INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2026IPPAX049

Thèse de doctorat

Contributions on Robust Distributed Learning

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à l'École polytechnique

École doctorale n°574 École doctorale de mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques Appliquées

Thèse présentée et soutenue à l'Institut Henri Poincaré à Paris, le 3 septembre 2026, par

RENAUD GAUCHER

Composition du Jury :

Jamal Atif Professeur, École polytechnique	Président
Po-Ling, Loh Professeure, Cambridge University	Rapporteur
Mikael, Johansson Professeur, KTH - Royal Institute of Technology	Rapporteur
El Mahdi El Mhamdi Professeur assistant, École polytechnique	Examinateur
Sonia Ben Mokhtar Directrice de recherche, CNRS	Examinateur
Hadrien Hendrikx Chargé de recherche, INRIA	Co-directeur de thèse
Aymeric Dieuleveut Professeur, École polytechnique	Directeur de thèse
Lydia Zakynthinou Professeure assistant, Johns Hopkins University	Invitée

Abstract

Modern machine learning is fed by data, which is usually generated by individuals and legal entities, or, more precisely, by their devices: smartphones, computers, IoT devices and other smart sensors. While sending all the data to a central server on which the model is trained can be simpler, ethical concerns and legal requirements have pushed for the development of distributed learning algorithms: keeping the data stored locally, and training the model in a distributed manner, is a first step towards ensuring the sovereignty of individuals over their data.

However, involving multiple devices in a distributed learning process opens up two new threats: some devices could be buggy, crash, or be controlled by hostile parties. The notion of Byzantine failure refers to such worst-case behavior. When no mitigation strategy is implemented, Byzantine devices can disrupt the optimization process of the machine learning model. When the communication is structured by a federating server that centralizes the learning process, simple mitigations such as a robust average of the information sent by the clients can be implemented. In contrast, when the devices communicate in a peer-to-peer – or decentralized – manner, naively implementing those mitigations yields overly biased algorithms.

The first set of contributions of this thesis focuses on proposing decentralized optimization algorithms robust to Byzantine devices. An algorithm is proposed to communicate robustly by adapting the gossip communication model, in which each client update consists of approximate averages of its neighbors' information. The robustness condition of this algorithm is analyzed, and proven to be tight up to a numerical factor. This algorithm is then used for decentralized optimization, and a specific attack for the setting is proposed. Then, we analyze those algorithms through the lens of the dual approach for decentralized optimization, which allows us to re-interpret the averaging methods, and understand the potential of the dual approach.

The second part of this thesis studies the communication complexity in the federating server setting in the presence of Byzantine devices. We propose an abstraction of the Byzantine optimization problem, which allows us to design two algorithms: the first one is a similarity-based algorithm that leverages a cheaply accessible proxy loss to speed up the optimization, and the second is an accelerated algorithm.

Distributing the optimization and keeping the data stored locally is insufficient in itself to guarantee the confidentiality of individuals' data. Ensuring this confidentiality requires relying on differential privacy. Yet, training machine learning algorithms with current differentially private optimization methods induces a significant accuracy overhead, as noise is added to obfuscate the potentially sensitive information. If using noise is necessary, some

of this overhead could be alleviated by carefully choosing the noise.

This motivates the last part of this thesis, in which we study the feasibility of private mean estimation of an anisotropic Gaussian distribution with a mild dependence on the dimension. We show that, when the covariance matrix is unknown and the number of samples is significantly smaller than the dimension, this is not possible, as the privacy overhead in the accuracy necessarily scales as a power of the dimension.

Résumé

L'apprentissage automatique moderne est nourri par les données, qui sont générées par des individus et des entités légales, ou, plus précisément, par leurs appareils électroniques: téléphones, ordinateurs, appareils IoT ou autres capteurs intelligents. S'il est plus simple d'envoyer toutes les données à un serveur sur lequel on centralise l'entraînement du modèle d'apprentissage, cela implique une perte de souveraineté des utilisateurs sur leurs données. Ainsi, des contraintes éthiques et légales ont conduit au développement d'algorithmes d'apprentissage distribués. En effet, stocker ses données localement, et entraîner les modèles de manière distribuée, est une première étape pour garantir la souveraineté des individus sur leurs données.

Cependant, effectuer l'apprentissage de manière distribuée en multipliant les appareils impliqués ouvre la voie à de nouvelles menaces: certains appareils peuvent avoir des bugs, être piratés ou contrôlés par des entités antagonistes. La notion de *défaillance Byzantine* modélise ce type de comportement hostile. Lorsqu'aucune défense n'est implémentée, des appareils Byzantins peuvent empêcher l'optimisation du modèle d'apprentissage machine. Lorsque la communication est structurée par un serveur fédérateur qui centralise les communications, celui-ci peut utiliser des défenses simples, comme des moyennes robustes des informations reçues, pour contrer l'impact de clients malveillants. Au contraire, lorsque les appareils communiquent de pair à pair, c'est-à-dire de manière décentralisée, l'implémentation naïve de ces moyennes robustes conduit à des algorithmes inutilement biaisés.

Le premier groupe de contributions de cette thèse étudie des algorithmes de communication décentralisés robustes aux attaques Byzantines. Un algorithme est proposé pour communiquer de manière robuste en adaptant le modèle de communication *gossip*, dans lequel chaque ordinateur actualise son paramètre en effectuant une combinaison linéaire des paramètres de ses voisins. La condition de robustesse de cet algorithme est analysée, et il est prouvé qu'elle est optimale à un facteur numérique près. Cet algorithme est ensuite utilisé en optimisation décentralisée, et une attaque spécifique au contexte décentralisé est proposée. Ensuite, nous analysons ces algorithmes de communication gossip par le biais de *l'approche duale* pour l'optimisation décentralisée, ce qui nous permet de les réinterpréter, et de comprendre le potentiel de l'approche duale dans le cadre Byzantin.

Dans la deuxième partie de cette thèse, nous étudions la complexité communicationnelle de l'optimisation robuste aux ordinateurs Byzantins dans le cadre du serveur fédérateur. Nous proposons une abstraction du problème d'optimisation Byzantine, ce qui nous permet de proposer deux algorithmes: le premier exploite la notion de similarité entre la fonction objective globale et un proxy de celle-ci facilement accessible, et le second est un algorithme

accéléré. Ces deux méthodes nous permettent d'améliorer très largement la complexité communicationnelle comparativement aux méthodes préexistantes.

Distribuer l'optimisation et garder les données stockées localement n'est pas suffisant, en soi, pour garantir la confidentialité de ces données. Pour cela, il est nécessaire d'entraîner les modèles de manière différentiellement confidentiels, ce qui induit une dégradation significative de leurs performances. Cela découle du bruit ajouté durant l'entraînement. Si l'utilisation de bruit est nécessaire à la garantie de confidentialité, une partie de la dégradation pourrait être évitée en utilisant un bruit optimal.

Cette remarque motive la dernière partie de cette thèse, dans laquelle nous étudions la faisabilité de l'estimation différentiellement confidentielle de la moyenne de vecteurs Gaussiens anisotropes avec une dégradation faible de la précision avec la dimension ambiante. L'étude de deux régimes de distributions a priori de la matrice de covariance: un régime où la matrice de covariance est une projection aléatoiren et un régime où la matrice de covariance est diagonale mais avec un spectre à queue de distribution lourde, nous apportons une réponse négative à cette question.

Remerciements

Écrire ses remerciements, c'est, il me semble, reconnaitre que la thèse que l'on vient d'écrire est quelque chose de sérieux, qui possède une ampleur - ou témoigne d'une expérience - justifiant l'écriture de ces pages de remerciements. Ou peut-être qu'il s'agit uniquement d'un occasion privilégiée pour reconnaitre par écrit que ce que l'on est et ce que l'on fait provient très largement de ce que notre entourage nous apporte. Étant très mal à l'aise avec la première hypothèse, je me raccroche volontiers à la seconde.

Mes premiers emerciements vont bien sûrs à mes directeurs de thèses, Aymeric et Hadrien, pour ces trois années passées. Merci pour m'avoir initié à la recherche, pour votre humanisme et tous les moments informels passés ensembles, la méta science les bières et toutes les discussions. Merci pour m'avoir donné les moyens de profiter pleinement de la vie de thèsard. Merci à toi Aymeric pour ta bonne humeur et ton enthousiasme permanent. Merci à toi Hadrien pour ta bienveillance, ta disponibilité sans faille et l'exemple que tu donnes par ton équilibre de vie. Et désolé pour ma facheuse tendance à tomber malade durant les périodes de rush.

I would also like to thank the members of my jury. First, the reviewers, Mikael and Po-Ling, for taking the time to review my manuscript and for their valuable and insightful feedback. I am also very grateful to Mahdi, Sonia and Aurelien for accepting to be part of the jury. It is truly an honor.

I would like to thank all the people who welcomed me in Berkeley - Mike team's and the others. Thank you Lydia for accepting the gamble of welcoming a student for a short visit on a new reasearch field. Thank you also Adam, Ephraim and Jonathan who warmly introduced me to DP, and, more importantly, iteratively broke all the predjudices I had on American's researchers. Thank you a lot for that, and for showing me an alternative way of doing research.

Aurélien, Rafael and Nirupam, I can't wait to work with you on exciting topics! Many thanks for this amazing opportunity.

Je voudrais ensuite remercier Christophe, pour avoir accueilli et aiguillé l'étudiant perdu que j'étais en 3A puis en 4A et qui te demandais naivement de lui trouver un avenir. Au dela de toi, merci aux instituteurs, enseignants et professeurs qui m'ont tant apris durant mes périodes de veille en classe.

J'aurais très certainement arrêté ma thèse si j'étais resté seul dans ma chambre sans profiter de votre présence (presque) au quotidien. Merci à tous les doctorants du cmap pour avoir été là. L'équipe Sympas, même si, pour certains, on se voyait quand même surtout en conf: Louis, Maxence, Margaux, Baptiste, Mahmoud, Pierre, Christophe, Louis,

Daniel, JB et les derniers arrivés Abel, Lucas et Andrei. Ceux arrivés en même temps que moi, et que j'abandonne à leur fin de thèse: Meggie, Alexandre, Loic, Luce et Adrien. Bon courage à vous. Et merci à tous les autres, Valentin, Martin, Solange, Manon, Antoine, et tous ceux que j'ai oublié, ou dont j'ai oublié le nom alors qu'on se parle régulièrement depuis trois ans.

Merci également aux permanents du cmap pour leur accueil - Mahdi, tu ne le sais peut être pas, mais c'est en regardant un vidéo de toi avec Lê que je me suis motivé à candidater à la thèse. Gauthier, Marie-Lou, et Luis, merci d'avoir été là. Merci à Adrien Taylor pour tes conseils et ton humanité.

Merci à l'équipe Thoth de l'inria grenoble, pour m'avoir aussi facilement et aussi facilement intégré alors que je n'ai passé que 7 semaines avec vous durant ces années. C'était toujours un plaisir de passer. Merci pour cela.

L'ambition des remerciements de thèse est parfois d'établir une généalogie de la thèse. Ne souhaitant pas faire un afront à ce que je crois être les enseignements élémentaires de la sociologie, je remercie ma famille pour m'avoir transmis le capital culturel m'ayant ouvert les portes de la réussite académique. Votre présence, votre amour et votre soutien me semblent si évidents qu'il me paraît presque grotesque de vous en remercier. Merci infinement pour cela.

Merci à mes amis d'avoir été là durant ces deux dernières années, d'accepter l'absence et d'être là même dans les moments difficiles. Julien, Joris, et tous les autres, merci.

Chose promite, choses due, merci au bl19 pour l'aide apportée à la diagonalisation d'une matrice (cf. Lemma 4.D.4).

Enfin, merci d'être là, pour tout. La vie est plus belle à tes côtés.

Contents

I Introduction **1**

1 Introduction **5**
1.1 Context . . . 5
1.2 Scope of the Thesis . . . 5
1.3 Outline of the Thesis . . . 7

2 Technical Background **9**
2.1 Supervised Learning . . . 9
2.2 Introduction to First Order Optimization . . . 11
2.3 Distributed Learning . . . 18
2.4 Adversarial Robustness . . . 25
2.5 Privacy . . . 32

3 Technical Summary of the Contributions **39**

II Byzantine Resilience in Decentralized Optimization **45**

4 A Unified Breakdown Analysis for Byzantine Robust Gossip **47**
4.1 Introduction . . . 49
4.2 Background . . . 50
4.3 The Robust Gossip Framework . . . 53
4.4 From the General Framework to Practical Decentralized Algorithms . . . 57
4.5 Attacking Robust Gossip Algorithms . . . 60
4.6 Experimental Evaluation . . . 62
4.7 Conclusion . . . 62

4.A Additional Discussion . . . 65
4.B Experiments . . . 67
4.C Analysis of the Robust Gossip Method . . . 72
4.D Proof of Theorem 4.3.5 - Upper Bound on the Breakdown Point . . . 77
4.E (b, ρ)-Robust Summation Rules . . . 83
4.F Proofs for D-SGD . . . 86

5 Byzantine-Robust Gossip: Insights from a Dual Approach **89**
5.1 Introduction . . . 91
5.2 Deriving EdgeClippedGossip . . . 92
5.3 Global Clipping . . . 96
5.4 Beyond Global Coordination and Averaging . . . 98

5.5 Experiments . . . 100
5.6 Conclusion . . . 101

5.A Experimental design . . . 103
5.B Proofs . . . 103

III Byzantine Resilience in Federated Optimization **113**

6 From Inexact Gradients to Byzantine Robustness: Acceleration and Optimization under Similarity **115**
6.1 Introduction . . . 117
6.2 Setting and Reduction . . . 119
6.3 Gradient Descent under Similarity . . . 123
6.4 An Accelerated Inexact Extra-gradient Method . . . 126
6.5 Experiments . . . 128
6.6 Conclusion . . . 129

6.A Related Concurrent Work Shi et al. (2025) . . . 131
6.B Robust Aggregation Rules . . . 131
6.C From Hessian Similarity to Bounded Heterogeneity . . . 136
6.D Convergence of Algorithm 4 and Algorithm 5 . . . 140
6.E From Devolder et al. (2013) acceleration to acceleration for $(\zeta^2, \alpha = 0)$-inexact oracles . . . 150
6.F Additional Experiments . . . 152
6.G Additional Discussion . . . 154

IV Privacy **157**

7 The Curse of Dimensionality in Privacy: High-dimensional Mean Estimation with Unknown Covariance **159**
7.1 Introduction . . . 161
7.2 Preliminaries . . . 164
7.3 The Non-Diagonal Case . . . 165
7.4 The Diagonal Case . . . 173

7.A Concentration Results . . . 187

V Conclusion and Perspectives **191**

Bibliography **195**

Part I

Introduction

Contents

1 Introduction **5**
1.1 Context . . . 5
1.2 Scope of the Thesis . . . 5
1.3 Outline of the Thesis . . . 7

2 Technical Background **9**
2.1 Supervised Learning . . . 9
2.2 Introduction to First Order Optimization . . . 11
2.2.1 Convergence of Gradient Descent . . . 12
2.2.2 Faster Algorithms . . . 15
2.2.3 Stochastic Gradient Descent . . . 16
2.3 Distributed Learning . . . 18
2.3.1 Introduction . . . 18
2.3.2 Federated Learning . . . 19
2.3.3 Decentralized Learning . . . 20
2.4 Adversarial Robustness . . . 25
2.4.1 Historical Perspectives . . . 26
2.4.2 The Machine Learning Setting . . . 27
2.4.3 Robust Algorithms . . . 28
2.4.4 Links with the Robust Statistics Literature . . . 31
2.5 Privacy . . . 32
2.5.1 Why Differential Privacy? . . . 32
2.5.2 Formal Guarantee . . . 33
2.5.3 Differentially Private Algorithms . . . 35

3 Technical Summary of the Contributions **39**

Chapter 1

Introduction

1.1 Context

The growth of the digital ecosystem, along with the increase in computational power of machines and the reduction of data storage costs, has led to the creation of massive amounts of data, which are in turn exploited using machine learning models. The use of those models now shape our societies: not only are recommendation models a pillar of the ad-targeting systems on which major companies are built, but they also shape our social interactions and influence the news we read. For instance, the 2026 digital news report of the Reuters Institute (Egan et al., 2026), covering 48 countries, showed that, for the first time, overall social media and video network usage overtook that of TV and news websites. The rise of large language models might accentuate this trend, as evidenced by the enormous investments made in LLMs-related companies and the necessary datacenters.

These technological changes inevitably come with societal challenges: for instance, the design and control of recommendation systems is becoming a democratic challenge, and overall AI development may disrupt the job market and further amplify inequalities (Korinek and Stiglitz, 2018). Yet, systematically understanding the (potentially undesirable) consequences of machine learning, as well as proposing efficient mitigations, goes far beyond the scope of this dissertation. Instead, we adopt the perspective of the European Commission's 2020 white paper on excellence and trust in AI (Commission, 2020), and contribute to two of the seven conditions necessary for the implementation of trustworthy AI, focusing on the requirements related to *technical robustness and security* and to *privacy and data governance*.

1.2 Scope of the Thesis

Throughout this thesis, we will consider robustness challenges in distributed learning algorithms: either the robustness against adversarial computers, or robustness against privacy leakage.

Distributed Learning. In Distributed Learning, multiple participants – also referred to as clients, nodes or devices – aim at collaboratively training a machine learning model

together. Each participant holds a database locally, and communicates with the others to optimize the model on the joint database. To achieve this goal, two communication settings are investigated:

- **Federated Learning**(Kairouz and McMahan, 2021) in which a server communicates directly with the clients to centralize the communication. This setting, in a context of distributed learning, is also referred to as the *centralized* setting, in opposition to the decentralized learning setting.

- **Decentralized Learning** in which clients communicate directly with each other, in a peer-to-peer manner, without relying on a central server. In this setting the communication network is modeled as a graph, in which clients are vertices and communicate with their neighbors in the graph through the edges. The communication can be synchronous or asynchronous communication, and the graph may also vary in time. In this thesis however, we will only consider synchronous communication with a fixed communication graph.

In both settings, the client's data may be heterogeneous, physical constraints may apply such as bandwidth constraint and a low-availability of the devices. In particular, we focus on the two following challenges.

Byzantine Failure. It is sometimes hard to ensure that all participants can be trusted: the distributed learning process may involve low-security devices, which could be hacked, have corrupted data, be buggy or controlled by adversarial parties. A common worst-case model is the so called *Byzantine failure* (Lamport et al., 1982) model, which refers to omniscient adversarial parties that send arbitrarily harmful information. One focus of this thesis is to design optimization algorithms that are guaranteed to converge to a neighborhood of the true solution despite such adversaries. In the decentralized setting, we investigate how to design efficient decentralized algorithms, and how many adversaries such a system can tolerate. In the federated setting we will focus on the design of fast and robust optimization algorithms.

Differential Privacy. One of the promises of distributed learning is to guarantee clients' sovereignty over their data, since optimization takes place without ever sending raw data to the central server. However, simply sending gradients or model parameters is usually not enough to guarantee the privacy of client data, as those gradients or parameters can directly reveal the corresponding dataset. Differential privacy (DP) formally guarantees that, to a certain extent, clients can participate in the learning process without revealing their data to unwanted third parties. However, designing distributed algorithms with differential privacy requires adding noise during the learning process: for example, the server adds isotropic Gaussian noise to the average of the clients' information. In practice, this noise can significantly reduce the accuracy of the optimization pipeline. It is therefore crucial to design private methods that add as little noise as possible while maintaining the same security guarantees. In this manuscript, we investigate to what extent differentially

private mean estimators can offer accuracy guarantees that depend tightly on the underlying structure of the problem, without suffering from a large ambient dimension.

1.3 Outline of the Thesis

This manuscript is divided in three main parts.

The rest of this introductory Part I is organized as follow: the remainder of this Chapter 1 provides a quick overview of the outline and the main contributions. Chapter 2 is an introduction to the scientific field studied in this thesis. We quickly present the overarching goal of statistical learning, then present how this goal can be achieved using first order optimization. Then, we present how these methods apply in distributed systems. This leads us to introducing the two focus of this manuscript: the robustness against corrupted (a.k.a. Byzantine) parties, and against privacy leakage. Chapter 3 gives a technical and detailed summary of the contributions.

Part II studies decentralized first-order optimization algorithms robust to Byzantine failures. In Chapter 4 (based on a joint work with Aymeric Dieuleveut and Hadrien Hendrikx) we propose a general method to make the gossip communication robust to adversaries. By quantifying the optimal breakdown condition of gossip algorithm with adversaries, we conclude that our method has an optimal breakdown up to small constant factors. This also allows us to propose a new attack to challenge distributed algorithms. In Chapter 5 (based on a joint work with Aymeric Dieuleveut and Hadrien Hendrikx) we study the dual method to design decentralized algorithms, as it usually offers a principled way to build efficient algorithms. We propose a communication algorithm that guarantee the decrease of the mean-square error, without searching for an asymptotic consensus. We then re-interpret existing robust algorithms using the dual method, and discuss its limitations in the Byzantine context.

Part III studies federated first-order optimization algorithms robust to Byzantine computers. In Chapter 6 (based on a joint work with Aymeric Dieuleveut and Hadrien Hendrikx) we explicit that distributed optimization with adversaries based on robust averaging function can be abstracted into an optimization problem with inexact gradient oracles. We show the strength of this abstraction by proposing both an accelerated distributed optimization method based on an extragradient scheme, and an optimization method under a similarity hypothesis of the hessian on the loss functions.

Part IV consider an other security-related topics, namely the one of differential privacy. In Chapter 7 (based on a joint work with Ephraim Linder, Adam Smith, Jonathan Ullman and Lydia Zakynthinou, initiated during a visit at UC Berkeley), we investigate the feasibility of achieving dimension-independent rates in high-dimensional private mean estimation, when the covariance matrix is of small effective dimension and is unknown to the analyst. We propose two complementary results: in the first one, for a number of samples smaller than the effective dimension of the problem, we show that the optimal error matches the

one of a naive private mean estimation algorithm, with a dimension-dependent error. In the second one, the effective dimension is nearly constant, and as long as the number of sample is a fractional power of the dimension, we show that the number of samples required to reach a given accuracy still scales as a power of the dimension. Those results contrast with previous work, that showed dimension-independent rates using an algorithm which knowns the covariance matrix in advance. *This chapter is still ongoing work, and while most of the technical part is mature, some aspects such as the introduction and the overall clarity require more work.*

Each chapter is self-contained, thus the notations may slightly vary from chapter to chapter.

Chap.	Paper title	Collaborators	Status
4	*A Unified Analysis of Byzantine Robust Gossip*	**R. Gaucher**, A. Dieuleveut, H. Hendrikx.	*ICML 2025.*
5	*Byzantine Robust Gossip: Insights from a Dual Approach*	**R. Gaucher**, A. Dieuleveut, H. Hendrikx.	*TMLR.*
6	*From Inexact Gradient to Byzantine Robustness: Acceleration and Optimization under Similarity*	**R. Gaucher**, A. Dieuleveut, H. Hendrikx.	*Under review.*
7	*The Curse of Dimensionality in Privacy: High-dimensional Mean Estimation with Unknown Covariance*	**R. Gaucher**, E. Linder, A. Smith, J. Ullman and L. Zakynthinou.	*In preparation for submission.*

Table 1.1: Summary of the scientific production

Chapter 2

Technical Background

We start this chapter by introducing the main underlying motivation of this thesis, which is to design methods applied to supervised learning. Section 2.1 aims at motivating the focus on finite-sum optimization problems, which is the main focus of this manuscript. We then introduce basic notions of optimization in Section 2.2, and then delve into the distributed optimization problem in Section 2.3. Section 2.4 and Section 2.5 eventually introduce important problems in distributed optimization, which are respectively studied in Parts II and III and – in a more theoretical way – in Part IV of this manuscript.

2.1 Supervised Learning

We begin by briefly introducing the notion of supervised learning. More information can be found in Mohri et al. (2018); Bach (2024).

Statistical machine learning aims at predicting a label Y given a covariate X. For instance, the covariate could be a picture of a cat or a dog, and we want the learning algorithm to predict its species. The covariate is supposed to provide some information on the label, which can be formalized by assuming that the couple (x, y) forms a random variable of an unknown distribution $\mathcal{D}$. Machine learning aims at automatically designing a measurable function $\mathcal{F}(\mathcal{X}, \mathcal{Y})$ called a *predictor* to predict y given x.

One can distinguish two classes of supervised learning problems:

1. The *regression* case, in which the label space is continuous, for instance $\mathcal{Y} = \mathcal{R}^k$,
2. The *classification* case, in which the label space is discrete, such as $\mathcal{Y} = \{1, \dots, k\}$.

In practice, classification predictors are often derived from a regressor $\hat{h}_r : \mathcal{X} \to \mathbb{R}^k$ using the plug-in approach: $\hat{h}_c(\mathrm{x}) = \arg\max_{i \in 1,\dots,k} [\hat{h}_r(\mathrm{x})]_i$.

Quality of the prediction. The quality of the prediction is measured by a loss function $\ell : (\mathcal{X} \times \mathcal{Y}) \times \mathcal{F}(\mathcal{X}, \mathcal{Y}) \to \mathbb{R}_+$, which is small when $\hat{h}(\mathrm{x})$ is close to the target y. Two classical loss functions are the squared loss $\ell((\mathrm{x}, \mathrm{y}), h) = \frac{1}{2}\|h(\mathrm{x}) - \mathrm{y}\|^2$ in regression and the 0-1 loss in classification $\ell((\mathrm{x}, \mathrm{y}), h) = 1_{h(\mathrm{x}) \neq \mathrm{y}}$. In the plug-in perspective, the 0-1 loss is often approximated by the logistic loss: $\ell((\mathrm{x}, \mathrm{y}), h) = -\log\left(e^{h(\mathrm{x})_\mathrm{y}} / \sum_{i \in \mathcal{Y}} e^{h(\mathrm{x})_i}\right)$.

Objective. The main goal we investigate is to construct a predictor $\hat{h}$ that minimizes the expected loss:

$$h^* \in \operatorname*{arg\,min}_{h \in \mathcal{F}(\mathcal{X},\mathcal{Y})} \left\{ \mathcal{R}(h) := \mathbb{E}_{(\mathrm{x},\mathrm{y})\sim\mathcal{D}}[\ell((\mathrm{x},\mathrm{y}),h)] \right\},$$

where the function $\mathcal{R}$ is referred to as the *population* risk. The measurable function that minimizes the population risk $h^* \in \arg\min_{h\in\mathcal{F}(\mathcal{X},\mathcal{Y})} \mathcal{R}(h)$ is referred to as the Bayes predictor. Since the distribution $\mathcal{D}$ is unknown, the Bayes predictor is inaccessible in practice, and must be approximated. We thus define the excess risk of a predictor $h \in \mathcal{F}(\mathcal{X},\mathcal{Y})$ as the additional risk of this predictor with respect to the Bayes risk: $\mathcal{L}(h,h^*) = \mathcal{R}(h) - \mathcal{R}(h^*)$.

Note that an alternative objective to the minimization of the loss in expectation is to have a small loss with high probability, which can be desirable in many real world scenarios, for instance when the safety of the prediction is important, and being good "on average" is not enough.

Model and Estimation Error. Searching the best possible predictor among all measurable functions $\mathcal{F}(\mathcal{X},\mathcal{Y})$ is intractable algorithmically. To simplify the problem, we rely on a *model* $\mathcal{M}(\mathcal{X},\mathcal{Y}) \subset \mathcal{F}(\mathcal{X},\mathcal{Y})$, which restricts the search of the best predictor within a sub-set of the measurable functions. This restriction comes however at the cost of some bias, which is referred to as the *approximation error*:

$$\mathcal{R}(h) - \min_{h\in\mathcal{F}(\mathcal{X},\mathcal{Y})} \mathcal{R}(h) = \underbrace{\mathcal{R}(h) - \min_{h\in\mathcal{M}(\mathcal{X},\mathcal{Y})} \mathcal{R}(h)}_{\text{estimation error}} + \underbrace{\min_{h\in\mathcal{M}(\mathcal{X},\mathcal{Y})} \mathcal{R}(h) - \min_{h\in\mathcal{F}(\mathcal{X},\mathcal{Y})} \mathcal{R}(h)}_{\text{approximation error}}.$$

In the remainder of this manuscript, we focus only on minimizing the estimation error.

Learning Rule. The machine learning algorithm, or the learning rule, aims at finding the best possible predictor among the model $\mathcal{M}(\mathcal{X},\mathcal{Y})$, by relying on an accessible dataset $\mathcal{D}_n = ((\mathrm{x}_i,\mathrm{y}_i)_{i\in[n]})$ of n independent random variables of distribution $\mathcal{D}$. Formally it is a measurable map

$$\hat{h} : \cup_{n\geq 1}(\mathcal{X}\times\mathcal{Y})^n \to \mathcal{M}(\mathcal{X},\mathcal{Y}).$$

For conciseness, we denote $\hat{h}(\mathcal{D}_n,\cdot)$ as $\hat{h}$.

Since we cannot access directly the risk function $\mathcal{R}$, the most common approach to leverage the dataset $\mathcal{D}_n$ is to propose a predictor that behaves well on this dataset. This is known as *empirical risk minimization* (ERM). Formally, denoting the empirical risk function $\hat{\mathcal{R}}_n(h) := \frac{1}{n}\sum_{i\in[n]} \ell((\mathrm{x}_i,\mathrm{y}_i),h)$, the empirical risk minimizer is

$$\hat{h}_{\mathrm{ERM}} \in \operatorname*{arg\,min}_{h\in\mathcal{M}(\mathcal{X},\mathcal{Y})} \hat{\mathcal{R}}_n(h).$$

Minimizing the empirical risk instead of the population risk comes however at the price of a new source of error in the estimation error, which, noting $h_{\mathcal{M}} = \arg\min_{h\in\mathcal{M}(\mathcal{X},\mathcal{Y})}$, can be

decomposed as

$$\begin{aligned}\mathcal{R}(h) - \mathcal{R}(h_{\mathcal{M}}) \leq & \mathcal{R}(h) - \mathcal{R}_n(h) + \mathcal{R}_n(h) - \mathcal{R}_n(\hat{h}_{ERM}) \\ & + \underbrace{\mathcal{R}_n(\hat{h}_{ERM}) - \mathcal{R}_n(h_{\mathcal{M}})}_{\leq 0} + \mathcal{R}_n(h_{\mathcal{M}}) - \mathcal{R}(h_{\mathcal{M}}) \\ \leq & \mathcal{R}_n(h) - \mathcal{R}_n(\hat{h}_{ERM}) + 2 \sup_{h \in \mathcal{M}(\mathcal{X},\mathcal{Y})} |\mathcal{R}(h) - \mathcal{R}_n(h)|.\end{aligned}$$

It follows that, when the empirical risk $\mathcal{R}_n$ is a good approximation of the population risk $\mathcal{R}$ on the model $\mathcal{M}(\mathcal{X}, \mathcal{Y})$, any approximate minimizer of the empirical risk is an approximate minimizer of the population risk. Guaranteeing such property – which is beyond the scope of this manuscript – is the purpose of the empirical process literature, and can be addressed with tools such as the Rademacher complexity or the Vapnik–Chervonenkis dimension.

Parametric Model. When the family of predictors is described by a parameter $\boldsymbol{\theta} \in \mathbb{R}^d$, we say that the model is *parametric*. This can be a convenient way to restrict the set of predictors, and encompasses a broad family of practical methods.

Example 2.1.1.

1. The most basic instance of a parametric model is the linear regression case. In linear regression, the covariate space is $\mathcal{X} = \mathbb{R}^d$, and the parameter space is $\Theta \subset \mathbb{R}^{k \times d}$. For any $\boldsymbol{\theta} \in \Theta$, we associate a linear predictor

$$\hat{h}_{\boldsymbol{\theta}}(\mathrm{x}) = \boldsymbol{\theta}\mathrm{x}.$$

2. Modern machine learning builds on parametric models such as deep neural networks or transformers, in which the parameters are in high-dimensional spaces, with up to $d \approx 10^9$ parameters for Large Language Models (LLMs).

For a parametric model, the supervised learning problem is phrased as

$$\hat{h} \approx \arg\min_{\boldsymbol{\theta} \in \Theta} \mathcal{R}(h_{\boldsymbol{\theta}}) \qquad \text{or} \qquad \hat{h} \approx \arg\min_{\boldsymbol{\theta} \in \Theta} \mathcal{R}_n(h_{\boldsymbol{\theta}}),$$

which motivates the efficient search for such minimizers, and consequently the next section.

2.2 Introduction to First Order Optimization

This section presents basic first-order optimization concepts that is used in the thesis. More detailed introductions can be found for instance in Boyd and Vandenberghe (2004); Bubeck (2015); Nesterov et al. (2018); Bach (2024). We denote $\mathcal{C}^p(\mathbb{R}^d)$ the set of p times continuously differentiable functions from $\mathbb{R}^d$ to $\mathbb{R}$. For $\mathcal{L} \in \mathcal{C}^2(\mathbb{R}^d)$, we denote $\nabla\mathcal{L}$ the gradient of $\mathcal{L}$ and $\nabla^2\mathcal{L}$ its Hessian. We study the following unconstrained optimization problem, for $\mathcal{L} \in \mathcal{C}^p(\mathbb{R}^d)$ with $p \geq 1$:

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{x}). \tag{2.1}$$

We assume that the Problem 2.1 is well-posed, i.e. $\mathcal{L}^* = \inf_{\theta \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{x})$ is finite and that there exists $\boldsymbol{x}^* \in \mathbb{R}^d$ such that $\mathcal{L}(\boldsymbol{x}^*) = \mathcal{L}^*$. We focus on first-order methods, meaning that the algorithms we consider rely only on the gradient $\nabla\mathcal{L}(\boldsymbol{x})$ to solve Part 2.1. The most basic of such algorithms is the *gradient descent* (GD) (Cauchy, 1847).

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_t \nabla\mathcal{L}(\boldsymbol{x}^t). \tag{2.2}$$

Since the gradient describes the growth direction of the function, GD corresponds to minimizing the first order approximation of the objective function along with a quadratic regularization $\mathcal{L}(\boldsymbol{x}^t) + \left\langle \nabla\mathcal{L}(\boldsymbol{x}^t) \ , \ \boldsymbol{x} - \boldsymbol{x}^t \right\rangle + \frac{1}{2\eta_t}\|\boldsymbol{x}^t - \boldsymbol{x}\|^2$. Updating the parameter by performing a step in the opposite direction allows, as long as the step-size η_t is sufficiently small, to decrease the value of the function, $\mathcal{L}(\boldsymbol{x}^{t+1}) \leq \mathcal{L}(\boldsymbol{x}^t)$. This intuition comes however with several questions:

1. When is η "sufficiently small"?
2. When does gradient descent converge, and where to?
3. How fast is the convergence?

Those questions cannot be answered without requiring that the function satisfies certain properties. We thus make additional assumptions on the regularity of the function to analyze gradient-based algorithms.

2.2.1 Convergence of Gradient Descent

While the gradient provides a local information on the growth direction of the function, we need to know for how long this information remains relevant to have non-zero step-sizes. This is directly related to how fast the gradient evolves, which is referred to in optimization as the *smoothness* of the objective function.

Definition 2.2.1 (Smoothness)**.** A function $\mathcal{L} \in \mathcal{C}^1(\mathbb{R}^d)$ is *L-smooth* if its gradient is L-Lipschitz, i.e. for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, we have

$$\|\nabla\mathcal{L}(\boldsymbol{x}) - \nabla\mathcal{L}(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|.$$

The smoothness of the function is also equivalent to

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, |\mathcal{L}(\boldsymbol{x}) - \mathcal{L}(\boldsymbol{y}) - \langle \nabla\mathcal{L}(\boldsymbol{y}) \ , \ \boldsymbol{x} - \boldsymbol{y} \rangle| \leq \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2,$$

and, when $\mathcal{L} \in \mathcal{C}^2(\mathbb{R}^d)$ to $\sup_{\boldsymbol{x} \in \mathbb{R}^d} \|\nabla^2\mathcal{L}(\boldsymbol{x})\|_2 \leq L$ (see Nesterov et al. (2018)).

The smoothness thus provides a quadratic upper-bound on the function value, which by itself allows guaranteeing a decrease of the function value for non-zero step-sizes.

Lemma 2.2.2 (Descent Lemma)**.** *Let $\mathcal{L} \in \mathcal{C}^1(\mathbb{R}^d)$ be L-smooth. Consider Equation (2.2)*

with step-size $\eta_t \leq 1/L$. Then, for any iteration $t \geq 0$, it holds that

$$\frac{\eta_t}{2}\|\nabla\mathcal{L}(\boldsymbol{x}^t)\|^2 \leq \mathcal{L}(\boldsymbol{x}^t) - \mathcal{L}(\boldsymbol{x}^{t+1}).$$

Proof. Definition 2.2.1 provides the quadratic upper-bound

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}^{t+1}) &\leq \mathcal{L}(\boldsymbol{x}^t) + \left\langle \nabla\mathcal{L}(\boldsymbol{x}^t)\ ,\ \boldsymbol{x}^{t+1} - \boldsymbol{x}^t \right\rangle + \frac{L}{2}\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\|. \\ &= \mathcal{L}(\boldsymbol{x}^t) - \eta_t(1 - \eta_t\frac{L}{2})\|\nabla\mathcal{L}(\boldsymbol{x}^t)\|^2\end{aligned}$$

Using $\eta \leq 1/L$ and rearranging terms gives the result. □

It follows that, after T iterations of gradient descent with constant step-size $\eta \leq 1/L$, the iterates verify

$$\sum_{t=1}^{T}\|\nabla\mathcal{L}(\boldsymbol{x}^t)\|^2 \leq \frac{2}{\eta}(\mathcal{L}(\boldsymbol{x}^0) - \mathcal{L}(\boldsymbol{x}^*)).$$

And thus $\nabla\mathcal{L}(\boldsymbol{x}^T) \to 0$, i.e. the iterates $\boldsymbol{x}^T$ converge to a critical point (which can be expected to be a local minimum since $\mathcal{L}(\boldsymbol{x}^{t+1}) \leq \mathcal{L}(\boldsymbol{x}^t)$). However, without further assumptions, both $\|\boldsymbol{x}^T - \boldsymbol{x}^*\|$ and $\mathcal{L}(\boldsymbol{x}^T) - \mathcal{L}(\boldsymbol{x}^*)$ can still be arbitrarily large.

Convexity is a property that allows to translate local information – such as the iterate being at a local minimum – into global information on the function – the local minimum is actually a global minimum.

Definition 2.2.3 (Convexity)**.** Let $\mathcal{L} \in \mathcal{C}^1(\mathbb{R}^d)$ and $\mu \geq 0$. Then $\mathcal{L}$ is *μ-strongly convex* if, for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$,

$$\mathcal{L}(\boldsymbol{x}) \geq \mathcal{L}(\boldsymbol{y}) + \langle \nabla\mathcal{L}(\boldsymbol{y})\ ,\ \boldsymbol{x} - \boldsymbol{y}\rangle + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

When $\mu = 0$, then $\mathcal{L}$ is merely *convex*.

The strong-convexity of a function $\mathcal{L}$ and its smoothness can be conveniently summarized by using the notion of Bregman divergence

Definition 2.2.4 (Bregman Divergence)**.** Let $\mathcal{L} \in \mathcal{C}^1(\mathbb{R}^d)$, the Bregman divergence of $\mathcal{L}$, noted $D_{\mathcal{L}}$ is defined as

$$D_{\mathcal{L}}(\boldsymbol{x}; \boldsymbol{y}) := \mathcal{L}(\boldsymbol{x}) - \mathcal{L}(\boldsymbol{y}) - \left\langle \nabla\mathcal{L}(\boldsymbol{y})\ ,\ \boldsymbol{x} - \boldsymbol{y}\right\rangle.$$

The Bregman divergence has many useful properties, the first one being that the smoothness and strong convexity of a function $\mathcal{L}$ is equivalent to lower and upper-bounds on the Bregman divergence of $\mathcal{L}$:

$$\frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 \leq D_{\mathcal{L}}(\boldsymbol{x}; \boldsymbol{y}) \leq \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2. \tag{2.3}$$

It follows from the above perspective that the Bregman divergence is sometimes considered as a generalization of the Euclidean distance. Note however that the Bregman divergence

is not symmetric, and that, if $D_{\mathcal{L}}$ is (strongly) convex in its first argument when $\mathcal{L}$ is (strongly) convex, it is not necessarily the case for the second argument.

We now state the convergence of the gradient descent method under convexity and strong convexity assumptions for smooth functions.

Proposition 2.2.5. *Let $\mathcal{L} \in \mathcal{C}^1(\mathbb{R}^d)$ be L-smooth and convex, then the iterates of GD with $\eta_t \leq 1/L$ yield*

$$\mathcal{L}(\boldsymbol{x}^T) - \mathcal{L}(\boldsymbol{x}^*) \leq \frac{1}{2\sum_{t=1}^T \eta_t} \|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2. \tag{2.4}$$

If additionally $\mathcal{L}$ is μ-strongly convex, then for $\eta_t = 1/L$, it holds

$$\|\boldsymbol{x}^T - \boldsymbol{x}^0\|^2 \leq \prod_{t=1}^T (1 - \eta_t \mu) \|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2. \tag{2.5}$$

Proof. Both proofs begin by decomposing the distance $\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\|^2$ as follows

$$\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\|^2 = \|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 - 2\eta_t \left\langle \boldsymbol{x}^t - \boldsymbol{x}^* \ , \ \nabla\mathcal{L}(\boldsymbol{x}^t) \right\rangle + \eta^2 \|\nabla\mathcal{L}(\boldsymbol{x}^t)\|^2$$
$$\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\|^2 = \|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 - 2\eta_t D_{\mathcal{L}}(\boldsymbol{x}^*; \boldsymbol{x}^t) - 2\eta_t(\mathcal{L}(\boldsymbol{x}^t) - \mathcal{L}(\boldsymbol{x}^*)) + \eta_t^2 \|\nabla\mathcal{L}(\boldsymbol{x}^t)\|^2.$$

Now, using the smoothness of $\mathcal{L}$ with the descent Lemma 2.2.2 yields

$$\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\|^2 \leq \|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 - 2\eta_t D_{\mathcal{L}}(\boldsymbol{x}^*; \boldsymbol{x}^t) - \eta_t(\mathcal{L}(\boldsymbol{x}^{t+1}) - \mathcal{L}(\boldsymbol{x}^*)). \tag{2.6}$$

1. If $\mathcal{L}$ is convex, then $D_{\mathcal{L}} \geq 0$, and it follows from Equation (2.6)

   $$2\eta_t(\mathcal{L}(\boldsymbol{x}^t) - \mathcal{L}(\boldsymbol{x}^*)) \leq \|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 - \|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\|^2.$$

   Summing and using that $\mathcal{L}(\boldsymbol{x}^T) \leq \mathcal{L}(\boldsymbol{x}^t)$ for $t \leq T$ according to Lemma 2.2.2 yields the result.

2. If $\mathcal{L}$ is μ-strongly convex, then using the strong convexity to lower bound $D_{\mathcal{L}}(\boldsymbol{x}^*; \boldsymbol{x}^t)$ and using $\mathcal{L}(\boldsymbol{x}^{t+1}) \geq \mathcal{L}(\boldsymbol{x}^*)$ in Equation (2.6) ensures that

   $$\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\|^2 \leq (1 - \eta_t \mu) \|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2.$$

   Chaining it for $t = 1$ to T yields the result.

□

Convergence rates are often alternatively compared by considering the *number of iterations* to reach an error ε. In the case of constant step-sizes $\eta_t = 1/L$, we thus say that according to Proposition 2.2.5 GD on smooth convex functions converges with the *sub-linear rate* of $\mathcal{O}(1/\varepsilon)$, while GD on smooth strongly functions converges with the *linear rate* of $\mathcal{O}(\kappa \log(1/\varepsilon))$, where

$$\kappa := \frac{L}{\mu}$$

is called the *condition number* of the loss function $\mathcal{L}$. The condition number of the objective function $\mathcal{L}$ is a key quantity to measure the convergence speed of first-order strongly convex optimization. As we will see in the next section, the convergence rate of gradient based methods are limited by the condition number.

2.2.2 Faster Algorithms

In first order optimization, and notably in machine learning, the main source of computational complexity often comes from computing the gradients. It follows that either reducing this cost, or reducing the number of oracle calls is of paramount importance. We first discuss how to reduce the number of oracle calls, and then the usage of stochastic oracles to rely on cheaper oracles.

Lower Bound. A natural question when studying first order methods is the following: How fast can an algorithm be on a given class of functions by relying only on first order oracles?

Theorem 2.2.6 (Nesterov et al. (2018))**.** *For any $\boldsymbol{x}^0 \in \mathbb{R}^\infty$, $\mu > 0$ and $\kappa > 1$, there exists a quadratic function $f : \mathbb{R}^\infty \to \mathbb{R}$ that is μ-strongly convex and $\kappa\mu$-smooth such that, for any iterative method $\mathcal{M}$ that generates a sequence of test points $\{\boldsymbol{x}^t\}$ such that $\boldsymbol{x}^t \in \boldsymbol{x}^0 + \operatorname{span}\big[\nabla\mathcal{L}(\boldsymbol{x}^0), \dots, \nabla\mathcal{L}(\boldsymbol{x}^{t-1})\big]$, we have for $t > 1$*

$$\mathcal{L}(\boldsymbol{x}^t) - \mathcal{L}(\boldsymbol{x}^*) \geq \frac{\mu}{2}\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{2t} \|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2,$$
$$\|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2 \geq \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{2t} \|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2.$$

At this point, we see that relying only on first order oracles prevents to have better convergence rates than linear ones, i.e. the rate of gradient descent in $\mathcal{O}(\kappa \log(1/\varepsilon))$ is tight when we do not take into account the dependence in κ. However, this dependence in κ can have a huge impact in practice when κ is large. A variation of gradient descent enjoys a convergence rate of $\mathcal{O}(\sqrt{\kappa}\log(1/\varepsilon))$, that matches the lower bound of Theorem 2.2.6. It is generally referred to as Nesterov's accelerated gradient descent (AGD), and follows the following equations

$$\begin{cases} \boldsymbol{x}^{t+1} &= \boldsymbol{y}^t - \eta^t \nabla\mathcal{L}(\boldsymbol{y}^t) \\ \boldsymbol{y}^{t+1} &= \boldsymbol{x}^{t+1} + \beta^t(\boldsymbol{x}^{t+1} - \boldsymbol{x}^t). \end{cases} \tag{AGD}$$

This algorithm corresponds to the gradient descent method in which the point at which the gradient is computed is extrapolated by considering the previous iterations. It enjoys the following guarantees

Theorem 2.2.7 (Nesterov et al. (2018))**.** *Let $\beta = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$, then the iterates $\{\boldsymbol{x}^t\}$ of*

Equation (AGD) *have the following convergence*

$$\mathcal{L}(\boldsymbol{x}^t) - \mathcal{L}(\boldsymbol{x}^*) \leq \frac{L+\mu}{2}\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2 e^{-t/\sqrt{\kappa}}.$$

Achieving such a linear rate of $\mathcal{O}(\sqrt{\kappa}\log(1/\varepsilon))$ is often referred to as achieving an accelerated rate of convergence on smooth strongly convex functions. Achieving such rate in an adversarial setting is one goal of the Chapter 6. One can however notice that such accelerated rate is vacuous when $\mu = 0$. Even if this is beyond the interest of this manuscript, we point out that acceleration on smooth convex functions refers to having a convergence rate of $\mathcal{O}(1/\sqrt{\varepsilon})$, and is also achieved by a specific parametrization of AGD. Additionally, this rate is tight on the class of smooth convex functions (Nesterov et al., 2018). For an extensive study on the design of accelerated methods, we refer the reader to d'Aspremont et al. (2021).

2.2.3 Stochastic Gradient Descent

Another approach to reducing the optimization cost is to rely on cheaper oracles. This is for instance very useful in machine learning where the ERM objective writes

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \mathcal{L}(\boldsymbol{x}) := \frac{1}{n}\sum_{i\in[n]} \mathcal{L}_i(\boldsymbol{x}), \tag{2.7}$$

where n denotes the number of samples in the training dataset and each function $\mathcal{L}_i$ corresponds to an empirical loss on a sample. In this case, computing the full gradient requires n times more computation than computing a single gradient $\nabla\mathcal{L}_i$, which can still be very informative. By picking at each iteration a sample at random, an unbiased stochastic oracle of the gradient is produced. This can be abstracted by the stochastic gradient descent, introduced by Robbins and Monro (1951):

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_t g_t \quad \text{where} \quad \mathbb{E}[g_t] = \nabla\mathcal{L}(\boldsymbol{x}^t). \tag{2.8}$$

This method is widely used in practice: not only does it alleviate the cost of computing the gradient on the full dataset, but it also allows directly optimizing the population risk when performing a single pass on the dataset (i.e. each training sample is used at most once), thus avoiding over-fitting issues (Nemirovskij and Yudin, 1983; Polyak and Juditsky, 1992). In order to analyze SGD, one needs to make assumptions on the noise $g_t - \nabla\mathcal{L}(\boldsymbol{x}^t)$. A basic assumption is to require a uniformly bounded variance of the gradient oracle.

Assumption 2.2.8. The stochastic gradients oracles g_t are i.i.d. and such that at every $\boldsymbol{x}^t \in \mathbb{R}^d$

$$\mathbb{E}[g_t] = \nabla\mathcal{L}(\boldsymbol{x}^t) \quad \text{and} \quad \mathbb{E}[\|g_t - \nabla\mathcal{L}(\boldsymbol{x}^t)\|^2] \leq \sigma^2.$$

Under Assumption 2.2.8, one can also show that stochastic gradient descent converges, either with high-probability, or, as we show now, in expectation.

Proposition 2.2.9. *Let $\mathcal{L}$ be μ-strongly convex and L-smooth. Consider $\{\boldsymbol{x}^t\}$ the sequence generated by Equation (2.8) with $\eta_t = \eta \leq 1/L$ where the stochastic oracle satisfies Assumption 2.2.8. Then if $\mu > 0$, we have*

$$\mathbb{E}[\|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2] \leq (1-\eta\mu)^T \|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2 + \eta\frac{\sigma^2}{\mu},$$

while, for $\mu = 0$, the average iterate $\bar{\boldsymbol{x}}^T = \frac{1}{T}\sum_{t=1}^T \boldsymbol{x}^t$ verifies

$$\mathbb{E}[\mathcal{L}(\bar{\boldsymbol{x}}^t)] - \mathcal{L}(\boldsymbol{x}^*) \leq \frac{\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2}{\eta T} + \eta\sigma^2.$$

Proof. Similarly to the proof of gradient descent, decomposing $\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\|^2$, using Assumption 2.2.8 and Lemma 2.2.2 (in expectation) yields,

$$\mathbb{E}[\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^*\|^2] \leq \mathbb{E}[\|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2] - 2\mathbb{E}[\eta_t D_{\mathcal{L}}(\boldsymbol{x}^*; \boldsymbol{x}^t)] - \eta_t(\mathbb{E}[\mathcal{L}(\boldsymbol{x}^{t+1})] - \mathcal{L}(\boldsymbol{x}^*)) + \eta_t^2\sigma^2.$$

Strongly convex case. Similarly to the non-stochastic case, if $\mu > 0$, we have $2D_{\mathcal{L}}(\boldsymbol{x}^*; \boldsymbol{x}^t) \geq \mu\|\boldsymbol{x}^t - \boldsymbol{x}^*\|^2$, and $\mathcal{L}(\boldsymbol{x}^t) - \mathcal{L}(\boldsymbol{x}^*) \geq 0$. This yields an arithmetic-geometric sequence which gives the result.

Convex case. If $\mu = 0$, then $\mathcal{D}_{\mathcal{L}}(\boldsymbol{x}^*, \boldsymbol{x}^t) \geq 0$ and similarly to the deterministic proof

$$\sum_{t=1}^T \eta_t\big(\mathbb{E}[\mathcal{L}(\boldsymbol{x}^t)] - \mathcal{L}(\boldsymbol{x}^*)\big) \leq \|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2 - \mathbb{E}[\|\boldsymbol{x}^T - \boldsymbol{x}^0\|^2] + \sum_{t=1}^T \eta_t^2\sigma^2.$$

Now, using the convexity of $\mathcal{L}$ we have $\mathcal{L}(\bar{\boldsymbol{x}}^T) \leq T^{-1}\sum_{t=1}^T \mathcal{L}(\boldsymbol{x}^t)$, and using $\eta_t = \eta$ gives the desired result. □

It follows from Proposition 2.2.9 that SGD converges at a rate $\tilde{\mathcal{O}}(1/\varepsilon)$ when $\mathcal{L}$ is strongly convex (take $\eta = {}^{\ln(T)}/_T$) and at a rate $\mathcal{O}(1/\varepsilon^2)$ when $\mathcal{L}$ is merely convex (take $\eta = {}^1/_{\sqrt{T}}$). Note that as explained above, those bounds directly bound the population risk when $T = n$ and the dataset is shuffled randomly at the beginning. Due to the important practical interest, SGD has been extensively studied (Bach and Moulines, 2013; Dieuleveut et al., 2020).

Variance Reduction. If the rates in Proposition 2.2.9 are un-improvable in general (Agarwal et al., 2009), leveraging the finite-sum structure of Equation (2.7) allows the design of linearly converging algorithms while still using mostly stochastic oracles of cost $\mathcal{O}(1)$. Those algorithms – SAG (Schmidt et al., 2017), SVRG (Johnson and Zhang, 2013), SAGA (Defazio et al., 2014) – referred to as *variance reduction* methods are beyond the scope of this manuscript.

2.3 Distributed Learning

2.3.1 Introduction

Motivation. In the previous section, we only considered basic optimization algorithms in which only one computer was involved. It follows that the algorithms are rather simple: an oracle is queried (a gradient or a stochastic gradient), the information is then directly returned to the computer that uses it to estimate a new iterate at which the oracle is again queried. Considering distributed systems allows for more complex algorithms, with both potential computational benefits and new constraints: on the one hand multiple oracles can be queried at the same time, opening the door to multi-processing and computational gain, but on the other hand the communication constraints, such as delays and bandwidth issues, can reduce those gains. The design of distributed systems thus requires answering several questions to identify the constraints and the goal to achieve. Where is the data? Who is participating? What are the communication and computational constraints? Can all the nodes be connected at the same time? Are there latency issues?

To illustrate those problematic, three canonical instances are considered: the *datacenter* setting, which corresponds to a cluster of computers designed for an efficient parallel computing task, the *cross-silo* setting, in which the parties are whole organizations such as hospitals or companies, and last the *cross-device* setting, in which the parties are IoT devices or personal devices such as smartphones and computers (Kairouz and McMahan, 2021).

- In the datacenter setting, nodes within a cluster have to train a model on a commonly available dataset, which can be balanced across nodes. The nodes can have a very high bandwidth, are highly reliable, with a high computational power, and are relatively few (although the rise of LLMs leads to ever bigger and bigger datacenters with up to $10^6 - 10^7$ GPUs).
- In the cross-silo setting, the parties are expected to be reliable, highly available, and relatively few (less than 100). However, they might have heterogeneous datasets which are kept locally with privacy concerns.
- In the cross-devices setting, devices can be highly unreliable, with up to 10^9 clients, that can disconnect unexpectedly, with heterogeneous datasets, low bandwidth and computational power. As in the cross-device setting, the data might be highly heterogeneous with privacy concerns.

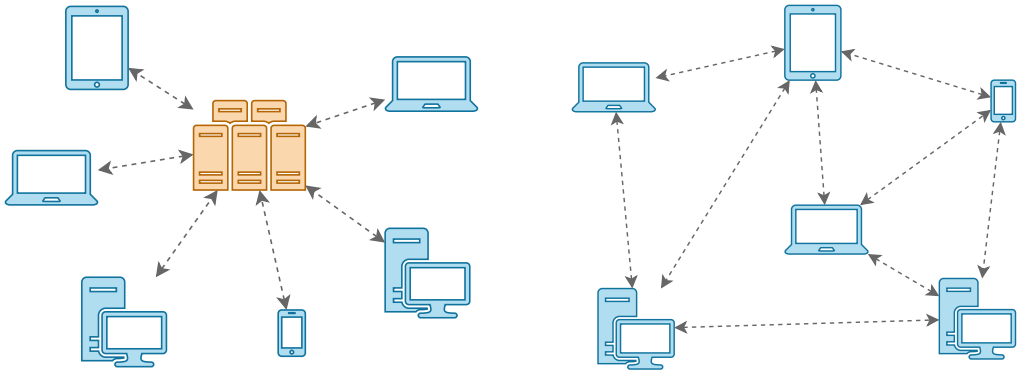
The practical implementation of federated learning methods requires understanding all the particular constraints and the specificity of the system. However, for the theoretician, studying the impact of the different constraints requires analyzing simpler models, in which the communication system is abstracted, where some constraints are translated into formal assumptions or formal requirements. In this thesis, we focus mostly on robustness concerns, by proposing either methods robust to unreliable devices that deviate from the prescribed behavior, or by understanding the privacy of statistical methods.

Formal Setting. In this section, we consider a system of n computers, also equivalently called *clients*, *nodes*, *devices* or *parties*. Each computer $i \in [n]$ has local access to a loss function $\mathcal{L}_i$, which corresponds typically to an empirical risk on a local dataset stored on the computer i, denoted $\mathcal{D}^i$.

Those computers aim at collaboratively minimizing the joint loss function

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} \frac{1}{n} \sum_{i \in [n]} \mathcal{L}_i(\boldsymbol{x}).$$

To do so, nodes collaborate by communicating within a network. This communication network is commonly modeled by assuming either that the communication is centralized by a server, or that the communication is performed in a decentralized manner, with clients communicating directly in a peer-to-peer manner (cf. Figure 2.1). We refer to the centralized communication as the federated learning setting, although the different use cases listed above (datacenters, cross-silo, cross-device) apply to the decentralized communication setting as well.



a) Federated Learning b) Decentralized Learning

Figure 2.1: Typology of communication systems.

2.3.2 Federated Learning

In federated learning, all clients communicate with a central server that supervises the optimization process. The server aggregates the information coming from the nodes, performs updates, and sends them back to the clients. A very simple algorithm to distribute the optimization process is to rely on distributed gradient descent:

1. The server sends the current parameter $\boldsymbol{x}^t$ to the clients.

2. Each client computes a gradient on its loss function $\nabla \mathcal{L}_i(\boldsymbol{x}^t)$ and sends it back to the server.

3. The server averages the gradients $\nabla \mathcal{L}(\boldsymbol{x}^t) = \frac{1}{n} \sum_{i \in [n]} \nabla \mathcal{L}_i(\boldsymbol{x}^t)$, and uses it to update the parameter $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta \nabla \mathcal{L}(\boldsymbol{x}^t)$.

Thanks to the linearity of the gradient, distributed gradient descent (DGD) is equivalent to gradient descent or any centralized gradient-based algorithms. In particular, DGD requires the same number of iterations as GD to converge. Assuming a communication time τ and 1 to compute a local loss function gradient, each iteration is performed in a time $\tau + 1$, while gradient descent on the server only would have required a time n to compute the full gradient. Thus, relying on distributed algorithm can reduce the time complexity by a factor $(\tau + 1)/n$. Depending on the communication time τ, this can either significantly speed-up, or slow down the process. If $\tau < 1$, the time complexity is $T_{\mathrm{GD}}(\varepsilon) \geq nT_{\mathrm{DGD}}(\varepsilon)$, thus distributed gradient descent achieves a *linear speed-up* compared to gradient descent. This phenomenon appears also in distributed stochastic gradient descent, in which averaging the stochastic oracles reduces the variance by a factor $1/n$.

Local Methods. A large body of research in federated learning focuses on local methods, in which each node performs multiple local optimization updates between each communication with the central server. This allows for potential speed-up when the communication delays are large, while still allowing to keep the data stored locally. One typical example of those methods is the Federated Averaging algorithm (McMahan et al., 2017; Stich, 2019), in which some users are selected at random at each round, receive the current model, compute a few gradient descent steps on their local loss function and send their local model back to the central server. The new global model is then obtained by averaging those local models. However, when the local loss functions are heterogeneous, the FedAvg algorithm converges only under vanishing step-sizes, and more involved methods involving control variates such as Scaffold (Karimireddy et al., 2020; Mangold et al., 2025) allow to achieve a linear speed-up with constant step-sizes while converging to the true solution.

2.3.3 Decentralized Learning

The hierarchical organization of federated learning has the major benefit of allowing a greater control on the learning process, which can be easily audited and modified by accessing the central server. This correlatively comes with the single point of failure vulnerability, and the centralizing server appears as both a computational and a communication bottleneck, since the load on this particular computer is drastically larger than on the simple clients. Decentralized optimization alleviates this issue by allowing the nodes to communicate directly with each other. Not only can this peer-to-peer communication speed-up the optimization (Lian et al., 2017), but it also provides more redundancy to the system, enhancing its resilience. Decentralized systems are also explored since they offer more sovereignty to the users as they can be more resilient to policy changes of the federating structure, as it happened for Twitter. This is for instance the key motivation of the Fediverse and, to some extent, of Bluesky.

Formally, decentralized learning consists of a communication network in which the devices communicate in a peer-to-peer manner. The communication network is commonly modeled as a graph, denoted $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ are the vertices of the graph, representing

the clients, and $\mathcal{E}$ the edges. In this graph, which is assumed to be connected, the clients can only communicate with their neighbors through the edges, as illustrated in Figure 2.1. Since all clients do not communicate directly with each other, transmitting information within this graph is non-trivial. The perspective investigated in this thesis is to consider *gossip algorithms* (Boyd et al., 2006; Nedic and Ozdaglar, 2009; Shi et al., 2015; Scaman et al., 2017; Kovalev et al., 2020) for optimizing the joint objective. In gossip algorithms, each computer iteratively updates its parameters through linear combinations of local information (such as gradient or past parameters) with information coming from the neighbors in the graph.

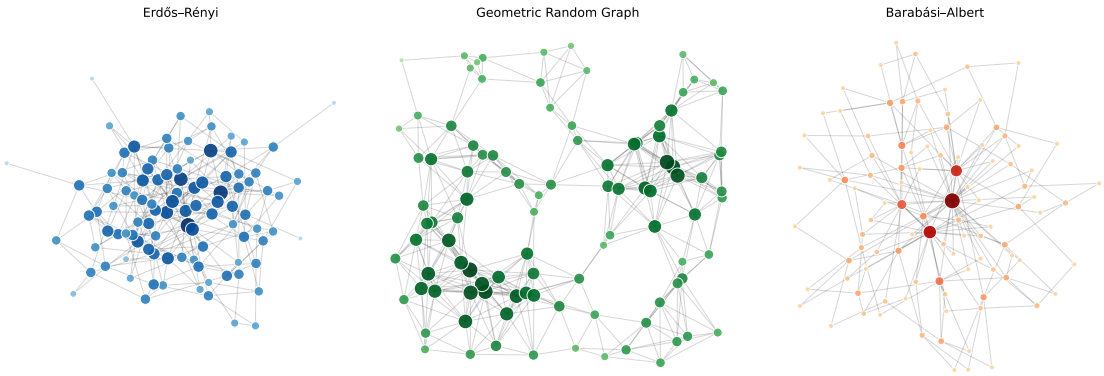


Figure 2.2: An Erdös Renyi, a geometric random graph and a Barabàsi-Albert graph with $n = 100$ nodes. Darker, larger circles correspond to nodes with larger degree.

2.3.3.1 The Averaging Problem

The most basic primitive of gossip algorithms is the one of gossip averaging. As in federated learning, in which the information from the clients is aggregated by averaging gradients or local parameters, understanding the averaging dynamic of gossip algorithms is key for designing gossip optimization methods. In the synchronous gossip averaging protocol, each client $i \in [n]$ synchronously updates its parameters by performing an approximate average of its parameter with the ones of its neighbors in the graph of communication $\mathcal{G}$ with updates of the form

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t - \eta \sum_{j \in [n]} w_{ij}(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t), \tag{2.9}$$

where w_{ij} is a weight associated with the edge $i \sim j \in \mathcal{E}$, and $\eta \geq 0$ denotes a communication step size. This update can also be compactly written by incorporating the weights $(w_{ij})_{ij \in [n]^2}$ into a gossip matrix $\boldsymbol{W}$ (Boyd et al., 2006; Scaman et al., 2017).

Definition 2.3.1 ((PSD) gossip matrix)**.** A gossip matrix of a connected graph $\mathcal{G}$ satisfies the following condition:

1. $\boldsymbol{W}$ is an $n \times n$ symmetric positive semi-definite matrix,
2. the kernel of $\boldsymbol{W}$ is restricted to the constant vector: $\ker(\boldsymbol{W}) = \text{span}(\mathbf{1}_n)$, where

$\mathbf{1}_n = (1, \dots, 1)^T \in \mathbb{R}^n$,

3. $\boldsymbol{W}_{ij} = 0$ if i and j are not neighbors in the graph $\mathcal{G}$, i.e. $(i, j) \notin \mathcal{E}$.

When the edges of $\mathcal{G}$ are associated with weights, then we canonically consider the Laplacian matrix of $\mathcal{G}$, defined as $\boldsymbol{W}_{ij} = -w_{ij}$ if $i \neq j$ and $\boldsymbol{W}_{ii} = \sum_{j \in [n] \setminus \{i\}} w_{ij}$.

The definition of gossip matrix allows to write the gossip update 2.9 as

$$\boldsymbol{X}^{t+1} = (\mathrm{I}_n - \eta \boldsymbol{W}) \boldsymbol{X}^t, \quad \text{where} \quad \boldsymbol{X}^t := (\boldsymbol{x}_1^t, \dots, \boldsymbol{x}_n^t)^T. \tag{2.10}$$

Note that the reader can consider $d = 1$ for convenience, in order to have vectors instead of matrices $\boldsymbol{X}$, without missing any major point.

We the eigenvalues of $\boldsymbol{W}$ as $\lambda_n(\boldsymbol{W}) \geq \dots, \geq \lambda_2(\boldsymbol{W}) > \lambda_1(\boldsymbol{W}) = 0$, where the last strict inequality holds as long as the graph is connected. We also define the *spectral gap* of the gossip matrix as $\gamma = \lambda_2/\lambda_n$. The spectral gap is essentially the inverse of condition number of the gossip matrix restricted to its span, and is characteristic of the worst-case convergence rates of gossip algorithms.

Proposition 2.3.2. *Consider the gossip update Equation (2.10), then if $\boldsymbol{W}$ is a gossip matrix and $\eta = 1/\lambda_n(\boldsymbol{W})$, it holds that all nodes converge linearly at rate $\gamma^{-1} \log(1/\varepsilon)$ to the average initial parameter, i.e.*

$$\frac{1}{n} \sum_{i \in [n]} \|\boldsymbol{x}_i^t - \overline{\boldsymbol{x}}^0\|^2 \leq (1 - \gamma)^t \frac{1}{n} \sum_{i \in [n]} \|\boldsymbol{x}_i^0 - \overline{\boldsymbol{x}}\|^2.$$

Remark 2.3.3. In fact, two definitions of a *gossip matrix* (and of the associated spectral gap) exist, which are equivalent, up to a factor 2 in the spectral gap.

- Definition 2.3.1 defining PSD gossip matrices $\boldsymbol{W}_{PSD}$,
- Requiring the gossip matrix $\boldsymbol{W}_{sto}$ to be symmetric bistochastic i.e. $\boldsymbol{W}_{sto} \in [0, 1]^{n \times n}$, $\boldsymbol{W}_{sto} \mathbf{1}_n = \mathbf{1}_n$ and $\mathbf{1}_n^T \boldsymbol{W}_{sto} = \mathbf{1}_n^T$. In this case, the spectral gap is defined as $\gamma_{sto} = 1 - \max\{|\lambda(\boldsymbol{W}_{sto})|, \lambda(\boldsymbol{W}_{sto}) \neq 1\}$.

One can go from one definition to the other as

- **Bistochastic to PSD:** consider $\boldsymbol{W}_{PSD} \leftarrow \mathrm{I} - \boldsymbol{W}_{sto}$,
- **PSD to Bistochastic:** consider $\boldsymbol{W}_{sto} \leftarrow \mathrm{I} - \frac{1}{\lambda_n(\boldsymbol{W}_{PSD})} \boldsymbol{W}_{PSD}$.

On the graph's connectivity. The spectral gap of the graph $\gamma = \lambda_2(\boldsymbol{W})/\lambda_n(\boldsymbol{W})$ is related to the *connectivity* of the graph, which, formally, can be defined as the minimal number of edges that need to be removed for disconnecting the graph. Even if the relation between the connectivity and the spectral gap is not strictly monotonic, weakly connected graphs (referred to as *sparse* graphs) tends to have smaller spectral gaps, which in turns

decreases the convergence speed of gossip algorithms. Since at the same time the number of bits send and received by each node during gossip communication grows with the number of their neighbors, taking both into account bandwidth constraints, a fast convergence requirement leads to search for graphs with both a small the maximal degree and large spectral gap. Such graphs exists, and are called expander graphs, which are graphs with strong connectivity properties. Such graphs can be created by using Erdös Renyi graphs or the hypercube graph with a good parametrization. We refer to Table 2.1 for examples of the spectral gap on standard graphs.

A related metric to the spectral gap is the *algebraic connectivity* of the graph, which corresponds to $\lambda_2(\boldsymbol{W})$, the second smallest eigenvalue of the graph's Laplacian. When the Laplacian is un-normalized, i.e. the weights on the edges are unitary, $\boldsymbol{W}_{ij} = -1$ for $(i,j) \in \mathcal{E}$, the algebraic connectivity is always smaller than the connectivity of the graph, while being larger than $1/(nD)$, where D is the diameter of the graph. On the other side, the largest eigenvalue of the graph is of the order of the largest degree, i.e. $\lambda_n(\boldsymbol{W}) \in [d_{\max}, 2d_{\max}]$, where both bounds are achieved by regular or bipartite graphs.

Graph	Complete	Expander	Torus of dimension d	Line	Circular
γ	1	$\mathcal{O}(1)$	$\mathcal{O}(d/n^{2/d})$	$\mathcal{O}(1/n^2)$	$\mathcal{O}(1/n^2)$
Degree	n-1	$\mathcal{O}(1)$	$2d$	2	2

Table 2.1: Spectral gap of several standard graphs as a function of the number of nodes.

Faster Updates. The plain gossip update can also be accelerated similarly to the gradient descent scheme. A simple method consists in applying the Chebyshev acceleration to the matrix updates, which yields a $\sqrt{\gamma}\log(1/\varepsilon)$ convergence rate to the average initial parameter(Scaman et al., 2017).

Asynchronous Gossip Update. Gossip can also be performed in an asynchronous manner. A simple model of it is to consider random asynchronous updates (Boyd et al., 2006): at each iteration, an edge $(i,j) \in \mathcal{E}$ is activated at random with probability $\propto w_{ij}$ and the two connected nodes average their parameters. In expectation, the update still follows the same gossip update, and thus the same rate is achieved. Achieving acceleration in the random asynchronous setting requires more involved tools than in the synchronous case (Even et al., 2021, 2024).

2.3.3.2 Decentralized (Stochastic) Gradient Descent

Once the averaging primitive is well-understood, it is natural, similarly to the federated setting, to combine the gossip averaging with a gradient update as

$$\boldsymbol{X}^{t+1} = (\mathrm{I}_n - \eta \boldsymbol{W})\boldsymbol{X}^t - \alpha \boldsymbol{G}(\boldsymbol{X}^t), \tag{2.11}$$

where $\boldsymbol{G}(\boldsymbol{X}^t) = (\nabla \mathcal{L}_1(\boldsymbol{x}_1^t), \dots, \nabla \mathcal{L}_n(\boldsymbol{x}_n^t))^T$, and α is an optimization step size. Yet, when the optimization step size is constant, Equation (2.11) is biased as it converges to the

fixed point $\boldsymbol{X}^*_{\alpha/\eta}$ of the equation $\alpha \boldsymbol{G}(\boldsymbol{X}^*_{\alpha/\eta}) + \eta \boldsymbol{W} \boldsymbol{X}^*_{\alpha/\eta} = 0$, which lies at a distance $\mathcal{O}(\alpha/\eta \sqrt{\kappa \Delta/\gamma})$ of the optimum when the loss functions are heterogeneous (Vogels et al., 2023), where Δ measures the heterogeneity of the loss functions in the pseudo-norm induced by $\boldsymbol{W}^\dagger$. This bias derives from the communication delays within the communication graph: the gossip communication does not compensate the local gradient descent steps that drive local iterates back to the local loss function optimum. Going beyond this non-vanishing error requires either using vanishing step sizes as in SGD or in sub-gradient methods (Nedic and Ozdaglar, 2009; Koloskova et al., 2020), i.e. slowing down the optimization in comparison to the communication, or modifying the update Equation (2.11) to incorporate a bias correction method. This is a key motivation of *gradient tracking* methods, as of *the dual approach*.

Gradient Tracking. Gradient tracking methods (Shi et al., 2015; Nedic et al., 2017) modify the gossip GD update in Equation (2.11) by proceeding to two consecutive updates tailored to ensure that all parameters converge to the same value under fixed optimization step size. For instance, the update of EXTRA (Shi et al., 2015) writes

$$\boldsymbol{X}^{t+1} = (\mathrm{I}_n - \frac{1}{2}\eta \boldsymbol{W}) \boldsymbol{X}^t - \alpha \boldsymbol{G}(\boldsymbol{X}^t) \tag{2.12}$$

$$\boldsymbol{X}^{t+2} = (\mathrm{I}_n - \eta \boldsymbol{W}) \boldsymbol{X}^{t+1} - \alpha \boldsymbol{G}(\boldsymbol{X}^{t+1}). \tag{2.13}$$

Inspecting the fixed point equation resulting from subtracting Equation (2.12) in Equation (2.13) guarantees that $\boldsymbol{W} \boldsymbol{X}^\infty = 0$, i.e. that the nodes are at consensus.

2.3.3.3 The Dual Approach

In this section, we present the dual approach to building decentralized algorithms. If optimal decentralized methods were eventually achieved using EXTRA-like algorithms (Li and Lin, 2020) with primal gradients and proofs, the dual approach offers a principled approach to both design optimal algorithms (Scaman et al., 2017; Hendrikx et al., 2019, 2021; Kovalev et al., 2020) and understand the gradient tracking methods (Jakovetić, 2018). We also detail this approach more thoroughly since it is the motivation of Chapter 5.

The dual approach consist in phrasing the finite sum problem 2.7 as an optimization under constraint problem, where the constraint depends on the communication network:

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^d} \sum_{i \in [n]} \mathcal{L}_i(\boldsymbol{x}) &= \min_{\substack{\boldsymbol{X} \in \mathbb{R}^{n \times d} \\ \boldsymbol{X}_i = \boldsymbol{X}_j \text{ for } (i,j) \in \mathcal{E}}} \sum_{i \in [n]} \mathcal{L}_i(\boldsymbol{X}_i) \\ &= \min_{\boldsymbol{X} \in \mathbb{R}^{n \times d}} \sup_{\boldsymbol{\Lambda} \in \mathbb{R}^{|\mathcal{E}| \times d}} \sum_{i \in [n]} \mathcal{L}(\boldsymbol{X}) - \mathrm{tr}\left(\boldsymbol{\Lambda}^T (\boldsymbol{C}^T \boldsymbol{X})\right), \end{aligned} \tag{2.14}$$

where $\boldsymbol{C} \in \mathbb{R}^{n \times |\mathcal{E}|}$ is a constraint matrix such that for $(i,j) \in \mathcal{E}$, $\boldsymbol{C}^T \boldsymbol{X} = \boldsymbol{X}_i - \boldsymbol{X}_j$, and by a slight abuse of notation $\mathcal{L}(\boldsymbol{X}) = \sum_{i \in [n]} \mathcal{L}_i(\boldsymbol{X}_i)$. Under convexity of the loss functions $\mathcal{L}$, the primal problem Equation (2.14) is equivalent to the dual problem

$$\min_{\boldsymbol{\Lambda} \in \mathbb{R}^{|\mathcal{E}| \times d}} \mathcal{L}^*(\boldsymbol{C}\boldsymbol{\Lambda}), \tag{2.15}$$

where $\mathcal{L}^*$ is the *convex conjugate* (a.k.a. Fenchel or Legendre transform) of $\mathcal{L}$. Recall that the convex conjugate of a function $f : \mathbb{R}^d \to \mathbb{R}$ is defined for $\boldsymbol{y} \in \mathbb{R}^d$ as $f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}\in\mathbb{R}^d}\{\langle \boldsymbol{x}\ ,\ \boldsymbol{y}\rangle - f(\boldsymbol{x})\}$, is convex, and if f is strongly convex, proper and differentiable, it holds that $\nabla f^* = (\nabla f)^{-1}$ (Boyd and Vandenberghe, 2004).

Duality thus allows to phrase the (exact) constrained optimization problem as an un-constrained optimization problem. Yet, there is no notion of decentralization at this point. To see how it can apply, consider that the gradient descent on the dual problem 2.15 writes $\boldsymbol{\Lambda}^{t+1} = \boldsymbol{\Lambda}^t - \eta \boldsymbol{C}^T \nabla \mathcal{L}^*(\boldsymbol{C}\boldsymbol{\Lambda})$. Now, multiplying this GD update by the matrix $\boldsymbol{C} \in \mathbb{R}^{n\times|\mathcal{E}|}$ yields the update

$$\boldsymbol{Y}^{t+1} = \boldsymbol{Y}^t - \eta \boldsymbol{C}\boldsymbol{C}^T \nabla \mathcal{L}^*(\boldsymbol{Y}^t). \tag{2.16}$$

At this point, a quick computation shows that $\boldsymbol{C}\boldsymbol{C}^T$ is the Laplacian matrix of the graph $\mathcal{G}$, while $\nabla\mathcal{L}^*(\boldsymbol{Y}) = (\nabla\mathcal{L}_1^*(\boldsymbol{Y}_1), \dots, \nabla\mathcal{L}_n^*(\boldsymbol{Y}_n))^T$. We can now point out that

1. the dual gradients $\nabla\mathcal{L}_i^*(\boldsymbol{Y}_i)$ can be computed locally, and the variable $(\boldsymbol{Y})_{i\in[n]}$ can be stored on the nodes
2. the multiplication by CC^T corresponds to a gossip communication step.

It follows that Equation (2.16) is a gossip algorithm, in which the nodes hold the dual variable $(\boldsymbol{Y}_i^t)_{i\in[n]}$ locally, iteratively compute the dual gradients, which correspond to primal variables $\boldsymbol{X}_i^t \leftarrow \nabla\mathcal{L}_i^*(\boldsymbol{Y}_i^t)$, and then perform a gossip communication of the $\boldsymbol{X}_i^t$ and use them to update the $\boldsymbol{Y}_i^t$. Then, if $\boldsymbol{Y}^0 \in \text{span}(C) = \text{span}(\mathbf{1}_n)^\perp \times \mathbb{R}^d$, $\boldsymbol{X}^t$ converges to the solution of Equation (2.14) at a linear rate that depends on the condition number of the dual function $\lambda \to \mathcal{L}^*(\boldsymbol{C}\boldsymbol{\Lambda})$, which can be upper-bounded by κ/γ.

The dual approach thus allows to design *exact* algorithms in a principled way, by applying standard optimization scheme on the identified loss function. For instance, minimizing Equation (2.15) with an accelerated scheme allows to achieve an linear rate of $\mathcal{O}(\sqrt{\kappa/\gamma}\log(1/\varepsilon))$, which is optimal (Scaman et al., 2017). Such formulations of the optimization problem as a constrained optimization problem have been key to propose exact decentralized optimization algorithms with optimal rates (Scaman et al., 2017; Hendrikx et al., 2021; Kovalev et al., 2020). It is also convenient for understanding the behavior of gradient tracking algorithms (Jakovetić, 2018). Note also that the usage of dual gradients can be alleviated by modeling the optimization problem with additional virtual nodes (Hendrikx et al., 2020a) which uncouple the optimization from the gossip communication problem.

2.4 Adversarial Robustness

In this section, we first present some historical background on the terminology of Byzantine Robustness, which was introduced by the distributed computing community. Then, we present the standard setup of distributed ML robust to Byzantine parties along with

standard algorithms. This lead us to a discussion on the overlaps with the robust statistics literature.

The presentation of the basic approach to design Byzantine-robust algorithms in Section 2.4.2 partially overlaps with Chapter 6, which presents an abstraction to present Byzantine distributed optimization.

2.4.1 Historical Perspectives

The notion of Byzantine failure was introduced in 1980 by Leslie Lamport, Robert Shostak and Marshall Pease (Lamport et al., 1982) by comparing the problem of leading multiple computers to agree on a common value to the *problem of Byzantine generals*: a Byzantine army is besieging a city, and its generals must agree on a plan while communicating only orally with each other. Unfortunately – for the army – some of the generals are traitors, and try to prevent the loyal generals from agreeing on a good plan. The problem is thus to design an algorithm A such that the honest generals find a consensus by following this algorithm, while the traitors behave arbitrarily, for instance they can send incoherent information to the other generals. Curiously, in this early article, the Byzantine generals correspond to the pool of all computers, while, by a strange reversal of terminology, the *Byzantine* computers in later articles refer to the dishonest (or adversarial) computers, in opposition to the honest (or good) computers.

Robust Consensus. Various forms of consensus robust to Byzantine failure have been studied over the last 45 years for decentralized systems. All those forms of consensus are studied in a distributed setting, in which n different computers communicate through two-party (a.k.a. peer-to-peer) messages and among which at most f are Byzantine failures. Initially, all computers hold locally a parameter.

- **The Consensus Problem** aims at leading all honest computers to agree on the exact same parameter after a finite number of communications steps. Note that this value should be non-trivial, in the sense that it should correspond to one of the initial values of the honest computers. If algorithms exist to solve this problem when $3f + 1 \leq n$ and the communication is synchronous (Lamport et al., 1982), some level of synchronicity is always required (Fischer et al., 1985).

- **The Approximate Consensus Problem** is a relaxation of the consensus problem, and requires that honest computers achieve an approximate agreement, in the sense that their values are close to each other, and that respects a validity condition, which is that the nodes' parameters should stay in the *convex hull* of the initial honest values. This problem was shown to be more tolerant to asynchronicity (Dolev et al., 1986) than the exact consensus problem. This approximate consensus with the convex hull constraint was later studied for sparse communication networks (LeBlanc et al., 2013; Vaidya et al., 2012). Eventually, Vaidya (2014) show that, when the

parameters are multivariate of dimension d, approximate consensus is infeasible as soon as $n < (d+2)f + 1$.

More recently, the terminology of Byzantine resilience was introduced in the field of machine learning by Blanchard et al. (2017). As we show in the next section, when applied to machine learning, the objective and robustness condition are slightly different from those of the Approximate Consensus Problem.

2.4.2 The Machine Learning Setting

We now present the basics of Byzantine-robust distributed optimization in a synchronous federated setting. Consider a system of n devices. Among them, an unknown subset $\mathcal{B} \subset [n]$ is composed of $|\mathcal{B}| = f$ Byzantine devices. We denote the subset of honest devices $\mathcal{H} = [n] \backslash \mathcal{B}$. Consider that each device $i \in [n]$ holds a local dataset $\mathcal{D}_m^i$. Each device also stores the parameter of a model $\boldsymbol{x}_i$, and can compute the empirical loss of this model on its local dataset $\mathcal{L}_i(\boldsymbol{x}_i) = \frac{1}{|\mathcal{D}_m^i|} \sum_{\mathrm{z} \in \mathcal{D}_m^i} \ell(\mathrm{z}; \boldsymbol{x})$.

The Byzantine optimization problem writes

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} \left\{ \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}) := \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \mathcal{L}_i(\boldsymbol{x}) \right\}. \tag{2.17}$$

It is impossible in general to solve Problem 2.17 up to an arbitrary precision when $\mathcal{B}$ is unknown. This can be evidenced easily by considering the crucial sub-problem of *robust-averaging*.

The Robust Averaging Problem. Assume that all computers hold a parameter $\boldsymbol{x}_i$. The goal is to compute

$$\overline{\boldsymbol{x}}_{\mathcal{H}} := \frac{1}{\mathcal{H}} \sum_{i \in \mathcal{H}} \boldsymbol{x}_i = \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} \sum_{i \in \mathcal{H}} \|\boldsymbol{x} - \boldsymbol{x}_i\|^2. \tag{2.18}$$

Example 2.4.1. Consider a server-client setting with 4 clients, among which 1 is Byzantine. Assume that the clients declare the values $(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) = (-1, 0, 0, 1)$. Since the Byzantine client is unknown, the true mean is among $\overline{\boldsymbol{x}}_{\mathcal{H}} \in \{0, -1/3, 1/3\}$. Thus, the best the server can hope for is to achieve an error of $\|\overline{\boldsymbol{x}}_{\mathcal{H}} - \hat{\boldsymbol{x}}\| = 1/3$.

This simple example is the building block of more generic lower bounds on the achievable error of the Problem 2.18. Formalizing the lower bound requires, at first, to define a level of heterogeneity among the losses of the honest clients.

Assumption 2.4.2 ((G, B)-Heterogeneity)**.** We say that the loss functions $(\mathcal{L}_i)_{i \in \mathcal{H}}$ are (G, B)-Heterogeneous if, for any $\boldsymbol{x} \in \mathbb{R}^d$, it holds

$$\frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \|\nabla \mathcal{L}_i(\boldsymbol{x}) - \nabla \mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\|^2 \leq G^2 + B^2 \|\nabla \mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\|^2 \quad \text{where } \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}) := \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \mathcal{L}_i(\boldsymbol{x}).$$

Assumption 2.4.2 can be seen as a decomposition into a heterogeneity error at the optimum, and a term that measures the distance between the Hessians of the local loss functions.

Remark 2.4.3. Let $\mathcal{L}_{\mathcal{H}}$ be μ-strongly convex and each local loss be convex and L-smooth. Denote also $\Delta = \sup_{i\in\mathcal{H},\boldsymbol{x}\in\mathbb{R}^d} \|\nabla^2\mathcal{L}_i(\boldsymbol{x}) - \nabla^2\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\|_{op}$ the maximum distance in operator norm between the Hessian of the loss functions. Then, we have that (cf Allouah et al. (2024) and Chapter 6)

$$G^2 \leq \frac{2}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2 \quad \text{and} \quad B^2 \leq \min\left(2\left(\frac{L}{\mu} - 1\right), \frac{\Delta^2}{\mu^2}\right).$$

In the specific case of the averaging problem 2.18 we have that $B^2 = 0$ while G^2 is the variance among honest parameters $G^2 = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2$.

Assumption 2.4.2 is convenient as it allows to understand the achievable error for a given level of heterogeneity.

Theorem 2.4.4 (Allouah et al. (2024))**.** *For any distributed optimization algorithm A on Problem 2.17, there exist quadratic functions $\{\mathcal{L}_i\}$ that meet (G, B)-Heterogeneity and such that $\mathcal{L}_{\mathcal{H}}$ is μ-strongly convex, for which the output $\hat{\boldsymbol{x}} = A((\mathcal{L}_i)_{i\in[n]})$ of the algorithm is such that*

$$\mathcal{L}_{\mathcal{H}}(\hat{\boldsymbol{x}}) - \min_{\boldsymbol{x}\in\mathbb{R}^d} \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}) \geq \frac{G^2}{8\mu}\cdot\frac{n}{n-(2+B^2)f} \quad \text{and} \quad \|\nabla\mathcal{L}_{\mathcal{H}}(\hat{\boldsymbol{x}})\|^2 \geq \frac{G^2}{4}\frac{n}{n-(2+B^2)f}.$$

In particular, if $(2+B^2)f \geq n$, it is impossible to ensure that $\mathcal{L}(\hat{\boldsymbol{x}})$ is finite. This condition is also known as the breakdown point.

This lower bound already presents two remarkable properties:

1. In the limit case of merely convex loss functions, with $\mu = 0$, it is impossible to guarantee that the optimization converges to an area of small loss.

2. The breakdown point depends on the B^2 parameter: when the Hessians of the loss functions are different, the optimal breakdown point decreases. In the next section, we show that this property can also be understood as the condition under which Byzantine clients can force any "robust average" of the gradients $(\nabla\mathcal{L}_i(\boldsymbol{x}))_{i\in[n]}$ to be zero, and thus non-informative.

We now make this statement precise by clarifying what we mean by "robust average".

2.4.3 Robust Algorithms

All Byzantine-robust optimization methods rely at some point on a robust-averaging tool. Indeed, consider for simplicity the federated (server-client) setting. The distributed gradient descent framework would require that the server sends a parameter $\boldsymbol{x}^t$ to all clients. Each client computes a gradient $\nabla\mathcal{L}_i(\boldsymbol{x}^t)$, and the server updates the model parameter as

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta\frac{1}{n}\sum_{i\in[n]} \nabla\mathcal{L}_i(\boldsymbol{x}^t). \tag{2.19}$$

Unsurprisingly, the above update is non-robust to any adversarial client, since the average is non robust, and any adversarial client could declare $\nabla\mathcal{L}_i(\boldsymbol{x}^t) = -\sum_{j\neq i}\nabla\mathcal{L}_i(\boldsymbol{x}^t)$ in order to block the algorithm at its starting point. Alleviating this issue requires some form of robust averaging tool, which we define as follows:

Definition 2.4.5 ((f,ν)-robustness)**.** A function $F : \mathbb{R}^{n\times d} \to \mathbb{R}^d$ is said to be a (f,ν) robust averaging rule if, for any $\boldsymbol{x}_1,\dots,\boldsymbol{x}_n \in \mathbb{R}^d$, and any subset $S \subset [n]$ of size $|S| = n - f$, it holds that

$$\|F(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n) - \overline{\boldsymbol{x}}_S\|^2 \leq \frac{\nu}{n}\sum_{i\in S}\|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_S\|^2 \quad \text{where} \quad \overline{\boldsymbol{x}}_S := \frac{1}{S}\sum_{i\in S}\boldsymbol{x}_i.$$

The above definition is somewhat different from the robustness considered in robust statistics: here the guarantee holds deterministically, without assuming any underlying distribution for the input parameters $\{\boldsymbol{x}_i\}_{i\in[n]}$. We discuss further this point in Section 2.4.4.

Instances of Robust Averaging Rules. The notion of robust average has been a long topic of interest in the robust statistics literature. Coherently, most robust averaging rules are standard in robust statistics, even though studying them through the lens of the deterministic guarantee of Definition 2.4.5 could be more novel. We hereafter provide an overview of some standard averaging rules.

Aggregation	Huber	CwTM	Krum	GM	CwM	Lower Bound
ν	$\frac{2f}{n-3f}$	$\frac{6f}{n-2f}\left(1+\frac{6f}{n-2f}\right)$	$6\left(1+\frac{6f}{n-2f}\right)$	$4\left(1+\frac{f}{n-2f}\right)^2$	$4\left(1+\frac{f}{n-2f}\right)^2$	$\frac{f}{n-2f}$

Table 2.2: (f,ν)-Robustness of the coordinate-wise trimmed mean (CwTM), Krum (Blanchard et al., 2017), the geometric median (GM), the coordinate-wise median (CwM), and the multivariate Huber estimator of Chapter 6.

A desirable property of the cited averaging rules in Section 2.4.3 is that they only take as input an upper-bound on the number of adversaries f (or are even parameter-free for the median), and do not use distributional assumptions, such as the variance of the honest parameters $(\boldsymbol{x}_i)_{i\in\mathcal{H}}$.

From Section 2.4.3 we can essentially conclude that, for $f < O(n)$, there exist (f,ν)-robust estimators with $\nu = O(f/(n-2f))$: take for instance the Huber estimator, which, under $n > 4f$, is (f,ν)-robust with $\nu \leq {}^{4f}/_{n-2f}$.

From Robust Average to Inexact Gradients Considering a (f,ν)-robust average F, we can consider the robust alternative of Equation (2.19):

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta F(\{\nabla\mathcal{L}_i(\boldsymbol{x}^t)\}_{i\in[n]}). \tag{2.20}$$

This allows to highlight the strength of both Assumption 2.4.2 and Definition 2.4.5.

Remark 2.4.6 (cf. Chapter 6). Let Assumption 2.4.2 hold, and consider Equation (2.20) with F being (f, ν)-robust. Then Equation (2.20) is an inexact gradient descent step, where the inexact gradient oracle $\tilde{\nabla}\mathcal{L}(\boldsymbol{x}^t)$ is characterized as

$$\|\tilde{\nabla}\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^t) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^t)\|^2 \leq \nu G^2 + \nu B^2 \|\nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^t)\|^2.$$

The above remark is at the core of Chapter 6, and allows by itself to design various distributed first order methods, for instance by relying on existing analysis of inexact gradient descent (Ajalloeian and Stich, 2020).

Proposition 2.4.7 (Folklore, (Ajalloeian and Stich, 2020)). *Let $\mathcal{L}$ be μ-strongly convex and L-smooth, assume $|\mathcal{B}| \leq f$ and consider Equation (2.20), where F is (f, ν)-robust. Then it holds for $\eta = 1/L$*

$$\mathcal{L}(\boldsymbol{x}^t) - \mathcal{L}(\boldsymbol{x}^*) \leq (1 - \kappa^{-1}(1 - \nu B^2))^t (\mathcal{L}(\boldsymbol{x}^0) - \mathcal{L}(\boldsymbol{x}^*)) + \frac{\nu G^2}{\mu(1 - \nu B^2)}.$$

In particular, when $\nu = \frac{f}{n-2f}$, the asymptotic error of Proposition 2.4.7 matches Theorem 2.4.4 up to constants.

Beyond Deterministic Gradient Descent. When the clients sample *stochastic* oracles instead of gradients, using a (f, ν)-robust averaging rule with gradient descent is also sufficient to show that robust D-SGD converges to a neighborhood of the solution. However, when the *identities* of the clients are not taken into account, using only *permutation invariant* averaging rules can only yield a sub-optimal asymptotic error. The stochasticity of the clients indeed appears as a heterogeneity among the loss functions, and the asymptotic error is at least $\mathcal{L}(\hat{\boldsymbol{x}}) - \mathcal{L}(\boldsymbol{x}^*) \geq \mathcal{O}(\frac{G^2}{\mu}\frac{f}{n-2f})$ (Karimireddy et al., 2021).

Sketch of proof. Since the server does not take into account the identity of the clients, it can not distinguish between the following two situations:

- At each iteration, the server sample the stochastic gradients (g_i^t), such that $\mathbb{E}[\|g_i^t - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^t)\|^2] \approx \sigma^2$,
- At each iteration, the server sample gradients from n new clients (denoted $\mathcal{N}_t = \mathcal{H}_t \cup \mathcal{B}_t$), whose loss satisfy $\mathbb{E}_{i\in\mathcal{H}_t}\|\nabla\mathcal{L}_i(\boldsymbol{x}^t) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^t)\|^2 \approx \sigma^2$.

In the second situation, Theorem 6.2.3 with $B = 0$ and $G^2 = \sigma^2$ yields the lower bound. □

Alleviating this issue is possible, either by increasing the batch size, or, similarly, by relying on moving average of the stochastic oracles. For instance Karimireddy et al. (2021); Farhadkhani et al. (2022) uses Polyak-momentum – which can be seen as an exponential moving average of the gradient – before averaging robustly to achieve an optimal asymptotic error, i.e. independent of σ^2. Quite surprisingly, using this exponential average of the

stochastic oracles does not degrade the convergence rate compared to D-SGD, at least asymptotically.

2.4.4 Links with the Robust Statistics Literature

In this section, we try to take a step back from the field of Byzantine optimization by positioning it within the vast literature of robust statistics. Robust statistics have been a field of research at least since the seminal work of Huber (1964), with a renewal of interest in the high-dimensional setting this last decade, as evidenced by the last Gödel Prize awarded to Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra and Alistair Stewart for their paper on computationally efficient high-dimensional robust mean estimation (Diakonikolas et al., 2019).

The most basic task in robust statistics corresponds to estimating the expectation of a distribution $\mathcal{D}$ given n "corrupted" samples, where the corruption can be of increasing strength:

- **Weak (or Huber) contamination**. The Huber model of corruption consists in assuming that the n samples are independent identically distributed from $\tilde{\mathcal{D}} = (1-\alpha)\mathcal{D} + \alpha\mathcal{D}_{\text{adv}}$, where $\mathcal{D}_{\text{adv}}$ is an adversarially crafted distribution, and $\alpha \in [0, 1/2]$.
- **Strong contamination**. In the strong model of corruption, n samples $\mathcal{D}_n = (x_1, \dots, x_n)$ are drawn i.i.d. from the distribution $\mathcal{D}$, then given to an adversary which can change up to $n\alpha$ samples to create a dataset $\mathcal{D}'$, by choosing both which sample to replace, and the replacement values, subject only to the constraint $d_H(\mathcal{D}_n, \mathcal{D}'_n) \leq n\alpha$.

The Byzantine corruption problem is the moral equivalent to the strong contamination model. However, a key distinction remains, since allowing a client to send arbitrary gradients at each iteration may be strictly stronger than allowing an honest client to have adversarially corrupted data. If the lower bound Theorem 2.4.4 and the upper bound Proposition 2.4.7 show that poisoning is as strong as Byzantine corruption when considering the optimization error, Boudou et al. (2025) argue, by considering the uniform stability of the optimization algorithm, that Byzantine corruption can be strictly worse than poisoning, in particular in terms of generalization.

Dimensional Tightness. The search for dimensionally-tight and computationally efficient robust mean estimators has received a lot of attention over the last decade. It is worth mentioning that, in contrast to the approach taken in Definition 2.4.5, which hides the dimensional dependence of the error in the $1/n \sum_{i \in \mathcal{H}} \|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2$ term, a fine characterization of the impact of the dimension is achievable. In mean estimation problems, the impact of the dimension happens throughout the covariance matrix Σ of the distribution. Two regimes of accuracy occur: if the underlying distribution is subgaussian, then the achievable accuracy (with high probability) grows as (Diakonikolas et al., 2019)

$$\|\hat{\mu} - \mu\|^2 = \tilde{\mathcal{O}}\left(\frac{\operatorname{tr}\Sigma}{n} + \alpha\|\Sigma\|_2\right), \tag{2.21}$$

while for more heavy tailed distribution with only a covariance matrix Σ, the achievable accuracy grows as (Lugosi and Mendelson, 2021)

$$\|\hat{\mu} - \mu\|^2 = \tilde{\mathcal{O}}\left(\frac{\operatorname{tr}\Sigma}{n} + \alpha^2\|\Sigma\|_2\right). \tag{2.22}$$

Importantly, in both results, the $\frac{\operatorname{tr}\Sigma}{n}$ factor is the error of the adversary-free mean estimation problem. While it can grow linearly with the dimension when the covariance matrix is isotropic, it also captures the potential low-dimensional structure of the problem. Thus, the impact of adversaries happens only in the second term, which is independent of the dimension. This contrasts with the guarantees of (f, ν)-robust aggregators, in which the error scales at best as $\frac{f}{n}\frac{1}{n}\sum_i \|\boldsymbol{x}_i - \overline{\boldsymbol{x}}\|^2 \approx \eta \operatorname{tr}(\Sigma)$.

A Deterministic Perspective. The assumptions made in Byzantine optimization, such as the one in the definition of (f, ν)-robustness, are deterministic. In contrast, most papers in robust statistics often assume that the data stems from an underlying distribution. Relying on a deterministic characterization of the heterogeneity can allow for more flexibility when dealing with highly heterogeneous settings, as it captures at the same time the heterogeneity due to distribution shifts among clients as well as the statistical heterogeneity due only to a finite number of i.i.d. samples among clients. Correlatively, designing robust mean estimators similar to the (f, ν)-robustness with tighter dependence on the dimension is possible (Allouah et al., 2025b). For such methods, the dependency on the proportion of adversaries is similar to the heavy-tail setting Equation (2.22).

2.5 Privacy

This section introduces the field of privacy. This concept primarily refers to societal and philosophical considerations, which the mathematical concept of differential privacy studied in this dissertation sheds technical light on, without, however, seeking to replace them. To go beyond this brief introduction, the reader can explore the seminal book of Dwork and Roth (2014), the privacy page of the Stanford Encyclopedia of Philosophy (Roessler and DeCew, 2023), the pedagogical blog-post of Damien Desfontaines (Desfontaines, 2026) or the nice introduction made by Cyffers (2024).

2.5.1 Why Differential Privacy?

As defined by Dwork and Roth (2014), privacy in the context of algorithmic data manipulation corresponds to a promise, made by an analyst to a data subject, that they will not be harmed as a consequence of sharing some personal information.

Early attempts to guarantee the privacy of the data subjects relied on methods such as *data-anonymization*, i.e. removing information that allows one to link the database entry to the associated subject. Unfortunately, this approach is insufficient to guarantee the privacy of the information, as some apparently benign information can be coupled with publicly available information to re-identify the data.

One example of such a failure is the Netflix Prize dataset (Narayanan and Shmatikov, 2008). The Netflix Prize dataset is made of movie ratings created by subscribers, where most identifying information was removed, but not all. In particular, correlating the date on which each movie was watched with publicly available reviews of movies on the Internet Movie Database allowed 99% of the records to be re-identified. Part of the watching history of some participants was not available in any publicly available database, and this leak could reveal sensitive information such as the sexual orientation of the data subject.

Another approach to guarantee privacy while disclosing analyses is to restrict the publicly available results to summary statistics on aggregated samples. One could think that, as long as each summary statistic is computed on enough people, re-identifying them would be impossible. However, the more summary statistics are published, the more this guarantee vanishes, and data may be reconstructed. In 2010, the US Census Bureau published its decennial study on the population and housing in the United States, with privacy protection based essentially on summary statistics and random swapping of data records. Yet, those protections were insufficient for protecting privacy, as the personal records of every resident in 70% of the census blocks could be exactly reconstructed using public data only (Abowd et al., 2023). This catastrophic failure – the census must protect the privacy of participants by law – led the Census Bureau to rely on differential privacy in the 2020 study.

Similarly to summary statistics, machine learning models are vulnerable to privacy leaks. This has been evidenced regularly over the last decades (Bun et al., 2014; Dwork et al., 2015; Shokri et al., 2017). The rise of large machine learning models only broadens the vulnerabilities: large language models are prone to reconstruction attacks, in which input training data can be reconstructed (Carlini et al., 2021; Nasr et al., 2023), and larger models only memorize more data. Importantly, memorization is not a bug of badly designed models, but a default behavior of complex models (Brown et al., 2021a). Ad-hoc privacy protections are still regularly proposed and broken, as evidenced by the privacy leak in Anthropic's Clio model, which relied on ad-hoc privacy guarantees (Sundaram Muthu Selva Annamalai et al., 2026).

In contrast to ad-hoc protection, the notion of Differential Privacy is a formal mathematical guarantee agnostic of the attack. Over the last decades, it has become the standard criterion for data privacy in statistics and machine learning, as it proposes a principled way to design secure and accurate analytic methods.

2.5.2 Formal Guarantee

Differential Privacy is a mathematical definition. Given a database, a notion of individual records, and an algorithm that takes the database as input, differential privacy corresponds, informally, to the guarantee that changing one record in the database does not drastically change the output of the algorithm.

Setup Let $\mathcal{D}$ be a dataset consisting of n records. One would like to query the dataset, i.e. to use a function f that takes the dataset as input and outputs a quantity of interest. This could correspond to outputting summary statistics such as the mean of the records, a whole dataset or a whole machine learning model. Then a private mechanism $\mathcal{M}$ is a pipeline that approximates the query f by adding noise to the process, such that the output $\mathcal{M}(\mathcal{D})$ is *statistically indistinguishable* from that of the mechanism $\mathcal{M}$ applied on a *neighboring dataset*, i.e. a dataset $\mathcal{D}'$ identical to the dataset $\mathcal{D}$ on all records except one.

Definition 2.5.1 (Differential Privacy)**.** Let $\mathcal{M}$ be a random mapping, $\varepsilon > 0$ and $\delta \in [0, 1]$. If $\mathcal{M}$ is (ε, δ)-differentially private, then for any input dataset $\mathcal{D}_n$ of n entries, and any neighboring dataset $\mathcal{D}'_n$, in the sense that the Hamming distance between $\mathcal{D}$ and $\mathcal{D}'$ is $d_H(\mathcal{D}, \mathcal{D}') = 1$, and any measurable set S

$$\Pr[\mathcal{M}(\mathcal{D}_n) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(\mathcal{D}'_n) \in S] + \delta.$$

The case $\delta = 0$ is commonly referred to as *pure* differential privacy, while $\delta > 0$ refers to *approximate* differential privacy.

The variable ε is the *privacy budget*, while δ corresponds to a catastrophic privacy leak probability, and must be significantly smaller than $1/n$ in order to have any significant privacy guarantee. Indeed: consider the task of computing the average of the n entries, then the "name and shame" algorithm which uniformly outputs one of the entries is $(0, 1/n)$-differentially private, although it always leaks one of the input data records.

Notion of Individual Records. The notion of individual records can vary depending on the context: it defines the unit of information on which the guarantee applies. For instance, in machine learning, it can either correspond to a single data entry, or to all the information related to an individual. In a distributed context, a client-level notion of record is typically considered, which corresponds to considering as 'record' the whole dataset locally stored by one of the clients.

Output of the Mechanism. The output of the mechanism defines what is actually accessible to an adversary, and what can then be used to infer information about the individuals. When computing simple analytics, such as the mean of the dataset, it corresponds to the noisy mean. In a distributed context, the output of the mechanism does not always correspond only to the last iterate of the optimization algorithm, depending on the trust assumptions on the clients and the server. We instantiate two classical settings in a federated learning context:

- In *Central Differential Privacy*, the server (and usually the other clients) are assumed to be trusted, and the adversary can only see the final output of the learning process. In some cases, only the server is trusted, while the other clients are honest-but-curious, and can leak information.

- In *Local Differential Privacy*, neither the server nor the other clients are trusted. It follows that the output of the mechanism corresponds to the transcript of all shared information among computers, and not only to the last iterate of the optimization pipeline.

Post-processing. The strength of differential privacy relies on its robustness to post-processing: any post-processing of the output of an (ε, δ)-differentially private algorithm is still (ε, δ)-differentially private.

Proposition 2.5.2 (post-processing)**.** *Let $\mathcal{M} : \mathcal{N}^{\mathcal{X}} \to \mathbb{R}^d$ be a randomized algorithm that is (ε, δ)-differentially private, then for any randomized mapping $f : \mathbb{R}^d \to \mathbb{R}^k$, $f\mathcal{M} : \mathcal{N}^{\mathcal{X}} \to \mathbb{R}^k$ is (ε, δ)-differentially private.*

More complex differentially private algorithms can also be designed by combining them.

Proposition 2.5.3 (Composition (Dwork et al., 2006, 2010; Kairouz et al., 2015))**.** *Consider $\mathcal{M}_1, \dots, \mathcal{M}_T$ $(\varepsilon_i, \delta_i)_{i\in[T]}$ DP algorithms. Consider the mechanism $\mathcal{M}(\mathcal{D}) = \mathcal{M}_T(\mathcal{D}, \mathcal{M}_{T-1}(\mathcal{D}, \dots, \mathcal{M}_2(\mathcal{D}, \mathcal{M}_1(\mathcal{D}))))$. Then*

1. *(Basic Composition) $\mathcal{M}$ is (ε, δ) DP with $\varepsilon = \sum_{t=1}^T \varepsilon_t$ and $\delta = \sum_{t=1}^T \delta_t$.*
2. *(Advanced Composition) $\mathcal{M}$ is $(\tilde{\varepsilon}_{\tilde{\delta}}, \tilde{\delta} + \sum_{t=1}^T \delta_t)$-DP where*

$$\tilde{\varepsilon} := \sum_{t=1}^T \frac{(e^{\varepsilon_t} - 1)\varepsilon_t}{e^{\varepsilon_t} + 1} + \sqrt{\sum_{t=1}^T \varepsilon_t^2 \log(1/\tilde{\delta})}.$$

If these results guarantee some level of differential privacy, it is known that the composition guarantees based on the (ε, δ)-definition of differential privacy are not tight. Other definitions of differential privacy, such as Rényi Differential Privacy (Mironov, 2017) (RDP), which relies on the Rényi Divergence, allow to tightly capture the privacy loss due to composition. Consequently, it is sometimes preferred over the (ε, δ) definition.

2.5.3 Differentially Private Algorithms

Although the privacy contributions in this manuscript mostly focus on impossibility results, rather than a methodological approach, we present some basic private algorithms, as it brings some perspective on our contributions.

Differentially private algorithms rely on some form of randomness to add noise to their output. To guarantee privacy, the noise scale must correspond to the *sensitivity* of the algorithm.

Definition 2.5.4 (Sensitivity)**.** Consider a function $f : \mathcal{X}^n \to \mathbb{R}^d$. Then the ℓ^p

sensitivity of the function f is defined as

$$\Delta_p := \sup_{\mathcal{D}\sim\mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\|_p.$$

The most basic private mechanism adds Laplace noise calibrated to the ℓ^1 sensitivity of the function f

Proposition 2.5.5 (Laplace Mechanism (Dwork et al., 2006)). *Consider a function $f : \mathcal{X}^n \to \mathbb{R}^d$. Then the Laplace mechanism writes*

$$\mathcal{M}(\mathcal{D}) = f(\mathcal{D}) + \frac{\Delta_1}{\varepsilon} Z, \quad \text{where } Z \text{ is of density} \quad p_Z(w) = \frac{1}{2^d}\exp(-\|w\|_1)$$

and is $(\varepsilon, 0)$-differentially private.

Another widely used mechanism is the Gaussian Mechanism, which, however, only satisfies approximate DP.

Definition 2.5.6 (Gaussian Mechanism (Dwork et al., 2006)). Consider the function $f = \mathcal{X}^n \to \mathbb{R}^d$ with l_2-sensitivity of Δ_2.

Then, the Gaussian mechanism, given by

$$\mathcal{M} : \mathcal{D}_n \longmapsto f(\mathcal{D}_n) + \mathcal{N}\left(0, \frac{2\Delta_2^2 \log(1.25/\delta)}{\varepsilon^2}\mathbb{I}_d\right)$$

is (ε, δ)-differentially private.

Alternative variants of differential privacy were also proposed to precisely capture the properties of the Gaussian mechanism. These variants are called *concentrated DP* (Dwork and Rothblum, 2016; Bun and Steinke, 2016) and *Gaussian DP* (Dong et al., 2019).

While the Laplace Mechanism enjoys the desirable property of achieving pure differential privacy, this is also over constraining in many settings, as pure DP is often achieved at the cost of a significant accuracy degradation compared to approximate DP.

Example 2.5.7 (Multivariate Private Mean Estimation). Consider i.i.d. random vectors $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ with $\|\mathbf{x}_i\|_2 \leq r$ almost surely. Then, the clipped sampled mean $f(\mathbf{X}) = \frac{1}{n}\text{clip}(\mathbf{x}_i; r)$ has ℓ^2 sensitivity $\Delta_2 = \frac{2r}{n}$. So the Gaussian mechanism yields

$$\mathbb{E}[\|\mathcal{M}_{\mathcal{N}}(\mathbf{X}) - \overline{\mathbf{x}}\|_2^2] = \mathbb{E}[\|\mathcal{N}\left(0, \frac{2\Delta_2^2 \log(1.25/\delta)}{\varepsilon^2}\,\mathrm{I}\right)\|_2^2] = \frac{2\log(1.25/\delta) r^2 d}{\varepsilon^2 n^2},$$

And thus $\mathbb{E}[\|\mathcal{M}_{\mathcal{N}}(\mathbf{X}) - \mathbb{E}[\mathbf{x}_i]\|_2^2] \leq \frac{r^2}{n} + \frac{2\log(1.25/\delta) r^2 d}{\varepsilon^2 n^2}$.

While this private mean estimator is often used in practice, it already raises a few questions:

1. How should we set the parameter r, since it impacts the accuracy of the estimation, while estimating it on the data might require using some data samples and some privacy budget?

2. The clipping operation is performed around the origin, and the mechanism is thus not translation-invariant.

3. Is the use of an isotropic covariance in the Gaussian noise optimal? It may obfuscate an underlying low-dimensional structure of the sample distribution, while adding an extra d factor in the squared error.

As mean estimation is a key building block of many algorithms, a significant body of work has been developed to design optimal private mean estimators and understand the fundamental limits of the task. The (sub-)Gaussian setting has seen particular interest, as it presents a simple problem of what can be achieved under favorable assumptions. In the univariate Gaussian setting, (Karwa and Vadhan, 2018) propose an optimal (ε, δ) mean-estimation algorithm that works even without assumptions on the localization of the expectation or the variance of the distribution. Note that this is only possible using approximate differential privacy, as (Hardt and Talwar, 2010; Bun et al., 2019) show that the sample complexity of $(\varepsilon, 0)$ private mean estimation grows as the logarithm of the diameter of the set of feasible expectations. Another line of work (Bun et al., 2019; Kamath et al., 2019; Aden-Ali et al., 2021; Liu et al., 2021; Brown et al., 2021b; Liu et al., 2022) focuses on designing estimators that are invariant to the choice of Euclidean norm, by considering the error using the *Mahalanobis* norm, i.e. $\|x\|^2_{\mathbf{\Sigma}^{-1}} = x^T \mathbf{\Sigma}^{-1} x$, where $\mathbf{\Sigma}$ is the covariance of the distribution. However, achieving a non-trivial error in the Mahalanobis norm requires at least d samples when the covariance matrix is unknown, since learning the covariance matrix is necessary to avoid adding noise in eigenspaces in the kernel of the covariance matrix. While, when the covariance matrix is diagonal, the optimal error of private mean estimation grows as $\alpha^2 = \theta\left(\frac{d}{n} + \frac{d^2}{n^2\varepsilon^2}\right)$, having a non-diagonal covariance matrix could allow for a better error with dependence on intrinsic-dimension-like quantities. For instance, Dagan et al. (2024) shows that knowing the covariance matrix allows one to design algorithms with ℓ^2 error $\alpha^2 \in \mathcal{O}(\frac{\operatorname{tr}\mathbf{\Sigma}}{n} + \frac{\left(\operatorname{tr}\sqrt{\mathbf{\Sigma}}\right)^2}{n^2\varepsilon^2})$. In Chapter 7, we investigate whether such rates can be obtained when the covariance matrix is unknown, and give a negative answer to the question.

Chapter 3

Technical Summary of the Contributions

In this chapter, we formally summarize the contributions of this thesis.

Contributions' summary of Chapter 4

We study the problem of Byzantine-robust decentralized optimization, with a particular focus on synchronous gossip communication. Our contribution is twofold:

Unified analysis We provide a general method to design robust gossip algorithms. The proposed approach consists, given a robust averaging rule F, in updating the parameters of honest nodes as:

$$\forall i \in \mathcal{H}, \quad \boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t - \eta F((\boldsymbol{x}_i - \boldsymbol{x}_j)_{j \in \text{neighbors}(i)}).$$

The above update is robust – in the sense that the mean square error decreases strictly – as soon as the number of Byzantine neighbors to honest node $b = \sup_{i \in \mathcal{H}} |\text{neighbors}(i) \cap \mathcal{B}|$ is smaller than half the smallest non-zero eigenvalue of the (un-normalized) Laplacian matrix of the honest sub-graph, i.e. $b < \lambda_2(\boldsymbol{W}_{\mathcal{H}})/2$. By denoting γ the spectral gap of the honest subgraph, and the breakdown assumption as $\delta := 2\rho b/\lambda_2(\boldsymbol{W}_{\mathcal{H}}) < 1$, where $\rho \geq 1$ measures the quality of the averaging rule F, we show that the iterates $\{\boldsymbol{x}_i^t; i \in \mathcal{H}, t \geq 0\}$ of the method satisfies

$$\frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \|\boldsymbol{x}_i^t - \overline{\boldsymbol{x}}_{\mathcal{H}}^t\|^2 \leq (1 - \gamma(1-\delta))^t \, \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \|\boldsymbol{x}_i^0 - \overline{\boldsymbol{x}}_{\mathcal{H}}^0\|^2$$

$$\|\overline{\boldsymbol{x}}_{\mathcal{H}}^t - \overline{\boldsymbol{x}}_{\mathcal{H}}^0\|^2 \leq \frac{4\delta}{\gamma(1-\delta)^2} \, \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \|\boldsymbol{x}_i^0 - \overline{\boldsymbol{x}}_{\mathcal{H}}^0\|^2.$$

It follows that honest nodes achieve an asymptotic consensus, while their average parameter has a bounded bias. This result provides a solid groundwork for the design of robust decentralized optimization algorithms, as we illustrate in the case of decentralized-SGD.

Spectral Tightness of the Breakdown Point. We show that the condition on the breakdown point (i.e., $b < \lambda_2(\boldsymbol{W}_{\mathcal{H}})/2$) is tight, in the sense that no better condition com-

paring the number of Byzantine neighbors per honest node b with eigenvalues of the graph can be established. Indeed, we show that there exist arbitrarily large and sparse graphs (in the sense that γ is small), for which $b = \lambda_{\max}(\mathcal{G}_{\mathcal{H}})/2$, and on which no communication algorithm can be robust, in the sense that the mean square distance to the initial mean paramter strictly decreases.

Attacks tailored to the decentralized setting. We stress the robust gossip communication by proposing an fully-decentralized attack to disrupt the communication. This attacks works by partitioning the graph in two clusters defined by the Fiedler vector of the graph, i.e. the eigenvector of the Laplacian matrix associated with the second smallest eigenvalue. Then, Byzantine nodes push honest ones into one or another direction depending on which cluster they are assigned to. This attack is shown to disrupt existing robust methods using less adversaries than previous approaches.

Contributions' summary of Chapter 5

We investigate the feasibility of the dual approach to design Byzantine-robust decentralized optimization methods. Using the notations of Section 2.3.3.3, we show that the most efficient methods of the framework of Chapter 4 can be seen as modifying the gradient in the gradient descent on the dual objective function Equation (2.15). Namely, that robust averaging methods can be seen as a control on the dual gradient in the gradient descent on the dual objective function.

However, some form of symmetry in the update between neighbors must be respected to be fully compatible with the dual approach. Followingly, we study with a dual perspective a communication algorithm based on a global clipping threshold. We show that, if the threshold is set according to

$$\sum_{i,j\in\mathcal{H},\ i\sim j} \|\boldsymbol{x}_i^t - \boldsymbol{x}_j^t\|^2 1_{\|\boldsymbol{x}_i^t-\boldsymbol{x}_j^t\|^2>\tau^t} \geq \Delta_\infty \sum_{i,j\in\mathcal{H},\ i\sim j} \|\boldsymbol{x}_i^t - \boldsymbol{x}_j^t\|^2 1_{\|\boldsymbol{x}_i^t-\boldsymbol{x}_j^t\|^2\geq\tau^t} + \eta b^2 \mathcal{H} \sum_{s=0}^{t} \tau^s,$$

where η is a step-size, and Δ_∞ a graph-related quantity that measures the influence of the Byzantine nodes on the communication, then the mean square error evolves as

$$\sum_{i\in\mathcal{H}} \|\boldsymbol{x}_i^{t+1} - \overline{\boldsymbol{x}}_{\mathcal{H}}^0\|^2 \leq \sum_{i\in\mathcal{H}} \|\boldsymbol{x}_i^t - \overline{\boldsymbol{x}}_{\mathcal{H}}^0\|^2 - \eta \sum_{i,j\in\mathcal{H},\ i\sim j} \|\boldsymbol{x}_i^t - \boldsymbol{x}_j^t\|^2 1_{\|\boldsymbol{x}_i^t-\boldsymbol{x}_j^t\|^2\leq\tau^t}.$$

If this ensures the decrease of the MSE, it does not provides a rate or some asymptotic consensus. Furthermore, the analysis is restricted to the squared norm loss function.

To extend this analysis, we phrase local-clipping methods in the dual perspective. If this allows to understand local aggregation methods as preconditioned clipped gradient descent, it is insufficient to overcome the challenge that Byzantine pose in the dual approach.

This leads us to discussing the major technical difficulty towards leveraging the dual approach to design Byzantine-robust decentralized algorithms: introducing Byzantine nodes breaks the key invariant of the dual perspective. Furthermore, while the strength

of the dual perspective usually relies on simple Lyapunov functions, this loss of invariant also breaks the usual Lyapunov function, thus restricting the interest of the methodology.

Contributions' summary of Chapter 6

Generic formalism. In this chapter, following Section 2.4.3, we explicit that Byzantine-robust distributed optimization exactly fits the formalism of first-order methods with inexact gradients oracle, with a simple additive and multiplicative error term, as

$$\|\tilde{\nabla}\mathcal{L}(\boldsymbol{x}) - \nabla\mathcal{L}(\boldsymbol{x})\|^2 \leq \zeta^2 + \alpha\|\nabla\mathcal{L}(\boldsymbol{x})\|^2,$$

where $\tilde{\nabla}\mathcal{L}$ is the inexact gradient oracle. To do so, we simply combine the standard (G, B)-Heterogeneity assumption with existing definitions of robust aggregation rules.

This allows us to directly 1) analyze Byzantine-robust distributed (stochastic) gradient descent using previous result on inexact gradient methods, and 2) use existing work on inexact gradients to show a first accelerated algorithm, although this acceleration holds for a restricted heterogeneity condition, and with a sub-optimal asymptotic error.

These first results both motivate and pave the way for a more systematic search for efficient algorithms, abstracting away the complexity of the Byzantine setting, and thus building on the inexact gradient formulation to propose a robust version of two standard fast algorithms.

Optimization under similarity. When an approximation $\hat{\mathcal{L}}$ (in the second-order sense) of the global honest loss is available, we give in a Byzantine-robust algorithm (with computationally more expensive updates than GD) called Prox Inexact Gradient method under Similarity (PIGS), which writes

$$\boldsymbol{x}^{t+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \left\{ \hat{\mathcal{L}}(\boldsymbol{x}) + \frac{1}{\eta}\|\boldsymbol{x} - \boldsymbol{x}^t\|^2 + (\tilde{\nabla}\mathcal{L}(\boldsymbol{x}^t) - \nabla\hat{\mathcal{L}}(\boldsymbol{x}^t))^T\boldsymbol{x} \right\}.$$

We show that PIGS enjoys a linear convergence rate of $\mathcal{O}(\Delta/\mu \log(1/\varepsilon))$ to reach an ε neighborhood of the optimal error, where Δ corresponds the the operator norm of the difference between the Hessian of the approximate loss and of the true honest one. Since Δ can be drastically smaller than the smoothness of the loss function, this results in an important gain in iteration complexity, which consequently reduces the communication time. The convergence theory of PIGS requires an extension of Woodworth et al. (2023) to handle the inexactness of the error term.

Acceleration. We adapt the extra-gradient method from (Kovalev et al., 2022) to the inexact oracle setting. We show that it converges in this setting towards the *optimal asymptotic error* with an *accelerated linear rate* of convergence $\mathcal{O}(\sqrt{L/\mu}\log(1/\varepsilon))$, where μ is the strong convexity of the considered loss function and L the smoothness. This result holds as long as the factor α in the oracle's relative error is smaller than $\sqrt{L/\mu}$. Our result outperforms previous works on accelerated inexact oracle first-order methods by a $\sqrt{L/\mu}$ factor both in the asymptotic error and in the condition on the relative error.

Additionally, the extra-gradient algorithm also enjoys an accelerated rate of convergence in the similarity setting, but only as long as the relative error in the gradient is small with respect to Δ^2/L^2.

Contributions' summary of Chapter 7

We study the problem of differentially private mean estimation of gaussian vectors, where data samples are distributed according to $\mathbf{x}^1, \dots, \mathbf{x}^n \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. When the covariance matrix of the data distribution is isotropic, i.e. $\boldsymbol{\Sigma} \propto \mathrm{I}_d$, it is well known that private mean estimators suffer from a dimensional overhead compared to non-private ones, as their accuracy typically scales as $\alpha^2 \leq \tilde{\mathcal{O}}\left(\frac{\mathrm{tr}(\boldsymbol{\Sigma})}{n} + \frac{d^2}{n^2\varepsilon^2}\right)$. Recently, Dagan et al. (2024) have shown that, when the covariance matrix was known, one could design a private mean estimator whose accuracy depends on the *effective dimension* of the problem, and not on the ambient dimension d. Formally, they propose an (ε, δ)-DP algorithm $\mathcal{A}$ which, when given the covariance matrix as input, enjoys the accuracy $\|\mathcal{A}(\mathbf{X}, \boldsymbol{\Sigma}) - \boldsymbol{\mu}\|^2 = \alpha^2 = \tilde{\mathcal{O}}\left(\frac{\mathrm{tr}(\boldsymbol{\Sigma})}{n} + \frac{\left(\mathrm{tr}\sqrt{\boldsymbol{\Sigma}}\right)^2}{n^2\varepsilon^2}\right)$ with high probability. This rate is also shown to be optimal on gaussian mean estimation.

Such dimension-free rate is important, as real-world data are often far from isotropic, with signals concentrated on a small number of principal components. That private mean estimation usually adds isotropic noise can seriously degrade the performance of downstream tasks, as evidenced when implementing differentially-private gradient descent on machine learning problems.

However, the covariance matrix of the data samples is rarely known in advance. To alleviate this issue, (Dagan et al., 2024) also propose a covariance-agnostic private mean estimator which, however, only achieves an accuracy of

$$\alpha^2 = \tilde{\mathcal{O}}\left(\frac{\mathrm{tr}(\boldsymbol{\Sigma})}{n} + \frac{\left(\sum_{i\in[d]}\sqrt{\boldsymbol{\Sigma}_{ii}}\right)^2}{n^2\varepsilon^2} + \frac{d\sum_{i\in[d]}\sqrt{\boldsymbol{\Sigma}_{ii}}}{n^4\varepsilon^4}\right),$$

which still depends on the dimension.

Question. *Can we design a private mean estimator for gaussian distribution, whose accuracy is independent of the ambient dimension d, but only of the effective-dimension-like quantities $\mathrm{tr}(\boldsymbol{\Sigma})/\|\boldsymbol{\Sigma}\|_2$ and $(\mathrm{tr}\sqrt{\boldsymbol{\Sigma}})^2/\|\boldsymbol{\Sigma}\|_2$?*

We answer this question negatively by comparing two complementary settings: a non-diagonal covariance setting, in which the covariance matrix is a random projector on a low-dimensional subspace, and a diagonal setting, in which the distribution on covariance matrices mimics a polynomial decrease in the eigenvalues, as $\boldsymbol{\Sigma}_{ii} \propto 1/i$.

Formally, we define the optimal ℓ_2 error with probability $\frac{2}{3}$ of (ε, δ)-DP algorithms for a given prior $\mathcal{P}$ distribution on the covariance matrices as

$$\mathfrak{R}_{\varepsilon,\delta}(\mathcal{P}, n) := \inf \left\{ \alpha^2 \geq 0 : \exists \mathcal{A} \ \ (\varepsilon, \delta)\text{-DP s.t.} \ \Pr_{\substack{\boldsymbol{\Sigma}\sim\mathcal{P}, \boldsymbol{\mu}\sim\mathcal{N}(0,\rho\,\mathrm{I}_d) \\ \mathbf{X}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})^{\otimes n}}} \left[\|\mathcal{A}(\mathbf{X}) - \boldsymbol{\mu}\|_2^2 \leq \alpha^2 \right] \geq \frac{2}{3} \right\},$$

where $\rho > 0$ is taken sufficiently large. Similarly we denote as $\widetilde{\mathfrak{R}}_{\varepsilon,\delta}$ the optimal error for algorithms that take both the dataset $\mathbf{X}$ and the covariance matrix $\boldsymbol{\Sigma}$ as input.

Non-Diagonal Setting. For $\mathcal{P}_1$ the uniform distribution on rank-k orthogonal projector of $\mathbb{R}^d$, we show that if $n \leq k$ and $\delta \leq \mathcal{O}(1/d)$, then (ε, δ) differentially private algorithms that do not take the covariance matrix as input suffer from a multiplicative overhead in their error of $\frac{d}{k}$, i.e.

$$\mathfrak{R}_{\varepsilon,\delta}(\mathcal{P}_1, n) \geq \frac{d}{k} \widetilde{\mathfrak{R}}_{\varepsilon,\delta}(\mathcal{P}_2, n).$$

Furthermore, this result shows that the naive algorithm which estimates the trace of the dataset and adds isotropic noise to the average of the (filtered) data samples is optimal in this low-dimensional regime of $n \leq k$.

Diagonal Setting. To go beyond the setting of n smaller than the effective dimension of the problem, we take as prior distribution that $\boldsymbol{\Sigma} = \mathrm{Diag}(\boldsymbol{\sigma})$, where $(\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_d) \sim \mathsf{inv}\chi^2(1,1)^{\otimes d}$, i.e. $1/\sigma_i$ follows a $\chi^2(1)$ distribution. For this prior distribution, if $\log(d) \lesssim n \lesssim d^{1/8.5}$ and $\delta \lesssim 1/n$, we have that

$$\mathfrak{R}_{\varepsilon,\delta}(\mathcal{P}_1, n) \geq \frac{\sqrt{d}}{n^{8.5}} \widetilde{\mathfrak{R}}_{\varepsilon,\delta}(\mathcal{P}_2, n).$$

Although the exact polynomial dependence on d and n might not be tight, this result shows that even when the intrinsic dimension is such that $\mathrm{tr}\sqrt{\boldsymbol{\Sigma}}/\|\boldsymbol{\Sigma}\|_2 = \Theta(\log(d))$ with high probability, achieving the covariance-aware optimal error when the covariance is unknown requires a number of samples which is a power of the ambient dimension.

Proof Technique. To show these results, we first show that any accurate and DP mean estimator can be converted into a DP "sampling" algorithm, whose output distribution has a covariance matrix “close” (in some sense) to the empirical covariance matrix of the dataset. We thus directly prove a lower bound on the accuracy of such differentially private sampling algorithms.

Part II

Byzantine Resilience in Decentralized Optimization

Chapter 4

A Unified Breakdown Analysis for Byzantine Robust Gossip

In decentralized machine learning, different devices communicate in a peer-to-peer manner to collaboratively learn from each other's data. Such approaches are vulnerable to misbehaving (or Byzantine) devices. We introduce F–RG, a general framework for building robust decentralized algorithms with guarantees arising from robust-sum-like aggregation rules F. We then investigate the notion of *breakdown point*, and show an upper bound on the number of adversaries that decentralized algorithms can tolerate. We introduce a practical robust aggregation rule, coined CS_+, such that CS_+–RG has a near-optimal breakdown. Other choices of aggregation rules lead to existing algorithms such as ClippedGossip or NNA. We give experimental evidence to validate the effectiveness of CS_+–RG and highlight the gap with NNA, in particular against a novel attack tailored to decentralized communications.

Contents

4.1 Introduction 49
4.2 Background 50
4.2.1 Decentralized Optimization 50
4.2.2 Byzantine-Robust Decentralized Optimization 51
4.3 The Robust Gossip Framework 53
4.3.1 The Robust Gossip Method: Adding Robust Differences Instead of Averaging Robustly 53
4.3.2 Robust Summation Functions 53
4.3.3 Convergence of RG under (b, ρ)-Robustness 54
4.3.4 Spectral Limit of r-Robust Decentralized Algorithms. 55
4.4 From the General Framework to Practical Decentralized Algorithms 57
4.4.1 Examples of (b, ρ)-Robust Rules 57
4.4.2 Byzantine Robust Distributed SGD on Graphs 59
4.5 Attacking Robust Gossip Algorithms 60
4.6 Experimental Evaluation 62
4.7 Conclusion 62

4.A Additional Discussion 65
4.B Experiments 67
4.C Analysis of the Robust Gossip Method 72
4.D Proof of Theorem 4.3.5 - Upper Bound on the Breakdown Point 77
4.E (b, ρ)-Robust Summation Rules 83
4.F Proofs for D-SGD 86

4.1 Introduction

Distributed machine learning, in which the training process is performed on multiple computing units (or nodes), responds to the increasingly distributed nature of data, its sensitivity, and the rising computational cost of optimizing models. We investigate the decentralized paradigm of distributed learning, in which nodes communicate in a peer-to-peer manner within a communication network, in opposition to distributed architectures relying on a central server that coordinates all units.

Distributing the training over a large number of devices introduces new security issues: software may be faulty, local data may be corrupted, and nodes can be hacked or even controlled by a hostile party. Such issues are modeled as *Byzantine* node failures (Lamport et al., 1982), defined as omniscient adversaries able to collude with each other. Standard distributed learning methods are known to be vulnerable to these Byzantine attacks (Blanchard et al., 2017). This has led to significant efforts in the development of robust distributed learning algorithms. From the first works on Byzantine-robust SGD (Blanchard et al., 2017; Yin et al., 2018; Alistarh et al., 2018; El-Mhamdi et al., 2020), methods have been developed to tackle stochastic noise using Polyak momentum (Karimireddy et al., 2021; Farhadkhani et al., 2022) and mixing strategies to handle heterogeneous loss functions (Karimireddy et al., 2023; Allouah et al., 2023a). In parallel to these robust algorithms, efficient attacks have been developed to challenge Byzantine-robust algorithms (Baruch et al., 2019; Xie et al., 2020). Recently, Allouah et al. (2025a) has used pre-aggregation adaptive clipping to improve robustness. To bridge the gap between algorithm performance and achievable accuracy in the Byzantine setting, tight lower bounds have been constructed for the heterogeneous setting (Karimireddy et al., 2023; Allouah et al., 2024). Yet, all these works rely on a trusted central server to coordinate the training.

In contrast, the decentralized case has been less explored. Especially when the communication network, abstracted as a graph where vertices represent computing units that communicate through edges, is not fully connected (a.k.a. sparse). For instance, understanding *how many Byzantine nodes can be tolerated over an arbitrary network before a communication protocol fails* is an open question (that we address in this paper). Indeed, the network is often assumed to be fully connected (El-Mhamdi et al., 2021; Farhadkhani et al., 2023). While the work of Fang et al. (2022) addresses the case of sparse networks, it only considers homogeneous losses. Closer to our setting, He et al. (2023); Wu et al. (2023) tackle Byzantine optimization on sparse networks with heterogeneous losses, however, they only achieve suboptimal robustness, as their guarantees vanish with either the number of nodes or the connectivity of the network. Moreover, the communication scheme proposed in He et al. (2023) relies on inaccessible information. Similarly, there is a lack of attacks designed to challenge decentralized optimization. To the best of our knowledge, only He et al. (2023) proposes one, named *Dissensus*. Still, a few previous works (Wu et al., 2023; Farhadkhani et al., 2023) have investigated generic criteria to go from communication schemes to robust distributed SGD frameworks.

Our work proposes a generic method to design algorithms that solve the aforementioned shortcomings. To do so, we carefully study the decentralized mean estimation problem. This

seemingly simple problem retains most of the difficulty of handling Byzantine nodes while allowing us to derive strong convergence and robustness guarantees. Our solution relies on robust adaptations of the *gossip* communication, a popular scheme for decentralized communication. We then tackle general (smooth non-convex) optimization problems through a reduction. Our contributions are the following:

1 - Unifying algorithmic framework. We develop a generic method (the RG method) to construct and analyze robust communication algorithms. It is based on the decentralized application of robust aggregation rules. This RG method recovers NNA (Farhadkhani et al., 2022) and ClippedGossip (He et al., 2023) in specific cases. We use the RG method to build CS$_+$–RG, a novel communication algorithm that is both practical and adapted to a sparse communication network.

2 - Tight theoretical guarantees. We show that RG provides robust convergence guarantees as soon as the underlying aggregation rule verifies (b, ρ)–robustness, a new robustness criterion that we introduce, and the weight of Byzantine nodes is not too large (with respect to the *algebraic connectivity*, a spectral property of the communication graph). We also show the converse result, that is, no robustness guarantees can be obtained if the weight of Byzantine nodes exceeds this threshold. Our bounds match each other for specific aggregation rules. These results generalize the standard fully-connected breakdown point of $1/3$ of Byzantines nodes to arbitrary sparse networks.

Besides these core contributions, we introduce a theoretically grounded attack, called *Spectral Heterogeneity*, specifically designed to challenge decentralized algorithms by leveraging spectral properties of the communication graph.

The remainder of the paper is organized as follows. Section 4.2 formalizes the problem of Byzantine-robust decentralized optimization. Section 4.3 presents the robust gossip framework as well as the main convergence guarantees. Then, Section 4.4 instantiates the general framework with several rules (and the associated robustness guarantees), and provides guarantees for a D-SGD algorithm based on F–RG. Finally, Section 4.5 presents the *Spectral Heterogeneity* attack, which is then used with other attacks to experimentally challenge several aggregation schemes in Section 4.6.

4.2 Background

4.2.1 Decentralized Optimization

We consider a system composed of m computing units that communicate synchronously through a communication network, which is represented as an undirected graph $\mathcal{G}$. We denote by $\mathcal{H}$ the set of honest nodes, and $\mathcal{B}$ the (unknown) set of Byzantine nodes. Each unit i holds a local parameter $\boldsymbol{x}_i \in \mathbb{R}^d$, a local loss function $f_i : \mathbb{R}^d \to \mathbb{R}$, and can communicate with its neighbors in the graph $\mathcal{G}$. We denote the set of neighbors of node i by $n(i)$ and by $n_{\mathcal{H}}(i)$ (resp. $n_{\mathcal{B}}(i)$) the set of honest (resp. Byzantine) ones. We study decentralized algorithms for solving

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\arg\min} \left\{ f_{\mathcal{H}}(\boldsymbol{x}) := \frac{1}{|\mathcal{H}|} \sum_{i\in\mathcal{H}} f_i(\boldsymbol{x}) \right\}. \tag{4.1}$$

Due to the averaging nature of Equation (4.1), centralized algorithms for solving this problem rely on global averaging of the gradients computed at each node. In the decentralized setting, we rely on performing local (node-wise) inexact averaging steps instead.

Gossip Communication. Standard decentralized optimization algorithms typically rely on the so-called *gossip* communication protocol (Boyd et al., 2006; Nedic and Ozdaglar, 2009; Scaman et al., 2017; Kovalev et al., 2020). The gossip protocol consists of updating the parameter of a node i through a linear combination of the parameters of its neighbors, with updates of the form $\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t - \eta \sum_{j=1}^m w_{ij}(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t)$, where w_{ij} is a weight associated with the edge (ij) of the graph and $\eta \geq 0$ will denote a communication step-size. By denoting $\boldsymbol{W}$ the Laplacian matrix associated with the weighted graph $\mathcal{G}$, i.e $\boldsymbol{W}_{ij} = -w_{ij}$ if $i \neq j$ and $\boldsymbol{W}_{ii} = \sum_{j \in n(i)} w_{ij}$, and denoting the matrix of parameters as $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_m)^T$, then the gossip update conveniently writes:

$$\boldsymbol{X}^{t+1} = \boldsymbol{X}^t - \eta \boldsymbol{W} \boldsymbol{X}^t. \tag{4.2}$$

The Laplacian matrix (a.k.a. gossip matrix), is symmetric non-negative. When the graph is connected, its kernel is restricted to the line of the constant vectors, i.e. $\text{span}(1, \dots, 1)^T$. In the following, we will always consider that the graph $\mathcal{G}$ is weighted, so that a unique Laplacian matrix is associated with each graph $\mathcal{G}$. Moreover, we denote by $\mu_{\max}(\mathcal{G}_{\mathcal{H}})$ and $\mu_2(\mathcal{G}_{\mathcal{H}})$ the largest and smallest non-zero eigenvalues of the Laplacian matrix $\boldsymbol{W}_{\mathcal{H}}$ of the *honest subgraph* $\mathcal{G}_{\mathcal{H}}$, and by $\gamma = \mu_2(\mathcal{G}_{\mathcal{H}})/\mu_{\max}(\mathcal{G}_{\mathcal{H}})$ its *spectral gap*. Spectral properties of the Laplacian matrix are known to characterize the convergence of gossip optimization methods. For instance, in the absence of Byzantine nodes, Equation (4.2) with step-size $\eta \leq \mu_{\max}(\mathcal{G})^{-1}$ leads to a linear convergence of the nodes parameter values towards the average of the initial parameters: $\|\boldsymbol{X}^t - \overline{\boldsymbol{X}}^0\|^2 \leq (1 - \eta\mu_2(\mathcal{G}))^t \|\boldsymbol{X}^0 - \overline{\boldsymbol{X}}^0\|^2$, for $\overline{\boldsymbol{X}}^0$ the matrix with columns $m^{-1} \sum_{j=1}^m \boldsymbol{x}_j^0$.

Robustness Issue. Gossip communication relies on updating the nodes' parameters by performing non-robust local averaging. As such, similarly to the centralized case, any Byzantine neighbor of node i can drive the update to any desired value (Blanchard et al., 2017). Then, the poisoned information spreads through gossip communications.

4.2.2 Byzantine-Robust Decentralized Optimization

In this section, we describe the threat model and the robustness criterion that we consider.

Threat model. We consider Byzantine nodes to be unknown omniscient adversaries, able to collude and send distinct values to each of their neighbors. We measure their influence by considering the weight of Byzantine nodes in the neighborhood of each honest node, $(b(i) := \sum_{j \in n_{\mathcal{B}}(i)} w_{ij})_{i \in \mathcal{H}}$ (as in LeBlanc et al. (2013); He et al. (2023)), instead of the total number of Byzantine nodes $|\mathcal{B}|$ as done for the centralized or fully-connected setting. Similarly, we denote by $h(i) = \sum_{j \in n_{\mathcal{H}}(i)} w_{ij}$ the weights of the honest nodes adjacent to the node i.

In the case of an arbitrary communication network, the number of honest nodes does not provide enough information by itself, as the results depend on *how* the nodes are

linked, i.e., the topology of the honest subgraph. Therefore, in the case of sparse topologies, it is necessary to consider a property of the graph related to its structure rather than relying solely on the number of honest nodes. We will show that spectral properties of the Laplacian of honest subgraph are relevant quantities for robustness analyses and introduce the following class of graphs.

Definition 4.2.1. For any $\mu_{\min} \geq 0$ and $b \geq 0$, we define the class of weighted graphs

$$\Gamma_{\mu_{\min},b} = \left\{ \mathcal{G} \;\text{ s.t. }\; \mu_2(\mathcal{G}_{\mathcal{H}}) \geq \mu_{\min} \text{ and } \max_{i \in \mathcal{H}} b(i) \leq b \right\}.$$

In other words, we introduce a subset of all possible graphs, partitioning in terms of (i) the second smallest eigenvalue of the honest subgraph, (a.k.a. the *algebraic connectivity*), that is restricted to be larger than a minimal value $\mu_{\min}$, and (ii) the maximal weight of Byzantines in the neighborhood of an honest node, which is restricted to be smaller than b.

Note that if all edges are equally weighted ($w_{ij} = \omega$), then $b(i) = |n_{\mathcal{B}}(i)| \cdot \omega$, and (ii) boils down to upper bounding the number of Byzantines in the neighborhood of honest nodes by $f := b \cdot \omega^{-1}$. Hence, Definition 4.2.1 is an extension of the standard *"there are at most f byzantine nodes among the $|\mathcal{B}| + |\mathcal{H}|$ nodes"* to the setting of arbitrary connected graphs. We point out that this class of graphs depends on the spectral properties *of the honest subgraph*, meaning that for a given communication network, these properties change depending on the location of Byzantine failures, and the associated graph can either fall within $\Gamma_{\mu_{\min},b}$ if Byzantines are "well spread", or not, e.g. if they are adversarially chosen.

Approximate Average Consensus. The average consensus problem consists in finding the average of vectors locally held by honest nodes $(\boldsymbol{x}_i)_{i \in \mathcal{H}}$. It is a specific case of Equation (4.1) obtained by considering $f_i(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}_i\|^2$. Due to adversarial attacks only an *approximate* estimate of the average of honest nodes vector $\overline{\boldsymbol{x}}_{\mathcal{H}} := |\mathcal{H}|^{-1} \sum_{i \in \mathcal{H}} \boldsymbol{x}_i$ can be reached (Karimireddy et al., 2023). We therefore introduce the following criterion to assess the robustness of a communication algorithm.

Definition 4.2.2 (r-robustness on $\mathcal{G}$.)**.** Let $r < 1$. A communication algorithm Alg is r-robust on a graph $\mathcal{G}$ if from any initial local parameters $(\boldsymbol{x}_i)_{i \in \mathcal{H}} \in (\mathbb{R}^d)^{\mathcal{H}}$, it allows any honest node i to compute a vector $\boldsymbol{x}_i^+$ such that

$$\frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \|\boldsymbol{x}_i^+ - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2 \leq r \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2.$$

Imposing $r < 1$ means the honest nodes' parameters have to be strictly closer (on average) to the initial mean after the communication than before. It thus requires that the reduction of the *variance* of nodes parameters $\mathrm{Var}_{\mathcal{H}}(\boldsymbol{x}) - \mathrm{Var}_{\mathcal{H}}(\boldsymbol{x}^+)$ is larger than the *bias* $\|\overline{\boldsymbol{x}}_{\mathcal{H}}^+ - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2$ introduced by the Byzantines, with $\mathrm{Var}_{\mathcal{H}}(\boldsymbol{x}) := |\mathcal{H}|^{-1} \sum_{i \in \mathcal{H}} \|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2$. Remark that the r-robustness of an algorithm on a graph $\mathcal{G}$ states that a *single use* of the algorithm strictly reduces the average quadratic error. However, it does not mean that multiple uses would result in a geometric decrease; indeed, we cannot simply use induction

as $\overline{\boldsymbol{x}}^+_{\mathcal{H}} \neq \overline{\boldsymbol{x}}_{\mathcal{H}}$.

4.3 The Robust Gossip Framework

We now introduce a generic framework, which relies on two key building blocks: (i) a generic update form, depending on an aggregation function $\mathrm{F} : (\mathbb{R}_+ \times \mathbb{R}^{\mathrm{d}})^{\mathrm{n}} \to \mathbb{R}^{\mathrm{d}}$, and (ii) a set of those aggregation functions F, referred to as *robust summation* functions. We then show that the combination of these two blocks, *i.e.*, the generic update used with a robust summation function F, leads to a robust decentralized algorithm, hence the name: Robust Gossip.

We finally show that this framework leads to (near-)optimal robustness guarantees for well-chosen F.

4.3.1 The Robust Gossip Method: Adding Robust Differences Instead of Averaging Robustly

We call *aggregation rule* a function $\mathrm{F} : (\mathbb{R}_+ \times \mathbb{R}^{\mathrm{d}})^{\mathrm{n}} \to \mathbb{R}^{\mathrm{d}}$, meant to aggregate a set of vectors $(z_i)_{i\in[n]} \in (\mathbb{R}^d)^n$ with weights $(w_i)_{i\in[n]} \in \mathbb{R}^n_+$. For $\eta > 0$ a communication step-size, the associated robust gossip algorithm (coined F–RG) consists, for any honest node $i \in \mathcal{H}$ of the communication network $\mathcal{G}$, in updating its parameter $\boldsymbol{x}_i$ using:

$$\boldsymbol{x}_i^+ := \boldsymbol{x}_i - \eta F\left((w_{ij}, \boldsymbol{x}_i - \boldsymbol{x}_j)_{j\in n(i)}\right). \tag{F–RG}$$

Crucially, each node applies the (robust) aggregation rule F to $(\omega_j, \boldsymbol{z}_j)_{j\in n(i)} := (w_{ij}, \boldsymbol{x}_i - \boldsymbol{x}_j)_{j\in n(i)}$, i.e. to *the differences of its parameter with those of its neighbors*, and uses this estimate to update its parameter. Thus, Equation (F–RG) recovers the standard gossip update from Equation (4.2) if F is the weighted sum operator (which is unfortunately not robust). Hence, we will look for aggregation functions that are robust versions of the weighted sum.

In contrast, directly averaging *the parameters* of the neighbors, even with a robust aggregation rule, would highly suffer from heterogeneity. Indeed, this leads to a *biased estimate* of the mean of initial parameters for sparse communication graphs. Extra assumptions, such as homogeneity of the local objectives, are thus needed to alleviate this problem (Fang et al., 2022).

Meanwhile, the RG method uses intrinsically decentralized updates, allowing for tight convergence guarantees in sparse communication graphs with heterogeneous local objectives. The strength of the RG framework is to turn any robust summation into a robust gossip algorithm. As such, one can focus on the design of robust aggregation functions without worrying about the decentralized aspect.

4.3.2 Robust Summation Functions

Our analysis of F–RG relies on aggregation rules that meet the following robustness conditions.

Definition 4.3.1 ((b,ρ)–robust summation)**.** Let $b, \rho \geq 0$. An aggregation rule $F : (\mathbb{R}_+ \times \mathbb{R}^d)^n \to \mathbb{R}^d$ is a (b,ρ)–*robust summation* if, for any vectors $(\boldsymbol{z}_i)_{i\in[n]} \in (\mathbb{R}^d)^n$, any weights $(\omega_i)_{i\in[n]} \in \mathbb{R}_+^n$ and any $S \subset [n]$ such that $\sum_{i\in\overline{S}} \omega_i \leq b$ (where $\overline{S} := [n]\backslash S$),

$$\left\| F\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]}\big) - \sum_{i\in S} \omega_i \boldsymbol{z}_i \right\|^2 \leq \rho b \sum_{i\in S} \omega_i \|\boldsymbol{z}_i\|^2.$$

In (F–RG), S is the set of honest neighbors $n_{\mathcal{H}}(i)$, while $\overline{S}$ is the set of Byzantine neighbors $n_{\mathcal{B}}(i)$. We will exhibit several (b,ρ)–robust summation rules, including a practical rule for $\rho = 2$, in Section 4.4.

Remark 4.3.2. The latter definition differs from (f,κ)–*robustness* (Allouah et al., 2023a) we upper bound the error using the *second moment* of vectors within S *instead of their variance*. Note also that (f,κ)–robustness is stated with constant weights only. Therefore, if F is (f,κ)–robust, then F is a (b,ρ)-robust summation with e.g. uniform weights $\omega_i = 1/(n-f)$ with $b = f/(n-f)$ and $\rho = \kappa/b$.

4.3.3 Convergence of RG under (b,ρ)-Robustness

As briefly discussed in the introduction, the goal of communicating is to reduce the variance, which comes at the price of bias. This is unavoidable since communicating allows nodes to inject wrong information which biases the system.

In the following core result, we show how (b,ρ)–robustness enables us to tightly quantify how much a single step of F–RG reduces the variance, and how much bias is injected. Recall that $\mathrm{Var}_{\mathcal{H}}(\boldsymbol{x}) = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2$ is the variance of honest nodes.

Theorem 4.3.3. *Let F be a (b,ρ)–robust summation, b and $\mu_{\min}$ be s.t. $2\rho b \leq \mu_{\min}$, and $\mathcal{G} \in \Gamma_{\mu_{\min},b}$. Then, assuming $\eta \leq \mu_{\max}(\mathcal{G}_{\mathcal{H}})^{-1}$, the output $(\boldsymbol{x}_i^+)_{i\in\mathcal{H}}$ of* F–RG *verifies:*

$$\begin{cases} \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \|\boldsymbol{x}_i^+ - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2 \leq (1 - \eta\,(\mu_{\min} - 2\rho b))\,\mathrm{Var}_{\mathcal{H}}(\boldsymbol{x}), \\ \|\overline{\boldsymbol{x}}_{\mathcal{H}}^+ - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2 \leq 2\rho b\,\eta\,\mathrm{Var}_{\mathcal{H}}(\boldsymbol{x}). \end{cases}$$

Thus, F–RG *is $(1-\eta\,(\mu_{\min} - 2\rho b))$–robust for $\mathcal{G} \in \Gamma_{\mu_{\min},b}$.*

While the bound on η depends on the honest subgraph, as $\mu_{\max}(\mathcal{G}_{\mathcal{H}}) \leq \mu_{\max}(\mathcal{G})$, η can be set conservatively by evaluating $\mu_{\max}$ on the whole graph. Note that while r–robustness is guaranteed for the whole class $\Gamma_{\mu_{\min},b}$, the value of r will depend on the actual graph within the class.

Chaining aggregation steps. When low variance levels are required, it is necessary to perform several robust gossip steps one after the other. This contrasts with the centralized setting, in which the variance can be brought to zero in one step. While the variance reduces at a linear rate, the bias accumulates as multiple aggregation steps are performed. We provide bounds for t steps of F–RG in the following Corollary.

Corollary 4.3.4. *Let F be a (b,ρ)–robust summation, let b and $\mu_{\min}$ be such that $2\rho b \leq \mu_{\min}$, and let $\mathcal{G} \in \Gamma_{\mu_{\min},b}$. We denote $\delta = \frac{2\rho b}{\mu_{\min}}$ and $\gamma = \mu_{\min}/\mu_{\max}(\mathcal{G}_{\mathcal{H}})$. Then, for $(\boldsymbol{x}_i^t)_{i\in\mathcal{G},\, t\geq 0}$ obtained from any $(\boldsymbol{x}_i^0)_{i\in\mathcal{G}}$ through $(\boldsymbol{x}_i^{t+1})_{i\in\mathcal{G}} = \text{F–RG}\big((\boldsymbol{x}^t)_{i\in\mathcal{G}}\big)$, with $\eta = \mu_{\max}(\mathcal{G}_{\mathcal{H}})^{-1}$,*

$$\begin{cases} \operatorname{Var}_{\mathcal{H}}(\boldsymbol{x}^t) \leq (1-\gamma(1-\delta))^t \operatorname{Var}_{\mathcal{H}}(\boldsymbol{x}^0), \\ \|\overline{\boldsymbol{x}}_{\mathcal{H}}^t - \overline{\boldsymbol{x}}_{\mathcal{H}}^0\| \leq \frac{\sqrt{\gamma\delta}\left(1-[1-\gamma(1-\delta)]^{t/2}\right)}{1-\sqrt{1-\gamma(1-\delta)}} \sqrt{\operatorname{Var}_{\mathcal{H}}(\boldsymbol{x}^0)}. \end{cases}$$

Consensus is thus reached, as $\operatorname{Var}_{\mathcal{H}}(\boldsymbol{x}^t) \to_{t\to\infty} 0$, and

$$\|\overline{\boldsymbol{x}}_{\mathcal{H}}^t - \overline{\boldsymbol{x}}_{\mathcal{H}}^0\|^2 \leq \frac{4\delta}{\gamma(1-\delta)^2} \operatorname{Var}_{\mathcal{H}}(\boldsymbol{x}^0). \tag{4.3}$$

While the total L2 error (bias plus variance) is guaranteed to decrease after a single step by Theorem 4.3.3, it may increase if several F–RG steps are performed because of bias accumulation. This happens when the factor multiplying the variance in Equation (4.3) is larger than 1, which essentially means $\gamma \leq \delta$. Despite this bias, the output of the resulting robust aggregation procedure is (arbitrarily) close to consensus, which can be desirable. Proofs of Theorem 4.3.3 and Corollary 4.3.4 are respectively given in Sections 4.C.1 and 4.C.2.

Dependence on the parameters. As expected, the bias increases with the amount of Byzantine corruption (through δ) and decreases as the graph becomes more connected (i.e, $\gamma \to 1$). One can then use parameter η (up to its maximum value) to control the bias-variance trade-off.

4.3.4 Spectral Limit of r-Robust Decentralized Algorithms.

We now provide an *upper bound* on the weight of Byzantine neighbors that can be tolerated by any algorithm running on a communication network in which the honest subgraph has a given algebraic connectivity.

Theorem 4.3.5. *Let $\mu_{\min} \geq 0$, $b \geq 0$ be such that $\mu_{\min} \leq 2b$. Then for any $h \geq 0$ and any algorithm* Alg*, there exists a graph $\mathcal{G} \in \Gamma_{\mu_{\min},b}$ in which all honest nodes have a weight of honest neighbors $h(i)$ larger than h, and such that for any $r < 1$,* Alg *is not r–robust on $\mathcal{G}$.*

We refer the reader to Section 4.D for the proof details. It follows from Theorem 4.3.5 that when a theoretical guarantee quantifies the robustness of an algorithm on a graph through $\mu_{\min}$, we must have $2b < \mu_{\min}$. Importantly, this upper bound on the breakdown point is *independent of the total weight of honest neighbors.*

For a fully-connected graph with uniform weights ω, we have $\mu_2(\mathcal{G}_{\mathcal{H}}) = \omega|\mathcal{H}|$ and $b(i) = \omega|\mathcal{B}|$. Then, the previous condition boils down to $|\mathcal{H}| > 2|\mathcal{B}|$, i.e. there is less than 1/3 of Byzantine nodes, which recovers known necessary robustness conditions of distributed system (Lamport et al., 1982; Vaidya et al., 2012; El-Mhamdi et al., 2021).

Near-optimal breakdown point. Theorem 4.3.5 states that uniformly ensuring r–robustness on $\Gamma_{\mu_{\min},b}$ is impossible as soon as $2b \geq \mu_{\min}$, and we know that update (F–RG) is r-robust as soon as $2\rho b < \mu_{\min}$. Therefore, no (ρ, b)-robust summand exists with $\rho < 1$, and (F–RG) achieves the optimal breakdown if a $(b, 1)$-robust aggregation rule is used. Such rules exist, as shown in Section 4.4, but are unfortunately not practical, as they require prior knowledge on the Byzantine nodes. It is an open question whether $\rho = 1$ can be achieved using a practical rule.

Nevertheless, we propose a practical robust summation rule that achieves $\rho = 2$, thus robust if $4b < \mu_{\min}$. This is a significant improvement over existing works. For example, in He et al. (2023), the $4b < \mu_{\min}$ condition is essentially replaced by $cb \leq \gamma\mu_{\min}$ (*e.g.*, for regular graphs), where $c > 0$ is a large constant, and $\gamma = {\mu_2(\mathcal{G}_\mathcal{H})}/{\mu_{\max}(\mathcal{G}_\mathcal{H})}$ the graph's spectral gap. This gap rapidly shrinks with the size and the lack of connectivity of the graph, making the condition orders of magnitude worse for large sparse graph. In Wu et al. (2023), the breakdown condition is $8b\sqrt{|\mathcal{H}|} \leq \mu_{\min}$, which means that the robustness guarantee decreases when the number of honest nodes increases. For instance, only a ${1}/{(9|\mathcal{H}|^{1/2})}$ fraction of Byzantine nodes is tolerated for a fully-connected network, whereas we tolerate up to ${1}/{5}$.

We conclude this section by two remarks on potential alternative characterizations of the breakdown point.

Remark 4.3.6 (On algebraic connectivity). Theorem 4.3.5 does not imply that a given communication algorithm systematically fails as soon as $2b \geq \mu_2(\mathcal{G})$, but rather that since there exists a graph for which it is the case, one can not have an r–robust algorithm with an assumption based on $\mu_2(\mathcal{G})$ and b looser than $\mu_2(\mathcal{G}) \geq 2b$. Yet, one can still prove breakdown points using other graph-related quantities, which might lead to tolerating $b > {\mu_{\min}}/{2}$ Byzantine nodes for specific graph architectures. This gap is standard in the decentralized optimization literature, where optimal algorithms depend on the (square root of the) *spectral gap* of the gossip matrix (Scaman et al., 2017; Kovalev et al., 2020), whereas iteration lower bounds are proven in terms of diameter of the communication graph.

Remark 4.3.7 (Dimension-dependent breakdown points). The Approximate Average Consensus problem (Section 4.2.2) is related to the *Approximate Consensus Problem* (ACP) (Dolev et al., 1986), in which the nodes must converge to the same value while *remaining within the convex hull* of the initial parameters. This is a harder problem, since the ACP cannot be solved using iterative communication on a system of m nodes with f Byzantine failures in dimension d if $m \leq (d+2)f + 1$ (Vaidya, 2014). This dependence on the dimension d is prohibitive for ML applications. On the contrary, our definition of r–robustness only requires the algorithm to improve the average squared distance to the target value, which is enough for D-SGD to converge, and enables us to prove *dimension-independent* breakdown. Yet, it would be interesting to link their notion of r-robust networks (LeBlanc et al., 2013) with algebraic connectivity.

4.4 From the General Framework to Practical Decentralized Algorithms

In this section, we first define robust summation rules and link our general framework with existing decentralized robust algorithms in Section 4.4.1. Then we prove convergence for Decentralized-SGD based on F–RG, in Section 4.4.2.

4.4.1 Examples of (b, ρ)-Robust Rules

Several methods have been proved to be (f, κ)-robust, including the Coordinate-Wise Trimmed Mean (CWTM) (Yin et al., 2018), the Coordinate-wise Median (CWM) (Yin et al., 2018), the Geometric Median (GM) (Yin et al., 2018; Pillutla et al., 2022) and Krum (Blanchard et al., 2017). It follows from Remark 4.3.2 that they are also (b, ρ)–robust summations.

However, since (b, ρ)–robust summation is weaker than (f, κ)–robustness, we can introduce robust aggregation methods using this new perspective with tighter robustness guarantees. The following aggregators either recover existing algorithms, or are tighter than previous approaches. In their definition, $(\omega_i, \boldsymbol{z}_i)_{i\in[n]} \in (\mathbb{R}_+ \times \mathbb{R}^d)^n$ and $S \subset [n]$ such that $\sum_{\overline{S}} \omega_i \leq b$.

Geometric Trimmed Sum (GTS). Assume w.l.o.g. that $(\|\boldsymbol{z}_i\|)_{i\in[n]}$ are sorted, i.e. $\|\boldsymbol{z}_1\| \leq \ldots \leq \|\boldsymbol{z}_n\|$, and we denote as $k^*(b) := \max\{k \in [n]; \sum_{i\geq k} \omega_i \geq b\}$ the index of the largest vector which has at least a weight b of vectors larger than it[1]. (GTS) computes $\tilde{\omega}_{k^*(b)} := \sum_{i\geq k^*(b)} \omega_i - b$, and outputs

$$\mathrm{GTS}\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]}\big) = \tilde{\omega}_{k^*(b)} \boldsymbol{z}_{k^*} + \sum_{i<k^*(b)} \omega_i \boldsymbol{z}_i.$$

In the simpler case where the weights are 1 and $b \in [n]$, GTS consists in discarding the b largest vectors within $(\boldsymbol{z}_i)_{i\in[n]}$ and summing the rest.

Clipped Sum (CS). Given a threshold function $\tau : (\mathbb{R}_+ \times \mathbb{R}^d)^n \mapsto \mathbb{R}_+$, output the mean of clipped vectors:

$$\mathrm{CS}\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]}; \tau\big) := \frac{1}{n} \sum_{i=1}^{n} \omega_i \mathrm{clip}\big(\boldsymbol{z}_i; \tau\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]}\big)\big),$$

$$\text{where } \forall \boldsymbol{z} \in \mathbb{R}^d, \tilde{\tau} \in \mathbb{R}_+, \mathrm{clip}(\boldsymbol{z}; \tilde{\tau}) := \frac{\boldsymbol{z}}{\|\boldsymbol{z}\|} \min(\|\boldsymbol{z}\|, \tilde{\tau}).$$

We propose the following threshold function[2], which leads to a practical and nearly optimal aggregator.

Practical Clipping (CS_+). Let $\mathrm{CS}_+ := \mathrm{CS}\big(\cdot\,; \tau_+\big)$, where

$$\tau_+\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]}\big) := \max\Big\{\tau \geq 0 : \sum_{i=1}^{n} \omega_i \mathbf{1}_{\|\boldsymbol{z}_i\|\geq\tau} \geq 2b\Big\}.$$

[1] When weights sum to 1, k^* is a quantile function.

[2] Note that Clipped Sum cannot be a (b, ρ)–robust summation when the threshold function is a fixed constant $\tau \geq 0$: the clipping threshold must be adaptive to the input vectors.

This rule corresponds, in the specific case of unitary weights $\omega_i = 1$ and $b \in [n]$, to defining the clipping threshold as the $2b^{th}$ largest value within $\|\boldsymbol{z}_1\|, \ldots, \|\boldsymbol{z}_n\|$ (*i.e.*, $\|z_{k^*(2b)}\|$). We now provide a robustness guarantee for these two aggregation rules in the following theorem.

Theorem 4.4.1. *Let $b \geq 0$, then* GTS *is (b, ρ)–robust with $\rho = 4$, and* CS_+ *is (b, ρ)–robust with $\rho = 2$.*

Remark 4.4.2 (Concurrent work). Allouah et al. (2025a) study the influence of an adaptive clipping scheme, named Adaptive Robust Clipping (ARC), with the same adaptive clipping threshold as CS_+. However, they use it *before* any aggregation function (f, κ)–robust F, and only analyze the robustness of F $\circ$ ARC, making their approach orthogonal to ours. Moreover, their focus is on the federated case.

Next, we define *oracle* (or.) threshold functions. Those require the knowledge of the set S to be computed, which eventually corresponds to being able to identify which node is honest and which node is Byzantine. This is obviously not a reasonable assumption in practice. Nevertheless, it shows that the optimality gap is not inherent to clipping, since an oracle choice of threshold is optimal. Furthermore, the guarantees from He et al. (2023) rely on such assumptions[3].

Oracle Clipping. Let $\text{CS}_+^{\text{or.}} := \text{CS}\big(\,\cdot\,; \tau_+^{\text{or.}}\big)$, where $\tau_+^{\text{or.}}\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]}; S\big) = \max\big\{\tau \geq 0 : \sum_{i\in S} \omega_i \mathbf{1}_{\|\boldsymbol{z}_i\|\geq\tau} \geq b\big\}$.

Oracle clipping (He et al., 2023). $\text{CS}_{\text{He}}^{\text{or.}} := \text{CS}\big(\,\cdot\,; \tau_{\text{He}}^{\text{or.}}\big)$, where $\tau_{\text{He}}^{\text{or.}}\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]}; S\big) = \sqrt{\frac{1}{b}\sum_{i\in S} \omega_i \|\boldsymbol{z}_i\|^2}$.

As shown below, $\text{CS}_+^{\text{or.}}$ leads to an optimal breakdown point. On the contrary, $\text{CS}_{\text{He}}^{\text{or.}}$ only achieves $\rho = 4$, despite its oracle aspect. Proofs of Theorems 4.4.1 and 4.4.3 are given in Section 4.E.

Theorem 4.4.3. *Let $b \geq 0$, then $\text{CS}_+^{or.}$ is (b, ρ)–robust with $\rho = 1$, and $\text{CS}_{He}^{or.}$ is (b, ρ)–robust with $\rho = 4$.*

When instantiated in specific settings, our framework gives tight convergence guarantees (improving on the existing ones) for existing algorithms.

Proposition 4.4.4. (F–RG) *recovers existing algorithms.*

1. GTS–RG*, on a fully connected communication network $\mathcal{G}$ with constant weight, corresponds to Nearest Neighbors Averaging (Farhadkhani et al., 2022,* NNA*).*
2. $\text{CS}_{He}^{or.}$–RG *recovers* ClippedGossip *(He et al., 2023).*

[3] In their experiments, they propose a practical (i.e. non-oracle) threshold function that is not supported by theory.

4.4.2 Byzantine Robust Distributed SGD on Graphs

We now give convergence results for a D-SGD-type algorithm that uses (F–RG) for robust decentralized aggregation. Several works on Byzantine-robust SGD abstract away the aggregation procedure by relying on contraction properties (Karimireddy et al., 2021; Wu et al., 2023; Farhadkhani et al., 2023), so that global D-SGD convergence follows from the robustness of the averaging procedure. Our Corollary 4.4.8 builds on the (α, λ)-reduction from Farhadkhani et al. (2023):

Definition 4.4.5 ((α, λ)-reduction)**.** A coordinating phase Ψ verifies an (α, λ)-reduction if, from any initial local parameters $(\boldsymbol{x}_i)_{i\in\mathcal{H}} \in (\mathbb{R}^d)^{\mathcal{H}}$, each honest nodes $i \in \mathcal{H}$ obtain a parameter vector $\boldsymbol{x}_i^+$ upon the completion of Ψ such that

$$\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{x}_i^+ - \overline{\boldsymbol{x}}_{\mathcal{H}}^+\|^2 \leq \alpha\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2,$$
$$\|\overline{\boldsymbol{x}}_{\mathcal{H}}^+ - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2 \leq \lambda\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2.$$

Remark 4.4.6. (α, λ)-reduction and r-robustness are closely related quantities since (α, λ) reductions implies r-robustness with $r \leq \alpha + \lambda$, and r-robustness implies $\alpha, \lambda \leq r$. Yet, r-robustness explicitly requires that $r < 1$, unlike (α, λ)-reduction. In essence, r-robustness expresses more precisely that *nodes benefit from the communication.*

The (α, λ) requirements on the aggregation procedure exactly match the guarantees of Theorem 4.3.3: using one single step of F–RG as a coordination phase leads to $\alpha = 1 - \gamma(1-\delta)$ and $\lambda = \gamma\delta$, and using multiple communication steps leads to $\alpha \approx 0$ and $\lambda = {}^{4\delta}/_{\gamma(1-\delta)^2}$. We build on this abstraction to propose a Byzantine robust decentralized stochastic gradient descent framework.

We consider Problem 4.1, where we assume that each local function f_i is a risk computed using a loss ℓ on a data distribution $\mathcal{D}_i$, i.e $f_i(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D}_i}[\nabla\ell(\boldsymbol{x}, \boldsymbol{\xi})]$. We solve Problem 4.1 using D-SGD over a communication network $\mathcal{G}$. Robustness to Byzantine nodes is obtained using (F–RG) as the aggregation rule, coupled with Polyak momentum used as a moving average to reduce the stochastic noise.

The convergence results of this algorithm rely on the following standard assumptions.

Assumption 4.4.7. Objective functions regularity.

1. **(Smoothness)** There exists $L \geq 0$, s.t. $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, $\|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$.
2. **(Bounded noise)** There exists $\sigma \geq 0$ s.t. $\forall \boldsymbol{x} \in \mathbb{R}^d$, $\mathbb{E}[\|\nabla\ell(\boldsymbol{x}, \xi_i) - \nabla f_i(\boldsymbol{x})\|^2] \leq \sigma^2$, for all $i \in [\mathcal{H}]$.
3. **(Heterogeneity)** There exist $\zeta \geq 0$ s.t. $\forall \boldsymbol{x} \in \mathbb{R}^d$, $\frac{1}{\mathcal{H}}\sum_{i\in\mathcal{H}}\|\nabla f_i(\boldsymbol{x}) - \nabla f_{\mathcal{H}}(\boldsymbol{x})\|^2 \leq \zeta^2$.

Algorithm 1 Byzantine-Resilient D-SGD with F–RG

Input: Initial model $\boldsymbol{x}_i^0 \in \mathbb{R}^d$, local loss functions f_i, initial momentum $m_i^0 = 0$, momentum coefficient $\beta = 0$, learning rate η_{op}, communication step size η, communication graph $\mathcal{G}$, upper bound on Byzantine weight b.
for $t = 0$ **to** T **do**
  for $i \in \mathcal{H}$ **in parallel do**
    Sample a noisy gradient: $\boldsymbol{g}_i^t = \nabla f_i(\boldsymbol{x}_i^t) + \boldsymbol{\xi}_i^t$.
    Update the momentum: $\boldsymbol{m}_i^t = \beta \boldsymbol{m}_i^{t-1} + (1-\beta)\boldsymbol{g}_i^t$.
    Optimization step: $\boldsymbol{x}_i^{t+1/2} = \boldsymbol{x}_i^t - \eta_{op}\boldsymbol{m}_i^t$.
    Send $\boldsymbol{x}_i^{t+1/2}$ to the neighbors $n(i)$.
    Update the model with F–RG
    $\boldsymbol{x}_i^{t+1} = \text{F–RG}\left(\boldsymbol{x}_i^{t+1/2}; \{\boldsymbol{x}_j^{t+1/2};\ j \in n(i)\}\right)$.
  end for
end for

We can now state the guarantees of Algorithm 1.

Corollary 4.4.8. *Let F a (b,ρ)–robust summation, let $b \geq 0$ and let $\mathcal{G}$ a weighted graph such that $\delta = 2\rho/\mu_2(\mathcal{G}_\mathcal{H}) < 1$ and of spectral gap $\gamma = \mu_2(\mathcal{G}_\mathcal{H})/\mu_{\max}(\mathcal{G}_\mathcal{H})$. Under Assumption 4.4.7, for all $i \in \mathcal{H}$, the iterates produced by Algorithm 1 on $\mathcal{G}$ with $\eta = 1/\mu_{\max}(\mathcal{G})$ and learning rate $\eta_{op} = \mathcal{O}(1/\sqrt{T})$ (depending also on problem parameters such as L, γ or δ), verify as T increases:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla f_\mathcal{H}(\boldsymbol{x}_i^t)\right\|^2\right] = \mathcal{O}\left(\frac{L\sigma}{\gamma(1-\delta)\sqrt{T}} + \frac{\zeta^2}{\gamma^2(1-\delta)^2}\right)$$
$$\mathrm{Var}_\mathcal{H}(\boldsymbol{x}^T) = \mathcal{O}\left(\frac{1}{T}\left(1 + \frac{\zeta^2}{\sigma^2}\right)\right).$$

Performing $\tilde{\mathcal{O}}(\gamma^{-1}(1-\delta)^{-1})$ steps of F–RG *between each gradient computation leads to:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla f_\mathcal{H}(\boldsymbol{x}_i^t)\right\|^2\right] = \mathcal{O}\left(\frac{L\sigma}{\sqrt{T}}\sqrt{\frac{\delta}{\gamma(1-\delta)^2}} + \frac{\delta\zeta^2}{\gamma(1-\delta)}\right).$$

It follows that those guarantees state that performing more aggregation steps between gradient computations improves the asymptotic error but under an additional communication cost.

This corollary is the consequence of the combination of our Theorem 4.3.3 with Theorem 1 of Farhadkhani et al. (2023). The result is simplified using $\delta \geq |\mathcal{H}|^{-1}$, and $\gamma \ll 1$. We refer the reader to Section 4.F for a more precise result and a detailed proof.

4.5 Attacking Robust Gossip Algorithms

In this section, we design an attack that aims to disrupt robust gossip algorithms. To do this, we model communications as perturbations of a gossip scheme, and analyze their impact on the variance among nodes, which allows us to deduce what perturbation Byzantines nodes

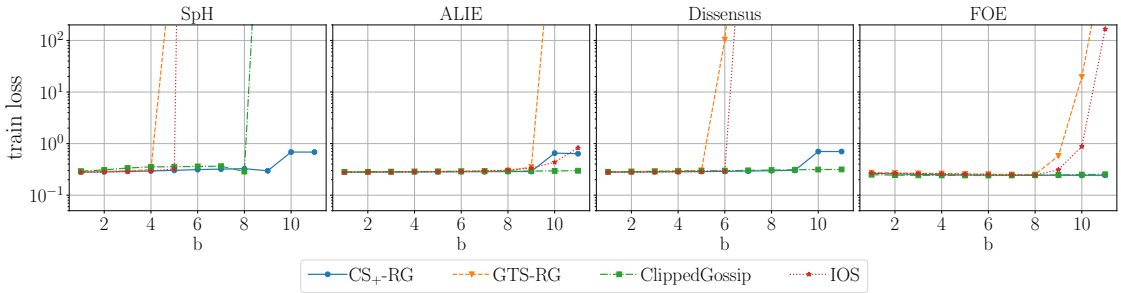


Figure 4.1: Training loss achieved by GTS–RG, CS$_+$–RG, ClippedGossip and IOS on MNIST ($\alpha = 1$) after 300 optimization and communication steps. The honest subgraph graph is $\mathcal{G}_\mathcal{H} := [\mathcal{G}_{m=13,k=8,c=1}]_\mathcal{H}$, as defined in Section 4.D. Thus, $\mu_2(\mathcal{G}_\mathcal{H}) = 16$.

should enforce for effective attacks. Recall that $\boldsymbol{X}^t_\mathcal{H} = (\boldsymbol{x}^t_1, \dots, \boldsymbol{x}^t_{|\mathcal{H}|})^T \in \mathbb{R}^{|\mathcal{H}| \times d}$ denotes the matrix of honest parameters at communication round t. Each step of F–RG can be decomposed as a perturbed gossip update (cf. Lemma 4.C.1),

$$\boldsymbol{X}^{t+1}_\mathcal{H} = (\boldsymbol{I}_\mathcal{H} - \eta \boldsymbol{W}_\mathcal{H}) \boldsymbol{X}^t_\mathcal{H} + \eta \boldsymbol{E}^t. \tag{4.4}$$

Where $\boldsymbol{E}^t$ is the perturbation term due to Byzantine nodes. In the following, we assume that $[\boldsymbol{E}^t]_i = \zeta^t_i \boldsymbol{a}^t_i$ for any honest node i, where $\boldsymbol{a}^t_i$ is the direction of attack on node i, and ζ^t_i is a scaling factor, which is chosen to bypass the defenses. Typically, if ζ^t_i is small, a Byzantine node $j \in n_\mathcal{B}(i)$ can declare to node i the parameter $\boldsymbol{x}^t_j = \boldsymbol{x}^t_i + \zeta^t_i \boldsymbol{a}^t_i$.

Dissensus Attack. Byzantine nodes can disrupt decentralized communication by maximizing the variance of the honest parameters. A natural decentralized notion of variance is the *Laplacian heterogeneity* $\sum_{i,j \in \mathcal{H}} w_{ij} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$, which corresponds to $\|\boldsymbol{X}_\mathcal{H}\|^2_{\boldsymbol{W}_\mathcal{H}}$. Finding $\boldsymbol{a}^t_i$ such that this heterogeneity is maximized at $t+1$ writes

$$\begin{aligned} &\underset{[\boldsymbol{E}^t]_i = \zeta^t_i \boldsymbol{a}^t_i}{\arg\max} \|(\boldsymbol{I}_\mathcal{H} - \eta \boldsymbol{W}_\mathcal{H}) \boldsymbol{X}^t_\mathcal{H} + \eta \boldsymbol{E}^t\|^2_{\boldsymbol{W}_\mathcal{H}} \\ &\qquad = \underset{[\boldsymbol{E}^t]_i = \zeta^t_i \boldsymbol{a}^t_i}{\arg\max} 2\eta \langle \boldsymbol{W}_\mathcal{H} \boldsymbol{X}^t_\mathcal{H}, \boldsymbol{E}^t \rangle + o(\eta^2). \end{aligned}$$

For small η, this suggests to take $\boldsymbol{a}^t_i = [\boldsymbol{W}^t_\mathcal{H} \boldsymbol{X}^t_\mathcal{H}]_i = \sum_{j \in n_\mathcal{H}(i)} w_{ij} (\boldsymbol{x}^t_i - \boldsymbol{x}^t_j)$. This choice of $\boldsymbol{a}^t_i$ corresponds to the *Dissensus* attack proposed in He et al. (2023). However, as gossip communication is usually operated with multiple communication rounds, maximizing only the pairwise differences at the next step is a short-sighted approach.

Spectral Heterogeneity Attack. Byzantine nodes can take into account the fact that several rounds of communication will occur, and focus on increasing the heterogeneity over the long term. This leads, at any time t, to maximizing for any $s \geq 0$ the pairwise differences at time $t+s$, i.e, finding

$$\underset{[\boldsymbol{E}^t]_i = \zeta^t_i \boldsymbol{a}^t_i}{\arg\max} 2\eta \langle \boldsymbol{W}_\mathcal{H} (\boldsymbol{I}_\mathcal{H} - \eta \boldsymbol{W}_\mathcal{H})^{2s+1} \boldsymbol{X}^t_\mathcal{H}, \boldsymbol{E}^t \rangle + o(\eta^2).$$

Taking $s \to +\infty$ leads to approximating $\boldsymbol{W}_\mathcal{H}(\boldsymbol{I}_\mathcal{H} - \eta \boldsymbol{W}_\mathcal{H})^{2s}$ as a projection on its eigenspace associated with the largest eigenvalue of $\boldsymbol{W}_\mathcal{H}(\boldsymbol{I}_\mathcal{H} - \eta \boldsymbol{W}_\mathcal{H})^{2s}$. This eigenspace corresponds to the space spanned by the eigenvector of $\boldsymbol{W}_\mathcal{H}$ associated with the smallest non-zero eigenvalue of $\boldsymbol{W}_\mathcal{H}$, i.e $\mu_2(\mathcal{G}_\mathcal{H})$. This eigenvector (denoted $\boldsymbol{e}_{fied}$) is commonly referred to

as the *Fiedler vector* of the graph. Its coordinates essentially sort the nodes of the graph with the two farthest nodes associated with the largest and smallest value. Hence, the signs of the values in the Fiedler vector are typically used to partition the graph into two (least-connected) components. Our *Spectral Heterogeneity* attack consists in taking $\boldsymbol{a}_i^t = [\boldsymbol{e}_{fied}\boldsymbol{e}_{fied}^T\boldsymbol{X}_{\mathcal{H}}^t]_i$, which leads Byzantine nodes to cut the graph into two by pushing honest nodes in either plus or minus $\boldsymbol{e}_{fied}^T\boldsymbol{X}_{\mathcal{H}}^t$.

4.6 Experimental Evaluation

We follow Farhadkhani et al. (2023) (on which the core of our code is based), and present results for classification tasks on MNIST and CIFAR-10 datasets, as well as plain averaging tasks. We refer to Section 4.B.2 for most of the experiments. Similarly to Farhadkhani et al. (2023), heterogeneity is simulated by sampling data from each class using a Dirichlet distribution of parameter α. We test the attacks Spectral Heterogeneity (SpH), Dissensus, A Little Is Enough (ALIE) (Baruch et al., 2019), and Fall of Empire (FOE) (Xie et al., 2020). The main differences with Farhadkhani et al. (2023) are the following.

1. We consider sparse communication networks.
2. We implement GTS–RG instead of NNA, and implement ClippedGossip (aka $\mathrm{CS}_{\mathrm{He}}$–RG) using the adaptive clipping rule of He et al. (2023) instead of a fixed threshold. We additionally implement IOS (Wu et al., 2023).
3. We add Dissensus and Spectral Heterogeneity attacks.
4. We modify the generic design of attacks to better adapt it to the decentralized setting.

See Section 4.B for a detailed experimental setup and our implementation available at `https://github.com/renaudgaucher/Byzantine-Robust-Gossip`.

In Figure 4.1, it appears that the SpH attack is more efficient in disrupting ClippedGossip, GTS–RG and IOS than Dissensus and ALIE, and that CS_{+}–RG is highly resilient in the setup considered. In this setting with a rather simple learning task, the connectivity of the graphs appears as the major limiting factor to the robustness of distributed algorithms, hence why Spectral Heterogeneity is very efficient.

4.7 Conclusion

This paper revisits robust averaging over sparse communication graphs. We introduce a general framework for robust decentralized averaging, which allows us to derive tight convergence guarantees for many robust summation rules. In particular, we introduce one that nearly matches an upper bound on the breakdown point, *i.e.*, the maximum number of Byzantine nodes an algorithm can tolerate. Our experiments confirm that our theory correctly sorts the breakdown points of the existing methods, and that some (such as NNA) fail before the optimal breakdown point. We introduce a new *Spectral Heterogeneity* attack that exploits the graph topology for sparse graphs to obtain this result.

An interesting future direction is the characterization of robustness when the constraint on the number of neighbors cannot be met globally, but convergence can be obtained within local neighborhoods. Conversely, this opens up questions on which nodes an attacker should corrupt to maximize their influence for a specific graph, in light of our results.

Appendix

4.A Additional Discussion

4.A.1 Asynchronous Communications

In the core of this paper, we assumed that all communications were synchronous. However, our F–RG framework can be readily adapted to operate in less synchronous settings.

Suppose that communication still occurs in rounds, but messages take a variable amount of time to be delivered. If each honest node waits to receive messages from all its neighbors before performing an aggregation step, then Byzantine nodes can prevent the honest nodes from updating simply by withholding their messages. To be robust against such behavior, the nodes should not wait for all messages before updating their parameters.

The F–RG framework adapts naturally to this setting. Specifically, consider a (ρ, b)-robust summand F, and assume that each honest node i performs the F–RG update as soon as it has received all messages except those corresponding to a total weight of at most b. Then, this asynchronous version of F–RG remains $(1 - \eta(\mu_{\min} - 4\rho b))$-robust, as the following proposition demonstrates.

Proposition 4.A.1. *Let $F : (\mathbb{R}_+ \times \mathbb{R}^d)^n \to \mathbb{R}^d$ be a (b, ρ)–robust summation. Let $F_{asyn.}$ denote the rule that applies F to all inputs $(w_i, \boldsymbol{x}_i)_{i\in[n]}$ excluding an arbitrary subset $S_{\text{delayed}} \subset [n]$ of smaller than b, i.e. $\sum_{i\in S_{\text{delayed}}} w_i \leq b$. Then $F_{asyn.}$ is a $(b, 2\rho)$–robust summation rule.*

Proof. Using the triangle inequality, $(a+b)^2 \leq 2(a^2+b^2)$ and Jensen inequality and the definition of robust summation yields

$$
\begin{aligned}
&\left\| F_{asyn.}\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]}\big) - \sum_{i\in S} \omega_i \boldsymbol{z}_i \right\|^2 \\
&= \left\| F\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]\setminus S_{\text{delayed}}}\big) - \sum_{i\in[n]\setminus S_{\text{delayed}}} \omega_i \boldsymbol{z}_i - \sum_{i\in S\cap S_{\text{delayed}}} \omega_i \boldsymbol{z}_i \right\|^2 \\
&\leq \left(\left\| F\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]\setminus S_{\text{delayed}}}\big) - \sum_{i\in[n]\setminus S_{\text{delayed}}} \omega_i \boldsymbol{z}_i \right\| + \left\| \sum_{i\in S\cap S_{\text{delayed}}} \omega_i \boldsymbol{z}_i \right\| \right)^2 \\
&\leq 2 \left(\left\| F\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]\setminus S_{\text{delayed}}}\big) - \sum_{i\in[n]\setminus S_{\text{delayed}}} \omega_i \boldsymbol{z}_i \right\|^2 + \left\| b \sum_{i\in S\cap S_{\text{delayed}}} \frac{\omega_i}{b} \boldsymbol{z}_i \right\|^2 \right)
\end{aligned}
$$

$$\leq 2\left(\rho b \sum_{i\in S\setminus S_{\text{delayed}}} \omega_i\|\boldsymbol{z}_i\|^2 + b\sum_{i\in S\cap S_{\text{delayed}}} \omega_i\|\boldsymbol{z}_i\|^2\right).$$

Now, since necessarily $\rho \geq 1$, it finally yields

$$\left\|F_{asyn.}\big((\omega_i,\boldsymbol{z}_i)_{i\in[n]}\big) - \sum_{i\in S}\omega_i\boldsymbol{z}_i\right\|^2 \leq 2\rho b\sum_{i\in S}\omega_i\|\boldsymbol{z}_i\|^2.$$

□

Remark 4.A.2. In the above result, we used the fact that there exists no (b,ρ)-robust summand with $\rho < 1$, as shown in Theorem 4.3.5.

Remark 4.A.3. The latter proof can be refined using $(a+b)^2 \leq (1+\varepsilon)a^2 + (1+\varepsilon^{-1})b^2$ with an optimal choice of ε, showing that $F_{asyn.}$ is at least $(b,\rho+\sqrt{\rho}))$-robust.

4.A.2 Choosing the Graph's Weights

Our analysis requires that each edge of the graph be associated with a nonnegative weight and that the communication step size satisfies $\eta \leq 1/\mu_{\max}(\mathcal{G}_{\mathcal{H}})$. Although unitary weights $w_{ij} = 1$ are convenient for identifying b as an upper bound on the *number of Byzantine neighbors*, they require adjusting the step size η using global information from the graph, such as $\mu_{\max}(\mathcal{G})^{-1}$. A practical way to circumvent this is to use bistochastic weights.

- **Generic Bistochastic Matrix.** Any bistochastic matrix $B \in [0,1]^{m\times m}$ can be used to define the weights of the graph using $w_{ij} = B_{ij}$. In this case, the Laplacian matrix is defined as $W = I - B$, and its largest eigenvalue is upper-bounded by 2. The communication step size can thus be chosen as $\eta = 1/2$. Note that, in such a case, the condition $\mu_2(\mathcal{G}_{\mathcal{H}}) \geq 2\rho b$ is implied by $\tilde{\gamma} \geq 2\rho b$, where $\tilde{\gamma}$ is generally named the *spectral gap* of the bistochastic matrix B, and is defined as $\tilde{\gamma} = 1 - \max_{\mu\in sp(B),\mu\neq 1}|\mu|$, where $sp(B)$ denotes the eigenvalues of the matrix B.

- **Metropolis-Hasting Weights (HASTINGS, 1970).** The Metropolis-Hasting algorithm constructs a bistochastic matrix by making each node i declare to their neighbors their degree d_i. Then, any pair of neighbors $(i,j) \in \mathbb{E}$ defines the weight on their edge as $w_{ij} = 1/\max(d_i,d_j)+1$. As pointed out in He et al. (2023) this algorithm is robust to corrupted nodes, since for $i \in \mathcal{H}$ and $j \in B$ the influence of j on i is bounded by $w_{ij} \leq \frac{1}{d_i+1}$. Interestingly, since the size of the communication step can be chosen as $\eta = 1/2$ without further knowledge of the global network properties, this choice of weights requires only local information to carry out the communication.

4.B Experiments

4.B.1 Detailed Experimental Setup

4.B.1.1 Attack Design

Our experimental setting is built on top of the code provided by Farhadkhani et al. (2023), with the following differences:

1. Each honest node receives different messages from Byzantine nodes: for an honest node $i \in \mathcal{H}$, the Byzantine node $j \in n_{\mathcal{B}}(i)$ declares to node i at time t the vector $\boldsymbol{x}_j^t = \boldsymbol{x}_i^t + \zeta_i^t \boldsymbol{a}_i^t$. The reference point taken is the parameter of node i, instead of the average of all parameters $\overline{\boldsymbol{x}}_{\mathcal{H}}^t$, as performed in (Farhadkhani et al., 2022). Indeed, $\overline{\boldsymbol{x}}_{\mathcal{H}}^t$ can be very far from the vectors in the honest neighborhood of node i since the network is not fully connected. Note that in opposition to (Farhadkhani et al., 2022), Byzantines declare *different* parameters to each of the honest nodes. Not only does it allow the use of attacks such as Dissensus and spectral heterogeneity (through the choice of $\boldsymbol{a}_i^t$), but it also allows us to tune ζ_i^t differently for each node.

2. Each scaling parameter ζ_i^t is designed separately through a linear search, such as to maximize for each honest node i

$$\left\| F\big((w_{ij}, \boldsymbol{x}_i - \boldsymbol{x}_j)_{j \in n(i)}\big) - \sum_{j \in n_{\mathcal{H}}(j)} w_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j) \right\|^2 .$$

3. The vector $\boldsymbol{a}_i^t$ is defined differently depending on the attack implemented: *Dissensus*, *Spectral Heterogeneity* (SpH), *Fall of Empire* (FOE) from Xie et al. (2020) or *A little is enough* (ALIE) from Baruch et al. (2019).

   - **Dissensus**. The Byzantines $j \in n_{\mathcal{H}}(i)$ take as attack vector $\boldsymbol{a}_i^t = [\boldsymbol{W}_{\mathcal{H}}^t \boldsymbol{X}_{\mathcal{H}}^t]_i = \sum_{j \in n_{\mathcal{H}}(i)} w_{ij}(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t)$.
   - **Spectral Heterogeneity**. The Byzantines $j \in n_{\mathcal{H}}(i)$ take as attack vector $\boldsymbol{a}_i^t = [\boldsymbol{e}_{fied} \boldsymbol{e}_{fied}^T \boldsymbol{X}_{\mathcal{H}}^t]_i$, where $\boldsymbol{e}_{fied}$ denotes an eigenvector of $\boldsymbol{W}_{\mathcal{H}}$ associated with $\mu_2(\boldsymbol{W}_{\mathcal{H}})$.
   - **ALIE**. The Byzantine nodes compute the mean of the honest parameters $\overline{\boldsymbol{x}}_{\mathcal{H}}^t$ and the coordinate-wise standard deviation $\boldsymbol{\sigma}^t$. Then they use the attack vector $\boldsymbol{a}_i^t = \boldsymbol{\sigma}^t$.
   - **FOE**. The Byzantine nodes uses $\boldsymbol{a}_i^t = -\overline{\boldsymbol{x}}_{\mathcal{H}}^t$.

Remark 4.B.1. In the case of trimming base rules, a badly designed attack leads Byzantine messages to be removed during aggregations, which induces the resulting algorithm to behave as a plain non-corrupted D-SGD algorithm. Thus, proposing non-over-confident experimental proofs of trimming-based aggregation requires a fine design of the attacks.

4.B.1.2 Algorithms Tested

Networks. Two topologies of the honest subgraph are investigated: 1) A "Two Worlds" graph, i.e. $\mathcal{G}_{\mathcal{H}} := [\mathcal{G}_{m=13,k=8,c=1}]_{\mathcal{H}}$. 2) Randomly sampled Erdos-Renyi graphs, with a varying probability of edge presence p. All graphs are equipped with unitary weights on the edges, for simplicity.

Communications. We compare

- GTS-RG;
- CS_+-RG;
- The version of ClippedGossip proposed in He et al. (2023) with an adaptive clipping rule which is not supported by any theory. Precisely, ClippedGossip is equivalent to $\mathrm{CS}_{\mathrm{He}}$-RG, with $\mathrm{CS}_{\mathrm{He}} := \mathrm{CS}\big(\cdot; \tau_{\mathrm{He}}\big)$, where, for $\|\boldsymbol{z}_1\| \leq \ldots \leq \|\boldsymbol{z}_n\|$, $\tau_{\mathrm{He}}\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]}\big) = \sqrt{\frac{1}{b}\sum_{i\leq|n_{\mathcal{H}}(i)|}\omega_i\|\boldsymbol{z}_i\|^2}$,
- Iterative Outlier Scissors (IOS), from (Wu et al., 2023).

F-RG based methods uses $\eta = \mu_{\max}(\mathcal{G}_{\mathcal{H}})^{-1}$.

Optimization. All learning experiments implement Algorithm 1, and only the communication part changes among experiments. It follows that, even though IOS is not explicitly combined with momentum in (Wu et al., 2023), we still implement it with momentum. We do this to have a fairer comparison between communication routines since (Farhadkhani et al., 2023) showed that momentum is key for robustness.

4.B.1.3 Dataset Pre-Processing

MNIST images receive an input image normalization of mean 0.1307 and standard deviation 0.3081. The images of CIFAR-10 are horizontally flipped, and a per-channel normalization is applied with means $(0.4914, 0.4822, 0.4465)$, and standard deviation $(0.2023, 0.1994, 0.2010)$.

4.B.1.4 Data Heterogeneity

We simulate data heterogeneity in the correct nodes' datasets following the method of (Farhadkhani et al., 2023) by making nodes sample from each class of the considered dataset (MNIST or CIFAR-10) using a Dirichlet distribution of parameter $\alpha > 0$: the smaller α, the more probable it is to sample from one class only.

4.B.1.5 Model Architecture and Hyper Parameters

To present the detailed architecture of the models used, we adopt the following compact notation:

L(#outputs) represents a **fully-connected linear layer**, C(output channels) represents a **2D-convolutional layer** of kernel size 3 and padding 1, R stands for **ReLU activation**, B stands for **batch-normalization**, and D represents **dropout** with probability 0.25, S stands for **log-softmax** and NLL for **negative log-likelihood loss**.

The architecture of the model used and the experimental setup are proposed in Table 4.1.

Table 4.1: Detailed experimental setting

Dataset	MNIST		CIFAR-10
Model type	CNN		CNN
Model architecture	C(16)-R-M-L(10)-S		C(32)-B-R-M-C(64)-B-R-M-C(128)-B-R-D-L(128)-R-D-L(10)-S
Loss	NLL		NLL
Batch size	64		64
Learning rate	$\eta_{op} = 0.1$		$\eta_{op} = 0.5$
Momentum	$\beta = 0.9$		$\beta = 0.99$
Number of Iterations	$T = 300$		$T = 5000$
Number of honest nodes	$\lvert\mathcal{H}\rvert = 26$	$\lvert\mathcal{H}\rvert = 20$	$\lvert\mathcal{H}\rvert = 16$
Graph	Two Worlds	Erdös Renyi	Two Worlds
Graph parameter	$k = 8$	$p \in [0.26, 1]$	$k = 6$
Data Heterogeneity	$\alpha = 1$	$\alpha = 1$	$\alpha = 5$
Byzantine weight	$b \in \{1, \dots, 11\}$	$b = 3$	$b = 3$
Number of seeds	1	1	1

4.B.2 Additional Empirical Results

In Section 4.B.2.1 we provide experiments with two world graphs taken as a communication network, both on learning tasks with MNIST and CIFAR-10 datasets, and on an averaging problem. On both the MNIST and averaging tasks, we use a varying amount of Byzantine weight to investigate the empirical robustness of each algorithm.

In Section 4.B.2.2 we provide experiments on Erdos Renyi networks on MNIST and an averaging task. We study here the influence of the connectivity of the network on the robustness by varying the probability of each pair of honest nodes being connected.

4.B.2.1 Experiments on 'Two Worlds' Graphs

For both the Averaging experiments and the MNIST experiments, we fixed the subgraph of honest nodes to be $\mathcal{G}_{\mathcal{H}} := [\mathcal{G}_{m=13,k=8,c=1}]_{\mathcal{H}}$, and the weight of Byzantine b varies. Note that Theorem 4.3.5 predicts that, on this graph, no algorithm can be r-robust when $b > 8$.

Averaging experiments (Figure 4.2). Recall that $\mathcal{G}_{\mathcal{H}} := [\mathcal{G}_{m=13,k=8,c=1}]_{\mathcal{H}}$, is built on two fully connected cliques of 13 honest nodes, which are then connected. Here we initialize the parameters of honest nodes with a $\mathcal{N}(\boldsymbol{u}, I_d/d)$ distribution ($d = 5$), where $\boldsymbol{u}$ is equal to $+(5, 0, \dots, 0)^T$ for one of the two cliques, and equal to $-(5, 0, \dots, 0)^T$ for the other one. For each setting (b, Communication Algorithm, Attack), we perform experiments with 6 random seeds. The error plotted correspond to the mean square error $\sum_{i\in\mathcal{H}} \|\boldsymbol{x}_i^t - \overline{\boldsymbol{x}}_{\mathcal{H}}^0\|^2 / \sum_{i\in\mathcal{H}} \|\boldsymbol{x}_i^0 - \overline{\boldsymbol{x}}_{\mathcal{H}}^0\|^2$ achieved after 100 communication steps. The line corresponds to the average value among seeds, while the confidence interval corresponds to the maximum and minimal values encountered.

MNIST experiments (Figure 4.3). Experiments are conducted with a heterogeneity parameter $\alpha = 1$. The error and accuracy displayed are after 300 iterations. Further

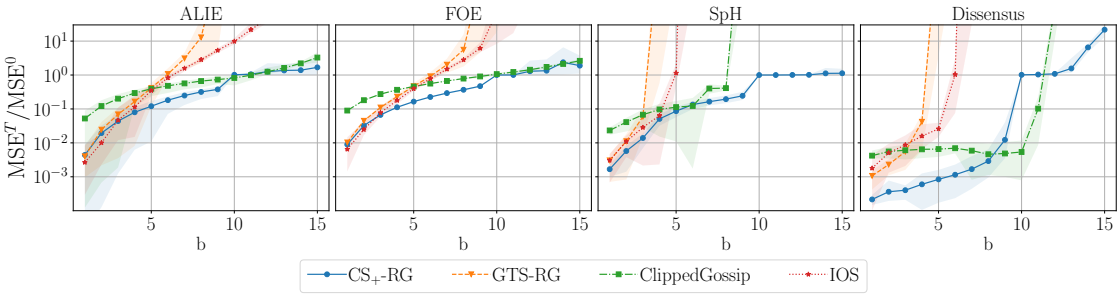


Figure 4.2: Relative MSE on an averaging task after 100 communication steps on a Two-Wold graph, with a varying weight of Byzantines b. Here $\mu_2(\mathcal{G}_\mathcal{H}) = 16$.

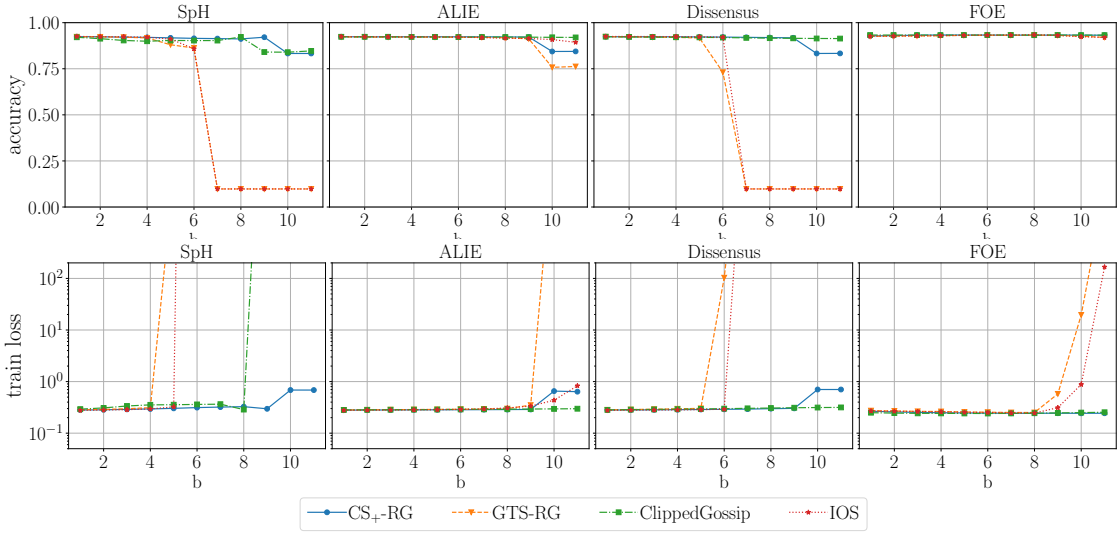


Figure 4.3: Accuracy and training loss on MNIST 300 optimization and communication steps, a Two Worlds graphs, with a varying weight of Byzantines neighbors. Here $\mu_2(\mathcal{G}_\mathcal{H}) = 16$, and $\alpha = 1$.

experimental details are in Table 4.1.

CIFAR-10 experiments (Figures 4.4 and 4.5). Experiments are conducted on CIFAR-10 Dataset with an heterogeneity parameters $\alpha = 5$, on a Two Worlds graph $\mathcal{G}_\mathcal{H} := [\mathcal{G}_{m=8,k=6,c=1}]_\mathcal{H}$ with $b = 1$. Further experimental details are in Table 4.1. We provide both test accuracy and train loss since the dynamic of the train loss does not always impact the test accuracy, specifically in the case of Spectral Heterogeneity and Dissensus attacks.

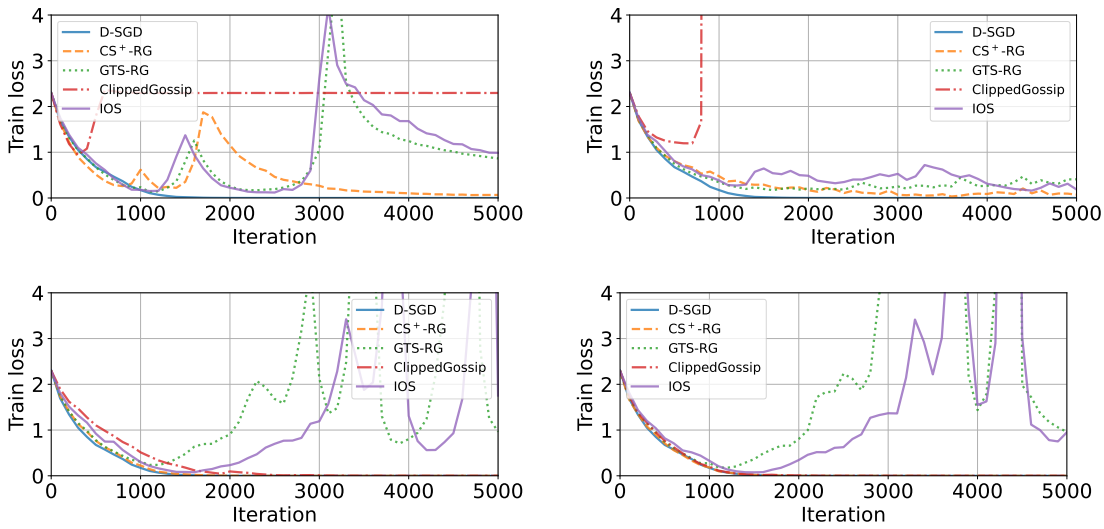


Figure 4.4: Train loss on CIFAR-10 ($\alpha = 5$) on a Two Worlds graph with $\mu_2(\mathcal{G}_\mathcal{H}) = 12$ and $b = 1$. Attacks tested are FOE (upper left), ALIE (upper right), SpH (lower left), and Dissensus (lower right).

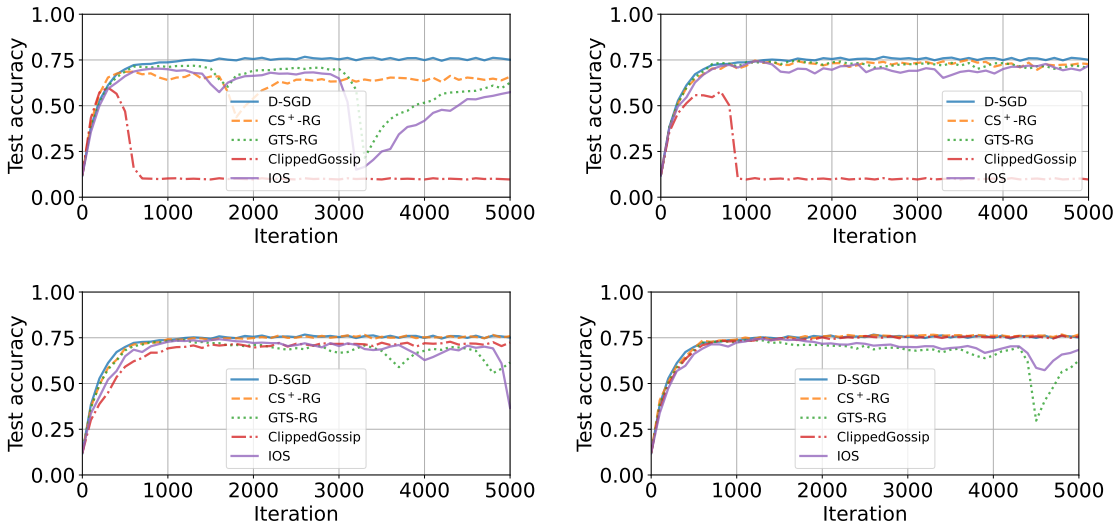


Figure 4.5: Accuracy on CIFAR-10 ($\alpha = 5$) on a Two Worlds graph with $\mu_2(\mathcal{G}_\mathcal{H}) = 12$ and $b = 1$. Attacks tested are FOE (upper left), ALIE (upper right), SpH (lower left), and Dissensus (lower right).

4.B.2.2 Experiments on Erdos-Renyi Graphs

Experiments are conducted by using, as a subgraph of honest nodes, a random Erdos Renyi graph with 20 honest nodes. Each honest node is always adjacent to 4 Byzantine nodes. On each seed, we test 12 different values of $p \in [0.25, 1]$, where p denotes the probability of an edge to exist. We plot the links between the algebraic connectivity of the graph (denoted μ_2, the second smallest eigenvalue of the unitary weighted Laplacian) and the losses.

Averaging task (Figure 4.6). Nodes' parameters are initialized using a $N(0, I_5)$ distribution. Nodes perform 100 (robust) gossip communication iterations, and the gain in

terms of mean square error is plotted. Experiments are conducted on 6 different seeds, and curves are smoothed using a moving average of size 4.

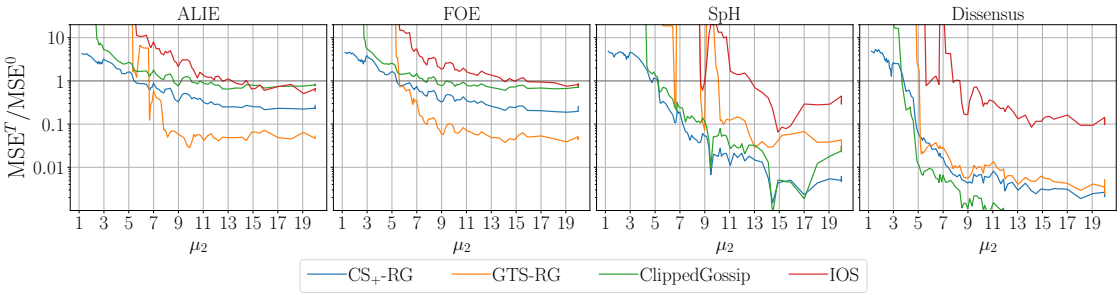


Figure 4.6: Averaging task. Relative MSE after 100 communication steps on randomly sampled Erdos-Renyi graphs with a fixed $b = 4$.

MNIST (Figure 4.7). The heterogeneity among nodes is set to $\alpha = 1$. All experiments run on the same seed.

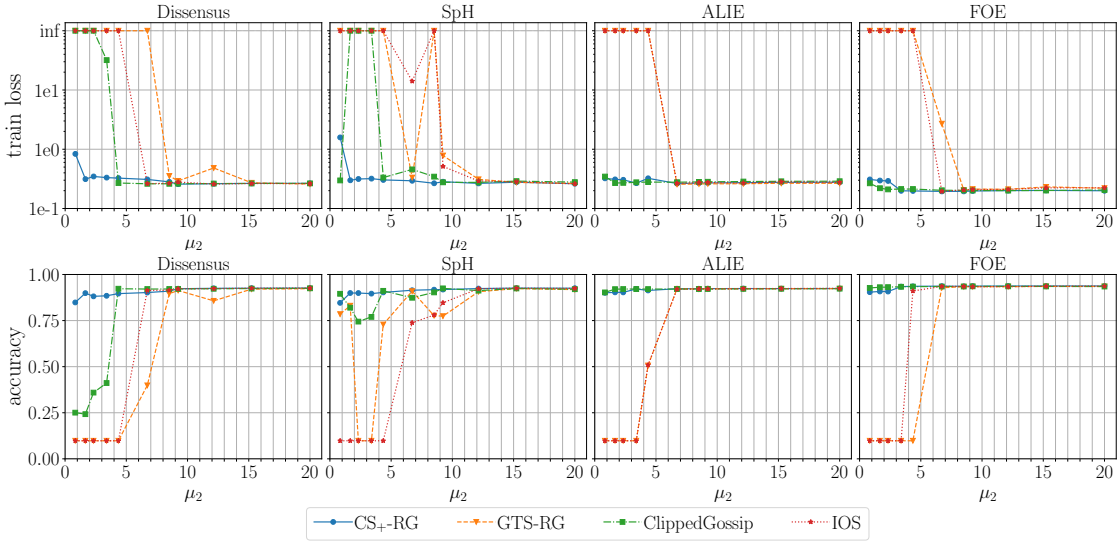


Figure 4.7: Accuracy and training loss on MNIST($\alpha = 1$) after 200 optimization and communication steps on randomly sampled Erdos-Renyi graphs with a fixed $b = 4$.

4.C Analysis of the Robust Gossip Method

4.C.1 Proof of Theorem 4.3.3

We now prove Theorem 4.3.3, and then use it to derive convergence for the Byzantine-robust decentralized stochastic gradient descent framework. Recall that nodes follow the update scheme below.

$$\begin{cases} \boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t - \eta F\left((w_{ij}; \boldsymbol{x}_j^t - \boldsymbol{x}_i^t)_{j \in n(i)}\right) & \text{if } i \in \mathcal{H} \\ \boldsymbol{x}_i^{t+1} = * \quad \text{if } i \in \mathcal{B}, \end{cases} \tag{4.5}$$

Before proving Theorem 4.3.3, we recall the following notations:

- $w_{ij} = -\boldsymbol{W}_{ij} \geq 0$ denote the weight associated with the edge $i \sim j$ on the graph.
- The matrix of honest parameters $\boldsymbol{X}_{\mathcal{H}}^t := \begin{pmatrix} (\boldsymbol{x}_1^t)^T \\ \vdots \\ (\boldsymbol{x}_{|\mathcal{H}|}^t)^T \end{pmatrix} \in \mathbb{R}^{|\mathcal{H}| \times d}$.
- The error due to robust aggregation and Byzantine corruption:

$$\forall i \in \mathcal{H}, \quad [\boldsymbol{E}^t]_i := \sum_{j \in n_{\mathcal{H}}(i)} w_{ij}(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t) - F\left((w_{ij}; \boldsymbol{x}_j^t - \boldsymbol{x}_i^t)_{j \in n(i)}\right).$$

Lemma 4.C.1. *Equation (4.5) writes*

$$\boldsymbol{X}_{\mathcal{H}}^{t+1} = (\boldsymbol{I}_{\mathcal{H}} - \eta \boldsymbol{W}_{\mathcal{H}})\boldsymbol{X}_{\mathcal{H}}^t + \eta \boldsymbol{E}^t.$$

Proof. Let $i \in \mathcal{H}$. We decompose the update due to the gossip scheme and consider the error term coming from both robust aggregation and the influence of Byzantine nodes.

$$\begin{aligned} \boldsymbol{x}_i^{t+1} &= \boldsymbol{x}_i^t - \eta F\left((w_{ij}; \boldsymbol{x}_j^t - \boldsymbol{x}_i^t)_{j \in n(i)}\right) \\ &= \boldsymbol{x}_i^t - \eta \sum_{j \in n_{\mathcal{H}}(i)} w_{ij}(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t) + \eta \left(\sum_{j \in n_{\mathcal{H}}(i)} w_{ij}(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t) - F\left((w_{ij}; \boldsymbol{x}_j^t - \boldsymbol{x}_i^t)_{j \in n(i)}\right) \right). \end{aligned}$$

Finally, the proof is concluded by remarking that $[W_{\mathcal{H}} \boldsymbol{X}_{\mathcal{H}}^t]_i = \sum_{j \in n_{\mathcal{H}}(i)} w_{ij}(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t)$. □

We begin by controlling the norm of the error term $\|\boldsymbol{E}^t\|_2^2$ in the case of CS$_+$–RG.

Lemma 4.C.2 (Control of the error)**.** *Assume F is a (b, ρ) robust summation. Then the error is controlled by the heterogeneity as measured by the Laplacian matrix:*

$$\|\boldsymbol{E}^t\|_2^2 \leq 2\rho b \|\boldsymbol{X}_{\mathcal{H}}^t\|_{\boldsymbol{W}_{\mathcal{H}}}^2 = \rho b \sum_{i \in \mathcal{H}, j \in n_{\mathcal{H}}(i)} w_{ij} \|\boldsymbol{x}_i^t - \boldsymbol{x}_j^t\|^2.$$

Proof. We recall that in this case,

$$\forall i \in \mathcal{H}, \quad [\boldsymbol{E}^t]_i := \sum_{j \in n_{\mathcal{H}}(i)} w_{ij}(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t) - F\left((w_{ij}; \boldsymbol{x}_j^t - \boldsymbol{x}_i^t)_{j \in n(i)}\right).$$

By assumption, for all honest node $i \in \mathcal{H}$, the weight of Byzantine in his neighborhood is smaller than b, i.e $\sum_{j \in n_{\mathcal{B}}(i)} w_{ij} \leq b$. Thus, applying the (b, ρ) robustness of F yields

$$\begin{aligned} \|\boldsymbol{E}^t\|^2 &= \sum_{i \in \mathcal{H}} \left\| \sum_{j \in n_{\mathcal{H}}(i)} w_{ij}(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t) - F\left((w_{ij}; \boldsymbol{x}_j^t - \boldsymbol{x}_i^t)_{j \in n(i)}\right) \right\|_2^2 \\ &\leq \sum_{i \in \mathcal{H}} \rho b \sum_{j \in n_{\mathcal{H}}(i)} w_{ij} \|\boldsymbol{x}_i^t - \boldsymbol{x}_j^t\|^2 \end{aligned}$$

$$= 2\rho b \|\boldsymbol{X}_{\mathcal{H}}^t\|_{\boldsymbol{W}_{\mathcal{H}}}^2,$$

Where the last equality follows by noting that $2\|\boldsymbol{X}_{\mathcal{H}}^t\|_{\boldsymbol{W}_{\mathcal{H}}}^2 = \sum_{i\in\mathcal{H}, j\in n_{\mathcal{H}}(i)} w_{ij}\|\boldsymbol{x}_i^t - \boldsymbol{x}_j^t\|^2$. Indeed, considering that $\mathcal{G}_{\mathcal{H}}$ is an undirected graph, $i \in n_{\mathcal{H}}(j) \iff j \in n_{\mathcal{H}}(i)$ and we have:

$$\begin{aligned}
\|\boldsymbol{X}_{\mathcal{H}}^t\|_{\boldsymbol{W}_{\mathcal{H}}}^2 &= \langle \boldsymbol{X}_{\mathcal{H}}, \boldsymbol{W}_{\mathcal{H}}\boldsymbol{X}_{\mathcal{H}}\rangle \\
&= \sum_{i\in\mathcal{H}} \left\langle \boldsymbol{x}_i^t, \sum_{j\in n_{\mathcal{H}}(i)} w_{ij}(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t)\right\rangle \\
&= \sum_{i\in\mathcal{H}}\sum_{j\in n_{\mathcal{H}}(i)} w_{ij}\left\langle \boldsymbol{x}_i^t, \boldsymbol{x}_i^t - \boldsymbol{x}_j^t\right\rangle \\
&= \frac{1}{2}\sum_{i\in\mathcal{H}}\sum_{j\in n_{\mathcal{H}}(i)} w_{ij}\left\langle \boldsymbol{x}_i^t - \boldsymbol{x}_j^t, \boldsymbol{x}_i^t - \boldsymbol{x}_j^t\right\rangle \\
\|\boldsymbol{X}_{\mathcal{H}}^t\|_{\boldsymbol{W}_{\mathcal{H}}}^2 &= \frac{1}{2}\sum_{i\in\mathcal{H},\, j\in n_{\mathcal{H}}(i)} w_{ij}\left\|\boldsymbol{x}_i^t - \boldsymbol{x}_j^t\right\|_2^2.
\end{aligned}$$

□

Now that we control the error term, we can conclude the proof of Theorem 4.3.3 using standard optimization arguments. Before proving this theorem, we prove the following one, from which Corollary 4.3.4 is direct.

Theorem 4.C.3. *Assume F is a (b,ρ) robust summation, and* RG *is the associated robust gossip algorithm. Let b and $\mu_{\min}$ be such that $2\rho b \leq \mu_{\min}$, and let $\mathcal{G} \in \Gamma_{\mu_{\min},b}$. Then, for any $\eta \leq \mu_{\max}(\mathcal{G}_{\mathcal{H}})^{-1}$, the output $\boldsymbol{y} = \mathrm{RG}(\boldsymbol{x})$ (obtained by one step of* RG *on $\mathcal{G}$ from $\boldsymbol{x}$) verifies:*

$$\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{x}_i^{t+1} - \overline{\boldsymbol{x}}_{\mathcal{H}}^{t+1}\|^2 \leq (1 - \eta\,(\mu_{\min} - 2\rho b))\,\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{x}_i^t - \overline{\boldsymbol{x}}_{\mathcal{H}}^t\|^2 \tag{4.6}$$

$$\|\overline{\boldsymbol{x}}_{\mathcal{H}}^{t+1} - \overline{\boldsymbol{x}}_{\mathcal{H}}^t\|^2 \leq \eta\frac{2\rho b}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{x}_i^t - \overline{\boldsymbol{x}}_{\mathcal{H}}^t\|^2. \tag{4.7}$$

Proof.

Part I: Equation (4.7).

Equation (4.7) is a direct consequence of Lemma 4.C.2. Indeed applying $\boldsymbol{P}_{\mathbf{1}_{\mathcal{H}}} := \frac{1}{|\mathcal{H}|}\mathbf{1}_{\mathcal{H}}\mathbf{1}_{\mathcal{H}}^T$ - the orthogonal projection on the kernel of $\boldsymbol{W}_{\mathcal{H}}$ - on Lemma 4.C.1 results in

$$\boldsymbol{P}_{\mathbf{1}_{\mathcal{H}}}\boldsymbol{X}_{\mathcal{H}}^{t+1} = \boldsymbol{P}_{\mathbf{1}_{\mathcal{H}}}(\boldsymbol{I}_{\mathcal{H}} - \eta\boldsymbol{W}_{\mathcal{H}})\boldsymbol{X}_{\mathcal{H}}^t + \eta\boldsymbol{P}_{\mathbf{1}_{\mathcal{H}}}\boldsymbol{E}^t = \boldsymbol{P}_{\mathbf{1}_{\mathcal{H}}}\boldsymbol{X}_{\mathcal{H}}^t + \eta\boldsymbol{P}_{\mathbf{1}_{\mathcal{H}}}\boldsymbol{E}^t.$$

Taking the norm yields

$$\|\boldsymbol{P}_{\mathbf{1}_{\mathcal{H}}}\boldsymbol{X}_{\mathcal{H}}^{t+1} - \boldsymbol{P}_{\mathbf{1}_{\mathcal{H}}}\boldsymbol{X}_{\mathcal{H}}^t\|^2 = \eta^2\|\boldsymbol{P}_{\mathbf{1}_{\mathcal{H}}}\boldsymbol{E}^t\|^2 \leq \eta^2\|\boldsymbol{E}^t\|^2. \tag{4.8}$$

We now apply Lemma 4.C.2, and use that $\mu_{\max}(\mathcal{G}_\mathcal{H})$ is the largest eigenvalue of $\boldsymbol{W}_\mathcal{H}$. It gives

$$\begin{aligned}\|\boldsymbol{P}_{\mathbf{1}_\mathcal{H}}\boldsymbol{X}_\mathcal{H}^{t+1} - \boldsymbol{P}_{\mathbf{1}_\mathcal{H}}\boldsymbol{X}_\mathcal{H}^{t}\|^2 &\leq \eta^2 2\rho b \|\boldsymbol{X}_\mathcal{H}^t\|^2_{\boldsymbol{W}_\mathcal{H}} \\ &\leq \mu_{\max}(\mathcal{G}_\mathcal{H})\eta^2 2\rho b \|(\boldsymbol{I}_\mathcal{H} - \boldsymbol{P}_{\mathbf{1}_\mathcal{H}})\boldsymbol{X}_\mathcal{H}^t\|^2\end{aligned}$$

Finally, Equation (4.7) derives from $[\boldsymbol{P}_{\mathbf{1}_\mathcal{H}}\boldsymbol{X}_\mathcal{H}^t]_{i\in\mathcal{H}} = [\sum_{j\in\mathcal{H}}\boldsymbol{x}_j^t]_{i\in\mathcal{H}} = [\overline{\boldsymbol{x}}_\mathcal{H}^t]_{i\in\mathcal{H}}$ and $\eta\mu_{\max}(\mathcal{G}_\mathcal{H}) \leq 1$.

Part II: Equation (4.6).

To prove Equation (4.6), we consider the objective function $\|(\boldsymbol{I}_\mathcal{H} - \boldsymbol{P}_{\mathbf{1}_\mathcal{H}})\boldsymbol{X}^t\|^2$. We denote by $\boldsymbol{W}_\mathcal{H}^\dagger$ the Moore-Penrose pseudo inverse of $\boldsymbol{W}_\mathcal{H}$. We begin by applying Lemma 4.C.1.

$$\begin{aligned}&\|(\boldsymbol{I}_\mathcal{H}-\boldsymbol{P}_{\mathbf{1}_\mathcal{H}})\boldsymbol{X}_\mathcal{H}^{t+1}\|^2 \\ &\quad=\|\boldsymbol{X}_\mathcal{H}^t - \eta\boldsymbol{W}_\mathcal{H}\boldsymbol{X}_\mathcal{H}^t + \eta\boldsymbol{E}^t\|^2_{(\boldsymbol{I}_\mathcal{H}-\boldsymbol{P}_{\mathbf{1}_\mathcal{H}})} \\ &\quad=\|\boldsymbol{X}_\mathcal{H}^t\|^2_{(\boldsymbol{I}_\mathcal{H}-\boldsymbol{P}_{\mathbf{1}_\mathcal{H}})} - 2\eta\left\langle \boldsymbol{X}_\mathcal{H}^t, \boldsymbol{W}_\mathcal{H}\boldsymbol{X}_\mathcal{H}^t - \boldsymbol{E}^t\right\rangle_{(\boldsymbol{I}_\mathcal{H}-\boldsymbol{P}_{\mathbf{1}_\mathcal{H}})} \\ &\qquad+\eta^2\left\|\boldsymbol{W}_\mathcal{H}\boldsymbol{X}_\mathcal{H}^t - \boldsymbol{E}^t\right\|_{(\boldsymbol{I}_\mathcal{H}-\boldsymbol{P}_{\mathbf{1}_\mathcal{H}})} \\ &\quad=\|\boldsymbol{X}_\mathcal{H}^t\|^2_{(\boldsymbol{I}_\mathcal{H}-\boldsymbol{P}_{\mathbf{1}_\mathcal{H}})} - 2\eta\left\langle \boldsymbol{X}_\mathcal{H}^t, \boldsymbol{X}_\mathcal{H}^t - \boldsymbol{W}_\mathcal{H}^\dagger\boldsymbol{E}^t\right\rangle_{\boldsymbol{W}_\mathcal{H}} \\ &\qquad+\eta^2\left\|\boldsymbol{X}_\mathcal{H}^t - \boldsymbol{W}_\mathcal{H}^\dagger\boldsymbol{E}^t\right\|_{\boldsymbol{W}_\mathcal{H}^2}.\end{aligned}$$

Applying $2\langle \boldsymbol{a}, \boldsymbol{b}\rangle = \|\boldsymbol{a}\|^2 + \|\boldsymbol{b}\|^2 - \|\boldsymbol{a}-\boldsymbol{b}\|^2$ leads to

$$\begin{aligned}\|\boldsymbol{X}_\mathcal{H}^{t+1}\|^2_{(\boldsymbol{I}_\mathcal{H}-\boldsymbol{P}_{\mathbf{1}_\mathcal{H}})} - \|\boldsymbol{X}_\mathcal{H}^{t}\|^2_{(\boldsymbol{I}_\mathcal{H}-\boldsymbol{P}_{\mathbf{1}_\mathcal{H}})} &= -\eta\left\|\boldsymbol{X}_\mathcal{H}^t\right\|^2_{\boldsymbol{W}_\mathcal{H}} - \eta\left\|\boldsymbol{X}_\mathcal{H}^t - \boldsymbol{W}_\mathcal{H}^\dagger\boldsymbol{E}^t\right\|^2_{\boldsymbol{W}_\mathcal{H}} \\ &\quad+\eta\left\|\boldsymbol{W}_\mathcal{H}^\dagger\boldsymbol{E}^t\right\|^2_{\boldsymbol{W}_\mathcal{H}} + \eta^2\left\|\boldsymbol{X}_\mathcal{H}^t - \boldsymbol{W}_\mathcal{H}^\dagger\boldsymbol{E}^t\right\|_{\boldsymbol{W}_\mathcal{H}^2} \\ &= -\eta\left\|\boldsymbol{X}_\mathcal{H}^t\right\|^2_{\boldsymbol{W}_\mathcal{H}} + \eta\left\|\boldsymbol{E}^t\right\|^2_{\boldsymbol{W}_\mathcal{H}^\dagger} \\ &\quad-\eta\left\|\boldsymbol{X}_\mathcal{H}^t - \boldsymbol{W}_\mathcal{H}^\dagger\boldsymbol{E}^t\right\|^2_{\boldsymbol{W}_\mathcal{H}} + \eta^2\left\|\boldsymbol{X}_\mathcal{H}^t - \boldsymbol{W}_\mathcal{H}^\dagger\boldsymbol{E}^t\right\|_{\boldsymbol{W}_\mathcal{H}^2}. \qquad (4.9)\end{aligned}$$

We now apply that $\mu_{\max}(\mathcal{G}_\mathcal{H})$ (resp. $\mu_2(\mathcal{G}_\mathcal{H})$) is the largest (resp. smallest) non-zero eigenvalue of $\boldsymbol{W}_\mathcal{H}$.

$$\begin{aligned}\|\boldsymbol{X}_\mathcal{H}^{t+1}\|^2_{(\boldsymbol{I}_\mathcal{H}-\boldsymbol{P}_{\mathbf{1}_\mathcal{H}})} - \|\boldsymbol{X}_\mathcal{H}^{t}\|^2_{(\boldsymbol{I}_\mathcal{H}-\boldsymbol{P}_{\mathbf{1}_\mathcal{H}})} &\leq -\eta\left\|\boldsymbol{X}_\mathcal{H}^t\right\|^2_{\boldsymbol{W}_\mathcal{H}} + \eta\frac{1}{\mu_2(\mathcal{G}_\mathcal{H})}\left\|\boldsymbol{E}^t\right\|^2 \\ &\quad-\eta(1-\mu_{\max}(\mathcal{G}_\mathcal{H})\eta)\left\|\boldsymbol{X}_\mathcal{H}^t - \boldsymbol{W}_\mathcal{H}^\dagger\boldsymbol{E}^t\right\|^2_{\boldsymbol{W}_\mathcal{H}}.\end{aligned}$$

Eventually Lemma 4.C.2 with the assumption $\eta \leq 1/\mu_{\max}(\mathcal{G}_\mathcal{H})$ yields the result

$$\|\boldsymbol{X}_\mathcal{H}^{t+1}\|^2_{(\boldsymbol{I}_\mathcal{H}-\boldsymbol{P}_{\mathbf{1}_\mathcal{H}})} \leq \|\boldsymbol{X}_\mathcal{H}^{t}\|^2_{(\boldsymbol{I}_\mathcal{H}-\boldsymbol{P}_{\mathbf{1}_\mathcal{H}})} - \eta\left(1 - \frac{2\rho b}{\mu_2(\mathcal{G}_\mathcal{H})}\right)\left\|\boldsymbol{X}_\mathcal{H}^t\right\|^2_{\boldsymbol{W}_\mathcal{H}}$$

$$\|\boldsymbol{X}_\mathcal{H}^{t+1}\|^2_{(\boldsymbol{I}_\mathcal{H}-\boldsymbol{P}_{\mathbf{1}_\mathcal{H}})} \leq \left(1 - \eta\mu_2(\mathcal{G}_\mathcal{H})\left(1 - \frac{2\rho b}{\mu_2(\mathcal{G}_\mathcal{H})}\right)\right)\left\|\boldsymbol{X}_\mathcal{H}^t\right\|^2_{(\boldsymbol{I}_\mathcal{H}-\boldsymbol{P}_{\mathbf{1}_\mathcal{H}})}.$$

□

To obtain Theorem 4.C.3, we note that we can actually control the one-step variation of the MSE using $(1-\eta(\mu_{\min}-2\rho b))$ only, thus strengthening the first inequality. We rewrite the first part of Theorem 4.3.3 below for completeness.

Corollary 4.C.4. *Assume F is a (b,ρ) robust summation, and* RG *is the associated robust gossip algorithm. Let b and $\mu_{\min}$ be such that $2\rho b \leq \mu_{\min}$, and let $\mathcal{G} \in \Gamma_{\mu_{\min},b}$. Then, for any $\eta \leq \mu_{\max}(\mathcal{G}_{\mathcal{H}})^{-1}$, the output $\boldsymbol{y} = \mathrm{RG}(\boldsymbol{x})$ (obtained by one step of* RG *on $\mathcal{G}$ from $\boldsymbol{x}$) verifies:*

$$\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{x}_i^{t+1}-\overline{\boldsymbol{x}}_{\mathcal{H}}^{t}\|^2 \leq (1-\eta\,(\mu_{\min}-2\rho b))\,\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{x}_i^{t}-\overline{\boldsymbol{x}}_{\mathcal{H}}^{t}\|^2$$

Proof. We consider Equation (4.8) and Equation (4.9), which write

$$\|\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}}\boldsymbol{X}_{\mathcal{H}}^{t+1}-\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}}\boldsymbol{X}_{\mathcal{H}}^{t}\|^2 = \eta^2\|\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}}\boldsymbol{E}^t\|^2.$$

$$\|\boldsymbol{X}_{\mathcal{H}}^{t+1}\|^2_{(\boldsymbol{I}_{\mathcal{H}}-\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}})}-\|\boldsymbol{X}_{\mathcal{H}}^{t}\|^2_{(\boldsymbol{I}_{\mathcal{H}}-\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}})} \leq -\eta\left\|\boldsymbol{X}_{\mathcal{H}}^{t}\right\|^2_{\boldsymbol{W}_{\mathcal{H}}}+\eta\left\|\boldsymbol{E}^t\right\|^2_{\boldsymbol{W}_{\mathcal{H}}^{\dagger}}.$$

It follows from the bias - variance decomposition of the MSE

$$\|\boldsymbol{X}_{\mathcal{H}}^{t+1}-\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}}\boldsymbol{X}_{\mathcal{H}}^{t}\|^2 = \|(\boldsymbol{I}_{\mathcal{H}}-\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}})\boldsymbol{X}_{\mathcal{H}}^{t+1}\|^2+\|\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}}\boldsymbol{X}_{\mathcal{H}}^{t+1}-\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}}\boldsymbol{X}_{\mathcal{H}}^{t}\|^2$$

that

$$\begin{aligned}\|\boldsymbol{X}_{\mathcal{H}}^{t+1}-\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}}\boldsymbol{X}_{\mathcal{H}}^{t}\|^2-&\|\boldsymbol{X}_{\mathcal{H}}^{t}\|^2_{(\boldsymbol{I}_{\mathcal{H}}-\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}})}\\ &\leq -\eta\left\|\boldsymbol{X}_{\mathcal{H}}^{t}\right\|^2_{\boldsymbol{W}_{\mathcal{H}}}+\eta\left\|\boldsymbol{E}^t\right\|^2_{\boldsymbol{W}_{\mathcal{H}}^{\dagger}}+\eta^2\|\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}}\boldsymbol{E}^t\|^2\\ &\leq -\eta\left\|\boldsymbol{X}_{\mathcal{H}}^{t}\right\|^2_{\boldsymbol{W}_{\mathcal{H}}}+\eta\frac{1}{\mu_2(\mathcal{G}_{\mathcal{H}})}\left\|\boldsymbol{E}^t\right\|^2_{(\boldsymbol{I}_{\mathcal{H}}-\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}})}+\eta^2\|\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}}\boldsymbol{E}^t\|^2\end{aligned}$$

As $\eta \leq \frac{1}{\mu_{\max}(\mathcal{G}_{\mathcal{H}})} \leq \frac{1}{\mu_2(\mathcal{G}_{\mathcal{H}})}$, we eventually get

$$\begin{aligned}\|\boldsymbol{X}_{\mathcal{H}}^{t+1}-\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}}\boldsymbol{X}_{\mathcal{H}}^{t}\|^2 &\leq \|\boldsymbol{X}_{\mathcal{H}}^{t}\|^2_{(\boldsymbol{I}_{\mathcal{H}}-\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}})}-\eta\left\|\boldsymbol{X}_{\mathcal{H}}^{t}\right\|^2_{\boldsymbol{W}_{\mathcal{H}}}+\eta\frac{1}{\mu_2(\mathcal{G}_{\mathcal{H}})}\left\|\boldsymbol{E}^t\right\|^2\\ &\leq \|\boldsymbol{X}_{\mathcal{H}}^{t}\|^2_{(\boldsymbol{I}_{\mathcal{H}}-\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}})}-\eta\left(1-\frac{2\rho b}{\mu_2(\mathcal{G}_{\mathcal{H}})}\right)\left\|\boldsymbol{X}_{\mathcal{H}}^{t}\right\|^2_{\boldsymbol{W}_{\mathcal{H}}}\\ &\leq \left(1-\eta\mu_2(\mathcal{G}_{\mathcal{H}})\left(1-\frac{2\rho b}{\mu_2(\mathcal{G}_{\mathcal{H}})}\right)\right)\|\boldsymbol{X}_{\mathcal{H}}^{t}\|^2_{(\boldsymbol{I}_{\mathcal{H}}-\boldsymbol{P}_{\boldsymbol{1}_{\mathcal{H}}})}.\end{aligned}$$

Which concludes the proof. □

4.C.2 Proof of Corollary 4.3.4

A direct consequence of the above results is Corollary 4.3.4, as we show below.

Proof. Using the $(\alpha,\,\lambda)$ reduction notations, we have:

$$\begin{cases}\alpha = 1-\gamma\,(1-\delta))\\ \lambda = \gamma\delta.\end{cases}$$

We denote here the drift increment $d_{t+1} = \|\boldsymbol{P}_{\mathbf{1}_\mathcal{H}}\boldsymbol{X}_\mathcal{H}^{t+1} - \boldsymbol{P}_{\mathbf{1}_\mathcal{H}}\boldsymbol{X}_\mathcal{H}^{t}\|$ and the variance at time t as $\sigma_t^2 = \|\boldsymbol{X}_\mathcal{H}^{t+1}\|^2_{(\boldsymbol{I}_\mathcal{H}-\boldsymbol{P}_{\mathbf{1}_\mathcal{H}})}$.

Corollary 4.C.4 ensures that

$$\sigma_{t+1}^2 + d_t^2 \leq \alpha\sigma_t^2.$$

Hence,we have $\sigma_{t+1}^2 + d_{t+1}^2 \leq \alpha\sigma_t^2$, and so $\sigma_{t+1}^2 \leq \alpha\sigma_t^2$, which implies that $\sigma_t \leq \alpha^{t/2}\sigma_0$. This proves the first part of the result. Using this, we write that Theorem 4.C.3 ensures that

$$d_{t+1} \leq \sqrt{\lambda}\sigma_t \leq \sqrt{\lambda}\beta^t\sigma 0,$$

leading to:

$$\sum_{t=1}^{T} d_t \leq \sqrt{\lambda}\sum_{t=0}^{T-1}\sigma_t \leq \sqrt{\lambda}\sum_{t=0}^{T-1}\alpha^{t/2}\sigma_0 \leq \frac{\sqrt{\lambda}(1-\alpha^{T/2})}{1-\alpha^{1/2}}\sigma 0,$$

which proves the second part. The last inequality is obtained by writing.

$$\|\boldsymbol{P}_{\mathbf{1}_\mathcal{H}}\boldsymbol{X}_\mathcal{H}^{T} - \boldsymbol{P}_{\mathbf{1}_\mathcal{H}}\boldsymbol{X}_\mathcal{H}^{0}\| \leq \sum_{t=1}^{T} d_t \leq \frac{\sqrt{\lambda}}{1-\sqrt{\alpha}}\sigma_0.$$

Then, we use that $0 \leq \frac{1}{1-\sqrt{1-x}} \leq \frac{2}{x}$ for $x \geq 0$, with $x = \gamma(1-\delta)$.

□

4.D Proof of Theorem 4.3.5 - Upper Bound on the Breakdown Point

We recall Theorem 4.3.5:

Theorem 4.D.1. *Let $\mu_{\min} \geq 0$, $b \geq 0$ be such that $\mu_{\min} \leq 2b$. Then for any $h \geq 0$ and any algorithm* Alg*, there exists a graph $\mathcal{G} \in \Gamma_{\mu_{\min},b}$ in which all honest nodes have a weight of honest neighbors $h(i)$ larger than h, and such that for any $r < 1$,* Alg *is not r–robust on $\mathcal{G}$.*

We recall that, since the Byzantine nodes are unknown, given a communication network and an algorithm Alg, the honest nodes follow Alg independently of the position of Byzantine nodes in the network. Thus, it is only when the position of the Byzantine nodes satisfies some hypothesis that Alg can be r-robust.

Here we consider the hypothesis $H = \{\mu_2(\mathcal{G}_\mathcal{H}) = 2b\} \cap \{\max_{i\in\mathcal{H}} b(i) \leq b\}$. We show that for any algorithm and any $h \geq 0$, there exists a communication network $\mathcal{N}$ on which any algorithm Alg is not r-robust for (at least) one configuration of the Byzantines that verifies the hypothesis H.

The structure of the proof is thus the following:

1. We introduce a family of unweighted communication networks $\{\mathcal{N}_{m,k}\}_{m\in\mathbb{N},k\in[m]}$, such that $\mathcal{N}_{m,k}$ admits three symmetric configurations of Byzantines nodes with $2m$ honest nodes and m Byzantines nodes, and each honest node is neighbor to k Byzantine

nodes and $m + k$ honest nodes. Then we show that an algorithm can not be r-robust in these three configurations with $r < 1$.

2. Now, for $c \geq 0$, let's denote $\mathcal{G}_{m,k,c}$ the weighted graph with a uniform weight c on the edges and such that $\mathcal{G}_{m,k,c}$ is a weighted version of one of the three above-mentioned configurations of $\mathcal{N}_{m,k}$. We show that:

   a) The weight of Byzantines in the neighborhood of any honest node is equal to $b = ck$.

   b) The algebraic connectivity of the honest subgraph $[\mathcal{G}_{m,k,c}]_{\mathcal{H}}$ verifies:

   $$\mu_2([\mathcal{G}_{m,k,c}]_{\mathcal{H}}) = 2ck = 2b.$$

   c) For any $h \geq b$, there exists a graph within $\{\mathcal{G}_{m,k,c}\}_{m\in\mathbb{N},k\in[m],c\in\mathbb{R}_+}$ such that all honest nodes have a weight associated to honest neighbors $h(i)$ larger than h.

This concludes the proof.

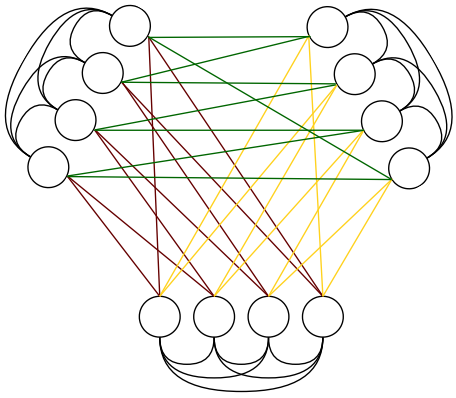


Figure 4.8: Topology of $\mathcal{N}_{m=4,k=2}$.

4.D.1 Definition of the Family of Graphs

Let $m \in \mathbb{N}^*$ and $k \in [m]$.

We define the communication network $\mathcal{N}_{m,k}$ as composed of three cliques of m nodes: C_1, C_2, and C_3. Each node in C_i is additionally connected to exactly k nodes in $C_{i+1 \mod 3}$ and to k nodes in $C_{i-1 \mod 3}$. Moreover, those connections are assumed to be *in circular order*, i.e., for any $j \in [m]$, node j in C_i is connected to nodes $j, \ldots, j + k \mod m$ in $C_{i+1 \mod 3}$.

We assume that one of the cliques is composed of Byzantine nodes, each honest node having k Byzantine neighbors, and there are m Byzantine nodes among the $3m$ nodes.

4.D.2 No Algorithm can be α-Robust on $\mathcal{N}_{m,k}$.

We now show that no algorithm can be robust on the communication network $\mathcal{N}_{m,k}$.

Lemma 4.D.2. *Let one unknown clique within C_1, C_2, and C_3 be composed of Byzantine nodes, then no communication algorithm is α-robust on $\mathcal{N}_{m,k}$.*

Assume an r–robust algorithm exists on the communication network $\mathcal{G}_{H,k}$. Informally:

1. We first show that if all nodes within one clique hold a unique parameter $\boldsymbol{x}$, and receive this parameter from nodes of either of the two other cliques, then they cannot change their parameter.

2. We then consider a setting where the two honest cliques hold different parameters, and we conclude that Byzantine nodes can force all honest nodes to keep their initial parameter at all times. This shows that in the considered setting, $r < 1$ is impossible.

Proof.
Part I. Let Alg be an algorithm r–robust on $\mathcal{N}_{m,k}$ with $r < 1$. We denote $\boldsymbol{x}_i^+$ the output from node i after running Alg.

We know that one of the cliques say C_1, is composed of honest nodes. Let the honest nodes within C_1 hold the parameter $\boldsymbol{x}$. Nodes in another clique, say C_2, declare the parameter $\boldsymbol{x}$ as well, while nodes in C_3 declare another parameter, say $\boldsymbol{y} \neq \boldsymbol{x}$. We show that all nodes $i \in C_1$ must output the parameter $\hat{\boldsymbol{x}}_i = \boldsymbol{x}$.

From the point of view of nodes in C_1 considering that the Byzantine clique is unknown, it is impossible to distinguish between these situations:

1. Situation I: C_2 is honest, and C_3 is Byzantine,

2. Situation II: C_2 is Byzantine, and C_3 is honest.

Thus the outputs of nodes in C_1 after running Alg, denoted $(\boldsymbol{x}_i^+)_{i \in C_1}$, are the same in both situations.

In Situation I, nodes of C_2 are honest, and nodes in C_1 and C_2 have the same initial parameter; hence, the initial quadratic error is 0. The r criterion writes

$$\sum_{i \in \mathcal{H}} \|\boldsymbol{x}_i^+ - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2 \leq r \sum_{i \in \mathcal{H}} \|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2 = 0.$$

It follows that for any node i in C_1, $\boldsymbol{x}_i^+ = \boldsymbol{x}$, *i.e.*, nodes in C_1 do not change their parameters (in both Situation I and Situation II).

Part II. Consider the setting where C_1 and C_2 are honest, while C_3 is Byzantine, and that nodes C_1 hold the parameter $\boldsymbol{x}$, while nodes in C_2 hold the parameter $\boldsymbol{y} \neq \boldsymbol{x}$.

As Byzantine nodes can declare different values to their different neighbors, nodes in C_3 can declare to nodes in C_1 that they hold the value $\boldsymbol{x}$, and to nodes in C_2 that they hold the value $\boldsymbol{y}$. Following Part I, nodes in C_1 and in C_2 cannot update their parameter, (i.e. $\forall i \in \mathcal{H}$, $\boldsymbol{x}_i^+ = \boldsymbol{x}_i$). Applying the r–robustness property brings:

$$\sum_{i \in \mathcal{H}} \|\boldsymbol{x}_i^+ - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2 \leq r \sum_{i \in \mathcal{H}} \|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2 = r \sum_{i \in \mathcal{H}} \|\boldsymbol{x}_i^+ - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2,$$

which implies that $r \geq 1$ since $\sum_{i\in\mathcal{H}} \|\boldsymbol{x}_i^+ - \overline{\boldsymbol{x}}_{\mathcal{H}}\|^2 > 0$. It follows that the Algorithm Alg is not r–robust on $\mathcal{N}_{m,k}$ for an $r < 1$.

□

4.D.3 Spectral Properties of the Graph

We define as $\mathcal{G}_{m,k,c}$ the graph associated with the network $\mathcal{N}_{m,k}$ where all edges have weight c and the nodes in one of the three cliques are Byzantine.

We show here that $\mu_2([\mathcal{G}_{m,k,c}]_{\mathcal{H}}) = 2kc$. This concludes the proof as we have:

1. The weight of Byzantines in the neighborhood of any honest node is equal to $b = ck$;
2. Which is linked to the algebraic connectivity of the honest subgraph $\mu_2([\mathcal{G}_{m,k,c}]_{\mathcal{H}}) = c2k$;
3. While the weight of honest nodes in the neighborhood of honest nodes is equal to $h = c(m + k)$, thus growths to infinity with m;

Analysis We denote $\boldsymbol{L}_{\mathcal{H}}$ the Laplacian of $[\mathcal{G}_{m,k,c=1}]_{\mathcal{H}}$, the honest subgraph of the network $\mathcal{N}_{m,k}$ in which we provided unitary weights to edges. We show that $\mu_2([\mathcal{G}_{m,k,c=1}]_{\mathcal{H}}) = 2k$, which brings that $\mu_2([\mathcal{G}_{m,k,c}]_{\mathcal{H}}) = 2kc$.

Lemma 4.D.3. *The second smallest eigenvalue of the (unitary weighted) Laplacian matrix $\boldsymbol{L}_{\mathcal{H}}$ of $[\mathcal{G}_{m,k,1}]_{\mathcal{H}}$ is equal to $\mu_2(\boldsymbol{L}_{\mathcal{H}}) = 2k$.*

Proof.

Recall that $m \triangleq m$. Let $\boldsymbol{M} \in \mathbb{R}^{m\times m}$ be a circulant matrix defined as $\boldsymbol{M} = \sum_{q=0}^{k-1} \boldsymbol{J}^q$, where $\boldsymbol{J}$ denotes the permutation

$$\boldsymbol{J} := \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & & & & 1 \\ 1 & 0 & 0 & \dots & 0 \end{pmatrix}.$$

The Laplacian matrix of the honest subgraph of $\mathcal{G}_{H,k}$ can be written as:

$$\boldsymbol{L}_{\mathcal{H}} = \begin{pmatrix} m\boldsymbol{I}_m - \mathbf{1}_m\mathbf{1}_m^T & 0 \\ 0 & m\boldsymbol{I}_m - \mathbf{1}_m\mathbf{1}_m^T \end{pmatrix} + \begin{pmatrix} k\boldsymbol{I}_m & -\boldsymbol{M} \\ -\boldsymbol{M}^T & k\boldsymbol{I}_m \end{pmatrix}$$

Hence

$$\boldsymbol{L}_{\mathcal{H}} = (k+m)\boldsymbol{I}_{2m} - \begin{pmatrix} \mathbf{1}_m\mathbf{1}_m^T & 0 \\ 0 & \mathbf{1}_m\mathbf{1}_m^T \end{pmatrix} - \begin{pmatrix} 0 & \boldsymbol{M} \\ \boldsymbol{M}^T & 0 \end{pmatrix}. \tag{4.10}$$

This matrix decomposition allows us to obtain the eigenvalues of the matrix $W_{\mathcal{G}}$.

Lemma 4.D.4. *The eigenvalues of* $\boldsymbol{L}_{\mathcal{H}}$ *are* $\{0, 2k\} \cup \{k + m \pm |\sum_{q=0}^{k-1} \omega^{pq}|; p \in \{1, \dots, m-1\}\}$ *where* $\omega := \exp(\frac{2i\pi}{m})$.

To prove the Lemma 4.D.4, we first need to following result.

Lemma 4.D.5. *If* $\boldsymbol{A}$ *is a symmetric matrix in* $\mathbb{R}^{2m \times 2m}$*, which can be decomposed as* $\boldsymbol{A} = \begin{pmatrix} 0 & \boldsymbol{M} \\ \boldsymbol{M}^T & 0 \end{pmatrix}$*, where* $\boldsymbol{M} \in \mathbb{R}^{m \times m}$ *is a matrix with complex eigenvalues* $\mu_0, \dots, \mu_{m-1}$.
Then the eigenvalues of $\boldsymbol{A}$ *are* $\{\pm|\mu_p|; p = 0, \dots, m-1\}$.

Proof of Lemma 4.D.5. Let $\boldsymbol{M} = \boldsymbol{U}^* \boldsymbol{D} \boldsymbol{U}$ be the diagonalization of $\boldsymbol{M}$ where $\boldsymbol{D} = \text{Diag}(\mu_0, \dots, \mu_{m-1})$, and $\boldsymbol{U}$ is a unitary matrix, i.e. $\boldsymbol{U}\boldsymbol{U}^* = \boldsymbol{I}$ where we denote as $\boldsymbol{U}^* = \overline{\boldsymbol{U}}^T$ the conjugate transpose of $\boldsymbol{U}$, where the (simple) conjugate matrix is denoted $\overline{\boldsymbol{U}}$.

Lemma 4.D.5 follows from

$$\begin{pmatrix} 0 & \boldsymbol{M} \\ \boldsymbol{M}^T & 0 \end{pmatrix} = \begin{pmatrix} 0 & \boldsymbol{U}^* \boldsymbol{D} \boldsymbol{U} \\ \boldsymbol{U}^T \boldsymbol{D} \overline{\boldsymbol{U}} & 0 \end{pmatrix} \underset{\overline{\boldsymbol{A}} = \boldsymbol{A}}{=} \begin{pmatrix} 0 & \boldsymbol{U}^* \boldsymbol{D} \boldsymbol{U} \\ \boldsymbol{U}^* \overline{\boldsymbol{D}} \boldsymbol{U} & 0 \end{pmatrix}.$$

Hence

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{U}^* & 0 \\ 0 & \boldsymbol{U}^* \end{pmatrix} \begin{pmatrix} 0 & \boldsymbol{D} \\ \overline{\boldsymbol{D}} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{U} & 0 \\ 0 & \boldsymbol{U} \end{pmatrix}.$$

A simple calculus (using that $\boldsymbol{D}$ is diagonal) yields that all eigenvalues of $\begin{pmatrix} 0 & \boldsymbol{D} \\ \overline{\boldsymbol{D}} & 0 \end{pmatrix}$ are $\{\pm|\boldsymbol{D}_p|; p = 0, \dots, m-1\}$. □

Proof of Lemma 4.D.4.

We start from the decomposition of Equation (4.10) :

$$\boldsymbol{L}_{\mathcal{H}} = (k+m)\boldsymbol{I}_{2m} - \begin{pmatrix} \mathbf{1}_m\mathbf{1}_m^T & 0 \\ 0 & \mathbf{1}_m\mathbf{1}_m^T \end{pmatrix} - \begin{pmatrix} 0 & \boldsymbol{M} \\ \boldsymbol{M}^T & 0 \end{pmatrix}.$$

We first notice that the subspace spanned by $(\mathbf{1}_m^T, +\mathbf{1}_m^T)^T$ and $(\mathbf{1}_m^T, -\mathbf{1}_m^T)^T$ is an eigenspace of $\begin{pmatrix} \mathbf{1}_m\mathbf{1}_m^T & 0 \\ 0 & \mathbf{1}_m\mathbf{1}_m^T \end{pmatrix}$ associated the eigenvalue m, and the orthogonal subspace is associated with 0. Furthermore these are eigenvectors of $\begin{pmatrix} 0 & \boldsymbol{M} \\ \boldsymbol{M}^T & 0 \end{pmatrix}$ associated with k and $-k$. It follows that they are eigenvectors of $\boldsymbol{L}_{\mathcal{H}}$ with eigenvalues 0 and $2k$. We notice as well that the three matrices of Equation (4.10) can be diagonalized in the same orthogonal basis.

The matrix $\boldsymbol{M}$ is a circulant matrix, so it can be diagonalized in $\mathbb{C}$. The eigenvalues are $\{\mu_p = \sum_{q=0}^{k-1}\omega^{pq}; p \in \{0,\dots,m-1\}\}$, where $\omega := \exp(\frac{2i\pi}{m})$. The eigenvector associated with μ_p is $x_p = (1, \omega^p, \dots, \omega^{(m-1)p})^T$. As such, with $U = (x_0, \dots, x_{m-1})$ and $\boldsymbol{D} = \mathrm{Diag}(\mu_0, \dots, \mu_{m-1})$, $\boldsymbol{M}$ writes:

$$\boldsymbol{M} = \boldsymbol{U}^*\boldsymbol{D}\boldsymbol{U}.$$

Considering Lemma 4.D.5, the eigenvalue of $\begin{pmatrix} 0 & \boldsymbol{M} \\ \boldsymbol{M}^T & 0 \end{pmatrix}$ are $\{\pm|\mu_p|; p = 0,\dots,m-1\}$, considering that $p = 0$ corresponds to the eigenvalues $+k$ and $-k$, hence the eigenvectors $(\mathbf{1}_m^T, \mathbf{1}_m^T)^T$ and $(\mathbf{1}_m^T, -\mathbf{1}_m^T)^T$, we deduce that the eigenvalues of $\boldsymbol{L}_{\mathcal{H}}$ are $\{\pm|\mu_p|; p = 0 \dots m-1\}$. □

End of the proof of Lemma 4.D.3.

To prove Lemma 4.D.3, considering the decomposition of Equation (4.10), we only have to show that $m-k$ is always the second largest eigenvalue of the matrix

$$\boldsymbol{B} := \begin{pmatrix} \mathbf{1}_m\mathbf{1}_m^T & 0 \\ 0 & \mathbf{1}_m\mathbf{1}_m^T \end{pmatrix} + \begin{pmatrix} 0 & \boldsymbol{M} \\ \boldsymbol{M}^T & 0 \end{pmatrix}.$$

First, considering Lemma 4.D.4, the eigenvalues of B are $\{m+k, m-k\} \cup \{\pm|\mu_p|; p \in \{1,\dots,m-1\}\}$ with $\mu_p = \sum_{q=0}^{k-1}\omega^{pq}$. As such showing that $|\mu_p| \leq m-k$ if $p \in \{1,\dots,m-1\}$ yields the result.

As $\omega^{mp} = \omega^{0p} = 1$, we have that $\sum_{q=0}^{m-1}\omega^{pq}(1-\omega^p) = 0$. Hence, for $p \in \{1,\dots,m-1\}$, as $\omega^p \neq 1$,

$$\sum_{q=0}^{m-1}\omega^{pq} = 0 \implies \mu_p = \sum_{q=0}^{k-1}\omega^{pq} = -\sum_{q=k}^{m-1}\omega^{pq}.$$

It follows from $|\omega| = 1$ that for $p \in \{1,\dots,m-1\}$, $|\mu_p| \leq m-k$. □

4.E (b, ρ)-Robust Summation Rules

We recall the definition of summation rules.

Definition 4.E.1 ((b, ρ)–robust summation)**.** Let $b, \rho \geq 0$. An aggregation rule $F : (\mathbb{R}_+ \times \mathbb{R}^d)^n \to \mathbb{R}^d$ is a (b, ρ)–*robust summation*, when, for any vectors $(\boldsymbol{z}_i)_{i\in[n]} \in (\mathbb{R}^d)^n$, any weights $(\omega_i)_{i\in[n]} \in \mathbb{R}_+^n$ and any set $S \subset [n]$ such that $\sum_{i\in\overline{S}} \omega_i \leq b$, there is

$$\left\| F\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]}\big) - \sum_{i\in S} \omega_i \boldsymbol{z}_i \right\|^2 \leq \rho b \sum_{i\in S} \omega_i \|\boldsymbol{z}_i\|^2.$$

where $\overline{S} := [n]\backslash S$.

4.E.1 Clipping

We first prove the robustness of the clipping-based summation.

Proposition 4.E.2. *Let $b \geq 0$, then*

1. *(Practical)* CS_+ *is (b, ρ)–robust with $\rho = 2$.*
2. *(Oracle)* $\mathrm{CS}_+^{or.}$ *is (b, ρ)–robust with $\rho = 1$.*
3. *(Oracle)* $\mathrm{CS}_{He}^{or.}$ *is (b, ρ)–robust with $\rho = 4$.*

Proof. Let $(\omega_i, \boldsymbol{z}_i)_{i\in[n]} \in (\mathbb{R}_+ \times \mathbb{R}^d)^n$, and $S \subset [n]$ such that $\sum_{i\in\overline{S}} \omega_i \leq b$,
We consider for $\tau \geq 0$

$$F\big((\omega_i, \boldsymbol{z}_i)_{i=1,\dots,n}\big) := \sum_{i=1}^n \omega_i \mathrm{clip}(\boldsymbol{z}_i; \tau).$$

Then, by applying the triangle inequality we have

$$\begin{aligned}
\left\| F\big((\omega_i, \boldsymbol{z}_i)_{i=1,\dots,n}\big) - \sum_{i\in S} \omega_i \boldsymbol{z}_i \right\|^2 &= \left\| \sum_{i\in S} \omega_i \left(\mathrm{clip}(\boldsymbol{z}_i; \tau) - \boldsymbol{z}_i\right) + \sum_{i\in\overline{S}} \omega_i \mathrm{clip}(\boldsymbol{z}_i; \tau) \right\|^2 \\
&\leq \left(\sum_{i\in S} \omega_i \|\boldsymbol{z}_i - \mathrm{clip}(\boldsymbol{z}_i; \tau)\| + \sum_{i\in\overline{S}} \omega_i \|\mathrm{clip}(\boldsymbol{z}_i; \tau)\| \right)^2 \\
&\leq \left(\sum_{i\in S} \omega_i (\|\boldsymbol{z}_i\| - \tau)_+ + b\tau \right)^2 . \qquad (4.11)
\end{aligned}$$

Where we used $\sum_{i\in\overline{S}} \omega_i \leq b$.

1. Case of our clipping threshold. We choose τ as

$$\tau = \max \left\{ \tau \geq 0 : \sum_{i=1}^n \omega_i \mathbf{1}_{\|\boldsymbol{z}_i\|\geq\tau} \geq 2b \right\} \triangleq \tau_{\mathrm{ours}}\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]}\big).$$

This corresponds to lowering the clipping threshold until the sum of the weights of clipped vectors is essentially equal to $2b$. This ensures that the total weight of the honest vectors that are clipped falls between b and $2b$. If there are ties at the clipping threshold, honest vectors can be arbitrarily denoted as *clipped* or *non-clipped*. Indeed, there is no clipping error incurred since the clipping threshold is the same as the actual value of the difference. Therefore, we do not have to accumulate error for the weight over $2b$, so the following equation always holds:

$$2b \geq \sum_{i \in S} \omega_i \mathbf{1}_{i \text{ clipped}} \geq b.$$

Which allows us to write:

$$\begin{aligned}\sum_{i \in S} \omega_i \left(\|\boldsymbol{z}_i\| - \tau\right)_+ + b\tau &\leq \sum_{i \in S} \omega_i (\|\boldsymbol{z}_i\| - \tau) \mathbf{1}_{i \text{ clipped}} + b\tau \\ &\leq \sum_{i \in S} \omega_i \|\boldsymbol{z}_i\| \mathbf{1}_{i \text{ clipped}}.\end{aligned}$$

We conclude the proof using the Cauchy-Schwarz inequality:

$$\begin{aligned}\left\| F\big((\omega_i, \boldsymbol{z}_i)_{i=1,\dots,n}\big) - \sum_{i \in S} \omega_i \boldsymbol{z}_i \right\|^2 &\leq \left(\sum_{i \in S} \sqrt{\omega_i} \|\boldsymbol{z}_i\| \cdot \sqrt{\omega_i} \mathbf{1}_{i \text{ clipped}} \right)^2 \\ &\leq \left(\sum_{i \in S} \omega_i \mathbf{1}_{i \text{ clipped}} \right) \left(\sum_{i \in S} \omega_i \|\boldsymbol{z}_i\|^2 \right) \\ &\leq 2b \sum_{i \in S} \omega_i \|\boldsymbol{z}_i\|^2.\end{aligned}$$

Where we used $\sum_{j \in n_{\mathcal{H}}(i)} \omega_{ij} \mathbf{1}_{j \text{ clipped}} \leq 2b$. Note that, if somehow it is possible to choose the clipping threshold such that the weight of clipped vectors within S is equal exactly to b then this factor 2 disappears. Which correspond to our oracle clipping threshold $\tau^{\text{or.}}_{\text{ours}}$. The same result can be achieved when it is possible to identify a subset of $\{1, \dots, n\}$ of weight $2b$ which includes the set $\overline{S}$, and if the weight with $\overline{S}$ sums exactly to b. This is for instance the case of the communication graph used in Section 4.D: nodes in (for instance) C_1 know that Byzantines nodes are among the nodes within C_2 and C_3, thus selecting all their neighbors that belong to these two other cliques leads to a subset of neighbors of weight $2b$ with exactly a weight b corresponding to Byzantine neighbors.

2. Clipping threshold of He et al. (2023) We plug in Equation (4.11) the following upper bound:

$$(\|\boldsymbol{z}_i\| - \tau)_+ = \tau \left(\frac{\|\boldsymbol{z}_i\|}{\tau} - 1 \right)_+ \leq \tau \left(\frac{\|\boldsymbol{z}_i\|^2}{\tau^2} - 1 \right)_+ \leq \frac{\|\boldsymbol{z}_i\|^2}{\tau}.$$

This yields

$$\left\| F\big((\omega_i, \boldsymbol{z}_i)_{i=1,\dots,n}\big) - \sum_{i \in S} \omega_i \boldsymbol{z}_i \right\|^2 \leq \left(\sum_{i \in S} \omega_i \frac{\|\boldsymbol{z}_i\|^2}{\tau} + b\tau \right)^2.$$

Then taking as clipping threshold the minimizer of the RHS, $\tau^* = \sqrt{\frac{1}{b}\sum_{i\in S}\omega_i\|\boldsymbol{z}_i\|^2} \triangleq \tau^{\text{or.}}_{He}\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]}\big)$ leads to:

$$\left\|F\big((\omega_i, \boldsymbol{z}_i)_{i=1,\dots,n}\big) - \sum_{i\in S}\omega_i\boldsymbol{z}_i\right\|^2 \leq 4b\sum_{i\in S}\omega_i\|\boldsymbol{z}_i\|^2.$$

However, the clipping threshold here requires an exact knowledge of the set S. Furthermore, it is unclear to what extent making an approximate estimate of this clipping threshold allows us to derive robustness guarantees. □

Remark 4.E.3. A key point here is that this oracle clipping threshold corresponds to the **unique minimizer** within each squared term of the sum. Hence, considering for instance the adaptive practical clipping rule of (He et al., 2023) leads to a **larger upper bound on the error**.

4.E.2 Geometric Trimming a.k.a. NNA

Let $(\omega_i, \boldsymbol{z}_i)_{i\in[n]} \in (\mathbb{R}_+ \times \mathbb{R}^d)^n$, and $S \subset [n]$ such that $\sum_{i\in\overline{S}}\omega_i \leq b$,

We recall the definition of GTS: Assume w.l.o.g. that $(\|\boldsymbol{z}_i\|)_{i\in[n]}$ are sorted, i.e. $\|\boldsymbol{z}_1\| \leq \ldots \leq \|\boldsymbol{z}_n\|$, and denote $k^*(b) := \max\{k \in [n]; \sum_{i\geq k}\omega_i \geq b\}$ the index of the largest vector which has at least a weight b of vector largest than him. (GTS) computes $\tilde{\omega}_{k^*(b)} := \sum_{i\geq k^*(b)}\omega_i - b$, and outputs

$$\mathrm{GTS}\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]}\big) = \tilde{\omega}_{k^*(b)}\boldsymbol{z}_{k^*} + \sum_{i<k^*(b)}\omega_i\boldsymbol{z}_i.$$

Lemma 4.E.4. *Geometric trimming is (b, ρ) -robust with $\rho = 4$.*

Proof. Let $(\omega_i, \boldsymbol{z}_i)_{i\in[n]} \in (\mathbb{R}_+ \times \mathbb{R}^d)^n$, and $S \subset [n]$ such that $\sum_{i\in\overline{S}}\omega_i \leq b$, Without loss of generality we assume that $\tilde{\omega}_{k^*(b)} = 0$[4].

Thus the aggregation rules write

$$F\big((\omega_i, \boldsymbol{z}_i)_{i=1,\dots,n}\big) := \sum_{i<k^*(b)}\omega_i\boldsymbol{z}_i. = \sum_{i=1}^{n}\omega_i\boldsymbol{z}_i\mathbf{1}_{i \text{ not removed}}$$

Then, by applying the triangle inequality we have

$$\begin{aligned}\left\|F\big((\omega_i, \boldsymbol{z}_i)_{i=1,\dots,n}\big) - \sum_{i\in S}\omega_i\boldsymbol{z}_i\right\|^2 &= \left\|\sum_{i\in S}\omega_i\boldsymbol{z}_i\mathbf{1}_{i \text{ removed}} + \sum_{i\in\overline{S}}\omega_i\boldsymbol{z}_i\mathbf{1}_{i \text{ not removed}}\right\|^2 \\ &\leq \left(\sum_{i\in S}\omega_i\|\boldsymbol{z}_i\|\mathbf{1}_{i \text{ removed}} + \sum_{i\in\overline{S}}\omega_i\|\boldsymbol{z}_i\|\mathbf{1}_{i \text{ not removed}}\right)^2.\end{aligned}$$

[4]Which can be ensured by adding artificially the entry $(\tilde{\omega}_{k^*(b)}, \boldsymbol{z}_{k^*(b)})$.

As $\sum_{i=1}^n \omega_i \mathbf{1}_{i \text{ removed}} = b$, it holds that

$$\begin{aligned} &\sum_{i\in\overline{S}} \omega_i \leq b = \sum_{i=1}^n \omega_i \mathbf{1}_{i \text{ removed}} = \sum_{i\in S} \omega_i \mathbf{1}_{i \text{ removed}} + \sum_{i\in\overline{S}} \omega_i \mathbf{1}_{i \text{ removed}} \\ \implies &\sum_{i\in\overline{S}} \omega_i \mathbf{1}_{i \text{ not removed}} \leq \sum_{i\in S} \omega_i \mathbf{1}_{i \text{ removed}}. \end{aligned}$$

Furthermore, if i is removed and j is not, then $\|\boldsymbol{z}_i\| \geq \|\boldsymbol{z}_j\|$. It follows that

$$\sum_{i\in S} \omega_i \|\boldsymbol{z}_i\| \mathbf{1}_{i \text{ removed}} \geq \sum_{i\in\overline{S}} \omega_i \|\boldsymbol{z}_i\| \mathbf{1}_{i \text{ not removed}}.$$

Consequently:

$$\begin{aligned} \left\| F\big((\omega_i, \boldsymbol{z}_i)_{i=1,\dots,n}\big) - \sum_{i\in S} \omega_i \boldsymbol{z}_i \right\|^2 &\leq \left(2 \sum_{i\in S} \omega_i \|\boldsymbol{z}_i\| \mathbf{1}_{i \text{ removed}} \right)^2 \\ &= 4 \left(\sum_{i\in S} \sqrt{\omega_i} \|\boldsymbol{z}_i\| \sqrt{\omega_i} \mathbf{1}_{i \text{ removed}} \right)^2 \\ &\leq 4 \sum_{i\in S} \omega_i \mathbf{1}_{i \text{ removed}} \sum_{i\in S} \omega_i \|\boldsymbol{z}_i\|^2 \end{aligned}$$

using Cauchy-Schwarz.

Thus:

$$\left\| F\big((\omega_i, \boldsymbol{z}_i)_{i=1,\dots,n}\big) - \sum_{i\in S} \omega_i \boldsymbol{z}_i \right\|^2 \leq 4b \sum_{i\in S} \omega_i \|\boldsymbol{z}_i\|^2.$$

□

4.F Proofs for D-SGD

Proof of Corollary 4.4.8.

This proof hinges on the fact that the proof of Farhadkhani et al. (2023, Theorem 1) does not actually require that communication is performed using NNA, but simply that the aggregation procedure respects (α, λ)-reduction, which they prove in their Lemma 2. Then, all subsequent results invoke this Lemma instead of the specific aggregation procedure. CS_+-RG also satisfies (α, λ)-reduction, as we prove in Theorem 4.3.3. We can then use their bounds on the errors out of the box.

Then, as T grows, and ignoring constant factors, only the first and last terms in their Theorem 3 remain, leading to, for $i \in \mathcal{H}$:

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E}\left[\left\| \nabla f_{\mathcal{H}}(\boldsymbol{x}_i^t) \right\|^2 \right] = \mathcal{O}\left(L\sigma \sqrt{\frac{f(\overline{\boldsymbol{x}}_{\mathcal{H}}^0) - f(\boldsymbol{x}^*)}{T} (|\mathcal{H}|^{-1} + C)} + C\zeta^2 \right),$$

where $C = c_1 + \lambda + \lambda c_1$, with $c_1 = \alpha(1+\alpha)/(1-\alpha)^2$. Note that we give $\mathcal{O}()$ versions of the Theorems for simplicity, but Farhadkhani et al. (2023, Theorem 1) allows to derive precise upper bounds for any $T \geq 1$.

One-step derivation. The one-step result is obtained by taking the values of $\alpha = 1 - \gamma(1-\delta)$ and $\lambda = \gamma\delta$. On the one hand

$$c_1 = \frac{(1-\gamma(1-\delta))(2-\gamma(1-\delta))}{\gamma^2(1-\delta)^2} = O\left(\frac{1-\gamma(1-\delta)}{\gamma^2(1-\delta)^2}\right).$$

On the other hand, $\lambda = \gamma\delta \leq 1$ implies that $C = O(c_1)$. Both lead to

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla f_{\mathcal{H}}(\boldsymbol{x}_i^t)\right\|^2\right]\\ &= \mathcal{O}\left(L\sigma\sqrt{\frac{f(\overline{\boldsymbol{x}}_{\mathcal{H}}^0)-f(\boldsymbol{x}^*)}{T}\left(|\mathcal{H}|^{-1}+\frac{1-\gamma(1-\delta)}{\gamma^2(1-\delta)^2}\right)}+\frac{1-\gamma(1-\delta)}{\gamma^2(1-\delta)^2}\zeta^2\right).\end{aligned}$$

The result of Corollary 4.4.8 derives from the case $\gamma \ll 1$ (otherwise, the guarantees are essentially the same as in Farhadkhani et al. (2023)), and $\delta \geq \frac{1}{\mathcal{H}}$ (otherwise there are essentially no Byzantines).

Multi-step derivation. In the previous case, we see that C is dominated by the c_1 term since $c_1 >> \lambda$. In particular, the guarantees would increase if we were able to trade-off some α for some λ, which is possible by using multiple communications steps. This is what we do, and take enough steps that $c_1 = o(\lambda)$ (i.e., $\alpha \approx 0$), so that $C \approx \lambda$. Following Corollary 4.3.4, it can be achieved by performing $\tilde{O}(\gamma^{-1}(1-\delta)^{-1})$ steps of F-RG, where logarithmic factors are hidden in the $\tilde{O}$ notation. We then plug the multi-step λ value from Corollary 4.3.4 to obtain the result: as $\lambda = O\left(\frac{\delta}{\gamma(1-\delta)}\right)$, the multi-communication steps convergence bound writes

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla f_{\mathcal{H}}(\boldsymbol{x}_i^t)\right\|^2\right]\\ &= \mathcal{O}\left(L\sigma\sqrt{\frac{f(\overline{\boldsymbol{x}}_{\mathcal{H}}^0)-f(\boldsymbol{x}^*)}{T}\left(|\mathcal{H}|^{-1}+\frac{\delta}{\gamma(1-\delta)^2}\right)}+\frac{\delta}{\gamma(1-\delta)^2}\zeta^2\right).\end{aligned}$$

□

Chapter 5

Byzantine-Robust Gossip: Insights from a Dual Approach

Distributed learning has many computational benefits but is vulnerable to attacks from a subset of devices transmitting incorrect information. This paper investigates Byzantine-resilient algorithms in a decentralized setting, where devices communicate directly in a peer-to-peer manner within a communication network. We leverage the so-called *dual approach* for decentralized optimization and propose a Byzantine-robust algorithm. We provide convergence guarantees in the average consensus subcase, discuss the potential of the dual approach beyond this subcase, and re-interpret existing algorithms using the dual framework. Lastly, we experimentally show the soundness of our method.

Contents

5.1 Introduction 91
5.2 Deriving EdgeClippedGossip 92
5.2.1 Gossip - Average Consensus Case 92
5.2.2 The Dual Approach in Decentralized Optimization 93
5.3 Global Clipping 96
5.3.1 The Global Clipping Rule 96
5.3.2 Convergence Result 97
5.4 Beyond Global Coordination and Averaging 98
5.4.1 Dual Interpretation of Existing Algorithms 98
5.4.2 Dual Formalism for Asymmetric Clipping 99
5.5 Experiments 100
5.6 Conclusion 101

5.A Experimental design 103
5.B Proofs 103

5.1 Introduction

Distributed optimization algorithms allow us to harness the multiplication of devices to tackle the increasing computational complexity of machine learning models. Yet, the involvement of many parties in the learning process opens up the threat of misbehaving devices, that can purposely or inadvertently send arbitrarily harmful messages. Such device failures are commonly termed *Byzantine* (Lamport et al., 1982). This paper studies Byzantine robustness in the decentralized communication paradigm, which alleviates the risk of a single point of failure due to a central server by making devices directly communicate with each other in a peer-to-peer manner.

We consider devices (a.k.a nodes) that communicate synchronously within a communication network, abstracted as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each vertex is a node that communicates with its neighbors in the graph through the edges. We denote as $\mathcal{V}_h$ the Byzantine nodes, and $\mathcal{V}_h = \mathcal{V} \backslash \mathcal{V}_b$ the remaining nodes, called honest. Similarly, we split the edges in $\mathcal{E} = \mathcal{E}_h \cup \mathcal{E}_b$, where $\mathcal{E}_h$ are the edges linking honest nodes, while edges in $\mathcal{E}_b$ link honest to Byzantine nodes. The subgraph of honest nodes $\mathcal{G}_h := (\mathcal{V}_h, \mathcal{E}_h)$ is always assumed to be connected. For any $i \in \mathcal{V}_h$, we denote $f_i : \mathbb{R}^d \to \mathbb{R}$ the local objective function on node i, and we study the optimization problem:

$$\min_{x \in \mathbb{R}^d} \sum_{i \in \mathcal{V}_h} f_i(x). \tag{5.1}$$

We study decentralized algorithms to solve this problem, which rely on the *gossip* communication protocol (Boyd et al., 2006; Nedic and Ozdaglar, 2009; Duchi et al., 2011; Scaman et al., 2017), in which each node $i \in \mathcal{V}$ maintains a *local parameter* $x_i \in \mathbb{R}^d$, and updates this parameter by performing approximate average of this parameter with those of its neighbors in the graph.

Such gossip algorithms naturally arise when solving a well-chosen dual formulation of Equation (5.1) with standard first-order methods. This *dual approach* gives a principled framework to design and analyze efficient decentralized algorithms, leading for instance to acceleration (Scaman et al., 2017; Kovalev et al., 2020), variance reduction (Hendrikx et al., 2019) and has strong links with gradient tracking (Jakovetić, 2018). This paper investigates the benefits of this approach in the Byzantine robust setting.

Related works. Since its introduction by Blanchard et al. (2017), the setting of Byzantine-robust learning has been extensively studied. The large majority of works focus on the *federated* case, in which a central server organizes the training and communicates directly with each node (Yin et al., 2018; Chen et al., 2017; Alistarh et al., 2018; Li et al., 2019; El-Mhamdi et al., 2020; Karimireddy et al., 2023, 2021; Farhadkhani et al., 2022; Allouah et al., 2023a). Some work investigated the robust decentralized optimization setting, either with fully connected networks (El-Mhamdi et al., 2021; Farhadkhani et al., 2023) or with more generic networks (He et al., 2023; Wu et al., 2023; Gaucher et al., 2025). However, those works only skimmed over the challenges of decentralized optimization in the Byzantine setting, and many key optimization techniques such as gradient tracking or acceleration are still unexplored in the presence of adversaries. Developing the dual framework in the presence of adversaries would thus open the way to tackle these challenges.

Contributions. In this paper, we (i) leverage the dual approach to derive EdgeClipped-Gossip, a decentralized robust algorithm based on clipped dual gradient descent, (ii) derive clipping rules and their associated convergence guarantees in the average consensus subproblem, in which EdgeClippedGossip recovers existing algorithms. Finally, (iii) we highlight the dynamics of the developed method compared to previous approaches and discuss its generalization beyond the average consensus case.

Notations. We denote $\mathbf{1}_n = (1, \dots, 1)^T$; $\|\cdot\|_2$ or simply $\|\cdot\|$ (resp. $\langle\cdot,\cdot\rangle$) the Euclidean norm (resp. inner product) on $\mathbb{R}^d$; $\|\cdot\|_F$ the Frobenius norm of any matrix in $\mathbb{R}^{d\times n}$, and $\langle M, N\rangle_F = \mathrm{tr}(M^T N)$ the corresponding inner product. We denote $X_{i,:}$ the i-th row of a matrix X. For a matrix M in $\mathbb{R}^{n\times d}$, we denote $\overline{M} := n^{-1}\mathbf{1}_n\mathbf{1}_n^T M$, and for $p \in \{1, 2, \infty\}$, we denote $\|M\|_{p,2}$, the p-norm of the vector of 2-norms of the rows of M, i.e., $\|(\|M_{i,:}\|_2)_{i\leq n}\|_p$. Moreover, $C^\dagger$ is the Moore Penrose inverse of C and I is the identity matrix. We identify $\mathbb{R}^{\mathcal{E}}$ to $\mathbb{R}^{|\mathcal{E}|}$ and $\mathbb{R}^{\mathcal{V}}$ to $\mathbb{R}^n$. We denote $a \wedge b = \min(a, b)$.

5.2 Deriving EdgeClippedGossip

This section describes a dual approach to Byzantine robustness. We first introduce a dual gossip algorithm, then show how it can be made robust using EdgeClippedGossip.

Setting: We denote $n_h = |\mathcal{V}_h|$ and $n_b = |\mathcal{V}_b|$ the total number of honest and Byzantine nodes, and $N_h(i)$ and $N_b(i)$ the number of honest and Byzantine neighbors of node i. Each agent owns and iteratively updates a local parameter x_i^t. At time t, we denote $X^t := [x_1^t, \dots, x_n^t]^T \in \mathbb{R}^{\mathcal{V}\times d}$ the parameter matrix, which we split into honest and Byzantine as $X^t = \begin{pmatrix} X_h^t \\ X_b^t \end{pmatrix}$, where $X_h^t = [(x_i^t)_{i\in\mathcal{V}_h}]^T$ (resp. X_b^t) is the sub-matrix containing the models held by the honest workers (resp. Byzantine). Note that Byzantine nodes can declare different values to each of their neighbors. As such, only the number and positions of edges with Byzantine nodes matter, and we can w.l.o.g identify the number of edges linking Byzantine and honest nodes $|\mathcal{E}_b|$ to the number of Byzantine workers n_b.

5.2.1 Gossip - Average Consensus Case

Gossip communications (Boyd et al., 2006) consist in sharing information between neighbors through local averaging steps. The standard gossip protocol writes $X^{t+1} = WX^t$, where $W \in [0, 1]^{n\times n}$ is a *gossip matrix*.

Definition 5.2.1 (Gossip matrix)**.** A matrix $W \in \mathbb{R}_+^{n\times n}$ is a *gossip matrix* for a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ if: (i) W is symmetric and doubly-stochastic: $W\mathbf{1}_n = \mathbf{1}_n$, and $\mathbf{1}_n^T W = \mathbf{1}_n^T$. (ii) W is supported on the edges of the network: $W_{ij} \neq 0$ if and only if $i = j$ or $(i \sim j) \in \mathcal{E}$.

Such a matrix W has eigenvalues $1 = \mu_1(W) \geq |\mu_2(W)| \geq \ldots \geq |\mu_n(W)|$. The propagation of information in the graph is characterized by the *spectral gap* of W, which is defined as $\gamma_W := 1 - |\mu_2(W)|$. Note that, if the graph $\mathcal{G}$ is connected, then $\gamma_W > 0$.

Example 5.2.2 (Gossip from Graph Laplacian). Consider the Laplacian matrix of the graph $L = D - A$ where D is the diagonal matrix of degrees and A is the adjacency matrix (i.e., $A_{ij} = 1_{(i\sim j)\in\mathcal{E}}$). Then, $W_L = \mathrm{I} - \eta L$ is a gossip matrix for $\eta \leq 1/\mu_n(W)$, with spectral gap $\gamma_{W_L} = \eta\mu^+_{\min}(L)$, where $\mu^+_{\min}(L)$ is the second smallest eigenvalue of L.

The *average consensus* problem is an important special case of Equation (5.1).

Definition 5.2.3. The approximate **average consensus problem** (ACP) consists in computing $\overline{x^*} = \frac{1}{n_h}\sum_{i\in\mathcal{V}_h} x_i^*$, where each node $i \in \mathcal{V}_h$ holds a value x_i^*, which is equivalent to Problem equation 5.1 with $f_i(x) = \|x - x_i^*\|^2$. In the presence of Byzantines, only an approximate estimation of $\overline{x^*}$ can be found.

When there are no Byzantine nodes, standard gossip-averaging results ensure linear convergence of the parameters to the average (Boyd et al., 2006): $\sum_{i\in\mathcal{V}} \|x_i^t - \overline{x^*}\|^2 \leq (1-\gamma)^{2t}\sum_{i\in\mathcal{V}} \|x_i^0 - \overline{x^*}\|^2$. Yet, this result is not robust to adversarial nodes: as the standard gossip communication uses averaging steps, it is *vulnerable to a single Byzantine node*, which can drive all honest nodes to any position (Blanchard et al., 2017).

5.2.2 The Dual Approach in Decentralized Optimization

The use of a dual formulation to solve decentralized optimization is very popular (Jakovetić et al., 2020; Uribe et al., 2020). Not only has it been at the heart of key advances such as acceleration (Scaman et al., 2017; Hendrikx et al., 2019; Kovalev et al., 2020) and variance reduction (Hendrikx et al., 2021), but it also allows us to understand methods such as gradient tracking (Jakovetić, 2018). It is thus natural to investigate whether the dual approach can be used for Byzantine robustness.

We first introduce the dual approach in a Byzantine-free setting. Then, we interpret dual variables and explain their use in the Byzantine-robust setting. Eventually, we instantiate the algorithm in the ACP setting and show that it recovers existing algorithms.

5.2.2.1 The Byzantine-free Dual Approach

We first show how the dual approach unrolls on Problem equation 5.1, in a setting without Byzantine agents. Let us write Problem equation 5.1 as a constrained optimization problem on $F(X) = \sum_{i\in\mathcal{V}} f_i(X_{i:})$, where the constraints write $X_{1:} = \ldots = X_{n:}$, or equivalently $\forall (i\sim j)\in\mathcal{E}, X_{i:} = X_{j:}$ for a connected graph. By encoding this constraint into a matrix $C^T \in \mathbb{R}^{\mathcal{E}\times\mathcal{V}}$, such that $[C^T X]_{(i\sim j),:} = X_{j:} - X_{i:}$ for any edge $(i\sim j)$ in $\mathcal{E}$, we obtain a *primal version* of Problem equation 5.1, which is, under convexity of functions f_i, equivalent to a *dual problem*:

$$\min_{X\in\mathbb{R}^{\mathcal{V}\times d},\ C^T X = 0} F(X) = \min_{X\in\mathbb{R}^{\mathcal{V}\times d}} \max_{\Lambda\in\mathbb{R}^{\mathcal{E}\times d}} \left\{F(X) - \langle \Lambda, C^T X\rangle\right\} \tag{Primal}$$

$$= -\min_{\Lambda\in\mathbb{R}^{\mathcal{E}\times d}} F^*(C\Lambda), \tag{Dual}$$

where $F^*(Y) := \sup_{X\in\mathbb{R}^{\mathcal{V}\times d}} \langle X, Y\rangle - F(X)$ is the Fenchel conjugate of F. Note that as F is separable, F^* can be decomposed as $F^*(Y) = \sum_{i\in\mathcal{V}} f_i^*(Y_{i:})$. The dual approach leverages

Equation (Dual) to design decentralized optimization algorithms, for instance by solving the dual problem using gradient descent (GD). If we denote $Y := C\Lambda \in \mathbb{R}^{\mathcal{V}\times d}$ the dual variable deriving from the Lagrangian multipliers $\Lambda \in \mathbb{R}^{\mathcal{E}\times d}$, GD with step-size η writes:

$$\Lambda^{t+1} = \Lambda^t - \eta C^T \nabla F^*(C\Lambda^t) \implies Y^{t+1} = Y^t - \eta CC^T \nabla F^*(Y^t). \tag{5.2}$$

Dual gossip update. The incidence matrix C is a root of the Laplacian matrix L, as $CC^T = L$. It follows that in the case of the average consensus problem where $\nabla f_i^*(y) = y + x_i^*$, Equation (5.2) consists in doing an update on the dual variable Y, which for the associated primal variable $X := \nabla F^*(Y) = Y + X^*$ recovers the standard gossip protocol $X^{t+1} = (I - \eta L)X^t$.

Convergence. Using standard GD results, Λ^t in Equation equation 5.2 converges linearly to Λ^*, the solution of equation Dual, under strong convexity and smoothness assumptions on the f_i. It follows that $X^t := \nabla F^*(C\Lambda^t)$ converges to the solution of equation Primal. Indeed:

1. *Invariant.* As $Y^t \triangleq C\Lambda^t$, $\ker C^T = \text{span}\mathbf{1}$, and $\nabla F(X^t) = Y^t$, the sum of (primal) gradients is null by design:

$$\textstyle\sum_{i\in\mathcal{V}_h} \nabla f_i(X_{i:}^t) = \mathbf{1}^T \nabla F(X^t) = \mathbf{1}^T Y^t = 0. \tag{Invariant}$$

2. *Asymptotic consensus.* Reaching a stationary point of Equation (5.2) means that $\nabla F^*(Y) \in \ker C^T$, thus $X_{i:} = X_{j:}$ for all $(i \sim j) \in \mathcal{E}$, i.e. consensus is reached among nodes.

Note that although dual gradients are arguably hard to compute in the general case, several approaches allow one to leverage the (primal-)dual approach while either bypassing the problem or alleviating this cost (Hendrikx et al., 2020a; Kovalev et al., 2020).

5.2.2.2 Duality in the Byzantine setting

In equation Dual, the Lagrangian multipliers $\Lambda^t \in \mathbb{R}^{\mathcal{E}\times d}$ correspond to the *influence* between nodes of the graph. Precisely, each row of Λ, indexed by edges as $\Lambda_{(i\sim j)}$, corresponds to the accumulated influence between nodes i and j through edge $(i \sim j)$. Indeed, the (primal) gradient is equal to the aggregation of neighboring Lagrangian multipliers, $\nabla f_i(x_i^t) \triangleq y_i^t = \sum_{(i\sim j)} \Lambda_{(i\sim j)}^t$, while the update of each $\Lambda_{(i\sim j)}^t$ multiplier involves only the two nodes of the edge: $\Lambda_{(i\sim j)}^{t+1} = \Lambda_{(i\sim j)}^t - \eta[\nabla f_j^*(y_j^t) - \nabla f_i^*(y_i^t)]$. This latter update thus corresponds to the *update of the influence* between node i and node j during a time step.

It follows that applying an operator on the update on the row $(i \sim j)$ of Λ^t corresponds to altering the influence between nodes i and j. For instance, setting the row $\Lambda_{(i\sim j)}$ to 0 corresponds to removing edge $(i \sim j)$ from the graph.

To address the vulnerability of plain gossip averaging to Byzantine parties, we regulate the influence update between nodes by clipping the update on the edges. Formally, for $\tau \geq 0$ and $x \in \mathbb{R}^d$, the projection of x onto the L_2 ball of radius τ writes:

$$\text{clip}(x;\tau) := \frac{x}{\|x\|_2} \min(\|x\|_2, \tau). \tag{5.3}$$

Algorithm 2 Byzantine-Resilient Decentralized Optimization with EdgeClippedGossip $(\boldsymbol{\tau}^t)$

Input: $\{f_j\}$, $\{y_j^0\}$, η, $\{\boldsymbol{\tau}^t\}$
for $t = 0$ **to** T **do**
  for $j = 1, \dots, n$ **in parallel do**
    Compute $x_j^t = \nabla f_j^*(y_j^t)$ if $j \in \mathcal{V}_h$ else $*$
    Share parameter x_j^t with neighbors
    $y_j^{t+1} = y_j^t - \eta \sum_{i \sim j} \mathrm{clip}\left(x_j^t - x_i^t; \boldsymbol{\tau}_{(i\sim j)}^t\right)$
  end for
end for

We thus consider for a general vector $\boldsymbol{\tau} \in \mathbb{R}^{\mathcal{E}}$, the regulation of the influence update on $(i \sim j)$, by setting $\Lambda_{(i\sim j)}^{t+1} = \Lambda_{(i\sim j)}^t - \eta \mathrm{clip}(\nabla f_j^*(y_j^t) - \nabla f_i^*(y_i^t), \boldsymbol{\tau}_{(i\sim j)}^t)$, which writes under matrix form as

$$\Lambda^{t+1} = \Lambda^t - \eta \mathrm{clip}\left(C^T \nabla F^*(C\Lambda^t), \boldsymbol{\tau}^t\right) \tag{5.4}$$

This gives Algorithm 2, called EdgeClippedGossip, which translates the update described above in terms of primal and dual iterates.

Oracle strategy. If the set of Byzantine edges $\mathcal{E}_b$ is known, then Equation 5.4 with $\tau_{(i\sim j)} = \infty \cdot 1_{(i\sim j) \notin \mathcal{E}_b}$ exactly recovers the generalized gossip algorithm 5.2 on the subgraph of honest nodes.

The previous derivations are summarized into the following result.

Proposition 5.2.4. *Consider a threshold $\tau > 0$ and $\boldsymbol{\tau} = \tau \mathbf{1}_{\mathcal{E}}$. Then, Algorithm 2 corresponds to a clipped gradient descent on the dual problem equation Dual: $\Lambda^{t+1} = \Lambda^t - \eta \Pi_\tau \left(C^T \nabla F^*(C\Lambda^t)\right)$, where Π_τ is the projection on a ball of radius τ for the infinite-2 norm $\|M\|_{\infty,2} := \sup_{(i\sim j)\in\mathcal{E}} \|M_{(i\sim j)}\|_2$.*

Next, we instantiate the algorithm in the ACP case, in which the ∇f_i^* have a special form, and which is the setup under which theoretical guarantees are given in Sections 5.3 and 5.4.

5.2.2.3 EdgeClippedGossip for the Average Consensus Problem

From now on, we focus on Algorithm 2 for the Average Consensus Problem (Definition 5.2.3).

We denote $X^* = [x_1^*, \dots, x_n^*]^T$, the matrix of node-wise optimal parameters. In the ACP setting, $X^t = \nabla F^*(C\Lambda^t) \overset{\text{ACP}}{=} C\Lambda^t + X^*$, thus Equation (5.4) writes

$$\Lambda^{t+1} = \Lambda^t - \eta \mathrm{clip}\left(C^T \nabla F^*(C\Lambda^t); \boldsymbol{\tau}^t\right) \Leftrightarrow X^{t+1} = X^t - \eta C \mathrm{clip}(C^T X^t; \boldsymbol{\tau}^t). \tag{5.5}$$

In order to distinguish the impact of honest and Byzantine nodes, we decompose Matrix C into blocks as $C = \begin{pmatrix} C_h & C_b \\ 0 & -\mathrm{I}_{\mathcal{E}_b} \end{pmatrix}$, and X as $\begin{pmatrix} X_h \\ X_b \end{pmatrix}$. The update on (Λ^t) in Proposition 5.2.4 thus writes as

$$X_h^{t+1} = X_h^t - \eta (C_h\, C_b) \mathrm{clip}(C^T X^t; \boldsymbol{\tau}^t); \quad X_b^{t+1} = *. \tag{5.6}$$

We analyze this sub-problem in the following section.

5.3 Global Clipping

We now investigate average consensus EdgeClippedGossip under a *global* clipping threshold, and highlight a choice of thresholds $(\boldsymbol{\tau}^t)_{t\geq 0}$ for which we derive convergence guarantees. All proofs for this section can be found in Appendix 5.B.2.

5.3.1 The Global Clipping Rule

To analyze Equation (5.6), we decompose the influence update between honest and Byzantine edges as $(C^T X^t)_h = C_h^T X_h^t \in \mathbb{R}^{\mathcal{E}_h \times d}$ and $(C^T X^t)_b = C_b^T X_h^t - X_b^t \in \mathbb{R}^{\mathcal{E}_b \times d}$. For a given threshold τ^t we define as κ^t the number of honest edges that are modified by $\mathrm{clip}(\cdot, \tau^t)$, i.e.,

$$\kappa^t = \mathrm{card}\left\{(i \sim j) \in \mathcal{E}_h,\ \|(C_h^T X_h^t)_{(i\sim j)}\|_2 > \tau^t\right\}. \tag{5.7}$$

We then denote $\|C_h^T X_h^t\|_{1,2;\kappa^t}$ the $1,2$-norm of the sub-matrix of clipped messages, i.e.:

$$\|C_h^T X_h^t\|_{1,2;\kappa^t} := \sum_{(i\sim j)\in\mathcal{E}_h,\ \ \kappa^t \text{ largest}} \|(C_h^T X_h^t)_{(i\sim j)}\|_2.$$

These quantities thus represent the fractions of weights of messages in $(C^T X)_h = ((x_i - x_j)_{(i\sim j)\in\mathcal{E}_h})$ that are affected by clipping. We now introduce

$$\Delta_\infty := \sup_{U_b\in\mathbb{R}^{\mathcal{E}_b\times d},\ \ \|U_b\|_{\infty,2}\leq 1} \|C_h^\dagger C_b U_b\|_{\infty,2},$$

which quantifies how much the Byzantine nodes can increase the variance of the honest nodes. We will need $\Delta_\infty < 1$ to obtain non-vacuous results, as otherwise, it means that Byzantine nodes introduce more variance than what is reduced by the contraction obtained through gossip averaging. Δ_∞ decreases when removing Byzantine edges from the graph, and can be bounded for some graphs.

Lemma 5.3.1. *For $\mathcal{G}_h$ fully connected where each honest node has exactly $N_b(i) = N_b$ Byzantine neighbors, $\Delta_\infty \leq \frac{2N_b}{n_h}\sqrt{d \wedge n_h N_b}$.*

To satisfy $\Delta_\infty < 1$, two regimes appear: it is sufficient that $n_h > 2N_b\sqrt{d}$ for small dimension d, but $n_h > 4N_b^3$ for large d, which is quickly untractable. Therefore, the limit fraction of Byzantine neighbors is significantly impacted by the dimension. We now define the following clipping threshold rule.

Definition 5.3.2 (Global Clipping Rule (GCR))**.** Given a step size η, a sequence of clipping thresholds $(\tau^s)_{s\geq 0}$, satisfies the *Global Clipping Rule* if for all $t \geq 0$, either $\tau^t = 0$, or

$$\|C_h^T X_h^t\|_{1,2;\kappa^t} \geq \Delta_\infty \|C_h^T X_h^t\|_{1,2} + \eta \frac{|\mathcal{E}_b|^2}{n_h} \sum_{0\leq s\leq t} \tau^s.$$

This clipping rule enforces that the clipping threshold is *below a certain value*, since decreasing τ^t increases κ^t, which in turn increases the left term. In particular, τ^t is reduced

until the sum of the norms of *clipped* honest edges is greater than the sum of the norms of all honest edges, shrunk by the contraction Δ_∞, plus a term proportional to the sum of previous thresholds (which is known by all nodes). This latter term, $\eta|\mathcal{E}_b|^2 \sum_{s\leq t} \tau^s/n_h$ can be understood as a bound on the maximum *bias* induced by Byzantine nodes after t steps, i.e. $1^T(\overline{X}_h^t - \overline{X}_h^0)$.

Over-clipping. The GCR defines an *upper bound* on τ^t, but any value below this threshold works. By denoting $\kappa^{t,*}$ the smallest κ^t such that the GCR holds, we must ensure that *at least* $\kappa^{t,*}$ honest edges are clipped. A solution to this end consists in clipping $\kappa^{t,*} + |\mathcal{E}_b|$ edges overall, by removing first the $|\mathcal{E}_b|$ largest edges, and computing $\kappa^{t,*}$ on the remaining edges afterwards. Note that the GCR still holds if the number of Byzantine edges is over-estimated. We provide in Section 5.B.3 a simplified (yet looser) version of the GCR.

Global coordination. Implementing a global clipping threshold that satisfies Definition 5.3.2 requires some global coordination, which is hard in general in this decentralized setting. As such, we propose a proof of concept on what can be achieved by this approach under a favorable assumption on their communication pattern, more than a practical decentralized algorithm.

5.3.2 Convergence Result

Recall that $\mu_{\max}(L_h)$ is the maximal eigenvalue of the honest subgraph Laplacian, and Δ_∞ depends only on the topology of the full graph $\mathcal{G}$. We note $\overline{X_h^*} = \mathbf{1}_h \mathbf{1}_h^T X_h^*/n_h$

Theorem 5.3.3. *Assuming $\Delta_\infty < 1$, let η be a constant step size: $\eta \leq 1/(1+|\mathcal{E}_h|(1-\Delta_\infty))\mu_{\max}(L_h)$. If the clipping thresholds $(\tau^s)_{s\geq 0}$ satisfy Definition 5.3.2 (GCR), then:*

$$\|X_h^{t+1} - \overline{X_h^*}\|_F^2 \leq \|X_h^t - \overline{X_h^*}\|_F^2 - \eta \sum_{i\sim j\in\mathcal{E}_h} \|x_i^t - x_j^t\|_2^2 \mathbf{1}_{\|x_i^t - x_j^t\|_2^2 \leq \tau^t}.$$

Theorem 5.3.3 ensures that the squared distance of honest models X_h^t to the global optimum $\overline{X_h^*}$ decreases at each step, although not necessarily all the way to 0. In essence, what happens is the following: each step of EdgeClippedGossip pulls nodes closer together but at the cost of allowing Byzantine nodes to introduce some bias so that $\overline{X_h^t} - \overline{X_h^*} \neq 0$. This trade-off appears in the GCR: at some point, further reducing the variance is not beneficial because it introduces too much bias, and so the GCR enforces that $\tau^t \approx 0$, which essentially stops the algorithm. Yet, regardless of the initial heterogeneity, nodes benefit from running EdgeClippedGossip.

Asymptotic error. This theorem ensures that finding small τ^t that satisfy the GCR reduces the distance to the optimum, yet it does not characterize the asymptotic error. Furthermore, the smallest τ^t satisfying the GCR is an oracle quantity, in the sense that it requires computing a sum of node-wise differences on honest neighbors only.

5.4 Beyond Global Coordination and Averaging

In this section, we investigate how to extend the dual approach beyond the average consensus case. We discuss the discrepancies between the developed dual framework and local clipping methods such as (He et al., 2023; Gaucher et al., 2025). Then we discuss on the difficulties to extend the dual framework beyond the average consensus setting.

5.4.1 Dual Interpretation of Existing Algorithms

Several works investigated the problem of robust average consensus. Among those, He et al. (2023); Farhadkhani et al. (2023); Gaucher et al. (2025) propose robust communication algorithms that rely, under the hood, on performing gossip communication while altering the update on the Lagrangian multipliers at each step. All those algorithms can be written as:

$$\begin{cases} x_i^{t+1} = x_i^t - \eta F\big((x_i^t - x_j^t)_{j\sim i}\big) & \text{if } i \in \mathcal{V}_h \\ x_i^{t+1} = * & \text{if } i \in \mathcal{V}_b. \end{cases} \tag{F-RG}$$

where F is an aggregation function that verifies a (b, ρ)-*robust summand properties*. The SOTA (b, ρ)-robust summand, Clipped Sum, relies on a clipping strategy: $\mathrm{CS}(z_1, \dots, z_n) := \sum_{i=1}^n \mathrm{clip}(z_i; \tau)$. Interestingly, the success of those methods relies partially on the control of the quantities $(x_i - x_j)_{i\sim j}$. In other words, they consist in altering the Lagrangian multipliers update $([C^T X^t]_{i\sim j})$ based on the norm of each entry.

Proposition 5.4.1. *When τ is a fixed constant, equation 5.6 is equivalent to Equation (F-RG) with Clipped Sum aggregation.*

Local Clipping thresholds. Another key element of the robustness of CS_+-RG, is that the clipping threshold is tailored locally, in a node-wise manner. Typically, the clipping threshold τ_i on a node i is defined as the $2b$ largest value within $(\|x_i - x_j\|)_{j\sim i}$. It follows that the influence update of two neighbors may not be symmetric when $\mathrm{clip}(x_i - x_j; \tau_i) \neq \mathrm{clip}(x_i - x_j; \tau_j)$. This asymmetry between clipping thresholds in CS-RG may bias the average of honest parameters independently of the attack of Byzantine nodes.

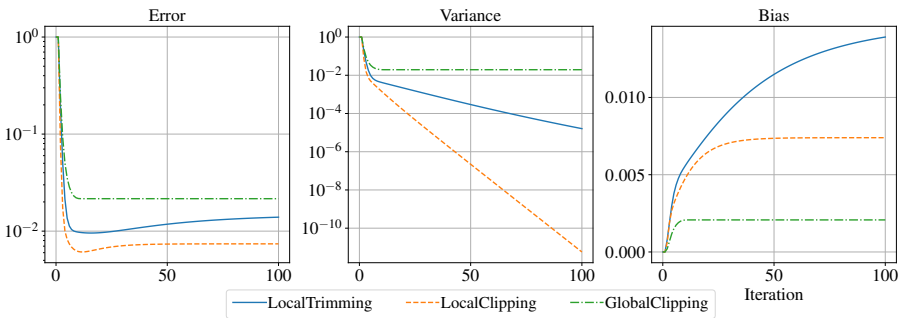


Figure 5.1: Bias - Variance decomposition of the parameters of honest nodes. GTS-RG (LocalTrimming), CS-RG (LocalClipping) and the GCR are tested under ALIE attack, in a Two Worlds graph (cf. Section 5.5) with 64 honest nodes in two cliques, each being neighbor to 16 nodes in the other clique, and to 3 Byzantine nodes.

Trading bias and consensus. This bias induced by the communication scheme is a fundamental difference with the dual approach such as exposed in Algorithm 2, where the bias is only induced by the Byzantines nodes. This discrepancy is reflected in the analysis: the bias of Algorithm 2 is controlled directly in the GCR, while for instance (Gaucher et al., 2025) controls the additional bias at each step, and sums them up by using that honest nodes converge quickly enough to the same value. Overall, Algorithm 2 with the GCR does not alter the average of honest parameters and thus can induce less bias in the communication process than local clipping methods. Yet, there is no free lunch: the variance of the honest parameters does not converge to 0 using the GCR, while CS-RG for instance ensures asymptotic consensus. We illustrate this point in Figure 5.1.

5.4.2 Dual Formalism for Asymmetric Clipping

Considering the weak convergence guarantee of the GCR, we modify slightly the dual formalism to allow local clipping thresholds. Then we discuss the major difficulties encountered when extending the dual approach beyond the case of average consensus.

From symmetric to asymmetric clipping. In Byzantine-free settings (see Section 5.2.2.1), dual gradient descent leads to gossip algorithms for which the influence of node i on node j is the opposite of the one of node j on node i, both are stored in $\Lambda^t_{i\sim j}$. The matrix of influence updates $C^T\nabla F^*(C\Lambda^t)$ thus has one row for each edge, the update can be written as equation 5.2, and the equation Invariant property is kept. This symmetry between the influence of two honest nodes $(i,j)\in\mathcal{V}_h$ remains when the influence update is clipped, as long as both honest neighbors clip the influence update on edge $(i\sim j)$ with the same threshold. Allowing local clipping thresholds requires storing in two different variables the influence from node i on node j and from node i to j.

To do so, we consider $\tilde{\mathcal{G}}=(\mathcal{V},\tilde{\mathcal{E}})$ the directed version of the undirected graph $\mathcal{G}$, we enumerate its set of edges as $\tilde{\mathcal{E}}=\{i\to j\}$, and consider the *directed* incidence matrix $\tilde{C}^T\in\mathbb{R}^{\tilde{\mathcal{E}}\times\mathcal{V}}$, such that for $X\in\mathbb{R}^{\mathcal{V}\times d}$, and $i\to j\in\tilde{\mathcal{E}}$, $(\tilde{C}^TX)_{i\to j}=X_{j,:}-X_{i,:}$. Furthermore, we define $\tilde{B}\in\mathbb{R}^{\mathcal{V}\times\tilde{\mathcal{E}}}$ as the positive coordinates of the incidence matrix $\tilde{C}$, such that for $\tilde{\Lambda}\in\mathbb{R}^{\tilde{\mathcal{E}}\times d}$ and $j\in\mathcal{V}$, we have $[\tilde{B}\tilde{\Lambda}]_j=\sum_{(i\to j)\in\tilde{\mathcal{E}}}\tilde{\Lambda}_{i\to j}$. Under these notations, Equation (5.2) can be written with directed edge-wise clipping thresholds $\boldsymbol{\tau}\in\mathbb{R}^{\tilde{\mathcal{E}}}$ as:

$$\tilde{\Lambda}^{t+1}=\tilde{\Lambda}^t-\eta\mathrm{clip}(\tilde{C}^T\nabla F^*(\tilde{B}\tilde{\Lambda}^t),\boldsymbol{\tau}^t)\implies\tilde{Y}^{t+1}=\tilde{Y}^t-\eta\tilde{B}\mathrm{clip}(\tilde{C}^T\nabla F^*(\tilde{Y}^t);\boldsymbol{\tau}^t).\quad(5.8)$$

And, in the average consensus sub-case, Equation (5.5) generalizes as

$$\tilde{X}^{t+1}=\tilde{X}^t-\eta\tilde{B}\mathrm{clip}(\tilde{C}^T\tilde{X}^t;\boldsymbol{\tau}^t).\quad(5.9)$$

Node-wise clipping thresholds. This formalism allows node-wise clipping thresholds as a sub-case of directed edge-wise clipping thresholds. Thus, communication algorithms with node-wise clipping thresholds as ClippedGossip in (He et al., 2023) or CS-RG in Gaucher et al. (2025) exactly correspond to Equation (5.9) for appropriate $\boldsymbol{\tau}^t$.

Proposition 5.4.2. *Denoting* $M := \tilde{B}^\dagger \tilde{C}$*, Equation* (5.8) *can be seen as a* clipped pre-conditioned *gradient descent on the dual objective* $\Lambda \to F^*(C\Lambda)$*:*

$$\tilde{\Lambda}^{t+1} = \tilde{\Lambda}^t - \eta \text{clip}(M^T \tilde{B}^T \nabla F^*(\tilde{B}\tilde{\Lambda}^t), \boldsymbol{\tau}). \tag{5.10}$$

In particular, we expect fixed points of this iteration to be minimizers of $\tilde{\Lambda} \mapsto F^*(\tilde{B}\tilde{\Lambda})$ *in the absence of Byzantines.*

Loss of the invariant. As noted before, even in the absence of Byzantines, the equation Invariant property, i.e. $\mathbf{1}^T \tilde{Y}^t = 0$, does not hold *by design* anymore, since $\mathbf{1}^T \tilde{B} \neq 0$.

Error metric. The main difficulty in extending the dual approach to more generic loss functions, even with a global clipping threshold, lies in this loss of invariant. The Fenchel transform $F^*(Y_h^t)$ is no longer informative of the distance between $\nabla F^*(Y_h^t)$ and the solution of the problem equation Primal when $\mathbf{1}^T Y_h^t \neq 0$. Local clipping analyses such as F-RG bypass this difficulty by directly proving a linear decrease for the *variance* of honest nodes.

Lack of Lyapunov function Seeing *Equation* (5.8) as noisy variation of *Equation* (5.2) leads to writing $\tilde{Y}^{t+1} = \tilde{Y}^t - \eta L \nabla F^*(\tilde{Y}^t) + \eta E^t$, where E^t is the error induced by both Byzantine corruption and clipping of honest edges. Note that E^t may lie in the kernel of L. If it is possible to use F-RG analysis to upper-bound the error as $\|E^t\|^2 \leq \rho b \|\sqrt{L}\nabla F^*(\tilde{Y}_h^t)\|^2$, it is much more involved to find a proper Lyapunov function to leverage this bound. One can consider $F^*(PY^t)$, with P being the projection on span$1^\perp$. However, at some point it is necessary to control $P\left(\nabla F^*(\tilde{Y}_h^t) - \nabla F^*(P\tilde{Y}_h^t)\right)$, which leads to vacuous results beyond the average consensus case.

Conclusion. In summary, the strength of the dual approach generally lies in its structure, which usually allows for building Lyapunov functions in a principled manner. The Byzantine case breaks this structure, which significantly hardens the analysis.

5.5 Experiments

We perform all experiments on a 'Two Worlds' graph with 64 nodes, made of two cliques of 32 honest nodes. Each node is connected to 16 nodes in the other clique. On such a graph, the theoretical maximal number of tolerated Byzantine neighbors per honest node is 16. The actual number of Byzantine neighbors per honest node depends on the experiment. We initialize the parameter of each honest node using a $N(0, I_5)$ distribution, each experimental configuration is run with 5 different seeds, and results are averaged. We implement ALIE (Baruch et al., 2019), FOE (Xie et al., 2020), Dissensus (He et al., 2023) and Spectral Heterogeneity (Gaucher et al., 2025), more details are available in Section 5.A.1. We compare 1) the Global Clipping rule, the threshold is the largest clipping threshold verifying the GCR, 2) Local Clipping denotes the algorithm CS-RG (Gaucher et al., 2025), 3) Local Trimming denotes GTS-RG, i.e. the implementation of NNA (Farhadkhani et al., 2022) in the case of sparse communication networks (Gaucher et al., 2025).

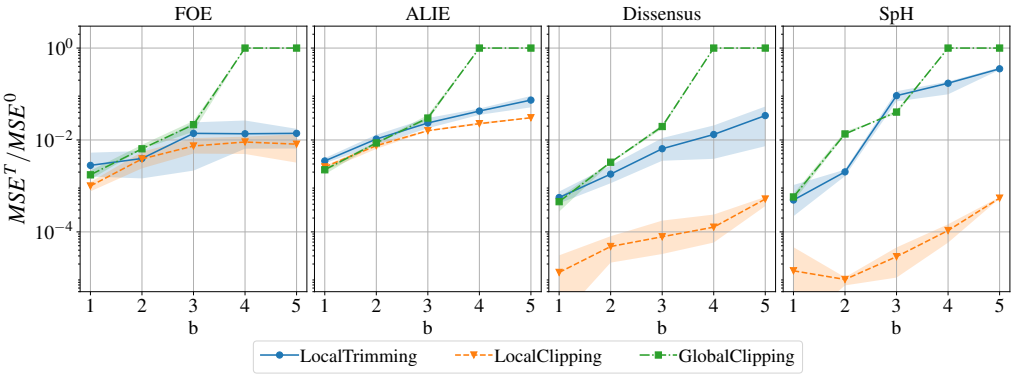


Figure 5.2: Relative mean square error after $T = 100$ communication steps, with a varying number of Byzantine neighbors to each honest node.

Analysis. We note on Figure 5.2 that the GCR has similar *worst-case*[1] performances on the investigated task than methods with local aggregation rules (i.e. GTS-RG and CS-RG). However, as soon as $\Delta_\infty \geq 1$, the GCR does not allow communication between nodes anymore, and we have $\text{MSE}^T/\text{MSE}^0 = 1$. This follows from the definition of the GCR itself: by design, the GCR is built such as to *ensure that nodes benefit from the communication*. As such, when there is no guarantee that this is the case anymore, the algorithms prevent communication between nodes. On the contrary, node-wise aggregation methods are provably robust up to a certain amount of Byzantines but do not have any guarantees beyond this breakdown point, and they do *increase* the mean square error by communicating – see e.g. Gaucher et al. (2025).

5.6 Conclusion

In this paper, we take the dual approach to design a byzantine-robust algorithm, which we then analyze in the average consensus case. We then use the dual approach to re-interpret existing efficient robust communication algorithms, and we experimentally compare our algorithms with them. Finally, we highlight that the usual strength of the dual approach to decentralized learning - its structure - is broken by the introduction of Byzantine nodes, which brings significant challenges to the analysis. Yet, interesting algorithms arise, which do not ensure consensus among nodes (as standard algorithms do, at the price of some bias). Finding new ways of better characterizing the convergence properties of such algorithms is an interesting open problem.

[1]Note that the empirical worst-case performance among all attacks is the meaningful evaluation of the resilience of an algorithm, and thus the superior performance of LocalClipping against Dissensus and SpH only denotes that these attacks are not optimal against LocalClipping, and not that LocalClipping has superior robustness.

Appendix

5.A Experimental design

5.A.1 Byzantine attacks in the federated case

In our experiments, we implement SOTA attacks developed for the federated SGD setting: *Fall of empire* (FOE) from Xie et al. (2020) and *A little is enough* (ALIE) from Baruch et al. (2019), *Dissensus* from He et al. (2023) and *Spectral Heterogeneity* for Gaucher et al. (2025). We implement all these methods by making a Byzantine node $j \in \mathcal{V}_b$ declare to an honest neighbor $i \in \mathcal{V}_h$ a vector $x_j^t = x_i^t + \varepsilon_t^t a_i^t$, where $a_i^t \in \mathbb{R}^d$ is the attack direction on node i and $\varepsilon_i^t \geq 0$ is the scale of the attack.

- **ALIE**. The Byzantine nodes compute the coordinate-wise standard deviation σ^t. Then they use $a_i^t = \sigma^t$.
- **FOE**. The Byzantine nodes compute the mean of the honest parameters $\overline{x^t}$, and declare $a_i^t = \overline{x^t}$.
- **Dissensus**. The Byzantines take $a_i^t = \sum_{k \sim j,\, k \in \mathcal{V}_h} x_i^t - x_k^t$.
- **Spectral Heterogeneity**. The Byzantines compute the Fiedler vector of the honest subgraph e_{fied}, i.e. the eigenvector associated with the second smallest eigenvalue of the Laplacian of the honest subgraph. Then, they take $a_i^t = [e_{fied}]_i \sum_{k \in \mathcal{V}_h} [e_{fied}]_k x_k^t$.

In all attacks we chose ε_i^t to maximize the error induced by Byzantines: for trimming we take it such that all Byzantine attacks are just below the trimming threshold, while for clipping strategies we chose $e_i^t >> 1$.

5.B Proofs

5.B.1 Notations

In this subsection we resume the different notations of the proofs. Table 5.1 summarizes notations used for primal and dual parameters, as well as Lagrange multipliers.

We recall the algorithm in the global clipping setting:

$$X_h^{t+1} = X_h^t - \eta(C_h\, C_b)\mathrm{clip}(C^T X^t; \boldsymbol{\tau}^t)$$
$$X_b^{t+1} = *.$$

Table 5.1: Summary of notations

Variable	Name	Size	Rows
X	primal parameter	$\mathcal{V} \times d$	$(x_i)_{i\in\mathcal{V}}$
Y	dual parameter	$\mathcal{V} \times d$	$(y_i)_{i\in\mathcal{V}}$
Λ	Lagrange multipliers	$\mathcal{E} \times d$	$(\lambda_{(i\sim j)})_{(i\sim j)\in\mathcal{E}}$
$C^T\nabla F^*(C\Lambda)$	Influence update on the edges	$\mathcal{E} \times d$	$(x_i - x_j)_{(i\sim j)\in\mathcal{E}}$

In the global clipping setting, we consider that, as $\boldsymbol{\tau}^t = \tau^t \mathbf{1}_{\mathcal{E}}$, we can overload the clipping operators writing for $\Lambda_h \in \mathbb{R}^{\mathcal{E}_h \times d}$ and $\Lambda_b \in \mathbb{R}^{\mathcal{E}_b \times d}$,

$$\mathrm{clip}(\Lambda_h; \tau^t) = [\mathrm{clip}(\lambda_{(i\sim j)}, \tau^t)]_{(i\sim j)\in\mathcal{E}_h} \tag{5.11}$$

$$\mathrm{clip}(\Lambda_b; \tau^t) = [\mathrm{clip}(\lambda_{(i\sim j)}, \tau^t)]_{(i\sim j)\in\mathcal{E}_b} \tag{5.12}$$

We use the following notations:

- For any matrix $M \in \mathbb{R}^{\mathcal{V}_h \times d}$, we denote
$$\overline{M} := \frac{1}{n_h} \mathbf{1}_{n_h} \mathbf{1}_{n_h}^T M.$$
- We note the primal parameter re-centered on the solution
$$Z_h^t := X_h^t - \overline{X_h^*}.$$
- The unitary matrix of attack on edges linking to Byzantine nodes
$$U_b^t := \frac{1}{\eta\tau^t}(\Lambda_b^{t+1} - \Lambda_b^t).$$
Thus $\|U_b^t\|_{\infty,2} \leq 1$ and $\Lambda_b^{t+1} = \Lambda_b^t - \eta\mathrm{clip}([C^T X^t]_b; \tau^t) = \Lambda_b^t + \eta\tau^t U_b^t$.
- The error term induced by clipping and Byzantine nodes
$$E^t := C_h \left(C_h^T Z_h^t - \mathrm{clip}(C_h^T Z_h^t; \tau^t)\right) + \tau^t C_b U_b^t,$$
- We denote for $p \in \{1, 2\}$ and $\kappa^t \in \{0, \ldots, |\mathcal{E}_h|\}$
$$\|C_h^T X_h^t\|_{p;\kappa^t}^p := \sum_{(i\sim j)\in\mathcal{E}_h,\ \ \kappa^t \text{ largest}} \|(C_h^T X_h^t)_{(i\sim j)}\|_2^p$$
$$\|C_h^T X_h^t\|_{p;-\kappa^t}^p := \|C_h^T X_h^t\|_p^p - \|C_h^T X_h^t\|_{p;\kappa^t}^p.$$
- We recall that by default $\|\cdot\|$ and $\|\cdot\|_2$ is the usual Euclidean norm (or the Frobenius norm in case of matrix).

And we recall some previous facts:

- $L_h = C_h C_h^T = \frac{1}{2}\tilde{C}_h \tilde{C}_h^T$ the Lagrangian matrix on the honest subgraph $\mathcal{G}_h$.
- $C_h^T \mathbf{1}_{n_h} = 0 \implies C_h^T \overline{X_h^t} = 0 \implies C_h^T X_h^t = C_h^T Z_h^t$.
- $C_h^\dagger$ is the Moore-Penrose pseudo inverse of C_h.

5.B.2 Analysis of Global Clipping Rule (GCR) Theorem 5.3.3

Lemma 5.B.1. *(Gossip - Error decomposition) The iterate update can be decomposed as*

$$Z_h^{t+1} = (I_h - \eta L_h) Z_h^t + \eta E^t.$$

Proof. The actualization of X_h^t writes

$$X_h^{t+1} = X_h^t - \eta C_h \text{clip}(C_h^T X_h^t; \tau^t) - \eta C_b \text{clip}([C^T X^t]_b; \tau^t).$$

Using $C_h^T X_h^t = C_h^T Z_h^t$, we get

$$\begin{aligned} Z_h^{t+1} &= Z_h^t - \eta C_h \text{clip}(C_h^T X_h^t; \tau^t) + \eta \tau^t C_b U_b^t \\ &= (I_h - \eta C_h C_h^T) Z_h^t + \eta \left\{ C_h \left(C_h^T Z_h^t - \text{clip}(C_h^T Z_h^t; \tau^t) \right) + \tau^t C_b U_b^t \right\} \\ &= (I_h - \eta L_h) Z_h^t + \eta E^t. \end{aligned}$$

□

Lemma 5.B.2. *(Sufficient decrease on the variance) We have the decomposition of the variance as a biased gradient descent:*

$$\|Z_h^{t+1} - \overline{Z_h^t}\|^2 = \|Z_h^t - \overline{Z_h^t}\|^2 + \eta^2 \|L_h Z_h^t - E^t\|^2 - 2\eta \|C_h^T Z_h^t\|^2 + 2\eta \langle Z_h^t - \overline{Z_h^t}, E^t \rangle.$$

Proof. we start from Lemma 5.B.1

$$\begin{aligned} \|Z_h^{t+1} - \overline{Z_h^t}\|^2 &= \|Z_h^t - \overline{Z_h^t} - \eta(L_h Z_h^t - E^t)\|^2 \\ &= \|Z_h^t - \overline{Z_h^t}\|^2 + \eta^2 \|L_h Z_h^t - E^t\|^2 - 2\eta \langle L_h Z_h^t, Z_h^t - \overline{Z_h^t} \rangle + 2\eta \langle Z_h^t - \overline{Z_h^t}, E^t \rangle \\ &= \|Z_h^t - \overline{Z_h^t}\|^2 + \eta^2 \|L_h Z_h^t - E^t\|^2 - 2\eta \|C_h^T Z_h^t\|^2 + 2\eta \langle Z_h^t - \overline{Z_h^t}, E^t \rangle, \end{aligned}$$

where we used that $C_h^T \overline{Z_h^t} = 0$ and $C_h C_h^T = L_h$. □

Lemma 5.B.3. *(complete sufficient decrease)*

$$\|Z_h^{t+1}\|^2 = \|Z_h^t\|^2 + \eta^2 \|L_h Z_h^t - E^t\|^2 - 2\eta \|C_h^T Z_h^t\|^2 + 2\eta \langle Z_h^t - \overline{Z_h^t}, E^t \rangle + 2\langle \overline{Z_h^t}, \overline{Z_h^{t+1}} - \overline{Z_h^t} \rangle.$$

Proof. We leverage that

$$\begin{aligned} \|Z_h^{t+1}\|^2 &= \|Z_h^{t+1} - \overline{Z_h^t} + \overline{Z_h^t}\|^2 \\ &= \|Z_h^{t+1} - \overline{Z_h^t}\|^2 + \|\overline{Z_h^t}\|^2 + 2\langle \overline{Z_h^t}, Z_h^{t+1} - \overline{Z_h^t} \rangle \\ &= \|Z_h^{t+1} - \overline{Z_h^t}\|^2 + \|\overline{Z_h^t}\|^2 + 2\langle \overline{Z_h^t}, \overline{Z_h^{t+1}} - \overline{Z_h^t} \rangle. \end{aligned}$$

Then we use Lemma 5.B.2. □

Assumption 5.B.4. Definition 5.3.2 Trade-off between clipping error and Byzantine influence. By denoting

$$\Delta_\infty := \sup_{\substack{U_b \in \mathbb{R}^{\mathcal{E}_b \times d} \\ \|U_b\|_{\infty,2} \leq 1}} \|C_h^\dagger C_b U_b\|_{\infty,2},$$

We assume that $\tau^t = 0$, or

$$\|C_h^T X_h^t\|_{\kappa^t} \geq \Delta_\infty \|C_h^T X_h^t\|_1 + \eta \frac{|\mathcal{E}_b|^2}{n_h} \sum_{0 \leq s \leq t} \tau^s.$$

Lemma 5.B.5. *(Join control of the first order error and second order bias) Under Definition 5.3.2, we have that*

$$\eta \langle Z_h^t - \overline{Z_h^t}, E^t \rangle + \langle \overline{Z_h^t}, \overline{Z_h^{t+1}} - \overline{Z_h^t} \rangle + \eta^2 \|\overline{E^t}\|_2^2 \leq \eta \|C_h^T Z_h^t\|_{\kappa^t}^2,$$

or $\tau^t = 0$.

Proof. We begin by upper bounding precisely both error terms:

1. Variance-induced error.

$$\begin{aligned}\langle Z_h^t - \overline{Z_h^t}, E^t \rangle &= \langle Z_h^t - \overline{Z_h^t}, C_h \left(C_h^T Z_h^t - \mathrm{clip}(C_h^T Z_h^t; \tau^t) \right) \rangle + \langle Z_h^t - \overline{Z_h^t}, \tau^t C_b U_b^t \rangle \\ &= \langle C_h^T Z_h^t, C_h^T Z_h^t - \mathrm{clip}(C_h^T Z_h^t; \tau^t) \rangle + \langle C_h^T Z_h^t, \tau^t C_h^\dagger C_b U_b^t \rangle,\end{aligned}$$

Using that $C_h^T \overline{Z_h^t} = 0$. On one side we have that,

$$\begin{aligned}\langle C_h^T Z_h^t, C_h^T Z_h^t - \mathrm{clip}(C_h^T Z_h^t; \tau^t) \rangle &= \sum_{(i \sim j) \in \mathcal{E}_h} \langle z_i^t - z_j^t, z_i^t - z_j^t - \mathrm{clip}(z_i^t - z_j^t; \tau^t) \rangle \\ &= \sum_{(i \sim j) \in \mathcal{E}_h} \|z_i^t - z_j^t\|_2 \left(\|z_i^t - z_j^t\|_2 - \tau^t \right)_+ .\end{aligned}$$

One the other side, by using that, for $M, N \in \mathbb{R}^{\mathcal{E}_h \times d}$, $\langle M, N \rangle \leq \|M\|_{1,2} \|N\|_{\infty,2}$, as:

$$\begin{aligned}\langle M, N \rangle &= \sum_{(i \sim j) \in \mathcal{E}_h} \langle M_{(i \sim j)}, N_{(i \sim j)} \rangle \\ &\leq \sum_{(i \sim j) \in \mathcal{E}_h} \|M_{(i \sim j)}\|_2 \|N_{(i \sim j)}\|_2 \\ &\leq \|N\|_{\infty,2} \sum_{(i \sim j) \in \mathcal{E}_h} \|M_{(i \sim j)}\|_2,\end{aligned}$$

we can upper bound the second term

$$\langle C_h^T Z_h^t, \tau^t C_h^\dagger C_b U_b^t \rangle \leq \tau^t \|C_h^T Z_h^T\|_{1,2} \|C_h^\dagger C_b U_b^t\|_{\infty,2}.$$

Then by defining $\Delta_\infty = \sup_{U_b \in \mathbb{R}^{\mathcal{E}_b \times d}} \|C_h^\dagger C_b U_b^t\|_{\infty,2}$, using $\|C_h^T Z_h^t\|_{1,2} = \sum_{(i\sim j)\in\mathcal{E}_h} \|z_i^t - z_j^t\|_2$, we have that

$$\begin{aligned}\langle Z_h^t - \overline{Z_h^t}, E^t\rangle &\leq \sum_{(i\sim j)\in\mathcal{E}_h} \|z_i^t - z_j^t\|_2 \left(\|z_i^t - z_j^t\|_2 - \tau^t\right)_+ + \tau^t \Delta_\infty \sum_{(i\sim j)\in\mathcal{E}_h} \|z_i^t - z_j^t\|_2 \\ &= \sum_{(i\sim j)\in\mathcal{E}_h} \|z_i^t - z_j^t\|_2^2 1_{\|z_i^t - z_j^t\|_2 > \tau^t} \\ &\quad - \tau^t \left(\sum_{(i\sim j)\in\mathcal{E}_h} \|z_i^t - z_j^t\|_2 1_{\|z_i^t - z_j^t\|_2 > \tau^t} - \Delta_\infty \sum_{(i\sim j)\in\mathcal{E}_h} \|z_i^t - z_j^t\|_2^2 \right).\end{aligned}$$

Let's consider any $\kappa^t \in \mathbb{N}$ such that

$$\kappa^t \in \left[\sum_{(i\sim j)\in\mathcal{E}_h} 1_{\|z_i^t - z_j^t\|_2 > \tau^t}, \sum_{(i\sim j)\in\mathcal{E}_h} 1_{\|z_i^t - z_j^t\|_2 \geq \tau^t} - 1 \right].$$

Then, using

$$\|C_h^T Z_h^t\|_{p,\kappa^t}^p := \sum_{\substack{(i\sim j)\in\mathcal{E}_h \\ \kappa^t \text{ largest}}} \|z_i^t - z_j^t\|_2^p,$$

the previous inequality writes

$$\langle Z_h^t - \overline{Z_h^t}, E^t\rangle \leq \|C_h^T Z_h^t\|_{2,\kappa^t}^2 - \tau^t \left(\|C_h^T Z_h^t\|_{1,\kappa^t} - \Delta_\infty \|C_h^T Z_h^t\|_1\right).$$

From this upper bound we could derive a clipping rule taking only into account the error induced by clipping and Byzantine nodes on the variance. In this analysis, we will include the second error term induced by the bias.

2. Bias-induced error.
We consider the term $\langle \overline{Z_h^t}, \overline{Z_h^{t+1}} - \overline{Z_h^t}\rangle$. We remark that, using that $Z_h^{t+1} = (I_h - C_h C_h^T) Z_h^t + \eta E^t$, and considering that $\mathbf{1}_{n_h}^T C_h = 0$, using the definition of E^t we get:

$$\mathbf{1}_{n_h}^T Z_h^{t+1} = \mathbf{1}_{n_h}^T Z_h^t + \eta\tau^t \mathbf{1}_{n_h}^T C_b U_b^t = \sum_{s=0}^t \eta\tau^s \mathbf{1}_{n_h}^T C_b U_b^s.$$

Using this, we remark that

$$\eta^2 \|\overline{E^t}\|_2^2 = \|\overline{Z^{t+1}} - \overline{Z^t}\|_2^2 \implies \eta^2 \|\overline{E^t}\|_2^2 + \langle \overline{Z_h^t}, \overline{Z_h^{t+1}} - \overline{Z_h^t}\rangle = \langle \overline{Z_h^{t+1}}, \overline{Z_h^{t+1}} - \overline{Z_h^t}\rangle.$$

Thus we get

$$\begin{aligned}\langle \overline{Z_h^{t+1}}, \overline{Z_h^{t+1}} - \overline{Z_h^t}\rangle &= \left\langle \sum_{s=0}^t \eta\tau^s \frac{1}{n_h} \mathbf{1}_{n_h} \mathbf{1}_{n_h}^T C_b U_b^s, \eta\tau^t \frac{1}{n_h} \mathbf{1}_{n_h} \mathbf{1}_{n_h}^T C_b U_b^t \right\rangle \\ &= \frac{\tau^t \eta^2}{n_h} \sum_{s=0}^t \tau^s \left\langle \mathbf{1}_{n_h}^T C_b U_b^s, \mathbf{1}_{n_h}^T C_b U_b^t \right\rangle \\ &= \frac{\tau^t \eta^2}{n_h} \sum_{s=0}^t \tau^s \left\langle \sum_{i\in\mathcal{V}_h} N_b(i) u_i^s, \sum_{i\in\mathcal{V}_h} N_b(i) u_i^t \right\rangle.\end{aligned}$$

Where u_i^t is the mean vector of Byzantine neighbors of node $i \in \mathcal{V}_h$, so that $\|u_i^t\|_2 \leq 1$, thus we get

$$\langle \overline{Z_h^t}, \overline{Z_h^{t+1}} - \overline{Z_h^t} \rangle \leq \frac{\tau^t \eta^2}{n_h} \sum_{s=0}^{t} \tau^s \left(\sum_{i \in \mathcal{V}_h} N_b(i) \right)^2 = \frac{\eta^2 \tau^t |\mathcal{E}_b|^2}{n_h} \sum_{s=0}^{t} \tau^s.$$

Now that we controlled both terms, we can mix both error terms:

$$\begin{aligned}
&\langle Z_h^t - \overline{Z_h^t}, E^t \rangle + \eta^{-1} \langle \overline{Z_h^t}, \overline{Z_h^{t+1}} - \overline{Z_h^t} \rangle \\
&\leq \|C_h^T Z_h^t\|_{2,\kappa^t}^2 - \tau^t \left(\|C_h^T Z_h^t\|_{1,\kappa^t} - \Delta_\infty \|C_h^T Z_h^t\|_1 \right) + \tau^t \frac{\eta |\mathcal{E}_b|^2}{n_h} \sum_{s=0}^{t-1} \tau^s \\
&= \|C_h^T Z_h^t\|_{2,\kappa^t}^2 - \tau^t \left(\|C_h^T Z_h^t\|_{1,\kappa^t} - \Delta_\infty \|C_h^T Z_h^t\|_1 - \frac{\eta |\mathcal{E}_b|^2}{n_h} \sum_{s=0}^{t} \tau^s \right).
\end{aligned}$$

Hence, as long as

$$\|C_h^T Z_h^t\|_{1,\kappa^t} \geq \Delta_\infty \|C_h^T Z_h^t\|_1 + \frac{\eta |\mathcal{E}_b|^2}{n_h} \sum_{s=0}^{t} \tau^s, \tag{5.13}$$

we have that

$$\langle Z_h^t - \overline{Z_h^t}, E^t \rangle + \eta^{-1} \langle \overline{Z_h^{t+1}}, \overline{Z_h^{t+1}} - \overline{Z_h^t} \rangle \leq \|C_h^T Z_h^t\|_{2,\kappa^t}^2.$$

When Equation (5.13) cannot be enforced using $\kappa^t \leq |\mathcal{E}_h| - 1$, then we take $\tau^t = 0$ (thus $\kappa^t = |\mathcal{E}_h|$), everything is clipped and nodes parameters don't move anymore. □

Lemma 5.B.6. *(Control of first and 2nd order error terms together) Assume Definition 5.3.2 and that*

$$\eta \leq \frac{1}{\mu_{\max}(L_h)},$$

then by denoting the first and 2nd order error term, divided by η as

$$\zeta^t := \eta \|L_h Z_h^t - E^t\|^2 + 2\langle Z_h^t - \overline{Z_h^t}, E^t \rangle + 2\eta^{-1} \langle \overline{Z_h^t}, \overline{Z_h^{t+1}} - \overline{Z_h^t} \rangle.$$

We have the control:

$$\zeta^t \leq \eta \mu_{\max}(L_h) \left(\|C_h^T Z_h^t\|_{2,-\kappa^t}^2 + (\tau^t)^2 \left[\kappa^t (1 - 2\Delta_\infty) + \Delta_\infty^2 |\mathcal{E}_h| \right] \right) + 2\|C_h^T Z_h^t\|_{2,\kappa^t}^2.$$

Proof. This part of the proof allows to compute the maximum stepsize usable. We want to control the discretization term $\eta^2 \|L_h Z_h^t - E^t\|_2^2$. If $E^t = 0$ it would have been done using a step-size small enough. In this case we will do the same, but we will monitor the interactions between the discretization and the control of the error. Note that

$$E^t - L_h Z_h^t = C_h \text{clip}(C_h^T Z_h^t; \tau^t) + \tau^t C_b U_b^t, \tag{5.14}$$

and that at the first order $\langle C_g \text{clip}(C_g^T Z_h^t; \tau^t), \tau^t C_b U_b^t \rangle \leq 0$ if Byzantine aims at preventing

the convergence. Thus, we will include this second-order term in the proof of the Lemma 5.B.2.

We denote by $\mu_{\max}(L_h)$ the largest eigenvalue of L_h.

We denote the first and 2nd order error term, divided by η as

$$\zeta^t := \eta\|L_h Z_h^t - E^t\|^2 + 2\langle Z_h^t - \overline{Z_h^t}, E^t\rangle + 2\eta^{-1}\langle \overline{Z_h^t}, \overline{Z_h^{t+1}} - \overline{Z_h^t}\rangle.$$

Hence $\|Z_h^{t+1}\|^2 \leq \|Z_h^t\|^2 + \eta\zeta^t - 2\eta\|C_h^T Z_h^t\|^2$.

Using

$$\begin{aligned}\|E^t - L_h Z_h^t\| &= \|C_h(\mathrm{clip}(C_h^T Z_h^t;\tau^t) - \tau^t C_h^\dagger C_b U_b^t) + \overline{E^t}\|^2 \\ &= \|C_h(\mathrm{clip}(C_h^T Z_h^t;\tau^t) - \tau^t C_h^\dagger C_b U_b^t)\|^2 + \|\overline{E^t}\|^2,\end{aligned}$$

we have:

$$\begin{aligned}\zeta^t &= \eta\|C_h(\mathrm{clip}(C_h^T Z_h^t;\tau^t) - \tau^t C_h^\dagger C_b U_b^t)\|_2^2 + \eta\|\overline{E^t}\|_2^2 \\ &\quad + 2\langle Z_h^t - \overline{Z_h^t}, E^t\rangle + 2\eta^{-1}\langle \overline{Z_h^t}, \overline{Z_h^{t+1}} - \overline{Z_h^t}\rangle \\ &\leq \eta\mu_{\max}(L_h)\left(\|\mathrm{clip}(C_h^T Z_h^t;\tau^t)\|^2 + \|\tau^t C_h^\dagger C_b U_b^t\|^2\right) \\ &\quad - 2\eta\mu_{\max}(L_h)\langle \mathrm{clip}(C_h^T Z_h^t) - C_h^T Z_h^t + C_h^T Z_h^t, \tau^t C_h^\dagger C_b U_b^t\rangle \\ &\quad + 2\langle C_h^T Z_h^t, \tau^t C_h^\dagger C_b U_b^t\rangle + 2\langle C_h^T Z_h^t, C_h^T Z_h^t - \mathrm{clip}(C_h^T Z_h^t;\tau^t)\rangle \\ &\quad + 2\eta^{-1}\langle \overline{Z_h^{t+1}}, \overline{Z_h^{t+1}} - \overline{Z_h^t}\rangle \\ &= \eta\mu_{\max}(L_h)\left(\|\mathrm{clip}(C_h^T Z_h^t;\tau^t)\|^2 + \|\tau^t C_h^\dagger C_b U_b^t\|^2\right) \\ &\quad + 2\langle C_h^T Z_h^t, C_h^T Z_h^t - \mathrm{clip}(C_h^T Z_h^t;\tau^t)\rangle + 2\eta^{-1}\langle \overline{Z_h^{t+1}}, \overline{Z_h^{t+1}} - \overline{Z_h^t}\rangle \\ &\quad + 2(1-\eta\mu_{\max}(L_h))\underbrace{\langle C_h^T Z_h^t, \tau^t C_h^\dagger C_b U_b^t\rangle}_{\leq \|C_h^T Z_h^t\|_1 \Delta_\infty \tau^t} \\ &\quad + 2\eta\mu_{\max}(L_h)\underbrace{\langle C_h^T Z_h^t - \mathrm{clip}(C_h^T Z_h^t;\tau^t), \tau^t C_h^\dagger C_b U_b^t\rangle}_{\leq \|C_h^T Z_h^t - \mathrm{clip}(C_h^T Z_h^t;\tau^t)\|_1 \Delta_\infty \tau^t}.\end{aligned}$$

Then we use that

$$\|C_h^T Z_h^t - \mathrm{clip}(C_h^T Z_h^t;\tau^t)\|_1 = \|C_h^T Z_h^t\|_1 - \|\mathrm{clip}(C_h^T Z_h^t;\tau^t)\|_1,$$

to get, for $\eta \leq \mu_{\max}(L_h)$,

$$\begin{aligned}\zeta^t &\leq \eta\mu_{\max}(L_h)\left(\|\mathrm{clip}(C_h^T Z_h^t;\tau^t)\|^2 + \|\tau^t C_h^\dagger C_b U_b^t\|^2\right) \\ &\quad + 2\langle C_h^T Z_h^t, C_h^T Z_h^t - \mathrm{clip}(C_h^T Z_h^t;\tau^t)\rangle \quad + 2\eta^{-1}\langle \overline{Z_h^{t+1}}, \overline{Z_h^{t+1}} - \overline{Z_h^t}\rangle \\ &\quad + 2(1-\eta\mu_{\max}(L_h))\|C_h^T Z_h^t\|_1\Delta_\infty\tau^t + 2\eta\mu_{\max}(L_h)\|C_h^T Z_h^t\|_1\Delta_\infty\tau^t \\ &\quad - 2\eta\mu_{\max}(L_h)\|\mathrm{clip}(C_h^T Z_h^t;\tau^t)\|_1\Delta_\infty\tau^t \\ &\leq \eta\mu_{\max}(L_h)\left(\|\mathrm{clip}(C_h^T Z_h^t;\tau^t)\|^2 + \|\tau^t C_h^\dagger C_b U_b^t\|^2 - 2\|\mathrm{clip}(C_h^T Z_h^t;\tau^t)\|_1\Delta_\infty\tau^t\right) \\ &\quad + 2\langle C_h^T Z_h^t, C_h^T Z_h^t - \mathrm{clip}(C_h^T Z_h^t;\tau^t)\rangle + 2\|C_h^T Z_h^t\|_1\Delta_\infty\tau^t \\ &\quad + 2\eta^{-1}\langle \overline{Z_h^{t+1}}, \overline{Z_h^{t+1}} - \overline{Z_h^t}\rangle.\end{aligned}$$

Then, the second term is controlled using the proof of Lemma 5.B.5.

By denoting

$$\Delta_2^2 := \sup_{\|U_b^t\|_{\infty,2}\leq 1} \|C_h^\dagger C_b U_b^t\|_2^2,$$

we have that, under Definition 5.3.2

$$\begin{aligned}\zeta^t &\leq \eta\mu_{\max}(L_h)\left(\|C_h^T Z_h^t\|_{2,-\kappa^t}^2 + \tau^t\left(\tau^t\kappa^t + \tau^t\Delta_2^2 - 2\tau^t\kappa^t\Delta_\infty - 2\Delta_\infty\|C_h^T Z_h^t\|_{1,-\kappa^t}\right)\right)\\ &\quad + 2\|C_h^T Z_h^t\|_{2,\kappa^t}^2\\ &\leq \eta\mu_{\max}(L_h)\left((1-\Delta_\infty)\|C_h^T Z_h^t\|_{2,-\kappa^t}^2 + (\tau^t)^2\left[\kappa^t(1-2\Delta_\infty) + \Delta_2^2\right]\right) + 2\|C_h^T Z_h^t\|_{2,\kappa^t}^2.\end{aligned}$$

Eventually, leveraging that $\Delta_2^2 \leq \Delta_\infty^2|\mathcal{E}_h|$, we have

$$\zeta^t \leq \eta\mu_{\max}(L_h)\left(\|C_h^T Z_h^t\|_{2,-\kappa^t}^2 + (\tau^t)^2\left[\kappa^t(1-2\Delta_\infty) + \Delta_\infty^2|\mathcal{E}_h|\right]\right) + 2\|C_h^T Z_h^t\|_{2,\kappa^t}^2.$$

□

Theorem 5.B.7. *Hence, under Definition 5.3.2, for*

$$\eta \leq \frac{\mu_{\max}(L_h)^{-1}}{1+|\mathcal{E}_h|(1-\Delta_\infty)},$$

we have

$$\|Z_h^{t+1}\|^2 \leq \|Z_h^t\|^2 - \eta\|C_h^T Z_h^t\|_{2,-\kappa^t}^2.$$

Proof. We start from Lemma 5.B.3 and use the majoration of the error from Lemma 5.B.6:

$$\begin{aligned}\|Z_h^{t+1}\|^2 &= \|Z_h^t\|^2 - 2\eta\|C_h^T Z_h^t\|^2 + \eta^2\|L_h Z_h^t - E^t\|^2 + 2\eta\langle Z_h^t - \overline{Z_h^t}, E^t\rangle + 2\langle\overline{Z_h^t}, \overline{Z_h^{t+1}} - \overline{Z_h^t}\rangle\\ &\leq \|Z_h^t\| - 2\eta\|C_h^T Z_h^t\|_2^2 + +2\eta\|C_h^T Z_h^t\|_{2,\kappa^t}^2\\ &\quad + \eta^2\mu_{\max}(L_h)\left(\|C_h^T Z_h^t\|_{2,-\kappa^t}^2 + (\tau^t)^2\left[\kappa^t(1-2\Delta_\infty) + \Delta_\infty^2|\mathcal{E}_h|\right]\right)\\ &= \|Z_h^t\| - 2\eta\|C_h^T Z_h^t\|_{2,-\kappa^t}^2\\ &\quad + \eta^2\mu_{\max}(L_h)\left(\|C_h^T Z_h^t\|_{2,-\kappa^t}^2 + (\tau^t)^2\left[\kappa^t(1-2\Delta_\infty) + \Delta_\infty^2|\mathcal{E}_h|\right]\right)\\ &\leq \|Z_h^t\| - \eta\left[2 - \eta\mu_{\max}(L_h)\left(1 + \kappa^t(1-2\Delta_\infty) + \Delta_\infty^2|\mathcal{E}_h|\right)\right]\|C_h^T Z_h^t\|_{2,-\kappa^t}^2.\end{aligned}$$

Where we used that, as $\kappa^t \leq \sum_{(i\sim j)\in\mathcal{E}_h} 1_{\|z_i^t - z_j^t\|\geq\tau^t} - 1$, the largest term in $\|C_h^T Z_h^t\|_{2,-\kappa^t}^2$ is $(\tau^t)^2$.

Hence, for

$$\eta \leq \frac{\mu_{\max}(L_h)^{-1}}{1+\kappa^t(1-2\Delta_\infty) + \Delta_\infty^2|\mathcal{E}_h|},$$

or, using a simpler (and stricter) condition

$$\eta \leq \frac{\mu_{\max}(L_h)^{-1}}{1+|\mathcal{E}_h|(1-\Delta_\infty)},$$

we have

$$\|Z_h^{t+1}\|^2 \leq \|Z_h^t\|^2 - \eta\|C_h^T Z_h^t\|_{2,-\kappa^t}^2.$$

□

5.B.2.1 On the computation of Δ_∞

We recall that

$$\Delta_\infty := \sup_{\substack{U_b \in \mathbb{R}^{\mathcal{E}_b \times d} \\ \|U_b\|_{\infty,2} \leq 1}} \|C_h^\dagger C_b U_b\|_{\infty,2}.$$

Proposition 5.B.8. ***(Computing Δ_∞)*** *Consider a fully connected graph $\mathcal{G}$, we have that*

$$\Delta_\infty \leq \sup_{(i\sim j)\in\mathcal{E}_h} \frac{2n_b}{n_h}\sqrt{d \wedge |\mathcal{E}_b|},$$

where n_b is the number of Byzantines nodes and n_h the number of honest nodes.

Proof. In a fully connected graph $L_h = n_h I_h - \mathbf{1}_h \mathbf{1}_h^T$, thus $C_h^\dagger = C_h^T L_h^\dagger = \frac{1}{n_h} C_h^T$. Using this, we compute:

$$\begin{aligned}\Delta_\infty &= \sup_{\substack{U_b \in \mathbb{R}^{\mathcal{E}_b \times d} \\ \|U_b\|_{\infty,2} \leq 1}} \|C_h^\dagger C_b U_b\|_{\infty,2} \\ &= \frac{1}{n_h} \sup_{\substack{U_b \in \mathbb{R}^{\mathcal{E}_b \times d} \\ \|U_b\|_{\infty,2} \leq 1}} \sup_{(i\sim j)\in\mathcal{E}_h} \|[C_h^T C_b U_b]_{(i\sim j),:}\|_2.\end{aligned}$$

For $(i \sim j) \in \mathcal{E}_h$,

$$\begin{aligned}\sup_{\substack{U_b \in \mathbb{R}^{\mathcal{E}_b \times d} \\ \|U_b\|_{\infty,2} \leq 1}} \|[C_h^T C_b U_b]_{(i\sim j),:}\|_2^2 &= \sup_{\substack{U_b \in \mathbb{R}^{\mathcal{E}_b \times d} \\ \|U_b\|_{\infty,2} \leq 1}} \sum_{k=1}^d \left([C_h^T C_b]_{(i\sim j),:}^T [U_b]_{:,k}\right)^2 \\ &= \sup_{\substack{U_b \in \mathbb{R}^{\mathcal{E}_b \times d} \\ \|U_b\|_{\infty,2} \leq 1}} \sum_{k=1}^d \left([[C_b]_{i,:} - [C_b]_{j,:}]^T [U_b]_{:,k}\right)^2 \\ &\leq \sup_{\substack{U_b \in \mathbb{R}^{\mathcal{E}_b \times d} \\ \|U_b\|_{\infty,2} \leq 1}} \sum_{k=1}^d \|[[C_b]_{i,:} - [C_b]_{j,:}]\|_1^2 \|[U_b]_{:,k}\|_\infty^2 \\ &\leq (N_b(i) + N_b(j))^2 \sup_{\substack{U_b \in \mathbb{R}^{\mathcal{E}_b \times d} \\ \|U_b\|_{\infty,2} \leq 1}} \sum_{k=1}^d \|[U_b]_{:,k}\|_\infty^2 \\ &\leq (N_b(i) + N_b(j))^2 (d \wedge |\mathcal{E}_b|).\end{aligned}$$

Hence, we get:

$$\Delta_\infty \leq \sup_{(i\sim j)\in\mathcal{E}_h} \frac{N_b(i) + N_b(j)}{n_h}\sqrt{d \wedge |\mathcal{E}_b|}.$$

Thus, assuming that, as we are in a fully-connected graph, every honest node is connected to n_b Byzantine node, we get the result. □

5.B.3 A simplified global clipping rule

The following proposition provides a sufficient condition for being a GCR that only requires κ_t to be large enough, without any dependence on X^t.

Proposition 5.B.9 (Simplified GCR). *If $(\tau^t)_{t\geq 0}$ is such that: for all $t \geq 0$, either $\tau^t = 0$ or*

$$\kappa^t \geq \Delta_\infty |\mathcal{E}_h| + \eta \frac{|\mathcal{E}_b|^2}{n_h} \sum_{s\leq t} \frac{\tau^s}{\tau^t}, \tag{5.15}$$

then $(\tau^s)_{s\geq 0}$ satisfies the GCR.

This result reads as a lower bound on the fraction $\kappa^t/|\mathcal{E}_h|$ of edges that are clipped.

Proof. (GCR), i.e Equation (5.13) writes

$$\|C_h^T Z_h^t\|_{1,\kappa^t} \geq \Delta_\infty \|C_h^T Z_h^t\|_1 + \frac{\eta|\mathcal{E}_b|^2}{n_h} \sum_{s=0}^{t} \tau^s.$$

We leverage that $\|C_h^T Z_h^t\|_{1,\kappa^t} \geq \tau^t \kappa^t$ and that $\|C_h^T Z_h^t\|_{1,-\kappa^t} \leq \tau^t(|\mathcal{E}_h| - \kappa^t)$ to write

$$\begin{aligned}
&\|C_h^T Z_h^t\|_{1,\kappa^t} \geq \Delta_\infty \|C_h^T Z_h^t\|_1 + \frac{\eta|\mathcal{E}_b|^2}{n_h} \sum_{s=0}^{t} \tau^s \\
&\iff (1-\Delta_\infty)\|C_h^T Z_h^t\|_{1,\kappa^t} \geq \Delta_\infty \|C_h^T Z_h^t\|_{1,-\kappa^t} + \frac{\eta|\mathcal{E}_b|^2}{n_h} \sum_{s=0}^{t} \tau^s \\
&\impliedby (1-\Delta_\infty)\kappa^t\tau^t \geq \Delta_\infty \tau^t(|\mathcal{E}_h| - \kappa^t) + \frac{\eta|\mathcal{E}_b|^2}{n_h} \sum_{s=0}^{t} \tau^s \\
&\iff \kappa^t\tau^t \geq \Delta_\infty \tau^t |\mathcal{E}_h| + \frac{\eta|\mathcal{E}_b|^2}{n_h} \sum_{s=0}^{t} \tau^s \\
&\iff \kappa^t \geq \Delta_\infty |\mathcal{E}_h| + \frac{\eta|\mathcal{E}_b|^2}{n_h} \sum_{s=0}^{t} \frac{\tau^s}{\tau^t}.
\end{aligned}$$

□

Part III

Byzantine Resilience in Federated Optimization

Chapter 6

From Inexact Gradients to Byzantine Robustness: Acceleration and Optimization under Similarity

Distributed learning algorithms are vulnerable to adversarial nodes, a.k.a. Byzantine failures. To solve this issue, robust algorithms have been developed, which typically replace parameter averaging by robust aggregations. While generic conditions on these aggregations exist to guarantee the convergence of (Stochastic) Gradient Descent (SGD), the analyses remain rather ad-hoc. This hinders the development of more complex robust algorithms, such as accelerated ones. In this work, we show that Byzantine-robust distributed optimization can, under standard generic assumptions, be cast as a general optimization with inexact gradient oracles, an active field of research. This allows to obtain state-of-the-art results for Byzantine-robust optimization from general inexact first-order analyses.

We first show that inexact GD on top of standard robust aggregation procedures obtains optimal asymptotic error in the Byzantine setting. Going further, we study an algorithm for *Optimization under Similarity*, in which the server leverages an auxiliary loss function that approximates the global loss. Then, we introduce an *Accelerated Extra-gradient* method, that yields acceleration in both the standard and similarity settings. We first give new general convergence results for these inexact schemes, and then instantiate these results in the Byzantine setting through our reduction. Both algorithms drastically reduce the communication complexity compared to previous methods, as we show theoretically and empirically.

Contents

6.1 Introduction . . . 117
6.2 Setting and Reduction . . . 119
6.2.1 Byzantine Robust Distributed Optimization . . . 119
6.2.2 Reduction to an Inexact Gradient Problem . . . 120
6.2.3 First Consequences of the Inexact Oracle Approach . . . 121
6.3 Gradient Descent under Similarity . . . 123
6.3.1 Setting . . . 123
6.3.2 Convergence Results . . . 124
6.4 An Accelerated Inexact Extra-gradient Method . . . 126
6.4.1 General Inexact Algorithm . . . 126
6.4.2 Acceleration Without Similarity . . . 127
6.4.3 Focus on the Byzantine Setting . . . 127
6.5 Experiments . . . 128
6.5.1 Setup . . . 128
6.5.2 Analysis . . . 129
6.6 Conclusion . . . 129

6.A Related Concurrent Work Shi et al. (2025) . . . 131
6.B Robust Aggregation Rules . . . 131
6.C From Hessian Similarity to Bounded Heterogeneity . . . 136
6.D Convergence of Algorithm 4 and Algorithm 5 . . . 140
6.E From Devolder et al. (2013) acceleration to acceleration for $(\zeta^2, \alpha = 0)$-inexact oracles . . . 150
6.F Additional Experiments . . . 152
6.G Additional Discussion . . . 154

6.1 Introduction

Distributed Learning is a paradigm in which multiple computers collaborate to train a machine learning model. This allows us to simultaneously leverage the computational power of multiple devices and to adapt to the distributed nature of data, which is often stored on smartphones, IOT devices, or by larger stakeholders such as companies or hospitals. However, involving multiple parties in an algorithmic process opens up to new threats: machines can crash, software can be buggy or even hacked, data can be corrupted or even adversarially crafted. Byzantine failure (Lamport et al., 1982) is a worst-case model of such behaviors, corresponding to omniscient adversarial parties.

Distributed Gradient Descent (DGD), the workhorse of modern distributed learning, is vulnerable to such failures (Blanchard et al., 2017): when the clients' gradients are averaged, one Byzantine node returning adversarially crafted gradients can corrupt the result arbitrarily, thus ruining the optimization. To address this critical issue, a large body of research has been devoted to developing and studying distributed learning methods that converge even with adversaries (Blanchard et al., 2017; Yin et al., 2018; Chen et al., 2017). Those methods crucially replace the gradients (or local parameters) averaging step with a robust aggregation scheme, such as the geometric median or coordinate-wise trimming. Over the years, analyses specific to each robust aggregation method have given way to generic characterizations and standard study assumptions (Allouah et al., 2023a; Karimireddy et al., 2023). Typically, one simply needs to guarantee that the error (*i.e.*, distance from the robust aggregation result to the average of the honest nodes) remains bounded. This bound depends on both the heterogeneity between honest nodes and the proportion of Byzantine nodes. By controlling these two sources of error, one can determine the optimal asymptotic error, and show that it can be achieved by several robust aggregation schemes (Farhadkhani et al., 2022; Karimireddy et al., 2021). Momentum-based methods were then proposed to alleviate the stochastic error in the Byzantine context (Karimireddy et al., 2021; Farhadkhani et al., 2022). Similarly, refined results taking precisely the dimension into account have been reached by leveraging high-dimensional averaging tools (Diakonikolas et al., 2019; Zhu et al., 2023). Robust distributed learning algorithms were also coupled with gradient compression (Rammal et al., 2024; Malinovsky et al., 2024; Liu et al., 2024) to reduce their communication cost.

As mentioned above, Byzantine-resilient algorithms work for a wide variety of robust aggregation methods, as long as they satisfy relevant error conditions. Yet, analyses of robust algorithms are still tailored to the Byzantine setting: proofs closely follow existing schemes, but are adapted to match the specificity of the distributed adversarial setting. Although this adaptation is rather transparent in simple settings, it becomes much more cumbersome for more involved proofs, with less leeway to accumulate error terms. This hinders the development of efficient algorithms, leaving critical questions open, and preventing the adaptation of efficient federated algorithms. For instance, Ghosh et al. (2020) proposed second-order methods, but require a very low heterogeneity level to actually converge. Similarly, Xu et al. (2025) studied Nesterov's acceleration in a relatively homogeneous setting, and do not actually obtain the expected accelerated convergence rates.

Recently, Shi et al. (2025) proposed an involved method and claimed to reach an accelerated rate, which was actually not true, as we detail in Section 6.A.

Contributions and outline

Our work alleviates the aforementioned shortcomings of existing theory. We cast Byzantine-robust optimization as a special case of optimization under inexact (gradient) oracles, and build on this to derive efficient robust algorithms with provably low iteration (and so communication) complexity.

Generic Formalism. After reviewing the setting in Section 6.2.1, we show in Section 6.2.2 that Byzantine-robust distributed optimization exactly fits into the formalism of first-order methods with inexact gradient oracles. To do so, we show that combining (G, B)-heterogeneity (Karimireddy et al., 2020), a standard assumptions on the heterogeneity of loss functions, with (f, ν)-robustness (Allouah et al., 2023a), a standard characterization of the employed robust aggregation method, leads to simple additive and multiplicative error terms on the mean of the honest gradient. Optimization under the provided inexact gradient formulation is an active field of research, from which we can thus directly reuse many results.

We then show in Section 6.2.3 that this abstraction is tight in the sense that using existing inexact GD results allows us to match the Byzantine-robust optimization lower bound on the asymptotic error under heterogeneity. We also leverage existing work on inexact gradients to show a first accelerated algorithm, but only under a stronger heterogeneity condition, and with sub-optimal asymptotic error.

These first results both motivate and pave the way for a more systematic search for efficient algorithms, abstracting away the complexity of the Byzantine setting, and thus building on the inexact gradient formulation to propose a robust version of two standard fast algorithms.

Optimization under similarity. When an approximation (in the second-order sense) of the global honest loss is available, we give in Section 6.3 a Byzantine-robust algorithm (with computationally more expensive updates than GD) called Prox Inexact Gradient method under Similarity (PIGS). PIGS enjoys a linear convergence rate of $\mathcal{O}(\Delta/\mu \log(1/\varepsilon))$ to reach an ε neighborhood of the optimal error, where Δ is a measure of the distance between the Hessian of the approximate loss and the true honest one. Since Δ can be drastically smaller than the smoothness of the loss function, this results in an important gain in iteration complexity, which consequently reduces the communication time. The convergence theory of PIGS requires an extension of Woodworth et al. (2023) to handle the inexactness of the error term.

Acceleration. In Section 6.4, we adapt the extra-gradient method from (Kovalev et al., 2022) to the inexact oracle setting. We show that it converges in this setting towards the *optimal asymptotic error* with an *accelerated linear rate* of convergence $\mathcal{O}(\sqrt{\mu/L})$, where μ is the strong convexity of the considered loss function and L the smoothness. This result holds as long as the relative error in the oracle is smaller than $\sqrt{\mu/L}$. Our result outperforms

previous works on accelerated inexact oracle first-order methods by a $\sqrt{\mu/L}$ factor both in the asymptotic error and in the condition on the relative error.

The extra-gradient algorithm also enjoys an accelerated rate of convergence in the similarity setting of Section 6.3, but only as long as the relative error in the gradient is small with respect to Δ^2/L^2.

Experiments. Last, we empirically show the robustness and performance of the proposed methods by comparing them in the Byzantine setting to existing methods in Section 6.5.

Notations. We denote $\langle \cdot, \cdot \rangle$ the standard dot product in $\mathbb{R}^d$, $\| \cdot \|$ the euclidean norm, and $\| \cdot \|_{op}$ the associated operator norm in $\mathbb{R}^{d \times d}$. For a real valued function $f : \mathbb{R}^d \to \mathbb{R}$, ∇f denotes its gradient, and $\nabla^2 f$ its Hessian. Let $f(\boldsymbol{x}) = \Omega(g(\boldsymbol{x}))$ and $g(\boldsymbol{x}) = \mathcal{O}(f(\boldsymbol{x}))$ denote that $\exists M > 0$ such that $f(\boldsymbol{x}) \geq M g(\boldsymbol{x})$. We denote $f(\boldsymbol{x}) = \Theta(g(\boldsymbol{x}))$ when both $f(\boldsymbol{x}) = \mathcal{O}(g(\boldsymbol{x}))$ and $f(\boldsymbol{x}) = \Omega(g(\boldsymbol{x}))$ hold. For $\boldsymbol{x} \in \mathbb{R}^d$, $\|\boldsymbol{x}\|$ denotes the canonical euclidean norm, and for $\boldsymbol{M} \in \mathbb{R}^{d \times d}$, $\|\boldsymbol{M}\|_{op}$ the associated operator norm.

6.2 Setting and Reduction

6.2.1 Byzantine Robust Distributed Optimization

We consider a server-client distributed system, with n clients. Among them, an unknown subset $\mathcal{B} \subset [n]$ is composed of $|\mathcal{B}| = f$ Byzantine clients. We denote $\mathcal{H} = [n] \backslash \mathcal{B}$ the $n - f$ honest clients, and $\mathcal{L}_i : \mathbb{R}^d \to \mathbb{R}$ the loss function of client i (*e.g.*, the empirical loss on the dataset stored by client i). We aim at finding

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} \left\{ \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}) := \frac{1}{n-f} \sum_{i \in \mathcal{H}} \mathcal{L}_i(\boldsymbol{x}) \right\}. \tag{6.1}$$

We assume that all $\mathcal{L}_i$ are differentiable, convex, and that $\mathcal{L}_{\mathcal{H}}$ is μ-strongly convex and L-smooth.

Definition 6.2.1. A function $\mathcal{L}$ is *μ-strongly convex*, and *L-smooth* if, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, $\frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 \leq \mathcal{L}(\boldsymbol{x}) - \mathcal{L}(\boldsymbol{y}) - \langle \nabla \mathcal{L}(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle \leq \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2$. $\mathcal{L}$ is *convex* if the left inequality holds with $\mu = 0$. The ratio $\kappa = L/\mu$ is the *condition number* of $\mathcal{L}$.

The goal of (distributed) optimization is to output a model $\hat{\boldsymbol{x}}$ with an error that vanishes with the computing budget. We say that an algorithm is resilient if it can reach a given error level ε despite the presence of adversaries, namely that the output $\hat{\boldsymbol{x}}$ satisfies $\mathcal{L}_{\mathcal{H}}(\hat{\boldsymbol{x}}) - \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^*) \leq \varepsilon$.

Hardness of Byzantine Optimization. Byzantine clients cannot be distinguished from honest clients with outlying losses. To obtain bounded errors, it is therefore necessary to assume that honest node losses have some level of similarity.

Assumption 6.2.2 ((G,B)-heterogeneity)**.** The local loss functions of honest clients are said to satisfy (G, B)-heterogeneity when, for any $\boldsymbol{x} \in \mathbb{R}^d$,

$$\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\nabla\mathcal{L}_i(\boldsymbol{x}) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\|^2 \leq G^2 + B^2\|\nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\|^2.$$

One can then derive the minimum error that any distributed algorithm has to pay, which increases with both the heterogeneity among honest clients and the proportion of Byzantine clients, up to a critical fraction known as the *breakdown point*, at which adversaries can arbitrarily disturb the system.

Theorem 6.2.3 (Allouah et al. (2024))**.** *For any distributed algorithm, there exist quadratic local loss functions satisfying Assumption 6.2.2, with an L-smooth and μ-strongly convex global loss $\mathcal{L}_{\mathcal{H}}$, for which algorithm's output $\hat{\boldsymbol{x}}$ can have a non-vacuous guarantee only if $\frac{f}{n} \leq \frac{1}{B^2+2}$, and such that*

$$\mathcal{L}_{\mathcal{H}}(\hat{\boldsymbol{x}}) - \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^*) \geq \frac{G^2}{8\mu}\frac{f}{n-(2+B^2)f}, \quad \textit{and} \quad \|\nabla\mathcal{L}_{\mathcal{H}}(\hat{\boldsymbol{x}})\|^2 \geq \frac{G^2}{4}\frac{f}{n-(2+B^2)f}. \tag{6.2}$$

The inequality $\frac{f}{n} \leq \frac{1}{B^2+2}$, gives an upper bound on the achievable breakdown point.

6.2.2 Reduction to an Inexact Gradient Problem

To protect against adversaries, distributed optimization methods rely on Algorithm 3: they robustly estimate the average of the honest gradients using an aggregator F, which is, for instance, a geometric median (Small, 1990; Pillutla et al., 2022), a coordinate-wise trimmed mean (Yin et al., 2018), or other methods such as Krum (Blanchard et al., 2017). Allouah et al. (2023a) proposed (f, ν)-robust, a criterion to simultaneously analyze those robust averaging schemes.

Definition 6.2.4 ((f,ν)-robustness (Allouah et al., 2023a))**.** An aggregation rule $F : \mathbb{R}^{d\times n} \to \mathbb{R}^d$ is said to be (f, ν)-robust, if $\forall \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$, and any set $S \subset [n]$ of size $n - f$,

$$\|F(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) - \overline{\boldsymbol{x}}_S\|^2 \leq \nu \cdot \frac{1}{|S|}\sum_{i\in S}\|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_S\|^2,$$

where $\overline{\boldsymbol{x}}_S = |S|^{-1}\sum_{i\in S}\boldsymbol{x}_i$. We call ν[a] the *robustness coefficient* of F.

[a] We denote it ν instead of κ as in Allouah et al. (2023a) due to notation incompatibility with the standard notation of the condition number.

Closed-form robustness coefficients can be found in Table 6.1, the Huber estimator cited is a side contribution (c.f. Section 6.B.1), and can be seen as a novel clipping threshold and analysis of CenteredClipping (Karimireddy et al., 2021). We discuss robustness aggregation techniques in Section 6.B, where we also generalize the mixing strategy of Allouah et al.

Algorithm 3 Inexact Oracles Sampler

Input: F, $\boldsymbol{x}$
Server sends $\boldsymbol{x}$ to clients
for client $i \in [n]$ in parallel **do**
  Computes $\boldsymbol{g}_i = \begin{cases} \nabla \mathcal{L}_i(\boldsymbol{x}) \text{ if } i \in \mathcal{H} \\ * \text{ if } i \in \mathcal{B} \end{cases}$
  Send $\boldsymbol{g}_i$ to Server
end for
Server computes $\tilde{\nabla}\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}) := F(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n)$

Table 6.1: Robustness coefficients of the coordinate-wise trimmed mean (CWTM), Krum, the geometric median (GM) (Allouah et al., 2023a), and a multivariate Huber estimator Algorithm 6.

Aggregation	Huber	CWTM	Krum	GM	Lower Bound
ν	$\frac{2f}{n-3f}$	$\frac{6f}{n-2f}\left(1+\frac{6f}{n-2f}\right)$	$6\left(1+\frac{6f}{n-2f}\right)$	$4\left(1+\frac{f}{n-2f}\right)^2$	$\frac{f}{n-2f}$

(2023a). The strength of Definition 6.2.4 is that under Assumption 6.2.2, aggregating gradients using (f, ν)-robust rules produces an inexact gradient oracle.

Definition 6.2.5 ((ζ^2, α)-inexact oracle)**.** A (ζ^2, α)-inexact gradient oracle of differentiable function $\mathcal{L}$, is an oracle $\tilde{\nabla}\mathcal{L}$ satisfying, for any $\boldsymbol{x} \in \mathbb{R}^d$, $\|\tilde{\nabla}\mathcal{L}(\boldsymbol{x}) - \nabla\mathcal{L}(\boldsymbol{x})\|^2 \leq \zeta^2 + \alpha\|\nabla\mathcal{L}(\boldsymbol{x})\|^2$.

Lemma 6.2.6. *If F is (f, ν)-robust and local losses are (G, B)-heterogeneous then $\tilde{\nabla}\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})$, the output of Algorithm 3 is a $(\nu G^2, \nu B^2)$-inexact gradient oracle.*

This reduction is of interest since instances of (ζ^2, α)-inexact oracle have been widely investigated (Ajalloeian and Stich, 2020), while mostly with either $\alpha = 0$ or $\zeta^2 = 0$ (De Klerk et al., 2020; Cyrus et al., 2018; Vasin et al., 2023). We can then reuse these algorithms to obtain Byzantine-robust methods. Moreover, developing algorithms based on the (ζ^2, α) inexact gradient assumption opens up to generic contributions, of interest beyond the Byzantine setting.

6.2.3 First Consequences of the Inexact Oracle Approach

We now leverage existing results of the literature with the inexact oracle perspective.

6.2.3.1 Gradient Descent

A first remarkable result is the **tightness of our reduction**: plugging Lemma 6.2.6 into a black-box inexact oracle result (Ajalloeian and Stich, 2020) leads to Corollary 6.2.7, which achieves the best possible asymptotic error.

Corollary 6.2.7 ((Ajalloeian and Stich, 2020)). *Let, $\mathcal{L}_{\mathcal{H}}$ be μ-strongly convex and L-smooth. Consider the Gradient Descent update $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta\tilde{\nabla}\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_k)$, where $\tilde{\nabla}\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_k)$ stems from Algorithm 3 with a (f,ν)-robust aggregation rule. Then, for $\eta \leq 1/L$ and $\nu B^2 < 1$, it holds*

$$\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_K) \leq \big(1 - \eta\mu\big(1-\nu B^2\big)\big)^K \big(\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_0) - \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^*)\big) + \frac{\nu G^2}{\mu(1-\nu B^2)}.$$

To back the tightness claim, consider a (f,ν)-robust aggregator with $\nu = \frac{cf}{n-2f}$, such as the Huber estimator ($c = 3$ for $n > 4f$, cf. Table 6.1). Then, for $\frac{f}{n} \leq \frac{1}{2(1+cB^2)}$, Algorithm 3 used in GD converges to the optimal error of Theorem 6.2.3 up to a multiplicative constant, i.e. $O(\frac{f}{n-2f}\frac{G^2}{\mu})$. This result is somehow surprising, as one may think that there would be more structure to the Byzantine problem than simply (f,ν)-robustness, for instance by requiring to observe the identities of the clients. Note that this latter point is mandatory for reaching an optimal error when the clients send stochastic i.i.d. gradient oracles (cf.Section 6.G and (Karimireddy et al., 2021)).

Tightness with respect to the dimension. The definition of (f,ν)-robust aggregator obfuscates the dependency on the dimension in the resulting error term $Gf/\mu(n-2f)$, even though a dimension-independent error can be achieved (Diakonikolas et al., 2019; Zhu et al., 2023; Allouah et al., 2025b). The Inexact Oracle Approach also tightly encompasses averaging rules with dimension-independent error, as we show in Section 6.C.2 but omit here for clarity.

A new perspective on the optimal error. We can reinterpret the lower bound in Theorem 6.2.3 as describing the region of $\boldsymbol{x} \in \mathbb{R}^d$ on which no information can be leveraged by the server. Indeed, 6.2 gives the gradient norm value below which a $(\zeta^2 = \nu G^2, \alpha = \nu B^2)$-inexact oracle resulting from Algorithm 3 may not be informative, as if $\|\nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\|^2 \leq \frac{G^2\nu}{1-B^2\nu}$, then the value $\tilde{\nabla}\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}) = 0$ is a valid $(\zeta^2 = \nu G^2, \alpha = \nu B^2)$-inexact oracle, while being obviously un-informative.

6.2.3.2 Fast Gradient Method

There are numerous works on accelerated first-order methods under inexact gradient oracles (d'Aspremont, 2008; Devolder et al., 2013, 2014). Building on Devolder et al. (2013), we give a Byzantine-resilient accelerated gradient method.

Corollary 6.2.8 (Fast Gradient Method w. Inexact Oracles (Devolder et al., 2014)). *Assume $\mathcal{L}_{\mathcal{H}}$ is L-smooth and μ-strongly convex, and Assumption 6.2.2 holds with $B = 0$. Then Algorithm 9 with a $(f, \nu = O(f/n-2f))$-robust aggregator yields*

$$\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_K) - \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^*) = O\left(L\,\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 \exp\left(-\frac{K}{4\sqrt{\kappa}}\right) + \sqrt{\kappa}\,\frac{G^2}{\mu}\,\frac{f}{n-2f}\right).$$

An accelerated rate. The above result is appealing in the sense that it shows an *accelerated rate* towards the asymptotic error, i.e the iteration complexity to reach a given

error scales as the square root of the condition number $\kappa = L/\mu$, rather than linearly. To our knowledge, this is the first correct proof of such an acceleration in the Byzantine setting (cf. Section 6.A), demonstrating the strength of the abstraction.

Limits. This result is a consequence of a notion of (δ, L, μ)- oracles (Devolder et al., 2013, 2014), which covers a broad class of inexact oracles. Yet, it has two main flaws: it covers *only an absolute error* term in the gradient, since $B = 0$, and the *asymptotic error is suboptimal*, inflated by $\sqrt{\kappa}$ factor w.r.t. the lower bound (6.2.3). This significant factor is unavoidable when using (δ, L, μ)-oracles (Devolder et al., 2014), but can be avoided when using (ζ^2, α)-inexact oracles, as shown in Section 6.4. We detail the link between (δ, L, μ)-inexact oracles and (ζ^2, α)-inexact oracles in Section 6.E.

6.3 Gradient Descent under Similarity

We now focus on another approach to reduce the communication complexity, which takes advantage of the similarity between local objectives (Shamir et al., 2014; Hendrikx et al., 2020b; Kovalev et al., 2022).

6.3.1 Setting

The key assumption is that the server has cheap (without communication) access to a proxy $\hat{\mathcal{L}}$ of the global loss function $\mathcal{L}_{\mathcal{H}}$ (which requires communication). This proxy may come from the empirical risk computed on a small toy dataset, or from a trusted client that uses its own loss function as the proxy, i.e., $\hat{\mathcal{L}} = \mathcal{L}_1$. In particular, the server can solve the following problem at each iteration:

$$\boldsymbol{x}_{k+1} \approx \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \left\{ \overbrace{\hat{\mathcal{L}}(\boldsymbol{x})}^{\text{Proxy Loss}} + \langle \nabla \overbrace{(\mathcal{L}_{\mathcal{H}} - \hat{\mathcal{L}})}^{\text{small Hessian}}(\boldsymbol{x}_k), \boldsymbol{x}\rangle + \frac{1}{2\eta}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2 \right\}. \tag{6.3}$$

This update can be seen from two perspectives, either as a mirror descent step on $\mathcal{L}_{\mathcal{H}}$ with step-size 1 and mirror map $\hat{\mathcal{L}} + \frac{1}{2\eta}\|\cdot\|^2$ (Shamir et al., 2014; Hendrikx et al., 2020b), or as a prox-gradient update with the forward step on $\mathcal{L}_{\mathcal{H}} - \hat{\mathcal{L}}$ (smooth) and the backward step on $\hat{\mathcal{L}}$ (cheap prox) (Kovalev et al., 2020; Woodworth et al., 2023). In both cases, these manipulations make the problem smoother when $\mathcal{L}_{\mathcal{H}}$ and $\hat{\mathcal{L}}$ are "similar" enough, speeding up the optimization by allowing significantly larger step sizes.

Assumption 6.3.1 (Similarity)**.** The difference between the global loss and the proxy $\mathcal{L}_{\mathcal{H}} - \hat{\mathcal{L}}$ is Δ-smooth.

Our result holds under the above assumption, as long as the inexact oracles satisfies an alternative definition of inexact oracles.

Definition 6.3.2 ($(\zeta^2, \tilde{\alpha})$-inexact oracle)**.** A $(\zeta^2, \tilde{\alpha})$-inexact gradient oracle of a μ-strongly convex function $\mathcal{L}$, is an oracle $\tilde{\nabla}\mathcal{L}$ satisfying, for $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x})$ and any $\boldsymbol{x} \in \mathbb{R}^d$,

$$\|\tilde{\nabla}\mathcal{L}(\boldsymbol{x}) - \nabla\mathcal{L}(\boldsymbol{x})\|^2 \leq \zeta^2 + \tilde{\alpha}\mu^2\|\boldsymbol{x} - \boldsymbol{x}^*\|^2.$$

Remark 6.3.3. Since $\mu^2\|\boldsymbol{x}-\boldsymbol{x}^*\|^2 \leq \|\nabla\mathcal{L}(\boldsymbol{x})\|^2$, any $(\zeta^2,\tilde{\alpha})$-inexact oracle is a (ζ^2,α)-inexact oracle with $\alpha=\tilde{\alpha}$.

When the clients satisfy the pairwise Hessian Similarity assumption, on the one hand Definition 6.3.2 arises from using as proxy a trusted client's loss, $\hat{\mathcal{L}}=\mathcal{L}_1$, and on the other hand using Algorithm 3 with an (f,ν)-aggregator yields a $(\zeta^2,\tilde{\alpha})$-inexact oracle 6.3.2.

Assumption 6.3.4 (Pairwise Hessian Similarity)**.** All local honest loss functions $(\mathcal{L}_i)_{i\in\mathcal{H}}$ are twice differentiable, and satisfy $\forall i,j\in\mathcal{H},\forall\boldsymbol{x}\in\mathbb{R}^d,\quad \|\nabla^2\mathcal{L}_i(\boldsymbol{x})-\nabla^2\mathcal{L}_j(\boldsymbol{x})\|_{op}\leq\Delta$.

Proposition 6.3.5. *Let $(\mathcal{L}_i)_{i\in\mathcal{H}}$ be convex functions. Then, Assumption 6.3.4 ensures that*

$$\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\nabla\mathcal{L}_i(\boldsymbol{x})-\nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\|^2\leq\frac{2}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\nabla\mathcal{L}_i(\boldsymbol{x}^*)\|^2+2\Delta^2\|\boldsymbol{x}-\boldsymbol{x}^*\|^2. \tag{6.4}$$

Which, if additionally $\mathcal{L}_{\mathcal{H}}$ is μ-strongly convex, implies (G,B)-Heterogeneity with

$$G^2=\frac{2}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\nabla\mathcal{L}_i(\boldsymbol{x}^*)\|^2 \text{ and } B^2=\frac{2\Delta^2}{\mu^2}. \tag{6.5}$$

We refer the reader to Section 6.C.1 for the proof. We thus obtain the following corollary.

Corollary 6.3.6. *If F is (f,ν)-robust and local losses $(\mathcal{L}_i)_{i\in\mathcal{H}}$ are convex, satisfy Assumption 6.3.4, and $\mathcal{L}_{\mathcal{H}}$ is μ-strongly convex, then $\tilde{\nabla}\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})$ the output of Algorithm 3 is a $(\zeta^2,\tilde{\alpha})$-inexact gradient oracle, with $\zeta^2=\nu G^2$, $\tilde{\alpha}=\nu B^2$ and (G^2,B^2) as in Equation (6.5).*

6.3.2 Convergence Results

We now state the complete Byzantine-robust distributed algorithm and its convergence guarantees.

Algorithm 4 Preconditioned Inexact Gradient with Similarity

Input: F, η, $\boldsymbol{x}_0$, $\hat{\mathcal{L}}$
for $k=0$ **to** $K-1$ **do**
  Sample $\tilde{\nabla}\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_k)$ using Algorithm 3.
  $\boldsymbol{x}_{k+1}\approx\arg\min_{\boldsymbol{x}\in\mathbb{R}^d}\left[\phi_\eta^k(\boldsymbol{x})=\langle\tilde{\nabla}\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_k)-\nabla\hat{\mathcal{L}}(\boldsymbol{x}_k),\boldsymbol{x}\rangle+\frac{1}{2\eta}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2+\hat{\mathcal{L}}(\boldsymbol{x})\right]$.
end for

Theorem 6.3.7. *Assume $\mathcal{L}_{\mathcal{H}}$ is μ-strongly convex and L-smooth, and $(\mathcal{L}_{\mathcal{H}}-\hat{\mathcal{L}})$ is Δ-smooth. Consider Algorithm 4 with a $(\zeta^2,\tilde{\alpha})$ inexact oracle (Definition 6.3.2) with*

$\tilde{\alpha} < 1/4$, a step-size $\eta \leq 1/2\Delta$, and where the proximal step is computed with precision

$$\|\nabla \phi_\eta^k(\boldsymbol{x}_{k+1})\|^2 \leq \frac{\mu\Delta}{4}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 + E^2. \tag{6.6}$$

Define $s_k = (1 + \frac{\eta\mu}{3})^k$, and $S_K = \sum_{k=0}^K s_k$. The weighted average $\hat{\boldsymbol{x}}_k := \frac{1}{S_K}\sum_{k=0}^K s_k \boldsymbol{x}_k$ satisfies

$$\mathcal{L}_{\mathcal{H}}(\hat{\boldsymbol{x}}_K) - \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^*) \leq \left(1 + \frac{\eta\mu}{3}\right)^{1-K} \frac{2}{\eta(1-4\tilde{\alpha})}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 + 2\frac{\zeta^2 + E^2}{\mu(1-4\tilde{\alpha})}. \tag{6.7}$$

Importantly, in Algorithm 4, each $\boldsymbol{x}_{k+1}$ is obtained by solving an optimization subroutine on the loss ϕ_η^k which the server can easily access. Equation (6.6) captures the error of this subroutine, and E^2 can be made arbitrarily small by allocating more computational budget on the server side, without involving more communication or requiring more homogeneity or honest nodes. This subproblem can also be warm-started, starting the minimization from $\boldsymbol{x}_k$, significantly reducing the computational burden. Interestingly, because of the absolute error due to adversarial nodes, taking $E^2 = o(\zeta^2)$ can save significant computational resources without affecting the precision.

The proof of Theorem 6.3.7 draws from Woodworth et al. (2023), who analyzed the gradient method under similarity with approximate proximal steps, but exact gradients. In Section 6.D.2, we introduce a new Lyapunov function to handle (ζ^2, α)-inexact gradients. We now instantiate Theorem 6.3.7 in the Byzantine setting.

Corollary 6.3.8. *Under the assumption of Corollary 6.3.6, consider Algorithm 4 with $\hat{\mathcal{L}} = \mathcal{L}_1$ and $\tilde{\nabla}\mathcal{L}_{\mathcal{H}}$ computed with Algorithm 3 and a $(f, \nu = \mathcal{O}(f/n-2f))$ robust-aggregation rule. Take $\eta = 1/2\Delta$, and $E^2 \leq O(\frac{f}{n-2f}G^2)$. Then, under $\frac{f}{n-2f}B^2 = O(1)$, we have after K iterations that*

$$K = \mathcal{O}\left(\frac{\Delta}{\mu}\log(1/\varepsilon)\right) \implies \mathcal{L}_{\mathcal{H}}(\hat{\boldsymbol{x}}_K) - \mathcal{L}_{\mathcal{H}}^* = \mathcal{O}\left(\frac{f}{n-2f}\frac{G^2}{\mu} + \varepsilon\right).$$

Remark 6.3.9. Note that Δ/μ can be much smaller than $\kappa = L/\mu$, the condition number of the global loss $\mathcal{L}_{\mathcal{H}}$, thus significantly speeding-up optimization. Indeed, assume that $\mathcal{L}_i(\boldsymbol{x}) := \frac{1}{|\mathcal{D}_i|}\sum_{a\in\mathcal{D}_i} l(a, \boldsymbol{x}) + \lambda/2\|\boldsymbol{x}\|^2$, where each dataset $\mathcal{D}_i$ consist of m i.i.d. samples from the same distribution $\mathcal{D}$. Assume $\|\nabla^2_{\boldsymbol{x}} l(a, \boldsymbol{x})\|_{op} \leq L$. Then w.l.o.g., $\|\nabla^2\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\|_{op} \leq L$, while Hoeffding's inequality for matrices (Tropp, 2015) yields, with probability at least $1 - \delta$,

$$\Delta = \|\nabla^2\mathcal{L}_1(\boldsymbol{x}) - \nabla^2\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\|_{op} = \mathcal{O}\left(\sqrt{\frac{L^2\log(d/\delta)}{m}}\right).$$

It follows $\Delta/\mu \leq \mathcal{O}\left(\kappa/\sqrt{m}\,\log(d/\delta)\right)$. Thus, the iteration complexity of Algorithm 4 to reach a given precision can be reduced by up to $\mathcal{O}(\sqrt{m})$ compared to Byzantine-robust D-GD, ignoring logarithmic factors. When local datasets are large, this yields a huge

improvement in the communication complexity. For refined and stronger control of Δ, including better dependencies on m, see Hendrikx et al. (2020b).

6.4 An Accelerated Inexact Extra-gradient Method

In this section, we adapt the accelerated method from Kovalev et al. (2022) to the inexact oracle setting, leading to Algorithm 5, an accelerated method for distributed optimization under similarity.

6.4.1 General Inexact Algorithm

We first give the convergence guarantees for Algorithm 5, independently of the Byzantine robust application.

Theorem 6.4.1. *Assume that $\mathcal{L}_{\mathcal{H}}$ is μ-strongly convex and L-smooth, that $\mathcal{L}_{\mathcal{H}} - \hat{\mathcal{L}}$ is Δ-smooth, and that $\tilde{\nabla}\mathcal{L}_{\mathcal{H}}$ is a (ζ^2, α)-inexact oracle with $\alpha < \min\left(\frac{\Delta^2}{4L^2}, \frac{1}{16}\right)$. Consider Algorithm 5 with the parametrization*

$$\gamma = \mu, \quad \theta = \frac{1}{2\sqrt{2}\Delta}, \quad \eta = \min\left(\frac{1}{2\mu}, \frac{\theta}{24\tau}\right), \quad \text{and} \quad \tau = \min\left(\frac{\theta\mu}{16\alpha}, \sqrt{\frac{\theta\mu}{48}}\right).$$

Then, if $\left\|\nabla\phi_\theta^k(\boldsymbol{x}_f^{k+1})\right\|^2 \leq \frac{1}{8\theta^2}\left\|\boldsymbol{x}_f^{k+1} - \boldsymbol{x}_g^k\right\|^2$, we have that

$$\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_f^K) - \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^*) = O\left((1-\rho)^K R + \frac{\zeta^2}{\mu}\right),$$

where $\rho = c\min\left(\sqrt{\frac{\mu}{\Delta}}, \frac{\mu}{\alpha\Delta}\right)$ with $c > 0$ a constant, and $R = \tau^2\Delta\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2 + \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^0) - \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^*)$.

We refer the reader to Section 6.D.3 for the proof of Theorem 6.4.1.

Algorithm 5 Accelerated Inexact Extra-gradient

1: **Input:** $\boldsymbol{x}^0 = \boldsymbol{x}_f^0 \in \mathbb{R}^d$, inexact oracle sampler $\tilde{\nabla}\mathcal{L}_{\mathcal{H}}$.
2: **Parameters:** $\tau \in (0,1)$, $\eta, \theta, \gamma > 0$, $K \in \{1, 2, \ldots\}$
3: **for** $k = 0, 1, 2, \ldots, K-1$ **do**
4: $\quad \boldsymbol{x}_g^k = \tau\boldsymbol{x}^k + (1-\tau)\boldsymbol{x}_f^k$
5: $\quad \boldsymbol{x}_f^{k+1} \approx \arg\min_{\boldsymbol{x}\in\mathbb{R}^d}\left[\phi_\theta^k(\boldsymbol{x}) := \langle\tilde{\nabla}\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_g^k) - \nabla\hat{\mathcal{L}}(\boldsymbol{x}_g^k), \boldsymbol{x} - \boldsymbol{x}_g^k\rangle + \frac{1}{2\theta}\left\|\boldsymbol{x} - \boldsymbol{x}_g^k\right\|^2 + \hat{\mathcal{L}}(\boldsymbol{x})\right]$
6: $\quad \boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \eta\gamma(\boldsymbol{x}_f^{k+1} - \boldsymbol{x}^k) - \eta\tilde{\nabla}\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_f^{k+1})$
7: **end for**
8: **Output:** $\boldsymbol{x}^K$

6.4.2 Acceleration Without Similarity

When no cheap proxy is available, one can take $\hat{\mathcal{L}} = 0$. This removes the proximal update from line 5 of Algorithm 5, leading to

$$\boldsymbol{x}_f^{k+1} = \boldsymbol{x}_g^k - \theta \tilde{\nabla} \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_g^k).$$

This provides a forward-only inexact first-order accelerated algorithm, with state-of-the-art guarantees.

Corollary 6.4.2 (Acceleration with inexact oracles)**.** *Let the setting of Theorem 6.4.1 holds, with $\hat{\mathcal{L}} = 0$, and thus $\Delta = L$. Then, if $\alpha < \frac{1}{\sqrt{\kappa}}$, Algorithm 5 ensures*

$$K = O(\sqrt{\kappa}\log(R/\varepsilon)) \implies \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_f^K) - \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^*) = O\left(\zeta^2/\mu + \varepsilon\right).$$

Tightness of the result. In this regime, Corollary 6.4.2 is, to our knowledge, the first convergence result of an accelerated algorithm for (ζ^2, α)-inexact oracles with both $\zeta^2 \neq 0$ and $\alpha \neq 0$. Additionally, previous analyses with an absolute error (Devolder et al., 2014; Vasin et al., 2023) suffered from an inflated asymptotic error in $\sqrt{\kappa}\zeta^2/\mu$ (see Section 6.2.3.2), which can be alleviated using the extra-gradient approach. Note that our condition on the relative error $\alpha < 1/\sqrt{\kappa}$ is also strictly better than previous work (Vasin et al., 2023; Kornilov et al., 2025), which required $\alpha < 1/\kappa$.

6.4.3 Focus on the Byzantine Setting

We now instantiate the convergence result of Algorithm 5 in the Byzantine setting, first in a standard distributed setting, without relying on a proxy loss $\hat{\mathcal{L}}$, and then in the similarity setting.

Corollary 6.4.3 (Byzantine-resilient plain acceleration)**.** *Let $\mathcal{L}_{\mathcal{H}}$ be L-smooth and μ-strongly convex, let Assumption 6.2.2 hold and assume $B^2\frac{f}{n-2f} \in O(\frac{1}{\sqrt{\kappa}})$. Then Algorithm 5 with an $(f, \nu = O(f/n-2f))$-aggregator, $\hat{\mathcal{L}} = 0$, after K iteration ensures*

$$K = O\left(\sqrt{\kappa}\log(1/\varepsilon)\right) \implies \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_f^K) - \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^*) = O\left(\frac{f}{n-2f}\frac{G^2}{\mu} + \varepsilon\right).$$

Corollary 6.4.4 (Byzantine-resilient acceleration under similarity)**.** *Under the assumptions of Corollary 6.3.8, if additionally $\frac{f}{n-2f} \in O(\frac{1}{\kappa^2})$, then Algorithm 5 with an $(f, \nu = O(f/n-2f))$-aggregator after K iteration ensures*

$$K = O\left(\sqrt{\Delta/\mu}\log(1/\varepsilon)\right) \implies \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_f^K) - \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^*) = O\left(\frac{f}{n-2f}\frac{G^2}{\mu} + \varepsilon\right).$$

Corollary 6.4.4 is the accelerated version of Corollary 6.3.8 (the PIGS result). Again, leveraging optimization under similarity replaces the standard dependence of the number of iterations on $\kappa = L/\mu$ by Δ/μ, which is smaller in general.

Yet, while the robustness regime of PIGS is $\frac{f}{n-2f} \leq O(\frac{1}{B^2}) = O(\frac{\mu^2}{\Delta^2})$ (when a robust aggregator with $\nu = O(f/n-2f)$ is used), which does not depends on the smoothness L at all, Theorem 6.4.1 requires $\frac{f}{n-2f} \leq \Omega(\kappa^{-2})$, which is much larger if we expect optimization under similarity to be worth the extra computational effort. We recover the same behavior as with standard acceleration: acceleration speeds up convergence, but can only be used if the number of Byzantine nodes is low enough (smaller breakdown point).

6.5 Experiments

6.5.1 Setup

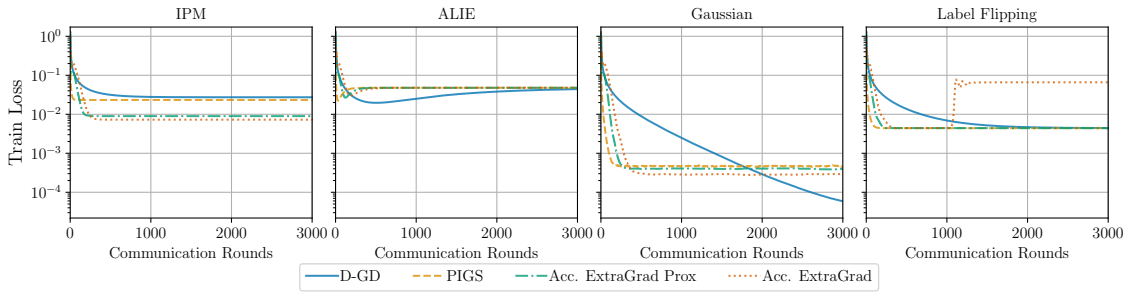


Figure 6.1: Performance on MNIST with a Huber aggregation under various attacks. There are 40 honest nodes with 1 adversary. Heterogeneous setting ($\beta = 1$).

Setup. We consider a l2-regularized Logistic Regression task on the MNIST dataset (LeCun et al., 1998). To simulate heterogeneity among clients, we follow the Dirichlet partitioning method from Hsu et al. (2019); Allouah et al. (2023a): each client's dataset class proportions are sampled from a Dirichlet distribution with parameter β. Lower β corresponds to higher heterogeneity. We refer to $\beta = 1$ as *heterogeneous* and to $\beta = 5$ as *mildly heterogeneous*.

Optimization Algorithms. We compare the accelerated extra-gradient with similarity (Acc. Extragrad Prox) and without (Acc. Extragrad), Distributed Gradient Descent (D-GD), and PIGS. In all methods, inexact gradients are computed using Algorithm 3 with a Huber aggregation (Algorithm 6). In Section 6.F.3 we also compare the Huber aggregation with Trimmed Mean (Yin et al., 2018), Krum (Blanchard et al., 2017) and Covariance-agnostic Filter (CAF) (Allouah et al., 2025b) (cf. Section 6.C.2 for theoretical guarantees of the optimization algorithm combined with CAF).

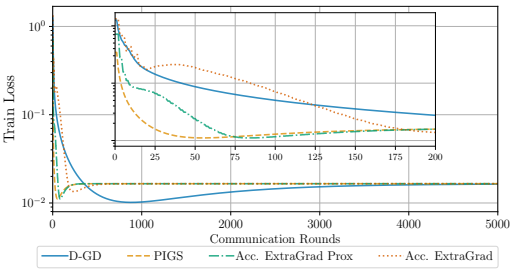


Figure 6.2: Distributed Logistic Regression, with 40 honest and 1 Byzantine client in the mildly heterogeneous setting ($\beta = 5$), with a $l2$-regularization $\mu = 10^{-2}$. Train loss $\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_k) - \mathcal{L}^*_{\mathcal{H}}$ under ALIE attack. Plot with a zoom on the first 200 iterations.

Parameters and Attacks. We tune optimizer parameters via hyperparameter search in the Byzantine-free setting. The attacks simulated are the Label Flipping attack (Allen-

Zhu et al., 2021), a Gaussian noise attack, as well as variants of A Little Is Enough (ALIE) (Baruch et al., 2019), Inner Product Manipulation (Xie et al., 2020), where the attack is scaled using a line search to maximize impact. Our code, accessible here, is built on top of the ByzFL library (González et al., 2025).

6.5.2 Analysis

Communication Complexity. As shown in all experiments, D-GD is significantly slower than the accelerated and/or similarity-based approaches. More surprisingly, in the considered setting, PIGS is stable with significantly larger step-sizes than the Accelerated Extra-gradient with similarity, hence converging faster. Note still that the prox extra-gradient algorithm is faster than the non-prox version.

6.6 Conclusion

In this work, we proposed to abstract away the problem of Byzantine Optimization by formalizing it as an inexact gradient oracle problem. Our approach opens up a new systematic approach to study and design Byzantine-resilient algorithms. We leverage this to propose both a non-accelerated optimization-under-similarity method, and an accelerated extra-gradient scheme which can be used both in the standard distributed setting and in the similarity setting. Those methods have better communication complexity (and a better computation complexity for the accelerated method without similarity), as we show theoretically and experimentally.

Acknowledgement

We would like to thank Adrien Taylor for the helpful discussions and feedback. The work of Aymeric Dieuleveut and Renaud Gaucher was supported by French State aid managed by the Agence Nationale de la Recherche (ANR) under the France 2030 program with the reference ANR-23-PEIA-005 (REDEEM project). The work of Aymeric Dieuleveut was also supported by Hi!Paris - FLAG chair.

Appendix

6.A Related Concurrent Work Shi et al. (2025)

In this section, we describe why Shi et al. (2025) do not actually prove an accelerated rate of convergence. This highlights that proving accelerated rates under adversarial corruptions is non-trivial, and emphasizes the strength of our abstraction: it allows to reduce the complexity of the problem, which helps to provide tighter analysis under weaker assumptions. Indeed, Shi et al. (2025) assume (G, B)-Heterogeneity with $B = 0$, while our results cover the setting $B > 0$.

The main convergence result in Shi et al. (2025) - Theorem 13 - requires that conditions on the parameters θ and α (and thus β), and on $(\chi_i)_{i\in[5]}$ in their Lemma 20 are met. Contrary to what is claimed in Corollary 13, choosing α and θ such that those conditions are met requires that the algorithm converges at a non-accelerated rate.

Specifically: the inequalities on χ_2 and χ_3 impose that $\beta(\theta+\beta) < 1$ must hold. Writing β using θ and κ yield the polynomial inequality $\theta^{3/2} - \kappa\theta + 4\sqrt{\kappa} > 0$. A quick numerical analysis shows that this inequality is met either for $\theta \geq \kappa + O(1/\kappa)$ or $\theta \leq O(1/\kappa)$. Since the convergence rate is $(1 - \sqrt{\theta/\kappa})^t$, the regime of large θ can be discarded (it breaks the convergence rate lower bound of strongly convex functions), and only the regime of small θ is feasible, in which the convergence rate is $(1 - O(1/\kappa))^t$, i.e. a *non-accelerated* one.

Note that Corollary 13 claims that choosing $\theta = 1$ and $\alpha = 0$ satisfies the inequalities of Lemma 20. This is only true when κ is very small, thus again not showing a universal acceleration result.

6.B Robust Aggregation Rules

In this section, we recall (f, ν)-robust averaging guarantees of standard aggregators. We also add two contributions to the vast literature of robust aggregators:

1. **A new analysis of the multivariate Huber estimator.** In Section 6.B.1, we discuss the relation with the multivariate estimator and the centered clipping scheme (Karimireddy et al., 2023). We prove the (f, ν)-robustness of a multivariate Huber estimator, which remarkably has state-of-the-art constant factors on ν.

2. **A generalization of the Nearest Neighbor Mixing (NNM) strategy of Allouah et al. (2023a).** Allouah et al. (2023a) show that relying on a mixing strategy *before* robustly averaging the input vectors allows for better robustness

guarantees. We generalize this strategy with the notion of *robust mixing*, and propose a pre-aggregator with better theoretical guarantees.

Overview of the (f,ν)-robustness. In Table 6.2 we recall the robustness guarantees (Allouah et al., 2023a) of the coordinate-wise trimmed mean, Krum, the geometric median and the coordinate-wise median. We additionally provide the robustness guarantees of our analysis of the multivariate Huber estimator.

Table 6.2: Robustness coefficients of the coordinate-wise trimmed mean (CWTM), Krum, the geometric median (GM), the coordinate-wise median (CWM), and a multivariate Huber estimator.

Aggregation	Huber	CWTM	Krum	GM	CWM	Lower Bound
ν	$\frac{2f}{n-3f}$	$\frac{6f}{n-2f}\left(1+\frac{6f}{n-2f}\right)$	$6\left(1+\frac{6f}{n-2f}\right)$	$4\left(1+\frac{f}{n-2f}\right)^2$	$4\left(1+\frac{f}{n-2f}\right)^2$	$\frac{f}{n-2f}$

Generic definition of a robust mixing. Allouah et al. (2023a) suggest improving aggregation rules by pre-processing the input values through a pre-aggregation step, and specifically the Nearest Neighbors Mixing (NNM) strategy. We generalize this mixing strategy in the following formal definition:

Definition 6.B.1. A map $F_1 : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}$ is said to be a (f,δ) robust mixing if, when $|\mathcal{B}| \leq f$, and given F_2 a (f,ν)-robust aggregation rule, $F_2 \circ F_1$ is a $(f,\tilde{\nu})$-robust aggregation rule with

$$\tilde{\nu} = \delta(1+\nu).$$

We list in Table 6.3 a few robust mixing strategies, which can be built generally using the robust-gossip framework, as we show in Section 6.B.2.

Table 6.3: Robust Pre-Aggregation / Mixing Strategies. F in F-RG is a $(\rho, b = \frac{f}{n-f})$-robust summand.

Pre-Aggregation	CS_+-RG	NNM/NNA	F-RG	Lower Bound
δ	$\frac{4f}{n-f}$	$\frac{8f}{n-f}$	$\frac{2\rho f}{n-f}$	
Breakdown	$\frac{1}{5}$	$\frac{1}{9}$	$\frac{1}{2\rho+1}$	$\frac{1}{3}$

6.B.1 Robustness of the Multivariate Huber Estimator.

In this section, we propose a multivariate Huber estimator (Huber, 1964) with a dynamic thresholding rule, and show that it is a (f,ν)-robust averaging rule.

The multivariate Huber robust estimator, relating back to the very beginning of robust statistics, is defined as

$$F_{Huber}(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n;\tau) = \arg\min_{\boldsymbol{x}} \left\{ \Psi(\boldsymbol{x};(\boldsymbol{x}_i)_{i\in[n]}) := \sum_{i\in[n]} \psi_\tau(\boldsymbol{x}_i - \boldsymbol{x}) \right\},$$

where $\tau > 0$ is a constant threshold and ψ_τ denotes the Huber loss defined as

$$\psi_\tau(\boldsymbol{x}) = \begin{cases} \frac{1}{2}\|\boldsymbol{x}\|^2 \text{ if } \|\boldsymbol{x}\| \leq \tau, \\ \tau\|\boldsymbol{x}\| - \frac{\tau}{2} \text{ if } \|\boldsymbol{x}\| > \tau. \end{cases}$$

Remark 6.B.2. The centered-clipping scheme (CC) (Karimireddy et al., 2023) is an approximation of the multivariate Huber estimator with constant threshold coincide. Indeed, the centered clipping aggregation rule starts from an estimate of the average $\boldsymbol{y}^0$, and iteratively improve this estimate during T iterations using

$$\boldsymbol{y}^{t+1} = \boldsymbol{y}^t - \eta \sum_{i\in[n]} \text{clip}(\boldsymbol{x}_i^t - \boldsymbol{y}^t; \tau), \quad \text{where} \quad \text{clip}(\boldsymbol{x}; \tau) := \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}\min(\tau, \|\boldsymbol{x}\|).$$

Which corresponds to gradient descent steps on the Huber empirical loss function $\Psi(\,\cdot\,; (\boldsymbol{x}_i)_{i\in[n]})$.
Note however that choice of the threshold τ in the multivariate Huber estimator (and CC) impacts the robustness of the procedure: indeed, choosing $\tau \to \infty$ yields a (non-robust) simple average, while $\tau \to 0$ outputs $\boldsymbol{y}_0$, i.e. the initial guess of the average.

In practice, the clipping threshold used in the theoretical guarantees in Karimireddy et al. (2021) can not be computed, as it depends on the variance of the honest inputs $(\boldsymbol{x}_i)_{i\in\mathcal{H}}$. To alleviate this flaw, we propose in Algorithm 6 a multivariate Huber estimator with a threshold choice agnostic to the (unknown) variance of the honest inputs. The estimator iteratively updates the threshold τ, such that the multivariate Huber estimator is, after a few iterations, (f, ν)-robust.

Algorithm 6 Byzantine-Robust Huber Estimator

Input: Initial average guess $\boldsymbol{y}_0$, Upper bound on the number of adversaries f, Input vectors $(\boldsymbol{x}_i)_{i\in[n]}$
for $t = 0$ **to** $T - 1$ **do**
  Sort $(\|\boldsymbol{y}^t - \boldsymbol{x}_i\|, i \in [n])$ with a permutation $\sigma_k : [n] \to [n]$ into

$$\|\boldsymbol{y}^t - \boldsymbol{x}_{\sigma_k(1)}\| \geq \ldots \geq \|\boldsymbol{y}^t - \boldsymbol{x}_{\sigma_k(n)}\|$$

  Set $\tau^t = \|\boldsymbol{y}^t - \boldsymbol{x}_{\sigma_k(2f)}\|$
  Update $\boldsymbol{y}^{t+1} = \boldsymbol{y}_t - \frac{1}{n-f}\sum_{i\in[n]} \text{clip}(\boldsymbol{y}^t - \boldsymbol{x}_i; \tau^t)$
end for
Output $\boldsymbol{y}^T$

The aggregation rule given by Algorithm 6 is probably robust.

Proposition 6.B.3. *Consider* $\nu_0 = \frac{2f}{n-f}$, *and* $\nu = \frac{2f}{n-3f}$, *then for any* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$, *and any* $S \subset [n]$ *of size* $n - f$, *after* T *iterations Algorithm 6 yields*

$$\|\boldsymbol{y}^T - \overline{\boldsymbol{x}}_S\|^2 \leq \nu_0^T\|\boldsymbol{y}_0 - \overline{\boldsymbol{x}}_S\|^2 + \nu(1 - \nu_0^T)\frac{1}{n-f}\sum_{i\in S}\|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_S\|^2.$$

Thus, the multivariate Huber estimator is $(f, \nu+\varepsilon)$-robust after

$$\frac{\log(\|\boldsymbol{y}_0 - \overline{\boldsymbol{x}}_S\|^2 - \frac{\nu}{n-f}\sum_{i\in S}\|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_S\|^2) + \log(1/\varepsilon)}{\log(1/\nu_0)}$$

iterations.

Proof. Writing the update of Algorithm 6 and applying the triangle inequality yields

$$\begin{aligned}\|\boldsymbol{y}^t - \overline{\boldsymbol{x}}_S\|^2 &= \left\|\frac{1}{n-f}\sum_{i\in S}\boldsymbol{y}^t - \boldsymbol{x}_i - \mathrm{clip}(\boldsymbol{y}^t - \boldsymbol{x}_i; \tau^k) + \frac{1}{n-f}\sum_{i\in[n]\setminus S}\mathrm{clip}(\boldsymbol{x}_i - \boldsymbol{y}^t; \tau^t)\right\|^2 \\ &\leq \left(\frac{1}{n-f}\sum_{i\in S}(\|\boldsymbol{y}^t - \boldsymbol{x}_i\| - \tau^t)_+ + \frac{f}{n-f}\tau^t\right)^2,\end{aligned}$$

where we denoted $(a)_+ = \max(a, 0)$.

Setting τ^t such that they are at least f terms in the left sum, and at most $2f$, for instance $\tau^t = \|\boldsymbol{y}^t - \boldsymbol{x}_{\sigma(2f)}\|$, yields using Cauchy-Schwarz inequality,

$$\begin{aligned}\|\boldsymbol{y}^t - \overline{\boldsymbol{x}}_S\|^2 &\leq \left(\frac{1}{n-f}\sum_{i\in S}\|\boldsymbol{y}^t - \boldsymbol{x}_i\|1_{\sigma_k(i)\leq 2f}\right)^2 \\ &\leq \frac{2f}{n-f}\frac{1}{n-f}\sum_{i\in S}\|\boldsymbol{y}^t - \boldsymbol{x}_i\|^2.\end{aligned}$$

The latter inequality can be decomposed, denoting $d^t = \|\boldsymbol{y}^t - \overline{\boldsymbol{x}}_S\|^2$ and $e = \frac{1}{n-f}\sum_{i\in S}\|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_S\|^2$, as

$$d^{t+1} \leq \nu_0(d^t + e).$$

This is an algebraic-arithmetic-like sequence. Consequently, it holds

$$\left(d^t - \frac{\nu_0}{1-\nu_0}e\right) \leq \nu_0^t\left(d_0 - \frac{\nu_0}{1-\nu_0}e\right).$$

From which the proposition derives, using $\nu = \frac{\nu_0}{1-\nu_0}$. □

Remark 6.B.4. To our knowledge, the multivariate Huber estimator is currently the (f, ν)-robust estimator with the tightest ν. For instance, investigating the range of $f \in [n]$ for which the condition $\nu < 1$ holds - which is required to have useful convergence guarantees when plugged in Algorithm 3 - yields $5f < n$ for the Huber estimator, while for instance CWTM requires approximately $25f < n$.
Similarly, the best (f, ν) robust averaging rule deriving from Proposition 6.B.6 is $\nu = \frac{2\rho f}{n-2(1+\rho)f}$, which, when considering that necessarily $\rho > 1$, is always worse than $\frac{2f}{n-3f}$.

6.B.2 From Byzantine-Robust Gossip to Byzantine-Robust Distributed Learning

Algorithm 7 Robust-Gossip based Pre-Aggregation

Input: F_2 $(\rho, \frac{f}{n-f})$-robust summand rule, F_1 (f, ν)-robust aggregation rule, vectors input $(\boldsymbol{x}_i)_{i\in[n]}$.
for Client $i \in [n]$ **do**
  Client Send $\boldsymbol{x}_i$ to Server
end for
Server Update $\boldsymbol{y}_i = \boldsymbol{x}_i - F((\boldsymbol{x}_i - \boldsymbol{x}_j, \frac{1}{n-f})_{j\in[n]})$

Nearest Neighbor Mixing (NNM) can be seen as an emulation of the Nearest Neighbor Averaging (NNA) algorithm (Farhadkhani et al., 2023). As such, the mixing strategy constitutes an intrinsically decentralized aggregation scheme, which can be further generalized within the Robust Gossip framework (Gaucher et al., 2025). More precisely, if the server simulates a single iteration of an F-Robust Gossip (Gaucher et al., 2025), with F a (b, ρ)-robust summand, the resulting update satisfies the mixing property required by NNM. This allows us to design even tighter pre-aggregation strategy.

F-Robust Gossip (or F-RG) is a decentralized algorithm, in the sense that all clients are seen as vertices in a communication graph $\mathcal{G}$, in which they communicate synchronously with their neighbors through the edges. The algorithm relies on so-called (b, ρ)-robust summand functions, defined as follows:

Definition 6.B.5 ((b, ρ)–robust summation)**.** Let $b, \rho \geq 0$. An aggregation rule $F : (\mathbb{R}_+ \times \mathbb{R}^d)^n \to \mathbb{R}^d$ is a (b, ρ)–*robust summation* if, for any vectors $(\boldsymbol{z}_i)_{i\in[n]} \in (\mathbb{R}^d)^n$, any weights $(\omega_i)_{i\in[n]} \in \mathbb{R}_+^n$ and any $S \subset [n]$ such that $\sum_{i\in\overline{S}} \omega_i \leq b$ (where $\overline{S} := [n]\backslash S$),

$$\left\| F\big((\omega_i, \boldsymbol{z}_i)_{i\in[n]}\big) - \sum_{i\in S} \omega_i \boldsymbol{z}_i \right\|^2 \leq \rho b \sum_{i\in S} \omega_i \|\boldsymbol{z}_i\|^2.$$

Given F a (b, ρ)-robust summand, and weights $(w_{ij})_{i,j\in[n]}$, one step of F-RG updates the client's parameter $(\boldsymbol{x}_i)_{i\in[n]}$ according to

$$\forall i \in \mathcal{H}, \quad \boldsymbol{y}_i = \boldsymbol{x}_i - \eta F((\boldsymbol{x}_i - \boldsymbol{x}_j, w_{ij})_{j\in[n]}).$$

One particular instance of F-RG simulated by a central server in a server–client architecture is the Nearest Neighbor Mixing (NNM) pre-aggregation scheme (Allouah et al., 2023a). This method corresponds alternatively to NNA (Farhadkhani et al., 2023) or to F-RG with weights $w_{ij} = 1/(n-f)$ and step size $\eta = 1$ when F is chosen as the geometrically trimmed sum (GTS), formally:

$$\mathrm{GTS}\left((\boldsymbol{z}_1, {}^1\!/_{n-f}), \ldots, (\boldsymbol{z}_n, {}^1\!/_{n-f})\right) := \frac{1}{n-f} \sum_{n-f \text{ smallest } \|\boldsymbol{z}_i\|} \boldsymbol{z}_i.$$

Proposition 6.B.6. *Let $\mathcal{G}$ be the fully-connected graph on $[n]$ with uniform weights on the edges $\frac{1}{n-f}$. Let F_1 be a (f,ν)-robust aggregator, and F_2 be a (ρ,b) robust summand, with $b=\frac{f}{n-f}$. Then $F_1\circ(F_2\text{-}RG)$ as in Algorithm 7 is a $(\tilde{\nu},f)$-robust aggregator, with*

$$\tilde{\nu}=\delta(1+\nu) \quad \textit{where} \quad \delta:=\frac{2\rho f}{n-f}.$$

We refer to δ as the mixing contraction factor.

Proof.

Gaucher et al. (2025) show that, for graph $\mathcal{G}$ of spectral gap γ, $\mathrm{F}_2-\mathrm{RG}$, with F_2 a (b,ρ)-robust aggregator with step size $\eta=\mu_{\max}^{-1}(\mathcal{G}_{\mathcal{H}})$, ensures that, for $(\boldsymbol{y}_i)_{i\in[n]}=\mathrm{F}_2-\mathrm{RG}(\boldsymbol{x}_i,i\in[n])$, and $b<\mu_2(\mathcal{G}_{\mathcal{H}})$

$$\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{y}_i-\overline{\boldsymbol{x}}_{\mathcal{H}}\|^2\leq(1-\gamma(1-\delta))\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{x}_i-\overline{\boldsymbol{x}}_{\mathcal{H}}\|^2, \tag{6.8}$$

where $\delta=\frac{2\rho b}{\mu_2(\mathcal{G}_{\mathcal{H}})}$.

Thus, plugging the $\boldsymbol{y}_i$ in F_2 leads to

$$\begin{aligned}
\|F_2(\boldsymbol{y}_1,\dots,\boldsymbol{y}_n)-\overline{\boldsymbol{x}}_{\mathcal{H}}\|^2 &\leq (1+\varepsilon^{-1})\|F_2(\boldsymbol{y}_1,\dots,\boldsymbol{y}_n)-\overline{\boldsymbol{y}}_{\mathcal{H}}\|^2+(1+\varepsilon)\|\overline{\boldsymbol{y}}_{\mathcal{H}}-\overline{\boldsymbol{x}}_{\mathcal{H}}\|^2 \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{Young's inequality} \\
&\leq \nu(1+\varepsilon^{-1})\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{y}_i-\overline{\boldsymbol{y}}_{\mathcal{H}}\|^2+(1+\varepsilon)\|\overline{\boldsymbol{y}}_{\mathcal{H}}-\overline{\boldsymbol{x}}_{\mathcal{H}}\|^2 \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (f,\nu)\text{-robustness} \\
&=(1+\nu)\left(\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{y}_i-\overline{\boldsymbol{y}}_{\mathcal{H}}\|^2+\|\overline{\boldsymbol{y}}_{\mathcal{H}}-\overline{\boldsymbol{x}}_{\mathcal{H}}\|^2\right) \quad \text{for } \varepsilon=\nu \\
&\leq (1+\nu)(1-\gamma(1-\delta))\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{x}_i-\overline{\boldsymbol{x}}_{\mathcal{H}}\|^2\ ,
\end{aligned}$$

where the last inequality stems from Equation (6.8) and a bias-variance decomposition.

Thus, if the central server emulates one step of F_2-RG on a fully connected graph, with uniform weights $\frac{1}{|\mathcal{H}|}$, we have that $\gamma=1$, $\mu_{\max}(\mathcal{G}_{\mathcal{H}})=\mu_{\min}(\mathcal{G}_{\mathcal{H}})=\eta^{-1}=1$, and thus $\delta=\frac{2\rho f}{n-f}$, and use F on the outputs (which corresponds to NNA for G=Trimmed Sum), we have that

$$\tilde{\nu}=\delta(1+\nu).$$

□

6.C From Hessian Similarity to Bounded Heterogeneity

In this section, we first show in Section 6.C.1 that the Hessian similarity assumption Assumption 6.3.1 implies the (G,B)-Heterogeneity assumption Assumption 6.2.2, as stated

in Proposition 6.3.5. We then show in Section 6.C.2 that this technique can also be used to control the error with a tighter dependency on the dimension.

6.C.1 Proof of Proposition 6.3.5

Proof.

Let us write the differences of gradients as an integral.

$$\nabla \mathcal{L}_i(\boldsymbol{x}) - \nabla \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}) = \nabla \mathcal{L}_i(\boldsymbol{x}^*) + \int_0^1 \nabla^2 \left(\mathcal{L}_i - \mathcal{L}_{\mathcal{H}}\right)\left(\boldsymbol{x}^* + t(\boldsymbol{x} - \boldsymbol{x}^*)\right) \cdot (\boldsymbol{x} - \boldsymbol{x}^*) dt.$$

We now use Young's inequality, then apply Jensen's inequality. This allows us to leverage the similarity Assumption 6.3.4, which, along with the convexity of both $\mathcal{L}_i$ and $\mathcal{L}_{\mathcal{H}}$, concludes the proof.

$$\begin{aligned}
\left\|\nabla \mathcal{L}_i(\boldsymbol{x}) - \nabla \mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\right\|^2 &\leq 2\left\|\nabla \mathcal{L}_i(\boldsymbol{x}^*)\right\|^2 \\
&\quad + 2\left\|\int_0^1 \nabla^2 \left(\mathcal{L}_i - \mathcal{L}_{\mathcal{H}}\right)\left(\boldsymbol{x}^* + t(\boldsymbol{x} - \boldsymbol{x}^*)\right) \cdot (\boldsymbol{x} - \boldsymbol{x}^*) dt\right\|^2 \\
&\leq 2\left\|\nabla \mathcal{L}_i(\boldsymbol{x}^*)\right\|^2 \\
&\quad + 2\int_0^1 (\boldsymbol{x} - \boldsymbol{x}^*)^T \left(\nabla^2 \left(\mathcal{L}_i - \mathcal{L}_{\mathcal{H}}\right)\left(\boldsymbol{x}^* + t(\boldsymbol{x} - \boldsymbol{x}^*)\right)\right)^2 \cdot (\boldsymbol{x} - \boldsymbol{x}^*) dt \\
&\leq 2\left\|\nabla \mathcal{L}_i(\boldsymbol{x}^*)\right\|^2 + 2\Delta^2 \left\|\boldsymbol{x} - \boldsymbol{x}^*\right\|^2
\end{aligned}$$

Averaging it on all $i \in \mathcal{H}$, and using that $\nabla \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^*) \triangleq 0$, yields

$$\frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \left\|\nabla \mathcal{L}_i(\boldsymbol{x}) - \nabla \mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\right\|^2 \leq \frac{2}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \left\|\nabla \mathcal{L}_i(\boldsymbol{x}^*)\right\|^2 + 2\Delta^2 \left\|\boldsymbol{x} - \boldsymbol{x}^*\right\|^2.$$

Using Equation (6.12), written here as $\left\|\boldsymbol{x} - \boldsymbol{x}^*\right\|^2 \leq \frac{1}{\mu^2}\left\|\nabla \mathcal{L}(\boldsymbol{x})\right\|^2$, gives the result. □

6.C.2 Extension to a Dimensionally-Tight Definition of Robustness

If the notion (f, ν)-robust aggregation rule is simple, and enjoys practical simple algorithms that achieve the criterion, it suffers from an obfuscation of the dependency on the dimension. Indeed, Theorem 6.2.3 is established using univariate counterexamples. As a consequence, it does not allows to understand the tightness of Byzantine-resilient algorithms for which the term $G^2 = \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \left\|\nabla \mathcal{L}_i(\boldsymbol{x}^*)\right\|^2$ might increase with the dimension.

And indeed, a line of work (Zhu et al., 2023; Allouah et al., 2023b, 2025b), inspired by the robust statistics literature, has shown that this dependency on the dimension can be alleviated, by suffering, instead of G, only a term $\tilde{G} := \left\|\frac{1}{\mathcal{H}} \nabla \mathcal{L}_i(\boldsymbol{x}^*) \nabla \mathcal{L}_i(\boldsymbol{x}^*)^T\right\|_{op}$, i.e. the largest eigenvalue of the empirical covariance matrix at the optimum.

This can be achieved by considering the broader notion of $(f, \nu, \|\cdot\|)$-robust aggregation rule.

Definition 6.C.1. . Let $\|\cdot\|$ be a norm on symmetric positive semi-definite matrices. An aggregation rule $F = \mathbb{R}^{n\times d} \to \mathbb{R}^d$ is said to be $(f, \nu, \|\cdot\|)$ robust if, $\forall\, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$, and any set $S \subset [n]$ of size $n - f$,

$$\|F(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) - \overline{\boldsymbol{x}}_S\|^2 \leq \nu \cdot \|\Sigma((\boldsymbol{x}_i)_{i\in S})\|,$$

with $\Sigma((\boldsymbol{x}_i)_{i\in S}) := \frac{1}{|S|}\sum_{i\in S}(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_S)(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_S)^T$.

Remark 6.C.2. As shown in Allouah et al. (2025b), an adaptation of the celebrated Filter (Diakonikolas et al., 2017) algorithm, coined Covariance-Agnostic Filter (CAF), is $(f, \nu = O(f/n-2f), \|\cdot\|_{op})$ robust, where $\|\cdot\|_{op}$ is the operator norm for matrices, i.e., for PSD matrices the largest absolute eigenvalue.

Remark 6.C.3. Considering $(f, \nu, \|\cdot\|_{op})$-robust aggregation rule may lead to a $\sqrt{d}$ dimension improvement w.r.t. (f, ν)-robust aggregation rule. Indeed, on the one hand we have that

$$\|\Sigma((\boldsymbol{x}_i)_{i\in S})\|_F \leq \frac{1}{|S|}\sum_{i\in S}\|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_S\|^2,$$

and, on the other hand,

$$\|\Sigma((\boldsymbol{x}_i)_{i\in S})\|_{op} \leq \|\Sigma((\boldsymbol{x}_i)_{i\in S})\|_F \leq \sqrt{d}\|\Sigma((\boldsymbol{x}_i)_{i\in S})\|_{op},$$

where both inequalities are tight for instances of PSD matrices. It follows that considering the robustness using the $(f, \nu, \|\cdot\|_{op})$ definition can give up to a $\sqrt{d}$ improvement when the covariance matrix is nearly isotropic.

Inexact gradients can be tight with respect to the dimension. Hopefully, the Inexact Oracle Framework works with $(f, \nu, \|\cdot\|_{op})$ robustness as well, and, in fact whenever $\|\cdot\|$ is unitary-invariant.

Proposition 6.C.4. *Let $\|\cdot\|$ be a unitary-invariant norm for PSD matrices. Under Assumption 6.3.4, if $(\mathcal{L}_i)_{i\in\mathcal{H}}$ are convex and $\mathcal{L}_{\mathcal{H}}$ is μ-strongly convex, Algorithm 3 with a $(f, \nu, \|\cdot\|)$-robust averaging rule (Definition 6.C.1) yields:*

$$\|\tilde{\nabla}\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\|^2 \leq \tilde{G}^2 + \alpha\|\nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\|^2, \tag{6.9}$$

where $\alpha = \nu\frac{2\Delta^2}{\mu^2}$ and $\tilde{G}^2 = 2\nu\left\|\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\nabla\mathcal{L}_i(\boldsymbol{x}^*)\nabla\mathcal{L}_i(\boldsymbol{x}^*)^T\right\|$.

Corollary 6.C.5. *Under Assumption 6.3.4, Algorithm 3 with CAF (Allouah et al.,*

2025b) yields a (ζ^2, α)-inexact oracle with

$$\zeta^2 = O\left(\frac{f}{n-2f}\left\|\frac{1}{\mathcal{H}}\sum_{i\in\mathcal{H}}\nabla\mathcal{L}_i(\boldsymbol{x}^*)\nabla\mathcal{L}_i(\boldsymbol{x}^*)^T\right\|_{op}\right) \quad \textit{and} \quad \alpha = O\left(\frac{f}{n-2f}\frac{\Delta^2}{\mu^2}\right).$$

This in turn can be used in PIGS Algorithm 4, GD, or the Extra-gradient method Algorithm 5 to achieve dimension-independent asymptotic errors.

Proof of Proposition 6.C.4.

To show the results, it is sufficient to show that

$$\|\Sigma((\nabla\mathcal{L}(\boldsymbol{x}))_{i\in\mathcal{H}})\| \leq 2\|\Sigma((\nabla\mathcal{L}_i(\boldsymbol{x}^*))_{i\in\mathcal{H}})\| + 2\frac{\Delta^2}{\mu^2}\|\nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\|^2.$$

We have that

$$\begin{aligned}
&\frac{1}{\mathcal{H}}\sum_{i\in\mathcal{H}}\left(\nabla\mathcal{L}_i(\boldsymbol{x}) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\right)\left(\nabla\mathcal{L}_i(\boldsymbol{x}) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\right)^T \\
&\qquad \preceq \frac{2}{\mathcal{H}}\sum_{i\in\mathcal{H}}\left(\nabla\mathcal{L}_i(\boldsymbol{x}) - \nabla\mathcal{L}_i(\boldsymbol{x}^*) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\right)\left(\nabla\mathcal{L}_i(\boldsymbol{x}) - \nabla\mathcal{L}_i(\boldsymbol{x}^*) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\right)^T \\
&\qquad\quad + \frac{2}{\mathcal{H}}\sum_{i\in\mathcal{H}}\nabla\mathcal{L}_i(\boldsymbol{x}^*)\nabla\mathcal{L}_i(\boldsymbol{x}^*)^T.
\end{aligned}$$

Now, using that for unitary-invariant norm for PSD matrices, if $M \preceq N$, then $\|M\| \leq \|N\|$. Thus

$$\begin{aligned}
&\left\|\frac{1}{\mathcal{H}}\sum_{i\in\mathcal{H}}\left(\nabla\mathcal{L}_i(\boldsymbol{x}) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\right)\left(\nabla\mathcal{L}_i(\boldsymbol{x}) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\right)^T\right\| \\
&\qquad \leq \left\|\frac{2}{\mathcal{H}}\sum_{i\in\mathcal{H}}\left(\nabla\mathcal{L}_i(\boldsymbol{x}) - \nabla\mathcal{L}_i(\boldsymbol{x}^*) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\right)\left(\nabla\mathcal{L}_i(\boldsymbol{x}) - \nabla\mathcal{L}_i(\boldsymbol{x}^*) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\right)^T\right\| \\
&\qquad\quad + \left\|\frac{2}{\mathcal{H}}\sum_{i\in\mathcal{H}}\nabla\mathcal{L}_i(\boldsymbol{x}^*)\nabla\mathcal{L}_i(\boldsymbol{x}^*)^T\right\|.
\end{aligned}$$

We control the first of the two terms as follows

$$\begin{aligned}
&\left\|\frac{2}{\mathcal{H}}\sum_{i\in\mathcal{H}}\left(\nabla\mathcal{L}_i(\boldsymbol{x}) - \nabla\mathcal{L}_i(\boldsymbol{x}^*) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\right)\left(\nabla\mathcal{L}_i(\boldsymbol{x}) - \nabla\mathcal{L}_i(\boldsymbol{x}^*) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\right)^T\right\| \\
&\leq \frac{2}{\mathcal{H}}\sum_{i\in\mathcal{H}}\left\|\left(\nabla\mathcal{L}_i(\boldsymbol{x}) - \nabla\mathcal{L}_i(\boldsymbol{x}^*) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\right)\left(\nabla\mathcal{L}_i(\boldsymbol{x}) - \nabla\mathcal{L}_i(\boldsymbol{x}^*) - \nabla\mathcal{L}_{\mathcal{H}}(\boldsymbol{x})\right)^T\right\| \\
&= \frac{2}{\mathcal{H}}\sum_{i\in\mathcal{H}}\left\|\int_0^1 \nabla^2\left(\mathcal{L}_i - \mathcal{L}_{\mathcal{H}}\right)\left(\boldsymbol{x}^* + t(\boldsymbol{x}-\boldsymbol{x}^*)\right)\cdot(\boldsymbol{x}-\boldsymbol{x}^*)dt\right\|_2^2 \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{Using } \|\mathbf{1}_d\mathbf{1}_d^T\| = \mathbf{1}_d^T\mathbf{1}_d = d \\
&\leq \frac{2}{\mathcal{H}}\sum_{i\in\mathcal{H}}\int_0^1 (\boldsymbol{x}-\boldsymbol{x}^*)^T\left(\nabla^2\left(\mathcal{L}_i - \mathcal{L}_{\mathcal{H}}\right)\left(\boldsymbol{x}^* + t(\boldsymbol{x}-\boldsymbol{x}^*)\right)\right)^2\cdot(\boldsymbol{x}-\boldsymbol{x}^*)dt
\end{aligned}$$

Using Jensen's inequality

Using Assumption 6.3.4, we obtain

$$\left\| \frac{2}{\mathcal{H}} \sum_{i \in \mathcal{H}} \Big(\nabla \mathcal{L}_i(\boldsymbol{x}) - \nabla \mathcal{L}_i(\boldsymbol{x}^*) - \nabla \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}) \Big) \Big(\nabla \mathcal{L}_i(\boldsymbol{x}) - \nabla \mathcal{L}_i(\boldsymbol{x}^*) - \nabla \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}) \Big)^T \right\| \leq \frac{2\Delta^2}{\mathcal{H}} \sum_{i \in \mathcal{H}} \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 = 2\Delta^2 \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 .$$

Eventually, combining the two terms yields

$$\|\Sigma_{\mathcal{H}}(\nabla \mathcal{L}_i(\boldsymbol{x}))\| \leq 2\|\Sigma((\nabla \mathcal{L}_i(\boldsymbol{x}^*)_{i \in \mathcal{H}})\| + 2\Delta^2 \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 .$$

Using Equation (6.12), written here as $\|\boldsymbol{x} - \boldsymbol{x}^*\|^2 \leq \frac{1}{\mu^2}\|\nabla \mathcal{L}(\boldsymbol{x})\|^2$, and using Definition 6.C.1 of $(f, \nu, \|\cdot\|)$ aggregators gives the result.

□

6.D Convergence of Algorithm 4 and Algorithm 5

In this section, we demonstrate the convergence of Algorithm 4 and Algorithm 5. For the sake of generality, we consider the following problem:

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} \{h(\boldsymbol{x}) = p(\boldsymbol{x}) + q(\boldsymbol{x})\}.$$

The guarantees of the Byzantine case then derives from considering $h = \mathcal{L}_{\mathcal{H}}$, $p = \mathcal{L}_{\mathcal{H}} - \hat{\mathcal{L}}$ and $q = \hat{\mathcal{L}}$.

6.D.1 Introduction of Standard Optimization tools

In the proof, we use several standard optimization tools that we recall hereafter.

Definition 6.D.1 (Bregman Divergence)**.** The Bregman divergence of a real-valued function h is defined as

$$D_h(\boldsymbol{x}, \boldsymbol{y}) := h(\boldsymbol{x}) - h(\boldsymbol{y}) - \langle \nabla h(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle.$$

The Bregman divergence has the following properties:

Proposition 6.D.2. *1. A function h is μ-strongly convex and L-smooth, if and only if*

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, \qquad \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 \leq D_h(\boldsymbol{x}; \boldsymbol{y}) \leq \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2. \tag{6.10}$$

2. ***Three points identity.*** *Let $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^d$, it holds:*

$$D_h(\boldsymbol{x}; \boldsymbol{y}) + D_h(\boldsymbol{y}; \boldsymbol{z}) - D_h(\boldsymbol{x}; \boldsymbol{z}) = \langle \nabla h(\boldsymbol{y}) - \nabla h(\boldsymbol{z}), \boldsymbol{y} - \boldsymbol{x} \rangle. \tag{6.11}$$

Furthermore, we will use the following property of strongly convex functions:

Proposition 6.D.3. *Let $h : \mathbb{R}^d \to \mathbb{R}$ be a μ-strongly convex L-smooth loss function, then:*

$$\langle \nabla h(\boldsymbol{x}) - \nabla h(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle \leq \frac{1}{\mu} \|\nabla h(\boldsymbol{x}) - \nabla h(\boldsymbol{y})\|^2. \tag{6.12}$$

Proof. We use that the Fenchel transform of h, defined as $h^* : \boldsymbol{u} \to \sup_{\boldsymbol{x} \in \mathbb{R}^d} \{\langle \boldsymbol{x}, \boldsymbol{u} \rangle - h(\boldsymbol{x})\}$, is $1/\mu$ smooth. This ensures that

Then, using Fenchel's identity, which here can be written as $\nabla h = (\nabla h^*)^{-1}$, and considering $\boldsymbol{x} = \nabla h^*(\boldsymbol{u})$ and $\boldsymbol{y} = \nabla h^*(\boldsymbol{v})$ yields the result. □

6.D.2 Convergence of PIGS (Algorithm 4)

We show the following theorem, of which Theorem 6.3.7 is a direct consequence.

Theorem 6.D.4. *Let $h : \mathbb{R}^d \to \mathbb{R}$ be a μ-strongly convex and L_h-smooth loss function, written as $h = p + q$, where p is L_p-smooth (i.e. $\|\nabla^2 p(\boldsymbol{x})\|_{op} \leq L_p$). Assume that we have access to an inexact gradient oracle $\tilde{\nabla} p$ satisfying*

$$\forall \boldsymbol{x} \in \mathbb{R}^d, \quad \|\tilde{\nabla} p(\boldsymbol{x}) - \nabla p(\boldsymbol{x})\|^2 \leq \zeta^2 + \alpha \mu^2 \|\boldsymbol{x} - \boldsymbol{x}^*\|^2, \tag{6.13}$$

where $\boldsymbol{x}^* := \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} h(\boldsymbol{x})$ and $\alpha < 1/8$. Consider $(\boldsymbol{x}_k)_{k \geq 0}$ such that for any k, given

$$\phi_\eta^k(\boldsymbol{x}) := \langle \tilde{\nabla} p(\boldsymbol{x}_k) + \nabla q(\boldsymbol{x}_k), \boldsymbol{x} \rangle + D_q(\boldsymbol{x}; \boldsymbol{x}_k) + \frac{1}{2\eta} \|\boldsymbol{x} - \boldsymbol{x}_k\|^2, \tag{6.14}$$

$\boldsymbol{x}_{k+1}$ is chosen such that

$$\|\nabla \phi_\eta^k(\boldsymbol{x}_{k+1})\|^2 \leq c \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 + E^2. \tag{6.15}$$

Denote $s_k = (1 + \frac{\eta\mu}{3})^k$, and $S_k = \sum_{k=0}^K s_k$. Then for $\eta \leq \frac{1}{L_p + 4c/\mu}$, and $\hat{\boldsymbol{x}}_k := \frac{1}{S_k} \sum_{k=0}^K s_k \boldsymbol{x}_k$, we have

$$h(\hat{\boldsymbol{x}}_K) - h(\boldsymbol{x}^*) \leq \left(1 + \frac{\eta\mu}{3}\right)^{1-K} \frac{2}{\eta(1 - 4\alpha)} \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 + 2 \frac{\zeta^2 + E^2}{\mu(1 - 4\alpha)}.$$

Note that we did lose some constant factors for the sake of clarity.

6.D.2.1 Proof of Theorem 6.D.4

This proof is an adaptation of the proof of Woodworth et al. (2023), in which we modify the Lyapunov function to allow for the multiplicative error term (i.e $\alpha \neq 0$) in the inexact gradient assumption.

We first show the following decomposition

Lemma 6.D.5. *Denote* $\boldsymbol{e}_k := \nabla p(\boldsymbol{x}_k) - \tilde{\nabla} p(\boldsymbol{x}_k)$*, such that, by definition* $\|\boldsymbol{e}_k\|^2 \leq \zeta^2 + \alpha\mu\langle \nabla h(\boldsymbol{x}_k), \boldsymbol{x}_k - \boldsymbol{x}^*\rangle$*. Then the following identity holds:*

$$\begin{aligned} h(\boldsymbol{x}_{k+1}) - h(\boldsymbol{x}^*) = &\left(\frac{1}{2\eta}\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 - D_p(\boldsymbol{x}^*; \boldsymbol{x}_k)\right) \\ &- \left(\frac{1}{2\eta}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 - D_p(\boldsymbol{x}^*; \boldsymbol{x}_{k+1})\right) \\ &- D_h(\boldsymbol{x}^*; \boldsymbol{x}_{k+1}) \\ &+ \langle \nabla \phi_\eta^k(\boldsymbol{x}_{k+1}) + \boldsymbol{e}_k, \boldsymbol{x}_{k+1} - \boldsymbol{x}^*\rangle \\ &+ D_p(\boldsymbol{x}_{k+1}; \boldsymbol{x}_k) - \frac{1}{2\eta}\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|^2. \end{aligned}$$

The first and second line are telescopic; the third line will be leveraged to ensure a strict decrease in the Lyapunov value; the fourth line corresponds to the error due to the approximate proximal step and the erroneous gradients. Eventually, the last line corresponds to a discretization error.

Proof.

Computing $\nabla \phi_\eta^k(\boldsymbol{x}_{k+1})$ using Equation (6.14) and $\nabla_{\boldsymbol{x}} D_q(\boldsymbol{x}, \boldsymbol{x}_k) = \nabla q(\boldsymbol{x}) - \nabla q(\boldsymbol{x}_k)$, defining $\boldsymbol{e}_k = \nabla p(\boldsymbol{x}_k) - \tilde{\nabla} p(\boldsymbol{x}_k)$ the error in the computation of $\nabla p(\boldsymbol{x}_k)$, and rearranging terms yields

$$\begin{aligned} \nabla \phi_\eta^k(\boldsymbol{x}_{k+1}) &= \tilde{\nabla} p(\boldsymbol{x}_k) + \nabla q(\boldsymbol{x}_k) + \nabla q(\boldsymbol{x}_{k+1}) - \nabla q(\boldsymbol{x}_k) + \frac{1}{\eta}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \\ &= \underbrace{\tilde{\nabla} p(\boldsymbol{x}_k) - \nabla p(\boldsymbol{x}_k)}_{=-\boldsymbol{e}_k} + \nabla p(\boldsymbol{x}_k) - \nabla p(\boldsymbol{x}_{k+1}) \\ &\quad + \underbrace{\nabla p(\boldsymbol{x}_{k+1}) + \nabla q(\boldsymbol{x}_{k+1})}_{=\nabla h(\boldsymbol{x}_{k+1})} + \frac{1}{\eta}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \\ \nabla h(\boldsymbol{x}_{k+1}) &= \nabla p(\boldsymbol{x}_{k+1}) - \nabla p(\boldsymbol{x}_k) + \nabla \phi_\eta^k(\boldsymbol{x}_{k+1}) + \boldsymbol{e}_k + \frac{1}{\eta}(\boldsymbol{x}_k - \boldsymbol{x}_{k+1}) \ . \end{aligned}$$

Furthermore, the definition of the Bregman divergence implies

$$h(\boldsymbol{x}_{k+1}) - h(\boldsymbol{x}^*) = \langle \nabla h(\boldsymbol{x}_{k+1}), \boldsymbol{x}_{k+1} - \boldsymbol{x}^*\rangle - D_h(\boldsymbol{x}^*; \boldsymbol{x}_{k+1}).$$

Substituting the previous identity into the latter one yields

$$\begin{aligned} &h(\boldsymbol{x}_{k+1}) - h(\boldsymbol{x}^*) \\ &\quad = \langle \nabla h(\boldsymbol{x}_{k+1}), \boldsymbol{x}_{k+1} - \boldsymbol{x}^*\rangle - D_h(\boldsymbol{x}^*; \boldsymbol{x}_{k+1}) \\ &\quad = \left\langle \nabla p(\boldsymbol{x}_{k+1}) - \nabla p(\boldsymbol{x}_k) + \nabla \phi_\eta^k(\boldsymbol{x}_{k+1}) + \boldsymbol{e}_k + \frac{1}{\eta}(\boldsymbol{x}_k - \boldsymbol{x}_{k+1}), \boldsymbol{x}_{k+1} - \boldsymbol{x}^* \right\rangle \\ &\qquad - D_h(\boldsymbol{x}^*; \boldsymbol{x}_{k+1}) \\ &\quad = D_p(\boldsymbol{x}^*; \boldsymbol{x}_{k+1}) + D_p(\boldsymbol{x}_{k+1}; \boldsymbol{x}_k) - D_p(\boldsymbol{x}^*; \boldsymbol{x}_k) \\ &\qquad + \langle \nabla \phi_\eta^k(\boldsymbol{x}_{k+1}) + \boldsymbol{e}_k, \boldsymbol{x}_{k+1} - \boldsymbol{x}^*\rangle \\ &\qquad - D_h(\boldsymbol{x}^*; \boldsymbol{x}_{k+1}) + \frac{1}{2\eta}\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 - \frac{1}{2\eta}\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|^2 - \frac{1}{2\eta}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2. \end{aligned}$$

Where we used the parallelogram identity,

$$\frac{1}{\eta}\langle \boldsymbol{x}_k - \boldsymbol{x}_{k+1}, \boldsymbol{x}_{k+1} - \boldsymbol{x}^*\rangle = \frac{1}{2\eta}\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 - \frac{1}{2\eta}\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|^2 - \frac{1}{2\eta}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2,$$

and the three-point identity Equation (6.11),

$$D_p(\boldsymbol{x}^*; \boldsymbol{x}_{k+1}) + D_p(\boldsymbol{x}_{k+1}; \boldsymbol{x}_k) - D_p(\boldsymbol{x}^*; \boldsymbol{x}_k) = \langle \nabla p(\boldsymbol{x}_{k+1}) - \nabla p(\boldsymbol{x}_k), \boldsymbol{x}_{k+1} - \boldsymbol{x}^*\rangle.$$

□

We now control the error due to the inexact oracle and the approximate proximal step as follows:

Lemma 6.D.6. *For any $\varepsilon > 1$, the following inequality holds, for any $k \geq 0$,*

$$\begin{aligned}&\langle \nabla \phi_\eta^k(\boldsymbol{x}_{k+1}) + \boldsymbol{e}_k, \boldsymbol{x}_{k+1} - \boldsymbol{x}^*\rangle \\ &\leq \frac{c}{2\varepsilon}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 + \frac{\alpha\mu^2}{2\varepsilon}\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 + \varepsilon\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 + \frac{E^2 + \zeta^2}{2\varepsilon}.\end{aligned}$$

Proof. Writing the inexact gradient assumption (Equation (6.13)) and Young's inequality, we have for any $\varepsilon > 0$,

$$\langle \boldsymbol{e}_k, \boldsymbol{x}_{k+1} - \boldsymbol{x}^*\rangle \leq \frac{1}{2\varepsilon}\left(\zeta^2 + \alpha\mu^2\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2\right) + \frac{\varepsilon}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2. \tag{6.16}$$

Using the assumption Equation (6.15) on the proximal step, namely

$$\|\nabla \phi_\eta^k(\boldsymbol{x}_{k+1}\|^2 \leq c\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 + E^2,$$

and Young's inequality, we have that

$$\langle \nabla \phi_\eta^k(\boldsymbol{x}_{k+1}), \boldsymbol{x}_{k+1} - \boldsymbol{x}^*\rangle \leq \frac{1}{2\varepsilon}\left(c\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 + E^2\right) + \frac{\varepsilon}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2. \tag{6.17}$$

Putting Equation (6.16) and Equation (6.17) together gives Lemma 6.D.6. □

We now define a Lyapunov energy function and show its decrease.

Lemma 6.D.7. *Let's consider the Lyapunov*

$$\Gamma_k := \left(\frac{1}{2\eta} + \frac{\alpha\mu^2}{2\varepsilon}\right)\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 - D_p(\boldsymbol{x}^*; \boldsymbol{x}_k). \tag{6.18}$$

Then, under the choice $\varepsilon = \frac{\mu}{4}$ and $\eta \leq 1/(L_p + 4c/\mu)$, we have

$$(1 - 4\alpha)\left(h(\boldsymbol{x}_{k+1}) - h(\boldsymbol{x}^*)\right) \leq \Gamma_k - \left(1 + \frac{\eta\mu}{3}\right)\Gamma_{k+1} + 2\frac{E^2 + \zeta^2}{\mu}.$$

Proof. Combining Lemmas 6.D.5 and 6.D.6, gives

$$\begin{aligned} h(\boldsymbol{x}_{k+1}) - h(\boldsymbol{x}^*) \leq & \left(\frac{1}{2\eta}\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 - D_p(\boldsymbol{x}^*; \boldsymbol{x}_k)\right) \\ & - \left(\frac{1}{2\eta}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 - D_p(\boldsymbol{x}^*; \boldsymbol{x}_{k+1})\right) \\ & - D_h(\boldsymbol{x}^*; \boldsymbol{x}_{k+1}) \\ & + \frac{c}{2\varepsilon}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 + \frac{\alpha\mu^2}{2\varepsilon}\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 + \varepsilon\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 + \frac{E^2 + \zeta^2}{2\varepsilon} \\ & + D_p(\boldsymbol{x}_{k+1}; \boldsymbol{x}_k) - \frac{1}{2\eta}\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|^2. \end{aligned}$$

Thus, removing on both sides $\frac{\alpha\mu^2}{2\varepsilon}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2$ and rearranging yields

$$\begin{aligned} (h(\boldsymbol{x}_{k+1}) - h(\boldsymbol{x}^*)) - \frac{\alpha\mu^2}{2\varepsilon}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 & \\ \leq & \left(\left(\frac{1}{2\eta} + \frac{\alpha\mu^2}{2\varepsilon}\right)\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 - D_p(\boldsymbol{x}^*; \boldsymbol{x}_k)\right) \\ & - \left(\left(\frac{1}{2\eta} + \frac{\alpha\mu^2}{2\varepsilon}\right)\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 - D_p(\boldsymbol{x}^*; \boldsymbol{x}_{k+1})\right) \\ & - D_h(\boldsymbol{x}^*; \boldsymbol{x}_{k+1}) \\ & + \frac{c}{2\varepsilon}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 + \varepsilon\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 + \frac{E^2 + \zeta^2}{2\varepsilon} \\ & + D_p(\boldsymbol{x}_{k+1}; \boldsymbol{x}_k) - \frac{1}{2\eta}\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|^2. \end{aligned}$$

Which, using the definition of Γ_k (Equation (6.18)), yields

$$\begin{aligned} h(\boldsymbol{x}_{k+1}) - h(\boldsymbol{x}^*) - \frac{\alpha\mu^2}{2\varepsilon}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 \leq & \Gamma_k - \Gamma_{k+1} - D_h(\boldsymbol{x}^*; \boldsymbol{x}_{k+1}) \\ & + \frac{c}{2\varepsilon}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 + \varepsilon\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 \\ & + \frac{E^2 + \zeta^2}{2\varepsilon} \\ & + D_p(\boldsymbol{x}_{k+1}; \boldsymbol{x}_k) - \frac{1}{2\eta}\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|^2. \end{aligned} \tag{6.19}$$

We now use the regularity property of h and p to derive the following inequalities:

$$D_h(\boldsymbol{x}^*; \boldsymbol{x}_{k+1}) \geq \frac{\mu}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2, \qquad \text{since } h \text{ is } \mu\text{-strongly convex} \tag{6.20}$$

$$h(\boldsymbol{x}_{k+1}) - h(\boldsymbol{x}^*) \geq \frac{\mu}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|, \qquad \text{since } \nabla h(\boldsymbol{x}^*) = 0 \tag{6.21}$$

$$|D_p(\boldsymbol{x}^*; \boldsymbol{x}_{k+1})| \leq \frac{L_p}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2, \qquad \text{since } p \text{ is } L_p\text{-smooth.} \tag{6.22}$$

And, using Equation (6.22), we have, for $\eta^{-1} \geq L_p$,

$$0 \leq \frac{1}{2\eta}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 - D_p(\boldsymbol{x}^*, \boldsymbol{x}_{k+1}) \leq \eta^{-1}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2.$$

Which yields,

$$0 \leq \Gamma_{k+1} \leq \left(\eta^{-1} + \frac{\alpha\mu^2}{2\varepsilon}\right)\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 \tag{6.23}$$

Combining Equation (6.20) and Equation (6.23) yields

$$D_h(\boldsymbol{x}^*; \boldsymbol{x}_{k+1}) \geq \frac{\mu}{4}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 + \frac{\mu}{2}\left(\eta^{-1} + \frac{\alpha\mu^2}{2\varepsilon}\right)^{-1}\Gamma_{k+1}. \tag{6.24}$$

Additionally, Equation (6.21) implies that

$$h(\boldsymbol{x}_{k+1}) - h(\boldsymbol{x}^*) - \frac{\alpha\mu^2}{2\varepsilon}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 \geq \left(1 - \frac{\alpha\mu}{\varepsilon}\right)(h(\boldsymbol{x}_{k+1}) - h(\boldsymbol{x}^*)) \tag{6.25}$$

We can now use Equation (6.22), Equation (6.24) and Equation (6.25) into Equation (6.19).

$$\begin{aligned}
&\left(1 - \frac{\alpha\mu}{\varepsilon}\right)(h(\boldsymbol{x}_{k+1}) - h(\boldsymbol{x}^*)) \\
&\qquad \leq \Gamma_k - \Gamma_{k+1} \\
&\qquad - \frac{\mu}{4}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 - \frac{\mu}{2}\left(\eta^{-1} + \frac{\alpha\mu^2}{2\varepsilon}\right)^{-1}\Gamma_{k+1} \\
&\qquad + \frac{c}{2\varepsilon}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 + \varepsilon\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 + \frac{E^2 + \zeta^2}{2\varepsilon} \\
&\qquad + D_p(\boldsymbol{x}_{k+1}; \boldsymbol{x}_k) - \frac{1}{2\eta}\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|^2.
\end{aligned}$$

Using the smoothness of p and rearranging yields

$$\begin{aligned}
&\left(1 - \frac{\alpha\mu}{\varepsilon}\right)(h(\boldsymbol{x}_{k+1}) - h(\boldsymbol{x}^*)) \\
&\qquad \leq \Gamma_k - \Gamma_{k+1} \\
&\qquad \left(\varepsilon - \frac{\mu}{4}\right)\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 - \frac{\mu}{2}\left(\eta^{-1} + \frac{\alpha\mu^2}{2\varepsilon}\right)^{-1}\Gamma_{k+1} \\
&\qquad + \left(\frac{L_p}{2} + \frac{c}{2\varepsilon} - \frac{1}{2\eta}\right)\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|^2 \\
&\qquad + \frac{E^2 + \zeta^2}{2\varepsilon}.
\end{aligned}$$

Choosing $\varepsilon = \frac{\mu}{4}$ and enforcing $\eta \leq 1/(L_p + \frac{4c}{\mu})$ yields

$$(1 - 4\alpha)(h(\boldsymbol{x}_{k+1}) - h(\boldsymbol{x}^*)) \leq \Gamma_k - \left(1 + \frac{\eta\mu}{2 + 4\alpha\eta\mu}\right)\Gamma_{k+1} + 2\frac{E^2 + \zeta^2}{\mu}.$$

Which, considering that for $\alpha < 1/4$, and $\mu < L_p$, we have

$$\frac{\eta\mu}{2 + 4\alpha\eta\mu} > \frac{\eta\mu}{3},$$

gives the desired result. □

We can now prove Theorem 6.D.4.

Proof. We denote $s_k := \left(1+\frac{\eta\mu}{3}\right)^{k-1}$ and $S_k := \sum_{k=1}^K s_k$ and considering

$$\hat{\boldsymbol{x}}_K := \frac{1}{S_k}\sum_{k=1}^K s_k \boldsymbol{x}_k,$$

allows us to write, by convexity of h,

$$\begin{aligned}(1-4\alpha)h(\hat{\boldsymbol{x}}_K) - h^* &\leq \frac{1-4\alpha}{S_k}\sum_{k=1}^K s_k(h(\boldsymbol{x}_k)-h^*)\\ &\leq \left(\frac{\Gamma_0}{S_k} + \frac{2\zeta^2}{\mu} + \frac{2E^2}{\mu}\right)\end{aligned}$$

using Lemma 6.D.7.

Now, using Equation (6.23), we have that

$$\Gamma_0 \leq (\frac{1}{\eta}+2\alpha\mu)\|\boldsymbol{x}_0-\boldsymbol{x}^*\|^2 \leq \frac{2}{\eta}\|\boldsymbol{x}_0-\boldsymbol{x}^*\|^2,$$

and considering that

$$\begin{aligned}S_k &= \frac{(1+\frac{\eta\mu}{3})^K - 1}{\eta\mu/3} = \frac{(1+\frac{\eta\mu}{3})^{K-1}\frac{\eta\mu}{3} + (1+\frac{\eta\mu}{3})^{K-1} - 1}{\eta\mu/3}\\ &\leq \left(1+\frac{\eta\mu}{3}\right)^{K-1},\end{aligned}$$

we have

$$\begin{aligned}h(\hat{\boldsymbol{x}}_K) - h(\boldsymbol{x}^*) &\leq \left(1+\frac{\eta\mu}{3}\right)^{1-K}\frac{\Gamma_0}{1-4\alpha} + 2\frac{\zeta^2+E^2}{\mu(1-4\alpha)}\\ h(\hat{\boldsymbol{x}}_K) - h(\boldsymbol{x}^*) &\leq \left(1+\frac{\eta\mu}{3}\right)^{1-K}\frac{2}{\eta(1-4\alpha)}\|\boldsymbol{x}_0-\boldsymbol{x}^*\|^2 + 2\frac{\zeta^2+E^2}{\mu(1-4\alpha)}.\end{aligned}$$

Which concludes the proof. □

6.D.3 Convergence of the Accelerated Inexact Extra-gradient Method (Algorithm 5)

Algorithm 8 Accelerated Inexact Extra-gradient

Input: $\boldsymbol{x}^0 = \boldsymbol{x}_f^0 \in \mathbb{R}^d$
Parameters: $\tau \in (0,1)$, $\eta, \theta, \gamma > 0$, $K \in \{1,2,\ldots\}$
for $k = 0, 1, 2, \ldots, K-1$ **do**
  $\boldsymbol{x}_g^k = \tau\boldsymbol{x}^k + (1-\tau)\boldsymbol{x}_f^k$
  $\boldsymbol{x}_f^{k+1} \approx \arg\min_{\boldsymbol{x}\in\mathbb{R}^d}\left[\phi_\theta^k(\boldsymbol{x}) := p(\boldsymbol{x}_g^k) + \langle\tilde{\nabla}p(\boldsymbol{x}_g^k), x-\boldsymbol{x}_g^k\rangle + \frac{1}{2\theta}\left\|\boldsymbol{x}-\boldsymbol{x}_g^k\right\|^2 + q(\boldsymbol{x})\right]$
  $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \eta\gamma(\boldsymbol{x}_f^{k+1}-\boldsymbol{x}^k) - \eta\tilde{\nabla}h(\boldsymbol{x}_f^{k+1})$
end for
Output: $\boldsymbol{x}^K$

Lemma 6.D.8. *Assume h is μ-strongly convex, and p and q are respectively L_p and L_q smooth. Consider Algorithm 8 with $\theta = \frac{1}{2\sqrt{2}L_p}$, the following inequality holds for all $\bar{\boldsymbol{x}} \in \mathbb{R}^d$*

$$\begin{aligned} 2\langle \bar{\boldsymbol{x}} - \boldsymbol{x}_g^k, \nabla h(\boldsymbol{x}_f^{k+1})\rangle \leq & 2\left[h(\bar{\boldsymbol{x}}) - h(\boldsymbol{x}_f^{k+1})\right] - \mu \left\|\boldsymbol{x}_f^{k+1} - \bar{\boldsymbol{x}}\right\|^2 - \theta \left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2 \\ & + 2\theta\left(\left\|\nabla \phi_\theta^k(\boldsymbol{x}_f^{k+1})\right\|^2 - \frac{1}{4\theta^2}\left\|\boldsymbol{x}_f^{k+1} - \boldsymbol{x}_g^k\right\|^2\right) \\ & + 4\theta(\zeta^2 + \alpha\left\|\nabla h(\boldsymbol{x}_g^k)\right\|^2). \end{aligned} \tag{6.26}$$

Proof. Using μ-strong convexity of $h(\boldsymbol{x})$, we get:

$$\begin{aligned} 2\langle \bar{\boldsymbol{x}} - \boldsymbol{x}_g^k, \nabla h(\boldsymbol{x}_f^{k+1})\rangle &= 2\langle \bar{\boldsymbol{x}} - \boldsymbol{x}_f^{k+1}, \nabla h(\boldsymbol{x}_f^{k+1})\rangle + 2\langle \boldsymbol{x}_f^{k+1} - \boldsymbol{x}_g^k, \nabla h(\boldsymbol{x}_f^{k+1})\rangle \\ &\leq 2\left[h(\bar{\boldsymbol{x}}) - h(\boldsymbol{x}_f^{k+1})\right] - \mu\left\|\boldsymbol{x}_f^{k+1} - \bar{\boldsymbol{x}}\right\|^2 \\ &\quad + 2\langle \boldsymbol{x}_f^{k+1} - \boldsymbol{x}_g^k, \nabla h(\boldsymbol{x}_f^{k+1})\rangle \\ &= 2\left[h(\bar{\boldsymbol{x}}) - h(\boldsymbol{x}_f^{k+1})\right] - \mu\left\|\boldsymbol{x}_f^{k+1} - \bar{\boldsymbol{x}}\right\|^2 \\ &\quad + 2\theta\langle \theta^{-1}(\boldsymbol{x}_f^{k+1} - \boldsymbol{x}_g^k), \nabla h(\boldsymbol{x}_f^{k+1})\rangle \\ &= 2\left[h(\bar{\boldsymbol{x}}) - h(\boldsymbol{x}_f^{k+1})\right] - \mu\left\|\boldsymbol{x}_f^{k+1} - \bar{\boldsymbol{x}}\right\|^2 \\ &\quad - \frac{1}{\theta}\left\|\boldsymbol{x}_f^{k+1} - \boldsymbol{x}_g^k\right\|^2 - \theta\left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2 \\ &\quad + \theta\left\|\theta^{-1}(\boldsymbol{x}_f^{k+1} - \boldsymbol{x}_g^k) + \nabla h(\boldsymbol{x}_f^{k+1})\right\|^2. \end{aligned}$$

The definition of $\phi_\theta^k(\boldsymbol{x})$, and the L_p-Lipschitzness of ∇p give:

$$\begin{aligned} 2\langle \bar{\boldsymbol{x}} - \boldsymbol{x}_g^k, \nabla h(\boldsymbol{x}_f^{k+1})\rangle &\leq 2\left[h(\bar{\boldsymbol{x}}) - h(\boldsymbol{x}_f^{k+1})\right] - \mu\left\|\boldsymbol{x}_f^{k+1} - \bar{\boldsymbol{x}}\right\|^2 \\ &\quad - \frac{1}{\theta}\left\|\boldsymbol{x}_f^{k+1} - \boldsymbol{x}_g^k\right\|^2 - \theta\left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2 \\ &\quad + \theta\left\|\nabla \phi_\theta^k(\boldsymbol{x}_f^{k+1}) + \nabla p(\boldsymbol{x}_f^{k+1}) - \nabla p(\boldsymbol{x}_g^k) + \nabla p(\boldsymbol{x}_g^k) - \tilde{\nabla} p(\boldsymbol{x}_g^k)\right\|^2 \\ &\leq 2\left[h(\bar{\boldsymbol{x}}) - h(\boldsymbol{x}_f^{k+1})\right] - \mu\left\|\boldsymbol{x}_f^{k+1} - \bar{\boldsymbol{x}}\right\|^2 \\ &\quad - \frac{1}{\theta}\left\|\boldsymbol{x}_f^{k+1} - \boldsymbol{x}_g^k\right\|^2 - \theta\left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2 \\ &\quad + 2\theta\left\|\nabla \phi_\theta^k(\boldsymbol{x}_f^{k+1})\right\|^2 + 4\theta L_p^2\left\|\boldsymbol{x}_f^{k+1} - \boldsymbol{x}_g^k\right\|^2 \\ &\quad + 4\theta\|\tilde{\nabla} p(\boldsymbol{x}_g^k) - \nabla p(\boldsymbol{x}_g^k)\|^2 \\ &= 2\left[h(\bar{\boldsymbol{x}}) - h(\boldsymbol{x}_f^{k+1})\right] - \mu\left\|\boldsymbol{x}_f^{k+1} - \bar{\boldsymbol{x}}\right\|^2 \\ &\quad - \frac{1}{\theta}\left(1 - 4\theta^2 L_p^2\right)\left\|\boldsymbol{x}_f^{k+1} - \boldsymbol{x}_g^k\right\|^2 - \theta\left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2 \\ &\quad + 2\theta\left\|\nabla \phi_\theta^k(\boldsymbol{x}_f^{k+1})\right\|^2 + 4\theta(\zeta^2 + \alpha\|\nabla h(\boldsymbol{x}_g^k)\|^2) \end{aligned}$$

With $\theta = \frac{1}{2\sqrt{2}L_p}$, we have

$$2\langle \bar{\boldsymbol{x}} - \boldsymbol{x}_g^k, \nabla h(\boldsymbol{x}_f^{k+1})\rangle \leq 2\left[h(\bar{\boldsymbol{x}}) - h(\boldsymbol{x}_f^{k+1})\right] - \mu\left\|\boldsymbol{x}_f^{k+1} - \bar{\boldsymbol{x}}\right\|^2 - \frac{1}{2\theta}\left\|\boldsymbol{x}_f^{k+1} - \boldsymbol{x}_g^k\right\|^2 - \theta\left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2 + 2\theta\left\|\nabla \phi_\theta^k(\boldsymbol{x}_f^{k+1})\right\|^2 + 4\theta(\zeta^2 + \alpha\left\|\nabla h(\boldsymbol{x}_g^k)\right\|^2).$$

This completes the proof of the lemma. □

Theorem 6.D.9 (Theorem 6.4.1). *Assume that $h = p + q$ is μ-strongly convex and L_h-smooth, that p and q are respectively L_p and L_q-smooth, and that $\alpha < \min\left(\frac{L_p^2}{4L_h^2}, \frac{1}{16}\right)$. Under the parametrization*

$$\gamma = \mu, \quad \theta = \frac{1}{2\sqrt{2}L_p}, \quad \eta = \min\left(\frac{1}{2\mu}, \frac{\theta}{24\tau}\right), \quad \text{and} \quad \tau = \min\left(\frac{\theta\mu}{16\alpha}, \sqrt{\frac{\theta\mu}{48}}\right),$$

Algorithm 8 with $\left\|\nabla \phi_\theta^k(\boldsymbol{x}_f^{k+1})\right\|^2 \leq \frac{1}{8\theta^2}\left\|\boldsymbol{x}_f^{k+1} - \boldsymbol{x}_g^k\right\|^2$, yields

$$h(\boldsymbol{x}_f^K) - h(\boldsymbol{x}^*) \leq O\Big((1-\rho)^K C + \frac{\zeta^2}{\mu}\Big),$$

where $\rho = O\left(\min\left(\frac{1}{\sqrt{\kappa}}, \frac{\mu}{\alpha L_p}\right)\right)$ and $R = \tau^2 L_p\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2 + h(\boldsymbol{x}^0) - h(\boldsymbol{x}^*)$.

Proof. Using line 6 of Algorithm 8, we get

$$\begin{aligned}
\frac{1}{\eta}\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\right\|^2 &= \frac{1}{\eta}\left\|\boldsymbol{x}^k - \boldsymbol{x}^*\right\|^2 + \frac{2}{\eta}\langle \boldsymbol{x}^{k+1} - \boldsymbol{x}^k, \boldsymbol{x}^k - \boldsymbol{x}^*\rangle + \frac{1}{\eta}\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\right\|^2 \\
&= \frac{1}{\eta}\left\|\boldsymbol{x}^k - \boldsymbol{x}^*\right\|^2 + 2\gamma\langle \boldsymbol{x}_f^{k+1} - \boldsymbol{x}^k, \boldsymbol{x}^k - \boldsymbol{x}^*\rangle \\
&\quad - 2\langle \nabla h(\boldsymbol{x}_f^{k+1}), \boldsymbol{x}^k - \boldsymbol{x}^*\rangle + \frac{1}{\eta}\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\right\|^2 \\
&\quad + 2\langle \nabla h(\boldsymbol{x}_f^{k+1}) - \tilde{\nabla} h(\boldsymbol{x}_f^{k+1}), \boldsymbol{x}^k - \boldsymbol{x}^*\rangle \\
&\leq \frac{1}{\eta}\left\|\boldsymbol{x}^k - \boldsymbol{x}^*\right\|^2 + \gamma\left\|\boldsymbol{x}_f^{k+1} - \boldsymbol{x}^*\right\|^2 - \gamma\left\|\boldsymbol{x}^k - \boldsymbol{x}^*\right\|^2 - \gamma\left\|\boldsymbol{x}_f^{k+1} - \boldsymbol{x}^k\right\|^2 \\
&\quad - 2\langle \nabla h(\boldsymbol{x}_f^{k+1}), \boldsymbol{x}^k - \boldsymbol{x}^*\rangle + \frac{1}{\eta}\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\right\|^2. \\
&\quad + \varepsilon\|\nabla h(\boldsymbol{x}_f^{k+1}) - \tilde{\nabla} h(\boldsymbol{x}_f^{k+1})\|^2 + \varepsilon^{-1}\left\|\boldsymbol{x}^k - \boldsymbol{x}^*\right\|^2
\end{aligned}$$

Line 4 of Algorithm 8 gives

$$\begin{aligned}
\frac{1}{\eta}\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\right\|^2 \leq \left(\frac{1}{\eta} - \gamma + \varepsilon^{-1}\right)\left\|\boldsymbol{x}^k - \boldsymbol{x}^*\right\|^2 + \gamma\left\|\boldsymbol{x}_f^{k+1} - \boldsymbol{x}^*\right\|^2 + \frac{1}{\eta}\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\right\|^2 \\
- \gamma\left\|\boldsymbol{x}_f^{k+1} - \boldsymbol{x}^k\right\|^2 + 2\langle \nabla h(\boldsymbol{x}_f^{k+1}), \boldsymbol{x}^* - \boldsymbol{x}_g^k\rangle \\
+ \frac{2(1-\tau)}{\tau}\langle \nabla h(\boldsymbol{x}_f^{k+1}), \boldsymbol{x}_f^k - \boldsymbol{x}_g^k\rangle + \varepsilon(\zeta^2 + \alpha\left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2).
\end{aligned}$$

Using Lemma 6.D.8 with $\bar{\boldsymbol{x}} = \boldsymbol{x}^*$ and $\bar{\boldsymbol{x}} = \boldsymbol{x}_f^k$, we get

$$\begin{aligned}
\frac{1}{\eta}\left\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^*\right\|^2 \leq & \left(\frac{1}{\eta}-\gamma+\varepsilon^{-1}\right)\left\|\boldsymbol{x}^k-\boldsymbol{x}^*\right\|^2+\gamma\left\|\boldsymbol{x}_f^{k+1}-\boldsymbol{x}^*\right\|^2+\frac{1}{\eta}\left\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^k\right\|^2 \\
& -\gamma\left\|\boldsymbol{x}_f^{k+1}-\boldsymbol{x}^k\right\|^2+2\left[h(\boldsymbol{x}^*)-h(\boldsymbol{x}_f^{k+1})\right]-\mu\left\|\boldsymbol{x}_f^{k+1}-\boldsymbol{x}^*\right\|^2 \\
& +\frac{2(1-\tau)}{\tau}\left[h(\boldsymbol{x}_f^k)-h(\boldsymbol{x}_f^{k+1})\right] \\
& -\frac{\theta}{\tau}\left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2+\frac{2\theta}{\tau}\left(\left\|\nabla\phi_\theta^k(\boldsymbol{x}_f^{k+1})\right\|^2-\frac{1}{4\theta^2}\left\|\boldsymbol{x}_f^{k+1}-\boldsymbol{x}_g^k\right\|^2\right) \\
& +\varepsilon(\zeta^2+\alpha\left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2)+\frac{4\theta}{\tau}(\zeta^2+\alpha\left\|\nabla h(\boldsymbol{x}_g^k)\right\|^2) \\
= & \left(\frac{1}{\eta}-\gamma+\varepsilon^{-1}\right)\left\|\boldsymbol{x}^k-\boldsymbol{x}^*\right\|^2+(\gamma-\mu)\left\|\boldsymbol{x}_f^{k+1}-\boldsymbol{x}^*\right\|^2 \\
& -\gamma\left\|\boldsymbol{x}_f^{k+1}-\boldsymbol{x}^k\right\|^2+\frac{1}{\eta}\left\|\eta\gamma(\boldsymbol{x}_f^{k+1}-\boldsymbol{x}^k)-\eta\tilde{\nabla}h(\boldsymbol{x}_f^{k+1})\right\|^2 \\
& -\frac{\theta}{\tau}\left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2+\frac{2(1-\tau)}{\tau}\left[h(\boldsymbol{x}_f^k)-h(\boldsymbol{x}^*)\right] \\
& -\frac{2}{\tau}\left[h(\boldsymbol{x}_f^{k+1})-h(\boldsymbol{x}^*)\right] \\
& +\frac{2\theta}{\tau}\left(\left\|\nabla\phi_\theta^k(\boldsymbol{x}_f^{k+1})\right\|^2-\frac{1}{4\theta^2}\left\|\boldsymbol{x}_f^{k+1}-\boldsymbol{x}_g^k\right\|^2\right) \\
& +(\varepsilon+\frac{4\theta}{\tau})\zeta^2+\varepsilon\alpha\left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2+\frac{4\theta}{\tau}\alpha\left\|\nabla h(\boldsymbol{x}_g^k)\right\|^2
\end{aligned}$$

Rearranging terms yields

$$\begin{aligned}
\frac{1}{\eta}\left\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^*\right\|^2 \leq & \left(\frac{1}{\eta}-\gamma+1/\varepsilon\right)\left\|\boldsymbol{x}^k-\boldsymbol{x}^*\right\|^2+(\gamma-\mu)\left\|\boldsymbol{x}_f^{k+1}-\boldsymbol{x}^*\right\|^2 \\
& +\gamma(2\eta\gamma-1)\left\|\boldsymbol{x}_f^{k+1}-\boldsymbol{x}^k\right\|^2+\left(2\eta+\varepsilon\alpha-\frac{\theta}{\tau}\right)\left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2 \\
& +\eta(\zeta^2+\alpha\left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2) \\
& +\frac{2(1-\tau)}{\tau}\left[h(\boldsymbol{x}_f^k)-h(\boldsymbol{x}^*)\right]-\frac{2}{\tau}\left[h(\boldsymbol{x}_f^{k+1})-h(\boldsymbol{x}^*)\right] \\
& +\frac{2\theta}{\tau}\left(\left\|\nabla\phi_\theta^k(\boldsymbol{x}_f^{k+1})\right\|^2-\frac{1}{4\theta^2}\left\|\boldsymbol{x}_f^{k+1}-\boldsymbol{x}_g^k\right\|^2\right) \\
& +(\varepsilon+\frac{4\theta}{\tau})\zeta^2+\frac{4\theta}{\tau}\alpha\left\|\nabla h(\boldsymbol{x}_g^k)\right\|^2.
\end{aligned}$$

Plugging in

$$\begin{aligned}
\left\|\nabla h(\boldsymbol{x}_g^k)\right\|^2 &\leq 2\left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2+2\left\|\nabla h(\boldsymbol{x}_f^{k+1})-\nabla h(\boldsymbol{x}_g^k)\right\|^2 \\
&\leq 2\left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2+2L_h^2\left\|\boldsymbol{x}_f^{k+1}-\boldsymbol{x}_g^k\right\|^2,
\end{aligned}$$

yields

$$\begin{aligned}
\frac{1}{\eta}\left\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^*\right\|^2 \leq & \left(\frac{1}{\eta}-\gamma+1/\varepsilon\right)\left\|\boldsymbol{x}^k-\boldsymbol{x}^*\right\|^2+(\gamma-\mu)\left\|\boldsymbol{x}_f^{k+1}-\boldsymbol{x}^*\right\|^2 \\
& +\gamma(2\eta\gamma-1)\left\|\boldsymbol{x}_f^{k+1}-\boldsymbol{x}^k\right\|^2 \\
& +\left(2\eta+(\eta+\varepsilon)\alpha-\frac{\theta}{2\tau}(1-8\alpha)\right)\left\|\nabla h(\boldsymbol{x}_f^{k+1})\right\|^2
\end{aligned}$$

$$+\frac{2(1-\tau)}{\tau}\left[h(\boldsymbol{x}_f^k)-h(\boldsymbol{x}^*)\right]-\frac{2}{\tau}\left[h(\boldsymbol{x}_f^{k+1})-h(\boldsymbol{x}^*)\right]$$
$$+\frac{2\theta}{\tau}\left(\left\|\nabla\phi_\theta^k(\boldsymbol{x}_f^{k+1})\right\|^2-(\frac{1}{4\theta^2}-4L_h^2\alpha)\left\|\boldsymbol{x}_f^{k+1}-\boldsymbol{x}_g^k\right\|^2\right)$$
$$+(\varepsilon+\eta+\frac{4\theta}{\tau})\zeta^2.$$

Considering $\gamma=\mu$, $\varepsilon=\frac{2}{\mu}$, $\eta=\min(\frac{1}{2\mu},\frac{1}{24}\frac{\theta}{\tau})$ and assuming $\alpha<\min\left(\frac{\theta\mu}{16\tau},\frac{1}{32\theta^2L_h^2}\right)$, ensures

$$\gamma(2\eta\gamma-1)<0,\qquad\left(2\eta+(\eta+\varepsilon)\alpha-\frac{\theta}{2\tau}(1-8\alpha)\right)<0,\quad\text{and}\qquad\frac{1}{4\theta^2}-4L_h^2\alpha>\frac{1}{8\theta^2}.$$

Thus, under

$$\left\|\nabla\phi_\theta^k(\boldsymbol{x}_f^{k+1})\right\|^2\leq\frac{1}{8\theta^2}\left\|\boldsymbol{x}_f^{k+1}-\boldsymbol{x}_g^k\right\|^2,$$

we have that

$$\frac{1}{\eta}\left\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^*\right\|^2\leq\left(\frac{1}{\eta}-\frac{\mu}{2}\right)\left\|\boldsymbol{x}^k-\boldsymbol{x}^*\right\|^2$$
$$+\frac{2(1-\tau)}{\tau}\left[h(\boldsymbol{x}_f^k)-h(\boldsymbol{x}^*)\right]-\frac{2}{\tau}\left[h(\boldsymbol{x}_f^{k+1})-h(\boldsymbol{x}^*)\right]$$
$$+(\frac{2}{\mu}+\frac{5\theta}{\tau})\zeta^2.$$

Now, denoting $\rho=\min(\tau,\frac{\eta\mu}{2})$, since $\theta=1/2\sqrt{2}L_p$ the following contraction holds

$$\frac{1}{\eta}\left\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^*\right\|^2+\frac{2}{\tau}\left[h(\boldsymbol{x}_f^{k+1})-h(\boldsymbol{x}^*)-\left(\frac{1}{\mu}+\frac{1}{\tau L_p}\right)\zeta^2\right]$$
$$\leq(1-\rho)\left[\frac{1}{\eta}\left\|\boldsymbol{x}^k-\boldsymbol{x}^*\right\|^2+\frac{2}{\tau}\left[h(\boldsymbol{x}_f^k)-h(\boldsymbol{x}^*)-\left(\frac{1}{\mu}+\frac{1}{\tau L_p}\right)\zeta^2\right]\right].$$

Maximizing $\rho(\tau)=\min\left(\tau,\frac{3}{4},\frac{1}{48}\frac{\theta\mu}{\tau}\right)$, by choosing $\tau=\min\left(\frac{\theta\mu}{16\alpha},\sqrt{\frac{\theta\mu}{48}}\right)$ yields, since $\tau\geq\mu/L_p$,

$$h(\boldsymbol{x}_f^K)-h(\boldsymbol{x}^*)\leq O\Big((1-\rho)^K R+\frac{\zeta^2}{\mu}\Big),$$

where $\rho=O\left(\min\left(\frac{1}{\sqrt{\kappa}},\frac{\mu}{\alpha L_p}\right)\right)$ and $R=\tau^2L_p\|\boldsymbol{x}^0-\boldsymbol{x}^*\|^2+h(\boldsymbol{x}^0)-h(\boldsymbol{x}^*)$. □

6.E From Devolder et al. (2013) acceleration to acceleration for $(\zeta^2,\alpha=0)$-inexact oracles

In the following section, we explain how using the existing notion of inexact gradient oracles (Devolder et al., 2013, 2014) allows us to directly show an accelerated gradient method for Byzantine first-order methods.

6.E.1 Link to First Order (δ, L, μ) Oracles

The study of inexact gradient oracles dates back at least to d'Aspremont (2008). Devolder et al. (2013) introduce the notion of (δ, L, μ)-oracle of a function f.

Definition 6.E.1. $(f_{\delta,L,\mu}, g_{\delta,L,\mu})$ are (δ, L, μ)-oracle of a convex function f if, for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, it satisfies

$$\frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 \leq f(\boldsymbol{x}) - f_{\delta,L,\mu}(\boldsymbol{y}) - \langle g_{\delta,L,\mu}(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y}\rangle \leq \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 + \delta.$$

The class of (δ, L, μ) oracles covers a wide range of scenarios: they appear when the gradients are evaluated at incorrect parameters, when gradients are only Hölder-continuous rather than L-Lipschitz, when the oracle relies on an optimization sub-routine, or when gradients come from of a smoothed version of the original loss (Devolder et al., 2013, 2014). Closer to our interest, (ζ^2, α)-inexact gradient oracles are also (δ, L, μ) oracles when $\alpha = 0$.

Proposition 6.E.2 (Devolder et al. (2014) Section 2.3)**.** ***1.*** *Assume that $(f_{\delta,L,\mu}, g_{\delta,L,\mu})$ is a (δ, L, μ)-oracle of f, and that f is continuous and μ_f strongly convex. Then for $\boldsymbol{x} \in \mathbb{R}^d$,*

$$0 \leq f(\boldsymbol{x}) - f_{\delta,L,\mu}(\boldsymbol{x}) \leq \delta,$$
$$\|\nabla f(\boldsymbol{x}) - g_{\delta,L,\mu}(\boldsymbol{x})\|^2 \leq 2(L - \mu_f)\delta.$$

2. *Conversely, if $\|\nabla f(\boldsymbol{x}) - g(\boldsymbol{x})\|^2 \leq \zeta^2$ and f is L_f-smooth and μ_f strongly convex, then $(f - \zeta^2/\mu_f, g)$ is an $\left((1/2L_f + 1/\mu_f)\zeta^2, 2L_f, \mu_f/2\right)$-oracle of $(f, \nabla f)$.*

The above proposition shows that for strongly convex and smooth losses, using the (δ, L, μ)-oracle formulation is equivalent – up to a factor κ in the inexactness – to the $(\zeta^2, \alpha = 0)$ inexact gradient definition. This allows us to ensure the feasibility of first-order acceleration in distributed optimization under Byzantine corruption.

6.E.2 Convergence Results

Algorithm 9 Inexact Oracle Fast Gradient Method

Input: $(\tilde{\nabla}\mathcal{L})$, η, β, $\gamma_0 = 1$, $(\tau_k)_k$, (γ_k), $\boldsymbol{x}_0$
for $k = 0$ **to** $K - 1$ **do**
  Sample $\tilde{\nabla}\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_k)$
  Compute $\boldsymbol{y}_k = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \left\{\langle \boldsymbol{x} - \boldsymbol{x}_k, \tilde{\nabla}\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_k)\rangle + L\|\boldsymbol{x} - \boldsymbol{x}_k\|^2\right\}$
  Set $\phi_k(\boldsymbol{x}) = L\|\boldsymbol{x} - \boldsymbol{x}_0\|^2 + \sum_{i=0}^{k} \gamma_i\left[\langle \tilde{\nabla}\mathcal{L}_{\mathcal{H}}(\boldsymbol{x}_i), \boldsymbol{x} - \boldsymbol{x}_i\rangle + \frac{\mu}{4}\|\boldsymbol{x} - \boldsymbol{x}_i\|^2\right]$
  Compute $\boldsymbol{z}_k = \arg\min_{\mathbb{R}^d} \phi_k(\boldsymbol{x})$
  Compute $\boldsymbol{x}_{k+1} = \tau_k \boldsymbol{z}_k + (1 - \tau_k)\boldsymbol{y}_k$
end for

We consider Algorithm 9, with a sequence $\{\gamma_k\}$ that satisfies $\gamma_0 = 1$ and $2L + \Gamma_k\mu/2 =$

$2L\gamma_{k+1}^2/\Gamma_{k+1}$ with $\Gamma_k := \sum_{i=0}^{k} \gamma_k$. This algorithm is an instantiation of the fast gradient method from (Devolder et al., 2014), in the case of unconstrained optimization, where we used the values in Proposition 6.E.2.

As required in Proposition 6.E.2, we assume Algorithm 9 uses $(\zeta^2, \alpha = 0)$-inexact gradient oracles, i.e. only an *absolute* error is present. In a federated setting, this corresponds to (G, B)-Heterogeneity assumptions with $B = 0$, a common hypothesis in the literature (e.g. Karimireddy et al. (2023)).

Theorem 6.E.3 (Fast Gradient Method w. Inexact Oracles)**.** *Algorithm 9 applied to a μ-strongly convex and L-smooth function $\mathcal{L}_{\mathcal{H}}$, with a $(\zeta^2, \alpha = 0)$-inexact gradient oracle, generates a sequence $(\boldsymbol{y}_k)_{k\geq 1}$ satisfying,*

$$\mathcal{L}_{\mathcal{H}}(\boldsymbol{y}_k) - \mathcal{L}_{\mathcal{H}}^* \leq \min\left(\frac{8LR}{k^2}, 2LR\exp\left(-\frac{k}{4}\sqrt{\frac{\mu}{L}}\right)\right) + \left(1 + \sqrt{\frac{L}{\mu}}\right)\frac{3\zeta^2}{2\mu},$$

where $R := \frac{1}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2$.

Proof: This follows from combining Theorem 7 in Devolder et al. (2014) with Proposition 6.E.2.

Corollary 6.E.4. *Assume $(G, B = 0)$-Heterogeneity (Assumption 6.2.2), that $\mathcal{L}_{\mathcal{H}}$ is μ-strongly convex and L-smooth, and that the robust aggregation rule F in Algorithm 3 is $(f, \nu = \mathcal{O}(f/n-2f))$-robust. Then, Algorithm 9 using inexact oracle from Algorithm 3, requires*

$$k = \mathcal{O}(\sqrt{\kappa}\log(1/\varepsilon))$$

iterations to ensure

$$\mathcal{L}_{\mathcal{H}}(\boldsymbol{y}_k) - \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}^*) \leq \mathcal{O}\left(\varepsilon + \frac{\sqrt{\kappa}}{\mu}\frac{f}{n-2f}G^2\right).$$

6.F Additional Experiments

6.F.1 Description of the Attacks

In our experiments we used the following attacks.

1. **Gaussian Attack.** The Byzantine Nodes send an $\mathcal{N}(0, I)$ gaussian vector at each iteration.

2. **Label Flipping.** Every adversarial worker computes its gradient on permuted labels. Since the labels are in [10] of MNIST, the gradients loss correspond to the one computed on the covariate c_i associated with the label $(y_i - 1) \bmod 10$.

3. **(Optimized) A Little Is Enough (Baruch et al., 2019).** The Byzantine nodes send

$$\mathrm{ALIE}(\boldsymbol{x}) = \nabla \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}) + \tau_{opt} \cdot \overline{\sigma}_{\mathcal{H}}$$

where $\overline{\sigma}_{\mathcal{H}}$ is the coordinate-wise standard deviation of $\nabla \mathcal{L}_i(\boldsymbol{x}), i \in \mathcal{H}$ and τ_{opt} is optimized through a linear search.

4. **(Optimized) Inner Product Manipulation.(Xie et al., 2020)** The Byzantine nodes send

$$\mathrm{IPM}(\boldsymbol{x}) = (1 - \tau_{opt}) \nabla \mathcal{L}_{\mathcal{H}}(\boldsymbol{x}).$$

Where τ_{opt} is optimized through a linear search.

6.F.2 Parameter Tuning

The choice of the parameters of all algorithm has been performed by an hyper-parameter search of the learning-rate in the Byzantine-free setting, and then applied to the Byzantine case.

To reduce the number of parameters in Algorithm 5, we define τ, η, γ from θ only using the theoretical values of the non-Byzantine setting. Namely, we took $\gamma = \mu$, $\tau = \min(1, \sqrt{\frac{\theta\mu}{2}})$, $\eta = \min(\frac{1}{2\mu}, \frac{\theta}{2\tau})$, with μ the l2-regularization of the empirical loss.

6.F.3 Empirical Results

In Figures 6.3 to 6.6 we compare the impact of the averaging rule used in Algorithm 3 on the performance of D-GD, the extragradient method with and without similarity. To do so, we compare the coordinate-wise trimmed mean (Yin et al., 2018), with our multivariate Huber estimator Section 6.B.1, Krum (Blanchard et al., 2017) and Covariance-agnostic Filter (CAF) (Allouah et al., 2025b) (cf. Section 6.C.2 for theoretical guarantees of the optimization methods used with CAF).

Interestingly, we note that the theoretical tightness of CAF translates into better asymptotic error. However, the trajectories of CAF-based algorithms are highly unstable compared to the other ones.

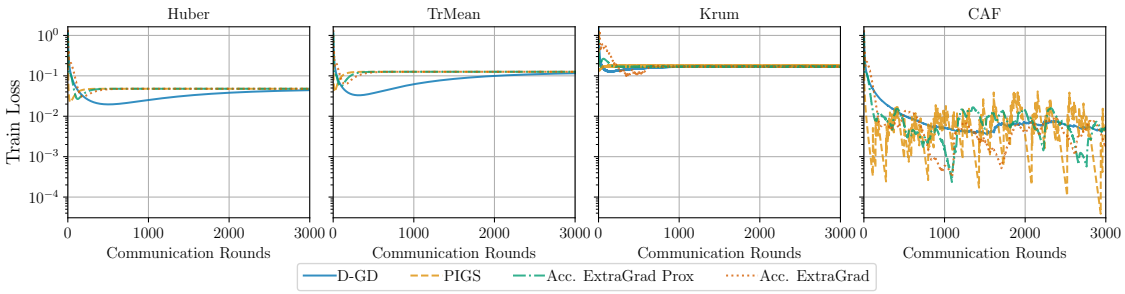


Figure 6.3: Impact of the aggregator under ALIE attack on MNIST dataset (heterogeneity $\beta = 1$, strong convexity $\mu = 0.01$, 40 honest clients and 1 adversary)

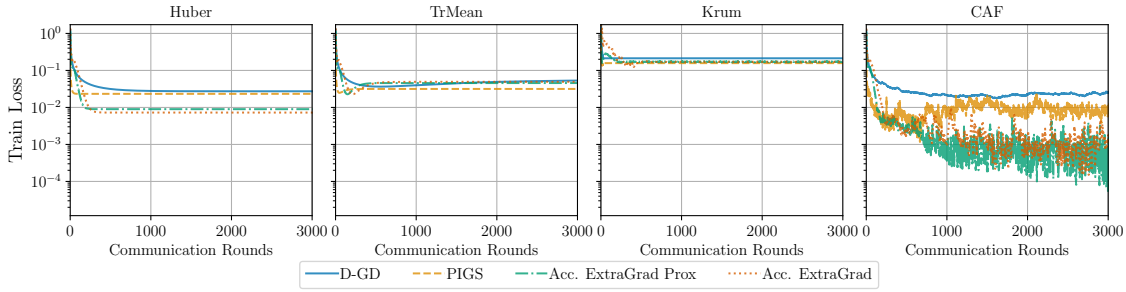


Figure 6.4: Impact of the aggregator under IPM attack on MNIST dataset (heterogeneity $\beta = 1$, strong convexity $\mu = 0.01$, 40 honest clients and 1 adversary)

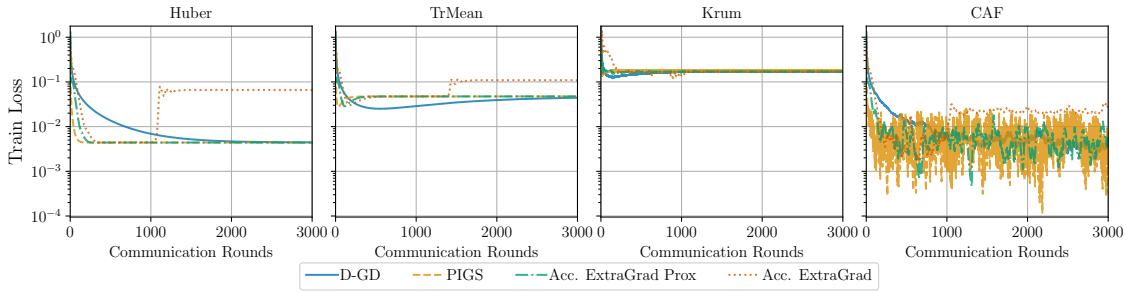


Figure 6.5: Impact of the aggregator under a LabelFlipping attack on MNIST dataset (heterogeneity $\beta = 1$, strong convexity $\mu = 0.01$, 40 honest clients and 1 adversary)

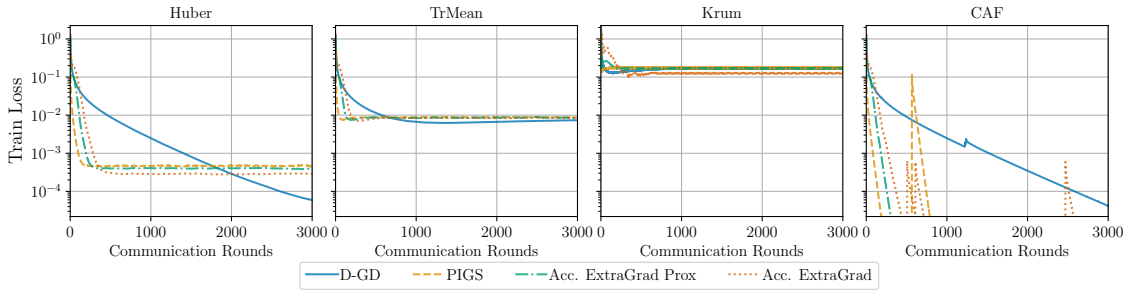


Figure 6.6: Impact of the aggregator under a Gaussian noise attack on MNIST dataset (heterogeneity $\beta = 1$, strong convexity $\mu = 0.01$, 40 honest clients and 1 adversary)

6.G Additional Discussion

6.G.1 Limitations

Strongly convex regime. In this work, we chose to focus on designing computational efficient algorithms for first order optimization. Considering the lower bound Theorem 6.2.3, convergence guaranties either of the loss $\mathcal{L}(\boldsymbol{x})$ or of the parameter $\|\boldsymbol{x} - \boldsymbol{x}^*\|^2$ can only be established for strongly convex loss functions. Even if it can be shown, for instance, that D-GD with smooth non-convex $\mathcal{L}_{\mathcal{H}}$ enjoys $\|\nabla \mathcal{L}_{\mathcal{H}}\|^2 = O(f/n-2fG^2)$, it is non-trivial to show accelerated rates without relying on function values. As a consequence, we leave the design

and analysis of accelerated Byzantine-robust algorithms beyond the strongly convex regime for future works.

Stochastic Setting. It is known since (Karimireddy et al., 2021) that permutation-invariant algorithms, i.e. algorithms in which the server average the information from the clients without taking account their identities, can not achieve an optimal error when the clients send i.i.d. stochastic oracles. Namely, when the stochastic gradient oracles send by the clients satisfies $E[\|\nabla \mathcal{L}_i(\boldsymbol{x}) - g_i(\boldsymbol{x})\|^2] \leq \sigma^2$, a permutation invariant Byzantine-resilient algorithm can only converge to a $O(\frac{f}{n-2f}\frac{\sigma^2}{\mu})$ error.

This impossibility result – which can be understood as a consequence of Theorem 6.2.3 with $B = 0$ – can be alleviated in two ways: either increase the batch size when the algorithm converge, or introduce a momentum term to the stochastic oracle (Karimireddy et al., 2021; Farhadkhani et al., 2022). If plugging our abstraction Algorithm 5 in the work of Ajalloeian and Stich (2020) gives convergence result for Byzantine resilient D-SGD, it does not cover the local momentum strategy.

Part IV

Privacy

Chapter 7

The Curse of Dimensionality in Privacy: High-dimensional Mean Estimation with Unknown Covariance

We present negative results on the feasibility of dimension-free private mean estimation when the covariance matrix is unknown. Previous work on private mean estimation has shown that, when the covariance matrix is known, it is possible to estimate privately the mean of a sub-gaussian distribution with an error in Euclidean distance $\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2 \leq \alpha$ with only $n \gtrsim \operatorname{tr} \boldsymbol{\Sigma}/\alpha^2 + \operatorname{tr} \sqrt{\boldsymbol{\Sigma}}/(\alpha\varepsilon)$ samples, i.e. with an additional sample complexity due to privacy independent of the dimension. In this work, we show that when the covariance matrix is unknown and the number of sample is small compared to the dimension, such dimension-free cannot be achieved. We support this claim by two analysis, in the first one, for a number of samples lower than the effective dimension of the problem, i.e. $n \lesssim \operatorname{tr} \boldsymbol{\Sigma}/\|\boldsymbol{\Sigma}\|_2$, the best achievable accuracy is the one of the naive private mean estimator with an isotropic gaussian mechanism, i.e $\alpha^2 \approx \frac{d \operatorname{tr} \boldsymbol{\Sigma}}{n^2}$. In the second one, we consider diagonal covariance matrices for which $\operatorname{tr} \boldsymbol{\Sigma}/\|\boldsymbol{\Sigma}\|_2 = \tilde{O}(1)$, and on which private algorithms that do not know the covariance matrix suffer a multiplicative penalty of $\sqrt{d}/n^{8.5}$ compared to covariance-aware algorithms. These two results highlight the necessity of a dimension-dependent number of samples to achieve a non-trivial error for private mean estimation under unknown covariance.

Contents

7.1 Introduction 161
7.1.1 Contributions 161
7.1.2 Proof Technique 162
7.1.3 Notation 163
7.2 Preliminaries 164
7.2.1 Differential Privacy 164
7.2.2 Risk Definition 164
7.3 The Non-Diagonal Case 165
7.3.1 Main Result 165
7.3.2 From Mean Estimation to Sampling 166
7.3.3 Lower-Bound on the Sampler Norm 168
7.4 The Diagonal Case 173
7.4.1 Prior Distribution 173
7.4.2 Main Result 174
7.4.3 From Mean Estimation to Sampling and Selection 175
7.4.4 Lower Bound on the Accuracy of the Selection Mechanism 178

7.A Concentration Results 187

7.1 Introduction

If machine learning models are now routinely used by a large proportion of the population, the security data on which they are trained is still a partially unresolved problem. It has been repeatedly shown that statistical and machine learning inference methods can disclose potentially sensitive information about the individual that contributed to the dataset, as evidenced via *reconstruction attacks* (Dinur and Nissim, 2003; Dwork et al., 2007; Dwork and Yekhanin, 2008; Kasiviswanathan et al., 2010), *membership-inference attacks* (Homer et al., 2008; Sankararaman et al., 2009; Bun et al., 2014; Dwork et al., 2015; Shokri et al., 2017; Yeom et al., 2018) and instances of unwanted *memorization of training data* (Carlini et al., 2019, 2021; Feldman, 2020; Brown et al., 2021a). To mitigate the risk of privacy violation, the notion of *Differential Privacy*(DP) has been introduced by Dwork et al. (2006) to quantify and mitigating the risk of information leak within data-using pipelines. Differentially privacy has become the gold standard for data privacy protection, and has been regularly adopted in industry (Erlingsson et al., 2014; Hartmann and Kairouz, 2023; Tastuggine and Mironov, 2020) and by governments (Haney et al., 2017; Abowd, 2018; Abowd et al., 2022).

However, training a machine learning model privately comes at the price of some deterioration of its accuracy: privacy is obtained by adding noise while optimizing the model on its training data with algorithms such as differentially private gradient descent (DP-SGD). When agnostic to the actual data distribution and its potentially low-dimensional structure, the added noise significantly increases the resulting error, which translates into poorer performance of the model. Designing a private mean estimator (for gradients) that adapts to the potentially low-dimensional structure of the data is thus key to better DP machine learning models.

The work of Dagan, Jordan, Yang, Zakynthinou and Zhivotovskiy (Dagan et al., 2024) opens the way to such an adaptive private mean estimator. They show that, when the covariance matrix of a sub-gaussian distribution is known, only $n \gtrsim \operatorname{tr}\boldsymbol{\Sigma}/\alpha^2 + \operatorname{tr}\sqrt{\boldsymbol{\Sigma}}/(\alpha\varepsilon)$ data samples are required to estimate the expectation $\boldsymbol{\mu}$ with accuracy in ℓ_2 norm $\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2 \leq \alpha$ with high probability, which is optimal. The case of an unknown covariance matrix is, however, less clear, since, in this setting, they propose another algorithm which still depends on the dimension when the covariance matrix is non-diagonal, or when the decay of the eigenvalues is pathological.

Question 1. *Is it possible to estimate privately the expectation of a sub-gaussian distribution of unknown covariance matrix with a relation between the accuracy α in ℓ_2 norm and the number of samples independent of the dimension d, but rather only depends on $\operatorname{tr}(\sqrt{\boldsymbol{\Sigma}})$ and $\operatorname{tr}(\boldsymbol{\Sigma})$?*

7.1.1 Contributions

In this article, we argue that Question 1 can not be answered positively, in the sense that there are settings in which the covariance matrix is such that $\operatorname{tr}(\boldsymbol{\Sigma})/\|\boldsymbol{\Sigma}\|_2 \leq \tilde{\mathcal{O}}(1)$ and $\operatorname{tr}(\sqrt{\boldsymbol{\Sigma}})/\|\boldsymbol{\Sigma}\|_2^{1/2} = \tilde{\mathcal{O}}(1)$, while the accuracy of any private algorithm is at least $\alpha \geq \Omega(d^{\Omega(1)})$.

In particular, we support this negative result by considering two settings: one in which $\mathbf{\Sigma}$ describes a k-dimensional subspace, with $n \lesssim k$, and one in which $\mathbf{\Sigma}$ is diagonal but with a pathological decay of its eigenvalues.

Non-diagonal covariance matrix case. We assume the covariance matrix $\mathbf{\Sigma}$ to be uniformly distributed among the projector on k-dimensional linear subspaces of $\mathbb{R}^d$, and that $n \lesssim k = \operatorname{tr}(\mathbf{\Sigma}) = \operatorname{tr}\left(\sqrt{\mathbf{\Sigma}}\right)$. In this low-sample regime, we show that, for reasonable values of ε and δ, the best possible accuracy of any (ε, δ)-DP estimator is $\alpha \geq \Omega\left(\frac{\sqrt{dk}}{n}\right)$. This accuracy coincides with that of the naive algorithm, which estimates $\operatorname{tr}\mathbf{\Sigma}$, and adds an isotropic Gaussian noise $\mathcal{N}(0, \frac{\operatorname{tr}\mathbf{\Sigma}\log(1.25/\delta)}{n^2\varepsilon^2} I_d)$ to the average of the filtered samples. It follows that no adaptive algorithm can perform better than the naive Gaussian mechanism in the unknown covariance case when the number of samples is smaller than the effective dimension of the problem.

Diagonal covariance matrix case. We consider a diagonal covariance matrix for which the entries follow an inverse chi-squared distribution, for which we have approximately $\{\mathbf{\Sigma}_{ii}; i \in [d]\} \approx \{\frac{d^2}{i^2}; i \in [d]\}$. In this setting, we show that, while with high probability $\operatorname{tr}\sqrt{\mathbf{\Sigma}}/\|\mathbf{\Sigma}\|_2^{1/2} = \tilde{\mathcal{O}}(1)$, any private algorithm that does not know the covariance matrix in advance must have an accuracy larger than $\alpha^2 \geq \tilde{\Omega}\left(\frac{\|\mathbf{\Sigma}\|_2}{n}\frac{\sqrt{d}}{n^{7.5}}\right)$, where $\tilde{\Omega}\left(\frac{\|\mathbf{\Sigma}\|_2}{n}\right)$ corresponds to the optimal accuracy of private estimator that takes the covariance matrix as input, which also coincide with the optimal accuracy of the (non-private) sample mean.

It follows that, even for diagonal the covariance matrices, and even for nearly constant effective dimension $\operatorname{tr}\sqrt{\mathbf{\Sigma}}/\|\mathbf{\Sigma}\|_2$, having a dimension-independent error for any private algorithm necessarily requires a number of samples that grows as a power of the dimension.

7.1.2 Proof Technique

Limitations of previous proof techniques Showing a relation between $\widetilde{\mathfrak{R}}_{\varepsilon,\delta}$ and $\mathfrak{R}_{\varepsilon,\delta}$ is non-trivial, as the standard lower bounds technique on mean estimation – the so-called *fingerprinting* or *tracing* method () – usually tracks only one source of randomness - the covariance or the expectation. For instance, (Cai et al., 2023; Lyu and Talwar, 2025) propose to abstract away the fingerprinting proof technique. They interpret it as an upper and lower bound on the divergence of the expected output of the algorithms with respect to the prior distribution, i.e. $\div g(\theta)$ where $g(\theta) := \mathbb{E}_{\mathbf{X}\sim\mathcal{P}(\theta)}[\mathcal{M}(\mathbf{X})]$, which can be expressed as an inner-product between the algorithm's statistical *score* and the input samples. On the one hand, the upper bound is established since, due to privacy, the algorithm's distribution is close to that of an algorithm independent of the input samples, and thus $\div(g(\theta))$ is close to zero. On the other hand, when the algorithm is accurate, its output evolves with the distribution parameter θ, and thus the divergence of $g(\theta)$ is large. This provides a condition linking the privacy of the algorithms with their accuracy.

However, in this approach, the private algorithm outputs the parameter of the distribution directly. In contrast, we want to show that when the covariance matrix is unknown to

the algorithm, the mean estimator suffers a penalty in its estimate. To do so, we must take into account that both the expectation and the covariance vary (e.g., are random), while the algorithm a priori outputs only an estimate of the expectation.

Intuition To avoid this issue that the algorithm does not directly output an estimate of the covariance matrix, we build on the following intuition: since the algorithm is differentially private, it should obfuscate its output with noise. When the algorithm is also close to the optimal algorithm, this obfuscating noise should be somehow similar to that of the optimal algorithm, i.e., with a covariance proportional to $\operatorname{tr}\sqrt{\boldsymbol{\Sigma}}$. However, when the covariance matrix is unknown, the algorithm can only rely on an estimate of the covariance matrix, which is either too pessimistic or correlated with the empirical covariance matrix of the dataset. It follows that, when the number of samples is not large enough to learn the covariance matrix privately, relying on a pessimistic estimate of the covariance matrix in the injected noise is necessary, which correlatively harms the accuracy of the estimator.

Sampling Mechanism To formalize this intuition, both lower bounds use the following proof structure:

1. For any differentially private mean estimator $\mathcal{A}$, we construct a "sampling" or a "selection" mechanism $\mathcal{M}$ to inspect the covariance of the error $\mathcal{A}(\mathbf{X}+\boldsymbol{\mu})-\boldsymbol{\mu}$, where $\boldsymbol{\mu}\sim\mathcal{N}(0,\mathbb{I})$ is sampled randomly by the sampling mechanism. We show that, if the initial mechanism $\mathcal{A}$ is accurate, then the covariance matrix of $\mathcal{A}(\mathbf{X}+\boldsymbol{\mu})-\boldsymbol{\mu}$ is must be close to the one of the data distribution $\boldsymbol{\Sigma}$

2. We then use that $\mathcal{M}$ is (ε,δ)-differentially private and independent of the distribution covariance matrix to argue that, if covariance matrix of $\mathcal{M}$ is a good approximation of $\boldsymbol{\Sigma}$, then it is also highly correlated with the empirical covariance of the input dataset. We then use that the mechanism is also private to find a contradiction. Going back to the original hypothesis, we deduce a lower bound on the accuracy of estimator $\mathcal{A}$

7.1.3 Notation

For $\mathbf{X}\in\mathbb{R}^{n\times d}$, and $\boldsymbol{\mu}\in\mathbb{R}^d$ we denote $\mathbf{X}+\boldsymbol{\mu}=(\mathbf{X}_i+\boldsymbol{\mu})_{i\in[n]}$. For $x\in\mathbb{R}^d$, $\|x\|$ denotes the standard Euclidean norm of x in $\mathbb{R}^d$, and for $\boldsymbol{\Sigma}$ a positive semi-definite (PSD) matrix, we denote the pseudo norm induced by $\boldsymbol{\Sigma}$ as $\|x\|_{\boldsymbol{\Sigma}}:=\sqrt{x^T\boldsymbol{\Sigma}x}$. We note $\mathbb{I}_d\in\mathbb{R}^{d\times d}$ the identity matrix. We denote $\preceq$ the partial ordering within PSD matrices, i.e. $\boldsymbol{\Sigma}\preceq\boldsymbol{\Sigma}'$ if for any vector x, $x^T(\boldsymbol{\Sigma}'-\boldsymbol{\Sigma})x\geq 0$. Pr denotes the probability of a measurable event. For variables u and v, we note $u\in\mathcal{O}(v)$ and $v\in\Omega(u)$ when $u\leq Cv$ with $C>0$ a numerical constant, and $u\in\Theta(v)$ if and only if both $u\in\mathcal{O}(v)$ and $u\in\Omega(v)$ hold. In $\mathcal{O}$, Θ and Ω, a tilde as $\tilde{\mathcal{O}}$ translates that logarithmic factors are omitted. Consider two random variables X and Y, and let f be a function. We denote $\mathbb{E}_{X|_Y}[f(X)]$ the conditional expectation of $f(X)$ given Y, and $\mathbb{E}[f(X)]$ the total expectation.

7.2 Preliminaries

7.2.1 Differential Privacy

Definition 7.2.1 ((ε, δ)-indistinguishability)**.** Two distributions $\mathcal{P}, \mathcal{Q}$ over a domain $\mathcal{W}$ are (ε, δ)-indistinguishable, denoted $\mathcal{P} \approx_{\varepsilon,\delta} \mathcal{Q}$, if, for any measurable subset $W \subset \mathcal{W}$,

$$\mathcal{P}(W) \leq e^{\varepsilon}\mathcal{Q}(W) + \delta \quad \text{and} \quad \mathcal{Q}(W) \leq e^{\varepsilon}\mathcal{P}(W) + \delta.$$

If x and y are random variable of distribution $\mathcal{P}$ and $\mathcal{Q}$ we denote x $\approx_{\varepsilon,\delta}$ y that $\mathcal{P} \approx_{\varepsilon,\delta} \mathcal{Q}$.

Definition 7.2.2 (Differential Privacy (Dwork et al., 2006))**.** A randomized algorithm $\mathcal{A} : \mathcal{X}^* \to \mathcal{W}$ is (ε, δ)-differentially private, if, for all neighboring data sets $\mathbf{X}, \mathbf{X}'$, i.e. datasets that differ on a single entry, we have $\mathcal{A}(\mathbf{X}) \approx_{\varepsilon,\delta} \mathcal{A}(\mathbf{X}')$.

7.2.2 Risk Definition

Formalizing our result require to define a notion of optimal error. To do so, we take a Bayesian perspective to measure the accuracy of the optimal private mean estimator. Given a prior distribution $\mathcal{P}$ on the expectation and the covariance of i.i.d. normal random vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, we consider the task of estimating the expectation of those normal random vectors privately, and compare algorithms agnostic to the covariance matrix of the input samples to covariance-aware algorithms.

Formally, we denote $A_{\varepsilon,\delta}$ the class of (ε, δ)-DP algorithms. We also define both the covariance-agnostic Bayes risk and the covariance-aware Bayes risk with constant probability.

Definition 7.2.3 (Bayes-risk)**.** Let $\mathcal{P}$ be a prior distribution over $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\boldsymbol{\Sigma} \in \S_+^d$. The *covariance-agnostic Bayes risk* with respect to the prior $\mathcal{P}$ is defined as:

$$\mathfrak{R}_{\varepsilon,\delta}(\mathcal{P}, n) = \inf\left\{\alpha^2 \geq 0 : \exists \mathcal{A} \in A_{\varepsilon,\delta} \text{ s.t. } \Pr_{\substack{(\boldsymbol{\mu},\boldsymbol{\Sigma})\sim\mathcal{P} \\ \mathbf{X}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})^{\otimes n}}}\left[\|\mathcal{A}(\mathbf{X}) - \boldsymbol{\mu}\|_2^2 \leq \alpha^2\right] \geq \frac{2}{3}\right\}.$$

We also define the *covariance-aware Bayes risk* with respect to the prior $\mathcal{P}$ by:

$$\widetilde{\mathfrak{R}}_{\varepsilon,\delta}(\mathcal{P}, n) = \inf\left\{\alpha^2 \geq 0 : \exists \mathcal{A} \in A_{\varepsilon,\delta} \text{ s.t. } \Pr_{\substack{(\boldsymbol{\mu},\boldsymbol{\Sigma})\sim\mathcal{P} \\ \mathbf{X}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})^{\otimes n}}}\left[\|\mathcal{A}(\mathbf{X}, \boldsymbol{\Sigma}) - \boldsymbol{\mu}\|_2^2 \leq \alpha^2\right] \geq \frac{2}{3}\right\}.$$

In this article, we will show that for some priors $\mathcal{P}$ on the covariance distribution (and on the expectation $\boldsymbol{\mu}$), the covariance-agnostic Bayes risk is larger than the covariance-aware Bayes risk, by a factor that depends on the ambient dimension d.

Remark 7.2.4. Defining the Bayes risk as in Definition 7.2.3 can seem over-complicated compared to a mean squared error definition, such as $\inf_{\mathcal{A}\in\mathcal{A}_{\varepsilon,\delta}} \mathbb{E}_{\mathbf{X},\boldsymbol{\mu},\boldsymbol{\Sigma}}[\|\mathcal{A}(\mathbf{X}) - \boldsymbol{\mu}\|^2]$. However, it enjoys the following properties:

1. Using Markov's inequality, the mean-squared-error is at least the Bayes risk of Definition 7.2.3, up to constants.
2. The mean-squared-error of private algorithms that rely on the propose-test-release paradigm (Dwork and Lei, 2009) – such as the ones in (Dagan et al., 2024) – is often ill-defined.
3. Considering Definition 7.2.3 allows for heavy-tailed distributions of the eigenvalues of the covariance matrix, which will be the case in Section 7.4.

7.3 The Non-Diagonal Case

This section provides the lower bound of the Bayes risk for random non-diagonal covariance matrices, uniformly distributed among the orthogonal projectors on k-dimensional linear subspaces of $\mathbb{R}^d$.

We first state the main result in Section 7.3.1, and then decompose its proof in Sections 7.3.2 and 7.3.3.

7.3.1 Main Result

Theorem 7.3.1. *Let $\mathcal{P}_1$ be a distribution on $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ such that $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are independent, $\boldsymbol{\mu} \sim \mathcal{N}(0, \rho\mathbb{I})$, and $\boldsymbol{\Sigma}$ is uniformly random on the orthogonal projectors on k-dimensional linear subspaces of $\mathbb{R}^d$.*
Then, there are universal constants $c, C > 0$, such that, if $\rho > C\max(1, k/n^2)$, $n \leq ck$, $\varepsilon = 1$ and $\delta \leq c/d$, we have

$$\Theta\left(\frac{dk}{n^2}\right) = \mathfrak{R}_{\varepsilon,\delta}(\mathcal{P}_1, n) \geq c\frac{d}{k}\,\widetilde{\mathfrak{R}}_{\varepsilon,\delta}(\mathcal{P}_1, n) = \Theta\left(\frac{k^2}{n^2}\right).$$

The proof of Theorem 7.3.1 is deferred to Section 7.3.3.

It follows from Theorem 7.3.1 that, in the *low-sample* regime, i.e. when the number of samples is smaller than the effective dimension of the problem – here k –, not knowing the covariance of the underlying distribution implies a necessary overhead of $\Theta\left(\frac{d}{k}\right)$ on the error of any private algorithm.

Remark 7.3.2 (Ongoing work). The assumption on δ, i.e., $\delta \in \mathcal{O}(1/d)$, is slightly stronger than what we could dream of. Indeed, the standard minimal assumption on δ is $\delta \leq \mathcal{O}(1/n)$, to exclude from the analysis the $(0, 1/n)$-DP "name and shame" algorithm, that randomly outputs a sample as $\mathcal{M}(\mathbf{X}) = \mathrm{X}^I$ with $I \sim \mathcal{U}(\{1, \dots, n\})$.

This point is an ongoing work, and we have an informal sketch of proof with a $\delta \leq c/n$ requirement, at the cost of the benignly stronger requirement of $n \log(n) \leq ck$.

7.3.1.1 Tightness

Lemma 7.3.3 (Optimal Known Covariance Algorithm)**.** *Consider the known (ε, δ)-private mean estimation algorithm under the known covariance of Dagan et al. (Dagan et al., 2024). This algorithm meets the Bayes risk under known covariance, as with probability at least $2/3$ over the randomness of $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, it satisfies*

$$\|\mathcal{A}(\mathbf{X}) - \boldsymbol{\mu}\|^2 \leq O\left(\frac{k}{n} + \frac{k^2 \log(1/\delta)}{n^2 \varepsilon^2}\right).$$

Consider the naive algorithm Algorithm 10. This algorithm achieves with probability at least $2/3$

$$\|\mathcal{A}(\mathbf{X}) - \boldsymbol{\mu}\|^2 \leq O\left(\frac{k}{n} + \frac{d^2 k^2 \log(1/\delta)}{n^2 \varepsilon^2}\right).$$

We now state the naive algorithm that achieves the optimal risk under unknown covariance in the low-sample regime.

Algorithm 10 Naive private mean estimation (Work in progress)

Input: $\mathbf{X} \in \mathbb{R}^{n \times d}$, ε, δ
Split $\mathbf{X}$ into two datasets of size $n/2$, $\mathbf{Y}$ and $\mathbf{Z}$.
Estimate $\hat{t} = \operatorname{tr} \boldsymbol{\Sigma}$ using the mean estimation algorithm of (Karwa and Vadhan, 2018) on $\{\|\mathbf{Z}^i - \mathbf{Z}^(i + n/4)\|^2; i \in [n/4]\}$
Use the covariance algorithm of (Dagan et al., 2024) with $\boldsymbol{M} \approx \hat{t}\mathbb{I}_d$.

Proof of Lemma 7.3.3. This proof is ongoing work. □

7.3.2 From Mean Estimation to Sampling

To prove Theorem 7.3.1, we first show that for any algorithm that achieves the mini-max optimal rate $\mathfrak{R}_{\varepsilon,\delta}(\mathcal{P}, n)$, we can construct an (ε, δ)-private algorithm $\mathcal{M}$ that correlates with the population covariance matrix.

Definition 7.3.4 ((λ, α)-sampler)**.** Let $\boldsymbol{\Sigma} \sim \mathcal{P}_1$, and fix $\lambda, \alpha > 0$. An algorithm $\mathcal{M}$ is a (λ, α)-sampler if there exists $n = n(d, k, \lambda, \alpha)$ such that,

$$\Pr_{\mathbf{X} \sim \mathcal{N}(0, \boldsymbol{\Sigma})^{\otimes n}, \boldsymbol{\Sigma}, \mathcal{M}} \left[\mathcal{M}(\mathbf{X})^T \boldsymbol{\Sigma} \mathcal{M}(\mathbf{X}) \geq \lambda^2\right] \geq \frac{1}{3} \quad \text{and} \quad \|\mathcal{M}(\mathbf{X})\|_2 \leq \alpha \quad \text{a.s.}$$

Remark 7.3.5. Not all values of λ and α are compatible with the existence of a (λ, α)-sampler. Trivially, it follows directly from the definition that necessarily $\alpha \geq \lambda$. In fact, we will show in Section 7.3.3 that, for $n \leq \mathcal{O}(k)$, α must be significantly larger than λ.

Algorithm 11 Sampler – Non-Diagonal Setting

Input: $\mathbf{X} \in \mathbb{R}^{n\times d}$, α, $\mathcal{A}$, ρ
Sample $\boldsymbol{\mu} \sim \mathcal{N}(0, \rho I_d)$.
Return $\mathcal{M}(\mathbf{X}) = \mathrm{proj}_\alpha(\mathcal{A}(\mathbf{X}+\boldsymbol{\mu}) - \boldsymbol{\mu})$

The considered "sampler" used in our lower bound consists essentially of the clipped error of the mean estimation mechanism $\mathcal{A}$. Namely, by denoting the clipping operator $\mathrm{proj}_\tau(x) = x\min(1, \|x\|/\tau)$, we have the following guarantee on Algorithm 11.

Proposition 7.3.6 (From mean estimation to sampling)**.** *Let $c, C > 0$ be numerical constants. Denote the Bayes optimal risk under unknown covariance as $\alpha = \mathfrak{R}_{\varepsilon,\delta}(\mathcal{P}_1, n)$, and let $\mathcal{A}$ be the (ε,δ)-DP algorithm that achieves this risk. Consider Algorithm 11, where $\rho > C\max(k/n^2, 1)$, and assume $\varepsilon \le 1$ and $\delta \le c/k$.*
Then $\mathcal{M}$ is (ε,δ)-differentially private, and a $\left(c\frac{k}{n}, \alpha\right)$-sampler.

To show the above proposition, we begin by stating a folklore result on the achievable accuracy of Gaussian private mean estimation

Proposition 7.3.7 (Folklore Result)**.** *Let $k, n \in \mathbb{N}$. There are numerical constants $c, C > 0$, such that for any (ε,δ)-mechanism $\mathcal{A} : \mathbb{R}^{k\times n} \to \mathbb{R}^k$ with $\delta \le \frac{c}{k}$, any $\varepsilon \le 1$, and any $\rho \ge C\max(k/n^2, 1)$,*

$$\Pr_{\mathbf{X}\sim\mathcal{N}(\boldsymbol{\mu},\mathbb{I}_k)^{\otimes n},\, \boldsymbol{\mu}\sim\mathcal{N}(0,\rho\mathbb{I}_k)}[\|\mathcal{A}(\mathbf{X}) - \boldsymbol{\mu}\| \ge \frac{ck}{n}] \ge \frac{2}{3}.$$

This folklore result is a Bayesian variation of well-known lower bounds on private mean estimation under a Gaussian distribution (Kamath et al., 2019) via fingerprinting arguments. This proof will be published as an independent note on arXiv.

Proof of Proposition 7.3.6. Recall that we denote $\|\cdot\|_{\boldsymbol{\Sigma}}$ the pseudo-norm induced by $\boldsymbol{\Sigma}$, e.g. $\|\mathcal{M}(\mathbf{X})\|_{\boldsymbol{\Sigma}}^2 = \mathcal{M}(\mathbf{X})^T\boldsymbol{\Sigma}\mathcal{M}(\mathbf{X})$. Using this notation, we have by definition of the clipping operator

$$\|\mathrm{proj}_{3\alpha}(\mathcal{A}(\mathbf{X}+\boldsymbol{\mu}) - \boldsymbol{\mu})\|_{\boldsymbol{\Sigma}}^2 = \|(\mathcal{A}(\mathbf{X}+\boldsymbol{\mu}) - \boldsymbol{\mu})\|_{\boldsymbol{\Sigma}}^2 \left[\min\left(1, \frac{3\alpha}{\|\mathcal{A}(\mathbf{X}+\boldsymbol{\mu}) - \boldsymbol{\mu}\|}\right)\right]^2.$$

Consider $G_1 := \{\|\mathcal{A}(\mathbf{X}+\boldsymbol{\mu}) - \boldsymbol{\mu}\| \le \alpha\}$ the event under which the clipping does not modify anything, and $G_2 := \left\{\|\mathcal{A}(\mathbf{X}+\boldsymbol{\mu}) - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}} \ge c\frac{k}{n}\right\}$ the event under which the input of the clipping is large. We show that both events hold with probability $2/3$ for $c > 0$, a small numerical constant.

1. By definition of $\mathcal{A}$, it holds that $\Pr(G_1) = \Pr[\|\mathcal{A}(\mathbf{X}+\boldsymbol{\mu}) - \boldsymbol{\mu}\|^2 \le \alpha] \ge 2/3$.

2. For any fixed rank-k orthogonal projection $\boldsymbol{\Sigma}$, we project the random gaussian vector $\boldsymbol{\mu}$ on $\mathrm{Span}(\boldsymbol{\Sigma})$ and $\mathrm{Span}(\boldsymbol{\Sigma})^\perp$ as $\boldsymbol{\mu} := (\boldsymbol{\mu}_{\boldsymbol{\Sigma}}, \boldsymbol{\mu}_{\mathbb{I}_d-\boldsymbol{\Sigma}})$. Note that $\boldsymbol{\mu}_{\boldsymbol{\Sigma}} \in \mathrm{Span}(\boldsymbol{\Sigma})$ is independent of $\boldsymbol{\mu}_{\mathbb{I}_d-\boldsymbol{\Sigma}} \in \mathrm{Span}(\mathbb{I}_d - \boldsymbol{\Sigma})$ since $\boldsymbol{\mu}$ is a gaussian vector. Fixing both,

$\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}_{\mathbb{I}_d-\boldsymbol{\Sigma}}$, we can consider the random function

$$\mathcal{A}_{\boldsymbol{\Sigma}} : \begin{cases} \mathrm{Span}(\boldsymbol{\Sigma})^n \to \mathrm{Span}(\boldsymbol{\Sigma}) \\ x \longmapsto \boldsymbol{\Sigma}\mathcal{A}\big((x, \boldsymbol{\mu}_{\mathbb{I}-\boldsymbol{\Sigma}})\big). \end{cases}$$

Since $\mathcal{A}$ is (ε, δ)-DP, $\mathcal{A}_{\boldsymbol{\Sigma}}$ is also (ε, δ)-DP by post-processing. Using Proposition 7.3.7, and that $\mathrm{rank}(\boldsymbol{\Sigma}) = k$, their exists numerical constant $c > 0$ such that, for $\delta \leq \frac{c}{k}$,

$$\Pr_{\boldsymbol{\mu}_{\boldsymbol{\Sigma}} \sim \mathcal{N}(0, \rho\boldsymbol{\Sigma}), \mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\Sigma}}, \boldsymbol{\Sigma})|_{\boldsymbol{\Sigma}}} [\|\mathcal{A}_{\boldsymbol{\Sigma}}(\mathbf{X}) - \boldsymbol{\mu}_{\boldsymbol{\Sigma}}\| \geq \frac{ck}{n}] \geq \frac{2}{3} \quad \text{a.s. w.r.t. } \boldsymbol{\Sigma} \text{ and } \boldsymbol{\mu}_{\mathbb{I}_d-\boldsymbol{\Sigma}}.$$

Eventually, since $\|\mathcal{A}(\mathbf{X} + \boldsymbol{\mu}) - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}} = \|\boldsymbol{\Sigma}\mathcal{A}(\mathbf{X} + \boldsymbol{\mu}_{\boldsymbol{\Sigma}}, \boldsymbol{\mu}_{\mathbb{I}_d-\boldsymbol{\Sigma}}) - \boldsymbol{\mu}_{\boldsymbol{\Sigma}}\|$, it holds that $\Pr[G_2] \geq \frac{2}{3}$.

A union bound yields the result. □

7.3.3 Lower-Bound on the Sampler Norm

We have shown that any differentially private mean estimator achieving the Bayes risk $\alpha = \mathfrak{R}_{\varepsilon,\delta}(\mathcal{P}_1, n)$ is associated with a sampler. We now show that, since the (λ, α) sampler $\mathcal{M}$ is agnostic to the covariance matrix, when n – the size of the input dataset – is smaller than k, α must be significantly larger than λ by a random projection argument.

Theorem 7.3.8 (Lower bound for private sampling)**.** *Let $\alpha \geq \lambda$, and $\mathcal{M}$ be a (λ, α)-sampler that is $(1, \delta)$-DP. It holds that there is a numerical constant $c > 0$ for which,*

$$\textit{If} \quad \max(n, \log(d)) \leq ck \quad \textit{then} \quad \alpha^2 \geq \Omega\left(\min\left\{\frac{d\lambda^2}{k}, \frac{\lambda^2}{n\delta\log(d)}\right\}\right).$$

We begin by combining Theorem 7.3.8 with Proposition 7.3.6 to prove Theorem 7.3.1, i.e., the general lower bound of the non-diagonal case.

Proof of Theorem 7.3.1. Let $\mathcal{A}$ be an algorithm achieving the Bayes risk under unknown covariance $\mathfrak{R}_{\varepsilon,\delta}(\mathcal{P}_1, n)$. Then, for $\delta \leq c/k$, Proposition 7.3.6 implies that there exists an (α, λ) sampling mechanism $\mathcal{M}$ with $\alpha = \mathfrak{R}_{\varepsilon,\delta}(\mathcal{P}_1, n)$ and $\lambda = \Omega\big(\frac{k}{n}\big)$. It follows from Theorem 7.3.8 that for $n < k/4e$, $\alpha^2 \geq \tilde{\Omega}\Big(\min\Big(\frac{kd}{n^2}, \frac{k^2}{n^3\delta}\Big)\Big)$. Requiring $\delta \leq ck/(nd)$ yields the desired result. □

7.3.3.1 A Finger-Printing Lower Bound on the Sampler's Norm

We now proceed to a fingerprinting-type approach to show the lower bound on the sampler Theorem 7.3.8. For this, we consider the following test statistics: for $z \in \mathbb{R}^d$, let

$$\mathcal{T}(z) := \langle \mathcal{M}(\mathbf{X}) \ , \ z\rangle^2 = \left\langle \mathcal{M}(\mathbf{X})\mathcal{M}(\mathbf{X})^T \ , \ zz^T \right\rangle. \tag{7.1}$$

This test statistic expresses the correlation between the covariance of the mechanism on input $\mathbf{X}$ and the covariance of the input random variable z. We compare the distribution of this test statistic in:

- The *OUT sample case* in which we consider $\mathcal{T}(\mathbf{z})$ with $\mathbf{z} \sim \mathcal{N}(0, \mathbf{\Sigma})$.
- The *IN sample case* in which we consider $\mathcal{T}(\mathrm{X}^I)$ with $I \sim \mathcal{U}(\{1, \dots, n\})$.

We first state the core result on the OUT and IN statistics in this subsection and combine them to prove Theorem 7.3.8. For clarity, we defer the analysis of the IN statistic in Section 7.3.3.2.

The following lemma in itself sketches the structure of the proof of Theorem 7.3.8.

Lemma 7.3.9 (OUT case)**.** *Assume $\mathcal{M}$ is a (λ, α)-sampling mechanism (Definition 7.3.4) and $n \leq k/2$, then we have*

$$\lambda^2 \leq \mathbb{E}[\mathcal{M}(\mathbf{X})^T \mathbf{\Sigma} \mathcal{M}(\mathbf{X})] = \mathbb{E}[\mathcal{T}(\mathbf{z})] \leq \mathbb{E}\left[\mathcal{M}(\mathbf{X})^T \boldsymbol{P}_{\mathbf{X}} \mathcal{M}(\mathbf{X})\right] + \frac{2k\alpha^2}{d},$$

where $\boldsymbol{P}_{\mathbf{X}} = \mathrm{proj}_{\mathrm{Span}(\mathrm{X}^1, \dots, \mathrm{X}^n)}$ is the orthogonal projector on the span of the input dataset.

Proof. We begin by considering that $\mathbf{z} \sim \mathcal{N}(0, \mathbf{\Sigma})$ is independent of $\mathcal{M}$ and $\mathbf{X}$. Thus, we have that $\mathbb{E}_{\mathbf{z}|\mathbf{\Sigma}, \mathcal{M}, \mathbf{X}}[\mathbf{z}\mathbf{z}^T] = \mathbf{\Sigma}$, and by the tower rule $\mathbb{E}[\mathcal{T}(\mathbf{z})] = \mathbb{E}[\mathcal{M}(\mathbf{X})^T \mathbf{\Sigma} \mathcal{M}(\mathbf{X})]$, which we decompose as

$$\mathbb{E}[\mathcal{T}(\mathbf{z})] = \mathbb{E}[\mathcal{M}(\mathbf{X})^T \boldsymbol{P}_{\mathbf{X}} \mathcal{M}(\mathbf{X})] + \mathbb{E}[\mathcal{M}(\mathbf{X})^T (\mathbf{\Sigma} - \boldsymbol{P}_{\mathbf{X}}) \mathcal{M}(\mathbf{X})]. \tag{7.2}$$

Remark that $\boldsymbol{P}_{\mathbf{X}}$ is almost surely a rank n orthogonal projector with $\mathrm{Span}(\boldsymbol{P}_{\mathbf{X}}) \subset \mathrm{Span}(\mathbf{\Sigma})$. Furthermore, since $\mathbf{\Sigma}$ is uniformly distributed on the rank k orthogonal projection in $\mathbb{R}^d$, we have that, conditionally on $\mathbf{X}$, $\mathbf{\Sigma} - \boldsymbol{P}_{\mathbf{X}}$ is a rank $k - n$ random orthogonal projection within the linear subspace $\mathrm{Span}(\boldsymbol{P}_{\mathbf{X}})^\perp$, and thus

$$\mathop{\mathbb{E}}_{\mathbf{\Sigma}|\mathbf{X}}[\mathbf{\Sigma} - \boldsymbol{P}_{\mathbf{X}}] = \frac{k-n}{d-n}(\mathbb{I}_d - \boldsymbol{P}_{\mathbf{X}}) \preceq \frac{2k}{d}\mathbb{I}_d, \tag{7.3}$$

where we used that $n \leq k/2$. The result eventually follows from plugging Equation (7.3) in Equation (7.2) with the tower rule, and from $\|\mathcal{M}(\mathbf{X})\| \leq \alpha^2$ a.s. The lower bound also derives from the definition of (λ, α)-sampler. □

The consequence of Lemma 7.3.9 is that, if the sampling mechanism is sufficiently correlated with the target distribution, i.e. $\mathbb{E}[\mathcal{M}(\mathbf{X})^T \mathbf{\Sigma} \mathcal{M}(\mathbf{X})]$ is lower bounded, then either 1) the sampling mechanism has a large variance in the linear subspace of the dataset, i.e. $\mathbb{E}\left[\mathcal{M}(\mathbf{X})^T P_{\mathbf{X}} \mathcal{M}(\mathbf{X})\right]$ is large, or 2) the norm of the sampling mechanism $\|\mathcal{M}(\mathbf{X})\|^2 \leq \alpha^2$ is very large and grows with the dimension.

Since the sampling mechanism is private, we will show that 1) cannot hold, which leads us to the desired result.

We detail point 1) by showing a relation between the IN statistic and the term $\mathbb{E}\left[\mathcal{M}(\mathbf{X})^T P_{\mathbf{X}} \mathcal{M}(\mathbf{X})\right]$.

Lemma 7.3.10 (IN case)**.** *Let* $\max(n, \log(2d/k)) \leq ck$*, where* $c > 0$ *is a universal constant. Assume additionally that* $\alpha \leq \frac{d}{4k}\lambda^2$*. Then, it holds that*

$$\mathbb{E}[\mathcal{M}(\mathbf{X})^T \boldsymbol{P}_{\mathbf{X}} \mathcal{M}(\mathbf{X})] \leq \frac{8n}{k} \mathbb{E}[\mathcal{M}(\mathbf{X})^T \hat{\boldsymbol{\Sigma}}_{\mathbf{X}} \mathcal{M}(\mathbf{X})] = \frac{8n}{k} \mathbb{E}[\mathcal{T}(\mathrm{X}^I)],$$

where $\hat{\boldsymbol{\Sigma}}_{\mathbf{X}} := \frac{1}{n}\sum_{i\in[n]}(\mathrm{X}^i)(\mathrm{X}^i)^T$.

The proof of Lemma 7.3.10 – which we defer to Section 7.3.3.2 for clarity – essentially relies on the fact that, in the regime of large k with $k/n \geq \Omega(1)$, we have that the empirical covariance matrix is highly concentrated as $\hat{\boldsymbol{\Sigma}} \approx \frac{k}{n}\boldsymbol{P}_{\mathbf{X}}$.

Eventually, we use a standard argument to show that the distributions of $\mathcal{T}(\mathrm{X}^I)$ and $\mathcal{T}(\mathbf{z})$ are (ε, δ)-indistinguishable, which allows us to control the expectation of the test in the two cases.

Lemma 7.3.11 (Privacy lemma)**.** *Let* $\mathcal{M}$ *be a* (α, λ)*-sampler* (ε, δ)*-DP. Then, we have that*

$$\mathbb{E}[\mathcal{T}(\mathrm{X}^I)] \leq e^{\varepsilon}\,\mathbb{E}[\mathcal{T}(\mathbf{z})] + \delta R + \int_{r\geq R} \Pr[\mathcal{T}(\mathrm{X}^I) \geq r]dr$$

where, for $R \geq Ck$ *with* $C > 0$ *a universal constant,*

$$\int_{r\geq R} \Pr[\mathcal{T}(\mathrm{X}^I) \geq r]dr \leq k\alpha^2 \exp\left(-R/C\right).$$

Proof. Denote $\tilde{\mathbf{X}}_i$ an adjacent data set of $\mathbf{X}$ obtained by replacing the sample on index i, i.e. X^i, with an i.i.d. copy. Since $\mathcal{M}$ is (ε, δ)-DP, we have,

$$\begin{aligned}\mathcal{T}(\mathbf{z}) = \langle \mathcal{M}(\mathbf{X}) \ , \ \mathbf{z}\rangle^2 =_{\text{dist.}} \left\langle \mathcal{M}(\tilde{\mathbf{X}}_i) \ , \ \mathrm{X}^i \right\rangle^2 &=_{\text{dist.}} \left\langle \mathcal{M}(\tilde{\mathbf{X}}_I) \ , \ \mathrm{X}^I \right\rangle^2 \\ &\approx_{(\varepsilon,\delta)} \left\langle \mathcal{M}(\mathbf{X}) \ , \ \mathrm{X}^I \right\rangle^2 = \mathcal{T}(\mathrm{X}^I).\end{aligned}$$

Thus, it holds that

$$\begin{aligned}\mathbb{E}\left[\mathcal{T}(\mathrm{X}^I)\right] &= \int_{0\leq r} \Pr[\mathcal{T}(\mathrm{X}^I) \geq r]dr \\ &\leq \int_{0\leq r\leq R} (e^{\varepsilon}\Pr[\mathcal{T}(\mathbf{z}) \geq r] + \delta)dr + \int_{R\leq r} \Pr[\mathcal{T}(\mathrm{X}^I) \geq r]dr \\ \mathbb{E}\left[\mathcal{T}(\mathrm{X}^I)\right] &\leq e^{\varepsilon}\,\mathbb{E}\left[\mathcal{T}(\mathbf{z})\right] + \delta R + \int_{R\leq r} \Pr[\mathcal{T}(\mathrm{X}^I) \geq r]dr.\end{aligned}$$

Now, considering that $\|\mathcal{M}(\mathbf{X})\| \leq \alpha$ almost surely, $\mathcal{T}(\mathrm{X}^I) = \left\langle \mathcal{M}(\mathbf{X}) \ , \ \mathrm{X}^I \right\rangle^2 \leq \alpha^2\|\mathrm{X}^I\|^2$. Since $\|\mathrm{X}^I\|^2 \sim \chi^2(k)$, we have that that $\int_{r\geq R}\Pr[\mathcal{T}(\mathrm{X}^I) \geq r]dr = \alpha^2\,\mathbb{E}[(\mathcal{T}(X)/\alpha^2 -$

$R\alpha^2)_+] \leq \alpha^2 \Pr(\chi^2(k+1) \geq R/\alpha^2)$. Thus, using Laurent-Massart (Laurent and Massart, 2000) tail bound on chi-squared distribution, and assuming $R \geq C\alpha^2(k+1)$ yields the result. □

We can now combine Lemmas 7.3.9 to 7.3.11 to prove Theorem 7.3.8.

Proof of Theorem 7.3.8. We start from Lemma 7.3.9, which writes

$$\mathbb{E}[\mathcal{T}(\mathbf{z})] \leq \mathbb{E}[\mathcal{M}(\mathbf{X})^T \boldsymbol{P}_X \mathcal{M}(\mathbf{X})] + \frac{2k\alpha^2}{d}.$$

Then, combining Lemmas 7.3.10 and 7.3.11 yields [1], $k \geq \max((4C)^2 n, 4C\log(8d/k))$, for $R \geq Ck$

$$\mathbb{E}[\mathcal{M}(\mathbf{X})^T \boldsymbol{P}_X \mathcal{M}(\mathbf{X})] \leq \frac{8n}{k}\left(e^{\varepsilon}\,\mathbb{E}[\mathcal{T}(\mathbf{z})] + \delta R + \alpha^2 k e^{-cR}\right).$$

Combining the previous inequalities and rearranging yields,

$$\left(1 - e^{\varepsilon}\frac{8n}{k}\right)\mathbb{E}[\mathcal{T}(\mathbf{z})] \leq \frac{8n}{k}\left(\delta R + k\alpha^2 \exp\left(-cR\right)\right) + \frac{2k\alpha^2}{d}.$$

Since $\mathbb{E}[\mathcal{T}(\mathbf{z})] \geq \lambda^2$, choosing $R = C\alpha^2 k\log(d)$ and absorbing the constant factors in C yields

$$\left(1 - e^{\varepsilon}\frac{8n}{k}\right)\lambda^2 \leq \frac{Cn}{k}\left(\alpha^2 k\log(d)\delta + \frac{k\alpha^2}{d}\right) + \frac{2k\alpha^2}{d}.$$

Eventually, for $n < k/(16e)$, and $\varepsilon \leq 1$, we have necessarily

$$\alpha^2 \geq \Omega\left(\min\left\{\frac{d\lambda^2}{k}, \frac{\lambda^2}{n\delta\log(d)}\right\}\right),$$

which completes the proof of Theorem 7.3.8.

□

7.3.3.2 Analysis of the In-Sample Test

This subsection serves as a proof of Lemma 7.3.10. We first note that, in expectation, the In-Sample test statistics can be expressed using the empirical covariance matrix.

Lemma 7.3.12. *Recall that $I \sim \mathcal{U}(\{1,\dots,n\})$, and denote $\hat{\boldsymbol{\Sigma}}_{\mathbf{X}} := \frac{1}{n}\sum_{i\in[n]}(\mathrm{X}^i)(\mathrm{X}^i)^T$. It holds that*

$$\mathbb{E}[\mathcal{T}(\mathrm{X}^I)] = \mathbb{E}[\mathcal{M}(\mathbf{X})^T \hat{\boldsymbol{\Sigma}}_{\mathbf{X}} \mathcal{M}(\mathbf{X})].$$

[1] Remark that, when the assumption $\alpha^2 \leq \frac{d\lambda^2}{4k}$ required for invoking Lemma 7.3.10 is not met, Theorem 7.3.8 trivially holds.

Proof. We simply use the tower rule with the randomness on I:

$$\begin{aligned}\mathbb{E}[\mathcal{T}(\mathrm{X}^I)] &= \mathbb{E}[\mathop{\mathbb{E}}_{I|_{\mathcal{M},\mathbf{X}}}[\mathcal{T}(\mathrm{X}^I)]] \\ &= \mathbb{E}[\frac{1}{n}\sum_{i\in[n]} \mathcal{M}(\mathbf{X})^T(\mathrm{X}^i)(\mathrm{X}^i)^T\mathcal{M}(\mathbf{X})] \\ &= \mathbb{E}[\mathcal{M}(\mathbf{X})^T\hat{\mathbf{\Sigma}}_{\mathbf{X}}\mathcal{M}(\mathbf{X})].\end{aligned}$$

□

We can now relate the IN sample test statistics with $\mathbb{E}[\mathcal{M}(\mathbf{X})^T\boldsymbol{P}_{\mathbf{X}}\mathcal{M}(\mathbf{X})]$ by using that, when the dataset is composed of n random samples of standard normal vectors in a k dimensional subspace, and when $n \ll k$, those samples are nearly orthogonal, and the empirical covariance matrix $\hat{\mathbf{\Sigma}}_{\mathbf{X}}$ is very concentrated around the orthogonal projection on the span of this dataset $\boldsymbol{P}_{\mathbf{X}}$, up to a $\frac{k}{n}$ multiplicative factor.

Lemma 7.3.13 (Random Matrix Concentration)**.** *There is a numerical constant $c > 0$, such that, if $n \leq ck$, we have conditionally on $\mathbf{\Sigma}$, that, with probability at least $1 - 2\exp(-ck)$ over the randomness of $\mathbf{X}$,*

$$\frac{k}{2n}\boldsymbol{P}_{\mathbf{X}} \preceq \hat{\mathbf{\Sigma}}_{\mathbf{X}} \preceq \frac{2k}{n}\boldsymbol{P}_{\mathbf{X}}.$$

Lemma 7.3.13 is a variation of standard matrix concentration results, and we refer to Section 7.A for the proof.

We now leverage the Lemma 7.3.13 and the Lemma 7.3.12 to conclude the analysis of the expectation of the in-sample test.

Proof of Lemma 7.3.10. Let G be the event $G = \left\{\frac{k}{2n}\boldsymbol{P}_{\mathbf{X}} \preceq \hat{\mathbf{\Sigma}}_{\mathbf{X}} \preceq \frac{2k}{n}\boldsymbol{P}_{\mathbf{X}}\right\}$. Using Lemma 7.3.13, for $n \leq ck$, $\Pr(G) \geq 1 - 2\exp(-ck) \geq \frac{1}{2}$. Since $\mathcal{M}(\mathbf{X})^T\hat{\mathbf{\Sigma}}_{\mathbf{X}}\mathcal{M}(\mathbf{X}) \geq 0$ a.s., it holds that

$$\mathbb{E}[\mathcal{M}(\mathbf{X})^T\hat{\mathbf{\Sigma}}_{\mathbf{X}}\mathcal{M}(\mathbf{X})] \geq \Pr(G)\mathop{\mathbb{E}}_{\mathbf{X},\mathcal{M}|_G}[\mathcal{M}(\mathbf{X})^T\hat{\mathbf{\Sigma}}_{\mathbf{X}}\mathcal{M}(\mathbf{X})] \geq \frac{k}{4n}\mathop{\mathbb{E}}_{\mathbf{X},\mathcal{M}|_G}[\mathcal{M}(\mathbf{X})^T\boldsymbol{P}_{\mathbf{X}}\mathcal{M}(\mathbf{X})].$$

Furthermore, by the law of total expectation gives

$$\begin{aligned}\mathbb{E}[\mathcal{M}(\mathbf{X})^T\boldsymbol{P}_{\mathbf{X}}\mathcal{M}(\mathbf{X})] = \Pr(G)&\mathop{\mathbb{E}}_{\mathbf{X},\mathcal{M}|_G}[\mathcal{M}(\mathbf{X})^T\boldsymbol{P}_{\mathbf{X}}\mathcal{M}(\mathbf{X})] \\ &+ (1-\Pr(G))\mathop{\mathbb{E}}_{\mathbf{X},\mathcal{M}|_{G^c}}[\mathcal{M}(\mathbf{X})^T\boldsymbol{P}_{\mathbf{X}}\mathcal{M}(\mathbf{X})],\end{aligned}$$

from which we deduce that

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathbf{X}|_G}\left[\mathcal{M}(\mathbf{X})^T\boldsymbol{P}_{\mathbf{X}}\mathcal{M}(\mathbf{X})\right] \geq &\frac{1}{2}\mathop{\mathbb{E}}_{\mathbf{X}}\left[\mathcal{M}(\mathbf{X})^T\boldsymbol{P}_{\mathbf{X}}\mathcal{M}(\mathbf{X})\right] + \Bigg(\frac{1}{2}\mathop{\mathbb{E}}_{\mathbf{X}}\left[\mathcal{M}(\mathbf{X})^T\boldsymbol{P}_{\mathbf{X}}\mathcal{M}(\mathbf{X})\right] \\ &- \mathop{\mathbb{E}}_{\mathbf{X}|_{G^c}}\left[\mathcal{M}(\mathbf{X})^T\boldsymbol{P}_{\mathbf{X}}\mathcal{M}(\mathbf{X})\right](1-\Pr(G))\Bigg).\end{aligned}$$

We now show that the right parenthesis is non-negative. We use that $\mathcal{M}$ is a (λ, α)-sampler and Lemma 7.3.9. It holds that

$$\begin{aligned}&\frac{1}{2}\mathop{\mathbb{E}}_{\mathbf{X}}\left[\mathcal{M}(\mathbf{X})^T \boldsymbol{P}_{\mathbf{X}} \mathcal{M}(\mathbf{X})\right] - \mathop{\mathbb{E}}_{\mathbf{X}|_{G^c}}\left[\mathcal{M}(\mathbf{X})^T \boldsymbol{P}_{\mathbf{X}} \mathcal{M}(\mathbf{X})\right](1-\Pr(G)) \\ \geq &\frac{1}{2}\lambda^2 - \left(\frac{k}{d} + 1 - \Pr(G)\right)\alpha^2.\end{aligned}$$

Since $1 - \Pr(G) \leq 2\exp(-ck)$, using $k \geq c\log(2d/k)$, and assuming $\alpha^2 \leq \frac{d}{4k}\lambda^2$ yields that the right parenthesis is non-negative. We thus have proven the desired result, i.e.

$$\mathbb{E}[\mathcal{T}(\mathrm{X}^I)] = \mathbb{E}[\mathcal{M}(\mathbf{X})^T \hat{\boldsymbol{\Sigma}}_{\mathbf{X}} \mathcal{M}(\mathbf{X})] \geq \frac{k}{8n}\,\mathbb{E}[\mathcal{M}(\mathbf{X})^T \boldsymbol{P}_{\mathbf{X}} \mathcal{M}(\mathbf{X})].$$

□

7.4 The Diagonal Case

This section provides the lower bound of the Bayes risk in the case of diagonal covariance matrices, whose eigenvalues are sampled from a scaled-inverse chi-squared distribution. We begin by discussing this choice of prior distribution, then state the main result under this choice of prior and discuss it, and eventually we prove it.

7.4.1 Prior Distribution

In this section, we consider $\boldsymbol{\Sigma} = \mathrm{Diag}(\boldsymbol{\sigma}^2)$ with $\boldsymbol{\sigma}^2 = (\boldsymbol{\sigma}_i^2)_{i\in[d]} \sim \mathsf{inv}\chi^2(1,1)^{\otimes d}$, and $\boldsymbol{\mu} \sim \mathcal{N}(0, \rho\mathbb{I})$, with $\rho > 0$. We denote this joint prior distribution as $(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim \mathcal{P}_2$.

We begin by recalling the definition of the $\mathsf{inv}\chi^2$ distribution.

Definition 7.4.1 ($\mathsf{inv}\chi^2(\tau,\nu)$)**.** The Scaled-inverse-chi-squared distribution with ν degrees of freedom and scale τ is the law with density, for $x \geq 0$, given by

$$f(x;\nu) \propto \frac{\exp\left[\frac{-\nu\tau^2}{2x}\right]}{x^{1+\nu/2}}.$$

On the prior distribution. This choice of diagonal distribution is not random: it reflects the failure mode of the unknown covariance algorithm of (Dagan et al., 2024) for a diagonal covariance matrix.

Indeed, when the covariance matrix is almost isotropic, the naive algorithm Algorithm 10 is optimal. And when the eigenvalues quickly decrease, as $\boldsymbol{\Sigma}_{ii} \propto 1/i^{2+c}$, with $c > 0$, then learning the first eigenvalues is enough to capture most of the signal as $(\sqrt{\boldsymbol{\Sigma}_{ii}})_{i=1}^{\infty}$ is summable. Thus, choosing $\boldsymbol{\Sigma} = \mathrm{Diag}((1/i^2)_{i\in[d]})$ is a pathological choice as relying on isotropic noise is highly sub-optimal, while the signal cannot be properly captured by considering only a few eigenvalues as required by the setting of a dimension-independent number of samples.

The $\mathsf{inv}\chi^2(1,1)$ distribution appears to be a continuous version of such a pathological covariance distribution. Indeed, consider the random variable sampled as follows:

1. Sample $I \sim \mathcal{U}(\{1,\dots,d\})$.
2. Output $\sigma^2_{\text{harmonic}} \sim \mathcal{U}([1/I^2, 1/(I+1)^2])$.

A quick computation shows that the density of σ_{harmonic} is $\rho(x) \propto x^{-3/2}$, which coincides with the tail density of the $\mathsf{inv}\chi^2(1,1)$ distribution. The choice of $\mathsf{inv}\chi^2(1,1)$ distribution is eventually motivated by its conjugate prior property, namely:

Fact 7.4.2. *Let $\sigma^2 \sim \mathsf{inv}\chi^2(\nu,\tau^2)$, then $\Pr(\sigma^2 > t) = C_{\nu,\tau}(t)t^{-\nu/2}$ with $C_{\nu,\tau}(t) \in [1, e^{1/2}]$ for $t \geq \nu\tau^2$, and $\sigma^2 = \frac{\nu\tau^2}{Y}$ with $Y \sim \chi^2(\nu)$.*
Additionally, if $\boldsymbol{X} = (\mathrm{X}^1,\dots,\mathrm{X}^n) \sim \mathcal{N}(0,\sigma^2)^{\otimes n}$, then it holds that:

$$\sigma^2 \mid \mathbf{X} \sim \mathsf{inv}\chi^2\left(\nu+n, \frac{\nu\tau^2 + \sum_{i\in[n]}(\mathrm{X}^i)^2}{\nu+n}\right).$$

Remark 7.4.3. While the $\mathsf{inv}\chi^2(1,1)$ distribution matches the intuitive worst-case distribution for eigenvalues of the harmonic series, it is of infinite expectation. Such a characteristic hardens the analysis, which will lead us to take a different approach from that of Section 7.3, by reducing the mean-estimation problem to the one of the problem of *selecting* the coordinates with the largest variance. Due to the heavy-tailed nature of the $\mathsf{inv}\chi^2(1,1)$ distribution, the exhibited test statistics are less interpretable, and the proof involves more involved analytical developments.

7.4.2 Main Result

We now state the main result that compares the Bayes risk of the covariance-agnostic and the covariance-aware settings.

Theorem 7.4.4. *Consider the prior distribution $\mathcal{P}_2$. There are universal constants $c, C > 0$, for which, if $\delta \leq c/n$, and $\varepsilon = 1$, and $\log(d)^2 \leq n \leq d^{1/8.5}$, we have that*

$$\mathfrak{R}_{\varepsilon,\delta}(\mathcal{P}_2, n) \geq \frac{\sqrt{d}}{n^{8.5}}\widetilde{\mathfrak{R}}_{\varepsilon,\delta}(\mathcal{P}_1, n).$$

It follows that the price of not knowing the covariance matrix implies a penalty on the accuracy of any algorithm that grows as a power of the dimension.

Lemma 7.4.5 (Optimal algorithms)**.** *The optimal algorithm under known covariance is the one of (Dagan et al., 2024), and ensures that*

$$\widetilde{\mathfrak{R}}_{\varepsilon,\delta}(\mathcal{P}_2, n) = \tilde{\Theta}\left(\frac{d^2}{n} + \frac{d^2\log(d)^2}{n^2\varepsilon^2}\right) = \tilde{\Theta}\left(\frac{d^2}{n}\right).$$

While the algorithm proposed in (Dagan et al., 2024) for the unknown covariance case

achieves the risk

$$\mathfrak{R}_{\varepsilon,\delta}(\mathcal{P}_2, n) \leq \tilde{\mathcal{O}}\left(\frac{d^2}{n} + \frac{d^3}{n^4\varepsilon^2}\right) = \tilde{\mathcal{O}}\left(\frac{d^3}{n^4}\right).$$

Importantly, the choice of normalization of the $\mathsf{inv}\chi^2(1,1)$ distribution implies that with high probability we have $\|\mathbf{\Sigma}\|_2^2 = \Theta(d^2)$. Phrasing the result of Lemma 7.4.5 with a standard unitary largest eigenvalue $\|\mathbf{\Sigma}\|_2 \approx 1$ requires dividing the error above by d^2.

Remark 7.4.6 (Privacy for free). Note that, for the prior distribution $\mathcal{P}_2$, the optimal risk under known covariance matches the risk for non-private algorithms. We say that we are in a *privacy for free* regime. This holds after a poly-logarithmic number of samples in the dimension d, which corresponds to the intrinsic dimension $(\operatorname{tr}\sqrt{\mathbf{\Sigma}})^2/\|\mathbf{\Sigma}\|_2$. In contrast, when the covariance matrix is unknown, the privacy-for-free regime happens only after a number of samples that grows as a power of the space dimension d.

Proof of Lemma 7.4.5. Since, the error of $\tilde{\mathcal{A}}$, the known covariance algorithm of (Dagan et al., 2024) is such that, for any fixed covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$ and expectation $\boldsymbol{\mu} \in \mathbb{R}^d$, with probability 9/10 over $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ and $\mathcal{A}$,

$$\|\tilde{\mathcal{A}}(\mathbf{X}, \mathbf{\Sigma}) - \boldsymbol{\mu}\|^2 \leq \tilde{\mathcal{O}}\left(\frac{\operatorname{tr}\mathbf{\Sigma}}{n} + \frac{\left(\operatorname{tr}\sqrt{\mathbf{\Sigma}}\right)^2 \log(1/\delta)}{n^2\varepsilon^2}\right),$$

while, for diagonal $\mathbf{\Sigma}$, the unknown covariance algorithm $\mathcal{A}$ of (Dagan et al., 2024) is such that, with probability 9/10 over $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ and $\mathcal{A}$,

$$\|\mathcal{A}(\mathbf{X}) - \boldsymbol{\mu}\|^2 \leq \tilde{\mathcal{O}}\left(\frac{\operatorname{tr}\mathbf{\Sigma}}{n} + \frac{\left(\operatorname{tr}\sqrt{\mathbf{\Sigma}}\right)^2 \log(1/\delta)}{n^2\varepsilon^2} + \frac{d\left(\operatorname{tr}\sqrt{\mathbf{\Sigma}}\right)^2 \log(1/\delta)^4}{n^4\varepsilon^4}\right).$$

Since, $\Pr(\boldsymbol{\sigma}_j^2 \geq t) \leq \sqrt{e}t^{-\frac{1}{2}}$ and $\Pr(\boldsymbol{\sigma}_j \geq t) \leq \sqrt{e}t^{-1}$ using Lemma 7.A.2, we have that

$$\Pr[\sum_{j\in[d]} \boldsymbol{\sigma}_j^2 \geq md^2] \leq K_{\frac{1}{2}}/m \qquad \text{and} \qquad \Pr[\sum_{j\in[d]} \boldsymbol{\sigma}_j \geq md] \leq \frac{K_1 \log(md)}{m},$$

taking m sufficiently large to ensure that all guarantees hold simultaneously with probability 2/3 yields the result. □

7.4.3 From Mean Estimation to Sampling and Selection

To prove Theorem 7.4.4, we proceed similarly to the diagonal setting, with a slight variation: instead of constructing a "sampler" only, we select the coordinates of this sampler with the largest error. We refer to such a mechanism as a *selection mechanism*.

Definition 7.4.7 ((k,ℓ)-selection mechanism)**.** Let $d, n \in \mathbb{N}$. A random algorithm $\mathcal{M}$ is a (k,ℓ)-selection mechanism with $k, \ell \in \mathbb{N}$, if for $\boldsymbol{\sigma} \sim \mathsf{inv}\chi^2(1,1)^{\otimes d}$, and $\boldsymbol{X} \sim \mathcal{N}(0, \mathrm{Diag}(\boldsymbol{\sigma}))$, the output $\mathbf{s} = \mathcal{M}(\mathbf{X}) \in \{0,1\}^d$ of the algorithm satisfies, with probability at least $1/3$:

1. **Boundedness:** $\sum_{i=1}^d \boldsymbol{b}_j \leq \ell$.
2. **Accuracy:** There is π a random permutation of $[d]$ such that, almost surely, $\boldsymbol{\sigma}^2_{\pi(1)} \geq \ldots \geq \boldsymbol{\sigma}^2_{\pi(d)}$, and
$$\sum_{i=1}^k \boldsymbol{s}_{\pi(j)} \geq \frac{k}{4}.$$

In other words, with probability $1/3$ over the randomness of the data and the algorithm, a selection mechanism outputs no more than ℓ indices, while selecting at least $k/4$ indices of the k largest variances.

Algorithm 12 Selection Mechanism – Diagonal Setting

Input: $\mathbf{X} \in \mathbb{R}^{n\times d}$.
Parameters: $\mathcal{A} : \mathbb{R}^{n\times d} \to \mathbb{R}^d$, τ
Sample $\boldsymbol{\mu} \sim \mathcal{N}(0, \rho \mathbb{I}_d)$.
Output $\boldsymbol{s} = \left(\mathbb{1}_{([\mathcal{A}(\mathbf{X}+\mu)]_j - \mu_j)^2 \geq \tau^2}\right)_{i\in[d]}$

Proposition 7.4.8 (From mean estimation to selection)**.** *Let $d \geq 114$, $n \in \mathbb{N}$, denote $\alpha = \mathfrak{R}_{\varepsilon,\delta}(\mathcal{P}_2, n)$ the Bayes optimal risk under unknown covariance for the prior $\mathcal{P}_2$, and let $\mathcal{A}$ be an (ε,δ)-DP algorithm that achieves this risk. For any $k \in [57, d/2]$, Algorithm 12 is (ε,δ)-DP and a (k,ℓ)-selection mechanism with $\ell = 49\frac{\alpha^2 k^2 n}{d^2}$.*

Proof. We begin by reasoning conditionally on $\boldsymbol{\sigma}$.

Error on each coordinate Consider Algorithm 12. Denoting $\mathbf{s} = \mathcal{M}(\mathbf{X})$ its outputs, it writes with $\boldsymbol{\mu} \sim \mathcal{N}(0, \rho\boldsymbol{\Sigma})$ and $\boldsymbol{X} \sim \mathcal{N}(0,\boldsymbol{\Sigma})^{\otimes n}$, $\mathbf{s} = \mathbb{1}_{|[\mathcal{A}(\mathbf{X}+\boldsymbol{\mu})-\boldsymbol{\mu}]_i| \geq \tau}$.

However, by gaussian conditioning (Lemma 7.A.4), it holds that, conditioned on $\boldsymbol{\Sigma}$ and $\mathbf{X} + \boldsymbol{\mu}$,
$$\boldsymbol{\mu}|_{\boldsymbol{\Sigma}, \mathbf{X}+\boldsymbol{\mu}} \sim \mathcal{N}\left(\overline{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}\right),$$
where $\overline{\boldsymbol{\mu}} := \left(\mathbb{I}_d + \frac{1}{n\rho}\boldsymbol{\Sigma}\right)^{-1} \frac{1}{n}\sum_{i\in[n]}(\mathrm{X}^i + \boldsymbol{\mu})$, and $\tilde{\boldsymbol{\Sigma}} := \left(\frac{1}{\rho}\mathbb{I}_d + n\boldsymbol{\Sigma}^{-1}\right)^{-1} = \mathrm{Diag}(\frac{1}{\rho} + n\boldsymbol{\sigma}_i^{-2})^{-1}$.

It follows that, when conditioning on $\boldsymbol{\Sigma}$, $\mathbf{X}+\boldsymbol{\mu}$ and $\mathcal{A}$, the covariance of $\overline{\boldsymbol{\mu}}$ is diagonal, and thus the $(\mathbf{s}_i)_{i\in[d]}$ are independent Bernoulli random variables of success probability,

for $i \in [d]$:

$$\begin{aligned}\Pr_{|_{\mathcal{A},\mathbf{X}+\boldsymbol{\mu},\boldsymbol{\Sigma}}}(\mathbf{s}_i = 1) &= \Pr_{|_{\mathcal{A},\mathbf{X}+\boldsymbol{\mu},\boldsymbol{\Sigma}}}\left(\left|\left[\mathcal{N}\left(\overline{\boldsymbol{\mu}} - \mathcal{A}(\mathbf{X}+\boldsymbol{\mu}), \frac{1}{n}\tilde{\boldsymbol{\Sigma}}\right)\right]_i\right| \geq \tau\right) \\ &\geq \Pr_{|_{\boldsymbol{\Sigma}}}\left(\left|\mathcal{N}\left(0, (\rho^{-1} + n\boldsymbol{\sigma}_i^{-2})^{-1}\right)\right| \geq \tau\right) \\ \Pr_{|_{\mathcal{A},\mathbf{X}+\boldsymbol{\mu},\boldsymbol{\Sigma}}}(\mathbf{s}_i = 1) &= 2\Phi\left(-\tau(n\boldsymbol{\sigma}_i^{-2} + \rho^{-1})^{\frac{1}{2}}\right) \qquad (7.4)\end{aligned}$$

Where Φ denotes the cdf of the standard normal distribution.

Selecting the largest coordinates. Assume that, conditionally on $\boldsymbol{\sigma}$, $\tau \leq (-\Phi^{-1}(1/6))\left(\frac{n}{\boldsymbol{\sigma}^2_{\pi(k)}} + \frac{1}{\rho}\right)^{-\frac{1}{2}}$. It follows from Equation (7.4) that for any $i \leq k$, $\Pr_{|_{\mathcal{A},\mathbf{X}+\boldsymbol{\mu},\boldsymbol{\Sigma}}}\left(\mathbf{s}_{\pi(i)} = 1\right) \geq 1/3$.

Next, using Hoeffding's inequality, it holds that

$$\Pr_{\boldsymbol{\mu}|_{\boldsymbol{\sigma},\mathcal{A},\mathbf{X}+\boldsymbol{\mu}}}\left(\sum_{i\leq k}\mathbf{s}_{\pi(i)} \geq k/4\right) \geq 1 - \exp(-2k/12^2) \geq 5/6,$$

where the last inequality holds for $k \geq \log(6)12^2/2 \approx 56$.

Selecting not too many coordinates. Considering the accuracy guarantees on $\mathcal{A}$, i.e.

$\Pr_{\mathbf{X}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma}),(\boldsymbol{\mu},\boldsymbol{\Sigma})\sim\mathcal{P}_2}\left(\|\mathcal{A}(\mathbf{X}) - \boldsymbol{\mu}\| \leq \alpha^2\right) \geq \frac{2}{3}$, it holds that

$$\Pr[\sum_{i\in[d]}\mathbf{s}_i \leq \ell] = \Pr[\sum_{i\in[d]}\mathbb{1}_{|[\mathcal{A}(\mathbf{X})]_i - \boldsymbol{\mu}_i|\geq\tau} \leq \ell] \geq \Pr[\|\mathcal{A}(X) - \mu\|^2 \leq \ell\tau^2] \geq \frac{2}{3} \quad \text{if } \ell = \alpha^2/\tau^2.$$

Fixing τ and ρ independently of $\boldsymbol{\sigma}$. We now choose ρ and τ such that $\tau \leq (-\Phi^{-1}(1/6))\left(\frac{n}{\boldsymbol{\sigma}^2_{\pi(k)}} + \frac{1}{\rho}\right)^{-\frac{1}{2}}$ holds with probability 5/6.

- *Fixing ρ.* Since $(\boldsymbol{\sigma}_i^2)_{i\in[d]} \sim \mathsf{inv}\chi^2(1,1)^{\otimes d}$, we have that $\Pr\left(\max_{i\in[d]}\boldsymbol{\sigma}_i^2 \geq t\right) \leq d\Pr(\boldsymbol{\sigma}_1^2 > t) \leq d\sqrt{e/t}$ for $t \geq 1$. So, setting $\rho = 12^2d^2/(ne)$, we have that $\rho^{-1} \leq n/\boldsymbol{\sigma}^2_{(k)}$ with probability at least $1 - 1/12$.
- *Fixing τ.* Let $k \in \{1, \ldots, d/2\}$, and set $\tau \leq \frac{d}{7k\sqrt{n}} \leq \frac{d}{4k\sqrt{2n}}\left(-\Phi^{-1}(1/6)\right)$. Then Lemma 7.A.3 ensures that $\tau \leq \left(-\Phi^{-1}(1/6)\right)\sqrt{\frac{\boldsymbol{\sigma}_{(k)}}{2n}}$ with probability at least $1 - 2/k$.

Thus, for $k \geq 24$, with probability at least $1 - 24/2 - 1/12 = 5/6$ over the randomness of $\boldsymbol{\sigma}$, the desired condition $\tau \leq (-\Phi^{-1}(1/6))\left(\frac{n}{\boldsymbol{\sigma}^2_{\pi(k)}} + \frac{1}{\rho}\right)^{-\frac{1}{2}}$ holds.

The result follows by a union bound and rounding constants: for $k \in [57, d/2]$, and $\tau = \frac{d}{7k\sqrt{n}}$, Algorithm 12 is a (k,ℓ)-selection mechanism, and $\ell = \alpha^2/\tau^2 = 49\frac{\alpha^2k^2n}{d^2}$. Additionally, by post-processing, Algorithm 12 is (ε,δ)-DP.

□

7.4.4 Lower Bound on the Accuracy of the Selection Mechanism

We have shown that any differentially private algorithm of accuracy α with probability $2/3$ under the prior distribution $\mathcal{P}_2$ is associated with a selection mechanism. We now show that the sampler cannot be simultaneously differentially private and sufficiently accurate.

Theorem 7.4.9 (Lower bound on the Sampling Accuracy)**.** *There are universal constants $c, C > 0$, such that, for any $n \geq C\log(d)^2$, any $k \geq Cn\log(nd)^{1/(1-a)}$, any $\ell \leq c\frac{d^a k^{1-a}}{\log(d)}$, for any $a \in (\frac{1}{2}, 1)$, if $\mathcal{M}$ is a (k,ℓ)-selection mechanism (Definition 7.4.7) that is (ε,δ)-DP, with $\varepsilon \leq 1$ and $\delta \leq \frac{c}{n}$, then, necessarily*

$$n \geq \Omega\Big(k^{\frac{1-a}{2+a}}\Big).$$

This, combined with the existence of (k,ℓ)-selection mechanism as a consequence of the mean-estimation mechanism, will allow us to show the main Theorem 7.4.4.

Proof. Let $\mathcal{A}$ be an (ε,δ)-DP algorithm that achieves the Bayes risk under unknown covariance $\alpha = \mathfrak{R}_{\varepsilon,\delta}(\mathcal{P}_2, n)$. Then, using Proposition 7.4.8, for any $k \in [57, d/2]$, their exists an (ε,δ)-DP (k,ℓ)-selection mechanism with $\ell = 49\frac{\alpha^2 k^2 n}{d^2}$.

It follows from Theorem 7.4.9, that, if $n \geq \Omega(\log(d)^2)$, $k \geq n\log(nd)$, choosing k such that

$$\ell = 49\frac{\alpha^2 k^2 n}{d^2} \leq c\frac{d^a k^{1-a}}{\log(d)} \iff k = c\left(\frac{d^{2+a}}{\alpha^2 n\log(d)}\right)^{1/(1+a)},$$

yields that, necessarily

$$n \geq ck^{\frac{1-a}{2+a}} = \frac{d^{\frac{1-a}{1+a}}}{(\alpha^2 n\log(d))^{\frac{1-a}{(2+a)(1+a)}}},$$

which implies that

$$\alpha^2 \geq \Omega\left(\frac{d^{2+a}}{\log(d) n^{1+\frac{(2+a)(1+a)}{1-a}}}\right).$$

Choosing (arbitrarily) a arbitrarily close to $1/2$ yields $\alpha^2 \geq \Omega\Big(\frac{d^2}{n}\cdot\frac{\sqrt{d}}{n^{8.5}}\Big)$, which is the stated result. One can check that, for $n \leq \mathcal{O}(d^{8.5})$, the assumption on $d/2 \geq k \geq \Omega(n\log(nd))$ is satisfied. □

7.4.4.1 A Private Selection Lower Bound

To prove Theorem 7.4.9, similarly to the diagonal case, we consider a test statistics that measure the correlation between the selection mechanism $\mathcal{M}$ and the empirical covariance of the dataset. Let $\boldsymbol{\sigma} \sim \mathsf{inv}\chi^2(1,1)^{\otimes d}$, and $\mathbf{X} \sim \mathcal{N}(0, \mathrm{Diag}(\boldsymbol{\sigma}))^{\otimes n}$. We consider for $\boldsymbol{z} \in \mathbb{R}^d$, the test

$$\mathcal{T}(\boldsymbol{z}) := \sum_{j\in[d]} \mathcal{T}_j(\boldsymbol{z}), \quad \text{where} \quad \mathcal{T}_j(\boldsymbol{z}) := \left(\frac{\boldsymbol{z}_j^2}{\boldsymbol{\sigma}_j^2} - 1\right)\boldsymbol{\sigma}_j^a \mathcal{M}(\mathbf{X})_j, \tag{7.5}$$

where $a \in (0,1)$ is a scale parameter which we will fix later. Similarly to Section 7.3, we compare the distribution of the two statistics:

- The *OUT sample case* in which we consider $\mathcal{T}(\mathbf{z})$ with $\mathbf{z} \sim \mathcal{N}(0, \mathrm{Diag}(\boldsymbol{\sigma}))$.
- The *IN sample case* in which we consider $\mathcal{T}(\mathrm{X}^I)$ with $I \sim \mathcal{U}(\{1,\dots,n\})$.

We begin by stating the analysis of the statistics of the Out- and In-sample cases, which, by combining them, will give the proof of Theorem 7.4.9. Then, in section Section 7.4.4.2, we analyze the Out-sample case, and the In-sample case in Section 7.4.4.3.

Lemma 7.4.10 (In Sample Test is Large)**.** *There are absolute constants $c, C > 0$, such that, for $k, d \geq 0$, any (k,ℓ)-selection mechanism $\mathcal{M}$, with $\ell \leq c\frac{d^a k^{1-a}}{\log d}$, if $n \geq C\log(d)^2$ and $k \geq Cn\log(nd)^{1/(1-a)}$, then*

$$\Pr_{\boldsymbol{\sigma}^2,\mathbf{X},I,\mathcal{M}(\mathbf{X})}\left[\mathcal{T}(\mathbf{X}^I) \geq \frac{cd^a k^{1-a}}{n^{1-a}}\right] = \Omega\left(\frac{1}{n}\right).$$

Lemma 7.4.11 (Out Sample Test is Small)**.** *Let $a \in (1/2, 1)$. There is $C(a) > 0$, such that, for any $n, d \geq 0$,*

$$\Pr_{\boldsymbol{\sigma},\mathbf{z}\sim\mathcal{N}(0,\mathrm{Diag}(\boldsymbol{\sigma}_j^2))}\left[\mathcal{T}(\mathbf{z}) \geq C(a)d^a n^{2a+1}\right] \leq \frac{1}{n^2}.$$

Combining Lemma 7.4.10 and Lemma 7.4.11 allows to show Theorem 7.4.9.

Proof of Theorem 7.4.9. Let $c > 0$ and $C(a)$ the constants of Lemmas 7.4.10 and 7.4.11. Assume that

$$\frac{k^{1-a}}{n^{1-a}} \geq \frac{C(a)}{c}n^{2a+1}, \text{ i.e } k \geq \left(\frac{C(a)}{c}\right)^{1/(1-a)} n^{\frac{2+a}{1-a}}.$$

By Lemmas 7.4.10 and 7.4.11 and since the mechanism $\mathcal{M}$ is (ε,δ)-DP we have that

$$\Omega\left(\frac{1}{n}\right) = \Pr\left[\mathcal{T}(X^I) \geq \frac{cd^a k^{1-a}}{n}\right] \leq e^{\varepsilon}\Pr\left[\mathcal{T}(Z) \geq \frac{cd^a k^{1-a}}{n}\right] + \delta = O\left(\frac{e^{\varepsilon}}{n^2}\right) + \delta.$$

Thus, for $\delta = c/n$ with c sufficiently small, we obtain a contradiction. It follows that necessarily

$$n = \Omega\left(k^{\frac{1-a}{2+a}}\right).$$

□

7.4.4.2 Out Sample Analysis

In this section, we analyze the test statistic T on a random sample $\mathbf{z} \sim \mathcal{N}(0, \mathrm{Diag}(\boldsymbol{\sigma}^2))$.

Lemma 7.4.12 (OUT case expectation and variance)**.** *Consider $\mathcal{T}$ the test defined in Equation (7.5). Then, it holds conditionally on $\boldsymbol{\sigma}$, $\mathbf{X}$ and $\mathcal{M}$ that, for*

$\mathbf{z} \sim \mathcal{N}(0, \mathrm{Diag}(\boldsymbol{\sigma}))$,

$$\mathop{\mathbb{E}}_{\mathbf{z}|_{\boldsymbol{\sigma},\mathbf{X},\mathcal{M}}}[\mathcal{T}(\mathbf{z})] = 0 \quad \textit{and} \quad \mathrm{Var}_{|\boldsymbol{\sigma},\mathbf{X},\mathcal{M}}[\mathcal{T}(\mathbf{z})] \leq 2\sum_{j\in[d]} \boldsymbol{\sigma}_j^{2a}\mathcal{M}(\mathbf{X})_j.$$

Proof. Since $\mathbb{E}[\mathbf{z}_j^2] = \boldsymbol{\sigma}_j^2$, we have $\mathbb{E}_{|\boldsymbol{\sigma},\mathbf{X},\mathcal{M}}[\mathcal{T}(\mathbf{z})] = 0$. Furthermore, $\mathbf{z}_j^2/\boldsymbol{\sigma}_j^2 \sim \chi^2(1)$. Thus,

$$\mathop{\mathbb{E}}_{|\boldsymbol{\sigma},\mathbf{X},\mathcal{M}}\left[\mathcal{T}_j(\mathbf{z})^2\right] = \mathop{\mathbb{E}}_{|\boldsymbol{\sigma},\mathbf{X},\mathcal{M}}\left[\left(\frac{\mathbf{z}_j^2}{\boldsymbol{\sigma}_j^2} - 1\right)^2\right]\boldsymbol{\sigma}_j^{2a}\mathcal{M}(\mathbf{X})_j^2 = 2\boldsymbol{\sigma}_j^{2a}\mathcal{M}(\mathbf{X})_j.$$

Since the $(\mathcal{T}_j(\mathbf{z}))_{j\in[d]}$ are independent conditionally on $\boldsymbol{\sigma}$, $\mathbf{X}$ and $\mathcal{M}$, using $\mathrm{Var}(\mathcal{T}(\mathbf{z})) = \sum_{i\in[d]}\mathrm{Var}(\mathcal{T}_j(\mathbf{z}))$ gives the result. □

Proof of Lemma 7.4.11. We decompose the probability based on the scale of $\sum_{j\in[d]}\boldsymbol{\sigma}_j^{2a}$, for any $m, t \geq 0$,

$$\mathbb{E}[\mathbb{1}_{\mathcal{T}(\mathbf{z})\geq t}] \leq \mathbb{E}\left[\mathbb{1}_{\sum_{j\in[d]}\boldsymbol{\sigma}_j^{2a}\geq md^{2a}}\right] + \mathbb{E}\left[\mathbb{1}_{\mathcal{T}(\mathbf{z})\geq t}\mathbb{1}_{\sum_{j\in[d]}\boldsymbol{\sigma}_j^{2a}\leq md^{2a}}\right]. \tag{7.6}$$

We control the first term using Lemma 7.A.2: since $\boldsymbol{\sigma}^2 \sim \mathsf{inv}\chi^2(1,1)^{\otimes d}$, we have for $t \geq 1$ that $\Pr(\boldsymbol{\sigma}_j^2 \geq t) \leq \sqrt{e/t}$ (cf. Fact 7.4.2), and thus $\Pr(\boldsymbol{\sigma}_j^{2a} \geq t) \leq \sqrt{e}t^{-1/2a}$. Using Lemma 7.A.2 with $\alpha = 1/2a$, it holds for $a \geq \frac{1}{2}$ and $m \geq 1$, that

$$\Pr\left(\sum_{j\in[d]}\boldsymbol{\sigma}_j^{2a} \geq md^{2a}\right) \leq K_{2a}m^{-1/2a}.$$

The second term is controlled by combining Lemma 7.4.12 with Chebyshev inequality (conditioned on $\boldsymbol{\sigma}$):

$$\mathbb{E}\left[\mathbb{1}_{\mathcal{T}(\mathbf{z})\geq t}\mathbb{1}_{\sum_{j\in[d]}\boldsymbol{\sigma}_j^{2a}\leq md^{2a}}\right] \leq \mathbb{E}\left[\mathbb{1}_{\sum_{j\in[d]}\boldsymbol{\sigma}_j^{2a}\leq md^{2a}}\frac{2\sum_{j\in[d]}\boldsymbol{\sigma}_j^{2a}\mathcal{M}(\mathbf{X})_j}{t^2}\right] \leq \frac{2md^{2a}}{t^2},$$

where we used that $\mathcal{M}(\mathbf{X}) \in \{0,1\}^d$. Combining the two in Equation (7.6) yields, that

$$\Pr(\mathcal{T}(\mathbf{z}) \geq t) \leq \frac{2md^{2a}}{t^2} + K_{2a}m^{-1/2a}.$$

We now proceed with a few algebraic manipulations: minimizing the upper bound in m with $m_t^{1+\frac{1}{2a}} = \frac{K_{2a}t^2}{4ad^{2a}}$, yields to

$$\Pr(\mathcal{T}(\mathbf{z}) \geq t) \leq m_t^{-1/2a}K_{2a}\left(1 + \frac{1}{2a}\right) = \left(\frac{K_{2a}t^2}{4ad^{2a}}\right)^{-1/(2a+1)}K_{2a}\frac{2a+1}{2a}.$$

Requiring the probability to be smaller than $1/n^2$, leads to choosing $t \geq \frac{\sqrt{2}(2a+1)^{a+\frac{1}{2}}K_{2a}^a}{(2a)^a}d^a n^{2a+1}$. Denoting $C(a) = \frac{\sqrt{2}(2a+1)^{a+\frac{1}{2}}K_{2a}^a}{(2a)^a}$ yields

$$\Pr\left(\mathcal{T}(\mathbf{z}) \geq C(a)d^a n^{2a+1}\right) \leq \frac{1}{n^2},$$

which is the desired result.

□

7.4.4.3 In Sample Analysis

In this section, we analyze the behavior of the test statistic in the in-sample case, i.e., the distribution of $\mathcal{T}(\mathrm{X}^I)$, where X^I is a random sample from the dataset X, and complete the proof of Lemma 7.4.10.

Lemma 7.4.13 (IN case expectation and variance)**.** *Denote* $\mathbf{q}_j = \frac{1}{n}\sum_{i\in[n]}(\mathrm{X}^i_j)^2$ *the empirical variance of the dataset* $\mathbf{X}$ *on coordinate* j *and* $\mathbf{q} = (\mathbf{q}_1, \dots, \mathbf{q}_n)$ *the vector of empirical variances. We have conditionally on* $\mathbf{X}$ *and* $\mathcal{M}(\mathbf{X})$ *that*

$$\mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2, I|\mathbf{X},\mathcal{M}}\left[\mathcal{T}(\mathrm{X}^I)\right] \geq \sum_{j\in[d]} \frac{\mathbf{q}_j^{a/2}}{n^{1-a}}\left(\frac{\mathbf{q}_j(1-a)-1}{\mathbf{q}_j+\frac{1}{n}}\right)\mathcal{M}(\mathbf{X})_j.$$

Additionally, if $n \geq 7$*, for each fixed* $i \in [n]$*,*

$$\mathrm{Var}_{\boldsymbol{\sigma}^2|\mathbf{X},\mathcal{M}}\left[\mathcal{T}(\mathrm{X}^i)\right] \leq 4\sum_{j\in[d]}(\mathrm{X}^i_j)^4\left(\frac{n+1}{1+n\mathbf{q}_j}\right)^{2-a} + 4\left(\frac{1+n\mathbf{q}_j}{n+1}\right)^a.$$

Proof. We first prove the lower bound on the expected value of $\mathcal{T}(\mathrm{X}^I)$. By linearity of expectation, we can consider each term $\mathcal{T}_j(\mathrm{X}^I) = \left(\frac{(\mathrm{X}^I_j)^2}{\boldsymbol{\sigma}^2_j} - 1\right)\boldsymbol{\sigma}^a_j\mathcal{M}(\mathbf{X})_j$ separately. Conditionally on $\mathcal{M}(\mathbf{X})$ and X, we have that

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2, I|\mathcal{M},\mathbf{X}}\left[\mathcal{T}_j(\mathrm{X}^I)\right] &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2|\mathbf{X},\mathcal{M}}\left[\left(\frac{\mathbb{E}_I[(\mathrm{X}^I_j)^2]}{\boldsymbol{\sigma}^2_j} - 1\right)\boldsymbol{\sigma}^a_j\mathcal{M}(\mathbf{X})_j\right] \\ &= \left(\mathbf{q}_j \mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2|\mathbf{X}}\left[\boldsymbol{\sigma}^{a-2}_j\right] - \mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2|\mathbf{X}}\left[\boldsymbol{\sigma}^a\right]\right)\mathcal{M}(\mathbf{X})_j.\end{aligned}$$

Recall that $\sigma \sim \mathsf{inv}\chi^2(1,1)$ is conjugate prior of $\mathcal{N}(0,\sigma^2)$. Thus, using Fact 7.4.2, it holds that $\boldsymbol{\sigma}^2_j|\mathbf{X} \sim \mathsf{inv}\chi^2(\nu,\tau^2)$ with $\nu = n+1$ and $\tau^2 = \frac{1+n\mathbf{q}_j}{\nu}$, i.e. $\boldsymbol{\sigma}^2_j|\mathbf{X} \sim \frac{\nu\tau^2}{Y}$ with $Y \sim \chi^2(\nu)$, whose moment have closed-form formula (cf. Proposition 7.A.8). It follows that

$$\mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2|\mathbf{X}}\left[\boldsymbol{\sigma}^{a-2}_j\right] = \nu\tau^2\,\mathbb{E}[Y^{-(a-2)/2}] = \nu\tau^2\frac{2^{-(a-2)/2}\Gamma(\frac{\nu}{2}+1-\frac{a}{2})}{\Gamma(\frac{\nu}{2})},$$

and

$$\mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2|\mathbf{X}}\left[\boldsymbol{\sigma}^{a}_j\right] = \nu\tau^2\,\mathbb{E}[Y^{-a/2}] = \nu\tau^2\frac{2^{-a/2}\Gamma(\frac{\nu}{2}-\frac{a}{2})}{\Gamma(\frac{\nu}{2})}.$$

Since $\Gamma(\frac{\nu}{2}-\frac{a}{2}+1) = (\frac{\nu}{2}-\frac{a}{2})\Gamma(\frac{\nu}{2}-\frac{a}{2})$, and after some manipulation, we obtain

$$\begin{aligned}\mathbf{q}_j \mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2}\left[\boldsymbol{\sigma}^{a-2}_j \mid X\right] - \mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2}\left[\boldsymbol{\sigma}^a \mid X\right] &= (\nu\tau^2)^{a/2}2^{-a/2}\left(\frac{\mathbf{q}_j(\nu-a)}{\nu\tau^2}-1\right)\frac{\Gamma(\frac{\nu-a}{2})}{\Gamma(\frac{\nu}{2})} \\ &\geq (\nu\tau^2)^{a/2}2^{-a/2}\left(\frac{\mathbf{q}_j(\nu-a)}{\nu\tau^2}-1\right)\left(\frac{\nu}{2}\right)^{-a/2} \\ &= \tau^a\left(\frac{\mathbf{q}_j(\nu-a)}{\nu\tau^2}-1\right),\end{aligned}$$

where the inequality follows from Gautschi's inequality (cf. Proposition 7.A.9) with $x = \frac{\nu}{2}-1$ and $s = 1-\frac{a}{2}$. Using the closed form expressions for ν and τ^2 we write the

above expression in terms of $\mathbf{q}_j$, n, and a

$$\begin{aligned}\tau^a\left(\frac{\mathbf{q}_j(\nu-a)}{\nu\tau^2}-1\right) &= \tau^{a-2}\left(\frac{\mathbf{q}_j(\nu-a)-\nu\tau^2}{\nu}\right)\\ &= \left(\frac{1+n\mathbf{q}_j}{n+1}\right)^{\frac{a}{2}-1}\left(\frac{\mathbf{q}_j(1-a)-1}{n+1}\right) = \frac{(n+1)^{\frac{a}{2}}}{(1+n\mathbf{q}_j)^{1-\frac{a}{2}}}(\mathbf{q}_j(1-a)-1)\\ &\geq \frac{n^a\mathbf{q}_j^{a/2}}{1+n\mathbf{q}_j}(\mathbf{q}_j(1-a)-1).\end{aligned}$$

Putting it all together, we have

$$\mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2,I|\mathbf{X},\mathcal{M}}\left[\mathcal{T}_j(\mathrm{X}^I)\right] \geq \sum_{j\in[d]}\frac{\mathbf{q}_j^{a/2}}{n^{1-a}}\left(\frac{\mathbf{q}_j(1-a)-1}{\mathbf{q}_j+\frac{1}{n}}\right)\mathcal{M}(\mathbf{X})_j$$

which completes the proof of the first part of the expectation.

Controlling the variance. We now prove the upper bound on the variance of $\mathcal{T}(\mathrm{X}^i)$ for fixed $i\in[n]$. Note that conditionally on $\mathbf{X}$ and $\mathcal{M}(\mathbf{X})$, the random variables $(\mathcal{T}_j(\mathrm{X}^i))_{j\in[d]}$ are independent from each other (they only depend on the $\boldsymbol{\sigma}_j^2$, which are independent). Thus,

$$\mathrm{Var}_{\boldsymbol{\sigma}^2|\mathbf{X},\mathcal{M}}\left[\mathcal{T}(\mathrm{X}^i)\right] = \sum_{j\in[d]}\mathrm{Var}_{\boldsymbol{\sigma}^2|\mathbf{X},\mathcal{M}}[\mathcal{T}_j(\mathrm{X}^i)] \leq \sum_{j\in[d]}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2|\mathbf{X},\mathcal{M}}[\mathcal{T}_j(\mathrm{X}^i)^2].$$

Recall $\mathcal{T}_j(\mathrm{X}^I) = \left(\frac{(\mathrm{X}_j^I)^2}{\boldsymbol{\sigma}_j^2}-1\right)\boldsymbol{\sigma}_j^a\mathcal{M}(\mathbf{X})_j$ for $j\in[d]$. Similarly to the expectation analysis, we can write

$$\mathcal{T}_j(\mathrm{X}^i)^2 \leq \left(\frac{(\mathrm{X}_j^i)^2}{\boldsymbol{\sigma}_j^2}-1\right)^2\boldsymbol{\sigma}_j^{2a} = \left(\frac{(\mathrm{X}_j^i)^2Y^{1-a/2}}{(\nu\tau^2)^{1-a/2}}-\left(\frac{\nu\tau^2}{Y}\right)^{a/2}\right)^2 \leq \frac{(\mathrm{X}_j^i)^4Y^{2-a}}{(\nu\tau^2)^{2-a}}+\left(\frac{\nu\tau^2}{Y}\right)^a.$$

Expressing again the moment of Y using the gamma function and using Stirling approximation (cf. Propositions 7.A.8 and 7.A.9) a quick computation shows that, for $p\in\{-a,2-a\}$, and $\nu\geq 8$,

$$\mathbb{E}[(Y/\nu)^p] = \frac{2^p\Gamma(\frac{\nu}{2}+p)}{\Gamma(\nu/2)\nu^p} \leq 4.$$

and thus, since $\nu=n+1$ and $\tau^2=(1+n\mathbf{q}_j)/(n+1)$,

$$\mathop{\mathbb{E}}_{\boldsymbol{\sigma}|\mathbf{X},\mathcal{M}}[\mathcal{T}_j(\mathrm{X}^i)^2] \leq 4(\mathrm{X}_j^i)^4\left(\frac{n+1}{1+n\mathbf{q}_j}\right)^{2-a}+4\left(\frac{1+n\mathbf{q}_j}{n+1}\right)^a$$

Summing over each $j\in[d]$ yields the desired variance bound. □

Now that we have computed the conditional expectation and the variance of the test in the In-sample case, we provide a constant probability lower and upper bound on the conditional expectation and variance over the randomness of $(\mathbf{X},\mathcal{M}(\mathbf{X}))$. This will allow us to conclude the analysis of the In-sample test statistic.

Lemma 7.4.14 ("Good" event)**.** *There exists numerical constants $c_1, c_2, c_3, c_4 > 0$ such that, we can define as G the set of $(\mathbf{X}, \mathcal{M}(\mathbf{X}))$ satisfying:*

$$\sum_{j\in[d]} \frac{\mathbf{q}_j^{a/2}}{n^{1-a}}\left(\frac{\mathbf{q}_j(1-a)-1}{\mathbf{q}_j+\frac{1}{n}}\right)\mathcal{M}(\mathbf{X})_j \geq c_1\frac{d^a k^{1-a}}{n^{1-a}}$$

$$\sum_{j\in[d]} (\mathrm{X}_j^i)^4\left(\frac{n+1}{1+n\mathbf{q}_j}\right)^{2-a} + \left(\frac{1+n\mathbf{q}_j}{n+1}\right)^a \leq c_2\log(nd)^2 d^{2a},$$

and such that, for any $n \geq c_3\log(d)^2$, $d \geq k \geq c_3$, we have

$$\Pr_{\boldsymbol{\sigma},\mathbf{X},\mathcal{M}}(G) \geq c_4.$$

Proof. We first argue that for all fixed $\boldsymbol{\sigma}^2 \in \mathbb{R}^d$ and all $i \in [n]$ and $j \in [d]$ we have that $(\mathrm{X}_j^i)^2, \mathbf{q}_j \approx \boldsymbol{\sigma}_j^2$ with high probability. We then argue that the bounds hold with constant probability when $(\mathrm{X}^i)^2$ and $\mathbf{q}$ are (approximately) replaced with $\boldsymbol{\sigma}^2$.

Let $C > 1$ be a sufficiently large absolute constant. Condition on $\boldsymbol{\sigma}^2$ and let B_0 denote the event:

$$B_0 = \left\{\exists i \in [n], \exists j \in [d] : \left\{(\mathrm{X}_j^i)^2 \notin \boldsymbol{\sigma}_j^2[1 \pm C\log(nd)]\right\} \cup \left\{\mathbf{q}_j \notin \boldsymbol{\sigma}_j^2[1 \pm Cn^{-1/2}\log(nd)]\right\}\right\}.$$

By standard concentration bounds for chi-squared random variables $\Pr_{\mathbf{X}|_{\boldsymbol{\sigma}^2}}[B_0] = o_{n\to\infty}(1)$.

Now, conditioned on B_0^c, for each $j \in [d]$ we have for $n \geq c\log^2(d)$, with $c > 0$ an absolute constant, that

$$\frac{\mathbf{q}_j^{a/2}}{n^{1-a}}\left(\frac{\mathbf{q}_j(1-a)-1}{\mathbf{q}_j+n^{-1}}\right) \geq \frac{\boldsymbol{\sigma}_j^a}{2n^{1-a}}\left(\frac{\boldsymbol{\sigma}_j^2(1-a)-2}{\boldsymbol{\sigma}_j^2}\right),$$

and that, absorbing numerical constants in C,

$$\begin{aligned}(\mathrm{X}_j^i)^4\left(\frac{n+1}{1+n\mathbf{q}_j}\right)^{2-a} + \left(\frac{1+n\mathbf{q}_j}{n+1}\right)^a &\leq C^2\log(nd)^2\boldsymbol{\sigma}_j^{2a}\left(\frac{(n+1)\boldsymbol{\sigma}_j^2}{1+n\boldsymbol{\sigma}_j^2/2}\right)^{2-a} + \left(\frac{1+2n\boldsymbol{\sigma}_j^2}{n+1}\right)^a \\ &\leq \frac{C}{(n+1)^a} + C\log(nd)^2\boldsymbol{\sigma}_j^{2a}.\end{aligned}$$

We complete the proof by arguing that the bound holds with constant probability over $\boldsymbol{\sigma}^2$.

We consider the following events:

1. Let $B_1 = \left\{\boldsymbol{\sigma}_{\min}^2\right\} \leq 1/s\log(d)$. Since $1/\boldsymbol{\sigma}_j^2 \sim \chi^2(1)$, we have using Proposition 7.A.7 that $\Pr(1/\boldsymbol{\sigma}_j^2 \geq 1+t) \leq ce^{-t/2}$ (with $c > 0$ a numerical constant) and thus $\Pr(B_1) \leq ce^{-s} \leq 1/10$ by a union bound and $s > \log(10c)$.
2. Let $B_2 = \left\{\boldsymbol{\sigma}_{(k)}^2 \leq (d/4k)^2\right\}$. By Lemma 7.A.3 we have that $\Pr(B_2) \leq \frac{2}{k}$.

3. Let B_3^c be the event that the (k,ℓ)-selection mechanism $\mathcal{M}$ is *accurate* and *bounded*. By definition of $\mathcal{M}$, we have that $\Pr(B_3) \leq 1/3$.

We have that, for $n, k \geq c$, where $c > 0$ is a numerical constant, that $\Pr[B_0 \cup B_1 \cup B_2 \cup B_3] \leq o_n(1) + 1/10 + 2/k + 1/3 \leq 1/2$.

Now, conditioned on $\overline{B_0 \cup B_1 \cup B_2 \cup B_3}$, we have, with $c, C > 0$ numerical constants,

$$\begin{aligned}\sum_{j\in[d]} \frac{\boldsymbol{\sigma}_j^a}{n^{1-a}}\left(\frac{\boldsymbol{\sigma}_j^2(1-a)-1}{\boldsymbol{\sigma}_j^2}\right)\mathcal{M}(\mathbf{X})_j \geq& \frac{1}{n^{1-a}}\sum_{j\in[k]} \boldsymbol{\sigma}_{(j)}^a\left(\frac{\boldsymbol{\sigma}_{(j)}^2(1-a)-1}{\boldsymbol{\sigma}_{(j)}^2}\right)\mathcal{M}(\mathbf{X})_j \\ &- C\ell\left(\frac{\log d}{n^{1-a}}\right) \\ &\text{using the boundedness of } \mathcal{M} \text{ and } B_3^c \\ \geq& c\frac{d^a k^{1-a}}{n^{1-a}} - C\ell\frac{\log d}{n^{1-a}} \\ &\text{using the accuracy of } \mathcal{M},\ B_2^c \text{ and } 1-a \geq \tfrac{1}{2} \\ \geq& c\frac{d^a k^{1-a}}{n^{1-a}} \\ &\text{assuming } \ell < \tfrac{c}{C}\frac{d^a k^{1-a}}{\log d}\end{aligned}$$

Where we absorbed numerical factors in c. This completes the proof of the first bound.

To prove the second bound, we consider for $m \geq 1$ the event $B_4 = \left\{\sum \boldsymbol{\sigma}_{j\in[d]}^{2a} \geq md^{2a}\right\}$.

Remember that $\Pr(\boldsymbol{\sigma}_j^{2a} \geq t) \leq \sqrt{e}t^{-1/2a}$. Thus, using Lemma 7.A.2 with $\alpha = 1/2a$, we have $\Pr[B_4] \leq K_{1/2a}m^{-1/2a}$, and conditioned on B_4^c, since $a \geq \frac{1}{2}$

$$\sum_{j\in[d]}\left\{\frac{C}{(n+1)^a} + C\log(nd)^2\boldsymbol{\sigma}_j^{2a}\right\} \leq Cm\log(nd)^2 d^{2a}.$$

Taking m sufficiently large, absorbing constant factors and applying the union bound over the events $\{B_i; i \in [5]\}$ suffices to complete the proof of the claim. □

We can now eventually conclude the analysis of the in-sample case by the proof of Lemma 7.4.10.

Proof of Lemma 7.4.10. We define conditionally on $\mathbf{X}$ and $\mathcal{M}(\mathbf{X})$ a random variable $i_{\mathbf{X},\mathcal{M}}$ such that, with $I \sim \mathcal{U}\{1,\dots,n\}$,

$$\mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2|_{\mathbf{X},\mathcal{M}}}[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}})] \geq \mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2,I|_{\mathbf{X},\mathcal{M}}}[\mathcal{T}(\mathrm{X}^{I})].$$

Note that, by the pigeonhole principle, there exists at least one index i that satisfies this condition.

Consider the event G from Lemma 7.4.14 and its numerical constants $(c_i)_{i\leq 4}$. By Lemma 7.4.13, $\mathbb{E}_{\boldsymbol{\sigma}^2,I|_{G,\mathbf{X},\mathcal{M}}}[\mathcal{T}(\mathrm{X}^I)] \geq c_1\frac{d^a k^{1-a}}{n^{1-a}}$. It follows that, conditionally on $\mathbf{X}, \mathcal{M}$ and $(\mathbf{X}, \mathcal{M}(\mathbf{X})) \in G$,

$$\Pr_{\boldsymbol{\sigma}^2,I|_{G,\mathbf{X},\mathcal{M}}}\left[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}}) \geq c_1\frac{d^a k^{1-a}}{n^{1-a}}\right] \geq \Pr_{\boldsymbol{\sigma}^2,I|_{G,\mathbf{X},\mathcal{M}}}\left[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}}) \geq \mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2|_{\mathbf{X},\mathcal{M}}}[\mathcal{T}(\mathrm{X}^I)]\right]$$

since $(\mathbf{X},\mathcal{M}(\mathbf{X})) \in G$

$$\geq \Pr_{\boldsymbol{\sigma}^2,I|_{G,\mathbf{X},\mathcal{M}}}\left[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}}) \geq \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2,I|_{G,\mathbf{X},\mathcal{M}}}[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}})]\right]$$

by the definition of $i_{\mathbf{X},\mathcal{M}}$

$$\geq \frac{1}{4}\frac{\mathbb{E}_{\boldsymbol{\sigma}^2,I|_{G,\mathbf{X},\mathcal{M}}}[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}})]^2}{\mathbb{E}_{\boldsymbol{\sigma}^2,I|_{G,\mathbf{X},\mathcal{M}}}[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}})^2]}$$

by Paley-Zygmund inequality

$$= \frac{1}{4}\frac{\mathbb{E}_{\boldsymbol{\sigma}^2|_{\mathbf{X},\mathcal{M}}}[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}})]^2}{\mathrm{Var}_{\boldsymbol{\sigma}^2|_{\mathbf{X},\mathcal{M}}}[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}})] + \mathbb{E}_{\boldsymbol{\sigma}^2|_{\mathbf{X},\mathcal{M}}}[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}})]^2}$$

$$\geq \frac{1}{8}\frac{\mathbb{E}_{\boldsymbol{\sigma}^2|_{\mathbf{X},\mathcal{M}}}[\mathcal{T}(\mathrm{X}^I)]^2 + \mathbb{E}_{\boldsymbol{\sigma}^2|_{\mathbf{X},\mathcal{M}}}[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}})]^2}{\mathrm{Var}_{\boldsymbol{\sigma}^2|_{\mathbf{X},\mathcal{M}}}[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}})] + \mathbb{E}_{\boldsymbol{\sigma}^2|_{\mathbf{X},\mathcal{M}}}[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}})]^2}.$$

by the definition of $i_{\mathbf{X},\mathcal{M}}$

The bounds on expectation and variance from Lemma 7.4.13, ensure that, for all $(\mathbf{X},\mathcal{M}(\mathbf{X})) \in G$,

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}^2|_{\mathbf{X},\mathcal{M},G}}[\mathcal{T}(\mathrm{X}^I)]^2 &\geq c_1^2\frac{d^{2a}k^{2-2a}}{n^{2-2a}}\frac{1}{c_2\log(nd)^2 d^{2a}}\mathrm{Var}_{\boldsymbol{\sigma}^2|_{\mathbf{X},\mathcal{M},G}}[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}})] \\ &= \Omega(\frac{k^{2-2a}}{n^{2-2a}\log(nd)^2})\mathrm{Var}_{\boldsymbol{\sigma}^2|_{\mathbf{X},\mathcal{M},G}}[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}})].\end{aligned}$$

Thus, for $k \geq \Omega(n\log(nd)^{1/(1-a)})$, we have $\Omega(\frac{k^{2-2a}}{n^{2-2a}\log(nd)^2}) \geq 1$, and thus it holds that

$$\Pr_{\boldsymbol{\sigma}^2,I|_{G,\mathbf{X},\mathcal{M}}}\left[\mathcal{T}(\mathrm{X}^{i_{\mathbf{X},\mathcal{M}}}) \geq c_1\frac{d^a k^{1-a}}{n^{1-a}}\right] \geq 1/8.$$

Let E be the event $E = G \cap \{I = i(X,S)\}$. Since I is uniform in $[n]$ and independent of $\mathbf{X}$ and $\mathcal{M}(\mathbf{X})$, Lemma 7.4.14 implies that $\Pr[E] = \Pr[I = i(X,S)]\Pr[G] = \Omega(1/n)$. Thus,

$$\Pr_{\boldsymbol{\sigma}^2,X,I,S}\left[\mathcal{T}(\mathrm{X}^I) \geq c_1\frac{d^a k^{1-a}}{n^{1-a}}\right] \geq \Pr_{\boldsymbol{\sigma}^2}\left[\mathcal{T}(\mathrm{X}^I) \gtrsim \frac{d^a k^{1-a}}{n^{1-a}} \mid E\right]\Pr[E] \geq \Omega\left(\frac{1}{n}\right),$$

Which completes the proof. □

Appendix

7.A Concentration Results

Lemma 7.A.1 (Random Matrix Concentration)**.** *Let $\mathbf{\Sigma}$ be uniformly distributed on the rank-k orthogonal projections in $\mathbb{R}^d$, and $\mathbf{X} \sim \mathcal{N}(0, \mathbf{\Sigma})$. There is a numerical constant $c > 0$, such that, if $n \leq ck$, we have conditionally on $\mathbf{\Sigma}$ that, with probability at least $1 - 2\exp(-ck)$ over the randomness of $\mathbf{X}$,*

$$\frac{k}{2n}\boldsymbol{P}_{\mathbf{X}} \preceq \hat{\mathbf{\Sigma}}_{\mathbf{X}} \preceq \frac{2k}{n}\boldsymbol{P}_{\mathbf{X}}.$$

Proof of Lemma 7.A.1. The singular value decomposition of $X = (X_1, \dots, X_n) \in \mathbb{R}^{d \times n}$ writes (considering $d > k > n$):

$$X = \sum_{i \in [n]} \boldsymbol{\sigma}_i(X) u_i(X) v_i(X)^T,$$

with $(u_i(X))_{i \in [n]} \subset \mathbb{R}^d$ and $(v_i(X))_{i \in [n]} \subset \mathbb{R}^n$ orthonormal families of vectors.

Considering the above notation, on the one hand, we have that

$$\hat{\mathbf{\Sigma}}_X = \frac{1}{n} X X^T = \sum_{i \in [n]} \frac{(\sigma_i(X))^2}{n} u_i(X) u_i(X)^T,$$

and on the other hand

$$P_X = \sum_{i \in [n]} u_i(X) u_i(X)^T.$$

Standard results on the eigenvalues of matrices with sub-gaussian entries (such as Vershynin (2018) Theorem 4.6.1), and the fact that $X^T|_{\mathbf{\Sigma}}$ is, up to a unitary transformation, a matrix with k i.i.d. isotropic Gaussian columns and $d - k$ columns of zeros, ensure that with probability at least $1 - \beta$ over the randomness of X,

$$\left\| \frac{1}{k} X^T X - I_n \right\|_2 \leq C \max\left(a, a^2\right) \quad \text{with} \quad a = \sqrt{\frac{n}{k}} + \sqrt{\frac{\log(2/\beta)}{k}},$$

with $C > 0$ a universal constant.

It follows that

$$\max(0, 1 - C\max(a, a^2)) P_X \preceq \frac{n}{k} \hat{\mathbf{\Sigma}}_X \preceq (1 + C\max(a, a^2)) P_X.$$

We then choose β in order to have, taking $C\max(a, a^2) = 1/2$. w.l.o.g, we can assume $C > 1$, which boils down to find β such that $\sqrt{n/k} + \sqrt{\frac{\log(2/\beta}{k}} = 1/2C$. Under $k/n > (4C)^2$, this exactly corresponds to

$$\beta = 2\exp\left(-k(1/2C - \sqrt{n/k})^2\right) \leq 2\exp(-k/(4C)^2).$$

Absorbing the constant factor in $c \leftarrow 1/(4C)^2$ yields the result. □

Lemma 7.A.2 (Sum of heavy tailed random variables)**.** *Let $\alpha \in (0,1), t_0 > 0$. Let Y be a nonnegative random variable such that $\Pr[Y \geq t] \leq Ct^{-\alpha}$ for all $t \geq t_0$. Let $Y_1, \ldots, Y_d$ be i.i.d. copies of Y. Then there exists $K = K(C, \alpha, t_0)$, such that for all $m \geq 1$,*

$$\Pr\left[\sum_{i=1}^{d} Y_i \geq md^{1/\alpha}\right] \leq Km^{-\alpha},$$

or equivalently, for any $\beta > 0$, $\sum_{i=1}^{d} Y_i \leq \left(\frac{Kd}{\beta}\right)^{1/\alpha}$ wp at least $1 - \beta$.
For $\alpha = 1$, it holds that

$$\Pr\left[\sum_{i=1}^{d} Y_i \geq md\right] \leq \frac{K_1 \log(md)}{m},$$

where $K_1 = (2C + t_0)$.

Proof. Let $A := \max\{C, t_0^\alpha\}$. Then for all $t > 0$, $\Pr[Y \geq t] \leq \min(1, Ct^{-\alpha})$. Let

$$S = \sum_{i=1}^{d} Y_i, \qquad M = \max_{1 \leq i \leq d} Y_i, \qquad x = md^{1/\alpha}.$$

By the union bound,

$$\Pr\left[M \geq \frac{x}{2}\right] \leq d\Pr\left[Y \geq \frac{x}{2}\right] \leq d\min\left(1, C\left(\frac{x}{2}\right)^{-\alpha}\right) = \min(d, C2^\alpha m^{-\alpha}) \leq 2^\alpha m^{-\alpha}.$$

Now let $T = \sum_{i=1}^{d} Y_i \mathbb{1}[Y_i \leq x/2]$. If $S \geq x$ and $M < x/2$, then $T = S \geq x$, so

$$\Pr[S \geq x] \leq \Pr\left(S \geq x,\ M < \frac{x}{2}\right) + \Pr(S \geq x,\ M \geq \frac{x}{2}) \leq \Pr(T \geq x) + \Pr(M \geq \frac{x}{2}).$$

Moreover, for $\alpha < 1$,

$$\begin{aligned}
\mathbb{E}[T] &= d\,\mathbb{E}\left[Y\mathbb{1}[Y \leq x/2]\right] \\
&\leq d\int_0^{x/2} \Pr[Y \geq t]\,dt \\
&\leq dt_0 + dC\int_{t_0}^{x/2} t^{-\alpha}\,dt \\
&= dt_0 + \frac{dC}{1-\alpha}\left(\left(\frac{x}{2}\right)^{1-\alpha} - t_0^{1-\alpha}\right),
\end{aligned}$$

thus $\mathbb{E}[T] \leq \frac{dC}{1-\alpha} 2^{\alpha-1} x^{1-\alpha}$ for $\tilde{C} = \max(C, \frac{t_0^\alpha}{1-\alpha})$. By Markov's inequality,

$$\Pr[T \geq x] \leq \frac{\mathbb{E}[T]}{x} \leq \frac{\tilde{C} 2^{\alpha-1}}{1-\alpha} d x^{-\alpha} \leq \frac{\tilde{C} 2^{\alpha-1}}{1-\alpha} m^{-\alpha}.$$

where the last inequality holds by the fact that $x = md^{1/\alpha}$, so $dx^{-\alpha} = m^{-\alpha}$. Combining the two:

$$\Pr[S \geq md^{1/\alpha}] \leq C 2^\alpha m^{-\alpha} + \frac{\tilde{C} 2^{\alpha-1}}{1-\alpha} m^{-\alpha} \leq K_{\alpha,t_0,C} m^{-\alpha}$$

where $K := \max\{C, t_0^\alpha\} \left(2^\alpha + \frac{2^{\alpha-1}}{1-\alpha}\right)$.

For $\alpha = 1$,

$$\mathbb{E}[T] \leq dt_0 + dC \int_{t_0}^{x/2} t^{-\alpha}\, dt = dt_0 + dC \log(\frac{x}{2t_0}).$$

Thus $\Pr[T \geq x] \leq \frac{t_0}{m} + \frac{C \log(md/2t_0)}{m}$ And

$$\Pr[S \geq md^{1/\alpha}] \leq \frac{2C}{m} + \frac{t_0}{m} + \frac{C \log(md/2t_0)}{m} \leq K_1 \frac{\log(md)}{m}$$

where $K_1 = (2C + t_0)$. □

Lemma 7.A.3 (Top-k samples from $\mathsf{inv}\chi^2(1,1)$)**.** *Let $\boldsymbol{\sigma} \sim \mathsf{inv}\chi^2(1,1)^{\otimes d}$, then for all $1 \leq k \leq d/2$,*

$$\Pr\left[\boldsymbol{\sigma}_{(k)} \geq \frac{d^2}{4k^2}\right] \geq 1 - \frac{2}{k}.$$

Proof. For all $i \in [d]$, since $\boldsymbol{\sigma}_i \sim \mathsf{inv}\chi^2(1,1)$, we have for $t \geq 1$ that $\Pr(\boldsymbol{\sigma}_i \geq t) = C(t) t^{-1/2}$, with $C(t) \in [1, \sqrt{e}]$.

Consider the events $H_i := \left\{\boldsymbol{\sigma}_i \geq d^2/(2k)^2\right\}$ and let $S = \sum_{i\in[d]} \mathbb{1}_{H_i}$. It holds that $\Pr(H_i) = C(2k/d)\frac{2k}{d}$, and thus $\mathbb{E}[S] = C(2k/d)2k$. Since S is the sum of independent Bernoulli random variables, Chebyshev's inequality yields

$$\begin{aligned}
\Pr\left(S \leq \frac{1}{2}\mathbb{E}[S]\right) &= \Pr\left(S - \mathbb{E}[S] \geq \frac{1}{2}\mathbb{E}[S]\right) \\
&\leq \Pr\left((S - \mathbb{E}[S])^2 \geq \frac{1}{2}\mathbb{E}[S]^2\right) \\
&\leq 2\frac{1 - \Pr[H_i]}{d \Pr[H_i]} \\
&= 2\frac{1 - C(2k/d)k/d}{C(2k/d)k}.
\end{aligned}$$

Using $C(2k/d) \geq 1$, it follows that $\boldsymbol{\sigma}^2_{(k)} \geq \frac{d^2}{4k^2}$ with probability at least $1 - \frac{2}{k} + \frac{2}{d}$.

□

Lemma 7.A.4 (Conditional Gaussian Expectation)**.** *Let $\boldsymbol{\mu} \sim \mathcal{N}(0, \rho I_d)$ and $\mathbf{X} = (\mathbf{X}^1, \ldots, \mathbf{X}^n) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ is as positive symmetric matrix. Then it*

holds that

$$\boldsymbol{\mu}|_{\mathbf{X}} \sim \mathcal{N}\left(\left(\mathbb{I}_d + \frac{1}{n\rho}\boldsymbol{\Sigma}\right)^{-1}\bar{\mathbf{x}}, \left(\frac{1}{\rho}\mathbb{I}_d + n\boldsymbol{\Sigma}^{-1}\right)^{-1}\right), \quad \text{where } \bar{\mathbf{x}} = \frac{1}{n}\sum_{i\in[n]} \mathrm{X}^i.$$

Proof. Since $(\mu, \mathrm{X}^1, \dots, \mathrm{X}^n)$ is a gaussian vector, so is $\boldsymbol{\mu}|_{\mathbf{X}}$ by Gaussian conditioning. By Bayes rule, we have that the density of $\boldsymbol{\mu}|_{\mathbf{X}=\boldsymbol{X}}$ is proportional to

$$p_{\boldsymbol{\mu}|_{\mathbf{X}=\boldsymbol{X}}}(\boldsymbol{u}) \propto \exp\left(-\frac{1}{2}\boldsymbol{u}^T(\rho^{-1}\mathbb{I})\boldsymbol{u}\right)\prod_{i\in[d]}\exp\left(-\frac{1}{2}(X^1-\boldsymbol{u})^T\boldsymbol{\Sigma}^{-1}(X^1-\boldsymbol{u})\right),$$

Dropping terms independent of $\boldsymbol{u}$, the density is thus

$$\begin{aligned} p_{\boldsymbol{\mu}|_{\mathbf{X}=\boldsymbol{X}}}(\boldsymbol{u}) &\propto \exp\left(-\frac{1}{2}\boldsymbol{u}^T\left(\frac{1}{\rho}\mathbb{I} + n\boldsymbol{\Sigma}^{-1}\right)\boldsymbol{u} + \boldsymbol{u}^T n\boldsymbol{\Sigma}^{-1}\bar{\mathbf{x}}\right)p_{\boldsymbol{\mu}|_{\mathbf{X}=\boldsymbol{X}}}(\boldsymbol{u}) \\ &\propto \exp\left(-\frac{1}{2}\left\|\boldsymbol{u} - \left(\frac{1}{\rho}\mathbb{I} + n\boldsymbol{\Sigma}^{-1}\right)n\boldsymbol{\Sigma}^{-1}\bar{\mathbf{x}}\right\|^2_{\frac{1}{\rho}\mathbb{I}+n\boldsymbol{\Sigma}^{-1}}\right). \end{aligned}$$

Up to a multiplicative term independent of $\boldsymbol{u}$, which concludes the proof. □

Proposition 7.A.5 (Pinsker's inequality)**.** *For all probability distributions P and Q we have $\mathrm{d_{TV}}(P,Q) \le \sqrt{\frac{1}{2}\mathrm{d_{KL}}(P,Q)}$.*

Proposition 7.A.6 (KL divergence between Gaussians)**.** *Fix $\sigma^2, \mu_1, \mu_2 \in \mathbb{R}$. Then $\mathrm{d_{KL}}\big(\mathcal{N}(\mu_1,\sigma^2),\mathcal{N}(\mu_2,\sigma^2)\big) = \frac{\log e}{2}\big(\frac{\mu_1-\mu_2}{\sigma}\big)^2$. Moreover, for all $n \in \mathbb{N}$ we have $\mathrm{d_{KL}}\big(\mathcal{N}(\mu_1,\sigma^2)^n,\mathcal{N}(\mu_2,\sigma^2)^n\big) \le \frac{n\log e}{2}\big(\frac{\mu_1-\mu_2}{\sigma}\big)^2$.*

Proposition 7.A.7 (Concentration of the $\chi^2(k)$ random variable (Laurent and Massart, 2000))**.** *Let $\mathrm{x} \sim \chi^2(k)$, then for any $t \ge 0$*

$$\Pr(\mathrm{x} - k \ge 2\sqrt{kt} + 2t) \le \exp(-t), \quad \Pr(k - \mathrm{x} \le 2\sqrt{kt}) \le \exp(-t).$$

Proposition 7.A.8. *Let $Y \sim \chi^2(k)$ be a chi-squared random variable with k degrees of freedom. Then for all $m > -k/2$*

$$\mathbb{E}\left[Y^m\right] = \frac{2^m\Gamma\left(\frac{k}{2}+m\right)}{\Gamma\left(\frac{k}{2}\right)}.$$

Proposition 7.A.9 (Gautschi's inequality)**.** *For all $x, s \in (0,1)$ we have $\Gamma(x+1) = x\Gamma(x)$ and $x^{1-s} \le \frac{\Gamma(x+1)}{\Gamma(x+s)} \le (x+1)^{1-s}$.*

Part V

Conclusion and Perspectives

Summary of the thesis

In this thesis, we study several aspects of robustness in statistical learning and optimization.

In Chapter 2, we introduce supervised learning, first-order convex optimization, distributed learning, Byzantine robustness, and differential privacy. This sets the stage for our main contributions by illustrating some fundamental concepts and situating them within the literature.

In Chapter 4 we propose a generic approach for designing gossip communication algorithms robust to Byzantine failures, which makes non-trivial use of robust averaging rules. The resulting communication algorithm converges linearly to a consensus with a controlled error. Its guarantees match those of the non-robust gossip in the absence of Byzantine failures, and degrades when the number of Byzantine failures increases, up to a breakdown point. This breakdown point relates the smallest nonzero eigenvalue of the Laplacian matrix of the honest subgraph to the number of Byzantine clients located in the neighborhood of each honest node. By providing examples on which communication cannot be robust, we show that the breakdown point of the robust gossip method with a clipping aggregation rule is optimal, up to a multiplicative factor of 2. By combining it with existing analysis, we propose a provably robust Decentralized-SGD algorithm. To test the robustness of decentralized optimization algorithms, we propose a new Byzantine attack that relies on spectral clustering of the communication graph. This allows us to compare the empirical performance of our method with that of previous ones on averaging and learning tasks, both against our attack and standard ones. These experiments validate the superior robustness of our method.

In Chapter 5 we investigate the feasibility of the dual approach for designing robust decentralized optimization algorithm in presence of Byzantine fault. Formulating the problem in a constrained optimization approach, we re-interpret existing robust communication approaches as a control on the dual variable. To go beyond this insight, we propose a clipping-based robust communication algorithm with a global clipping threshold and identify under which condition it provably decreases the mean-squared-error. Then, we discuss the key challenges for dual-approach-based algorithms and proofs in the Byzantine context.

In Chapter 6 we formalize centralized learning with Byzantine fault as a an optimization task with inexact gradient oracles. This allows both to recover tight convergence guarantees for existing algorithm as D-GD, and pave the way for a more systematic design of efficient robust algorithms. We demonstrate this point by proposing two fast algorithms. The first one follows the "optimization under similarity" perspective. It uses a cheap access to a proxy of the loss function to drastically reduce the number of communications round, in particular when the hessian of the proxy is close to the one of the global loss. The second one is an extragradient method, which enjoys an accelerated rate towards the optimal error under both a relative and an absolute error term in the gradient. This result outperforms previous works on acceleration with such inexact oracles in both the asymptotic error, and in the condition of the relative error. Eventually, this extra-gradient method can also be applied in the similarity setting, although under a more restrictive heterogeneity assumption.

In Chapter 7 we investigate the feasibility of private mean estimator whose sampling

complexity for a given error does not depend on the ambient dimension, but rather on quantities such as the trace of the (square root of) the covariance matrix of the distribution. We show that, when the covariance matrix is unknown, there is an unavoidable accuracy penalty compared to optimal estimators that know the covariance matrix, and that this penalty grows as a power of the dimension. This conclusion is supported by two analyses: in the first, we consider covariance matrices that are orthogonal projection, and show that the naive private mean estimation algorithm is optimal when the number of samples is smaller than the rank of the covariance matrix. In the second example we consider diagonal covariance matrices whose effective dimension is logarithmic with respect to the dimension. We show that covariance-agnostic estimators always suffer from a penalty that grows as a power of the ambient dimension, as long as the number of samples is less than a fractional power of the ambient dimension. This work provides a negative answer to the question of the feasibility of dimension-independent private mean estimator in the unknown covariance setting.

Open directions

Following these works, several exciting directions appear.

However, considering the rise of LLMs-based automatic search for open directions and subsequent production of "scientific" papers, those open directions shall remain private. If you are interested in a direct follow-up from one of my papers in the next few years, do not hesitate to reach me out to see if I'm planning to work on this, this could lead to a fruitfull collaboration!

Bibliography

J. M. Abowd. The US Census Bureau adopts differential privacy. In *ACM International Conference on Knowledge Discovery & Data Mining*, KDD '18, 2867–2867, 2018. (p. 161.)

J. M. Abowd, R. Ashmead, R. Cumings-Menon, S. Garfinkel, M. Heineck, C. Heiss, R. Johns, D. Kifer, P. Leclerc, A. Machanavajjhala, B. Moran, W. Sexton, M. Spence, and P. Zhuravlev. The 2020 Census Disclosure Avoidance System TopDown Algorithm. *Harvard Data Science Review*, Special Issue 2, June 2022. (p. 161.)

J. M. Abowd, T. Adams, R. Ashmead, D. Darais, S. Dey, S. L. Garfinkel, N. Goldschlag, D. Kifer, P. Leclerc, E. Lew, et al. A simulated reconstruction and reidentification attack on the 2010 us census: Full technical report. Technical report, National Bureau of Economic Research, 2023. (p. 33.)

I. Aden-Ali, H. Ashtiani, and G. Kamath. On the sample complexity of privately learning unbounded high-dimensional Gaussians. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, ALT '21, March 2021. (p. 37.)

A. Agarwal, M. J. Wainwright, P. Bartlett, and P. Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. *Advances in Neural Information Processing Systems*, 22, 2009. (p. 17.)

A. Ajalloeian and S. U. Stich. On the convergence of sgd with biased gradients. *arXiv preprint arXiv:2008.00051*, 2020. (pp. 30, 121, 122, and 155.)

D. Alistarh, Z. Allen-Zhu, and J. Li. Byzantine stochastic gradient descent. *Advances in neural information processing systems*, 31, 2018. (pp. 49 and 91.)

Z. Allen-Zhu, F. Ebrahimianghazani, J. Li, and D. Alistarh. Byzantine-resilient non-convex stochastic gradient descent. In *International Conference on Learning Representations*, 2021. (p. 128.)

Y. Allouah, S. Farhadkhani, R. Guerraoui, N. Gupta, R. Pinot, and J. Stephan. Fixing by mixing: A recipe for optimal byzantine ml under heterogeneity. In *International Conference on Artificial Intelligence and Statistics*, 1232–1300. PMLR, 2023a. (pp. 49, 54, 91, 117, 118, 120, 121, 128, 131, 132, and 135.)

Y. Allouah, R. Guerraoui, N. Gupta, R. Pinot, and J. Stephan. On the privacy-robustness-utility trilemma in distributed learning. In *International Conference on Machine Learning*, 569–626. PMLR, 2023b. (p. 137.)

Y. Allouah, R. Guerraoui, N. Gupta, R. Pinot, and G. Rizk. Robust distributed learning: tight error bounds and breakdown point under data heterogeneity. *Advances in Neural Information Processing Systems*, 36, 2024. (pp. 28, 49, and 120.)

Y. Allouah, R. Guerraoui, N. Gupta, A. Jellouli, G. Rizk, and J. Stephan. Adaptive gradient clipping for robust federated learning. In *International Conference on Learning Representations*, 2025, 84251–84295, 2025a. (pp. 49 and 58.)

Y. Allouah, R. Guerraoui, and J. Stephan. Towards trustworthy federated learning with untrusted participants. In *International Conference on Machine Learning*, 1184–1227. PMLR, 2025b. (pp. 32, 122, 128, 137, 138, and 153.)

F. Bach. *Learning theory from first principles*. MIT press, 2024. (pp. 9 and 11.)

F. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate o (1/n). *Advances in neural information processing systems*, 26, 2013. (p. 17.)

G. Baruch, M. Baruch, and Y. Goldberg. A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems*, 32, 2019. (pp. 49, 62, 67, 100, 103, 129, and 153.)

P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017. (pp. 27, 29, 49, 51, 57, 91, 93, 117, 120, 128, and 153.)

T. Boudou, B. Le Bars, N. Gupta, and A. Bellet. Generalization under byzantine & poisoning attacks: Tight stability bounds in robust distributed learning. 2025. (p. 31.)

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004. (pp. 11 and 25.)

S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE transactions on information theory*, 52(6):2508–2530, 2006. (pp. 21, 23, 51, 91, 92, and 93.)

G. Brown, M. Bun, V. Feldman, A. Smith, and K. Talwar. When is memorization of irrelevant training data necessary for high-accuracy learning? *arXiv*, 2012.06421, 2021a. (pp. 33 and 161.)

G. Brown, M. Gaboardi, A. Smith, J. Ullman, and L. Zakynthinou. Covariance-aware private mean estimation without private covariance estimation. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21, 7950–7964, 2021b. (p. 37.)

S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and trends in Machine Learning*, 8(3-4):231–357, 2015. (p. 11.)

M. Bun and T. Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, TCC '16, 635–658, Beijing, China, 2016. https://arxiv.org/abs/1605.02065. (p. 36.)

M. Bun, J. Ullman, and S. Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *ACM Symposium on the Theory of Computing*, STOC '14, 1–10, 2014. (pp. 33 and 161.)

M. Bun, G. Kamath, T. Steinke, and Z. S. Wu. Private hypothesis selection. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, 156–167, 2019. (p. 37.)

T. T. Cai, Y. Wang, and L. Zhang. Score attack: A lower bound technique for optimal differentially private learning. *arXiv preprint arXiv:2303.07152*, 2023. (p. 162.)

N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In {*USENIX*} *Security Symposium*, ({USENIX} '19), 267–284, 2019. https://arxiv.org/abs/1802.08232. (p. 161.)

N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, 2633–2650, 2021. (pp. 33 and 161.)

A. Cauchy. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25:536–538, 1847. (p. 12.)

Y. Chen, L. Su, and J. 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, 2017. (pp. 91 and 117.)

E. Commission. On artificial intelligence—a european approach to excellence and trust. *White Paper*, 2020. (p. 5.)

E. Cyffers. *Differential Privacy for Decentralized Learning*. PhD thesis, Université de Lille, 2024. (p. 32.)

S. Cyrus, B. Hu, B. Van Scoy, and L. Lessard. A robust accelerated optimization algorithm for strongly convex functions. In *2018 Annual American Control Conference (ACC)*, 1376–1381. IEEE, 2018. (p. 121.)

Y. Dagan, M. Jordan, X. Yang, L. Zakynthinou, and N. Zhivotovskiy. Dimension-free private mean estimation for anisotropic distributions. In *Advances in Neural Information Processing Systems*, NeurIPS '24, 2024. https://arxiv.org/abs/2411.00775. (pp. 37, 42, 161, 165, 166, 173, 174, and 175.)

A. d'Aspremont. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008. (pp. 122 and 151.)

E. De Klerk, F. Glineur, and A. B. Taylor. Worst-case convergence analysis of inexact gradient and newton methods through semidefinite programming performance estimation. *SIAM Journal on Optimization*, 30(3):2053–2082, 2020. (p. 121.)

A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014. (p. 17.)

D. Desfontaines. Ted is writing things (personal blog). https://desfontain.es/blog/posts.html, 06 2026. (p. 32.)

O. Devolder, F. Glineur, Y. Nesterov, et al. First-order methods with inexact oracle: the strongly convex case. *CORE discussion papers*, 2013016:47, 2013. (pp. iv, 116, 122, 123, 150, and 151.)

O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1):37–75, 2014. (pp. 122, 123, 127, 150, 151, and 152.)

I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart. Being robust (in high dimensions) can be practical. In *International Conference on Machine Learning*, 999–1008. PMLR, 2017. (p. 138.)

I. Diakonikolas, G. Kamath, D. Kane, J. Li, A. Moitra, and A. Stewart. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019. (pp. 31, 117, and 122.)

A. Dieuleveut, A. Durmus, and F. Bach. Bridging the gap between constant step size stochastic gradient descent and markov chains. 2020. (p. 17.)

I. Dinur and K. Nissim. Revealing information while preserving privacy. In *Proceedings of the 22nd ACM Symposium on Principles of Database Systems*, PODS '03, 202–210. ACM, 2003. (p. 161.)

D. Dolev, N. A. Lynch, S. S. Pinter, E. W. Stark, and W. E. Weihl. Reaching approximate agreement in the presence of faults. *Journal of the ACM (JACM)*, 33(3):499–516, 1986. (pp. 26 and 56.)

J. Dong, A. Roth, and W. J. Su. Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*, 2019. https://arxiv.org/abs/1905.02383. (p. 36.)

J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57 (3):592–606, 2011. (p. 91.)

C. Dwork and J. Lei. Differential privacy and robust statistics. In *Proceedings of the 41st ACM Symposium on Theory of Computing*, STOC '09, 371–380. ACM, 2009. (p. 165.)

C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014. (p. 32.)

C. Dwork and G. N. Rothblum. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*, 2016. https://arxiv.org/abs/1603.01887. (p. 36.)

C. Dwork and S. Yekhanin. New efficient attacks on statistical disclosure control mechanisms. In *Annual International Cryptology Conference*, 469–480. Springer, 2008. (p. 161.)

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Conference on Theory of Cryptography*, TCC '06, 265–284, 2006. (pp. 35, 36, 161, and 164.)

C. Dwork, F. McSherry, and K. Talwar. The price of privacy and the limits of LP decoding. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, 85–94. ACM, 2007. (p. 161.)

C. Dwork, G. N. Rothblum, and S. P. Vadhan. Boosting and differential privacy. In *Proceedings of the 51st IEEE Annual Symposium on Foundations of Computer Science*, FOCS '10, 51–60. IEEE, 2010. (p. 35.)

C. Dwork, A. Smith, T. Steinke, J. Ullman, and S. Vadhan. Robust traceability from trace amounts. In *Proceedings of the 56th IEEE Annual Symposium on Foundations of Computer Science*, FOCS '15, 2015. (pp. 33 and 161.)

A. d'Aspremont, D. Scieur, and A. Taylor. Acceleration methods. *Foundations and Trends in Optimization*, 5(1-2):1–245, 2021. (p. 16.)

J. Egan, C. Robertson, A. Ross Arguedas, N. Newman, R. Nielsen, M. Mukherjee, and R. Fletcher. Reuters institute digital news report 2026. 2026. (p. 5.)

E.-M. El-Mhamdi, R. Guerraoui, A. Guirguis, L. N. Hoang, and S. Rouault. Genuinely distributed byzantine machine learning. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, 355–364, 2020. (pp. 49 and 91.)

E. M. El-Mhamdi, S. Farhadkhani, R. Guerraoui, A. Guirguis, L.-N. Hoang, and S. Rouault. Collaborative learning in the jungle (decentralized, byzantine, heterogeneous, asynchronous and nonconvex learning). *Advances in Neural Information Processing Systems*, 34:25044–25057, 2021. (pp. 49, 55, and 91.)

Ú. Erlingsson, V. Pihur, and A. Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *ACM Conference on Computer and Communications Security*, CCS '14, 2014. (p. 161.)

M. Even, R. Berthier, F. Bach, N. Flammarion, H. Hendrikx, P. Gaillard, L. Massoulié, and A. Taylor. Continuized accelerations of deterministic and stochastic gradient descents, and of gossip algorithms. *Advances in Neural Information Processing Systems*, 34:28054–28066, 2021. (p. 23.)

M. Even, H. Hendrikx, and L. Massoulié. Asynchronous speedup in decentralized optimization. *IEEE Transactions on Automatic Control*, 70(3):1467–1482, 2024. (p. 23.)

C. Fang, Z. Yang, and W. U. Bajwa. Bridge: Byzantine-resilient decentralized gradient descent. *IEEE Transactions on Signal and Information Processing over Networks*, 8: 610–626, 2022. (pp. 49 and 53.)

S. Farhadkhani, R. Guerraoui, N. Gupta, R. Pinot, and J. Stephan. Byzantine machine learning made easy by resilient averaging of momentums. In *International Conference on Machine Learning*, 6246–6283. PMLR, 2022. (pp. 30, 49, 50, 58, 67, 91, 100, 117, and 155.)

S. Farhadkhani, R. Guerraoui, N. Gupta, L.-N. Hoang, R. Pinot, and J. Stephan. Robust collaborative learning with linear gradient overhead. In *International Conference on Machine Learning*, 9761–9813. PMLR, 2023. (pp. 49, 59, 60, 62, 67, 68, 86, 87, 91, 98, and 135.)

V. Feldman. Does learning require memorization? A short tale about a long tail. In *ACM Symposium on Theory of Computing*, STOC '20, 954–959, 2020. https://arxiv.org/abs/1906.05271. (p. 161.)

M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985. (p. 26.)

R. Gaucher, A. Dieuleveut, and H. Hendrikx. Unified breakdown analysis for byzantine robust gossip. In *Forty-second International Conference on Machine Learning*. PMLR, 2025. (pp. 91, 98, 99, 100, 101, 103, 135, and 136.)

A. Ghosh, R. K. Maity, and A. Mazumdar. Distributed newton can communicate less and resist byzantine workers. *Advances in Neural Information Processing Systems*, 33: 18028–18038, 2020. (p. 117.)

M. González, R. Guerraoui, R. Pinot, G. Rizk, J. Stephan, and F. Taïani. Byzfl: Research framework for robust federated learning, 2025. (p. 129.)

S. Haney, A. Machanavajjhala, J. M. Abowd, M. Graham, M. Kutzbach, and L. Vilhuber. Utility cost of formal privacy for releasing national employer-employee statistics. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, 1339–1354. ACM, 2017. (p. 161.)

M. Hardt and K. Talwar. On the geometry of differential privacy. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, STOC '10, 2010. (p. 37.)

F. Hartmann and P. Kairouz. Distributed differential privacy for federated learning, 2023. https://research.google/blog/distributed-differential-privacy-for-federated-learning/. (p. 161.)

W. HASTINGS. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97, 1970. (p. 66.)

L. He, S. P. Karimireddy, and M. Jaggi. Byzantine-robust decentralized learning via clippedgossip, 2023. (pp. 49, 50, 51, 56, 58, 61, 62, 66, 68, 84, 85, 91, 98, 99, 100, and 103.)

H. Hendrikx, F. Bach, and L. Massoulié. Accelerated decentralized optimization with local updates for smooth and strongly convex objectives. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 897–906. PMLR, 2019. (pp. 24, 91, and 93.)

H. Hendrikx, F. Bach, and L. Massoulié. Dual-free stochastic decentralized optimization with variance reduction. *Advances in neural information processing systems*, 33:19455–19466, 2020a. (pp. 25 and 94.)

H. Hendrikx, L. Xiao, S. Bubeck, F. Bach, and L. Massoulie. Statistically preconditioned accelerated gradient method for distributed optimization. In *International conference on machine learning*, 4203–4227. PMLR, 2020b. (pp. 123 and 126.)

H. Hendrikx, F. Bach, and L. Massoulie. An optimal algorithm for decentralized finite-sum optimization. *SIAM Journal on Optimization*, 31(4):2753–2783, 2021. (pp. 24, 25, and 93.)

N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genetics*, 4(8):e1000167, 2008. (p. 161.)

T.-M. H. Hsu, H. Qi, and M. Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019. (p. 128.)

P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. (pp. 31 and 132.)

D. Jakovetić. A unification and generalization of exact distributed first-order methods. *IEEE Transactions on Signal and Information Processing over Networks*, 5(1):31–46, 2018. (pp. 24, 25, 91, and 93.)

D. Jakovetić, D. Bajović, J. Xavier, and J. M. Moura. Primal–dual methods for large-scale and distributed convex optimization and data analytics. *Proceedings of the IEEE*, 108 (11):1923–1938, 2020. (p. 93.)

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013. (p. 17.)

P. Kairouz and H. B. McMahan. Advances and open problems in federated learning. *Foundations and trends in machine learning*, 14(1-2):1–210, 2021. (pp. 6 and 18.)

P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. In *Proceedings of the 32nd International Conference on Machine Learning*, ICML '15, 1376–1385, 2015. (p. 35.)

G. Kamath, J. Li, V. Singhal, and J. Ullman. Privately learning high dimensional distributions. In *Proceedings of the 32nd Annual Conference on Learning Theory*, COLT '19. JMLR, 2019. (pp. 37 and 167.)

S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, 5132–5143. PMLR, 2020. (pp. 20 and 118.)

S. P. Karimireddy, L. He, and M. Jaggi. Learning from history for byzantine robust optimization. In *International Conference on Machine Learning*, 5311–5319. PMLR, 2021. (pp. 30, 49, 59, 91, 117, 120, 122, 133, and 155.)

S. P. Karimireddy, L. He, and M. Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing, 2023. (pp. 49, 52, 91, 117, 131, 133, and 152.)

V. Karwa and S. Vadhan. Finite sample differentially private confidence intervals. In *Proceedings of the 9th Conference on Innovations in Theoretical Computer Science*, ITCS '18, 44:1–44:9, 2018. (pp. 37 and 166.)

S. P. Kasiviswanathan, M. Rudelson, A. Smith, and J. Ullman. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, STOC '10, 775–784. ACM, 2010. (p. 161.)

A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, 5381–5393. PMLR, 2020. (p. 24.)

A. Korinek and J. E. Stiglitz. Artificial intelligence and its implications for income distribution and unemployment. In *The economics of artificial intelligence: An agenda*, 349–390. University of Chicago Press, 2018. (p. 5.)

N. Kornilov, M. Alkousa, E. Gorbunov, F. Stonyakin, P. Dvurechensky, and A. Gasnikov. Intermediate gradient methods with relative inexactness. *Journal of Optimization Theory and Applications*, 207(3):62, 2025. (p. 127.)

D. Kovalev, A. Salim, and P. Richtárik. Optimal and practical algorithms for smooth and strongly convex decentralized optimization. *Advances in Neural Information Processing Systems*, 33:18342–18352, 2020. (pp. 21, 24, 25, 51, 56, 91, 93, 94, and 123.)

D. Kovalev, A. Beznosikov, E. Borodich, A. Gasnikov, and G. Scutari. Optimal gradient sliding and its application to optimal distributed optimization under similarity. *Advances in Neural Information Processing Systems*, 35:33494–33507, 2022. (pp. 41, 118, 123, and 126.)

L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982. (pp. 6, 26, 49, 55, 91, and 117.)

B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, 1302–1338, 2000. (pp. 171 and 190.)

H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram. Resilient asymptotic consensus in robust networks. *IEEE Journal on Selected Areas in Communications*, 31(4):766–781, 2013. (pp. 26, 51, and 56.)

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. (p. 128.)

H. Li and Z. Lin. Revisiting extra for smooth distributed optimization. *SIAM Journal on Optimization*, 30(3):1795–1821, 2020. (p. 24.)

L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling. Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 1544–1551, 2019. (p. 91.)

X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in neural information processing systems*, 30, 2017. (p. 20.)

C. Liu, Y. Li, Y. Yi, and K. H. Johansson. Byzantine-robust and communication-efficient distributed learning via compressed momentum filtering. *arXiv preprint arXiv:2409.08640*, 2024. (p. 117.)

X. Liu, W. Kong, S. Kakade, and S. Oh. Robust and differentially private mean estimation. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21, 3887–3901, 2021. (p. 37.)

X. Liu, W. Kong, and S. Oh. Differential privacy and robust statistics in high dimensions. In *Proceedings of the 35th Annual Conference on Learning Theory*, COLT '22, 1167–1246. PMLR, Jul 2022. (p. 37.)

G. Lugosi and S. Mendelson. Robust multivariate mean estimation: the optimality of trimmed mean. *The Annals of Statistics*, 49(1):393–410, 2021. (p. 32.)

X. Lyu and K. Talwar. Fingerprinting codes meet geometry: Improved lower bounds for private query release and adaptive data analysis. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, 2374–2385, 2025. (p. 162.)

G. Malinovsky, P. Richtárik, S. Horváth, and E. Gorbunov. Byzantine robustness and partial participation can be achieved at once: Just clip gradient differences. *Advances in Neural Information Processing Systems*, 37:34900–34979, 2024. (p. 117.)

P. Mangold, A. Durmus, A. Dieuleveut, and E. Moulines. Scaffold with stochastic gradients: New analysis with linear speed-up. *arXiv preprint arXiv:2503.07594*, 2025. (p. 20.)

B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. Pmlr, 2017. (p. 20.)

I. Mironov. Rényi differential privacy. In *IEEE Computer Security Foundations Symposium*, CSF '17, 263–275, Santa Barbara, CA, USA, 2017. https://arxiv.org/abs/1702.07476. (p. 35.)

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018. (p. 9.)

A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*, 111–125, 2008. (p. 33.)

M. Nasr, N. Carlini, J. Hayase, M. Jagielski, A. F. Cooper, D. Ippolito, C. A. Choquette-Choo, E. Wallace, F. Tramèr, and K. Lee. Scalable extraction of training data from (production) language models. arxiv. *Preprint posted online on Nov*, 28, 2023. (p. 33.)

A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on automatic control*, 54(1):48–61, 2009. (pp. 21, 24, 51, and 91.)

A. Nedic, A. Olshevsky, and W. Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017. (p. 24.)

A. S. Nemirovskij and D. B. Yudin. Problem complexity and method efficiency in optimization. 1983. (p. 16.)

Y. Nesterov et al. *Lectures on convex optimization*, 137. Springer, 2018. (pp. 11, 12, 15, and 16.)

K. Pillutla, S. M. Kakade, and Z. Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022. (pp. 57 and 120.)

B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992. (p. 16.)

A. Rammal, K. Gruntkowska, N. Fedin, E. Gorbunov, and P. Richtárik. Communication compression for byzantine robust learning: New efficient algorithms and improved rates. In *International Conference on Artificial Intelligence and Statistics*, 1207–1215. PMLR, 2024. (p. 117.)

H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951. (p. 16.)

B. Roessler and J. DeCew. Privacy. In E. N. Zalta and U. Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2023 edition, 2023. (p. 32.)

S. Sankararaman, G. Obozinski, M. I. Jordan, and E. Halperin. Genomic privacy and limits of individual detection in a pool. *Nature Genetics*, 41(9):965–967, 2009. (p. 161.)

K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *international conference on machine learning*, 3027–3036. PMLR, 2017. (pp. 21, 23, 24, 25, 51, 56, 91, and 93.)

M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1):83–112, 2017. (p. 17.)

O. Shamir, N. Srebro, and T. Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *International conference on machine learning*, 1000–1008. PMLR, 2014. (p. 123.)

Q. Shi, J. Peng, K. Yuan, X. Wang, and Q. Ling. Optimal complexity in byzantine-robust distributed stochastic optimization with data heterogeneity. *Journal of Machine Learning Research*, 26(268):1–58, 2025. (pp. iv, 116, 118, and 131.)

W. Shi, Q. Ling, G. Wu, and W. Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015. (pp. 21 and 24.)

R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (S&P), Oakland*, 2017. (pp. 33 and 161.)

C. G. Small. A survey of multidimensional medians. *International Statistical Review/Revue Internationale de Statistique*, 263–277, 1990. (p. 120.)

S. U. Stich. Local sgd converges fast and communicates little. In *International Conference on Learning Representations*, 2019. (p. 20.)

M. Sundaram Muthu Selva Annamalai, E. De Cristofaro, and P. Kairouz. Cliopatra: Extracting private information from llm insights. *arXiv e-prints*, arXiv–2603, 2026. (p. 33.)

D. Tastuggine and I. Mironov. Introducing Opacus: A high-speed library for training PyTorch models with differential privacy. Facebook AI Blog, 2020. https://ai.facebook.com/blog/introducing-opacus-a-high-speed-library-for-training-pytorch-models-with-differential-privacy/. (p. 161.)

J. A. Tropp. An introduction to matrix concentration inequalities. *Foundations and trends® in machine learning*, 8(1-2):1–230, 2015. (p. 125.)

C. A. Uribe, S. Lee, A. Gasnikov, and A. Nedić. A dual approach for optimal algorithms in distributed optimization over networks. In *2020 Information Theory and Applications Workshop (ITA)*, 1–37. IEEE, 2020. (p. 93.)

N. H. Vaidya. Iterative byzantine vector consensus in incomplete graphs. In *Distributed Computing and Networking: 15th International Conference, ICDCN 2014, Coimbatore, India, January 4-7, 2014. Proceedings 15*, 14–28. Springer, 2014. (pp. 26 and 56.)

N. H. Vaidya, L. Tseng, and G. Liang. Iterative approximate byzantine consensus in arbitrary directed graphs. In *Proceedings of the 2012 ACM symposium on Principles of distributed computing*, 365–374, 2012. (pp. 26 and 55.)

A. Vasin, A. Gasnikov, P. Dvurechensky, and V. Spokoiny. Accelerated gradient methods with absolute and relative noise in the gradient. *Optimization Methods and Software*, 38 (6):1180–1229, 2023. (pp. 121 and 127.)

R. Vershynin. *High-dimensional probability: An introduction with applications in data science*, 47. Cambridge university press, 2018. (p. 187.)

T. Vogels, H. Hendrikx, and M. Jaggi. Beyond spectral gap: The role of the topology in decentralized learning. *Journal of Machine Learning Research*, 24(355):1–31, 2023. (p. 24.)

B. Woodworth, K. Mishchenko, and F. Bach. Two losses are better than one: Faster optimization using a cheaper proxy. In *International Conference on Machine Learning*, 37273–37292. PMLR, 2023. (pp. 41, 118, 123, 125, and 141.)

Z. Wu, T. Chen, and Q. Ling. Byzantine-resilient decentralized stochastic optimization with robust aggregation rules. *IEEE transactions on signal processing*, 2023. (pp. 49, 56, 59, 62, 68, and 91.)

C. Xie, O. Koyejo, and I. Gupta. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *Uncertainty in Artificial Intelligence*, 261–270. PMLR, 2020. (pp. 49, 62, 67, 100, 103, 129, and 153.)

L. Xu, Y. Dong, G. Wang, R. Zeng, X. Fan, and X. Hu. Nesterov-accelerated robust federated learning over byzantine adversaries. *arXiv preprint arXiv:2511.02657*, 2025. (p. 117.)

S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE Computer Security Foundations Symposium*, CSF '18, 268–282, 2018. (p. 161.)

D. Yin, Y. Chen, R. Kannan, and P. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, 5650–5659. PMLR, 2018. (pp. 49, 57, 91, 117, 120, 128, and 153.)

B. Zhu, L. Wang, Q. Pang, S. Wang, J. Jiao, D. Song, and M. I. Jordan. Byzantine-robust federated learning with optimal statistical rates. In *International Conference on Artificial Intelligence and Statistics*, 3151–3178. PMLR, 2023. (pp. 117, 122, and 137.)

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

Titre : Contributions en apprentissage distribué robuste

Mots clés : Optimisation convexe, Robustesse adversariale, confidentialité différentielle, apprentissage distribué, systèmes distribués

Résumé : Utilisation d' algorithmes d'optimisation distribués en machine learning permet à la fois de laisser les données stockés sur les appareils d'où elles sont issus - offrant plus de souveraineté à leur auteurs - et d'exploiter la capacité des calculs de ces appareils. Cependant, l'utilité pratique de ces méthodes se heurte à deux challenges : garantir la confidentialité des données, mêmes stockés en local, requière l'utilisation d'algorithmes avec garanties formelles de confidentialité ; et 2. la démultiplication d'appareils dans un processus d'optimisation collaborative implique un risque accru d'appareils défaillants (nommé défaillance Byzantine), qui pourrait corrompre l'optimisation en transmettant des messages erronés.
Pour sécuriser les algorithmes à la présence de défaillance Byzantine, nous commençons par proposer des algorithmes de communication robustes lorsque les appareils communiquent de pair à pair dans un graphe de communication. Nous analysons précisément le point de rupture de ces algorithmes par une approche spectral, et montrons leur efficacité pour l'optimisation. Dans un second temps, ces algorithmes sont étudiés par la biais d'une approche duale. Dans une seconde partie, nous proposons une abstraction du problème d'optimisation distribuée robuste en présence d'un serveur central, ainsi que deux méthodes permettant de réduire la complexité communicationnelle. Enfin, nous étudions la faisabilité de moyenne avec garanties de confidentialité dont l'erreur ne dépend de la dimension que à travers la matrice de covariance de la distribution sous-jacente des données, et apportons une réponse négative à cette question en l'absence d'a priori fort sur la matrice de covariance.

Title : Contributions on Robust Distributed Learning

Keywords : Convex optimization, Adversarial Robustness, Differential Privacy, Distributed Learning, Distributed Systems

Abstract : The use of distributed optimization algorithms in machine learning makes it possible both to keep data stored on the devices from which it originates—thereby offering greater control to its creators—and to leverage the computational power of those devices. However, the practical utility of these methods faces two challenges : ensuring data privacy, even when stored locally, requires the use of algorithms with formal privacy guarantees ; and 2. the proliferation of devices in a collaborative optimization process entails an increased risk of faulty devices (known as Byzantine faults), which could corrupt the optimization by transmitting erroneous messages.
To secure algorithms against Byzantine faults, we begin by proposing robust communication algorithms for scenarios where devices communicate peer-to-peer within a communication graph. We precisely analyze the failure point of these algorithms using a spectral approach and demonstrate their effectiveness for optimization. Next, we study these algorithms using a dual approach. In the second part, we propose an abstraction of the robust distributed optimization problem in the presence of a central server, along with two methods for reducing communication complexity. Finally, we investigate the feasibility of mean estimation with confidentiality guarantees, where the error depends on the dimension only through the covariance matrix of the underlying data distribution, and conclude that this is not feasible in the absence of strong prior assumptions about the covariance matrix.

Institut Polytechnique de Paris
91120 Palaiseau, France