{\max}}{n} \sum_{j=1}^n \sum_{t'=[t-\tau_{\max}]_+}^{t-1} \mathbb{E}\|\nabla F_j(\mathbf{x}_j^{(t')}, \xi_j^{(t')}) - \nabla f_j(\mathbf{x}_j^{(t')})\|^2 \leq 3\gamma^2\tau_{\max} \sum_{t'=[t-\tau_{\max}]_+}^{t-1} (L^2\Xi_{t'} + 2Le_{t'} + \bar{\sigma}^2).$$

The second term T_3 can be bounded by adding $0 = \pm \nabla f_j(\bar{\mathbf{x}}^{(t')}) \pm \nabla f_j(\mathbf{x}^*)$ inside the norm

$$\begin{aligned} T_3 &\leq \frac{\gamma^2\tau_{\max}}{n} \sum_{j=1}^n \sum_{t'=[t-\tau_{\max}]_+}^{t-1} \|\nabla f_j(\mathbf{x}_j^{(t')}) \pm \nabla f_j(\bar{\mathbf{x}}^{(t')}) \pm \nabla f_j(\mathbf{x}^*)\|^2 \\ &\leq \frac{3\gamma^2\tau_{\max}}{n} \sum_{j=1}^n \sum_{t'=[t-\tau_{\max}]_+}^{t-1} \left(L^2\|\mathbf{x}_j^{(t')} - \bar{\mathbf{x}}^{(t')}\|^2 + \|\nabla f_j(\bar{\mathbf{x}}^{(t')}) - \nabla f_j(\mathbf{x}^*)\|^2 + \|\nabla f_j(\mathbf{x}^*)\|^2 \right) \\ &= 3\gamma^2\tau_{\max} \sum_{t'=[t-\tau_{\max}]_+}^{t-1} \left(L^2\Xi_{t'} + \frac{1}{n} \sum_{j=1}^n \|\nabla f_j(\bar{\mathbf{x}}^{(t')}) - \nabla f_j(\mathbf{x}^*)\|^2 + \bar{\zeta}^2 \right) \\ &\stackrel{(D.4)}{\leq} 3\gamma^2\tau_{\max} \sum_{t'=[t-\tau_{\max}]_+}^{t-1} \left(L^2\Xi_{t'} + 2L(f(\bar{\mathbf{x}}^{(t')}) - f(\mathbf{x}^*)) + \bar{\zeta}^2 \right) \end{aligned}$$

Therefore

$$\mathbb{E}\|e_i^t\|^2 \leq 3\gamma^2\tau_{\max} \sum_{t'=[t-\tau_{\max}]_+}^{t-1} (2L^2\Xi_{t'} + 4Le_{t'} + (\bar{\sigma}^2 + \bar{\zeta}^2)).$$

Average over i on both sides and note the right hand side does not depend on index i ,

$$\Xi_t = \frac{1}{n} \sum_{i=1}^n \mathbb{E}\|e_i^t\|^2 \leq 3\gamma^2\tau_{\max} \sum_{t'=[t-\tau_{\max}]_+}^{t-1} (2L^2\Xi_{t'} + 4Le_{t'} + \bar{\sigma}^2).$$

Multiply both sides by w_t and sum over t gives

$$\begin{aligned} \frac{1}{W_T} \sum_{t=0}^T w_t \Xi_t &\leq \frac{3\gamma^2 \tau_{\max}^2}{W_T} \sum_{t=0}^T w_t (2L^2 \Xi_t + 4Le_t + \bar{\sigma}^2) \\ &= \frac{6\gamma^2 L^2 \tau_{\max}^2}{W_T} \sum_{t=0}^T w_t \Xi_t + \frac{12\gamma^2 L \tau_{\max}^2}{W_T} \sum_{t=0}^T w_t e_t + 3\gamma^2 \tau_{\max} (\bar{\sigma}^2 + \bar{\zeta}^2) \end{aligned}$$

where $W_T := \sum_{t=0}^T w_t$. Rearrange the terms and let $\gamma \leq \frac{1}{10L\tau_{\max}}$ give

$$\begin{aligned} \frac{1}{W_T} \sum_{t=0}^T w_t \Xi_t &\leq \frac{1}{1 - 6\gamma^2 L^2 \tau_{\max}^2} \left(\frac{12\gamma^2 L \tau_{\max}^2}{W_T} \sum_{t=0}^T w_t e_t + \frac{3\gamma^2 \tau_{\max}^2}{n} (\bar{\sigma}^2 + \bar{\zeta}^2) \right) \\ &\leq \frac{1}{6LW_T} \sum_{t=0}^T w_t e_t + 6\gamma^2 \tau_{\max}^2 (\bar{\sigma}^2 + \bar{\zeta}^2) \end{aligned}$$

□

Theorem D.10. *For convex objective, we have*

$$\frac{1}{T+1} \sum_{t=0}^T \left(f(\bar{\mathbf{x}}^{(t)}) - f(\mathbf{x}^*) \right) \leq 4 \left(\frac{3\bar{\sigma}^2 r_0}{n(T+1)} \right)^{\frac{1}{2}} + 4 \left(\frac{6\tau_{\max} \sqrt{L(\bar{\sigma}^2 + \bar{\zeta}^2)} r_0}{T+1} \right)^{\frac{2}{3}} + \frac{10L(\tau_{\max} + 1)r_0}{T+1}.$$

where $r_0 = \|\mathbf{x}^0 - \mathbf{x}^*\|^2$.

Remark 19. *For target accuracy $\epsilon > 0$, then $\frac{1}{T+1} \sum_{t=0}^T (f(\bar{\mathbf{x}}^{(t)}) - f(\mathbf{x}^*)) < \epsilon$ after*

$$\mathcal{O} \left(\frac{\bar{\sigma}^2 r_0}{n\epsilon^2} + \frac{\tau_{\max} \sqrt{L(\bar{\sigma}^2 + \bar{\zeta}^2)} r_0}{\epsilon^{3/2}} + \frac{10L(\tau_{\max} + 1)r_0}{\epsilon} \right)$$

iterations. This result is similar to [Koloskova et al., 2020b, Theorem 2] except that here we replace spectral gap p with the inverse of maximum delay $\frac{1}{\tau_{\max}}$.

Proof. Consider Lemma D.17 and multiply both sides with $\frac{w_t}{\gamma}$ and average over time

$$\begin{aligned} \frac{1}{W_T} \sum_{t=0}^T w_t e_t &\leq \frac{1}{W_T} \sum_{t=0}^T \left(\frac{w_t}{\gamma} r_t - \frac{w_t}{\gamma} r_{t+1} \right) + \frac{3L}{W_T} \sum_{t=0}^T w_t \Xi_t + \frac{3\gamma}{nW_T} \sum_{t=0}^T w_t \bar{\sigma}^2 \\ &\leq \frac{1}{W_T} \sum_{t=0}^T \left(\frac{w_t}{\gamma} r_t - \frac{w_t}{\gamma} r_{t+1} \right) + \frac{1}{2W_T} \sum_{t=0}^T w_t e_t + 18\gamma^2 \tau_{\max}^2 L(\bar{\sigma}^2 + \bar{\zeta}^2) + \frac{3\gamma \bar{\sigma}^2}{n} \end{aligned}$$

where the second inequality comes from Lemma D.18. Then

$$\frac{1}{2W_T} \sum_{t=0}^T w_t e_t \leq \frac{1}{W_T} \sum_{t=0}^T \left(\frac{w_t}{\gamma} r_t - \frac{w_t}{\gamma} r_{t+1} + \frac{3\bar{\sigma}^2}{n} \gamma + 18\tau_{\max}^2 L(\bar{\sigma}^2 + \bar{\zeta}^2) \gamma^2 \right).$$

We can further consider

$$\begin{aligned} \frac{3L}{W_T} \sum_{t=0}^T w_t \Xi_t &= \frac{1}{2W_T} \sum_{t=0}^T w_t e_t + 18\tau_{\max}^2 L(\bar{\sigma}^2 + \bar{\zeta}^2) \gamma^2 \\ &\leq \frac{1}{W_T} \sum_{t=0}^T \left(\frac{w_t}{\gamma} r_t - \frac{w_t}{\gamma} r_{t+1} + \frac{3\bar{\sigma}^2}{n} \gamma + 36\tau_{\max}^2 L(\bar{\sigma}^2 + \bar{\zeta}^2) \gamma^2 \right) =: \Psi_T. \end{aligned}$$

Taking $\{w_t = 1\}_{t \geq 0}$, then

$$\Psi_T \leq \frac{r_0}{\gamma(T+1)} + \frac{3\bar{\sigma}^2}{n}\gamma + 36\tau_{\max}^2 L(\bar{\sigma}^2 + \bar{\zeta}^2)\gamma^2.$$

Apply Lemma D.13 we have

$$\Psi_T \leq 2 \left(\frac{3\bar{\sigma}^2 r_0}{n(T+1)} \right)^{\frac{1}{2}} + 2 \left(\frac{6\tau_{\max} \sqrt{L(\bar{\sigma}^2 + \bar{\zeta}^2)} r_0}{T+1} \right)^{\frac{2}{3}} + \frac{dr_0}{T+1}.$$

where $d = \max\{10L, 10L\tau_{\max}\} \leq 10L(\tau_{\max} + 1)$ and at the same time

$$\begin{aligned} \frac{1}{2(T+1)} \sum_{t=0}^T e_t &\leq 2 \left(\frac{3\bar{\sigma}^2 r_0}{n(T+1)} \right)^{\frac{1}{2}} + 2 \left(\frac{6\tau_{\max} \sqrt{L(\bar{\sigma}^2 + \bar{\zeta}^2)} r_0}{T+1} \right)^{\frac{2}{3}} + \frac{dr_0}{T+1} \\ \frac{3L}{T+1} \sum_{t=0}^T \Xi_t &\leq 2 \left(\frac{3\bar{\sigma}^2 r_0}{n(T+1)} \right)^{\frac{1}{2}} + 2 \left(\frac{6\tau_{\max} \sqrt{L(\bar{\sigma}^2 + \bar{\zeta}^2)} r_0}{T+1} \right)^{\frac{2}{3}} + \frac{dr_0}{T+1} \end{aligned}$$

□

D.7.2 Empirical analysis of RelaySGD/Grad

In Table D.9, we compare RelaySGD/Grad to RelaySGD on deep-learning based image classification on Cifar-10 with VGG-11. Without momentum, and with low levels of heterogeneity, RelaySGD/Grad sometimes outperforms RelaySGD.

Figure D.6 illustrates a key difference between RelaySGD/Grad and RelaySGD. While RelaySGD behaves independently of heterogeneity, and converges linearly with a fixed step size, RelaySGD/Grad reaches a plateau based on the learning rate and level of heterogeneity.

Table D.9 Comparing RelaySGD/Grad with RelaySGD on Cifar-10 Krizhevsky [2012] with the VGG-11 architecture. We vary the data heterogeneity α [Lin et al., 2021b] between 16 workers. For low-heterogeneity cases and without momentum, RelaySGD/Grad sometimes performs better than RelaySGD.

Algorithm	Topology	$\alpha = 1.00$ (most homogeneous)	$\alpha = 0.1$	$\alpha = .01$ (most heterogeneous)
All-reduce (baseline) +momentum	fully connected	87.0% $\longrightarrow \# \longrightarrow$ 90.2% $\longrightarrow \#$	87.0% $\longrightarrow \# \longrightarrow$ 90.2% $\longrightarrow \#$	87.0% $\longrightarrow \# \longrightarrow$ 90.2% $\longrightarrow \#$
RelaySGD +local momentum	chain	87.3% $\longrightarrow \# \longrightarrow$ 89.5% $\longrightarrow \# \longrightarrow$	87.2% $\longrightarrow \# \longrightarrow$ 89.2% $\longrightarrow \# \longrightarrow$	86.5% $\longrightarrow \# \longrightarrow$ 88.4% $\longrightarrow \# \longrightarrow$
RelaySGD/Grad +local momentum	chain	88.8% $\longrightarrow \# \longrightarrow$ 86.9% $\longrightarrow \# \longrightarrow$	88.5% $\longrightarrow \# \longrightarrow$ 87.8% $\longrightarrow \# \longrightarrow$	83.5% $\longrightarrow \# \longrightarrow$ 68.6% $\longrightarrow \# \longrightarrow$

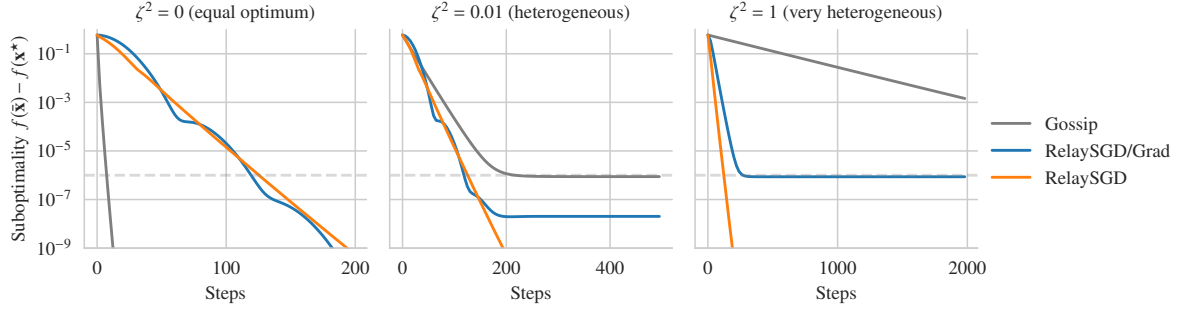


Fig. D.6 Comparing RelaySGD/Grad against RelaySGD on random quadratics with varying levels of heterogeneity ζ^2 , without stochastic noise, on a ring/chain of 32 nodes. Learning rates are tuned to reach suboptimality $\leq 10^{-6}$ as quickly as possible. In contrast to RelaySGD, RelaySGD/Grad with a fixed learning rate does not converge linearly. Compared to DP-SGD (Gossip), RelaySGD/Grad is still less sensitive to data heterogeneity.

Appendix E

Debiasing Conditional Stochastic Optimization

E.1 Missing Pseudocodes

We present pseudocodes of E-BSGD and E-BSpiderBoost scheme in Algorithms 16 and 17 respectively.

Algorithm 16 E-BSGD

- 1: **Input:** $\mathbf{x}^0 \in \mathbb{R}^d$, step-size γ , batch sizes m
 - 2: **for** $t = 0, 1, \dots, T - 1$ **do**
 - 3: Draw one sample ξ and compute extrapolated gradient $G_{\text{E-BSGD}}^{t+1}$ from (6.7)
 - 4: $\mathbf{x}^{t+1} \leftarrow \mathbf{x}^t - \gamma G_{\text{E-BSGD}}^{t+1}$
 - 5: **Output:** \mathbf{x}^s picked uniformly at random from $\{\mathbf{x}^t\}_{t=0}^{T-1}$
-

E.2 Missing Details from § 6.2

E.2.1 Other Related Work

CSO. Dai et al. [2017] proposed a primal-dual stochastic approximation algorithm to solve a min-max reformulation of CSO, employing the kernel embedding techniques. However, this method requires convexity of f_ξ and linearity of g_η , which are not satisfied by general applications when neural networks are involved. Goda and Kitade [2022] showed that a special class of CSO problems can be unbiased, e.g., when f_ξ measures the squared error between some $u(\xi)$ and $\mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}; \xi)]$, giving rise to this objective function $\mathbb{E}_\xi[(u(\xi) - \mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}; \xi)])^2]$. However, they did not show any improvement over the sample complexity of BSGD (i.e., $\mathcal{O}(\epsilon^{-6})$). Hu et al. [2020b] also analyzed lower bounds on the minimax error for the CSO problem and showed that for a specific class of biased gradients with $\mathcal{O}(\epsilon)$ bias (same bias as BSGD) and variance $\mathcal{O}(1)$ the

Algorithm 17 E-BSpiderBoost

```

1: Input:  $\mathbf{x}^0 \in \mathbb{R}^d$ , step-size  $\gamma$ , batch sizes  $B_1, B_2$ , Probability  $p_{\text{out}}$ 
2: for  $t = 0, 1, \dots, T - 1$  do
3:   Draw  $\chi_{\text{out}}$  from Bernoulli( $p_{\text{out}}$ )
4:   if ( $t = 0$ ) or ( $\chi_{\text{out}} = 1$ ) then ▷ Large batch
5:     Draw  $\mathcal{B}_1$  outer samples  $\{\xi_1, \dots, \xi_{B_1}\}$ 
6:     Compute extrapolated gradient  $G_{\text{E-BSGD}}^{t+1}$  with (6.7)

$$G_{\text{E-BSB}}^{t+1} = \frac{1}{B_1} \sum_{\xi \in \mathcal{B}_1} G_{\text{E-BSGD}}^{t+1}$$

7:   else ▷ Small batch
8:     Draw  $\mathcal{B}_2$  outer samples  $\{\xi_1, \dots, \xi_{B_2}\}$ 
9:     Compute extrapolated gradient  $G_{\text{E-BSGD}}^{t+1}$  with (6.7)

$$G_{\text{E-BSB}}^{t+1} = G_{\text{E-BSB}}^t + \frac{1}{B_2} \sum_{\xi \in \mathcal{B}_2} (G_{\text{E-BSGD}}^{t+1} - G_{\text{E-BSGD}}^t)$$

10:    $\mathbf{x}^{t+1} = \mathbf{x}^t - \gamma G_{\text{E-BSB}}^{t+1}$ 
11: Output:  $\mathbf{x}^s$  picked uniformly at random from  $\{\mathbf{x}^t\}_{t=0}^{T-1}$ 

```

bound achieved by BSpiderBoost is tight. However, these lower bounds are not applicable in settings such as ours (and also to [Hu et al., 2021]) where the bias is smaller than the BSGD bias.

Variance Reduction. The reduction of variance in stochastic optimization is a crucial approach to decrease sample complexity, particularly when dealing with finite-sum formulations of the form $\min_{\mathbf{x}} \frac{1}{n} \sum_{i=1}^n f_i(\mathbf{x})$. Pioneering works such as Stochastic Average Gradient (SAG) [Schmidt et al., 2017], Stochastic Variance Reduced Gradient (SVRG) [Johnson and Zhang, 2013; Reddi et al., 2016a], and SAGA [Defazio et al., 2014; Reddi et al., 2016b] improved the iteration complexity from $\mathcal{O}(\epsilon^{-4})$ in Stochastic Gradient Descent (SGD) to $\mathcal{O}(\epsilon^{-2})$. Subsequent research, including Stochastic Path-Integrated Differential Estimator (SPIDER) [Fang et al., 2018] and Stochastic Recursive Gradient Algorithm (SARAH) [Nguyen et al., 2017], expanded the application of these techniques to both finite-sum and online scenarios, where n is large or possibly infinite. These methods boast an improved sample complexity of $\min(\sqrt{n}\epsilon^{-2}, \epsilon^{-3})$. SpiderBoost [Wang et al., 2019], achieves the same near-optimal complexity performance as SPIDER, but allows a much larger step size and hence runs faster in practice than SPIDER. In this paper, we use a probabilistic variant of SpiderBoost as the variance reduction module for CSO and FCCO problems. We highlight that alternative techniques, such as SARAH, can also be applied and offer similar guarantees.

Bias Correction. One of the classic problems in statistics is to design procedures to reduce the bias of estimators. Well-established general bias correction techniques, such as the jackknife [Tukey, 1958], bootstrap [Efron, 1992], Taylor series [Han et al., 2020; Withers, 1987], have been extensively studied and applied in various contexts [Jiao and Han, 2020]. However, these

methods are predominantly examined in relation to standard statistical distributions, with limited emphasis on their adaptability to optimization problems. Our proposed extrapolation-based approach is derived from sample-splitting methods [Han et al., 2020], specifically tailored and analyzed for optimization problems involving unknown distributions.

Stochastic Composition Optimization. Finally, a closely related class of problems, called stochastic composition optimization, has been extensively studied (e.g., [Ermoliev and Norkin, 2013; Wang et al., 2016, 2017; Yermol'yev, 1971]) in the literature where the goal is:

$$\min_{\mathbf{x} \in \mathbb{R}^d} \mathbb{E}_{\xi} [f_{\xi}(\mathbb{E}_{\eta} [g_{\eta}(\mathbf{x})])]. \quad (\text{E.1})$$

Despite having nested expectations in their formulations (CSO) and (E.1) are fundamentally different: a) in stochastic composite optimization the inner randomness η is conditionally dependent on the outer randomness ξ and b) in CSO the inner random function $g_{\eta}(\mathbf{x}, \xi)$ depends on both ξ and η . These differences lead to quite different sample complexity bounds for these problems, as explored in Hu et al. [2020a]. In fact, Zhang and Xiao [2021] presented a near optimal complexity of $\mathcal{O}(\min(\epsilon^{-3}, \sqrt{n}\epsilon^{-2}))$ for stochastic composite optimization problems using nested variance reduction. While Wang et al. [2016] also use the "extrapolation" technique, their motivation and formula are significantly different from ours and cannot reduce the bias in the CSO problem.

E.3 Missing Details from § 6.3

Lemma E.1 (Moments of \mathcal{D}_m). *The moments of $\delta \in \mathcal{D}_m$ are bounded as follows*

$$\mathbb{E}[(\delta - \mathbb{E}[\delta])^2] = \frac{\sigma_2}{m}, \quad |\mathbb{E}[(\delta - \mathbb{E}[\delta])^3]| = \frac{\sigma_3}{m^2}, \quad \mathbb{E}[(\delta - \mathbb{E}[\delta])^4] = \frac{\sigma_4}{m^3} + \frac{3(m-1)\sigma_2^2}{m^3}.$$

More generally, for $k \geq 2$, $|\mathbb{E}[(\delta - \mathbb{E}[\delta])^k]| = \mathcal{O}(m^{-\lceil k/2 \rceil})$.

Proof. Define $\hat{\delta} = \delta - \mathbb{E}[\delta]$ as the centered random variable. Now

$$\mathbb{E}[(\delta - \mathbb{E}[\delta])^k] = \mathbb{E}[\hat{\delta}^k].$$

So we focus on $\mathbb{E}[\hat{\delta}^k]$ in the remainder of the proof. For $k = 2$,

$$|\mathbb{E}[\hat{\delta}^2]| = \frac{1}{m^2} |\mathbb{E}[\sum_{i=1}^m \hat{\delta}_i^2]| = \frac{1}{m^2} \left| \mathbb{E} \left[\sum_i \hat{\delta}_i^2 + 2 \sum_{i < j} \hat{\delta}_i \hat{\delta}_j \right] \right| = \frac{\sigma_2}{m}.$$

For $k = 3$,

$$\begin{aligned} |\mathbb{E}[\hat{\delta}^3]| &= \frac{1}{m^3} |\mathbb{E}[\sum_{i=1}^m \hat{\delta}_i]^3| \\ &= \frac{1}{m^3} \left| \mathbb{E} \left[\sum_i \hat{\delta}_i^3 + 3 \sum_{i \neq j} \hat{\delta}_i^2 \hat{\delta}_j + 6 \sum_{i < j < k} \hat{\delta}_i \hat{\delta}_j \hat{\delta}_k \right] \right| \\ &= \frac{\sigma_3}{m^2}. \end{aligned}$$

For $k = 4$,

$$\begin{aligned} |\mathbb{E}[\hat{\delta}^4]| &= \frac{1}{m^4} |\mathbb{E}[\sum_{i=1}^m \hat{\delta}_i]^4| \\ &= \frac{1}{m^4} \left| \mathbb{E} \left[\sum_i \hat{\delta}_i^4 + 4 \sum_{i \neq j} \hat{\delta}_i^3 \hat{\delta}_j + 6 \sum_{i < j} \hat{\delta}_i^2 \hat{\delta}_j^2 + 24 \sum_{i < j < k < l} \hat{\delta}_i \hat{\delta}_j \hat{\delta}_k \hat{\delta}_l \right] \right| \\ &= \frac{1}{m^4} \left| m \mathbb{E}[\hat{\delta}_i^4] + 6 \frac{m(m-1)}{2} \mathbb{E}[\hat{\delta}_i^2] \mathbb{E}[\hat{\delta}_j^2] \right| \\ &= \frac{\sigma_4}{m^3} + \frac{3(m-1)\sigma_2^2}{m^3}. \end{aligned}$$

For $k = 5$,

$$\begin{aligned} |\mathbb{E}[\hat{\delta}^5]| &= \frac{1}{m^5} |\mathbb{E}[\sum_{i=1}^m \hat{\delta}_i]^5| \\ &= \frac{1}{m^5} \left| \mathbb{E} \left[\sum_i \hat{\delta}_i^5 + 10 \sum_{i \neq j} \hat{\delta}_i^3 \hat{\delta}_j^2 \right] \right| \\ &= \frac{1}{m^5} \left| m \mathbb{E}[\hat{\delta}_i^5] + 10m(m-1) \mathbb{E}[\hat{\delta}_i^3] \mathbb{E}[\hat{\delta}_j^2] \right| \\ &= \frac{\sigma_5}{m^4} + \frac{10(m-1)\sigma_3\sigma_2}{m^4}. \end{aligned}$$

For general $k > 0$, we expand the following term as a function of m

$$|\mathbb{E}[\hat{\delta}^k]| = \frac{1}{m^k} |\mathbb{E}[\sum_{i=1}^m \hat{\delta}_i]^k|.$$

As $\mathbb{E}[\hat{\delta}_i] = 0$ and $\hat{\delta}_i$ and $\hat{\delta}_j$ are independent for different i and j , the outcome has the following form

$$|\mathbb{E}[\hat{\delta}^k]| = \frac{1}{m^k} \mathcal{O} \left(\sum_{\substack{2a_2+3a_3+\dots+ka_k=k \\ a_i \geq 0 \ \forall i}} m^{\sum_{i=2}^k a_i} \sigma_2^{a_2} \sigma_3^{a_3} \dots \sigma_k^{a_k} \right) \quad (\text{E.2})$$

where $\sum_{i=2}^k a_i$ is the count of independent $\{\hat{\delta}_i\}$ used in $\sigma_2^{a_2} \sigma_3^{a_3} \dots \sigma_k^{a_k}$. Among the terms in (E.2), the dominating one in terms of m is one with largest $\sum_{i=2}^k a_i$, i.e.

$$|\mathbb{E}[\hat{\delta}^k]| = \begin{cases} \frac{1}{m^k} \mathcal{O}(m^{k/2}) \sigma_2^{k/2} & \text{if } k \text{ even,} \\ \frac{1}{m^k} \mathcal{O}(m^{\lfloor k/2 \rfloor}) \sigma_2^{\lfloor k/2 \rfloor - 1} \sigma_3 & \text{if } k \text{ odd.} \end{cases}$$

Then, we can simplify the upper right-hand side with

$$|\mathbb{E}[\hat{\delta}^k]| = \mathcal{O}(m^{-k+\lfloor k/2 \rfloor}),$$

which gives all the desired results. \square

Proposition E.1 (First-order Guarantee). *Assume that \mathcal{D}_m and $q(\cdot)$ satisfy Assumption B and C respectively with $k = 1$. Then, $\forall s \in \mathbb{R}$, $\left| \mathbb{E} \left[\mathcal{L}_{\mathcal{D}_m}^{(1)} q(s) \right] - q(s + \mathbb{E}[\delta]) \right| \leq a_2 \sigma_2 / (2m)$.*

Proof. Let $h = \mathbb{E}[\delta]$. If the function $q \in \mathcal{C}^2$, then the Taylor expansion at $s + h$ with remainders leads to

$$\mathbb{E}[q(s + \delta)] = q(s + h) + q'(s + h) \mathbb{E}[\delta - h] + \frac{1}{2} \mathbb{E}[q''(\phi_1)(\delta - h)^2]$$

where ϕ_1 between $s + h$ and $s + \delta$. Then the error of extrapolation becomes

$$|\mathbb{E}[q(s + \delta)] - q(s + h)| = \left| \frac{1}{2} \mathbb{E}[q''(\phi_1)(\delta - h)^2] \right| \leq \frac{a_2}{2} \mathbb{E}[(\delta - h)^2].$$

By Assumption C and Lemma E.1, we have that

$$|\mathbb{E}[q(s + \delta)] - q(s + h)| \leq \frac{a_2}{2} \mathbb{E}[(\delta - h)^2] = \frac{a_2}{2} \mathbb{E}[(\delta - h)^2] = \frac{a_2 \sigma_2}{2m}.$$

This completes the proof. \square

Proposition 6.1 (Second-order Guarantee). *Assume that distribution \mathcal{D}_m and $q(\cdot)$ satisfies Assumption B and C respectively with $k = 2$. Then, for all $s \in \mathbb{R}$, $\left| \mathbb{E} \left[\mathcal{L}_{\mathcal{D}_m}^{(2)} q(s) \right] - q(s + \mathbb{E}[\delta]) \right| \leq \frac{4a_3\sigma_3 + 9a_4\sigma_2^2}{48m^2} + \frac{5a_4}{96} \frac{\sigma_4 - 3\sigma_2^2}{m^3}$.*

Proof. Let $h = \mathbb{E}[\delta]$. If the function $q \in \mathcal{C}^4$, then the Taylor expansion at $s + h$ with remainders leads to

$$\begin{aligned} \mathbb{E}[q(s + \delta_1)] &= q(s + h) + q'(s + h) \mathbb{E}[\delta_1 - h] + \frac{q''(s+h)}{2} \mathbb{E}[(\delta_1 - h)^2] + \frac{q^{(3)}(s+h)}{6} \mathbb{E}[(\delta_1 - h)^3] \\ &\quad + \frac{1}{24} \mathbb{E}[q^{(4)}(\phi_1)(\delta_1 - h)^4] \\ \mathbb{E}[q(s + \delta_2)] &= q(s + h) + q'(s + h) \mathbb{E}[\delta_2 - h] + \frac{q''(s+h)}{2} \mathbb{E}[(\delta_2 - h)^2] + \frac{q^{(3)}(s+h)}{6} \mathbb{E}[(\delta_2 - h)^3] \\ &\quad + \frac{1}{24} \mathbb{E}[q^{(4)}(\phi_2)(\delta_2 - h)^4] \\ \mathbb{E}[q(s + \frac{\delta_1 + \delta_2}{2})] &= q(s + h) + q'(s + h) \mathbb{E}[\frac{\delta_1 + \delta_2}{2} - h] + \frac{q''(s+h)}{2} \mathbb{E}[(\frac{\delta_1 + \delta_2}{2} - h)^2] \\ &\quad + \frac{q^{(3)}(s+h)}{6} \mathbb{E}[(\frac{\delta_1 + \delta_2}{2} - h)^3] + \frac{1}{24} \mathbb{E}[q^{(4)}(\phi_3) (\frac{\delta_1 + \delta_2}{2} - h)^4] \end{aligned}$$

where ϕ_1, ϕ_2, ϕ_3 between $s + h$ and $s + \delta_1, s + \delta_2, s + \delta_3$ respectively.

As $\mathbb{E}[\delta - h] = 0$, the error of extrapolation becomes

$$\begin{aligned}
& |\mathbb{E}[\mathcal{L}_{\mathcal{D}_m}^2 q(s)] - q(s+h)| \\
& \leq \left| 2 \mathbb{E} \left[\frac{q^{(3)}(s+h)}{6} \left(\frac{\delta_1 + \delta_2}{2} - h \right)^3 \right] - \frac{1}{2} \left(\mathbb{E} \left[\frac{q^{(3)}(s+h)}{6} (\delta_1 - h)^3 \right] + \mathbb{E} \left[\frac{q^{(3)}(s+h)}{6} (\delta_2 - h)^3 \right] \right) \right| \\
& \quad + \left| 2 \mathbb{E} \left[\frac{q^{(4)}(\phi_3)}{24} \left(\frac{\delta_1 + \delta_2}{2} - h \right)^4 \right] - \frac{1}{2} \left(\mathbb{E} \left[\frac{q^{(4)}(\phi_1)}{24} (\delta_1 - h)^4 \right] + \mathbb{E} \left[\frac{q^{(4)}(\phi_2)}{24} (\delta_2 - h)^4 \right] \right) \right| \\
& \leq \frac{a_3}{6} \left| 2 \mathbb{E} \left[\left(\frac{\delta_1 + \delta_2}{2} - h \right)^3 \right] - \frac{1}{2} (\mathbb{E}[(\delta_1 - h)^3] + \mathbb{E}[(\delta_2 - h)^3]) \right| \\
& \quad + \left| 2 \mathbb{E} \left[\frac{q^{(4)}(\phi_3)}{24} \left(\frac{\delta_1 + \delta_2}{2} - h \right)^4 \right] - \frac{1}{2} \left(\mathbb{E} \left[\frac{q^{(4)}(\phi_1)}{24} (\delta_1 - h)^4 \right] + \mathbb{E} \left[\frac{q^{(4)}(\phi_2)}{24} (\delta_2 - h)^4 \right] \right) \right| \\
& \leq \frac{a_3}{6} \left| 2 \mathbb{E} \left[\left(\frac{\delta_1 + \delta_2}{2} - h \right)^3 \right] - \frac{1}{2} (\mathbb{E}[(\delta_1 - h)^3] + \mathbb{E}[(\delta_2 - h)^3]) \right| \\
& \quad + \left| 2 \mathbb{E} \left[\frac{|q^{(4)}(\phi_3)|}{24} \left(\frac{\delta_1 + \delta_2}{2} - h \right)^4 \right] + \frac{1}{2} \left(\mathbb{E} \left[\frac{|q^{(4)}(\phi_1)|}{24} (\delta_1 - h)^4 \right] + \mathbb{E} \left[\frac{|q^{(4)}(\phi_2)|}{24} (\delta_2 - h)^4 \right] \right) \right| \\
& \leq \frac{a_3}{6} \left| 2 \mathbb{E} \left[\left(\frac{\delta_1 + \delta_2}{2} - h \right)^3 \right] - \frac{1}{2} (\mathbb{E}[(\delta_1 - h)^3] + \mathbb{E}[(\delta_2 - h)^3]) \right| \\
& \quad + \frac{a_4}{24} \left| 2 \mathbb{E} \left[\left(\frac{\delta_1 + \delta_2}{2} - h \right)^4 \right] + \frac{1}{2} (\mathbb{E}[(\delta_1 - h)^4] + \mathbb{E}[(\delta_2 - h)^4]) \right|.
\end{aligned}$$

where the second inequality uses the upper bound on $q^{(3)}(\cdot)$ (Assumption C) and the third inequality uses $(\delta - h)^4$ is non-negative and the last inequality uses the uniform bound on $q^{(4)}(\cdot)$ (Assumption C). Then

$$\begin{aligned}
& |\mathbb{E}[\mathcal{L}_{\mathcal{D}_m}^2 q(s)] - q(s+h)| \\
& \leq \frac{a_3}{12} |\mathbb{E}[(\delta_1 - h)^3]| + \frac{a_4}{24} \left(2 \mathbb{E} \left[\left(\frac{\delta_1 + \delta_2}{2} - h \right)^4 \right] + \mathbb{E}[(\delta_1 - h)^4] \right) \\
& \leq \frac{a_3 \sigma_3}{12m^2} + \frac{a_4}{24} \left(\frac{\sigma_4}{4m^3} + \frac{3(2m-1)\sigma_2^2}{4m^3} + \frac{\sigma_4}{m^3} + \frac{3(m-1)\sigma_2^2}{m^3} \right) \\
& \leq \frac{a_3 \sigma_3}{12m^2} + \frac{a_4}{24} \left(\frac{9\sigma_2^2}{2m^2} + \frac{5(\sigma_4 - 3\sigma_2^2)}{4m^3} \right) \\
& \leq \frac{4a_3 \sigma_3 + 9a_4 \sigma_2^2}{48m^2} + \frac{5a_4}{96} \frac{\sigma_4 - 3\sigma_2^2}{m^3}.
\end{aligned}$$

we first use that $\mathbb{E}[(\delta_1 - h)^3] = \mathbb{E}[(\delta_2 - h)^3] = 4 \mathbb{E}[(\frac{\delta_1 + \delta_2}{2} - h)^3]$ and the uses the bound on moments in Lemma E.1. Note that $\mathbb{E} \left[\left(\frac{\delta_1 + \delta_2}{2} - h \right)^4 \right]$ can be seen as the 4th order moments of a batch size of $2m$.

□

Proposition E.2. Assume $q \in \mathcal{C}^6$. Then $\mathcal{L}_{\mathcal{D}_m}^{(3)}$ as defined below is a third-order extrapolation operator.

$$\mathcal{L}_{\mathcal{D}_m}^{(3)} q : s \mapsto \left(-\frac{1}{36} \mathcal{L}_{\mathcal{D}_m}^{(2)} + \frac{5}{9} \mathcal{L}_{\mathcal{D}_{2m}}^{(2)} - \frac{3}{4} \mathcal{L}_{\mathcal{D}_{3m}}^{(2)} - \frac{16}{9} \mathcal{L}_{\mathcal{D}_{4m}}^{(2)} + 3 \mathcal{L}_{\mathcal{D}_{6m}}^{(2)} \right) q(s).$$

Proof. Let $h = \mathbb{E}[\delta]$. If $q \in \mathcal{C}^{2k}$, then q has the following Taylor expansion

$$\begin{aligned} \mathbb{E}[q(s + \delta)] = & \underbrace{q(s + h)}_{\text{zero order term}} + q'(s + h) \mathbb{E}[\delta - h] + \underbrace{\frac{q''(s+h)}{2} \mathbb{E}[(\delta - h)^2]}_{\text{second order term}} + \dots \\ & + \frac{q^{(2k-1)}(s+h)}{(2k-1)!} \mathbb{E}[(\delta - h)^{2k-1}] + \frac{1}{2k!} \mathbb{E}[q^{(2k)}(\phi)(\delta - h)^{2k}]. \end{aligned}$$

Eliminate the third order term in the Taylor expansion. Consider the following affine combination which

$$\mathcal{F}_{\mathcal{D}_m}^{(3)} q : s \mapsto \alpha_1 \mathcal{L}_{\mathcal{D}_m}^{(2)} q(s) + \alpha_2 \mathcal{L}_{\mathcal{D}_{2m}}^{(2)} q(s).$$

We determine α_1 and α_2 by expanding $\mathcal{L}_{\mathcal{D}_m}^{(2)} q(s)$ and $\mathcal{L}_{\mathcal{D}_{2m}}^{(2)} q(s)$ and analyze the coefficients of terms:

- **(Affine).** Taylor expansion of $\mathcal{F}_{\mathcal{D}_m}^{(3)} q(s)$ at $s + h$ should have zero order term $q(s + h)$, i.e.

$$\alpha_1 q(s + h) + \alpha_2 q(s + h) = q(s + h).$$

- **(Eliminate third term).** Taylor expansion of $\mathcal{F}_{\mathcal{D}_m}^{(3)} q(s)$ at $s + h$ should have third order term $\mathbb{E}[(\delta - h)^3]$. That is,

$$\alpha_1 \mathbb{E}[(\delta_1 - h)^3] + \alpha_2 \mathbb{E}\left[\left(\frac{\delta_1 + \delta_2}{2} - h\right)^3\right] = 0.$$

This is equivalent to

$$\alpha_1 \mathbb{E}[(\delta_1 - h)^3] + \frac{\alpha_2}{4} \mathbb{E}[(\delta_1 - h)^3] = 0.$$

Therefore, α_1 and α_2 can be determined through the following linear system

$$\begin{aligned} \alpha_1 + \alpha_2 &= 1 \\ \alpha_1 + \frac{1}{4}\alpha_2 &= 0. \end{aligned}$$

The solution is $\alpha_1 = -\frac{1}{3}$ and $\alpha_2 = \frac{4}{3}$.

For $k = 3$ order extrapolation, consider the following

$$\mathcal{L}_{\mathcal{D}_m}^{(3)} q : s \mapsto \alpha'_1 \mathcal{F}_{\mathcal{D}_m}^{(3)} q(s) + \alpha'_2 \mathcal{F}_{\mathcal{D}_{2m}}^{(3)} q(s) + \alpha'_3 \mathcal{F}_{\mathcal{D}_{3m}}^{(3)} q(s).$$

We determine α'_1 , α'_2 and α'_3 by satisfying the following two conditions

- **(Affine).** Taylor expansion of $\mathcal{L}_{\mathcal{D}_m}^{(3)} q(s)$ at $s + h$ should have zero order term $q(x + h)$, i.e.

$$(\alpha'_1 + \alpha'_2 + \alpha'_3)q(x + h) = q(x + h).$$

- Taylor expansion of $\mathcal{L}_{\mathcal{D}_m}^{(3)} q(s)$ at $s + h$ should have 4th order term $\mathbb{E}[(\delta - h)^4]$. That is

$$\alpha'_1 \mathbb{E}[(\delta_1 - h)^4] + \alpha'_2 \mathbb{E} \left[\left(\frac{\delta_1 + \delta_2}{2} - h \right)^4 \right] + \alpha'_3 \mathbb{E} \left[\left(\frac{\delta_1 + \delta_2 + \delta_3}{3} - h \right)^4 \right] = 0.$$

This is equivalent to

$$\begin{aligned} \left(\alpha'_1 + \frac{\alpha'_2}{8} + \frac{\alpha'_3}{27} \right) \mathbb{E}[(\delta_1 - h)^4] &= 0 \\ \left(\frac{3}{8}\alpha'_2 + \frac{2}{9}\alpha'_3 \right) (\mathbb{E}[(\delta_1 - h)^2])^2 &= 0. \end{aligned}$$

Therefore, α'_1 , α'_2 and α'_3 can be determined through the following linear system

$$\begin{aligned} \alpha'_1 + \alpha'_2 + \alpha'_3 &= 1 \\ \alpha'_1 + \frac{1}{8}\alpha'_2 + \frac{1}{27}\alpha'_3 &= 0 \\ \alpha'_1 + \frac{3}{8}\alpha'_2 + \frac{2}{9}\alpha'_3 &= 0. \end{aligned}$$

The solution is $\alpha'_1 = \frac{1}{12}$, $\alpha'_2 = -\frac{4}{3}$ and $\alpha'_3 = \frac{9}{4}$. Then consider the Taylor expansion of $\mathcal{L}_{\mathcal{D}_m}^{(3)} q(s)$ at $s + h$ with (6.2), we can

$$|\mathbb{E}[\mathcal{L}_{\mathcal{D}_m}^{(3)} q(s)] - q(s + h)| \lesssim \left| q^{(5)}(s + h) \mathbb{E}[(\delta - h)^5] \right| + \left| \mathbb{E}[q^{(6)}(\phi_\delta)(\delta - h)^6] \right| \lesssim \mathcal{O}((a_5 + a_6)m^{-3})$$

where the first inequality uses the fact that $\mathcal{L}_{\mathcal{D}_m}^{(3)}$ is an affine mapping and the last inequality uses Lemma E.1. Therefore, $\mathcal{L}_{\mathcal{D}_m}^{(3)}$ is a 3rd-order extrapolation operator. We can expand it into

$$\begin{aligned} \mathcal{L}_{\mathcal{D}_m}^{(3)} q : s &\mapsto \frac{1}{12} \left(-\frac{1}{3}\mathcal{L}_{\mathcal{D}_m}^{(2)} q(s) + \frac{4}{3}\mathcal{L}_{\mathcal{D}_{2m}}^{(2)} q(s) \right) - \frac{4}{3} \left(-\frac{1}{3}\mathcal{L}_{\mathcal{D}_{2m}}^{(2)} q(s) + \frac{4}{3}\mathcal{L}_{\mathcal{D}_{4m}}^{(2)} q(s) \right) \\ &\quad + \frac{9}{4} \left(-\frac{1}{3}\mathcal{L}_{\mathcal{D}_{3m}}^{(2)} q(s) + \frac{4}{3}\mathcal{L}_{\mathcal{D}_{6m}}^{(2)} q(s) \right) \\ &= \left(-\frac{1}{36}\mathcal{L}_{\mathcal{D}_m}^{(2)} + \frac{5}{9}\mathcal{L}_{\mathcal{D}_{2m}}^{(2)} - \frac{3}{4}\mathcal{L}_{\mathcal{D}_{3m}}^{(2)} - \frac{16}{9}\mathcal{L}_{\mathcal{D}_{4m}}^{(2)} + 3\mathcal{L}_{\mathcal{D}_{6m}}^{(2)} \right) q(s). \end{aligned}$$

□

Lemma E.2 (Variance Bound). *Assume that $q : \mathbb{R}^p \rightarrow \mathbb{R}^\ell$ is in \mathcal{C}^4 and \mathcal{D}_m is the distribution in Assumption B. Suppose that the variance of $q(\mathbf{s} + \boldsymbol{\delta})$ is bounded as*

$$\mathbb{E}[\|q(\mathbf{s} + \boldsymbol{\delta}) - \mathbb{E}[q(\mathbf{s} + \boldsymbol{\delta})]\|_2^2] \leq \frac{V^2}{m} + C.$$

Then the variance of extrapolation $\mathcal{L}_{\mathcal{D}_m}^{(2)} q(\mathbf{s})$ is upper bounded by

$$\mathbb{E} \left[\left\| \mathcal{L}_{\mathcal{D}_m}^{(2)} q(\mathbf{s}) - \mathbb{E}[\mathcal{L}_{\mathcal{D}_m}^{(2)} q(\mathbf{s})] \right\|_2^2 \right] \leq 14 \left(\frac{V^2}{m} + C \right).$$

Proof. Let us use the definition of $\mathcal{L}_{\mathcal{D}_m}^{(2)} q(\mathbf{s})$:

$$\begin{aligned} & \mathbb{E} \left[\left\| \mathcal{L}_{\mathcal{D}_m}^{(2)} q(\mathbf{s}) - \mathbb{E}[\mathcal{L}_{\mathcal{D}_m}^{(2)} q(\mathbf{s})] \right\|_2^2 \right] \\ & \leq \mathbb{E} \left[\left\| 2q\left(\mathbf{s} + \frac{\boldsymbol{\delta}_1 + \boldsymbol{\delta}_2}{2}\right) - \frac{q(\mathbf{s} + \boldsymbol{\delta}_1) + q(\mathbf{s} + \boldsymbol{\delta}_2)}{2} - \mathbb{E} \left[2q\left(\mathbf{s} + \frac{\boldsymbol{\delta}_1 + \boldsymbol{\delta}_2}{2}\right) - \frac{q(\mathbf{s} + \boldsymbol{\delta}_1) + q(\mathbf{s} + \boldsymbol{\delta}_2)}{2} \right] \right\|_2^2 \right] \\ & \leq 3 \mathbb{E} \left[\left\| 2q\left(\mathbf{s} + \frac{\boldsymbol{\delta}_1 + \boldsymbol{\delta}_2}{2}\right) - \mathbb{E} \left[2q\left(\mathbf{s} + \frac{\boldsymbol{\delta}_1 + \boldsymbol{\delta}_2}{2}\right) \right] \right\|_2^2 \right] + 3 \mathbb{E} \left[\left\| \frac{q(\mathbf{s} + \boldsymbol{\delta}_1)}{2} - \mathbb{E} \left[\frac{q(\mathbf{s} + \boldsymbol{\delta}_1)}{2} \right] \right\|_2^2 \right] \\ & \quad + 3 \mathbb{E} \left[\left\| \frac{q(\mathbf{s} + \boldsymbol{\delta}_2)}{2} - \mathbb{E} \left[\frac{q(\mathbf{s} + \boldsymbol{\delta}_2)}{2} \right] \right\|_2^2 \right] \\ & \leq 12 \left(\frac{V^2}{2m} + C \right) + \frac{3}{4} \left(\frac{V^2}{m} + C \right) + \frac{3}{4} \left(\frac{V^2}{m} + C \right) \\ & = \frac{15V^2}{2m} + \frac{27C}{2}. \end{aligned}$$

This completes the proof. \square

E.4 Stationary Point Convergence Proofs from § 6.4 (CSO)

In this section, we provide the convergence proofs for the CSO problem. We start by establishing some helpful lemmas in § E.4.1. In § E.4.2, we reanalyze the BSGD algorithm to obtain explicit bias and variance bounds, which are then useful when we analyze E-BSGD in § E.4.3. Similarly, we reanalyze BSpiderBoost in § E.4.4 and use the resulting bias and variance bounds for the analysis of E-BSpiderBoost in § E.4.5.

Note that throughout our analyses, we define $\mathbb{E}^{t+1}[\cdot|t]$ as the expectation of randomness at time $t+1$ conditioning on the randomness until time t . When there is no ambiguity, we use $\mathbb{E}[\cdot]$ instead of $\mathbb{E}^{t+1}[\cdot|t]$.

E.4.1 Helpful Lemmas

Lemma E.3 (Sufficient Decrease). *Suppose Assumption I holds true and $\gamma \leq \frac{1}{2L_F}$ then*

$$\|\nabla F(\mathbf{x}^t)\|_2^2 \leq \frac{2(\mathbb{E}[F(\mathbf{x}^{t+1})] - F(\mathbf{x}^t))}{\gamma} + L_F \gamma \mathcal{E}_{var}^{t+1} + \mathcal{E}_{bias}^{t+1},$$

where $\mathbb{E}[\cdot]$ denote conditional expectation over the randomness at time t conditioned on all of the past randomness until time t .

Proof. In this proof, we use $\mathbb{E}[\cdot]$ to denote conditional expectation over the randomness at time t conditioned on all the past randomness until time t .

Let us expand $F(\mathbf{x}^{t+1})$ and apply the L_F -smoothness of F

$$\mathbb{E}[F(\mathbf{x}^{t+1})] \leq F(\mathbf{x}^t) - \gamma \mathbb{E}[\langle \nabla F(\mathbf{x}^t), G^{t+1} \rangle] + \frac{L_F \gamma^2}{2} \mathbb{E}[\|G^{t+1}\|_2^2].$$

Since $\mathbb{E}[\|G^{t+1}\|_2^2] = \mathbb{E}[\|G^{t+1} - \mathbb{E}[G^{t+1}]\|_2^2] + \|\mathbb{E}[G^{t+1}]\|_2^2 = \mathcal{E}_{\text{var}}^{t+1} + \|\mathbb{E}[G^{t+1}]\|_2^2$, then

$$\mathbb{E}[F(\mathbf{x}^{t+1})] \leq F(\mathbf{x}^t) - \gamma \mathbb{E}[\langle \nabla F(\mathbf{x}^t), G^{t+1} \rangle] + \frac{L_F \gamma^2}{2} (\mathcal{E}_{\text{var}}^{t+1} + \|\mathbb{E}[G^{t+1}]\|_2^2).$$

Expand the middle term with

$$\begin{aligned} -\gamma \mathbb{E}[\langle \nabla F(\mathbf{x}^t), G^{t+1} \rangle] &= -\frac{\gamma}{2} \|\nabla F(\mathbf{x}^t)\|_2^2 - \frac{\gamma}{2} \|\mathbb{E}[G^{t+1}]\|_2^2 + \frac{\gamma}{2} \|\nabla F(\mathbf{x}^t) - \mathbb{E}[G^{t+1}]\|_2^2 \\ &= -\frac{\gamma}{2} \|\nabla F(\mathbf{x}^t)\|_2^2 - \frac{\gamma}{2} \|\mathbb{E}[G^{t+1}]\|_2^2 + \frac{\gamma}{2} \mathcal{E}_{\text{bias}}^{t+1}. \end{aligned}$$

Combine with the inequality

$$\mathbb{E}[F(\mathbf{x}^{t+1})] \leq F(\mathbf{x}^t) - \frac{\gamma}{2} \|\nabla F(\mathbf{x}^t)\|_2^2 - \frac{\gamma}{2} (1 - L_F \gamma) \|\mathbb{E}[G^{t+1}]\|_2^2 + \frac{\gamma}{2} \mathcal{E}_{\text{bias}}^{t+1} + \frac{L_F \gamma^2}{2} \mathcal{E}_{\text{var}}^{t+1}.$$

By taking $\gamma \leq \frac{1}{2L_F}$, we have that

$$\mathbb{E}[F(\mathbf{x}^{t+1})] \leq F(\mathbf{x}^t) - \frac{\gamma}{2} \|\nabla F(\mathbf{x}^t)\|_2^2 - \frac{\gamma}{4} \|\mathbb{E}[G^{t+1}]\|_2^2 + \frac{\gamma}{2} \mathcal{E}_{\text{bias}}^{t+1} + \frac{L_F \gamma^2}{2} \mathcal{E}_{\text{var}}^{t+1}.$$

Re-arranging the terms we get the desired inequality. \square

A consequence of Lemma E.3 is the following result.

Lemma E.4 (Descent Lemma). *Suppose Assumption I holds true. By taking $\gamma \leq \frac{1}{2L_F}$, we have,*

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{x}^t)\|_2^2] + \frac{1}{2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\mathbb{E}^t[G^{t+1}|t]\|_2^2] \\ \leq \frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T} + \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{bias}}^{t+1}] + \frac{L_F \gamma}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{var}}^{t+1}] \end{aligned}$$

where the expectation is taken over all randomness from $t = 0$ to T .

Proof. We denote the conditional expectation at time t in the descent lemma (Lemma E.3) as $\mathbb{E}^{t+1}[\cdot|t]$ which conditions on all past randomness until time t . Then the descent lemma can be written as

$$\mathbb{E}^{t+1}[F(\mathbf{x}^{t+1})|t] \leq F(\mathbf{x}^t) - \frac{\gamma}{2} \|\nabla F(\mathbf{x}^t)\|_2^2 - \frac{\gamma}{4} \|\mathbb{E}^{t+1}[G^{t+1}|t]\|_2^2 + \frac{\gamma}{2} \mathbb{E}^{t+1}[\mathcal{E}_{\text{bias}}^{t+1}|t] + \frac{L_F \gamma^2}{2} \mathbb{E}^{t+1}[\mathcal{E}_{\text{var}}^{t+1}|t].$$

If we additionally consider the randomness at time $t - 1$, and apply $\mathbb{E}^t[\cdot|t - 1]$ to both sides

$$\begin{aligned}\mathbb{E}^t [\mathbb{E}^{t+1}[F(\mathbf{x}^{t+1})|t]|t - 1] &\leq \mathbb{E}^t[F(\mathbf{x}^t)|t - 1] - \frac{\gamma}{2} \mathbb{E}^t[\|\nabla F(\mathbf{x}^t)\|_2^2|t - 1] \\ &\quad - \mathbb{E}^t \left[\frac{\gamma}{4} \|\mathbb{E}^{t+1}[G^{t+1}|t]\|_2^2|t - 1 \right] + \frac{\gamma}{2} \mathbb{E}^t [\mathbb{E}^{t+1}[\mathcal{E}_{\text{bias}}^{t+1}|t]|t - 1] \\ &\quad + \frac{L_F\gamma^2}{2} \mathbb{E}^{t-1} [\mathbb{E}^{t+1}[\mathcal{E}_{\text{var}}^{t+1}|t]|t - 1].\end{aligned}$$

By the law of iterative expectations, we have $\mathbb{E}^t [\mathbb{E}^{t+1}[\cdot|t]|t - 1] = \mathbb{E}^t \mathbb{E}^{t+1} [\cdot|t - 1]$

$$\begin{aligned}\mathbb{E}^t[\mathbb{E}^{t+1} [F(\mathbf{x}^{t+1})|t - 1]] &\leq \mathbb{E}^t[F(\mathbf{x}^t)|t - 1] - \frac{\gamma}{2} \mathbb{E}^t[\|\nabla F(\mathbf{x}^t)\|_2^2|t - 1] \\ &\quad - \mathbb{E}^t \left[\frac{\gamma}{4} \|\mathbb{E}^{t+1}[G^{t+1}|t]\|_2^2|t - 1 \right] + \frac{\gamma}{2} \mathbb{E}^t [\mathbb{E}^{t+1} [\mathcal{E}_{\text{bias}}^{t+1}|t - 1]] \\ &\quad + \frac{L_F\gamma^2}{2} \mathbb{E}^t[\mathbb{E}^t [\mathcal{E}_{\text{var}}^{t+1}|t - 1]].\end{aligned}$$

Similarly, we can apply $\mathbb{E}^{t-1}[\cdot|t - 2]$, $\mathbb{E}^{t-2}[\cdot|t - 3]$, \dots , $\mathbb{E}^2[\cdot|1]$ and finally $\mathbb{E}^1[\cdot]$

$$\begin{aligned}\mathbb{E}^1 \dots [\mathbb{E}^{t+1} [F(\mathbf{x}^{t+1})]] &\leq \mathbb{E}^1 \dots [\mathbb{E}^t[F(\mathbf{x}^t)]] - \frac{\gamma}{2} \mathbb{E}^1 \dots [\mathbb{E}^t[\|\nabla F(\mathbf{x}^t)\|_2^2]] \\ &\quad - \mathbb{E}^1 \dots [\mathbb{E}^t[\frac{\gamma}{4} \|\mathbb{E}^{t+1}[G^{t+1}|t]\|_2^2]] + \frac{\gamma}{2} \mathbb{E}^1 \dots [\mathbb{E}^{t+1} [\mathcal{E}_{\text{bias}}^{t+1}]] \\ &\quad + \frac{L_F\gamma^2}{2} \mathbb{E}^1 \dots [\mathbb{E}^t [\mathcal{E}_{\text{var}}^{t+1}]].\end{aligned}$$

Now that both sides of the inequality have no randomness, we can simplify the notation by applying $\mathbb{E}^{t+1} \dots [\mathbb{E}^t[\cdot]]$ to both sides and by denoting

$$\mathbb{E}[\cdot] = \mathbb{E}^1 \dots [\mathbb{E}^{t+1}[\cdot]].$$

Then the descent lemma becomes

$$\mathbb{E}[F(\mathbf{x}^{t+1})] \leq \mathbb{E}[F(\mathbf{x}^t)] - \frac{\gamma}{2} \mathbb{E}[\|\nabla F(\mathbf{x}^t)\|_2^2] - \frac{\gamma}{4} \mathbb{E}[\|\mathbb{E}^{t+1}[G^{t+1}|t]\|_2^2] + \frac{\gamma}{2} \mathbb{E}[\mathcal{E}_{\text{bias}}^{t+1}] + \frac{L_F\gamma^2}{2} \mathbb{E}[\mathcal{E}_{\text{var}}^{t+1}].$$

Now we can sum the descent lemmas from $t = 0$ to $T - 1$

$$\begin{aligned}\sum_{t=0}^{T-1} \mathbb{E}[F(\mathbf{x}^{t+1})] &\leq \sum_{t=0}^{T-1} \mathbb{E}[F(\mathbf{x}^t)] - \frac{\gamma}{2} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{x}^t)\|_2^2] \\ &\quad - \frac{\gamma}{4} \sum_{t=0}^{T-1} \mathbb{E}[\|\mathbb{E}^{t+1}[G^{t+1}|t]\|_2^2] + \frac{\gamma}{2} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{bias}}^{t+1}] + \frac{L_F\gamma^2}{2} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{var}}^{t+1}].\end{aligned}$$

After simplification and division by T , we get

$$\begin{aligned}&\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{x}^t)\|_2^2] + \frac{1}{2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\mathbb{E}^{t+1}[G^{t+1}|t]\|_2^2] \\ &\leq \frac{2(\mathbb{E}[F(\mathbf{x}^T)] - \mathbb{E}[F(\mathbf{x}^0)])}{\gamma T} + \frac{\gamma}{2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{bias}}^{t+1}] + \frac{L_F\gamma^2}{2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{var}}^{t+1}] \\ &\leq \frac{2(\mathbb{E}[F(\mathbf{x}^T)] - F^*)}{\gamma T} + \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{bias}}^{t+1}] + \frac{L_F\gamma}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{var}}^{t+1}].\end{aligned}$$

□

The following corollary is a consequence of Proposition 6.1. Assume ∇f_ξ in CSO satisfies

$$a_l := \sup_{\mathbf{x}} \sup_{\xi} \|\nabla^{l+1} f_\xi(\mathbf{x})\|_2 < \infty, \quad l = 1, 2, 3, 4.$$

Let's further assume that the higher order moments of $g_\eta(\cdot)$ are bounded,

$$\sigma_k = \sup_{\mathbf{x}} \sup_{\xi} \mathbb{E}_{\eta|\xi} \left[\sum_{i=1}^p [g_\eta(\mathbf{x}) - \mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x})]]_i^k \right] < \infty, \quad k = 1, 2, 3, 4$$

where $[\cdot]_i$ refers to the i -th coordinate of a vector. Consider the $\mathcal{L}_{\mathcal{D}_{g,\xi}^{t+1}}^{(2)} \nabla f_\xi(0)$ defined in (6.6), then

$$\|\mathbb{E} \left[\mathcal{L}_{\mathcal{D}_{g,\xi}^{t+1}}^{(2)} \nabla f_\xi(0) \right] - \nabla f_\xi(\mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}^t)])\|_2^2 \leq \frac{C_e^2}{m^4} \quad \forall \xi,$$

where $C_e^2(f; g) := \left(\frac{8a_3\sigma_3 + 18a_4\sigma_2^2 + 5a_4\sigma_4}{96} \right)^2$.

Proof. The Proposition 6.1 gives the following upper bound

$$\|\mathbb{E} \left[\mathcal{L}_{\mathcal{D}_{g,\xi}^{t+1}}^{(2)} \nabla f_\xi(0) \right] - \nabla f_\xi(\mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}^t)])\|_2^2 \leq \left(\frac{4a_3\sigma_3 + 9a_4\sigma_2^2}{48m^2} + \frac{5a_4}{96} \frac{\sigma_4 - 3\sigma_2^2}{m^3} \right)^2.$$

For simplicity, we can relax the upper bound to

$$\|\mathbb{E} \left[\mathcal{L}_{\mathcal{D}_{g,\xi}^{t+1}}^{(2)} \nabla f_\xi(0) \right] - \nabla f_\xi(\mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}^t)])\|_2^2 \leq \frac{1}{m^4} \left(\frac{8a_3\sigma_3 + 18a_4\sigma_2^2 + 5a_4\sigma_4}{96} \right)^2.$$

□

E.4.2 Convergence of BSGD

In this section, we reanalyze the BSGD algorithm of [Hu et al., 2020b] to obtain bounds on bias and variance of its gradient estimates. Theorem E.3 shows that BSGD achieves an $\mathcal{O}(\epsilon^{-6})$ sample complexity.

Lemma E.5 (Bias and Variance of BSGD). *The bias and variance of BSGD are*

$$\mathcal{E}_{bias}^{t+1} \leq \frac{\sigma_{bias}^2}{m}, \quad \mathcal{E}_{var}^{t+1} \leq \frac{\sigma_{in}^2}{m} + \sigma_{out}^2$$

where $\sigma_{in}^2 = \zeta_g^2 C_f^2 + \sigma_g^2 C_g^2 L_f^2$, $\sigma_{out}^2 = C_F^2$, and $\sigma_{bias}^2 = \sigma_g^2 C_g^2 L_f^2$.

Proof. Denote $G^{t+1} = G_{BSGD}^{t+1}$ (6.5) and denote $\mathbb{E}[\cdot]$ as the conditional expectation $\mathbb{E}^{t+1}[\cdot|t]$ which conditions on all past randomness until time t . Note that the $\nabla g_{\tilde{\eta}}$ can be estimated without bias, i.e.

$$\mathbb{E}_{\tilde{\eta}|\xi} \left[\frac{1}{m} \sum_{\tilde{\eta} \in \tilde{H}_\xi} \nabla g_{\tilde{\eta}}(\mathbf{x}) \right] = \mathbb{E}_{\tilde{\eta}|\xi} [\nabla g_{\tilde{\eta}}(\mathbf{x})],$$

Then let's first look at the bias of BSGD

$$\begin{aligned}
\mathcal{E}_{\text{bias}}^{t+1} &= \|\nabla F(\mathbf{x}^{t+1}) - \mathbb{E}[G^{t+1}]\|_2^2 \\
&= \|\mathbb{E}_\xi \left[(\mathbb{E}_{\tilde{\eta}|\xi}[\nabla g_{\tilde{\eta}}(\mathbf{x}^t)])^\top \left(\nabla f_\xi(\mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}^t)]) - \mathbb{E}_{\eta|\xi}[\nabla f_\xi(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t))] \right) \right] \|^2 \\
&\leq C_g^2 \mathbb{E}_\xi \left[\|\nabla f_\xi(\mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}^t)]) - \mathbb{E}_{\eta|\xi}[\nabla f_\xi(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t))]\|_2^2 \right] \\
&\leq C_g^2 L_f^2 \mathbb{E}_\xi \left[\mathbb{E}_{\eta|\xi} \left[\|\mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}^t)] - \frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t)\|_2^2 \right] \right] \\
&\leq \frac{C_g^2 L_f^2}{m} \mathbb{E}_\xi \left[\mathbb{E}_{\eta|\xi} [\|\mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}^t)] - g_\eta(\mathbf{x}^t)\|_2^2] \right] \\
&= \frac{\sigma_g^2 C_g^2 L_f^2}{m} = \frac{\sigma_{\text{bias}}^2}{m}.
\end{aligned}$$

For the first inequality, we take the expectation outside the norm and bound $\nabla g_{\tilde{\eta}}$ with C_g .

On the other hand, the variance of BSGD can be decomposed into inner variance and outer variance

$$\begin{aligned}
\mathcal{E}_{\text{var}}^{t+1} &= \mathbb{E}_\xi [\mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi} [\|G^{t+1} - \mathbb{E}_\xi [\mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi} [G^{t+1}]]\|_2^2]] \\
&= \mathbb{E}_\xi [\mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi} [\| (G^{t+1} - \mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi} [G^{t+1}]) + (\mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi} [G^{t+1}] - \mathbb{E}_\xi [\mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi} [G^{t+1}]])\|_2^2]] \\
&= \underbrace{\mathbb{E}_\xi [\mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi} [\|G^{t+1} - \mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi} [G^{t+1}]]\|_2^2]}_{\text{Inner variance}} + \underbrace{\mathbb{E}_\xi [\|\mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi} [G^{t+1}] - \mathbb{E}_\xi [\mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi} [G^{t+1}]]\|_2^2]}_{\text{Outer variance}}.
\end{aligned}$$

The inner variance is bounded as follows

$$\begin{aligned}
&\mathbb{E}_\xi [\mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi} [\|G^{t+1} - \mathbb{E}_\xi [\mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi} [G^{t+1}]]\|_2^2]] \\
&= \mathbb{E}_\xi \left[\mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi} \left[\left\| \left(\frac{1}{m} \sum_{\tilde{\eta} \in \tilde{H}_\xi} \nabla g_{\tilde{\eta}}(\mathbf{x}^t) - \mathbb{E}_{\tilde{\eta}|\xi} [\nabla g_{\tilde{\eta}}(\mathbf{x}^t)] \right)^\top \nabla f_\xi \left(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t) \right) \right\|_2^2 \right] \right] \\
&\quad + \mathbb{E}_\xi \left[\mathbb{E}_{\eta|\xi} \left[\left\| (\mathbb{E}_{\tilde{\eta}|\xi} [\nabla g_{\tilde{\eta}}(\mathbf{x}^t)])^\top \left(\nabla f_\xi \left(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t) \right) - \mathbb{E}_{\eta|\xi} [\nabla f_\xi \left(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t) \right)] \right) \right\|_2^2 \right] \right] \\
&\leq C_f^2 \mathbb{E}_\xi \left[\mathbb{E}_{\tilde{\eta}|\xi} \left[\left\| \frac{1}{m} \sum_{\tilde{\eta} \in \tilde{H}_\xi} \nabla g_{\tilde{\eta}}(\mathbf{x}^t) - \mathbb{E}_{\tilde{\eta}|\xi} [\nabla g_{\tilde{\eta}}(\mathbf{x}^t)] \right\|_2^2 \right] \right] \\
&\quad + C_g^2 \mathbb{E}_\xi \left[\mathbb{E}_{\eta|\xi} \left[\left\| \nabla f_\xi \left(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t) \right) - \mathbb{E}_{\eta|\xi} [\nabla f_\xi \left(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t) \right)] \right\|_2^2 \right] \right] \\
&= C_f^2 \mathbb{E}_\xi \left[\mathbb{E}_{\tilde{\eta}|\xi} \left[\left\| \frac{1}{m} \sum_{\tilde{\eta} \in \tilde{H}_\xi} \nabla g_{\tilde{\eta}}(\mathbf{x}^t) - \mathbb{E}_{\tilde{\eta}|\xi} [\nabla g_{\tilde{\eta}}(\mathbf{x}^t)] \right\|_2^2 \right] \right] \\
&\quad + C_g^2 \mathbb{E}_\xi \left[\mathbb{E}_{\eta|\xi} \left[\left\| \nabla f_\xi \left(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t) \right) - \nabla f_\xi(\mathbb{E}_\eta[g_\eta(\mathbf{x}^t)]) \right\|_2^2 \right] \right] \\
&\quad - C_g^2 \mathbb{E}_\xi \left[\left\| \nabla f_\xi(\mathbb{E}_\eta[g_\eta(\mathbf{x}^t)]) - \mathbb{E}_{\eta|\xi} [\nabla f_\xi \left(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t) \right)] \right\|_2^2 \right] \\
&\leq \frac{\zeta_g^2 C_f^2}{m} + C_g^2 L_f \mathbb{E}_\xi \left[\mathbb{E}_{\eta|\xi} \left[\left\| \frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t) - \mathbb{E}_{\eta|\xi} [g_\eta(\mathbf{x}^t)] \right\|_2^2 \right] \right] \\
&\leq \frac{C_f^2 \zeta_g^2}{m} + \frac{C_g^2 L_f}{m} \mathbb{E}_\xi \left[\mathbb{E}_{\eta|\xi} [\|g_\eta(\mathbf{x}^t) - \mathbb{E}_{\eta|\xi} [g_\eta(\mathbf{x}^t)]\|_2^2] \right] \\
&\leq \frac{\zeta_g^2 C_f^2 + \sigma_g^2 C_g^2 L_f^2}{m} = \frac{\sigma_{\text{in}}^2}{m}.
\end{aligned}$$

The outer variance is independent of the inner batch size and can be bounded by

$$\mathbb{E}_\xi[\|\mathbb{E}_{\eta|\xi,\tilde{\eta}|\xi}[G^{t+1}] - \mathbb{E}_\xi[\mathbb{E}_{\eta|\xi,\tilde{\eta}|\xi}[G^{t+1}]]\|_2^2] \leq \mathbb{E}_\xi[\|\mathbb{E}_{\eta|\xi,\tilde{\eta}|\xi}[G^{t+1}]\|_2^2] \leq C_f^2 C_g^2 = C_F^2 = \sigma_{\text{out}}^2$$

Therefore, the variance is bounded as follows

$$\mathcal{E}_{\text{var}}^{t+1} \leq \frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2.$$

This completes the proof. \square

Theorem E.3 (BSGD Convergence). *Consider the (CSO) problem. Suppose Assumptions G, H, I holds true. Let step size $\gamma \leq 1/(2L_F)$. Then for BSGD, \mathbf{x}^s picked uniformly at random among $\{\mathbf{x}^t\}_{t=0}^{T-1}$ satisfies: $\mathbb{E}[\|\nabla F(\mathbf{x}^s)\|_2^2] \leq \epsilon^2$, for nonconvex F , if the inner batch size $m = \Omega(\sigma_{\text{bias}}^2 \epsilon^{-2})$ and the number of iterations $T = \Omega((F(\mathbf{x}^0) - F^*)L_F(\sigma_{\text{in}}^2/m + \sigma_{\text{out}}^2)\epsilon^{-4})$, where $\sigma_{\text{in}}^2 = \zeta_g^2 C_f^2 + \sigma_g^2 C_g^2 L_f^2$, $\sigma_{\text{out}}^2 = C_F^2$, and $\sigma_{\text{bias}}^2 = \sigma_g^2 C_g^2 L_f^2$.*

Proof. Denote $G^{t+1} = G_{\text{BSGD}}^{t+1}$ (6.1). Using descent lemma (Lemma E.4) and bias-variance bounds of BSGD (Lemma E.5)

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{x}^t)\|_2^2] + \frac{1}{2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\mathbb{E}^{t+1}[G^{t+1}|t]\|_2^2] \\ \leq \frac{2(\mathbb{E}[F(\mathbf{x}^T)] - F^*)}{\gamma T} + L_F \gamma (\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2) + \frac{\sigma_{\text{bias}}^2}{m} \end{aligned}$$

Then we can minimize the right-hand size by optimizing γ to

$$\gamma = \sqrt{\frac{2(F(\mathbf{x}^0) - F^*)}{L_F(\sigma_{\text{in}}^2/m + \sigma_{\text{out}}^2)T}}$$

which is smaller than the bound of step size $\gamma \leq \frac{1}{2L_F}$ if T is greater than the following constant which does not rely on the target precision ϵ

$$T \geq \frac{8L_F(F(\mathbf{x}^0) - F^*)}{\sigma_{\text{in}}^2/m + \sigma_{\text{out}}^2}.$$

Then the upper bound of gradient becomes

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{x}^t)\|_2^2] \leq \sqrt{\frac{2(F(\mathbf{x}^0) - F^*)L_F(\sigma_{\text{in}}^2/m + \sigma_{\text{out}}^2)}{T}} + \frac{\sigma_{\text{bias}}^2}{m}.$$

By taking inner batch size of at least

$$m \geq \frac{\sigma_{\text{bias}}^2}{\epsilon^2},$$

and iteration T greater than

$$T \geq \frac{2(F(\mathbf{x}^0) - F^*)L_F(\sigma_{\text{in}}^2/m + \sigma_{\text{out}}^2)}{\epsilon^4},$$

we have that

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\|\nabla F(\mathbf{x}^t)\|_2^2] \leq 2\epsilon^2.$$

By picking \mathbf{x}^s uniformly at random among $\{\mathbf{x}^t\}_{t=0}^{T-1}$, we get the desired guarantee. \square

The resulting sample complexity of BSGD to get to an ϵ -stationary point is $\mathcal{O}(\epsilon^{-6})$.

E.4.3 Convergence of E-BSGD

In this section, we analyze the sample complexity of Algorithm 16 (E-BSGD) for the CSO problem.

Lemma E.6 (Bias and Variance of E-BSGD). *The bias and variance of E-BSGD are*

$$\mathcal{E}_{\text{bias}}^{t+1} \leq \frac{\tilde{\sigma}_{\text{bias}}^2}{m^4}, \quad \mathcal{E}_{\text{var}}^{t+1} \leq 14\left(\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2\right)$$

where $\sigma_{\text{in}}^2 = \zeta_g^2 C_f^2 + \sigma_g^2 C_g^2 L_f^2$, $\sigma_{\text{out}}^2 = C_F^2$, and $\tilde{\sigma}_{\text{bias}}^2 = C_g^2 C_e^2$ with C_e^2 defined in § E.4.1.

Proof. Denote $G^{t+1} = G_{\text{E-BSGD}}^{t+1}$ (6.7). Like previously (Lemma E.5), let $\mathbb{E}[\cdot]$ denote the conditional expectation $\mathbb{E}^{t+1}[\cdot|t]$ which conditions on all past randomness until time t . In E-BSGD, we apply extrapolation to $\nabla f_\xi(\cdot)$. The bias can be estimated with the help of § E.4.1 as

$$\begin{aligned} \mathcal{E}_{\text{bias}}^{t+1} &= \|\nabla F(\mathbf{x}^{t+1}) - \mathbb{E}[G^{t+1}]\|_2^2 \\ &\leq C_g^2 \mathbb{E}_\xi \left[\|\nabla f_\xi(\mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}^t)]) - \mathbb{E} \left[\mathcal{L}_{\mathcal{D}_{g,\xi}^{t+1}}^{(2)} \nabla f_\xi(0) \right] \|_2^2 \right] \\ &\leq \frac{C_g^2 C_e^2}{m^4}. \end{aligned}$$

Since the variance of BSGD in Lemma E.5 is upper bounded by $\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2$, then Lemma E.2 gives

$$\mathcal{E}_{\text{var}}^{t+1} \leq 14\left(\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2\right).$$

This proves the claimed bounds. \square

Theorem 6.2. *[E-BSGD Convergence] Consider the (CSO) problem. Suppose Assumptions G, H, I, J hold true and $L_F, C_F, \tilde{L}_F, C_g, F^*$ are constants and $C_e(f; g) := \frac{8a_3\sigma_3 + 18a_4\sigma_2^2 + 5a_4\sigma_4}{96}$ defined in § E.4.1 are associated with second order extrapolation in the CSO problem. Let step*

size $\gamma \leq 1/(2L_F)$. Then the output \mathbf{x}^s of E-BSGD (Algorithm 16) satisfies: $\mathbb{E}[\|\nabla F(\mathbf{x}^s)\|_2^2] \leq \epsilon^2$, for nonconvex F , if the inner batch size $m = \Omega(C_e C_g \epsilon^{-1/2})$, and the number of iterations

$$T = \Omega(L_F(F(\mathbf{x}^0) - F^*)(\tilde{L}_F^2/m + C_F^2)\epsilon^{-4}).$$

Proof. The proof is very similar to Theorem E.3. Denote $G^{t+1} = G_{\text{E-BSGD}}^{t+1}$ (6.7). Using descent lemma (Lemma E.4) and bias-variance bounds of E-BSGD (Lemma E.6)

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{x}^t)\|_2^2] + \frac{1}{2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\mathbb{E}^{t+1}[G^{t+1}|t]\|_2^2] \\ \leq \frac{2(\mathbb{E}[F(\mathbf{x}^T)] - F^*)}{\gamma T} + 14L_F\gamma(\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2) + \frac{C_g^2 C_e^2}{m^4}. \end{aligned}$$

Then we optimize γ to

$$\gamma = \sqrt{\frac{(F(\mathbf{x}^0) - F^*)}{7L_F(\sigma_{\text{in}}^2/m + \sigma_{\text{out}}^2)T}}$$

which is smaller than the bound of step size $\gamma \leq \frac{1}{2L_F}$ if T is greater than the following constant which does not rely on the target precision ϵ

$$T \geq \frac{4L_F(F(\mathbf{x}^0) - F^*)}{7(\sigma_{\text{in}}^2/m + \sigma_{\text{out}}^2)}.$$

Then the gradient norm has the following upper bound.

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla F(\mathbf{x}^t)\|_2^2 \leq 4\sqrt{\frac{7(F(\mathbf{x}^0) - F^*)L_F(\sigma_{\text{in}}^2/m + \sigma_{\text{out}}^2)}{T}} + \frac{\tilde{\sigma}_{\text{bias}}^2}{m^4}.$$

In order to reach ϵ -stationary point, i.e.

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla F(\mathbf{x}^t)\|_2^2 \leq \epsilon^2,$$

we can enforce

$$4\sqrt{\frac{7(F(\mathbf{x}^0) - F^*)L_F(\sigma_{\text{in}}^2/m + \sigma_{\text{out}}^2)}{T}} \leq \epsilon^2, \quad \frac{C_g^2 C_e^2}{m^4} \leq \epsilon^2.$$

By taking inner batch size of at least

$$m = \Omega(\tilde{\sigma}_{\text{bias}}^{1/2} \epsilon^{-1/2}),$$

and iteration T greater than

$$T \geq \frac{112(F(\mathbf{x}^0) - F^*)L_F(\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2)}{\epsilon^4},$$

we have that

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla F(\mathbf{x}^t)\|_2^2 \leq 3\epsilon^2.$$

By picking \mathbf{x}^s uniformly at random among $\{\mathbf{x}^t\}_{t=0}^{T-1}$, we get the desired guarantee. \square

E.4.4 Convergence of BSpiderBoost

In this section, we reanalyze the BSpiderBoost algorithm of [Hu et al., 2020b] to obtain bounds on bias and variance of its gradient estimates. Theorem E.3 shows that BSpiderBoost achieves an $\mathcal{O}(\epsilon^{-5})$ sample complexity.

Let G_{BSB}^{t+1} as the BSpiderBoost gradient estimate

$$G_{\text{BSB}}^{t+1} = \begin{cases} G_{\text{BSB}}^t + \frac{1}{B_2} \sum_{\xi \in \mathcal{B}_2} (G_{\text{BSGD}}^{t+1} - G_{\text{BSGD}}^t) & \text{with prob. } 1 - p_{\text{out}} \\ \frac{1}{B_1} \sum_{\xi \in \mathcal{B}_1} G_{\text{BSGD}}^{t+1} & \text{with prob. } p_{\text{out}}. \end{cases}$$

Lemma E.7 (Bias and Variance of BSpiderBoost). *If $\gamma \leq \min\{\frac{1}{2L_F}, \frac{\sqrt{B_2}}{6L_F}\}$, then the bias and variance of BSpiderBoost are*

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{bias}}^{t+1}] &\leq \frac{2\sigma_{\text{bias}}^2}{m} + \frac{(1-p_{\text{out}})^3}{p_{\text{out}}B_2} \frac{56L_F^2\gamma^2}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\mathbb{E}^{t+1}[G^{t+1}|t]\|_2^2] + (\frac{1}{Tp_{\text{out}}} + 1) \frac{4(1-p_{\text{out}})^2}{B_1} (\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2) \\ \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{var}}^{t+1}] &\leq \frac{28(1-p_{\text{out}})L_F^2\gamma^2}{B_2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\mathbb{E}^{t+1}[G^{t+1}|t]\|_2^2] + (\frac{1}{T} + p_{\text{out}}) \frac{2}{B_1} (\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2), \end{aligned}$$

where $\sigma_{\text{in}}^2 = \zeta_g^2 C_f^2 + \sigma_g^2 C_g^2 L_f^2$, $\sigma_{\text{out}} = C_F^2$, and $\sigma_{\text{bias}}^2 = \sigma_g^2 C_g^2 L_f^2$.

Proof. Denote $G^{t+1} = G_{\text{BSB}}^{t+1}$ (6.8). Like previously (Lemma E.5), let $\mathbb{E}[\cdot]$ denote the conditional expectation $\mathbb{E}^{t+1}[\cdot|t]$ which conditions on all past randomness until time t . Denote G_L^{t+1} and G_S^{t+1} as the large batch and small batch in BSpiderBoost separately, i.e.,

$$\begin{cases} G_L^{t+1} = \frac{1}{B_1} \sum_{\xi \in \mathcal{B}_1} G_{\text{BSGD}}^{t+1} & \text{with prob. } p_{\text{out}} \\ G_S^{t+1} = G^t + \frac{1}{B_2} \sum_{\xi \in \mathcal{B}_2} (G_{\text{BSGD}}^{t+1} - G_{\text{BSGD}}^t) & \text{with prob. } 1 - p_{\text{out}}. \end{cases}$$

The bias of BSpiderBoost can be decomposed to its distance to BSGD and the distance from BSGD to the full gradient, i.e.,

$$\begin{aligned} \mathcal{E}_{\text{bias}}^{t+1} &= \|\nabla F(\mathbf{x}^{t+1}) - \mathbb{E}[G^{t+1}]\|_2^2 \\ &\leq 2\|\nabla F(\mathbf{x}^{t+1}) - \mathbb{E}[G_{\text{BSGD}}^{t+1}]\|_2^2 + 2\|\mathbb{E}[G_{\text{BSGD}}^{t+1}] - \mathbb{E}[G^{t+1}]\|_2^2 \\ &\leq \frac{2\sigma_{\text{bias}}^2}{m} + 2\|\mathbb{E}[G_{\text{BSGD}}^{t+1}] - \mathbb{E}[G^{t+1}]\|_2^2. \end{aligned} \tag{E.3}$$

where the last inequality uses the bias of BSGD from Lemma E.5. Then the second term can be bounded as follows

$$\begin{aligned}\|\mathbb{E}[G_{\text{BSGD}}^{t+1}] - \mathbb{E}[G^{t+1}]\|_2^2 &= (1 - p_{\text{out}})^2 \|\mathbb{E}[G_{\text{BSGD}}^{t+1}] - \mathbb{E}[G_S^{t+1}]\|_2^2 \\ &= (1 - p_{\text{out}})^2 \|\mathbb{E}[G_{\text{BSGD}}^t] - G^t\|_2^2.\end{aligned}$$

By taking the expectation of randomness of G^t

$$\begin{aligned}\|\mathbb{E}[G_{\text{BSGD}}^{t+1}] - \mathbb{E}[G^{t+1}]\|_2^2 &= (1 - p_{\text{out}})^2 (\|\mathbb{E}[G_{\text{BSGD}}^t] - \mathbb{E}[G^t]\|_2^2 + \mathbb{E}\|G^t - \mathbb{E}[G^t]\|_2^2) \\ &= (1 - p_{\text{out}})^2 (\|\mathbb{E}[G_{\text{BSGD}}^t] - \mathbb{E}[G^t]\|_2^2 + \mathcal{E}_{\text{var}}^t)\end{aligned}$$

Note that $\|\mathbb{E}[G_{\text{BSGD}}^1] - \mathbb{E}[G^1]\|_2^2 = 0$ as the first iteration always chooses the large batch. Then as we always use large batch at $t = 0$ we know that

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\mathbb{E}[G_{\text{BSGD}}^{t+1}] - \mathbb{E}[G^{t+1}]\|_2^2 \leq \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}. \quad (\text{E.4})$$

Therefore combine (E.3) and (E.4) we can upper bound the bias

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{bias}}^{t+1} \leq \frac{2\sigma_{\text{bias}}^2}{m} + \frac{2(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}. \quad (\text{E.5})$$

Variance. Now we consider the variance,

$$\begin{aligned}\mathcal{E}_{\text{var}}^{t+1} &= \mathbb{E} [\|G^{t+1} - \mathbb{E}[G^{t+1}]\|_2^2] \\ &\leq (1 - p_{\text{out}}) \mathbb{E} [\|G_S^{t+1} - \mathbb{E}[G_S^{t+1}]\|_2^2] + p_{\text{out}} \mathbb{E} [\|G_L^{t+1} - \mathbb{E}[G_L^{t+1}]\|_2^2] \\ &= \frac{(1-p_{\text{out}})}{B_2} \mathbb{E} [\|G_{\text{BSGD}}^{t+1} - G_{\text{BSGD}}^t - \mathbb{E}[G_{\text{BSGD}}^{t+1} - G_{\text{BSGD}}^t]\|_2^2] + \frac{p_{\text{out}}}{B_1} \mathbb{E} [\|G_{\text{BSGD}}^{t+1} - \mathbb{E}[G_{\text{BSGD}}^{t+1}]\|_2^2] \\ &\leq \frac{1-p_{\text{out}}}{B_2} \mathbb{E} [\|G_{\text{BSGD}}^{t+1} - G_{\text{BSGD}}^t - \mathbb{E}[G_{\text{BSGD}}^{t+1} - G_{\text{BSGD}}^t]\|_2^2] + \frac{p_{\text{out}}}{B_1} (\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2)\end{aligned} \quad (\text{E.6})$$

where the last equality is because the large batch in BSpiderBoost is similar to BSGD.

$$\mathcal{E}_{\text{var}}^1 = \mathbb{E} [\|G^1 - \mathbb{E}[G^1]\|_2^2] = \mathbb{E} [\|G_L^1 - \mathbb{E}[G_L^1]\|_2^2] = \frac{1}{B_1} \mathbb{E} [\|G_{\text{BSGD}}^1 - \mathbb{E}[G_{\text{BSGD}}^1]\|_2^2] \leq \frac{1}{B_1} (\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2). \quad (\text{E.7})$$

Finally, we expand the variance at small batch size epoch

$$\begin{aligned}&\mathbb{E} [\|G_{\text{BSGD}}^{t+1} - G_{\text{BSGD}}^t - \mathbb{E}[G_{\text{BSGD}}^{t+1} - G_{\text{BSGD}}^t]\|_2^2] \\ &= \underbrace{\mathbb{E} [\|G_{\text{BSGD}}^{t+1} - G_{\text{BSGD}}^t - \mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi}[G_{\text{BSGD}}^{t+1} - G_{\text{BSGD}}^t]\|_2^2]}_{\text{Inner variance } \mathcal{T}_{\text{in}}} \\ &\quad + \underbrace{\mathbb{E}_{\xi} [\|\mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi}[G_{\text{BSGD}}^{t+1} - G_{\text{BSGD}}^t] - \mathbb{E}[G_{\text{BSGD}}^{t+1} - G_{\text{BSGD}}^t]\|_2^2]}_{\text{Outer variance } \mathcal{T}_{\text{out}}}.\end{aligned}$$

The outer variance \mathcal{T}_{out} can be upper bounded as

$$\begin{aligned}
\mathcal{T}_{\text{out}} &\leq \mathbb{E}_{\xi} [\|\mathbb{E}_{\eta|\xi, \tilde{\eta}|\xi}[G_{\text{BSGD}}^{t+1} - G_{\text{BSGD}}^t]\|_2^2] \\
&= \mathbb{E}_{\xi} \left[\left\| (\mathbb{E}_{\tilde{\eta}|\xi}[\nabla g_{\tilde{\eta}}(\mathbf{x}^t)])^\top \mathbb{E}_{\eta|\xi}[\nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^t))] - (\mathbb{E}_{\tilde{\eta}|\xi}[\nabla g_{\tilde{\eta}}(\mathbf{x}^{t-1})])^\top \mathbb{E}_{\eta|\xi}[\nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^{t-1}))] \right\|_2^2 \right] \\
&\leq 2 \mathbb{E}_{\xi} \left[\left\| (\mathbb{E}_{\tilde{\eta}|\xi}[\nabla g_{\tilde{\eta}}(\mathbf{x}^t)] - \mathbb{E}_{\tilde{\eta}|\xi}[\nabla g_{\tilde{\eta}}(\mathbf{x}^{t-1})])^\top \mathbb{E}_{\eta|\xi}[\nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^t))] \right\|_2^2 \right] \\
&\quad + 2 \mathbb{E}_{\xi} \left[\left\| (\mathbb{E}_{\tilde{\eta}|\xi}[\nabla g_{\tilde{\eta}}(\mathbf{x}^{t-1})])^\top \mathbb{E}_{\eta|\xi}[\nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^t)) - \nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^{t-1}))] \right\|_2^2 \right] \\
&\leq 2L_g^2 C_f^2 \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 + 2C_g^4 L_f^2 \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 \\
&= 2L_F^2 \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 \\
&= 2L_F^2 \gamma^2 \|G^t\|_2^2.
\end{aligned}$$

The inner variance can be bounded by

$$\begin{aligned}
\mathcal{T}_{\text{in}} &\leq 4 \mathbb{E} \left[\left\| (\frac{1}{m} \sum_{\tilde{\eta} \in \tilde{H}_{\xi}} (\nabla g_{\tilde{\eta}}(\mathbf{x}^t) - \nabla g_{\tilde{\eta}}(\mathbf{x}^{t-1})) - \mathbb{E}_{\tilde{\eta}|\xi}[\nabla g_{\tilde{\eta}}(\mathbf{x}^t) - \nabla g_{\tilde{\eta}}(\mathbf{x}^{t-1})])^\top \nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^t)) \right\|_2^2 \right] \\
&\quad + 4 \mathbb{E} \left[\left\| (\mathbb{E}_{\tilde{\eta}|\xi}[\nabla g_{\tilde{\eta}}(\mathbf{x}^t) - \nabla g_{\tilde{\eta}}(\mathbf{x}^{t-1})])^\top (\nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^t)) - \mathbb{E}_{\eta|\xi}[\nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^t))]) \right\|_2^2 \right] \\
&\quad + 4 \mathbb{E} \left[\left\| \left((\frac{1}{m} \sum_{\tilde{\eta} \in \tilde{H}_{\xi}} \nabla g_{\tilde{\eta}}(\mathbf{x}^{t-1}))^\top \left(\nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^t)) - \nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^{t-1})) \right) \right. \right. \right. \\
&\quad \left. \left. \left. - \mathbb{E}_{\eta|\xi} \left[\nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^t)) - \nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^{t-1})) \right] \right) \right\|_2^2 \right] \\
&\quad + 4 \mathbb{E} \left[\left\| (\frac{1}{m} \sum_{\tilde{\eta} \in \tilde{H}_{\xi}} \nabla g_{\tilde{\eta}}(\mathbf{x}^{t-1}) - \mathbb{E}_{\tilde{\eta}|\xi}[\nabla g_{\tilde{\eta}}(\mathbf{x}^{t-1})])^\top (\nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^t)) - \nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^{t-1}))) \right\|_2^2 \right] \\
&\leq \frac{4C_f^2}{m} \mathbb{E} [\|\nabla g_{\tilde{\eta}}(\mathbf{x}^t) - \nabla g_{\tilde{\eta}}(\mathbf{x}^{t-1}) - \mathbb{E}_{\tilde{\eta}|\xi}[\nabla g_{\tilde{\eta}}(\mathbf{x}^t) - \nabla g_{\tilde{\eta}}(\mathbf{x}^{t-1})]\|_2^2] + \frac{4L_g^2 C_f^2}{m} \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 \\
&\quad + 4C_g^2 \mathbb{E} \left[\left\| \nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^t)) - \nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^{t-1})) \right. \right. \\
&\quad \left. \left. - \mathbb{E}_{\eta|\xi}[\nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^t)) - \nabla f_{\xi}(\frac{1}{m} \sum_{\eta \in H_{\xi}} g_{\eta}(\mathbf{x}^{t-1}))] \right\|_2^2 \right] \\
&\quad + \frac{4C_g^4 L_f^2}{m} \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2.
\end{aligned}$$

Then we have that

$$\begin{aligned}
\mathcal{T}_{\text{in}} &\leq \frac{4L_g^2 C_f^2}{m} \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 + \frac{4L_g^2 C_f^2}{m} \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 \\
&\quad + 4C_g^2 \mathbb{E} \left[\left\| \nabla f_\xi \left(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t) \right) - \nabla f_\xi \left(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^{t-1}) \right) - (\nabla f_\xi(\mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}^t)]) - \nabla f_\xi(\mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}^{t-1})])) \right\|_2^2 \right] \\
&\quad + 4C_g^2 \mathbb{E} \left[\left\| (\nabla f_\xi(\mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}^t)]) - \nabla f_\xi(\mathbb{E}_{\eta|\xi}[g_\eta(\mathbf{x}^{t-1})])) - \mathbb{E}_{\eta|\xi}[\nabla f_\xi \left(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t) \right) - \nabla f_\xi \left(\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^{t-1}) \right)] \right\|_2^2 \right] \\
&\quad + \frac{4C_g^4 L_f^2}{m} \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 \\
&\leq \frac{8L_g^2 C_f^2}{m} \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 + \frac{8C_g^4 L_f^2}{m} \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 + \frac{4C_g^4 L_f^2}{m} \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 \\
&\leq \frac{12L_F^2}{m} \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 \\
&= \frac{12L_F^2 \gamma^2}{m} \|G^t\|_2^2.
\end{aligned}$$

To sum up, the variance is bounded by

$$\begin{aligned}
\mathcal{E}_{\text{var}}^{t+1} &\leq \frac{2(1-p_{\text{out}})L_F^2 \gamma^2}{B_2} \left(1 + \frac{6}{m}\right) \|G^t\|_2^2 + \frac{p_{\text{out}}}{B_1} \left(\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2\right) \\
&\leq \frac{14(1-p_{\text{out}})L_F^2 \gamma^2}{B_2} \|G^t\|_2^2 + \frac{p_{\text{out}}}{B_1} \left(\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2\right) \\
&= \frac{14(1-p_{\text{out}})L_F^2 \gamma^2}{B_2} \mathbb{E}^t[\|G^t\|_2^2 | t-1] + \frac{p_{\text{out}}}{B_1} \left(\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2\right).
\end{aligned}$$

Then averaging over time and now we redefine $\mathbb{E}[\cdot] = \mathbb{E}^T \dots [\mathbb{E}^0[\cdot]]$

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{var}}^{t+1}] &\leq \frac{14(1-p_{\text{out}})L_F^2 \gamma^2}{B_2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] + \frac{\mathcal{E}_{\text{var}}^1}{T} + \frac{p_{\text{out}}}{B_1} \left(\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2\right) \\
&= \frac{14(1-p_{\text{out}})L_F^2 \gamma^2}{B_2} \frac{1}{T} \sum_{t=0}^{T-1} (\mathbb{E}[\mathcal{E}_{\text{var}}^{t+1}] + \mathbb{E}[\|\mathbb{E}^t[G^{t+1}|t]\|_2^2]) + \frac{\mathcal{E}_{\text{var}}^1}{T} + \frac{p_{\text{out}}}{B_1} \left(\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2\right).
\end{aligned}$$

If we take $\gamma \leq \frac{\sqrt{B_2}}{6L_F}$, then $\frac{14(1-p_{\text{out}})L_F^2 \gamma^2}{B_2} \leq \frac{1}{2}$, therefore

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{var}}^{t+1}] &\leq \frac{28(1-p_{\text{out}})L_F^2 \gamma^2}{B_2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\mathbb{E}^t[G^{t+1}|t]\|_2^2] + \frac{\mathcal{E}_{\text{var}}^1}{T} + \frac{2p_{\text{out}}}{B_1} \left(\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2\right) \\
&\stackrel{(\text{E.7})}{\leq} \frac{28(1-p_{\text{out}})L_F^2 \gamma^2}{B_2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\mathbb{E}^t[G^{t+1}|t]\|_2^2] + \left(\frac{1}{T} + p_{\text{out}}\right) \frac{2}{B_1} \left(\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2\right)
\end{aligned}$$

Then with (E.5), we can bound the bias by

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{bias}}^{t+1}] \leq \frac{2\sigma_{\text{bias}}^2}{m} + \frac{(1-p_{\text{out}})^3}{p_{\text{out}} B_2} \frac{56L_F^2 \gamma^2}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\mathbb{E}^t[G^{t+1}|t]\|_2^2] + \left(\frac{1}{T p_{\text{out}}} + 1\right) \frac{4(1-p_{\text{out}})^2}{B_1} \left(\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2\right)$$

□

Theorem E.4 (BSpiderBoost Convergence). *Consider the (CSO) problem. Suppose Assumptions G, H, I holds true. Let step size $\gamma \leq 1/(13L_F)$. Then for BSpiderBoost, \mathbf{x}^s picked*

uniformly at random among $\{\mathbf{x}^t\}_{t=0}^{T-1}$ satisfies: $\mathbb{E}[\|\nabla F(\mathbf{x}^s)\|_2^2] \leq \epsilon^2$, for nonconvex F , if the inner batch size $m = \Omega(\sigma_{\text{bias}}^2 \epsilon^{-2})$, the hyperparameters of the outer loop of BSpiderBoost are $B_1 = (\sigma_{\text{in}}^2/m + \sigma_{\text{out}}^2)\epsilon^{-2}$, $B_2 = \mathcal{O}(\epsilon^{-1})$, $p_{\text{out}} = 1/B_2$, and the number of iterations $T = \Omega(L_F(F(\mathbf{x}^0) - F^*)\epsilon^{-2})$, where $\sigma_{\text{in}}^2 = \zeta_g^2 C_f^2 + \sigma_g^2 C_g^2 L_f^2$, $\sigma_{\text{out}}^2 = C_F^2$, and $\sigma_{\text{bias}}^2 = \sigma_g^2 C_g^2 L_f^2$.

Proof. Denote $G^{t+1} = G_{\text{BSB}}^{t+1}$ (6.8). Using descent lemma (Lemma E.4) and bias-variance bounds of BSpiderBoost (Lemma E.7)

$$\begin{aligned} & \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{x}^t)\|_2^2] + \frac{1}{2T} \sum_{t=0}^{T-1} \mathbb{E}[\|\mathbb{E}^t[G^{t+1}|t]\|_2^2] \\ & \leq \frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T} + L_F \gamma \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{var}}^{t+1}] + \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{bias}}^{t+1}] \\ & \leq \frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T} + L_F \gamma \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{var}}^{t+1}] + \frac{2\sigma_{\text{bias}}^2}{m} + \frac{2}{p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{var}}^{t+1}] \\ & \leq \frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T} + \frac{2\sigma_{\text{bias}}^2}{m} + \frac{3}{p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\text{var}}^{t+1}] \end{aligned}$$

where the last inequality use $\gamma \leq \frac{1}{2L_F}$. Use the variance estimation of G^{t+1} and choose $B_2 p_{\text{out}} = 1$

$$\begin{aligned} & \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{x}^t)\|_2^2] + \frac{1}{2T} \sum_{t=0}^{T-1} \mathbb{E}[\|\mathbb{E}^t[G^{t+1}|t]\|_2^2] \\ & \leq \frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T} + \frac{2\sigma_{\text{bias}}^2}{m} + 84L_F^2 \gamma^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\mathbb{E}^t[G^{t+1}|t]\|_2^2] + (\frac{1}{Tp_{\text{out}}} + 1) \frac{6}{B_1} (\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2). \end{aligned}$$

Now we can let $\gamma \leq \frac{1}{13L_F}$ such that $84L_F^2 \gamma^2 \leq \frac{1}{2}$

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{x}^t)\|_2^2] \leq \frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T} + \frac{2\sigma_{\text{bias}}^2}{m} + (\frac{1}{Tp_{\text{out}}} + 1) \frac{6}{B_1} (\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2).$$

In order for the right-hand side to be ϵ^2 , the inner batch size

$$m \geq \frac{2\sigma_{\text{bias}}^2}{\epsilon^2},$$

and the outer batch size

$$B_1 = \frac{\sigma_{\text{in}}^2/m + \sigma_{\text{out}}^2}{\epsilon^2}, \quad B_2 = \sqrt{B_1}, \quad p_{\text{out}} = \frac{1}{B_2}.$$

The step size γ is upper bounded by $\min\{\frac{1}{2L_F}, \frac{\sqrt{B_2}}{6L_F}, \frac{1}{13L_F}\}$. As $B_2 \geq 1$, we can take $\gamma = \frac{1}{13L_F}$. So we need iteration T greater than

$$T \geq \frac{26L_F(F(\mathbf{x}^0) - F^*)}{\epsilon^2}.$$

By picking \mathbf{x}^s uniformly at random among $\{\mathbf{x}^t\}_{t=0}^{T-1}$, we get the desired guarantee. \square

The resulting sample complexity of BSpiderBoost to get to an ϵ -stationary point is $\mathcal{O}(\epsilon^{-5})$.

E.4.5 Convergence of E-BSpiderBoost

In this section, we analyze the sample complexity of Algorithm 17 (E-BSpiderBoost) for the CSO problem.

Lemma E.8 (Bias and Variance of E-BSpiderBoost). *The bias and variance of E-BSpiderBoost are*

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{var}^{t+1}] &\leq \frac{28(1-p_{out})L_F^2\gamma^2}{B_2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\mathbb{E}^t[G^{t+1}|t]\|_2^2] + \left(\frac{1}{Tp_{out}} + 1\right) \frac{28p_{out}}{B_1} \left(\frac{\sigma_{in}^2}{m} + \sigma_{out}^2\right) \\ \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{bias}^{t+1}] &\leq \frac{2\tilde{\sigma}_{bias}^2}{m^4} + \frac{2}{p_{out}} \frac{(1-p_{out})^2}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{var}^{t+1}], \end{aligned}$$

where $\sigma_{in}^2 := \zeta_g^2 C_f^2 + \sigma_g^2 C_g^2 L_f^2$, $\sigma_{out} = C_F^2$, and $\tilde{\sigma}_{bias}^2 = C_g^2 C_e^2$ with C_e^2 defined in § E.4.1.

Proof. Denote $G^{t+1} = G_{E-BSB}^{t+1}$ (6.9). Like previously (Lemma E.5), let $\mathbb{E}[\cdot]$ denote the conditional expectation $\mathbb{E}^t[\cdot|t]$ which conditions on all past randomness until time t . Let $G^{t+1} = G_{E-BSB}^{t+1}$ be the E-BSpiderBoost update. We expand the bias as follows

$$\begin{aligned} \mathcal{E}_{bias}^{t+1} &= \|\nabla F(\mathbf{x}^{t+1}) - \mathbb{E}[G^{t+1}]\|_2^2 \\ &\leq 2\|\nabla F(\mathbf{x}^{t+1}) - \mathbb{E}[G_{E-BSGD}^{t+1}]\|_2^2 + 2\|\mathbb{E}[G_{E-BSGD}^{t+1}] - \mathbb{E}[G^{t+1}]\|_2^2. \end{aligned}$$

From Lemma E.6, we know that

$$\|\nabla F(\mathbf{x}^{t+1}) - \mathbb{E}[G_{E-BSGD}^{t+1}]\|_2^2 \leq \frac{\tilde{\sigma}_{bias}^2}{m^4}.$$

The distance between $\mathbb{E}[G_{E-BSGD}^{t+1}]$ and $\mathbb{E}[G^{t+1}]$ can be bounded as follows.

$$\begin{aligned} \|\mathbb{E}[G_{E-BSGD}^{t+1}] - \mathbb{E}[G^{t+1}]\|_2^2 &= (1-p_{out})^2 \|\mathbb{E}[G_{E-BSGD}^{t+1}] - (G^t + \mathbb{E}[G_{E-BSGD}^{t+1} - G_{E-BSGD}^t])\|_2^2 \\ &= (1-p_{out})^2 \|\mathbb{E}[G_{E-BSGD}^t] - G^t\|_2^2 \end{aligned}$$

Taking expectation with respect to G^t

$$\|\mathbb{E}[G_{E-BSGD}^{t+1}] - \mathbb{E}[G^{t+1}]\|_2^2 \leq (1-p_{out})^2 (\|\mathbb{E}[G_{E-BSGD}^t] - \mathbb{E}[G^t]\|_2^2 + \|G^t - \mathbb{E}[G^t]\|_2^2).$$

where $\|\mathbb{E}[G_{E-BSGD}^1] - \mathbb{E}[G^1]\|_2^2 = 0$. By averaging over time we have

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\mathbb{E}[G_{E-BSGD}^{t+1}] - \mathbb{E}[G^{t+1}]\|_2^2 \leq \frac{1}{p_{out}} \frac{(1-p_{out})^2}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{var}^{t+1}].$$

Then the bias is bounded by

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{bias}^{t+1}] \leq \frac{2\tilde{\sigma}_{bias}^2}{m^4} + \frac{2}{p_{out}} \frac{(1-p_{out})^2}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{var}^{t+1}].$$

Variance. Since the extrapolation only gives a constant overhead given Lemma E.2

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\|G_{\text{BSB}}^{t+1} - \mathbb{E}^t[G_{\text{BSB}}^{t+1}|t]\|_2^2] \leq \frac{28(1-p_{\text{out}})L_F^2\gamma^2}{B_2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\|\mathbb{E}^t[G_{\text{BSB}}^{t+1}|t]\|_2^2] + (\frac{1}{T} + p_{\text{out}}) \frac{28}{B_1} (\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2).$$

Then the variance is bounded by

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\mathcal{E}_{\text{var}}^{t+1}] \leq \frac{28(1-p_{\text{out}})L_F^2\gamma^2}{B_2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\|\mathbb{E}^t[G^{t+1}|t]\|_2^2] + (\frac{1}{Tp_{\text{out}}} + 1) \frac{28p_{\text{out}}}{B_1} (\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2).$$

□

Theorem 6.3. *[E-BSpiderBoost Convergence] Consider the (CSO) problem under the same assumptions as Theorem 6.2. Let step size $\gamma \leq 1/(13L_F)$. Then the output \mathbf{x}^s of E-BSpiderBoost (Algorithm 17) satisfies: $\mathbb{E}[\|\nabla F(\mathbf{x}^s)\|_2^2] \leq \epsilon^2$, for nonconvex F , if the inner batch size $m = \mathcal{O}(C_e C_g \epsilon^{-0.5})$, the hyperparameters of the outer loop of E-BSpiderBoost $B_1 = (\tilde{L}_F^2/m + C_F^2)\epsilon^{-2}$, $B_2 = \sqrt{B_1}$, $p_{\text{out}} = 1/B_2$, and the number of iterations*

$$T = \Omega(L_F(F(\mathbf{x}^0) - F^*)\epsilon^{-2}).$$

Proof. Denote $G^{t+1} = G_{\text{E-BSB}}^{t+1}$ (6.9). Using descent lemma (Lemma E.4) and bias-variance bounds of E-BSpiderBoost (Lemma E.8)

$$\begin{aligned} & \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\|\nabla F(\mathbf{x}^t)\|_2^2] + \frac{1}{2T} \sum_{t=0}^{T-1} \mathbb{E} [\|\mathbb{E}^t[G^{t+1}|t]\|_2^2] \\ & \leq \frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T} + L_F \gamma \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\mathcal{E}_{\text{var}}^{t+1}] + \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\mathcal{E}_{\text{bias}}^{t+1}] \\ & \leq \frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T} + L_F \gamma \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\mathcal{E}_{\text{var}}^{t+1}] + \frac{2\tilde{\sigma}_{\text{bias}}^2}{m^4} + \frac{2}{p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\mathcal{E}_{\text{var}}^{t+1}] \\ & \leq \frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T} + \frac{2\tilde{\sigma}_{\text{bias}}^2}{m^4} + \frac{3}{p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\mathcal{E}_{\text{var}}^{t+1}] \end{aligned}$$

where the last inequality use $\gamma \leq \frac{1}{2L_F}$. Use the variance estimation of G^{t+1} and choose $B_2 p_{\text{out}} = 1$

$$\begin{aligned} & \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\|\nabla F(\mathbf{x}^t)\|_2^2] + \frac{1}{2T} \sum_{t=0}^{T-1} \mathbb{E} [\|\mathbb{E}^t[G^{t+1}|t]\|_2^2] \\ & \leq \frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T} + \frac{2\tilde{\sigma}_{\text{bias}}^2}{m^4} + 84L_F^2\gamma^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\|\mathbb{E}^t[G^{t+1}|t]\|_2^2] + (\frac{1}{Tp_{\text{out}}} + 1) \frac{84}{B_1} (\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2). \end{aligned}$$

Now we can let $\gamma \leq \frac{1}{13L_F}$ such that $84L_F^2\gamma^2 \leq \frac{1}{2}$

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\|\nabla F(\mathbf{x}^t)\|_2^2] \leq \frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T} + \frac{2\tilde{\sigma}_{\text{bias}}^2}{m^4} + (\frac{1}{Tp_{\text{out}}} + 1) \frac{84}{B_1} (\frac{\sigma_{\text{in}}^2}{m} + \sigma_{\text{out}}^2). \quad (\text{E.8})$$

In order to make the right-hand side ϵ^2 , the inner batch size

$$m = \Omega(\tilde{\sigma}_{\text{bias}}^2 \epsilon^{-0.5}),$$

and the outer batch size

$$B_1 = \frac{(\sigma_{\text{in}}^2/m + \sigma_{\text{out}}^2)}{\epsilon^2}, \quad B_2 = \sqrt{B_1}, \quad p_{\text{out}} = \frac{1}{B_2}.$$

The step size γ is upper bounded by $\min\{\frac{1}{2L_F}, \frac{\sqrt{B_2}}{6L_F}, \frac{1}{13L_F}\}$. As $B_2 \geq 1$, we can take $\gamma = \frac{1}{13L_F}$. So we need iteration T greater than

$$T \geq \frac{26L_F(F(\mathbf{x}^0) - F^*)}{\epsilon^2}.$$

By picking \mathbf{x}^s uniformly at random among $\{\mathbf{x}^t\}_{t=0}^{T-1}$, we get the desired guarantee. \square

E.5 Stationary Point Convergence Proofs from § 6.5 (FCCO)

In this section, we provide the convergence proofs for the FCCO problem. We start by analyzing a variant of BSpiderBoost (Algorithm 17) for this case in § E.5.1. In § E.5.2, we present a multi-level variance reduction approach (called NestedVR) that applies variance reduction in both outer (over the random variable i) and inner (over the random variable $\eta|i$) loops. In § E.5.3, we analyze E-NestedVR. As in the case of CSO analyses, our proofs go via bounds on bias and variance terms of these algorithms.

E.5.1 E-BSpiderBoost for FCCO problem

Theorem E.5. *Consider the (FCCO) problem. Suppose Assumptions G, H, I, J holds true. Let step size $\gamma = \mathcal{O}(1/L_F)$. Then the output of E-BSpiderBoost (Algorithm 17) satisfies: $\mathbb{E}[\|\nabla F(\mathbf{x}^s)\|_2^2] \leq \epsilon^2$, for nonconvex F , if the inner batch size $m = \Omega(\max\{C_e C_g \epsilon^{-1/2}, \sigma_{\text{in}}^2 n^{-1} \epsilon^{-2}\})$, the hyperparameters of the outer loop of E-BSpiderBoost $B_1 = n, B_2 = \sqrt{n}, p_{\text{out}} = 1/B_2$, and the number of iterations $T = \Omega(L_F(F(\mathbf{x}^0) - F^*)\epsilon^{-2})$. The resulting sample complexity is*

$$\mathcal{O}\left(L_F(F(\mathbf{x}^0) - F^*) \max\left\{\frac{\sqrt{n}C_e C_g}{\epsilon^{2.5}}, \frac{\sigma_{\text{in}}^2}{\sqrt{n}\epsilon^4}\right\}\right).$$

Remark 9. *The sample complexity depends on the relation between n and ϵ*

- When $n = \mathcal{O}(1)$, we have a complexity of $\mathcal{O}(\epsilon^{-4})$. This happens because we did not apply variance reduction for the inner loop.
- When $n = \Theta(\epsilon^{-2/3})$, E-BSpiderBoost has same performance as MSVR-V2 [Jiang et al., 2022] of $\mathcal{O}(n\epsilon^{-3}) = \mathcal{O}(\epsilon^{-11/3})$.
- When $n = \Theta(\epsilon^{-1.5})$, E-BSpiderBoost achieves a better sample complexity of $\mathcal{O}(\epsilon^{-3.25})$ than $\mathcal{O}(\epsilon^{-4.5})$ from MSVR-V2 [Jiang et al., 2022].
- When $n = \Theta(\epsilon^{-2})$, we recover $\mathcal{O}(\epsilon^{-3.5})$ sample complexity as in Theorem 6.3.

Proof. Denote $G^{t+1} = G_{\text{E-BSB}}^{t+1}$ (6.9). As we are using the finite-sum variant of SpiderBoost for the outer loop of the CSO problem, we only need to change the (E.6) and (E.7) to reflect that the outer variance is 0 now instead of $\frac{\sigma_{\text{out}}^2}{B_1}$ in the general CSO case. More concretely, we update (E.6) to

$$\begin{aligned} \mathcal{E}_{\text{var}}^{t+1} &= \mathbb{E}[\|G^{t+1} - \mathbb{E}[G^{t+1}]\|_2^2] \\ &\leq (1 - p_{\text{out}}) \mathbb{E}[\|G_S^{t+1} - \mathbb{E}[G_S^{t+1}]\|_2^2] + p_{\text{out}} \mathbb{E}[\|G_L^{t+1} - \mathbb{E}[G_L^{t+1}]\|_2^2] \\ &= \frac{(1-p_{\text{out}})}{B_2} \mathbb{E}[\|G_{\text{E-BSGD}}^{t+1} - G_{\text{E-BSGD}}^t - \mathbb{E}[G_{\text{E-BSGD}}^{t+1} - G_{\text{E-BSGD}}^t]\|_2^2] + \frac{p_{\text{out}}}{B_1} \mathbb{E}[\|G_{\text{E-BSGD}}^{t+1} - \mathbb{E}[G_{\text{E-BSGD}}^{t+1}]\|_2^2] \\ &\leq \frac{1-p_{\text{out}}}{B_2} \mathbb{E}[\|G_{\text{E-BSGD}}^{t+1} - G_{\text{E-BSGD}}^t - \mathbb{E}[G_{\text{E-BSGD}}^{t+1} - G_{\text{E-BSGD}}^t]\|_2^2] + \frac{p_{\text{out}}}{B_1} \frac{\sigma_{\text{in}}^2}{m}. \end{aligned} \quad (\text{E.9})$$

and change (E.7) to

$$\mathcal{E}_{\text{var}}^1 = \mathbb{E}[\|G^1 - \mathbb{E}[G^1]\|_2^2] = \mathbb{E}[\|G_L^1 - \mathbb{E}[G_L^1]\|_2^2] = \frac{1}{B_1} \mathbb{E}[\|G_{\text{E-BSGD}}^1 - \mathbb{E}[G_{\text{E-BSGD}}^1]\|_2^2] \leq \frac{1}{B_1} \frac{\sigma_{\text{in}}^2}{m}. \quad (\text{E.10})$$

Then our analysis only has to start from the updated version of (E.8)

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{x}^t)\|_2^2] \leq \frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T} + \frac{2\tilde{\sigma}_{\text{bias}}^2}{m^4} + \left(\frac{1}{Tp_{\text{out}}} + 1\right) \frac{84}{B_1} \frac{\sigma_{\text{in}}^2}{m}.$$

We would like all terms on the right-hand side to be bounded by ϵ^2 . From $\frac{2\tilde{\sigma}_{\text{bias}}^2}{m^4} \leq \epsilon^2$ we know that

$$m = \Omega\left(\frac{\tilde{\sigma}_{\text{bias}}^{1/2}}{\epsilon^{1/2}}\right).$$

From $\left(\frac{1}{Tp_{\text{out}}} + 1\right) \frac{84}{B_1} \frac{\sigma_{\text{in}}^2}{m} \leq \epsilon^2$, we know that

$$m = \Omega\left(\frac{\sigma_{\text{in}}^2}{n\epsilon^2}\right).$$

From $\frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T} \leq \epsilon^2$, we can choose that

$$\gamma = \mathcal{O}\left(\frac{1}{L_F}\right), \quad T = \Omega\left(\frac{L_F(F(\mathbf{x}^0) - F^*)}{\epsilon^2}\right).$$

Now the total sample complexity for E-BSpiderBoost for the FCCO problem becomes

$$B_2 m T = \mathcal{O}\left(L_F^2(F(\mathbf{x}^0) - F^*) \max\left\{\frac{\sqrt{n}\tilde{\sigma}_{\text{bias}}^{1/2}}{\epsilon^{2.5}}, \frac{\sigma_{\text{in}}^2}{\sqrt{n}\epsilon^4}\right\}\right).$$

By picking \mathbf{x}^s uniformly at random among $\{\mathbf{x}^t\}_{t=0}^{T-1}$, we get the desired guarantee. \square

E.5.2 Convergence of NestedVR

NestedVR Algorithm. We start by describing the NestedVR construction. We maintain states \mathbf{y}_i^{t+1} and \mathbf{z}_i^{t+1} to approximate

$$\mathbf{y}_i^{t+1} \approx \mathbb{E}_{\eta|i}[g_\eta(\mathbf{x}^t)], \quad \mathbf{z}_i^{t+1} \approx \mathbb{E}_{\tilde{\eta}|i}[\nabla g_{\tilde{\eta}}(\mathbf{x}^t)].$$

In iteration $t + 1$, if i is selected, then the state \mathbf{y}_i^{t+1} is updated as follows

$$\mathbf{y}_i^{t+1} = \begin{cases} \frac{1}{S_1} \sum_{\eta \in H_i} g_\eta(\mathbf{x}^t) & \text{with prob. } p_{\text{in}} \\ \mathbf{y}_i^t + \frac{1}{S_2} \sum_{\eta \in H_i} (g_\eta(\mathbf{x}^t) - g_\eta(\phi_i^t)) & \text{with prob. } 1 - p_{\text{in}}, \end{cases}$$

where ϕ_i^t is the last time node i is visited. If i is not selected, then

$$\mathbf{y}_i^{t+1} = \mathbf{y}_i^t.$$

In this case, \mathbf{y}_i^{t+1} was never used to compute $\nabla f_i(\mathbf{y}_i^{t+1})$ because i is not selected at the time $t + 1$. We use the following quantities

$$\hat{\mathbf{z}}_i^{t+1} = \mathbb{E}_{\tilde{\eta}|i}[\nabla g_{\tilde{\eta}}(\mathbf{x}^t)], \quad \mathbf{z}_i^{t+1} = \frac{1}{m} \sum_{\tilde{\eta} \in \tilde{H}_i} \nabla g_{\tilde{\eta}}(\mathbf{x}^t). \quad (\text{E.11})$$

We use G_{NVR}^{t+1} as the actual updates,

$$G_{\text{NVR}}^{t+1} = \begin{cases} \frac{1}{B_1} \sum_{i \in \mathcal{B}_1} (\mathbf{z}_i^{t+1})^\top \nabla f_i(\mathbf{y}_i^{t+1}) & \text{with prob. } p_{\text{out}} \\ G_{\text{NVR}}^t + \frac{1}{B_2} \sum_{i \in \mathcal{B}_2} ((\mathbf{z}_i^{t+1})^\top \nabla f_i(\mathbf{y}_i^{t+1}) - (\mathbf{z}_i^t)^\top \nabla f_i(\tilde{\mathbf{y}}_i^t)) & \text{with prob. } 1 - p_{\text{out}}. \end{cases}$$

We can also use the following quantity $\hat{G}_{\text{NVR}}^{t+1}$ as an auxiliary

$$\hat{G}_{\text{NVR}}^{t+1} = \begin{cases} \frac{1}{B_1} \sum_{i \in \mathcal{B}_1} (\hat{\mathbf{z}}_i^{t+1})^\top \nabla f_i(\mathbf{y}_i^{t+1}) & \text{with prob. } p_{\text{out}} \\ \hat{G}_{\text{NVR}}^t + \frac{1}{B_2} \sum_{i \in \mathcal{B}_2} ((\hat{\mathbf{z}}_i^{t+1})^\top \nabla f_i(\mathbf{y}_i^{t+1}) - (\hat{\mathbf{z}}_i^t)^\top \nabla f_i(\tilde{\mathbf{y}}_i^t)) & \text{with prob. } 1 - p_{\text{out}}. \end{cases}$$

Here we use $\tilde{\mathbf{y}}_i^t$ to represent an i.i.d. copy of \mathbf{y}_i^t where i is selected at time t .

The iterate \mathbf{x}^{t+1} is therefore updated

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \gamma G_{\text{NVR}}^{t+1}.$$

Lemma E.10. *The error between G_{NVR}^{t+1} and $\hat{G}_{\text{NVR}}^{t+1}$ can be upper bounded as follows*

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \left[\|G_{\text{NVR}}^{t+1} - \hat{G}_{\text{NVR}}^{t+1}\|_2^2 \right] \leq \frac{1}{B_1} \frac{C_f^2 \sigma_g^2}{m} + \frac{4(1-p_{\text{out}})}{B_2 m p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} (\mathbb{E}[\|G_i^{t+1} - G_i^t\|_2^2]).$$

Proof. In this proof, we ignore the subscript in G_{NVR}^{t+1} and $\hat{G}_{\text{NVR}}^{t+1}$, we bound the error between G^{t+1} and associated \hat{G}^{t+1} where

$$\begin{aligned} G_i^{t+1} &= (\frac{1}{m} \sum_{\tilde{\eta} \in \tilde{H}_i} \nabla g_{\tilde{\eta}}(\mathbf{x}))^\top \nabla f_i(\mathbf{y}_i^{t+1}), \\ \hat{G}_i^{t+1} &= (\mathbb{E}_{\tilde{\eta}|i}[\nabla g_{\tilde{\eta}}(\mathbf{x})])^\top \nabla f_i(\mathbf{y}_i^{t+1}). \end{aligned}$$

Let's only consider the expectation over the randomness of $\nabla g_{\tilde{\eta}}$,

$$\begin{aligned} \mathbb{E}_{H_i} [\|G_i^{t+1} - \hat{G}_i^{t+1}\|_2^2] &\leq \mathbb{E}_{\tilde{\eta}|i} [\|(\frac{1}{m} \sum_{\tilde{\eta} \in \tilde{H}_i} \nabla g_{\tilde{\eta}}(\mathbf{x}) - \mathbb{E}_{\tilde{\eta}|i}[\nabla g_{\tilde{\eta}}(\mathbf{x})])\|_2^2] \mathbb{E}[\|\nabla f_i(\mathbf{y}_i^{t+1})\|_2^2] \\ &\leq \frac{C_f^2}{m} \mathbb{E}_{\tilde{\eta}|i} [\|\nabla g_{\tilde{\eta}}(\mathbf{x}) - \mathbb{E}_{\tilde{\eta}|i}[\nabla g_{\tilde{\eta}}(\mathbf{x})]\|_2^2] \\ &\leq \frac{C_f^2 \sigma_g^2}{m}. \end{aligned}$$

Then we can bound the error as follows

$$\begin{aligned} \mathbb{E} [\mathbb{E}_{H_i} [\|G^{t+1} - \hat{G}^{t+1}\|_2^2]] &= \frac{p_{\text{out}}}{B_1} \mathbb{E} [\mathbb{E}_{H_i} [\|G_i^{t+1} - \hat{G}_i^{t+1}\|_2^2]] \\ &\quad + (1 - p_{\text{out}}) \left(\|G^t - \hat{G}^t\|_2^2 + \frac{1}{B_2} \mathbb{E} [\mathbb{E}_{H_i} [\|G_i^{t+1} - G_i^t - \hat{G}_i^{t+1} - \hat{G}_i^t\|_2^2]] \right) \\ &\leq \frac{p_{\text{out}}}{B_1} \frac{C_f^2 \sigma_g^2}{m} + (1 - p_{\text{out}}) \|G^t - \hat{G}^t\|_2^2 \\ &\quad + \frac{(1-p_{\text{out}})}{B_2 m} (\mathbb{E} [\|G_i^{t+1} - G_i^t\|_2^2]) \\ &\leq \frac{p_{\text{out}}}{B_1} \frac{C_f^2 \sigma_g^2}{m} + (1 - p_{\text{out}}) \|G^t - \hat{G}^t\|_2^2 + \frac{(1-p_{\text{out}})}{B_2 m} (\mathbb{E} [\|G_i^{t+1} - G_i^t\|_2^2]). \end{aligned}$$

Unroll the recursion gives

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\|G^{t+1} - \hat{G}^{t+1}\|_2^2] \leq \frac{1}{B_1} \frac{C_f^2 \sigma_g^2}{m} + \frac{4(1-p_{\text{out}})}{B_2 m p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\|G_i^{t+1} - G_i^t\|_2^2].$$

□

Lemma E.11 (Staleness). *Define the staleness of iterates at time t as $\Xi^t := \frac{1}{n} \sum_{j=1}^n \|\mathbf{x}^t - \phi_j^t\|_2^2$ and let G^{t+1} be the gradient estimate, then*

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\Xi^t] \leq \frac{6n^2}{B_2^2} \gamma^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2]. \quad (\text{E.12})$$

Proof. Like previously (Lemma E.5), let $\mathbb{E}[\cdot]$ denote the expectation conditioned on all previous randomness until $t-1$. It is clear that $\Xi^0 = 0$, so we only consider $t > 0$. We upper bound $\mathbb{E}[\Xi^t]$ as follows,

$$\mathbb{E}[\Xi^t] = (1 - p_{\text{out}}) \underbrace{\frac{1}{n} \sum_{j=1}^n \mathbb{E}[\|\mathbf{x}^t - \phi_j^t\|_2^2]}_{\text{if time } t \text{ takes } \mathcal{B}_2} + p_{\text{out}} \underbrace{\frac{1}{n} \sum_{j=1}^n \mathbb{E}[\|\mathbf{x}^t - \phi_j^t\|_2^2]}_{\text{if time } t \text{ takes } \mathcal{B}_1(\phi_j^t = \mathbf{x}^{t-1})}.$$

Then we can expand $\mathbb{E}[\Xi^t]$ as follows

$$\begin{aligned}\mathbb{E}[\Xi^t] &= \frac{1-p_{\text{out}}}{n} \sum_{j=1}^n \mathbb{E}[\|\mathbf{x}^t - \phi_j^t\|_2^2] + \frac{p_{\text{out}}}{n} \sum_{j=1}^n \mathbb{E}[\|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2] \\ &\leq \frac{1-p_{\text{out}}}{n} \sum_{j=1}^n \left((1 + \frac{1}{\beta}) \mathbb{E}_i[\|\mathbf{x}^{t-1} - \phi_j^t\|_2^2] + (1 + \beta) \|\mathbf{x}^{t-1} - \mathbf{x}^t\|_2^2 \right) + p_{\text{out}} \gamma^2 \mathbb{E}[\|G^t\|_2^2] \\ &\leq \frac{1}{n} \sum_{j=1}^n (1 + \frac{1}{\beta}) \mathbb{E}_i[\|\mathbf{x}^{t-1} - \phi_j^t\|_2^2] + (1 + \beta) \gamma^2 \mathbb{E}[\|G^t\|_2^2]\end{aligned}$$

where we use Cauchy-Schwarz inequality with coefficient $\beta > 0$. By the definition of ϕ_j^t ,

$$\begin{aligned}\mathbb{E}[\Xi^t] &\leq \frac{1}{n} \sum_{j=1}^n (1 + \frac{1}{\beta}) \left(\frac{n - B_2}{n} \|\mathbf{x}^{t-1} - \phi_j^{t-1}\|_2^2 + \frac{B_2}{n} \|\mathbf{x}^{t-1} - \mathbf{x}^{t-1}\|_2^2 \right) + (1 + \beta) \gamma^2 \mathbb{E}[\|G^t\|_2^2] \\ &= (1 + \frac{1}{\beta}) (1 - \frac{B_2}{n}) \Xi^{t-1} + (1 + \beta) \gamma^2 \mathbb{E}[\|G^t\|_2^2].\end{aligned}$$

By taking $\beta = 2n/B_2$, we have that $(1 + \frac{1}{\beta})(1 - \frac{B_2}{n}) \leq 1 - \frac{B_2}{2n}$ and thus

$$\mathbb{E}[\Xi^t] \leq (1 - \frac{B_2}{2n}) \Xi^{t-1} + (1 + \frac{2n}{B_2}) \gamma^2 \mathbb{E}[\|G^t\|_2^2].$$

Note that $\mathbb{E}[\Xi^0] = 0$.

$$\begin{aligned}\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\Xi^t] &\leq \frac{2n}{B_2} (1 + \frac{2n}{B_2}) \gamma^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] \\ &\leq \frac{6n^2}{B_2^2} \gamma^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2].\end{aligned}$$

□

The following lemma describes how the inner variable changes inside the variance.

Lemma E.12. Denote $\mathcal{E}_y^{t+1} := \mathbb{E}[\|\mathbf{y}_i^{t+1} - \mathbb{E}_{\eta|i}[g_\eta(\mathbf{x}^t)]\|_2^2]$ to be the error from inner variance and $p_{\text{out}}T \leq 1$. Then

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_y^{t+1} \leq \frac{(1-p_{\text{in}})C_g^2}{p_{\text{in}}S_2} \frac{1}{T} \sum_{t=0}^{T-1} \Xi^t + \frac{2\sigma_g^2}{S_1}.$$

Meanwhile, $\mathcal{E}_y^1 = \mathbb{E}[\|\mathbf{y}_i^1 - \mathbb{E}_{\eta|i}[g_\eta(\mathbf{x}^0)]\|_2^2] = \frac{\sigma_g^2}{S_1}$.

Proof.

$$\begin{aligned}\mathcal{E}_y^{t+1} &\leq p_{\text{in}} \frac{\sigma_g^2}{S_1} + (1 - p_{\text{in}}) \mathbb{E}_i[\mathbb{E}_{\eta|i}[\|\mathbf{y}_i^t - \mathbb{E}_{\eta|i}[g_\eta(\phi_i^t)]\|_2^2]] \\ &\quad + \frac{1-p_{\text{in}}}{S_2} \mathbb{E}_i[\mathbb{E}_{\eta|i}[\|g_\eta(\mathbf{x}^t) - g_\eta(\phi_i^t)\|_2^2]] \\ &\leq (1 - p_{\text{in}}) \mathcal{E}_y^t + \frac{(1-p_{\text{in}})C_g^2}{S_2} \mathbb{E}_i[\mathbb{E}_{\eta|i}[\|\mathbf{x}^t - \phi_i^t\|_2^2]] + p_{\text{in}} \frac{\sigma_g^2}{S_1}.\end{aligned}$$

As $t = 0$ always uses the large batch, $\mathcal{E}_y^1 = \mathbb{E}[\|\mathbf{y}_i^1 - \mathbb{E}_{\eta|i}[g_\eta(\mathbf{x}^0)]\|_2^2] = \frac{\sigma_g^2}{S_1}$. Then

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_y^{t+1} &\leq \frac{(1-p_{\text{in}})C_g^2}{p_{\text{in}}S_2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}_i[\mathbb{E}_{\eta|i}[\|\mathbf{x}^t - \phi_i^t\|_2^2]] + \frac{\sigma_g^2}{S_1} + \frac{\mathcal{E}_y^1}{p_{\text{in}}T} \\ &\leq \frac{(1-p_{\text{in}})C_g^2}{p_{\text{in}}S_2} \frac{1}{T} \sum_{t=0}^{T-1} \Xi^t + \frac{2\sigma_g^2}{S_1}. \end{aligned}$$

□

Lemma E.13. *The error $\mathbb{E}_i[\mathbb{E}_{p_{\text{in}}}[\mathbb{E}_{\eta|i}[\|\mathbf{y}_i^{t+1} - \tilde{\mathbf{y}}_i^t\|_2^2]]]$ satisfies*

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^{T-1} \mathbb{E}_i[\mathbb{E}_{p_{\text{in}}}[\mathbb{E}_{\eta|i}[\|\mathbf{y}_i^{t+1} - \tilde{\mathbf{y}}_i^t\|_2^2]]] &\leq \frac{4C_g^2\gamma^2}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] + \frac{4(1-p_{\text{in}})C_g^2}{S_2} \frac{1}{T} \sum_{t=0}^{T-1} \Xi^{t+1} \\ &\quad + \frac{6(1-p_{\text{in}})}{T} \sum_{t=0}^{T-1} \mathcal{E}_y^{t+1}. \end{aligned}$$

Note that when $p_{\text{in}} = 1$ and $S_1 = S_2 = m$, we recover the following

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^{T-1} \mathbb{E}_i[\mathbb{E}_{p_{\text{in}}}[\mathbb{E}_{\eta|i}[\|\mathbf{y}_i^{t+1} - \tilde{\mathbf{y}}_i^t\|_2^2]]] &= \frac{1}{T} \sum_{t=1}^{T-1} \mathbb{E}_i[\mathbb{E}_{p_{\text{in}}}[\mathbb{E}_{\eta|i}[\|\frac{1}{m} \sum_{\eta \in H_\xi} g_\eta(\mathbf{x}^t) - g_\eta(\mathbf{x}^{t-1})\|_2^2]]] \\ &\leq \frac{4C_g^2\gamma^2}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2]. \end{aligned}$$

Proof. For $t \geq 2$, $\mathbb{E}_i[\mathbb{E}_{p_{\text{in}}}[\mathbb{E}_{\eta|i}[\|\mathbf{y}_i^{t+1} - \tilde{\mathbf{y}}_i^t\|_2^2]]]$ can be upper bounded as follows

$$\begin{aligned} \mathbb{E}_i[\mathbb{E}_{p_{\text{in}}}[\mathbb{E}_{\eta|i}[\|\mathbf{y}_i^{t+1} - \tilde{\mathbf{y}}_i^t\|_2^2]]] &= p_{\text{in}} \mathbb{E}_i \left[\mathbb{E}_{\eta|i} \left[\left\| \frac{1}{S_1} \sum_{\eta \in H_i} (g_\eta(\mathbf{x}^t) - g_\eta(\mathbf{x}^{t-1})) \right\|_2^2 \right] \right] \\ &\quad + (1-p_{\text{in}}) \mathbb{E}_i \left[\mathbb{E}_{\eta|i} \left[\left\| \mathbf{y}_i^t - \mathbf{y}_i^{t-1} + \frac{1}{S_2} \sum_{\eta \in H_i} (g_\eta(\mathbf{x}^t) - g_\eta(\phi_i^t)) - (g_\eta(\mathbf{x}^{t-1}) - g_\eta(\phi_i^{t-1})) \right\|_2^2 \right] \right] \\ &\leq p_{\text{in}} C_g^2 \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 + \frac{1-p_{\text{in}}}{S_2} \mathbb{E}_i \left[\mathbb{E}_{\eta|i} \left[\left\| (g_\eta(\mathbf{x}^t) - g_\eta(\phi_i^t)) - (g_\eta(\mathbf{x}^{t-1}) - g_\eta(\phi_i^{t-1})) \right\|_2^2 \right] \right] \\ &\quad + 3(1-p_{\text{in}}) (\mathbb{E}_i [\|\mathbf{y}_i^t - \mathbb{E}_{\eta|i}[g_\eta(\phi_i^t)]\|_2^2] + \mathbb{E}_i [\|\mathbf{y}_i^{t-1} - \mathbb{E}_{\eta|i}[g_\eta(\phi_i^{t-1})]\|_2^2] + C_g^2 \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2) \\ &\leq p_{\text{in}} C_g^2 \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 + \frac{2(1-p_{\text{in}})C_g^2}{S_2} (\Xi^t + \Xi^{t-1}) + 3(1-p_{\text{in}}) (\mathcal{E}_y^t + \mathcal{E}_y^{t-1} + C_g^2 \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2) \\ &\leq (p_{\text{in}} + 3(1-p_{\text{in}})) C_g^2 \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 + \frac{2(1-p_{\text{in}})C_g^2}{S_2} (\Xi^t + \Xi^{t-1}) + 3(1-p_{\text{in}}) (\mathcal{E}_y^t + \mathcal{E}_y^{t-1}). \end{aligned}$$

For $t = 1$, we choose $\tilde{\mathbf{y}}_i^1 = \mathbf{y}_i^1$

$$\begin{aligned} \mathbb{E}_i[\mathbb{E}_{p_{\text{in}}}[\mathbb{E}_{\eta|i}[\|\mathbf{y}_i^2 - \tilde{\mathbf{y}}_i^1\|_2^2]]] &= p_{\text{in}} \mathbb{E}_i \left[\mathbb{E}_{\eta|i} \left[\left\| \frac{1}{S_1} \sum_{\eta \in H_i} (g_\eta(\mathbf{x}^1) - g_\eta(\mathbf{x}^0)) \right\|_2^2 \right] \right] \\ &\quad + (1-p_{\text{in}}) \mathbb{E}_i \left[\mathbb{E}_{\eta|i} \left[\left\| \mathbf{y}_i^1 - \frac{1}{S_2} \sum_{\eta \in H_i} (g_\eta(\mathbf{x}^1) - g_\eta(\mathbf{x}^0)) - \tilde{\mathbf{y}}_i^1 \right\|_2^2 \right] \right] \\ &\leq C_g^2 \|\mathbf{x}^1 - \mathbf{x}^0\|_2^2. \end{aligned}$$

Then for summing up $t = 1$ to $T - 1$

$$\begin{aligned}
& \sum_{t=2}^{T-1} \mathbb{E}_i[\mathbb{E}_{p_{\text{in}}}[\mathbb{E}_{\eta|i}[\|\mathbf{y}_i^{t+1} - \tilde{\mathbf{y}}_i^t\|_2^2]]] + \mathbb{E}_i[\mathbb{E}_{p_{\text{in}}}[\mathbb{E}_{\eta|i}[\|\mathbf{y}_i^2 - \tilde{\mathbf{y}}_i^1\|_2^2]]] \\
& \leq (p_{\text{in}} + 3(1 - p_{\text{in}}))C_g^2 \sum_{t=2}^{T-1} \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 + \frac{2(1-p_{\text{in}})C_g^2}{S_2} \left(\sum_{t=2}^{T-1} \Xi^t + \sum_{t=2}^{T-1} \Xi^{t-1} \right) \\
& \quad + 3(1 - p_{\text{in}}) \left(\sum_{t=2}^{T-1} \mathcal{E}_y^t + \sum_{t=2}^{T-1} \mathcal{E}_y^{t-1} \right) + C_g^2 \|\mathbf{x}^1 - \mathbf{x}^0\|_2^2 \\
& \leq 4C_g^2 \sum_{t=1}^{T-1} \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 + \frac{4(1-p_{\text{in}})C_g^2}{S_2} \sum_{t=0}^{T-1} \Xi^{t+1} + 6(1 - p_{\text{in}}) \sum_{t=0}^{T-1} \mathcal{E}_y^{t+1}.
\end{aligned}$$

Finally, the error has the following upper bound

$$\begin{aligned}
& \frac{1}{T} \sum_{t=1}^{T-1} \mathbb{E}_i[\mathbb{E}_{p_{\text{in}}}[\mathbb{E}_{\eta|i}[\|\mathbf{y}_i^{t+1} - \tilde{\mathbf{y}}_i^t\|_2^2]]] \\
& \leq \frac{4C_g^2\gamma^2}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] + \frac{4(1-p_{\text{in}})C_g^2}{S_2} \frac{1}{T} \sum_{t=0}^{T-1} \Xi^{t+1} + \frac{6(1-p_{\text{in}})}{T} \sum_{t=0}^{T-1} \mathcal{E}_y^{t+1}.
\end{aligned}$$

□

Lemma E.14 (Bias and Variance of NestedVR). *If the step size γ satisfies,*

$$\gamma^2 L_F^2 \max \left\{ \frac{(1-p_{\text{in}})}{p_{\text{in}}S_2} \frac{18}{B_2}, \frac{1-p_{\text{out}}}{B_2} \frac{(1-p_{\text{in}})}{p_{\text{in}}S_2} \frac{18n^2}{B_2^2}, \frac{(1-p_{\text{out}})}{B_2} \right\} \leq \frac{1}{16} \cdot \frac{1}{6}$$

then the variance and bias of NestedVR are

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1} & \leq 32 \left(\left(\frac{p_{\text{out}}}{B_1} + \frac{1-p_{\text{out}}}{B_2} \right) \frac{(1-p_{\text{in}})}{p_{\text{in}}S_2} \frac{18n^2}{B_2^2} + \frac{(1-p_{\text{out}})}{B_2} \right) \frac{\gamma^2 L_F^2}{T} \sum_{t=0}^{T-1} \|\mathbb{E}[G^{t+1}]\|_2^2 \\
& \quad + 96 \left(\frac{p_{\text{out}}}{B_1} + \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} \right) \frac{\tilde{L}_F^2}{S_1} + \frac{(1-p_{\text{out}})}{T} \frac{8\tilde{L}_F^2}{B_1 S_1} \\
\frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{bias}}^{t+1} & \leq \frac{12(1-p_{\text{in}})}{p_{\text{in}}S_2} \frac{n^2}{B_2^2} \frac{L_F^2 \gamma^2}{T} \sum_{t=0}^{T-1} \|\mathbb{E}[G^{t+1}]\|_2^2 + \frac{4\tilde{L}_F^2}{S_1} \\
& \quad + \left(\frac{12(1-p_{\text{in}})}{p_{\text{in}}S_2} \frac{n^2}{B_2^2} L_F^2 \gamma^2 + \frac{2(1-p_{\text{out}})^2}{p_{\text{out}}} \right) \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}.
\end{aligned}$$

Proof. Notations. Let us define the following terms,

$$G_i^{t+1} := (\mathbf{z}_i^{t+1})^\top \nabla f_i(\mathbf{y}_i^{t+1}), \quad \tilde{G}_i^t := (\mathbf{z}_i^t)^\top \nabla f_i(\tilde{\mathbf{y}}_i^t).$$

Note that the \tilde{G}^t computed at time $t + 1$ has same expectation as G^t

$$\mathbb{E}^{t+1}[\tilde{G}^t|t] = \mathbb{E}^t[G^t|t-1]. \tag{E.13}$$

Computing the bias. First consider the two cases in the outer loop

$$\begin{aligned}
\mathcal{E}_{\text{bias}}^{t+1} & = \|\nabla F(\mathbf{x}^t) - \mathbb{E}^{t+1}[G^{t+1}|t]\|_2^2 \\
& \leq 2 \underbrace{\|\nabla F(\mathbf{x}^t) - \mathbb{E}^{t+1}[G_i^{t+1}|t]\|_2^2}_{A_1^{t+1}} + 2 \underbrace{\|\mathbb{E}^{t+1}[G_i^{t+1}|t] - \mathbb{E}^{t+1}[G^{t+1}|t]\|_2^2}_{A_2^{t+1}}.
\end{aligned}$$

We expand A_2^{t+1} as follows

$$\begin{aligned}
A_2^{t+1} &= \|\mathbb{E}^{t+1}[G_i^{t+1}|t] - \mathbb{E}^{t+1}[G^{t+1}|t]\|_2^2 \\
&= \|\mathbb{E}^{t+1}[G_i^{t+1}|t] - p_{\text{out}} \mathbb{E}^{t+1}[G_i^{t+1}|t] - (1 - p_{\text{out}})(G^t + \mathbb{E}^{t+1}[G_i^{t+1} - \tilde{G}_i^t|t])\|_2^2 \\
&= (1 - p_{\text{out}})^2 \|G^t - \mathbb{E}^{t+1}[\tilde{G}_i^t|t]\|_2^2 \\
&= (1 - p_{\text{out}})^2 \|G^t - \mathbb{E}^t[G_i^t|t - 1]\|_2^2
\end{aligned}$$

where we use (E.13) in the last equality. Now we take expectation with respect to randomness at t such that G^t is a random variable, then

$$\begin{aligned}
A_2^{t+1} &= (1 - p_{\text{out}})^2 \mathbb{E}^t [\|G^t - \mathbb{E}^t[G_i^t|t - 1]\|_2^2 | t - 1] \\
&= (1 - p_{\text{out}})^2 (\|\mathbb{E}^t[G^t|t - 1] - \mathbb{E}^t[G_i^t|t - 1]\|_2^2 + \mathcal{E}_{\text{var}}^t) \\
&= (1 - p_{\text{out}})^2 (A_2^t + \mathcal{E}_{\text{var}}^t)
\end{aligned}$$

while at initialization we always use large batch

$$A_2^1 = \|\mathbb{E}^1[G_i^1] - \mathbb{E}^1[G^1]\|_2^2 = \|\mathbb{E}^1[G_i^1] - \mathbb{E}^1[G_i^1]\|_2^2 = 0.$$

Therefore, when we average over time t

$$\frac{1}{T} \sum_{t=0}^{T-1} A_2^{t+1} \leq \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}. \quad (\text{E.14})$$

On the other hand, let us consider the upper bound on A_1^{t+1}

$$A_1^{t+1} \leq C_g^2 L_f^2 \mathbb{E}[\|\mathbf{y}_i^{t+1} - \mathbb{E}_{\eta|i}[g_\eta(\mathbf{x}^t)]\|_2^2] = C_g^2 L_f^2 \mathcal{E}_y^{t+1}.$$

From Lemma E.12 we know that

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^{T-1} A_1^{t+1} &\leq C_g^2 L_f^2 \left(\frac{(1-p_{\text{in}})C_g^2}{p_{\text{in}}S_2} \frac{1}{T} \sum_{t=0}^{T-1} \Xi^t + \frac{2\sigma_g^2}{S_1} \right) \\
&\leq \frac{(1-p_{\text{in}})L_F^2}{p_{\text{in}}S_2} \frac{1}{T} \sum_{t=0}^{T-1} \Xi^t + \frac{2\tilde{L}_F^2}{S_1}.
\end{aligned}$$

From Lemma E.11 we know that

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^{T-1} A_1^{t+1} &\leq \frac{(1-p_{\text{in}})L_F^2}{p_{\text{in}}S_2} \left(\frac{6n^2}{B_2^2} \gamma^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] \right) + \frac{2\tilde{L}_F^2}{S_1} \\
&= \frac{6(1-p_{\text{in}})}{p_{\text{in}}S_2} \frac{n^2}{B_2^2} \frac{L_F^2 \gamma^2}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] + \frac{2\tilde{L}_F^2}{S_1} + \frac{6(1-p_{\text{in}})}{p_{\text{in}}S_2} \frac{n^2}{B_2^2} \frac{L_F^2 \gamma^2}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}.
\end{aligned}$$

Therefore, the bias has the following bound

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{bias}}^{t+1} &\leq \frac{12(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{n^2}{B_2^2} \frac{L_F^2 \gamma^2}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] + \frac{4\tilde{L}_F^2}{S_1} \\ &\quad + \left(\frac{12(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{n^2}{B_2^2} L_F^2 \gamma^2 + \frac{2(1-p_{\text{out}})^2}{p_{\text{out}}} \right) \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}. \end{aligned} \quad (\text{E.15})$$

Note that when $p_{\text{in}} = 1$ and $S_1 = S_2 = m$, then this bias recovers BSpiderBoost in (E.5)

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{bias}}^{t+1} \leq \frac{4\tilde{L}_F^2}{m} + \frac{2(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}.$$

Computing the variance. Let us decompose the variance into 3 parts:

$$\begin{aligned} \mathcal{E}_{\text{var}}^{t+1} &= \mathbb{E}[\|G^{t+1} - \mathbb{E}[G^{t+1}]\|_2^2] \\ &= \mathbb{E}[\|G^{t+1} \pm \hat{G}^{t+1} \pm \mathbb{E}_{\eta|i}[\hat{G}^{t+1}] - \mathbb{E}_i[\mathbb{E}_{\eta|i}[\hat{G}^{t+1}]]\|_2^2] \\ &= \underbrace{\mathbb{E}[\|G^{t+1} - \hat{G}^{t+1}\|_2^2]}_{\mathcal{E}_{\nabla g}^{t+1}} + \underbrace{\mathbb{E}_i[\|\mathbb{E}_{\eta|i}[G^{t+1}] - \mathbb{E}_i[\mathbb{E}_{\eta|i}[G^{t+1}]]\|_2^2]}_{\mathcal{E}_{\text{var,out}}^{t+1}} + \underbrace{\mathbb{E}[\|G^{t+1} - \mathbb{E}_{\eta|i}[G^{t+1}]\|_2^2]}_{\mathcal{E}_{\text{var,in}}^{t+1}} \end{aligned}$$

where $\mathcal{E}_{\text{var,out}}^{t+1}$ and $\mathcal{E}_{\text{var,in}}^{t+1}$ are the variance of outer loop and inner loop.

Inner Variance. For $t \geq 1$, we expand the inner variance

$$\begin{aligned} \mathcal{E}_{\text{var,in}}^{t+1} &= p_{\text{out}} \mathbb{E} \left[\left\| \frac{1}{B_1} \sum_i (\mathbb{E}_{\tilde{\eta}|i}[\nabla g_{\tilde{\eta}}(\mathbf{x}^t)])^\top (\nabla f_i(\mathbf{y}_i^{t+1}) - \mathbb{E}_{\eta|i}[\nabla f_i(\mathbf{y}_i^{t+1})]) \right\|_2^2 \right] \\ &\quad + (1 - p_{\text{out}}) \mathbb{E} \left[\left\| \frac{1}{B_2} \sum_i (G_i^{t+1} - \tilde{G}_i^t) - \mathbb{E}_{\eta|i}[G_i^{t+1} - \tilde{G}_i^t] \right\|_2^2 \right] \\ &\leq \frac{p_{\text{out}}}{B_1} C_g^2 \mathbb{E}[\|\nabla f_i(\mathbf{y}_i^{t+1}) - \mathbb{E}_{\eta|i}[\nabla f_i(\mathbf{y}_i^{t+1})]\|_2^2] + \frac{1-p_{\text{out}}}{B_2} \mathbb{E}_i \left[\mathbb{E}_{\eta|i} \left[\|G_i^{t+1} - \tilde{G}_i^t\|_2^2 \right] \right] \\ &\leq \frac{p_{\text{out}}}{B_1} 4C_g^2 L_f^2 \mathcal{E}_y^{t+1} + \frac{1-p_{\text{out}}}{B_2} \mathbb{E}_i \left[\mathbb{E}_{p_{\text{in}}} \left[\mathbb{E}_{\eta|i} \left[\|G_i^{t+1} - \tilde{G}_i^t\|_2^2 \right] \right] \right]. \end{aligned} \quad (\text{E.16})$$

We bound the outer variance as

$$\begin{aligned} \mathbb{E}_i \left[\mathbb{E}_{p_{\text{in}}} \left[\mathbb{E}_{\eta|i} \left[\|G_i^{t+1} - \tilde{G}_i^t\|_2^2 \right] \right] \right] &= \mathbb{E}_i \left[\mathbb{E}_{p_{\text{in}}} \left[\mathbb{E}_{\eta|i} \left[\|G_i^{t+1} \pm (\mathbb{E}_{\tilde{\eta}|i}[\nabla g_{\tilde{\eta}}(\mathbf{x}^{t-1})])^\top \nabla f_i(\mathbf{y}_i^{t+1}) - \tilde{G}_i^t\|_2^2 \right] \right] \right] \\ &\leq 2 \mathbb{E}_i \left[\mathbb{E}_{p_{\text{in}}} \left[\mathbb{E}_{\eta|i} \left[\|G_i^{t+1} - (\mathbb{E}_{\tilde{\eta}|i}[\nabla g_{\tilde{\eta}}(\mathbf{x}^{t-1})])^\top \nabla f_i(\mathbf{y}_i^{t+1})\|_2^2 \right] \right] \right] \\ &\quad + 2 \mathbb{E}_i \left[\mathbb{E}_{p_{\text{in}}} \left[\mathbb{E}_{\eta|i} \left[\|(\mathbb{E}_{\tilde{\eta}|i}[\nabla g_{\tilde{\eta}}(\mathbf{x}^{t-1})])^\top \nabla f_i(\mathbf{y}_i^{t+1}) - \tilde{G}_i^t\|_2^2 \right] \right] \right] \\ &\leq 2C_f^2 L_g^2 \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 + 2C_g^2 L_f^2 \mathbb{E}_i[\mathbb{E}_{p_{\text{in}}}[\mathbb{E}_{\eta|i}[\|\mathbf{y}_i^{t+1} - \tilde{\mathbf{y}}_i^t\|_2^2]]]. \end{aligned} \quad (\text{E.17})$$

For $t = 0$, as we only use large and small batch in the

$$\begin{aligned}
\mathcal{E}_{\text{var,in}}^1 &= \mathbb{E} \left[\left\| \frac{1}{B_1} \sum_i (\mathbb{E}_{\tilde{\eta}|i} [\nabla g_{\tilde{\eta}}(\mathbf{x}^0)])^\top (\nabla f_i(\mathbf{y}_i^1) - \mathbb{E}_{\eta|i} [\nabla f_i(\mathbf{y}_i^1)]) \right\|_2^2 \right] \\
&\leq \frac{1}{B_1} C_g^2 \mathbb{E} [\| \nabla f_i(\mathbf{y}_i^1) - \mathbb{E}_{\eta|i} [\nabla f_i(\mathbf{y}_i^1)] \|_2^2] \\
&\leq \frac{1}{B_1} 4C_g^2 L_f^2 \mathcal{E}_y^1 \\
&\leq \frac{1}{B_1} 4C_g^2 L_f^2 \frac{\sigma_g^2}{S_1} \\
&\leq \frac{4\tilde{L}_F^2}{B_1 S_1}.
\end{aligned} \tag{E.18}$$

Therefore, average over time $t = 0, \dots, T-1$ gives

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var,in}}^{t+1} &\leq \frac{p_{\text{out}}}{B_1} 4C_g^2 L_f^2 \frac{1}{T} \sum_{t=1}^{T-1} \mathcal{E}_y^{t+1} + \frac{1-p_{\text{out}}}{B_2} \frac{1}{T} \sum_{t=1}^{T-1} \mathbb{E}_i [\mathbb{E}_{p_{\text{in}}} [\mathbb{E}_{\eta|i} [\|G_i^{t+1} - \tilde{G}_i^t\|_2^2]]] + \frac{\mathcal{E}_{\text{var,in}}^1}{T} \\
&= \frac{p_{\text{out}}}{B_1} 4C_g^2 L_f^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_y^{t+1} + \frac{1-p_{\text{out}}}{B_2} \frac{1}{T} \sum_{t=1}^{T-1} \mathbb{E}_i \left[\mathbb{E}_{p_{\text{in}}} \left[\mathbb{E}_{\eta|i} \left[\|G_i^{t+1} - \tilde{G}_i^t\|_2^2 \right] \right] \right] + \frac{(1-p_{\text{out}})\mathcal{E}_{\text{var,in}}^1}{T} \\
&\leq \frac{p_{\text{out}}}{B_1} 4C_g^2 L_f^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_y^{t+1} + \frac{(1-p_{\text{out}})\mathcal{E}_{\text{var,in}}^1}{T} \\
&\quad + \frac{2(1-p_{\text{out}})}{B_2} \left(\frac{C_f^2 L_g^2}{T} \sum_{t=1}^{T-1} \|\mathbf{x}^t - \mathbf{x}^{t-1}\|_2^2 + C_g^2 L_f^2 \frac{1}{T} \sum_{t=1}^{T-1} \mathbb{E}_i [\mathbb{E}_{p_{\text{in}}} [\mathbb{E}_{\eta|i} [\|\mathbf{y}_i^{t+1} - \tilde{\mathbf{y}}_i^t\|_2^2]]] \right) \\
&\leq \frac{p_{\text{out}}}{B_1} 4C_g^2 L_f^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_y^{t+1} + \frac{(1-p_{\text{out}})\mathcal{E}_{\text{var,in}}^1}{T} \\
&\quad + \frac{2(1-p_{\text{out}})}{B_2} \frac{C_f^2 L_g^2 \gamma^2}{T} \sum_{t=0}^{T-1} \mathbb{E} [\|G^{t+1}\|_2^2] \\
&\quad + \frac{2(1-p_{\text{out}})}{B_2} C_g^2 L_f^2 \frac{1}{T} \sum_{t=1}^{T-1} \mathbb{E}_i [\mathbb{E}_{p_{\text{in}}} [\mathbb{E}_{\eta|i} [\|\mathbf{y}_i^{t+1} - \tilde{\mathbf{y}}_i^t\|_2^2]]].
\end{aligned}$$

Let us first apply Lemma E.13

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var,in}}^{t+1} &\leq \frac{p_{\text{out}}}{B_1} 4C_g^2 L_f^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_y^{t+1} + \frac{(1-p_{\text{out}})\mathcal{E}_{\text{var,in}}^1}{T} + \frac{2(1-p_{\text{out}})}{B_2} \frac{C_f^2 L_g^2 \gamma^2}{T} \sum_{t=0}^{T-1} \mathbb{E} [\|G^{t+1}\|_2^2] \\
&\quad + \frac{2(1-p_{\text{out}})C_g^2 L_f^2}{B_2} \left(\frac{4C_g^2 \gamma^2}{T} \sum_{t=0}^{T-1} \mathbb{E} [\|G^{t+1}\|_2^2] + \frac{4(1-p_{\text{in}})C_g^2}{S_2} \frac{1}{T} \sum_{t=0}^{T-1} \Xi^t + \frac{6(1-p_{\text{in}})}{T} \sum_{t=0}^{T-1} \mathcal{E}_y^{t+1} \right) \\
&\leq \left(\frac{p_{\text{out}}}{B_1} + \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} \right) \frac{12C_g^2 L_f^2}{T} \sum_{t=0}^{T-1} \mathcal{E}_y^{t+1} + \frac{8(1-p_{\text{out}})L_F^2 \gamma^2}{B_2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\|G^{t+1}\|_2^2] \\
&\quad + \frac{8(1-p_{\text{out}})(1-p_{\text{in}})C_g^4 L_f^2}{B_2 S_2} \frac{1}{T} \sum_{t=0}^{T-1} \Xi^t + \frac{(1-p_{\text{out}})\mathcal{E}_{\text{var,in}}^1}{T}
\end{aligned}$$

Then we apply Lemma E.12 on the bound of $\frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_y^{t+1}$

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var},\text{in}}^{t+1} &\leq 24 \left(\frac{p_{\text{out}}}{B_1} + \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} \right) \left(\frac{(1-p_{\text{in}})L_F^2}{p_{\text{in}}S_2} \frac{1}{T} \sum_{t=0}^{T-1} \Xi^t + \frac{\tilde{L}_F^2}{S_1} \right) \\
&\quad + \frac{8(1-p_{\text{out}})L_F^2\gamma^2}{B_2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] \\
&\quad + \frac{8(1-p_{\text{out}})(1-p_{\text{in}})C_g^4L_f^2}{B_2S_2} \frac{1}{T} \sum_{t=0}^{T-1} \Xi^t + \frac{(1-p_{\text{out}})\mathcal{E}_{\text{var},\text{in}}^1}{T} \\
&\leq 24 \left(\frac{p_{\text{out}}}{B_1} + \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} + \frac{p_{\text{in}}(1-p_{\text{out}})}{B_2} \right) \frac{(1-p_{\text{in}})L_F^2}{p_{\text{in}}S_2} \frac{1}{T} \sum_{t=0}^{T-1} \Xi^t \\
&\quad + \frac{8(1-p_{\text{out}})L_F^2\gamma^2}{B_2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] \\
&\quad + 24 \left(\frac{p_{\text{out}}}{B_1} + \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} \right) \frac{\tilde{L}_F^2}{S_1} + \frac{(1-p_{\text{out}})\mathcal{E}_{\text{var},\text{in}}^1}{T} \\
&\leq 24 \left(\frac{p_{\text{out}}}{B_1} + \frac{1-p_{\text{out}}}{B_2} \right) \frac{(1-p_{\text{in}})L_F^2}{p_{\text{in}}S_2} \frac{1}{T} \sum_{t=0}^{T-1} \Xi^t \\
&\quad + \frac{8(1-p_{\text{out}})L_F^2\gamma^2}{B_2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] \\
&\quad + 24 \left(\frac{p_{\text{out}}}{B_1} + \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} \right) \frac{\tilde{L}_F^2}{S_1} + \frac{(1-p_{\text{out}})}{T} \mathcal{E}_{\text{var},\text{in}}^1.
\end{aligned}$$

From Lemma E.11, we plug in the upper bound of $\frac{1}{T} \sum_{t=0}^{T-1} \Xi^t$

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var},\text{in}}^{t+1} &\leq 24 \left(\frac{p_{\text{out}}}{B_1} + \frac{1-p_{\text{out}}}{B_2} \right) \frac{(1-p_{\text{in}})L_F^2}{p_{\text{in}}S_2} \left(\frac{6n^2}{B_2^2} \gamma^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] \right) \\
&\quad + \frac{8(1-p_{\text{out}})L_F^2\gamma^2}{B_2} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] \\
&\quad + 24 \left(\frac{p_{\text{out}}}{B_1} + \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} \right) \frac{\tilde{L}_F^2}{S_1} + \frac{(1-p_{\text{out}})}{T} \mathcal{E}_{\text{var},\text{in}}^1 \\
&\leq 8 \left(\left(\frac{p_{\text{out}}}{B_1} + \frac{1-p_{\text{out}}}{B_2} \right) \frac{(1-p_{\text{in}})}{p_{\text{in}}S_2} \frac{18n^2}{B_2^2} + \frac{(1-p_{\text{out}})}{B_2} \right) \frac{\gamma^2 L_F^2}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] \\
&\quad + 24 \left(\frac{p_{\text{out}}}{B_1} + \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} \right) \frac{\tilde{L}_F^2}{S_1} + \frac{(1-p_{\text{out}})}{T} \mathcal{E}_{\text{var},\text{in}}^1.
\end{aligned}$$

Finally, we add the upper bound on with $\mathcal{E}_{\text{var},\text{in}}^1$ with (E.18)

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var},\text{in}}^{t+1} &\leq 8 \left(\left(\frac{p_{\text{out}}}{B_1} + \frac{1-p_{\text{out}}}{B_2} \right) \frac{(1-p_{\text{in}})}{p_{\text{in}}S_2} \frac{18n^2}{B_2^2} + \frac{(1-p_{\text{out}})}{B_2} \right) \frac{\gamma^2 L_F^2}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] \\
&\quad + 24 \left(\frac{p_{\text{out}}}{B_1} + \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} \right) \frac{\tilde{L}_F^2}{S_1} + \frac{(1-p_{\text{out}})}{T} \frac{4\tilde{L}_F^2}{B_1S_1}. \tag{E.19}
\end{aligned}$$

Outer Variance. Now we consider the outer variance for $t \geq 1$

$$\begin{aligned}
\mathcal{E}_{\text{var},\text{out}}^{t+1} &\leq \frac{(1-p_{\text{out}})^2}{B_2} \mathbb{E}_i \left[\|\mathbb{E}_{\eta|i}[G_i^{t+1}] - \mathbb{E}_{\eta|i}[\tilde{G}_i^t]\|_2^2 \right] \\
&\leq \frac{(1-p_{\text{out}})^2}{B_2} \mathbb{E}_i \left[\mathbb{E}_{\eta|i} \left[\|G_i^{t+1} - \tilde{G}_i^t\|_2^2 \right] \right].
\end{aligned}$$

Compared to (E.16) we know that the upper bound of is smaller than that of $\mathcal{E}_{\text{var},\text{in}}^{t+1}$. Besides, whereas $\mathcal{E}_{\text{var},\text{out}}^1 = 0$ as we use large batch at $t = 0$. Therefore, the upper bound of $\mathcal{E}_{\text{var}}^{t+1}$ is upper bounded by $2^*(\text{E.19})$.

Variance of $\nabla g_{\tilde{\eta}}$. From Lemma E.10, we know that

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{E}_{\nabla g}^{t+1}] &\leq \frac{1}{B_1} \frac{C_f^2 \sigma_g^2}{m} + \frac{4(1-p_{\text{out}})}{B_2 m p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \left(\mathbb{E} \left[\|G_i^{t+1} - \tilde{G}_i^t\|_2^2 \right] \right) \\ &\leq \frac{1}{B_1} \frac{C_f^2 \sigma_g^2}{m} + \frac{1}{m} \mathcal{E}_{\text{var}}^{t+1} \end{aligned}$$

Finally, we use $\mathbb{E}[\|G^{t+1}\|_2^2] = \mathbb{E}[G^{t+1}]\|_2^2 + \mathcal{E}_{\text{var}}^{t+1}$.

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1} &\leq 16 \left(\left(\frac{p_{\text{out}}}{B_1} + \frac{1-p_{\text{out}}}{B_2} \right) \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18n^2}{B_2^2} + \frac{(1-p_{\text{out}})}{B_2} \right) \frac{\gamma^2 L_F^2}{T} \sum_{t=0}^{T-1} \|\mathbb{E}[G^{t+1}]\|_2^2 \\ &\quad + 16 \left(\left(\frac{p_{\text{out}}}{B_1} + \frac{1-p_{\text{out}}}{B_2} \right) \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18n^2}{B_2^2} + \frac{(1-p_{\text{out}})}{B_2} \right) \frac{\gamma^2 L_F^2}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1} \\ &\quad + 48 \left(\frac{p_{\text{out}}}{B_1} + \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} \right) \frac{\tilde{L}_F^2}{S_1} + \frac{(1-p_{\text{out}})}{T} \frac{8\tilde{L}_F^2}{B_1 S_1}. \end{aligned}$$

By taking step size γ to satisfy

$$\gamma^2 L_F^2 \max \left\{ \frac{p_{\text{out}}}{B_1} \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18n^2}{B_2^2}, \frac{1-p_{\text{out}}}{B_2} \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18n^2}{B_2^2}, \frac{(1-p_{\text{out}})}{B_2} \right\} \leq \frac{1}{16} \cdot \frac{1}{6}$$

which can be simplified to

$$\gamma^2 L_F^2 \max \left\{ \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18}{B_2}, \frac{1-p_{\text{out}}}{B_2} \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18n^2}{B_2^2}, \frac{(1-p_{\text{out}})}{B_2} \right\} \leq \frac{1}{16} \cdot \frac{1}{6}.$$

Then the coefficient of $\frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}$ is bounded by $\frac{1}{2}$

$$16 \left(\left(\frac{p_{\text{out}}}{B_1} + \frac{1-p_{\text{out}}}{B_2} \right) \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18n^2}{B_2^2} + \frac{(1-p_{\text{out}})}{B_2} \right) \gamma^2 L_F^2 \leq \frac{1}{2}.$$

The the variance has the following bound

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1} &\leq 32 \left(\left(\frac{p_{\text{out}}}{B_1} + \frac{1-p_{\text{out}}}{B_2} \right) \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18n^2}{B_2^2} + \frac{(1-p_{\text{out}})}{B_2} \right) \frac{\gamma^2 L_F^2}{T} \sum_{t=0}^{T-1} \|\mathbb{E}[G^{t+1}]\|_2^2 \\ &\quad + 96 \left(\frac{p_{\text{out}}}{B_1} + \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} \right) \frac{\tilde{L}_F^2}{S_1} + \frac{(1-p_{\text{out}})}{T} \frac{8\tilde{L}_F^2}{B_1 S_1}. \end{aligned}$$

□

Theorem E.6. Consider the (FCCO) problem. Suppose Assumptions G, H, I holds true. Let step size $\gamma = \mathcal{O}(\frac{1}{\sqrt{n}L_F})$. Then for NestedVR, \mathbf{x}^s picked uniformly at random among $\{\mathbf{x}^t\}_{t=0}^{T-1}$ satisfies: $\mathbb{E}[\|\nabla F(\mathbf{x}^s)\|_2^2] \leq \epsilon^2$, for nonconvex F , if the hyperparameters of the inner loop $S_1 = \mathcal{O}(\tilde{L}_F^2 \epsilon^{-2})$, $S_2 = \mathcal{O}(\tilde{L}_F \epsilon^{-1})$, $p_{\text{in}} = \mathcal{O}(1/S_2)$, the hyperparameters of the outer loop $B_1 = n$, $B_2 = \sqrt{n}$, $p_{\text{out}} = 1/B_2$, and the number of iterations

$$T = \Omega \left(\frac{\sqrt{n}L_F(F(\mathbf{x}^0) - F^*)}{\epsilon^2} \right).$$

The resulting sample complexity is

$$\mathcal{O}\left(\frac{nL_F\tilde{L}_F(F(\mathbf{x}^0)-F^*)}{\epsilon^3}\right).$$

In fact, it reaches this sample complexity for all $\frac{p_{in}p_{out}}{\sqrt{1-p_{in}}} \lesssim \epsilon$.

Proof. Using descent lemma (Lemma E.4) and bias-variance bounds of NestedVR (Lemma E.14)

$$\begin{aligned} & \frac{1}{T} \sum_{t=0}^{T-1} \|\nabla F(\mathbf{x}^t)\|_2^2 + \frac{1}{2T} \sum_{t=0}^{T-1} \|\mathbb{E}[G^{t+1}]\|_2^2 \\ & \leq \frac{2(F(\mathbf{x}^0)-F^*)}{\gamma T} + \frac{L_F\gamma}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1} + \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{bias}}^{t+1} \\ & \leq \underbrace{\frac{2(F(\mathbf{x}^0)-F^*)}{\gamma T}}_{\mathcal{T}_0} + \underbrace{\frac{4\tilde{L}_F^2}{S_1}}_{\mathcal{T}_1} + \underbrace{\frac{12(1-p_{in})}{p_{in}S_2} \frac{n^2}{B_2^2} \frac{L_F^2\gamma^2}{T} \sum_{t=0}^{T-1} \|\mathbb{E}[G^{t+1}]\|_2^2}_{\mathcal{T}_2} \\ & \quad + \underbrace{\left(\frac{12(1-p_{in})}{p_{in}S_2} \frac{n^2}{B_2^2} L_F^2\gamma^2 + \frac{2(1-p_{out})^2}{p_{out}} + \gamma L_F \right) \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}}_{\mathcal{T}_3}. \end{aligned}$$

Compute \mathcal{T}_0 . In order to let $\mathcal{T}_0 \leq \epsilon^2$, we require that

$$\gamma T \geq \epsilon^{-2}. \quad (\text{E.20})$$

Compute \mathcal{T}_1 . In order to let \mathcal{T}_1 to be smaller than ϵ^2 , we need

$$S_1 = \frac{4\tilde{L}_F^2}{\epsilon^2}.$$

Compute \mathcal{T}_2 . In order to let the coefficient of $\frac{1}{T} \sum_{t=0}^{T-1} \|\mathbb{E}[G^{t+1}]\|_2^2$ in \mathcal{T}_2 to be less than $\frac{1}{4}$, i.e.

$$\frac{12(1-p_{in})}{p_{in}S_2} \frac{n^2}{B_2^2} L_F^2\gamma^2 \leq \frac{1}{4}, \quad (\text{E.21})$$

which requires γ

$$\gamma \leq \frac{B_2\sqrt{p_{in}S_2}}{7L_F n\sqrt{1-p_{in}}} = \frac{p_{out}p_{in}\tilde{L}_F}{7\epsilon L_F\sqrt{1-p_{in}}}. \quad (\text{E.22})$$

Compute \mathcal{T}_3 . Let us now focus on \mathcal{T}_3 and notice that the middle term $\frac{2(1-p_{out})^2}{p_{out}}$

$$\frac{2(1-p_{out})^2}{p_{out}} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}.$$

Using Lemma E.14 we have that

$$\begin{aligned}
& \frac{2(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1} \\
& \leq \underbrace{32 \frac{2(1-p_{\text{out}})^2}{p_{\text{out}}} \left(\left(\frac{p_{\text{out}}}{B_1} + \frac{1-p_{\text{out}}}{B_2} \right) \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18n^2}{B_2^2} + \frac{(1-p_{\text{out}})}{B_2} \right)}_{\mathcal{T}_{3,1}} \gamma^2 L_F^2 \frac{1}{T} \sum_{t=0}^{T-1} \|\mathbb{E}[G^{t+1}]\|_2^2 \\
& \quad + \underbrace{96 \frac{2(1-p_{\text{out}})^2}{p_{\text{out}}} \left(\frac{p_{\text{out}}}{B_1} + \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} \right)}_{\mathcal{T}_{3,2}} \frac{\tilde{L}_F^2}{S_1} + \underbrace{\frac{2(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{(1-p_{\text{out}})}{T} \frac{8\tilde{L}_F^2}{B_1 S_1}}_{\mathcal{T}_{3,3}}.
\end{aligned}$$

- Compute $\mathcal{T}_{3,3}$: As we already know that $S_1 = \mathcal{O}(\epsilon^{-2})$ and $T \geq 1$ and $B_1 p_{\text{out}} \geq 1$. This imposes no more constraints, i.e.

$$S_1 = \mathcal{O}\left(\frac{\tilde{L}_F^2}{\epsilon^2}\right).$$

- Compute $\mathcal{T}_{3,2}$: As $S_1 = \mathcal{O}(\epsilon^{-2})$ and $B_1 = n$ and $B_2 = B_1 p_{\text{out}}$, then it requires

$$\frac{(1-p_{\text{in}})(1-p_{\text{out}})^3}{p_{\text{out}}^2} \leq n.$$

- Compute $\mathcal{T}_{3,1}$: In order to satisfy the following

$$32 \frac{2(1-p_{\text{out}})^2}{p_{\text{out}}} \left(\left(\frac{p_{\text{out}}}{B_1} + \frac{1-p_{\text{out}}}{B_2} \right) \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18n^2}{B_2^2} + \frac{(1-p_{\text{out}})}{B_2} \right) \gamma^2 L_F^2 \leq \frac{1}{12}$$

we need to enforce

$$\gamma \leq \frac{p_{\text{in}} p_{\text{out}} \tilde{L}_F}{\epsilon L_F (1-p_{\text{in}})^{1/2} (1-p_{\text{out}})^{3/2}}. \quad (\text{E.23})$$

Now we go back to \mathcal{T}_3 and compare the other two coefficients

$$\frac{12(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{n^2}{B_2^2} L_F^2 \gamma^2 + \frac{2(1-p_{\text{out}})^2}{p_{\text{out}}} + \gamma L_F.$$

As $\gamma L_F \leq \frac{1}{2} \lesssim \frac{2(1-p_{\text{out}})^2}{p_{\text{out}}}$ we can safely ignore γL_F . On the other hand, from (E.21) we know that the first term is also have

$$\frac{12(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{n^2}{B_2^2} L_F^2 \gamma^2 \leq \frac{1}{4} \lesssim \frac{2(1-p_{\text{out}})^2}{p_{\text{out}}}.$$

Constraints from the Bias-Variance Lemma (Lemma E.14). By setting $B_1 = n$ and $S_1 = \mathcal{O}(\frac{\tilde{L}_F^2}{\epsilon^2})$, this constraint translates to

$$\gamma^2 L_F^2 \max \left\{ \frac{(1-p_{\text{in}})}{p_{\text{in}}^2} \frac{\epsilon^2}{B_2}, \frac{1-p_{\text{out}}}{B_2} \frac{(1-p_{\text{in}})\epsilon^2}{p_{\text{in}}^2} \frac{1}{p_{\text{out}}^2}, \frac{(1-p_{\text{out}})}{B_2} \right\} \lesssim 1$$

which is weaker than (E.22).

Summary on the Limit on γ . Combine (E.22) and (E.23) and $\gamma \leq \frac{1}{2L_F}$, we have a final limit on step size γ

$$\gamma \lesssim \min \left\{ \frac{p_{\text{out}} p_{\text{in}} \tilde{L}_F}{\epsilon L_F \sqrt{1-p_{\text{in}}}}, \frac{1}{L_F} \right\} \quad (\text{E.24})$$

Then the total sample complexity of NestedVR can be computed as

$$(\# \text{ of iters } T) \times (\text{Avg. outer batch size } B_2 = B_1 p_{\text{out}}) \times (\text{Avg. inner batch size } S_2 = S_1 p_{\text{in}}).$$

This sample complexity has the following requirement

$$B_2 S_2 T = \frac{B_2 S_2 (T\gamma)}{\gamma} \stackrel{(\text{E.20})}{\geq} \frac{B_2 S_2}{\epsilon^2 \gamma} = \frac{n \epsilon^{-2}}{\epsilon^2} \frac{p_{\text{in}} p_{\text{out}}}{\gamma} \stackrel{(\text{E.24})}{\gtrsim} n \epsilon^{-3}.$$

The lower bound $n \epsilon^{-3}$ is reached when in (E.24) we have

$$\frac{p_{\text{out}} p_{\text{in}} \tilde{L}_F}{\epsilon L_F \sqrt{1-p_{\text{in}}}} \lesssim \frac{1}{L_F}.$$

That is, $\frac{p_{\text{out}} p_{\text{in}}}{\sqrt{1-p_{\text{in}}}} \lesssim \epsilon$.

In particular, we can choose the following hyperparameters to reach $\mathcal{O}(n \epsilon^{-3})$ sample complexity

$$B_1 = n, \quad B_2 = \sqrt{n}, \quad p_{\text{out}} = \frac{1}{\sqrt{n}}, \quad S_1 = \mathcal{O}(\tilde{L}_F^2 \epsilon^{-2}), \quad S_2 = \mathcal{O}(\tilde{L}_F \epsilon^{-1}), \quad p_{\text{in}} = \mathcal{O}(\tilde{L}_F^{-1} \epsilon)$$

The step size γ can be chosen as

$$\gamma \lesssim \frac{1}{\sqrt{n} L_F}.$$

and the iteration complexity

$$T = \Omega \left(\frac{\sqrt{n} L_F (F(\mathbf{x}^0) - F^*)}{\epsilon^2} \right).$$

Putting these together gives the claimed sample complexity bound. By picking \mathbf{x}^s uniformly at random among $\{\mathbf{x}^t\}_{t=0}^{T-1}$, we get the desired guarantee. \square

E.5.3 Convergence of E-NestedVR

In this section, we analyze the sample complexity of Algorithm 7 (E-NestedVR) for the FCCO problem with

$$G_{\text{E-NVR}}^{t+1} = \begin{cases} \frac{1}{B_1} \sum_i (z_i^{t+1})^\top \mathcal{L}_{\mathcal{D}_{\mathbf{y},i}^{t+1}}^{(2)} \nabla f_i(0) & \text{with prob. } p_{\text{out}} \\ G_{\text{E-NVR}}^t + \frac{1}{B_2} \sum_i \left((z_i^{t+1})^\top \mathcal{L}_{\mathcal{D}_{\mathbf{y},i}^{t+1}}^{(2)} \nabla f_i(0) - (z_i^t)^\top \mathcal{L}_{\mathcal{D}_{\mathbf{y},i}^t}^{(2)} \nabla f_i(0) \right) & \text{with prob. } 1 - p_{\text{out}}. \end{cases} \quad (\text{E.25})$$

Lemma E.15 (Bias and Variance of E-NestedVR). *If the step size γ satisfies*

$$\gamma^2 L_F^2 \max \left\{ \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18}{B_2}, \frac{1-p_{\text{out}}}{B_2} \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18n^2}{B_2^2}, \frac{(1-p_{\text{out}})}{B_2} \right\} \leq \frac{1}{16} \cdot \frac{1}{6}$$

then the variance and bias of E-NestedVR are

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1} &\leq 14 \cdot 32 \left(\left(\frac{p_{\text{out}}}{B_1} + \frac{1-p_{\text{out}}}{B_2} \right) \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18n^2}{B_2^2} + \frac{(1-p_{\text{out}})}{B_2} \right) \frac{\gamma^2 L_F^2}{T} \sum_{t=0}^{T-1} \|\mathbb{E}[G^{t+1}]\|_2^2 \\ &\quad + 14 \cdot 96 \left(\frac{p_{\text{out}}}{B_1} + \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} \right) \frac{\tilde{L}_F^2}{S_1} + \frac{(1-p_{\text{out}})}{T} \frac{8\tilde{L}_F^2}{B_1 S_1}. \\ \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{bias}}^{t+1} &\leq \frac{(1-p_{\text{in}})^3 \tilde{L}_F^2}{p_{\text{in}} S_2} \frac{6n^2}{B_2^2} \gamma^2 \frac{1}{T} \sum_{t=0}^{T-1} \|\mathbb{E}[G^{t+1}]\|_2^2 + \frac{2(1-p_{\text{in}})^2 \tilde{L}_F^2}{S_1} + \frac{C_\epsilon^2}{S_2^4} \\ &\quad + \left(\frac{(1-p_{\text{in}})^3 \tilde{L}_F^2}{p_{\text{in}} S_2} \frac{6n^2}{B_2^2} \gamma^2 + \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \right) \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}. \end{aligned}$$

Proof. Note that this proof is very similar to NestedVR so we highlight the differences. Let $G^{t+1} = G_{\text{E-NVR}}^{t+1}$ (E.25) be the E-NestedVR update and define

$$G_i^{t+1} := (z_i^{t+1})^\top \mathcal{L}_{\mathcal{D}_{\mathbf{y},i}^{t+1}}^{(2)} \nabla f_i(0)$$

We expand the bias by inserting $\mathbb{E}_{i,p_{\text{in}},\eta|i}[G_i^{t+1}]$

$$\begin{aligned} \mathcal{E}_{\text{bias}}^{t+1} &= \|\nabla F(\mathbf{x}^{t+1}) - \mathbb{E}[G^{t+1}]\|_2^2 \\ &\leq 2 \underbrace{\|\nabla F(\mathbf{x}^{t+1}) - \mathbb{E}_{i,p_{\text{in}},\eta,\tilde{\eta}|i}[G_i^{t+1}]\|_2^2}_{A_1^{t+1}} + 2 \underbrace{\|\mathbb{E}_{i,p_{\text{in}},\eta,\tilde{\eta}|i}[G_i^{t+1}] - \mathbb{E}[G^{t+1}]\|_2^2}_{A_2^{t+1}}. \end{aligned}$$

Consider A_1^{t+1} . The term A_1^{t+1} captures the difference between full gradient and extrapolated gradient

$$\begin{aligned}
A_1^{t+1} &= \|\mathbb{E}_i \left[(\mathbb{E}_{\tilde{\eta}|i}[\nabla g_{\tilde{\eta}}(\mathbf{x}^t)])^\top \nabla f_i(\mathbb{E}[g_{\eta}(\mathbf{x}^t)]) - \mathbb{E}_{p_{\text{in}}, \eta|i} \left[(\mathbb{E}_{\tilde{\eta}|i}[\nabla g_{\tilde{\eta}}(\mathbf{x}^t)])^\top \mathcal{L}_{\mathcal{D}_{\mathbf{y},i}^{t+1}}^{(2)} \nabla f_i(0) \right] \right]\|_2^2 \\
&\leq C_g^2 \mathbb{E}_i \left[\|\nabla f_i(\mathbb{E}_{\eta|i}[g_{\eta}(\mathbf{x}^t)]) - \mathbb{E}_{p_{\text{in}}, \eta|i} \left[\mathcal{L}_{\mathcal{D}_{\mathbf{y},i}^{t+1}}^{(2)} \nabla f_i(0) \right]\|_2^2 \right] \\
&\leq 2C_g^2 \underbrace{\mathbb{E}_i [\|\nabla f_i(\mathbb{E}_{\eta|i}[g_{\eta}(\mathbf{x}^t)]) - \mathbb{E}_{p_{\text{in}}}[\nabla f_i(\mathbb{E}_{\eta|i}[\mathbf{y}_i^{t+1}])]\|_2^2]}_{=: A_{1,1}^{t+1}} \\
&\quad + 2C_g^2 \underbrace{\mathbb{E}_i \left[\|\mathbb{E}_{p_{\text{in}}}[\nabla f_i(\mathbb{E}_{\eta|i}[\mathbf{y}_i^{t+1}])]\|_2 - \mathbb{E}_{p_{\text{in}}, \eta|i} \left[\mathcal{L}_{\mathcal{D}_{\mathbf{y},i}^{t+1}}^{(2)} \nabla f_i(0) \right]\|_2 \right]^2}_{=: A_{1,2}^{t+1}}.
\end{aligned}$$

The first term $A_{1,1}^{t+1}$ can be upper bounded through smoothness of f_{ξ} , for $t \geq 1$

$$\begin{aligned}
A_{1,1}^{t+1} &= \mathbb{E}_i [\|\nabla f_i(\mathbb{E}_{\eta|i}[g_{\eta}(\mathbf{x}^t)]) - p_{\text{in}} \nabla f_i(\mathbb{E}_{\eta|i}[g_{\eta}(\mathbf{x}^t)]) - (1 - p_{\text{in}}) \nabla f_i(\mathbf{y}_i^t + \mathbb{E}_{\eta|i}[g_{\eta}(\mathbf{x}^t) - g_{\eta}(\phi_i^t)])\|_2^2] \\
&= (1 - p_{\text{in}})^2 \mathbb{E}_i [\|\nabla f_i(\mathbb{E}[g_{\eta}(\mathbf{x}^t)]) - \nabla f_i(\mathbf{y}_i^t + \mathbb{E}_{\eta|i}[g_{\eta}(\mathbf{x}^t) - g_{\eta}(\phi_i^t)])\|_2^2] \\
&\leq (1 - p_{\text{in}})^2 L_f^2 \mathbb{E}_i [\|\mathbb{E}_{\eta|i}[g_{\eta}(\mathbf{x}^t)] - (\mathbf{y}_i^t + \mathbb{E}_{\eta|i}[g_{\eta}(\mathbf{x}^t) - g_{\eta}(\phi_i^t)])\|_2^2] \\
&= (1 - p_{\text{in}})^2 L_f^2 \mathbb{E}_i [\|\mathbf{y}_i^t - \mathbb{E}_{\eta|i}[g_{\eta}(\phi_i^t)]\|_2^2] \\
&= (1 - p_{\text{in}})^2 L_f^2 \mathcal{E}_y^t.
\end{aligned}$$

For $t = 0$, $A_{1,1}^1 = 0$, then

$$\frac{1}{T} \sum_{t=0}^{T-1} A_{1,1}^{t+1} \leq (1 - p_{\text{in}})^2 L_f^2 C_g^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_y^{t+1}. \quad (\text{E.26})$$

On the other hand, with Lemma E.6

$$\begin{aligned}
A_{1,2}^{t+1} &\leq p_{\text{in}} \mathbb{E}_i \left[\|\nabla f_i(\mathbb{E}_{\eta|i}[g_{\eta}(\mathbf{x}^t)]) - \mathbb{E}_{\eta|i} \left[\mathcal{L}_{\mathcal{D}_{\mathbf{y},S_1,i}^{t+1}}^{(2)} \nabla f_i(0) \right]\|_2^2 \right] \\
&\quad + (1 - p_{\text{in}}) \mathbb{E}_i \left[\|\nabla f_i(\mathbf{y}_i^t + \mathbb{E}_{\eta|i}[g_{\eta}(\mathbf{x}^t) - g_{\eta}(\phi_i^t)]) - \mathbb{E}_{\eta|i} \left[\mathcal{L}_{\mathcal{D}_{\mathbf{y},S_2,i}^{(2)}}^{(2)} \nabla f_i(0) \right]\|_2^2 \right] \\
&\leq \frac{p_{\text{in}} C_e^2}{S_1^4} + \frac{(1 - p_{\text{in}}) C_e^2}{S_2^4} \\
&\leq \frac{C_e^2}{S_2^4}
\end{aligned}$$

where $\mathcal{D}_{\mathbf{y},S_1,i}^{t+1}$ is the distribution of $\frac{1}{S_1} \sum_{\eta \in S_1} g_{\eta}(\mathbf{x}^t)$ and $\mathcal{D}_{\mathbf{y},S_2,i}^{t+1}$ is the distribution of

$$\mathbf{y}_i^t + \frac{1}{S_2} \sum_{\eta \in S_2} (g_{\eta}(\mathbf{x}^t) - \mathbb{E}[g_{\eta}(\phi_i^t)]).$$

Thus the A_1^{t+1} has the following upper bound

$$\frac{1}{T} \sum_{t=0}^{T-1} A_1^{t+1} \leq (1 - p_{\text{in}})^2 L_f^2 C_g^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_y^{t+1} + \frac{C_e^2}{S_2^4}. \quad (\text{E.27})$$

Consider A_2^{t+1} . Let us expand A_2^{t+1} through recursion

$$\begin{aligned} A_2^{t+1} &= \|\mathbb{E}_{i,p_{\text{in}},\eta,\tilde{\eta}|i}[G_i^{t+1}] - \mathbb{E}[G^{t+1}]\|_2^2 \\ &= (1 - p_{\text{out}})^2 \|G^t - \mathbb{E}_i[\mathbb{E}_{\eta,\tilde{\eta}|i}[\tilde{G}_i^t]]\|_2^2 \\ &= (1 - p_{\text{out}})^2 \left(\|\mathbb{E}[G^t] - \mathbb{E}_i[\mathbb{E}_{\eta,\tilde{\eta}|i}[\tilde{G}_i^t]]\|_2^2 + \mathcal{E}_{\text{var}}^t \right) \\ &= (1 - p_{\text{out}})^2 (A_2^t + \mathcal{E}_{\text{var}}^t). \end{aligned}$$

For $t = 0$, we have that $A_2^1 = 0$, then average over time gives

$$\frac{1}{T} \sum_{t=0}^{T-1} A_2^{t+1} \leq \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}.$$

Therefore, the bias has the following bound

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{bias}}^{t+1} \leq (1 - p_{\text{in}})^2 L_f^2 C_g^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_y^{t+1} + \frac{C_e^2}{S_2^4} + \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}.$$

Using Lemma E.12

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{bias}}^{t+1} &\leq (1 - p_{\text{in}})^2 L_f^2 C_g^2 \left(\frac{(1-p_{\text{in}})C_g^2}{p_{\text{in}}S_2} \frac{1}{T} \sum_{t=0}^{T-1} \Xi^t + \frac{2\sigma_g^2}{S_1} \right) \\ &\quad + \frac{C_e^2}{S_2^4} + \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1} \\ &\leq \frac{(1-p_{\text{in}})^3 \tilde{L}_F^2}{p_{\text{in}}S_2} \frac{1}{T} \sum_{t=0}^{T-1} \Xi^t + \frac{2(1-p_{\text{in}})^2 \tilde{L}_F^2}{S_1} + \frac{C_e^2}{S_2^4} + \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}. \end{aligned}$$

Using Lemma E.11 we have that

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{bias}}^{t+1} &\leq \frac{(1-p_{\text{in}})^3 \tilde{L}_F^2}{p_{\text{in}}S_2} \left(\frac{6n^2}{B_2^2} \gamma^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] \right) + \frac{2(1-p_{\text{in}})^2 \tilde{L}_F^2}{S_1} + \frac{C_e^2}{S_2^4} + \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1} \\ &\leq \frac{(1-p_{\text{in}})^3 \tilde{L}_F^2}{p_{\text{in}}S_2} \frac{6n^2}{B_2^2} \gamma^2 \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] + \frac{2(1-p_{\text{in}})^2 \tilde{L}_F^2}{S_1} + \frac{C_e^2}{S_2^4} \\ &\quad + \left(\frac{(1-p_{\text{in}})^3 \tilde{L}_F^2}{p_{\text{in}}S_2} \frac{6n^2}{B_2^2} \gamma^2 + \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \right) \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}. \end{aligned}$$

Variance. Combine the variance of NestedVR in Lemma E.14 and Lemma E.2 gives

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1} &\leq 14 \cdot 32 \left(\left(\frac{p_{\text{out}}}{B_1} + \frac{1-p_{\text{out}}}{B_2} \right) \frac{(1-p_{\text{in}})}{p_{\text{in}}S_2} \frac{18n^2}{B_2^2} + \frac{(1-p_{\text{out}})}{B_2} \right) \frac{\gamma^2 L_F^2}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|G^{t+1}\|_2^2] \\ &\quad + 14 \cdot 96 \left(\frac{p_{\text{out}}}{B_1} + \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} \right) \frac{\tilde{L}_F^2}{S_1} + \frac{(1-p_{\text{out}})}{T} \frac{8\tilde{L}_F^2}{B_1 S_1}. \end{aligned}$$

□

Theorem 6.4. [*E-NestedVR Convergence*] Consider the (FCCO) problem. Under the same assumptions as Theorem 6.2.

- If $n = \mathcal{O}(\epsilon^{-2/3})$, then we choose the hyperparameters of E-NestedVR (Algorithm 7) as $B_1 = B_2 = n, p_{out} = 1, S_1 = \tilde{L}_F^2 \epsilon^{-2}, S_2 = \tilde{L}_F \epsilon^{-1}, p_{in} = \tilde{L}_F^{-1} \epsilon, \gamma = \mathcal{O}(\frac{1}{L_F})$.
- If $n = \Omega(\epsilon^{-2/3})$, then we choose the hyperparameters of E-NestedVR as $B_1 = n, B_2 = \sqrt{n}, p_{out} = 1/\sqrt{n}, S_1 = S_2 = \max \left\{ C_e C_g \epsilon^{-1/2}, \tilde{L}_F^2 / (n \epsilon^2) \right\}, p_{in} = 1, \gamma = \mathcal{O}(\frac{1}{L_F})$.

Then the output \mathbf{x}^s of E-NestedVR satisfies: $\mathbb{E}[\|\nabla F(\mathbf{x}^s)\|_2^2] \leq \epsilon^2$, for nonconvex F with iterations

$$T = \Omega \left(L_F (F(\mathbf{x}^0) - F^*) \epsilon^{-2} \right).$$

Proof. Denote. $G^{t+1} = G_{\text{E-NVR}}^{t+1}$ (E.25). Using descent lemma (Lemma E.3) and bias-variance of E-NestedVR (Lemma E.15)

$$\begin{aligned} & \frac{1}{T} \sum_{t=0}^{T-1} \|\nabla F(\mathbf{x}^t)\|_2^2 + \frac{1}{2T} \sum_{t=0}^{T-1} \|\mathbb{E}[G^{t+1}]\|_2^2 \\ & \leq \frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T} + \frac{L_F \gamma}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1} + \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{bias}}^{t+1} \\ & \leq \frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T} + \frac{(1-p_{in})^3 \tilde{L}_F^2}{p_{in} S_2} \frac{6n^2}{B_2^2} \gamma^2 \frac{1}{T} \sum_{t=0}^{T-1} \|\mathbb{E}[G^{t+1}]\|_2^2 + \frac{2(1-p_{in})^2 \tilde{L}_F^2}{S_1} + \frac{C_e^2}{S_2^4} \\ & \quad + \left(\frac{(1-p_{in})^3 \tilde{L}_F^2}{p_{in} S_2} \frac{6n^2}{B_2^2} \gamma^2 + \frac{(1-p_{out})^2}{p_{out}} + L_F \gamma \right) \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1}. \end{aligned}$$

As we would like the right-hand side to be bounded by either $\frac{1}{T} \sum_{t=0}^{T-1} \|\mathbb{E}[G^{t+1}]\|_2^2$ or ϵ^2 .

- **Bound on $\frac{2(F(\mathbf{x}^0) - F^*)}{\gamma T}$ with ϵ^2** , i.e.

$$\gamma T \gtrsim (F(\mathbf{x}^0) - F^*) \epsilon^{-2} \quad (\text{E.28})$$

- **Coefficient of $\frac{1}{T} \sum_{t=0}^{T-1} \|\mathbb{E}[G^{t+1}]\|_2^2$ is bounded by $\frac{1}{4}$** , i.e.

$$\frac{(1-p_{in})^3 \tilde{L}_F^2}{p_{in} S_2} \frac{6n^2}{B_2^2} \gamma^2 \leq \frac{1}{4}$$

which can be achieved by choosing the following step size

$$\gamma \leq \frac{p_{out} p_{in} \sqrt{S_1}}{5 \tilde{L}_F (1-p_{in})^{3/2}}. \quad (\text{E.29})$$

- **Bound on $\frac{2(1-p_{in})^2 \tilde{L}_F^2}{S_1}$ with ϵ^2**

$$\frac{2(1-p_{in})^2 \tilde{L}_F^2}{S_1} \leq \epsilon^2. \quad (\text{E.30})$$

- **Bound $\frac{C_e^2}{S_2^4}$ with ϵ^2** . This leads to

$$S_2 \geq \sqrt{\frac{C_e}{\epsilon}}. \quad (\text{E.31})$$

- **Bound on the variance.** First notice from (E.29) and $\gamma \leq \frac{1}{2L_F}$,

$$\begin{aligned} \frac{(1-p_{\text{in}})^3 \tilde{L}_F^2}{p_{\text{in}} S_2} \frac{6n^2}{B_2^2} \gamma^2 &\leq \frac{1}{4} \lesssim \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \\ L_F \gamma &\leq \frac{1}{2} \lesssim \frac{(1-p_{\text{out}})^2}{p_{\text{out}}}. \end{aligned}$$

Therefore, we only need to consider the upper bound on

$$\begin{aligned} &\frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{E}_{\text{var}}^{t+1} \\ &\leq 14 \cdot 32 \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \left(\left(\frac{p_{\text{out}}}{B_1} + \frac{1-p_{\text{out}}}{B_2} \right) \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18n^2}{B_2^2} + \frac{(1-p_{\text{out}})}{B_2} \right) \frac{\gamma^2 L_F^2}{T} \sum_{t=0}^{T-1} \|\mathbb{E}[G^{t+1}]\|_2^2 \\ &\quad + 14 \cdot 96 \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \left(\frac{p_{\text{out}}}{B_1} + \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} \right) \frac{\tilde{L}_F^2}{S_1} + \frac{(1-p_{\text{out}})^3}{p_{\text{out}} T} \frac{8\tilde{L}_F^2}{B_1 S_1}. \end{aligned}$$

We impose the constraints for each term

$$\begin{aligned} \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{p_{\text{out}}}{B_1} \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18n^2}{B_2^2} L_F^2 \gamma^2 &\lesssim 1 \\ \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{1-p_{\text{out}}}{B_2} \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18n^2}{B_2^2} L_F^2 \gamma^2 &\lesssim 1 \\ \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{1-p_{\text{out}}}{B_2} L_F^2 \gamma^2 &\lesssim 1 \\ \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{p_{\text{out}}}{B_1} \frac{\tilde{L}_F^2}{S_1} &\lesssim \epsilon^2 \\ \frac{(1-p_{\text{out}})^2}{p_{\text{out}}} \frac{(1-p_{\text{in}})(1-p_{\text{out}})}{B_2} \frac{\tilde{L}_F^2}{S_1} &\lesssim \epsilon^2 \\ \frac{(1-p_{\text{out}})^3}{p_{\text{out}} T} \frac{8\tilde{L}_F^2}{B_1 S_1} &\lesssim \epsilon^2. \end{aligned}$$

These can be simplified as

$$\gamma \lesssim \frac{p_{\text{in}} p_{\text{out}} \sqrt{B_1} \sqrt{S_1}}{(1-p_{\text{out}}) \sqrt{1-p_{\text{in}}}} \frac{1}{L_F} \quad (\text{E.32})$$

$$\gamma \lesssim \frac{p_{\text{in}} p_{\text{out}}^2 \sqrt{B_1} \sqrt{S_1}}{(1-p_{\text{out}})^{3/2} \sqrt{1-p_{\text{in}}}} \frac{1}{L_F} \quad (\text{E.33})$$

$$\gamma \lesssim \frac{\sqrt{B_1}}{(1-p_{\text{out}})^{3/2}} \frac{1}{L_F} \quad (\text{E.34})$$

$$B_1 S_1 \gtrsim \frac{(1-p_{\text{out}})^2 \tilde{L}_F^2}{\epsilon^2} \quad (\text{E.35})$$

$$B_1 S_1 \gtrsim \frac{(1-p_{\text{out}})^3 (1-p_{\text{in}}) \tilde{L}_F^2}{\epsilon^2 p_{\text{out}}^2} \quad (\text{E.36})$$

$$B_1 S_1 \gtrsim \frac{(1-p_{\text{out}})^3 \tilde{L}_F^2}{T \epsilon^2 p_{\text{out}}} \quad (\text{E.37})$$

- Constraints from Lemma E.15

$$\gamma^2 L_F^2 \max \left\{ \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18}{B_2}, \frac{1-p_{\text{out}}}{B_2} \frac{(1-p_{\text{in}})}{p_{\text{in}} S_2} \frac{18n^2}{B_2^2}, \frac{(1-p_{\text{out}})}{B_2} \right\} \leq \frac{1}{16} \cdot \frac{1}{6}$$

which can be translated to

$$\gamma \lesssim \frac{p_{\text{in}}\sqrt{S_1}\sqrt{B_2}}{L_F\sqrt{1-p_{\text{in}}}} \quad (\text{E.38})$$

$$\gamma \lesssim \frac{p_{\text{in}}p_{\text{out}}\sqrt{S_1}\sqrt{B_2}}{L_F\sqrt{1-p_{\text{in}}}\sqrt{1-p_{\text{out}}}} \quad (\text{E.39})$$

$$\gamma \lesssim \frac{\sqrt{B_2}}{L_F\sqrt{1-p_{\text{out}}}} \quad (\text{E.40})$$

- Constraint from sufficient decrease lemma:

$$\gamma \leq \frac{1}{2L_F}. \quad (\text{E.41})$$

We simplify the conditions noticing that 1) (E.37) is weaker than (E.35); 2) (E.34) and (E.40) are weaker than (E.41). Combine all the constraints on γ , i.e. (E.32), (E.33), (E.38), (E.39), (E.41)

$$\gamma \lesssim \frac{1}{L_F} \min \left\{ \min \left\{ 1, \frac{p_{\text{out}}}{\sqrt{1-p_{\text{out}}}} \right\} \frac{p_{\text{in}}p_{\text{out}}\sqrt{B_1}\sqrt{S_1}}{(1-p_{\text{out}})\sqrt{1-p_{\text{in}}}} \frac{1}{L_F}, \min \left\{ 1, \frac{p_{\text{out}}}{\sqrt{1-p_{\text{out}}}} \right\} \frac{p_{\text{in}}\sqrt{S_1}\sqrt{B_2}}{\sqrt{1-p_{\text{in}}}}, 1, \frac{p_{\text{out}}p_{\text{in}}\sqrt{S_1}}{5L_F(1-p_{\text{in}})^{3/2}} \right\}.$$

This can be simplified as an upper bound

$$\gamma \lesssim \frac{1}{L_F} \min \left\{ \frac{p_{\text{in}}p_{\text{out}}\sqrt{S_1}}{\sqrt{1-p_{\text{in}}}}, \frac{p_{\text{in}}p_{\text{out}}\sqrt{S_1}\sqrt{B_1}}{\sqrt{1-p_{\text{out}}}}, \frac{p_{\text{in}}p_{\text{out}}^2\sqrt{S_1}\sqrt{B_1}}{\sqrt{1-p_{\text{in}}}\sqrt{1-p_{\text{out}}}}, 1 \right\}.$$

Now we consider two sets of hyperparameters depending on the size of n **Case 1:** For $n = \mathcal{O}(\epsilon^{-2/3})$, we choose the following set of hyperparameters

$$B_1 = B_2 = n, \quad p_{\text{out}} = 1, \quad S_1 = \tilde{L}_F^2 \epsilon^{-2}, \quad S_2 = \tilde{L}_F \epsilon^{-1}, \quad p_{\text{in}} = \tilde{L}_F^{-1} \epsilon.$$

Then we have $\gamma \lesssim \frac{1}{L_F} \min \left\{ \frac{p_{\text{in}}\sqrt{S_1}}{\sqrt{1-p_{\text{in}}}}, 1 \right\} = \frac{1}{L_F}$, we have the total sample complexity of

$$B_2 S_2 T = \frac{B_2 S_2 T \gamma}{\gamma} \stackrel{(\text{E.28})}{=} \frac{F(\mathbf{x}^0) - F^*}{\epsilon^2} \frac{B_2 S_2}{\gamma} = \frac{(F(\mathbf{x}^0) - F^*) n \tilde{L}_F L_F}{\epsilon^3}$$

Case 2: For $n = \Omega(\epsilon^{-2/3})$, we choose the following set of hyperparameters

$$B_1 = n, \quad B_2 = \sqrt{n}, \quad p_{\text{out}} = \frac{1}{\sqrt{n}}.$$

In this case, (E.35) is stronger than (E.36) which requires $S_1 \gtrsim \frac{\tilde{L}_F^2}{n\epsilon^2}$

$$S_1 = S_2 = \max \left\{ \tilde{\sigma}_{\text{bias}}^{1/2} \epsilon^{-1/2}, \frac{\sigma_{\text{in}}^2}{n\epsilon^2} \right\}, \quad p_{\text{in}} = 1$$

Then we have $\gamma \lesssim \frac{1}{L_F} \min\{\frac{p_{\text{out}}\sqrt{n}\sqrt{S_1}}{\sqrt{1-p_{\text{out}}}}, 1\} = \frac{1}{L_F}$, we have the total sample complexity of

$$B_2 S_2 T = \frac{B_2 S_2 T \gamma}{\gamma} \stackrel{\text{(E.28)}}{=} \frac{F(\mathbf{x}^0) - F^*}{\epsilon^2} \frac{B_2 S_2}{\gamma} = (F(\mathbf{x}^0) - F^*) \max\left\{\frac{\sqrt{n}\tilde{\sigma}_{\text{bias}}^{1/2}}{\epsilon^{2.5}}, \frac{\sigma_{\text{in}}^2}{\sqrt{n}\epsilon^4}\right\}.$$

By picking \mathbf{x}^s uniformly at random among $\{\mathbf{x}^t\}_{t=0}^{T-1}$, we get the desired guarantee. \square

E.6 Missing Details from Section 2.7

E.6.1 Application of First-order MAML

Over the past few years, the MAML framework [Finn et al., 2017] has become quite popular for few-shot supervised learning and meta reinforcement learning tasks. The first-order Model-Agnostic Meta-Learning (MAML) can be formulated mathematically as follows:

$$\min_{\mathbf{x}} \mathbb{E}_{i \sim p, \mathcal{D}_{\text{query}}^i} \ell_i \left(\mathbb{E}_{\mathcal{D}_{\text{supp}}^i} (\mathbf{x} - \alpha \nabla \ell_i(\mathbf{x}, \mathcal{D}_{\text{supp}}^i)), \mathcal{D}_{\text{query}}^i \right)$$

where α is the step size, $\mathcal{D}_{\text{supp}}^i$ and $\mathcal{D}_{\text{query}}^i$ are meta-training and meta-testing data respectively and ℓ_i being the loss function of task i . Stated in the CSO framework, $f_{\xi}(\mathbf{x}) := \ell_i(\mathbf{x}, \mathcal{D}_{\text{query}}^i)$ and $g_{\eta}(\mathbf{x}, \xi) := \mathbf{x} - \alpha \nabla \ell_i(\mathbf{x}, \mathcal{D}_{\text{supp}}^i)$ where $\xi = (i, \mathcal{D}_{\text{query}}^i)$ and $\eta = \mathcal{D}_{\text{supp}}^i$.

In this context, lots of popular choices for f_{ξ} are smooth. For illustration purposes, we now discuss a widely used sine-wave few-shot regression task as appearing from the work of Finn et al. [2017], where the goal is to do a few-shot learning of a sine wave, $A \sin(t - \phi)$, using a neural network $\Phi_{\mathbf{x}}(t)$ with smooth activations, where A and ϕ represent the unknown amplitude and phase, and \mathbf{x} denotes the model weight. Each task i is characterized by $(A^i, \phi^i, \mathcal{D}_{\text{query}}^i)$. In the first-order MAML training process, we randomly select a task i , and draw training data $\eta = \mathcal{D}_{\text{supp}}^i$. Define the loss function for a given dataset \mathcal{D} as $\ell_i(\Phi_{\mathbf{x}}; \mathcal{D}) = \frac{1}{2} \mathbb{E}_{t \sim \mathcal{D}} \|A^i \sin(t - \phi^i) - \Phi_{\mathbf{x}}(t)\|_2^2$. We then establish the outer function $f_i(\mathbf{x}) = \ell_i(\Phi_{\mathbf{x}}; \mathcal{D}_{\text{query}}^i)$ and inner function $g_{\eta}(\mathbf{x}) = \mathbf{x} - \alpha \nabla_{\mathbf{x}} \ell_i(\Phi_{\mathbf{x}}; \mathcal{D}_{\text{supp}}^i)$. As f_i is smooth, our results are applicable.

In Figure E.1, we show the results of BSGD and E-BSGD applied to this problem. In this experiment, the amplitude A is drawn from a uniform distribution $\mathcal{U}(0.1, 5)$ and the phase ϕ is drawn from $\mathcal{U}(0, \pi)$. Both $\mathcal{D}_{\text{supp}}$ and $\mathcal{D}_{\text{query}}$ are independently drawn from $\mathcal{U}(-5, 5)$. The step size is set to $\alpha = 0.01$. The batch size is fixed to 10. The performances of BSGD and E-BSGD are very close. This is not surprising because finetuning step size α is chosen to be small which significantly reduces the variance of g_{η} , making the bias of meta gradient to be very small ($\mathcal{O}(\alpha^2)$). Therefore, we observe similar performance of BSGD and E-BSGD. Similar trend also holds for BSpiderBoost and NestedVR compared to their extrapolated variants.

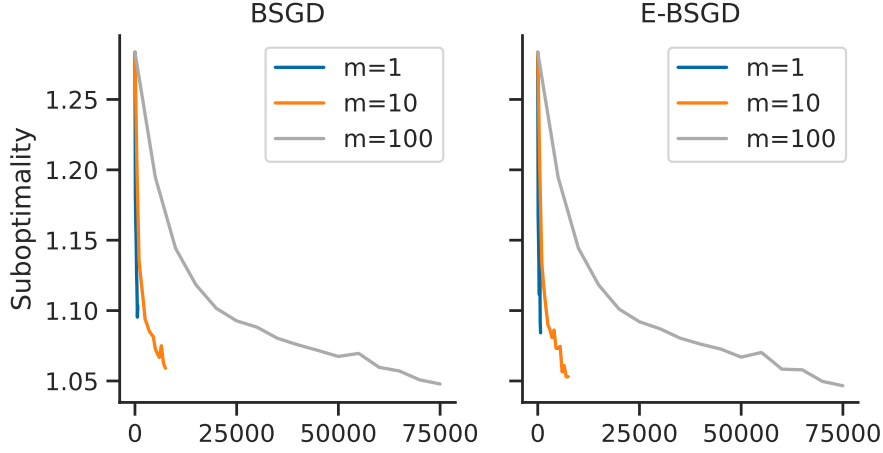


Fig. E.1 Performance of BSGD vs. E-BSGD on the few-shot sinusoid regression task.

E.6.2 Application of Deep Average Precision Maximization

The areas under precision-recall curve (AUPRC) has an unbiased point estimator that maximizes average precision (AP) [Qi et al., 2021a; Wang et al., 2022a]. Let \mathcal{S}_+ and \mathcal{S}_- be the set of positive and negative samples and $\mathcal{S} = \mathcal{S}_- \cup \mathcal{S}_+$. Let $h_{\mathbf{w}}(\cdot)$ be a classifier parameterized with \mathbf{w} and ℓ be a surrogate function, such as logistic or sigmoid. A smooth surrogate objective for maximizing average precision can be formulated as [Wang and Yang, 2022]:

$$F(\mathbf{w}) = -\frac{1}{|\mathcal{S}_+|} \sum_{\mathbf{x}_i \in \mathcal{S}_+} \frac{\sum_{\mathbf{x} \in \mathcal{S}_+} \ell(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}_i))}{\sum_{\mathbf{x} \in \mathcal{S}} \ell(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}_i))}$$

This problem can be seen as a conditional stochastic optimization problem with $g_i(\mathbf{w}) = [\sum_{\mathbf{x} \in \mathcal{S}_+} \ell(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}_i)), \sum_{\mathbf{x} \in \mathcal{S}} \ell(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}_i))]$ and $f_i : \mathbb{R} \times \mathbb{R} \setminus \{0\} \rightarrow \mathbb{R}$ is defined as $f_i(\mathbf{y}) = -\frac{[\mathbf{y}]_1}{[\mathbf{y}]_2}$ where $[\mathbf{y}]_k$ denotes the k th coordinate of a vector $\mathbf{y} \in \mathbb{R} \times \mathbb{R} \setminus \{0\}$. During the stochastic optimization of this objective, we draw uniformly at random $\xi := \mathbf{x}_i$ (drawn from the set \mathcal{S}_+) as a positive sample and $\eta|\xi = [\mathcal{F}_{\mathbf{x}_1}, \mathcal{F}_{\mathbf{x}_2}]$ where set \mathbf{x}_1 is drawn uniformly at random from \mathcal{S}_+ and \mathbf{x}_2 is drawn uniformly at random from \mathcal{S} and functional $\mathcal{F}_{\mathbf{x}}(\mathbf{w}) := \ell(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}_i))$. Note that $f_i \in \mathcal{C}^\infty$ is smooth with gradient

$$\nabla f_i(\mathbf{y}) = \begin{bmatrix} -\frac{1}{[\mathbf{y}]_2} \\ \frac{[\mathbf{y}]_1}{([\mathbf{y}]_2)^2} \end{bmatrix}.$$

Therefore, our results from Sections 6.4 and 6.5 again apply.

E.6.3 Necessity of Additional Smoothness Conditions

Throughout the paper, we assume bounded moments (Assumption B) and a smoothness condition (Assumption C) to derive our extrapolation technique. However, it is worth noting that the technique itself does not explicitly depend on higher-order derivatives. Our theoretical framework does not address the behavior of extrapolation in the absence of these smoothness constraints. In this section, we investigate the application of extrapolation to two non-smooth functions:

- ReLU function given by $q(x) = \max\{x, 0\}$;
- Perturbed quadratics represented as $q(x) = x^2/2 + \text{TriangleWave}(x) + 1$. The function $\text{TriangleWave}(x)$ has a period of 2 and spans the range $[-1, 1]$, defined as:

$$\text{TriangleWave}(x) = 2 \left| 2 \left(\frac{x}{2} - \left\lfloor \frac{x}{2} + \frac{1}{2} \right\rfloor \right) \right| - 1$$

Visual representations of these functions can be found in Figure E.2c. We set $s = 0$ and consider a random variable $\delta \sim \mathcal{N}(10, 100)$ with $m = 1$. We then apply first-, second-, and third-order extrapolation. The outcomes are depicted in Figure E.2. Remarkably, both the ReLU and the perturbed quadratic functions do not conform to the differentiability assumptions inherent to our stochastic extrapolation schemes. Nonetheless, as indicated by Figure E.2a and Figure E.2b, our proposed second- and third-order extrapolation techniques yield a superior approximation of $q(\mathbb{E}[\delta])$.

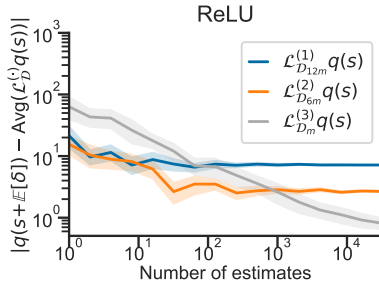


Fig. E.2a

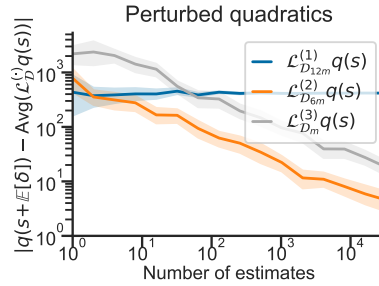


Fig. E.2b

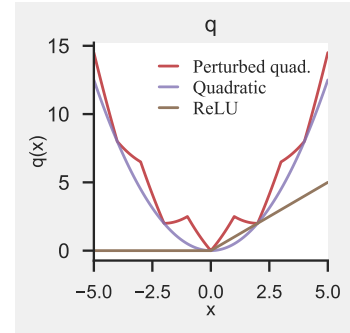


Fig. E.2c

Fig. E.2 (a) Fig. E.2a: Error in estimating $q(s + \mathbb{E}[\delta])$ for our proposed first-, second-, and third-order extrapolation schemes applied to ReLU $q(x) = \max\{x, 0\}$, $s = 0$, $\delta \sim \mathcal{N}(10, 100)$, $m = 1$. (b) Fig E.2b: Error in estimating $q(s + \mathbb{E}[\delta])$ for our proposed first-, second-, and third-order extrapolation schemes applied to a perturbed quadratic $q(x) = x^2/2 + \text{TriangleWave}(x) + 1$, $s = 0$, $\delta \sim \mathcal{N}(10, 100)$, $m = 1$. The $\text{TriangleWave}(x)$ has a period of 2 and spans the range $[-1, 1]$, i.e. $2|2(\frac{x}{2} - \lfloor \frac{x}{2} + \frac{1}{2} \rfloor)| - 1$. (c) Fig E.2c: The ReLU and perturbed quadratic used in the Fig. 5a and 5b along with quadratic curves.

References

- Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.
- Ittai Abraham, T-H. Hubert Chan, Danny Dolev, Kartik Nayak, Rafael Pass, Ling Ren, and Elaine Shi. Communication complexity of byzantine agreement, revisited, 2020.
- Anish Acharya, Abolfazl Hashemi, Prateek Jain, Sujay Sanghavi, Inderjit S Dhillon, and Ufuk Topcu. Robust training in high dimensions via block coordinate geometric median descent. *ArXiv preprint*, abs/2106.08882, 2021. URL <https://arxiv.org/abs/2106.08882>.
- Alekh Agarwal and John C. Duchi. Distributed delayed stochastic optimization. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 873–881, 2011. URL <https://proceedings.neurips.cc/paper/2011/hash/f0e52b27a7a5d6a1a87373dffa53dbe5-Abstract.html>.
- Mehrdad Aliasgari, Marina Blanton, Yihua Zhang, and Aaron Steele. Secure computation on floating point numbers. In *NDSS*, 2013.
- Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine stochastic gradient descent. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 4618–4628, 2018. URL <https://proceedings.neurips.cc/paper/2018/hash/a07c2f3b3b907aaf8436a26c6d77f0a2-Abstract.html>.
- Zeyuan Allen-Zhu, Faeze Ebrahimiaghazani, Jerry Li, and Dan Alistarh. Byzantine-resilient non-convex stochastic gradient descent. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL <https://openreview.net/forum?id=PbEHqvFtcS>.

- Zeyuan Allen-Zhu, Faeze Ebrahimiaghazani, Jerry Li, and Dan Alistarh. Byzantine-resilient non-convex stochastic gradient descent. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b. URL <https://openreview.net/forum?id=PbEHqvFtcS>.
- Fabio Anselmi, Joel Z Leibo, Lorenzo Rosasco, Jim Mutch, Andrea Tacchetti, and Tomaso Poggio. Unsupervised learning of invariant representations. *Theoretical Computer Science*, 633:112–121, 2016.
- Yossi Arjevani, Yair Carmon, John C. Duchi, Dylan J. Foster, Ayush Sekhari, and Karthik Sridharan. Second-order information in non-convex stochastic optimization: Power and limitations. In Jacob D. Abernethy and Shivani Agarwal, editors, *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 242–299. PMLR, 2020. URL <http://proceedings.mlr.press/v125/arjevani20a.html>.
- Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, pages 1–50, 2022.
- Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Michael G. Rabbat. Stochastic gradient push for distributed deep learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 344–353. PMLR, 2019a. URL <http://proceedings.mlr.press/v97/assran19a.html>.
- Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Michael G. Rabbat. Stochastic gradient push for distributed deep learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 344–353. PMLR, 2019b. URL <http://proceedings.mlr.press/v97/assran19a.html>.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2016.
- Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 2938–2948. PMLR, 2020a. URL <http://proceedings.mlr.press/v108/bagdasaryan20a.html>.

- Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 2938–2948. PMLR, 2020b. URL <http://proceedings.mlr.press/v108/bagdasaryan20a.html>.
- Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8632–8642, 2019. URL <https://proceedings.neurips.cc/paper/2019/hash/ec1c59141046cd1866bbbcdfb6ae31d4-Abstract.html>.
- Donald Beaver. Efficient multiparty protocols using circuit randomization. In *Annual International Cryptology Conference*, pages 420–432. Springer, 1991.
- Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd with majority vote is communication efficient and fault tolerant. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019a. URL <https://openreview.net/forum?id=BJxhijAcY7>.
- Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd with majority vote is communication efficient and fault tolerant. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019b. URL <https://openreview.net/forum?id=BJxhijAcY7>.
- Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin B. Calo. Analyzing federated learning through an adversarial lens. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 634–643. PMLR, 2019. URL <http://proceedings.mlr.press/v97/bhagoji19a.html>.
- Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 119–129, 2017. URL <https://proceedings.neurips.cc/paper/2017/hash/f4b9ec30ad9f68f89b29639786cb62ef-Abstract.html>.

- Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM journal on Computing*, 13(4):850–864, 1984.
- Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy preserving machine learning. *IACR Cryptol. ePrint Arch.*, 2017:281, 2017. URL <http://eprint.iacr.org/2017/281>.
- Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konečný, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. In Ameet Talwalkar, Virginia Smith, and Matei Zaharia, editors, *Proceedings of Machine Learning and Systems 2019, MLSys 2019, Stanford, CA, USA, March 31 - April 2, 2019*. mlsys.org, 2019. URL <https://proceedings.mlsys.org/book/271.pdf>.
- Maria Borge, Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, and Bryan Ford. Proof-of-personhood: Redemocratizing permissionless cryptocurrencies. In *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 23–26. IEEE, 2017.
- Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE transactions on information theory*, 52(6):2508–2530, 2006.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- Sébastien Bubeck, Qijia Jiang, Yin Tat Lee, Yuanzhi Li, and Aaron Sidford. Near-optimal method for highly smooth convex optimization. In Alina Beygelzimer and Daniel Hsu, editors, *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pages 492–507. PMLR, 2019. URL <http://proceedings.mlr.press/v99/bubeck19a.html>.
- Laurent Bulteau, Gal Shahaf, Ehud Shapiro, and Nimrod Talmon. Aggregation over metric spaces: Proposing and voting in elections, budgeting, and legislation. *Journal of Artificial Intelligence Research*, 70:1413–1439, 2021.
- Lukas Burkhalter, Hidde Lycklama, Alexander Viand, Nicolas Küchler, and Anwar Hithnawi. Roff: Attestable robustness for secure federated learning, 2021.
- William Cappelletti. Byzantine-robust decentralized optimization for Machine Learning, 20c. URL <https://arxiv.org/abs/c>.

- Melissa Chase, Ran Gilad-Bachrach, Kim Laine, Kristin E Lauter, and Peter Rindal. Private collaborative neural network learning. *IACR Cryptology ePrint Archive*, 2017:762, 2017.
- Lingjiao Chen, Hongyi Wang, Zachary B. Charles, and Dimitris S. Papailiopoulos. DRACO: byzantine-resilient distributed training via redundant gradients. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 902–911. PMLR, 2018. URL <http://proceedings.mlr.press/v80/chen18l.html>.
- Valerie Chen, Valerio Pastro, and Mariana Raykova. Secure computation for machine learning with spdz. *ArXiv preprint*, abs/1901.00329, 2019. URL <https://arxiv.org/abs/1901.00329>.
- Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. In *PERV*, 2017a. URL <https://api.semanticscholar.org/CorpusID:58534983>.
- Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2): 1–25, 2017b. ISSN 2476-1249. doi: 10.1145/3154503. URL <http://dx.doi.org/10.1145/3154503>.
- Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25, 2017c.
- Edward Chou, Josh Beal, Daniel Levy, Serena Yeung, Albert Haque, and Li Fei-Fei. Faster cryptonets: Leveraging sparsity for real-world encrypted inference. *ArXiv preprint*, abs/1811.09953, 2018. URL <https://arxiv.org/abs/1811.09953>.
- Michael B. Cohen, Yin Tat Lee, Gary L. Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 9–21. ACM, 2016. doi: 10.1145/2897518.2897647. URL <https://doi.org/10.1145/2897518.2897647>.
- Henry Corrigan-Gibbs and Dan Boneh. Prio: Private, robust, and scalable computation of aggregate statistics. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, pages 259–282, 2017.
- Bo Dai, Niao He, Yunpeng Pan, Byron Boots, and Le Song. Learning from conditional distributions via dual embeddings. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1458–1467. PMLR, 2017. URL <http://proceedings.mlr.press/v54/dai17a.html>.

- Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. SBEED: convergent reinforcement learning with nonlinear function approximation. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1133–1142. PMLR, 2018. URL <http://proceedings.mlr.press/v80/dai18c.html>.
- Georgios Damaskinos, El Mahdi El Mhamdi, Rachid Guerraoui, Rhicheck Patra, and Mahsa Taziki. Asynchronous byzantine machine learning (the case of SGD). In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1153–1162. PMLR, 2018. URL <http://proceedings.mlr.press/v80/damaskinos18a.html>.
- Georgios Damaskinos, El-Mahdi El-Mhamdi, Rachid Guerraoui, Arsany Guirguis, and Sébastien Rouault. AGGREGATHOR: byzantine machine learning via robust gradient aggregation. In Ameet Talwalkar, Virginia Smith, and Matei Zaharia, editors, *Proceedings of Machine Learning and Systems 2019, MLSys 2019, Stanford, CA, USA, March 31 - April 2, 2019*. mlsys.org, 2019. URL <https://proceedings.mlsys.org/book/280.pdf>.
- Deepesh Data and Suhas Diggavi. Byzantine-resilient sgd in high dimensions on heterogeneous data. *arXiv 2005.07866*, 2020.
- Deepesh Data and Suhas Diggavi. Byzantine-resilient sgd in high dimensions on heterogeneous data. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 2310–2315. IEEE, 2021a.
- Deepesh Data and Suhas N. Diggavi. Byzantine-resilient high-dimensional SGD with local iterations on heterogeneous data. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 2478–2488. PMLR, 2021b. URL <http://proceedings.mlr.press/v139/data21a.html>.
- Allison Davis, Burleigh Bradford Gardner, and Mary R Gardner. *Deep South: A social anthropological study of caste and class*. Univ of South Carolina Press, 1930.
- Aaron Defazio, Francis R. Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1646–1654, 2014. URL <https://proceedings.neurips.cc/paper/2014/hash/ede7e2b6d13a41ddf9f4bdef84fdc737-Abstract.html>.

- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 20-25 June 2009, Miami, Florida, USA, pages 248–255. IEEE Computer Society, 2009. doi: 10.1109/CVPR.2009.5206848. URL <https://doi.org/10.1109/CVPR.2009.5206848>.
- Danny Dolev and H. Raymond Strong. Authenticated algorithms for byzantine agreement. *SIAM J. Comput.*, 12:656–666, 1983.
- Danny Dolev, Nancy A Lynch, Shlomit S Pinter, Eugene W Stark, and William E Weihl. Reaching approximate agreement in the presence of faults. *Journal of the ACM (JACM)*, 33(3):499–516, 1986.
- Bradley Efron. *Bootstrap methods: another look at the jackknife*. Springer, 1992.
- El Mahdi El-Mhamdi, Sadegh Farhadkhani, Rachid Guerraoui, Arsany Guirguis, Lê-Nguyên Hoang, and Sébastien Rouault. Collaborative learning in the jungle (decentralized, byzantine, heterogeneous, asynchronous and nonconvex learning). *Advances in Neural Information Processing Systems*, 34, 2021.
- Yuri M Ermoliev and Vladimir I Norkin. Sample average approximation method for compound stochastic optimization problems. *SIAM Journal on Optimization*, 23(4):2231–2263, 2013.
- Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 211–222, 2003.
- Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. SPIDER: near-optimal non-convex optimization via stochastic path-integrated differential estimator. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 687–697, 2018. URL <https://proceedings.neurips.cc/paper/2018/hash/1543843a4723ed2ab08e18053ae6dc5b-Abstract.html>.
- Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 2017. URL <http://proceedings.mlr.press/v70/finn17a.html>.
- Michael J Fischer, Nancy A Lynch, and Michael Merritt. Easy impossibility proofs for distributed consensus problems. *Distributed Computing*, 1(1):26–39, 1986.

- Bryan Ford. 10. technologizing democracy or democratizing technology? a layered-architecture perspective on potentials and challenges. In *Digital Technology and Democratic Theory*, pages 274–321. University of Chicago Press, 2021.
- Leonidas Georgopoulos. *Definitive Consensus for Distributed Data Inference*. PhD thesis, EPFL, 2011.
- Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.
- Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. Robust federated learning in a heterogeneous environment. *ArXiv preprint*, abs/1906.06629, 2019. URL <https://arxiv.org/abs/1906.06629>.
- Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin E. Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 201–210. JMLR.org, 2016. URL <http://proceedings.mlr.press/v48/gilad-bachrach16.html>.
- Takashi Goda and Wataru Kitade. Constructing unbiased gradient estimators with finite variance for conditional stochastic optimization. *ArXiv preprint*, abs/2206.01991, 2022. URL <https://arxiv.org/abs/2206.01991>.
- Eduard Gorbunov, Alexander Borzunov, Michael Diskin, and Max Ryabinin. Secure distributed training at scale, 2021.
- Rachid Guerraoui, Sébastien Rouault, et al. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*, pages 3521–3530. PMLR, 2018.
- Shangwei Guo, Tianwei Zhang, Xiaofei Xie, Lei Ma, Tao Xiang, and Yang Liu. Towards byzantine-resilient learning in decentralized systems. *arXiv 2002.08569*, 2020.
- Shangwei Guo, Tianwei Zhang, Han Yu, Xiaofei Xie, Lei Ma, Tao Xiang, and Yang Liu. Byzantine-resilient decentralized stochastic gradient descent, 2021.
- Nirupam Gupta and Nitin H Vaidya. Resilience in collaborative optimization: redundant and independent cost functions. *ArXiv preprint*, abs/2003.09675, 2020. URL <https://arxiv.org/abs/2003.09675>.
- Nirupam Gupta, Thinh T Doan, and Nitin Vaidya. Byzantine fault-tolerance in federated local sgd under 2f-redundancy. *ArXiv preprint*, abs/2108.11769, 2021. URL <https://arxiv.org/abs/2108.11769>.

- Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1737–1746. JMLR.org, 2015. URL <http://proceedings.mlr.press/v37/gupta15.html>.
- Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008a.
- Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using NetworkX. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008b.
- Yanjun Han, Jiantao Jiao, and Tsachy Weissman. Minimax estimation of divergences between discrete distributions. *IEEE Journal on Selected Areas in Information Theory*, 1(3):814–823, 2020.
- W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
- Lie He and Shiva Prasad Kasiviswanathan. Debiasing conditional stochastic optimization. *CoRR*, abs/2304.10613, 2023. doi: 10.48550/arXiv.2304.10613. URL <https://doi.org/10.48550/arXiv.2304.10613>.
- Lie He, Sai Praneeth Karimireddy, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via resampling, 2020a.
- Lie He, Sai Praneeth Karimireddy, and Martin Jaggi. Secure byzantine-robust machine learning. *ArXiv preprint*, abs/2006.04747, 2020b. URL <https://arxiv.org/abs/2006.04747>.
- Lie He, Sai Praneeth Karimireddy, and Martin Jaggi. Byzantine-robust decentralized learning via clippedgossip. *ArXiv preprint*, abs/2202.01545, 2022. URL <https://arxiv.org/abs/2202.01545>.
- Julien M. Hendrickx, Raphaël M. Jungers, Alexander Olshevsky, and Guillaume Vankeerberghen. Graph diameter, eigenvalues, and minimum-time consensus. *Automatica*, 50(2):635–640, 2014. doi: <https://doi.org/10.1016/j.automatica.2013.11.034>. URL <https://www.sciencedirect.com/science/article/pii/S0005109813005517>.

- Ehsan Hesamifard, Hassan Takabi, and Mehdi Ghasemi. Cryptodl: Deep neural networks over encrypted data. *ArXiv preprint*, abs/1711.05189, 2017. URL <https://arxiv.org/abs/1711.05189>.
- Martin Hirt and Pavel Raykov. Multi-valued byzantine broadcast: The $t < n$ case. In *ASIACRYPT*, 2014.
- Yifan Hu, Xin Chen, and Niao He. Sample complexity of sample average approximation for conditional stochastic optimization. *SIAM Journal on Optimization*, 30(3):2103–2133, 2020a.
- Yifan Hu, Siqi Zhang, Xin Chen, and Niao He. Biased stochastic first-order methods for conditional stochastic optimization and applications in meta learning. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020b. URL <https://proceedings.neurips.cc/paper/2020/hash/1cdf14d1e3699d61d237cf76ce1c2dca-Abstract.html>.
- Yifan Hu, Xin Chen, and Niao He. On the bias-variance-cost tradeoff of stochastic optimization. *Advances in Neural Information Processing Systems*, 34:22119–22131, 2021.
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org, 2015. URL <http://proceedings.mlr.press/v37/ioffe15.html>.
- Prateek Jain, Purushottam Kar, et al. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–363, 2017.
- Sylvain Jeaugey. Massively scale your deep learning training with NCCL 2.4. <https://devblogs.nvidia.com/massively-scale-deep-learning-training-nccl-2-4/>, 2019. [Online; accessed 21-May-2019].
- Wei Jiang, Gang Li, Yibo Wang, Lijun Zhang, and Tianbao Yang. Multi-block-single-probe variance reduced estimator for coupled compositional optimization. *ArXiv preprint*, abs/2207.08540, 2022. URL <https://arxiv.org/abs/2207.08540>.
- Jiantao Jiao and Yanjun Han. Bias correction with jackknife, bootstrap, and taylor series. *IEEE Transactions on Information Theory*, 66(7):4392–4418, 2020.
- Björn Johansson, Maben Rabi, and Mikael Johansson. A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM J. Optim.*, 20(3):1157–1170, 2009. doi: 10.1137/08073038X. URL <https://doi.org/10.1137/08073038X>.

- Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 315–323, 2013. URL <https://proceedings.neurips.cc/paper/2013/hash/ac1dd209cbcc5e5d1c6e28598e8cbbe8-Abstract.html>.
- Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. GAZELLE: A low latency framework for secure neural network inference. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1651–1669, 2018.
- Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D’Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrede Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *ArXiv preprint*, abs/1912.04977, 2019. URL <https://arxiv.org/abs/1912.04977>.
- Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian U. Stich, and Martin Jaggi. Error feedback fixes signsgd and other gradient compression schemes. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3252–3261. PMLR, 2019. URL <http://proceedings.mlr.press/v97/karimireddy19a.html>.
- Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. *ArXiv preprint*, abs/2006.09365, 2020a. URL <https://arxiv.org/abs/2006.09365>.
- Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR, 2020b. URL <http://proceedings.mlr.press/v119/karimireddy20a.html>.

- Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for byzantine robust optimization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5311–5319. PMLR, 2021a. URL <http://proceedings.mlr.press/v139/karimireddy21a.html>.
- Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for byzantine robust optimization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5311–5319. PMLR, 2021b. URL <http://proceedings.mlr.press/v139/karimireddy21a.html>.
- Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing, 2021c.
- Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.
- Marcel Keller, Emmanuela Orsini, and Peter Scholl. Mascot: faster malicious arithmetic secure computation with oblivious transfer. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 830–842, 2016.
- Marcel Keller, Valerio Pastro, and Dragos Rotaru. Overdrive: making SPDZ great again. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 158–189. Springer, 2018.
- David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 482–491. IEEE Computer Society, 2003. doi: 10.1109/SFCS.2003.1238221. URL <https://doi.org/10.1109/SFCS.2003.1238221>.
- Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local SGD on identical and heterogeneous data. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 4519–4529. PMLR, 2020. URL <http://proceedings.mlr.press/v108/bayoumi20a.html>.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL <http://arxiv.org/abs/1412.6980>.
- Chih-Kai Ko. *On Matrix Factorization and Scheduling for Finite-time Average-consensus*. PhD thesis, California Institute of Technology, 2010.

- Anastasia Koloskova, Sebastian U. Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3478–3487. PMLR, 2019. URL <http://proceedings.mlr.press/v97/koloskova19a.html>.
- Anastasia Koloskova, Tao Lin, Sebastian U. Stich, and Martin Jaggi. Decentralized deep learning with arbitrary communication compression. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020a. URL <https://openreview.net/forum?id=SkgGCKrKvH>.
- Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U. Stich. A unified theory of decentralized SGD with changing topology and local updates. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5381–5393. PMLR, 2020b. URL <http://proceedings.mlr.press/v119/koloskova20a.html>.
- Anastasiia Koloskova, Tao Lin, and Sebastian U Stich. An improved analysis of gradient tracking for decentralized machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.
- Lingjing Kong, Tao Lin, Anastasia Koloskova, Martin Jaggi, and Sebastian U. Stich. Consensus control for decentralized deep learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5686–5696. PMLR, 2021. URL <http://proceedings.mlr.press/v139/kong21a.html>.
- Dmitry Kovalev, Anastasia Koloskova, Martin Jaggi, Peter Richtárik, and Sebastian U. Stich. A linearly convergent algorithm for decentralized optimization: Sending less bits for free! In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 4087–4095. PMLR, 2021. URL <http://proceedings.mlr.press/v130/kovalev21a.html>.
- Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 2012.
- Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (Canadian Institute for Advanced Research). URL <http://www.cs.toronto.edu/~kriz/cifar.html>.
- Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pages 203–226. 2019.

- Heath J LeBlanc, Haotian Zhang, Xenofon Koutsoukos, and Shreyas Sundaram. Resilient asymptotic consensus in robust networks. *IEEE Journal on Selected Areas in Communications*, 31(4):766–781, 2013.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Liping Li, Wei Xu, Tianyi Chen, Georgios B. Giannakis, and Qing Ling. RSA: byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 1544–1551. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33011544. URL <https://doi.org/10.1609/aaai.v33i01.33011544>.
- Xiang Li, Wenhao Yang, Shusen Wang, and Zhihua Zhang. Communication-efficient local decentralized sgd methods, 2021.
- Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5330–5340, 2017a. URL <https://proceedings.neurips.cc/paper/2017/hash/f75526659f31040afeb61cb7133e4e6d-Abstract.html>.
- Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5330–5340, 2017b. URL <https://proceedings.neurips.cc/paper/2017/hash/f75526659f31040afeb61cb7133e4e6d-Abstract.html>.
- Tao Lin, Sai Praneeth Karimireddy, Sebastian U. Stich, and Martin Jaggi. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6654–6665. PMLR, 2021a. URL <http://proceedings.mlr.press/v139/lin21c.html>.

- Tao Lin, Sai Praneeth Karimireddy, Sebastian U. Stich, and Martin Jaggi. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6654–6665. PMLR, 2021b. URL <http://proceedings.mlr.press/v139/lin21c.html>.
- Hanxiao Liu, Andy Brock, Karen Simonyan, and Quoc Le. Evolving normalization-activation layers. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/9d4c03631b8b0c85ae08bf05eda37d0f-Abstract.html>.
- Jian Liu, Mika Juuti, Yao Lu, and Nadarajah Asokan. Oblivious neural network predictions via minionn transformations. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 619–631. ACM, 2017.
- Paolo Di Lorenzo and Gesualdo Scutari. Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016. doi: 10.1109/TSIPN.2016.2524588.
- Yucheng Lu and Christopher De Sa. Optimal complexity in decentralized training. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 7111–7123. PMLR, 2021. URL <http://proceedings.mlr.press/v139/lu21a.html>.
- Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3331–3340. PMLR, 2018. URL <http://proceedings.mlr.press/v80/ma18a.html>.
- Kalikinkar Mandal, Guang Gong, and Chuyi Liu. Nike-based fast privacy-preserving highdimensional data aggregation for mobile devices. Technical report, CACR Technical Report, CACR 2018-10, University of Waterloo, Canada, 2018.
- Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial*

- Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017a. URL <http://proceedings.mlr.press/v54/mcmahan17a.html>.
- Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017b. URL <http://proceedings.mlr.press/v54/mcmahan17a.html>.
- Edward Meeds, Remco Hendriks, Said Al Faraby, Magiel Bruntink, and Max Welling. Mlittb: machine learning in the browser. *PeerJ Computer Science*, 1:e11, 2015. ISSN 2376-5992. doi: 10.7717/peerj-cs.11. URL <http://dx.doi.org/10.7717/peerj-cs.11>.
- Si Yi Meng, Sharan Vaswani, Issam Hadj Laradji, Mark Schmidt, and Simon Lacoste-Julien. Fast and furious convergence: Stochastic second order methods under interpolation. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 1375–1386. PMLR, 2020. URL <http://proceedings.mlr.press/v108/meng20a.html>.
- El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in byzantium. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3518–3527. PMLR, 2018. URL <http://proceedings.mlr.press/v80/mhamdi18a.html>.
- El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. Distributed momentum for byzantine-resilient stochastic gradient descent. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL <https://openreview.net/forum?id=H8UHdhWG6A3>.
- El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. Distributed momentum for byzantine-resilient stochastic gradient descent. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b. URL <https://openreview.net/forum?id=H8UHdhWG6A3>.
- Stanislav Minsker et al. Geometric median and robust estimation in banach spaces. *Bernoulli*, 21(4):2308–2335, 2015.

- Ken Miura and Tatsuya Harada. Implementation of a practical distributed calculation system with browsers and javascript, and application to distributed deep learning. *arXiv 1503.05743*, 2015.
- Payman Mohassel and Yupeng Zhang. SecureML: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 19–38. IEEE, 2017.
- Youssef Mroueh, Stephen Voinea, and Tomaso A. Poggio. Learning with group invariant features: A kernel perspective. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1558–1566, 2015. URL <https://proceedings.neurips.cc/paper/2015/hash/6602294be910b1e3c4571bd98c4d5484-Abstract.html>.
- Krikamol Muandet, Arash Mehrjou, Si Kai Lee, and Anant Raj. Dual instrumental variable regression. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/1c383cd30b7c298ab50293adfecb7b18-Abstract.html>.
- Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. *ArXiv preprint*, abs/1708.08689, 2017. URL <https://arxiv.org/abs/1708.08689>.
- Luis Muñoz-González, Kenneth T. Co, and Emil C. Lupu. Byzantine-robust federated machine learning through adaptive model averaging. *ArXiv preprint*, abs/1909.05125, 2019. URL <https://arxiv.org/abs/1909.05125>.
- Ofir Nachum and Bo Dai. Reinforcement learning via fenchel-rockafellar duality. *ArXiv preprint*, abs/2001.01866, 2020. URL <https://arxiv.org/abs/2001.01866>.
- Angelia Nedic. Distributed gradient methods for convex machine learning problems in networks: Distributed optimization. *IEEE Signal Processing Magazine*, 37(3):92–101, 2020.
- Angelia Nedic and Alex Olshevsky. Stochastic gradient-push for strongly convex functions on time-varying directed graphs. *IEEE Trans. Autom. Control.*, 61(12):3936–3947, 2016. doi: 10.1109/TAC.2016.2529285. URL <https://doi.org/10.1109/TAC.2016.2529285>.
- Angelia Nedić and Asuman Ozdaglar. Convergence rate for consensus with delays. *Journal of Global Optimization*, 47(3):437–456, 2010.

- Angelia Nedic and Asuman E. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control.*, 54(1):48–61, 2009. doi: 10.1109/TAC.2008.2009515. URL <https://doi.org/10.1109/TAC.2008.2009515>.
- Angelia Nedic, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM J. Optim.*, 27(4):2597–2633, 2017. doi: 10.1137/16M1084316. URL <https://doi.org/10.1137/16M1084316>.
- Arvind Neelakantan, Luke Vilnis, Quoc V Le, Lukasz Kaiser, Karol Kurach, Ilya Sutskever, and James Martens. Adding gradient noise improves learning for very deep networks. In *ICLR*, 2016.
- Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takác. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2613–2621. PMLR, 2017. URL <http://proceedings.mlr.press/v70/nguyen17b.html>.
- Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019. URL <https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html>.
- Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *Journal of the ACM (JACM)*, 27(2):228–234, 1980a.
- Marshall C. Pease, Robert E. Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27:228–234, 1980b.
- Jie Peng and Qing Ling. Byzantine-robust decentralized stochastic optimization. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 5935–5939. IEEE, 2020. doi: 10.1109/ICASSP40776.2020.9054377. URL <https://doi.org/10.1109/ICASSP40776.2020.9054377>.

- Radia J. Perlman. An algorithm for distributed computation of a spanningtree in an extended LAN. In William Lidinsky and Bart W. Stuck, editors, *SIGCOMM '85, Proceedings of the Ninth Symposium on Data Communications, British Columbia, Canada, September 10-12, 1985*, pages 44–53. ACM, 1985. doi: 10.1145/319056.319004. URL <https://doi.org/10.1145/319056.319004>.
- Krishna Pillutla, Sham M. Kakade, and Zaid Harchaoui. Robust Aggregation for Federated Learning. *ArXiv preprint*, abs/1912.13445, 2019. URL <https://arxiv.org/abs/1912.13445>.
- Ouri Poupko, Gal Shahaf, Ehud Shapiro, and Nimrod Talmon. Building a sybil-resilient digital community utilizing trust-graph connectivity. *IEEE/ACM Transactions on Networking*, 2021.
- Shi Pu and Angelia Nedic. Distributed stochastic gradient tracking methods. *ArXiv preprint*, abs/1805.11454, 2018. URL <https://arxiv.org/abs/1805.11454>.
- Shi Pu, Wei Shi, Jinming Xu, and Angelia Nedic. Push-pull gradient methods for distributed optimization in networks. *IEEE Trans. Autom. Control.*, 66(1):1–16, 2021. doi: 10.1109/TAC.2020.2972824. URL <https://doi.org/10.1109/TAC.2020.2972824>.
- Qi Qi, Youzhi Luo, Zhao Xu, Shuiwang Ji, and Tianbao Yang. Stochastic optimization of areas under precision-recall curves with provable convergence. In Marc’Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 1752–1765, 2021a. URL <https://proceedings.neurips.cc/paper/2021/hash/0dd1bc593a91620daecf7723d2235624-Abstract.html>.
- Qi Qi, Youzhi Luo, Zhao Xu, Shuiwang Ji, and Tianbao Yang. Stochastic optimization of areas under precision-recall curves with provable convergence. *Advances in Neural Information Processing Systems*, 34:1752–1765, 2021b.
- Shashank Rajput, Hongyi Wang, Zachary B. Charles, and Dimitris S. Papailiopoulos. DETOX: A redundancy-based framework for faster and more robust gradient aggregation. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 10320–10330, 2019. URL <https://proceedings.neurips.cc/paper/2019/hash/415185ea244ea2b2bedeb0449b926802-Abstract.html>.
- Daniel Ramage and Stefano Mazzocchi. Federated analytics: Collaborative data science without data collection. <https://ai.googleblog.com/2020/05/federated-analytics-collaborative-data.html>, 2020.

- Sashank J. Reddi, Ahmed Hefny, Suvrit Sra, Barnabás Póczos, and Alexander J. Smola. Stochastic variance reduction for nonconvex optimization. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 314–323. JMLR.org, 2016a. URL <http://proceedings.mlr.press/v48/reddi16.html>.
- Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Fast incremental method for nonconvex optimization. *ArXiv preprint*, abs/1603.06159, 2016b. URL <https://arxiv.org/abs/1603.06159>.
- Jayanth Regatti, Hao Chen, and Abhishek Gupta. Bygars: Byzantine sgd with arbitrary number of attackers, 2020.
- M. Sadegh Riazi, Mohammad Samragh, Hao Chen, Kim Laine, Kristin Lauter, and Farinaz Koushanfar. Xonn: Xnor-based oblivious deep neural network inference. *ArXiv preprint*, abs/1902.07342, 2019. URL <https://arxiv.org/abs/1902.07342>.
- Bitan Darvish Rouhani, M. Sadegh Riazi, and Farinaz Koushanfar. Deepsecure: Scalable provably-secure deep learning. *ArXiv preprint*, abs/1705.08963, 2017. URL <https://arxiv.org/abs/1705.08963>.
- Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*. John Wiley & sons, 2005.
- Theo Ryffel, Edouard Dufour-Sans, Romain Gay, Francis Bach, and David Pointcheval. Partially encrypted machine learning using functional encryption. *ArXiv preprint*, abs/1905.10214, 2019. URL <https://arxiv.org/abs/1905.10214>.
- Peter Sanders, Jochen Speck, and Jesper Larsson Träff. Two-tree algorithms for full bandwidth broadcast, reduction and scan. *Parallel Comput.*, 35(12):581–594, 2009. doi: 10.1016/j.parco.2009.09.001. URL <https://doi.org/10.1016/j.parco.2009.09.001>.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv preprint*, abs/1910.01108, 2019. URL <https://arxiv.org/abs/1910.01108>.
- Felix Sattler, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. On the byzantine robustness of clustered federated learning. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 8861–8865. IEEE, 2020. doi: 10.1109/ICASSP40776.2020.9054676. URL <https://doi.org/10.1109/ICASSP40776.2020.9054676>.

- Mark Schmidt and Nicolas Le Roux. Fast convergence of stochastic gradient descent under a strong growth condition. *ArXiv preprint*, abs/1308.6370, 2013. URL <https://arxiv.org/abs/1308.6370>.
- Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1):83–112, 2017.
- Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321. ACM, 2015.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL <http://arxiv.org/abs/1409.1556>.
- Nigel P. Smart and Titouan Tanguy. TaaS: Commodity MPC via Triples-as-a-Service. In *CCSW’19 - Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop*, CCSW’19, page 105–116, 2019. doi: 10.1145/3338466.3358918. URL <https://doi.org/10.1145/3338466.3358918>.
- Jy-yong Sohn, Dong-Jun Han, Beongjun Choi, and Jaekyun Moon. Election coding for distributed learning: Protecting signsgd against byzantine attacks. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/a7f0d2b95c60161b3f3c82f764b1d1c9-Abstract.html>.
- Sebastian U. Stich. Unified optimal analysis of the (stochastic) gradient method. *ArXiv preprint*, abs/1907.04232, 2019. URL <https://arxiv.org/abs/1907.04232>.
- Lili Su and Nitin Vaidya. Multi-agent optimization in the presence of byzantine adversaries: Fundamental limits. In *2016 American Control Conference (ACC)*, pages 7183–7188. IEEE, 2016a.
- Lili Su and Nitin H Vaidya. Robust multi-agent optimization: coping with byzantine agents with input redundancy. In *International Symposium on Stabilization, Safety, and Security of Distributed Systems*, pages 368–382. Springer, 2016b.
- Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv 1911.07963*, 2019.
- Shreyas Sundaram and Bahman Ghahsifard. Distributed optimization under adversarial nodes. *IEEE Transactions on Automatic Control*, 64(3):1063–1076, 2018.

- Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D²: Decentralized training over decentralized data. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4855–4863. PMLR, 2018. URL <http://proceedings.mlr.press/v80/tang18a.html>.
- Konstantinos I. Tsianos and Michael G. Rabbat. Distributed consensus and optimization under communication delays. In *49th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2011, Allerton Park & Retreat Center, Monticello, IL, USA, 28-30 September, 2011*, pages 974–982. IEEE, 2011. doi: 10.1109/Allerton.2011.6120272. URL <https://doi.org/10.1109/Allerton.2011.6120272>.
- Konstantinos I. Tsianos, Sean F. Lawlor, and Michael G. Rabbat. Push-sum distributed dual averaging for convex optimization. In *Proceedings of the 51th IEEE Conference on Decision and Control, CDC 2012, December 10-13, 2012, Maui, HI, USA*, pages 5453–5458. IEEE, 2012. doi: 10.1109/CDC.2012.6426375. URL <https://doi.org/10.1109/CDC.2012.6426375>.
- John Tukey. Bias and confidence in not quite large samples. *Ann. Math. Statist.*, 29:614, 1958.
- Sharan Vaswani, Francis R. Bach, and Mark Schmidt. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 1195–1204. PMLR, 2019a. URL <http://proceedings.mlr.press/v89/vaswani19a.html>.
- Sharan Vaswani, Aaron Mishkin, Issam H. Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien. Painless stochastic gradient: Interpolation, line-search, and convergence rates. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3727–3740, 2019b. URL <https://proceedings.neurips.cc/paper/2019/hash/2557911c1bf75c2b643afb4ecbfc8ec2-Abstract.html>.
- Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. Practical low-rank communication compression in decentralized deep learning. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/a376802c0811f1b9088828288eb0d3f0-Abstract.html>.

- Thijs Vogels, Lie He, Anastasia Koloskova, Tao Lin, Sai Praneeth Karimireddy, Sebastian U. Stich, and Martin Jaggi. Relaysun for decentralized deep learning on heterogeneous data, 2021.
- Bokun Wang and Tianbao Yang. Finite-sum coupled compositional stochastic optimization: Theory and applications. In *International Conference on Machine Learning*, pages 23292–23317. PMLR, 2022.
- Guanghui Wang, Ming Yang, Lijun Zhang, and Tianbao Yang. Momentum accelerates the convergence of stochastic AUPRC maximization. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pages 3753–3771. PMLR, 2022a. URL <https://proceedings.mlr.press/v151/wang22b.html>.
- Guanghui Wang, Ming Yang, Lijun Zhang, and Tianbao Yang. Momentum accelerates the convergence of stochastic auprc maximization. In *International Conference on Artificial Intelligence and Statistics*, pages 3753–3771. PMLR, 2022b.
- Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris S. Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/b8ffa41d4e492f0fad2f13e29e1762eb-Abstract.html>.
- Mengdi Wang, Ji Liu, and Ethan X. Fang. Accelerating stochastic composition optimization. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1714–1722, 2016. URL <https://proceedings.neurips.cc/paper/2016/hash/92262bf907af914b95a0fc33c3f33bf6-Abstract.html>.
- Mengdi Wang, Ethan X Fang, and Han Liu. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161:419–449, 2017.
- Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. Spiderboost and momentum: Faster variance reduction algorithms. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC*,

- Canada, pages 2403–2413, 2019. URL <https://proceedings.neurips.cc/paper/2019/hash/512c5cad6c37edb98ae91c8a76c3a291-Abstract.html>.
- Duncan J Watts and Steven H Strogatz. Collective dynamics of ‘small-world’ networks. *nature*, 393(6684):440–442, 1998.
- Christopher Stroude Withers. Bias reduction by taylor series. *Communications in Statistics-Theory and Methods*, 16(8):2369–2383, 1987.
- Chenguang Xi and Usman A. Khan. DEXTRA: A fast algorithm for optimization over directed graphs. *IEEE Trans. Automat. Contr.*, 62(10):4980–4993, 2017. doi: 10.1109/TAC.2017.2672698. URL <https://doi.org/10.1109/TAC.2017.2672698>.
- Chenguang Xi, Van Sy Mai, Ran Xin, Eyad H. Abed, and Usman A. Khan. Linear convergence in optimization over directed graphs with row-stochastic matrices. *IEEE Trans. Autom. Control.*, 63(10):3558–3565, 2018. doi: 10.1109/TAC.2018.2797164. URL <https://doi.org/10.1109/TAC.2018.2797164>.
- Lin Xiao and Stephen P. Boyd. Fast linear iterations for distributed averaging. *Syst. Control. Lett.*, 53(1):65–78, 2004. doi: 10.1016/j.sysconle.2004.02.022. URL <https://doi.org/10.1016/j.sysconle.2004.02.022>.
- Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Generalized byzantine-tolerant sgd. *ArXiv preprint*, abs/1802.10116, 2018a. URL <https://arxiv.org/abs/1802.10116>.
- Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Phocas: dimensional byzantine-resilient stochastic gradient descent. *ArXiv preprint*, abs/1805.09682, 2018b. URL <https://arxiv.org/abs/1805.09682>.
- Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant SGD by inner product manipulation. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 261–270. AUAI Press, 2019a. URL <http://proceedings.mlr.press/v115/xie20a.html>.
- Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant SGD by inner product manipulation. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 261–270. AUAI Press, 2019b. URL <http://proceedings.mlr.press/v115/xie20a.html>.
- Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June*

- 2019, Long Beach, California, USA, volume 97 of *Proceedings of Machine Learning Research*, pages 6893–6901. PMLR, 2019c. URL <http://proceedings.mlr.press/v97/xie19b.html>.
- Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno++: Robust fully asynchronous SGD. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10495–10503. PMLR, 2020a. URL <http://proceedings.mlr.press/v119/xie20c.html>.
- Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno++: Robust fully asynchronous SGD. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10495–10503. PMLR, 2020b. URL <http://proceedings.mlr.press/v119/xie20c.html>.
- Ran Xin and Usman A. Khan. A linear algorithm for optimization over directed graphs with geometric convergence. *IEEE Control. Syst. Lett.*, 2(3):315–320, 2018. doi: 10.1109/LCSYS.2018.2834316. URL <https://doi.org/10.1109/LCSYS.2018.2834316>.
- Ran Xin and Usman A. Khan. Distributed heavy-ball: A generalization and acceleration of first-order methods with gradient tracking. *IEEE Trans. Autom. Control.*, 65(6):2627–2633, 2020. doi: 10.1109/TAC.2019.2942513. URL <https://doi.org/10.1109/TAC.2019.2942513>.
- Ran Xin, Chenguang Xi, and Usman A. Khan. FROST - fast row-stochastic optimization with uncoordinated step-sizes. *EURASIP J. Adv. Signal Process.*, 2019:1, 2019. doi: 10.1186/s13634-018-0596-y. URL <https://doi.org/10.1186/s13634-018-0596-y>.
- Yi-Rui Yang and Wu-Jun Li. BASGD: buffered asynchronous SGD for byzantine learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 11751–11761. PMLR, 2021a. URL <http://proceedings.mlr.press/v139/yang21e.html>.
- Yi-Rui Yang and Wu-Jun Li. BASGD: buffered asynchronous SGD for byzantine learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 11751–11761. PMLR, 2021b. URL <http://proceedings.mlr.press/v139/yang21e.html>.
- Zhixiong Yang and Waheed U Bajwa. Bridge: Byzantine-resilient decentralized gradient descent. *arXiv 1908.08098*, 2019a.
- Zhixiong Yang and Waheed U Bajwa. Byrdie: Byzantine-resilient distributed coordinate descent for decentralized learning. *IEEE Transactions on Signal and Information Processing over Networks*, 2019b.

- Andrew C Yao. Theory and application of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, pages 80–91. IEEE, 1982.
- Yu M Yermol'yev. A general stochastic programming problem. 1971.
- Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5636–5645. PMLR, 2018a. URL <http://proceedings.mlr.press/v80/yin18a.html>.
- Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5636–5645. PMLR, 2018b. URL <http://proceedings.mlr.press/v80/yin18a.html>.
- Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett. Defending against saddle point attack in byzantine-robust distributed learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7074–7084. PMLR, 2019. URL <http://proceedings.mlr.press/v97/yin19a.html>.
- Bicheng Ying, Kun Yuan, Yiming Chen, Hanbin Hu, Pan Pan, and Wotao Yin. Exponential graph is provably efficient for decentralized deep training. *Advances in Neural Information Processing Systems*, 34, 2021a.
- Bicheng Ying, Kun Yuan, Hanbin Hu, Yiming Chen, and Wotao Yin. Bluefog: Make decentralized algorithms practical for optimization and deep learning. *ArXiv preprint*, abs/2111.04287, 2021b. URL <https://arxiv.org/abs/2111.04287>.
- Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 5693–5700. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33015693. URL <https://doi.org/10.1609/aaai.v33i01.33015693>.

- Kun Yuan, Bicheng Ying, Xiaochuan Zhao, and Ali H. Sayed. Exact diffusion for distributed optimization and learning - part I: algorithm development. *IEEE Trans. Signal Process.*, 67(3):708–723, 2019. doi: 10.1109/TSP.2018.2875898. URL <https://doi.org/10.1109/TSP.2018.2875898>.
- Kun Yuan, Yiming Chen, Xinmeng Huang, Yingya Zhang, Pan Pan, Yinghui Xu, and Wotao Yin. Decentlam: Decentralized momentum SGD for large-batch deep training. *ArXiv preprint*, abs/2104.11981, 2021. URL <https://arxiv.org/abs/2104.11981>.
- Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan H. Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7252–7261. PMLR, 2019. URL <http://proceedings.mlr.press/v97/yurochkin19a.html>.
- Jiaqi Zhang and Keyou You. Decentralized stochastic gradient tracking for non-convex empirical risk minimization, 2020.
- Junyu Zhang and Lin Xiao. Multilevel composite stochastic optimization via nested variance reduction. *SIAM Journal on Optimization*, 31(2):1131–1157, 2021.
- Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657, 2015. URL <https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html>.
- Zhaorong Zhang, Kan Xie, Qianqian Cai, and Minyue Fu. A bp-like distributed algorithm for weighted average consensus. In *12th Asian Control Conference, ASCC 2019, Kitakyushu-shi, Japan, June 9-12, 2019*, pages 728–733. IEEE, 2019. URL <https://ieeexplore.ieee.org/document/8765066>.
- Chengcheng Zhao, Jianping He, and Qing-Guo Wang. Resilient distributed optimization algorithm against adversarial attacks. *IEEE Transactions on Automatic Control*, 65(10):4308–4315, 2019.
- Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC*,

Canada, pages 14747–14756, 2019. URL <https://proceedings.neurips.cc/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html>.

Lie He | Curriculum Vitae

INJ 335, EPFL – Ecublens 1024, Switzerland

✉ lie.he@epfl.ch • 🌐 GitHub • 🎓 Google Scholar

Education

École Polytechnique Fédérale de Lausanne (EPFL) <i>Ph.D. in Computer Science</i> Thesis: <i>Distributed Optimization with Byzantine Robustness Guarantees</i> . Advisor: Prof. Martin Jaggi.	Lausanne, Switzerland 2019–2023
École Polytechnique Fédérale de Lausanne (EPFL) <i>MSc in Computational Science and Engineering</i> Thesis: <i>COLA: Decentralized Linear Learning</i> . Advisor: Prof. Martin Jaggi.	Lausanne, Switzerland 2015–2018
University of Science and Technology of China (USTC) <i>BSc in Mathematics</i> Thesis: <i>Numerical Fluxes of Finite Volumes Method for Euler Equations</i> . Advisor: Prof. Yinhua Xia.	Hefei, China 2011–2015

Work Experience

Amazon Inc. <i>Applied Scientist Intern</i> <ul style="list-style-type: none">Developed a novel technique to identify and mitigate biases in optimization algorithms commonly used in machine learning, achieving orders-of-magnitude improvement in sample complexity.Paper accepted for presentation at NeurIPS 2023.	Tübingen, Germany June–October 2022
Google Inc. <i>Research Intern</i> <ul style="list-style-type: none">Engineered multi-organizational federated learning algorithms for iNaturalist datasets with hierarchical structure.Partnered with cross-disciplinary teams to incorporate research findings into broader organizational research agendas.	New York, USA April–July 2019
Machine Learning and Optimization Lab at EPFL <i>Software Engineer Intern</i> <ul style="list-style-type: none">Developed an open-source project MLBench from scratch which offers a benchmark suite for distributed machine learning algorithms.Implemented and benchmarked popular distributed training algorithms for deep learning.	Lausanne, Switzerland Jul–Dec 2018

Honors, Awards and Fundings

- 2022: Google Research Collab Program** awarded by Google to fund research student
- 2019: EDIC Fellowship** awarded by EPFL to selected PhD students
- 2015: Outstanding Undergraduate Scholarships** awarded by USTC
- 2014: Exchange Student Scholarship** awarded by HKUST for summer exchange program

Academic Services

Conference reviewer:

- International Conference on Machine Learning (ICML): '23, '22, '21
- Conference on Neural Information Processing Systems (NeurIPS): '22, '21
- International Conference on Learning Representations (ICLR): '22, '21

Journal reviewer:

- Journal of Machine Learning Research (JMLR)
- Transactions on Machine Learning Research (TMLR)
- IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)

Open Source Projects

MLBench: A framework for benchmarking distributed machine learning algorithms

DecentralizedAI: A cross-platform framework for collaborative and privacy-preserving training of machine learning models

Selected Publications

Note: * indicates that the authors with equal contributions.

Peer-reviewed conference and journal publications.....

1. **Towards Provably Personalized Federated Learning via Threshold-Clustering of Similar Clients.**
Mariel Werner, Lie He, Sai Praneeth Karimireddy, Michael Jordan, Martin Jaggi
TMLR 2023 and a shorter version accepted at NeurIPS 2022 FL Workshop.
2. **Debiasing Conditional Stochastic Optimization.**
Lie He, Shiva Kasiviswanathan
NeurIPS 2023.
3. **Byzantine-Robust Learning on Heterogeneous Datasets via Bucketing.**
Sai Praneeth Karimireddy, Lie He*, and Martin Jaggi.*
ICLR 2022 **Spotlight** and a shorter version accepted at NeurIPS 2020 SpicyFL workshop.
4. **Relaysum for Decentralized Deep Learning on Heterogeneous Data.**
Thijs Vogels, Lie He*, Koloskova Anastasia, Sai Praneeth Karimireddy, Tao Lin, Sebastian Stich, and Martin Jaggi.*
NeurIPS 2021.
5. **Learning from History for Byzantine Robust Optimization.**
Sai Praneeth Karimireddy, Lie He, and Martin Jaggi.
ICML 2021.
6. **COLA: Decentralized Linear Learning.**
Lie He, An Bian*, and Martin Jaggi.*
NeurIPS 2018.

Peer-reviewed workshop papers.....

1. **Secure Byzantine-Robust Machine Learning.**
Lie He, Sai Praneeth Karimireddy, and Martin Jaggi.
NeurIPS 2020 SpicyFL Workshop.

Preprints.....

1. **Byzantine-Robust Decentralized Learning via ClippedGossip.**
Lie He, Sai Praneeth Karimireddy*, Martin Jaggi*
Arxiv.