DEFENDING DISTRIBUTED SYSTEMS AGAINST ADVERSARIAL ATTACKS: CONSENSUS, CONSENSUS-BASED LEARNING, AND STATISTICAL LEARNING

BY

LILI SU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Doctoral Committee:

Professor Nitin H. Vaidya, Chair
Professor Hagit Attiya, Technion, Israel
Professor Bruce Hajek
Professor R. Srikant

ABSTRACT

A distributed system consists of networked components that interact with each other in order to achieve a common goal. Given the ubiquity of distributed systems and their vulnerability to adversarial attacks, it is crucial to design systems that are "provably secured". In this dissertation, we propose and explore the problems of performing consensus, consensus-based learning, and statistical learning in the presence of malicious components.

- **Consensus**: In this dissertation, we explore the influence of communication range on the computability of reaching iterative approximate consensus. Particularly, we characterize the tight topological condition on the networks for consensus to be achievable in the presence of Byzantine components. Our results bridge the gap of previous work.

- **Consensus-Based Learning**: We propose, to the best of our knowledge, consensus-based Byzantine-tolerant learning problems: *Consensus-Based Multi-Agent Optimization* and *Consensus-Based Distributed Hypothesis Testing*. For the former, we characterize the performance degradation, and design efficient algorithms that can achieve the optimal fault-tolerance performance. For the latter, we propose, as far as we know, the first learning algorithm under which the good agents can collaboratively identify the underlying truth.

- **Statistical Learning**: Finally, we explore distributed statistical learning, where the distributed system is captured by the server-client model. We develop a distributed machine learning algorithm that is able to (1) tolerate Byzantine failures, (2) accurately learn a highly complex model with low local data volume, and (3) converge exponentially fast using logarithmic communication rounds.

To Jiaming Xu.

ACKNOWLEDGMENTS

I was such a lucky person to have the chance to meet the most beautiful minds in the world at UIUC. It is really a hard time for me to say goodbye. I would like to devote the first paragraph of my acknowledgement to UIUC.

I owe my deepest gratitude to my advisor Professor Nitin H. Vaidya, for his guidance and support. I remember clearly, during our first meeting, he emphasized the importance of defining and tackling research problems that are both applicable in practice and technically interesting in theory. This is the first thing he taught me. Many years have passed, and I never ever forgot those words. I highly appreciate all his contributions of time and energy. Prof. Vaidya was always ready to spend hours and days working on the problems together with me. Whenever I had some idea – even if it was a very vague one, he would always be more than happy to discuss with me immediately to explore this idea. I also would like to thank Prof. Hajek, Prof. Srikant, and Prof. Attiya for serving on my doctoral committee. Their insightful comments and suggestions helped improving the dissertation significantly. More importantly, I really appreciate their suggestions on my career plan. My research was financially supported in part by NSF.

Many thanks to my friends, Yingyan Lin, Yuanyuan Hu, Jonathon Ligo, James Yifei Yang, Farzad Farnoud, Christopher Quinn, Xun Gong, and many others, for their companionship and encouragement. My colleagues in the DISC group, Lewis Tseng, Guanfeng Liang, Shegufta B Ahsan, Syeda Persia Aziz, Hang Cui, Zhuolun Xiang, and Shripad Gade, have been very helpful in many ways; I thank all of them. I thank my co-authors, Yudong Chen and Jiaming Xu, for providing the guidance in statistical learning. I also thank Carol Wisniewski for her kind help with various administrative matters.

Finally, I would like to thank my mother, father, brother and sister-in-law for their love and support. Most importantly, I am lucky to have my husband Jiaming Xu always by my side.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This dissertation considers security problems in distributed systems.

A distributed system consists of networked components that communicate and coordinate their actions by passing messages [1].[1] The components interact with each other in order to achieve a common goal. Distributed systems are ubiquitous in both industry and our daily life. For example, we use clusters and networked workstations to analyze large amounts of data, the world wide web for information and resource sharing, and the Internet of Things (IoT) to access a much wider variety of resources.

In distributed systems, components are more vulnerable to adversarial attacks. We are no longer surprised when we are told that some websites, companies, and even cloud systems were attacked by hackers. Sony pictures entertainment and iCloud, respectively, were hacked in 2014. Given the ubiquitousness of distributed systems and their vulnerability to adversarial attacks, it is crucial to design systems that are "provably secured".

In this dissertation, to capture the adversarial behaviors of an unknown fraction of components, we consider the general fault model – the Byzantine fault model, which was introduced in [2] and has received much attention for decades. In this model, it is assumed that up to a certain fraction of the computing components may be compromised by a system adversary, and the compromised components are reprogrammed to behave under the control of the system adversary. In addition, the system adversary is also assumed to have complete knowledge of the system, including the live status of each component (including the non-compromised components). The Byzantine fault model is fundamental in distributed computing and real-world systems for the following reasons.

- Due to the constraint of domain knowledge, detailed descriptions of

[1] Note other information exchanges models exist, for example, shared memory.

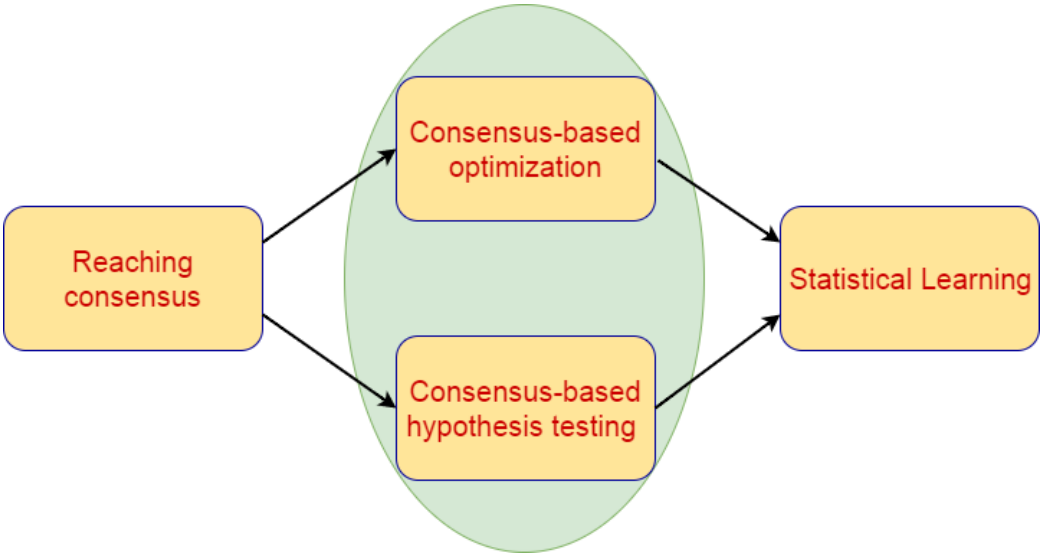

Figure 1.1: Overview of the dissertation

the adversarial attacks may not always be available. If the systems are required to have high fault-tolerance guarantee, the Byzantine fault model can be used.

- Due to its generality, the Byzantine fault model suggests a good starting point to investigate the adversarial attacks in general by providing fundamental insights into the effects of adversarial agents/components.

The focus of this dissertation is to develop, via the concrete topics such as reaching consensus, multi-agent optimization, distributed hypothesis testing, and statistical learning, approaches for charactering the fundamental limits of the system's performance in the presence of Byzantine computing components, and designing efficient algorithms with optimal or near optimal performance.

1.1 Dissertation Overview

We start the dissertation by investigating the consensus problem (Chapter 2). In this problem, a collection of networked processes/agents interact with each other using simple coordination rules in order to aggregate scattered information. Following the consensus chapter, we present two lines of research: Consensus-based multi-agent optimization (Chapter 3) and consensus-based

distributed hypothesis testing (Chapter 4). In both of the above two chapters (Chapters 3 and 4), we mainly focus on the family of algorithms in which the agents/processes interact with each other using the simple coordination rules that are similar to the one discussed in Chapter 2. In the subsequent chapter (Chapter 5), observing the trends in collaborative machine learning (mobile + cloud computing), we explore the problem of performing distributed machine learning in the adversarial setting. One key distinction between the distributed system assumed in Chapter 5 and that discussed in Chapters 2, 3 and 4 is the existence of a parameter server, which is used for the inter-agent coordination.

The slightly detailed problem descriptions and our contributions can be found in the following sections.

1.2 Reaching Consensus

In a consensus problem, a collection of components (referred as processors in distributed computing) are required to reach a common decision. Reaching consensus in a distributed system is one of the fundamental problems, and thus has received intense attention. The existing work assumes either local communication or full message forwarding. In the former, each processor can only exchange messages with its neighbors. In the latter, processors can exchange information with each other as long as there is a route. We observe that in some communication networks, processors' communication power is stronger than that in the local model, but is still limited – supporting multi-hop communication.

This dissertation addresses the impact of communication range on the computability of reaching consensus asymptotically. Specifically, we assume that in each iteration the processors can only communicate with other processors that are up to ℓ hops away, where ℓ is a positive integer. For a given ℓ, we identified a necessary and sufficient condition on the network structure for the existence of correct iterative algorithms that achieve asymptotical consensus in the presence of Byzantine agents. Our results bridged the above two lines of literature. In particular, our tight condition generalized the tight condition identified in existing work for $\ell = 1$, i.e., local communication. For $\ell \geq \ell^*$, where ℓ^* is the length of a longest cycle-free path in the given network,

our condition is equivalent to the tight conditions obtained for full message forwarding communication.

1.3 Consensus-Based Multi-Agent Optimization

In consensus-based multi-agent optimization, each agent keeps a *local* cost function that is initially known only to itself, and the networked agents want to collectively reach agreement on a global decision x such that a global objective that properly aggregates these local costs is minimized. From the previous description, it can be seen that consensus is part of the requirements satisfied by any solution for the optimization problem of interest.

Assuming that every agent is non-faulty, a typical goal of the multi-agent optimization problem [3, 4, 5, 6, 7, 8] is to minimize the average of the local cost functions of individual agents. Precisely, let h_i be the local cost function associated with agent i. The goal of (failure-free) multi-agent optimization is to have all the agents reach agreement on x that minimizes

$$\frac{1}{n}\sum_{i=1}^{n} h_i, \tag{1.1}$$

where n is the total number of agents in the system. Due to its many potential applications, multi-agent optimization has been a topic of significant research activity [3, 4, 5, 6, 7, 8]. The applications include distributed machine learning and distributed resource allocation. In a distributed machine learning problem [9], x represents parameters that need to be learned, using data available to a collection of agents. The local objective h_i denotes a loss function for agent i that depend on data initially available to agent i only. In the resource allocation problem, the argument x represents allocation of shared resources to the agents, and the local cost functions depends on the *fairness* of the resource allocation. The global objective is to allow the agents to collaboratively agree on the most fair resource allocation. Many problems in distributed robotics are also represented in the above form [10].

While the failure-free version of the above problem is well-understood, very little attention has been paid to the scenario when some agents may be malicious. In this dissertation, we characterize the performance degradation caused by the existence of malicious agents, and design efficient algorithms

that can achieve the optimal (the best possible) performance.

1.4 Consensus-Based Distributed Hypothesis Testing

Collaborative distributed hypothesis testing over multi-agent networks has received a significant amount of attention. To avoid the complexity of Bayesian learning, a non-Bayesian learning framework that combines local Bayesian learning with consensus was proposed by Jadbabaie et al. [11], and has attracted much attention since then. The prior work implicitly assumes that the networked agents are reliable in the sense that they correctly follow the specified distributed algorithm. However, in some practical multi-agent networks, this assumption may not hold. For example, in social networks, it is possible that some agents are adversarial, and try to prevent the true state from being learned by the good agents. This dissertation addresses the problem of developing distributed learning algorithms that are robust to adversarial attacks. We proposed the first Byzantine-resilient learning algorithm [12], and characterized a tight network identifiability condition in [13] – the extended version of [12]. At first glance, our learning rule is counter-intuitive: by applying the cumulative likelihood, the "old information" contained in the previous signals is used again and again in updating local pseudo beliefs. It turns out that this learning rule enables us to deal with the dependency between the pseudo beliefs and the effective message propagation. This dependency is crucial in our adversarial attacks setting.

1.5 Distributed Statistical Learning

Many efficient distributed machine learning algorithms [14, 15] and system implementations [16, 17, 18, 19] have been proposed and studied. Prior work mostly focuses on the traditional "training within cloud" framework where the model training process is carried out within the cloud infrastructures. In this framework, distributed machine learning is secured via system architectures, hardware devices, and monitoring [20, 21, 22]. This framework faces significant privacy risk, as the data has to be collected from owners and stored within the clouds. Although a variety of privacy-preserving solutions

have been developed [23, 24], privacy breaches still occur frequently, with recent examples including iCloud leaks of celebrity photos [25] and PRISM surveillance program [26].

To address privacy concerns, a new machine learning paradigm called *Federated Learning* was proposed by Google researchers [27, 28]. It aims at learning an accurate model without collecting data from owners and storing the data in the cloud. The training data is kept locally on the owners' computing devices, which are recruited to participate directly in the model training process and hence function as working machines. Google has been intensively testing this new paradigm in their recent projects such as *Gboard* [28], the Google Keyboard. Compared to "training within cloud", Federated Learning has lower privacy risk, but inevitably becomes less secure. In particular, it faces the following three key challenges:

- Security: The devices of the recruited data owners can be easily reprogrammed and completely controlled by external attackers, and thus behave adversarially.

- Small local datasets versus high model complexity: While the total number of data samples over all data owners may be large, each individual owner may keep only a small amount of data, which by itself is insufficient for learning a complex model.

- Communication constraints: Data transmission between the recruited devices and the cloud may suffer from high latency and low throughout. Communication between them is therefore a scarce resource, whose usage should be minimized.

In the last part of this dissertation, we address the above challenges faced by Federated Learning by developing a new iterative distributed machine learning algorithm that is able to (1) tolerate Byzantine failures, (2) accurately learn a highly complex model with low local data volume, and (3) converge exponentially fast using logarithmic communication rounds.

CHAPTER 2

REACHING CONSENSUS

2.1 Introduction

The problem of reaching consensus concerns a collection of processes that are connected by a network. Among the networked processes, some unknown processes may be compromised by an adversary, and be reprogrammed to behave arbitrarily, and adversarially try to degrade the behavior of the system. This fault model is referred to as Byzantine fault [29]. In this chapter, we are interested in the approximate Byzantine consensus problem, wherein all the *faulty-free* processes reach consensus asymptotically (approximately in finite time). In particular, we focus on the algorithms under which each process communicates with other processes that are up to l hops away via synchronous FIFO (first-in-first-out) communication channels and maintains *minimal* states across iterations – no messages received during previous iterations will be used in the state updates.

The Byzantine fault-tolerance problem was first introduced in [30], and is one of the most fundamental problems in distributed computing. [31] showed that the fault-tolerant consensus problem cannot be solved in an *asynchronous* system even in the presence of only one crash failure. A process suffering crash faults may unexpectedly stop participating in the specified algorithms/protocols. As one way to circumvent this impossibility result, the notion of *approximate consensus* was introduced in [32] by requiring that the nodes agree with each other only asymptotically (approximately in finite time). The notion of approximate consensus is of interest in *synchronous* system as well [32, 33, 34]. The discussion in this chapter applies to synchronous systems.

Let n be the total number of processes and f be the upper bound on the number of faulty processes in the system. The actual number of compro-

mised (faulty) processes may vary across executions, and may not known to the fault-free processes. However, each fault-free process knows that in each execution at most f processes may be faulty. In networks with bidirectional links, approximate consensus is achievable if and only if the network node-connectivity is at least $2f + 1$ and less than one third of the processes can be faulty, i.e., $n \geq 3f + 1$ [35]. Relaxing the bidirectional communication assumption, a tight condition for directed graphs was presented in [36]. There has been increasing interest in designing iterative variants of approximate Byzantine consensus where only local knowledge of the network topology (and local communication) is needed, and processes carry minimal state across iterations [37, 38, 39, 33, 34]. [39] studied the convergence rate of approximate consensus algorithms over complete networks. [33, 34] considered arbitrary directed networks and derived tight (necessary and sufficient) topological conditions on the communication network. While [34] investigated the Byzantine fault model, [33] considered a restricted fault model in which the faulty nodes are restricted to sending identical messages to their neighbors. When $f = 0$, such iterative approximate consensus algorithms have been well-studied in the cooperative control community [40, 41, 42].

To the best of our knowledge, no attempts have been made to investigate the impact of each process's communication range on the network condition for a correct iterative approximate consensus algorithm to exist. In this chapter, we model the network as a directed graph, and we focus on the family of algorithms in which a process communicates with processes that are up to l hops away by forwarding messages through intermediate processes. The directed graph model is motivated by the presence of directed links in wireless networks. Our goal is to identify a necessary and sufficient condition on the network structure for the algorithms of interest to exist.

Contributions Our main contribution is to identify a necessary and sufficient condition on the network structure for a given l, named Condition NC for a given l. Informally speaking, our Condition NC states that for any four set process partition L, R, C, and F such that both L and R are nonempty and $|F| \leq f$, with up to l–hop communication, at least one process in L is influenced by processes in $R \cup C$ or at least one process in R is influenced by processes in $L \cup C$. Condition NC will be formally stated in Section 2.3. Our sufficiency proof is shown by constructing a simple iterative algorithm,

whose trim function is defined based on a *minimal messages cover* property that we introduce in this chapter.

The tight condition we found is consistent with the tight condition identified in [34] when only local communication is allowed, i.e., $l = 1$. For $l \geq l^*$, where l^* is the length of a longest cycle-free path in the given network, our condition is equivalent to the tight condition for consensus in undirected networks [35] as well as exact consensus in directed networks [36].

Organization The rest of this chapter is organized as follows. Section 2.2 presents our models and the structure of the iterative algorithms considered in our work. Our necessary condition is presented in Section 2.3, and its sufficiency is proved constructively in Section 2.4. The correspondence between our condition and the results in [32, 35, 36] is discussed in Section 2.5. Section 2.6 discusses possible relaxations of our fault model and concludes the chapter.

2.2 Problem Setup and Structure of Iterative Algorithms

Communication model The system is assumed to be *synchronous*. The communication network is modeled as a simple *directed* graph G with self-loop at each process. Denote $\mathcal{V}(G) = \{1, \ldots, n\}$ as the set of n processes, where $n \geq 2$, and $\mathcal{E}(G)$ as the set of directed links between processes in $\mathcal{V}(G)$. In general, $\mathcal{V}(\cdot)$ and $\mathcal{E}(\cdot)$ are two functions defined over graphs that return the vertex set and the edge set, respectively, for a given graph. For instance, let H be a graph, then $\mathcal{V}(H)$ and $\mathcal{E}(H)$ are the vertex set and edge set of H. In this chapter, we use "process" and "node" interchangeably, and use "link" and "edge" interchangeably.

Let l be a positive integer. For each node i, let N_i^{l-} be the set of nodes that can reach node i via at most l hops. Similarly, denote the set of nodes that are reachable from node i via at most l hops by N_i^{l+}. Note that $i \in N_i^{l-}$ and $i \in N_i^{l+}$. When $l = 1$, we write N_i^{1-} and N_i^{1+} as N_i^- and N_i^+, respectively, for simplicity. We also assume each node i knows the entire network topology.

Node i may send messages to node j via different i, j–paths with intermediate nodes on an i, j–path forwarding messages accordingly. To capture

this distinction in transmission routes, we represent a message as a tuple $m = (w, P)$, where $w \in \mathbb{R}$ is the message content, and P indicates the path via which message m should be transmitted. It is assumed that the network layer in the system delivers the messages along the specified paths. The intermediate nodes on the paths do not view the message values (i.e., the message values are not used by intermediate nodes in performing consensus). Four functions are defined over message m, corresponding to message content, transmission route, message source, and message destination, respectively. Specifically, for $m = (w, P)$, let function value be $\mathsf{value}(m) = w$ and let path be $\mathsf{path}(m) = P$, whose images are the first entry (message content) and the second entry (message route), respectively, of message tuple $m = (w, P)$. In addition, functions source and $\mathsf{destination}$ are defined by $\mathsf{source}(m) = i$ and $\mathsf{destination}(m) = j$ if P is an i, j–path, i.e., message $m = (w, P)$ is sent from node i (source) to node j (destination).

Fault model Let $\mathcal{F} \subseteq \mathcal{V}(G)$ be the collection of faulty nodes in the system. We consider the Byzantine fault model with up to f nodes becoming faulty, i.e., $|\mathcal{F}| \leq f$. We assume that each fault-free node knows f, but does not know the actual number of faulty agents $|\mathcal{F}|$ in a given execution. A faulty node may tamper with the message arbitrarily. Possible misbehavior includes sending incorrect and mismatching (or inconsistent) messages to different neighbors. In addition, a faulty node $k \in \mathcal{F}$ may tamper with message m if it is in the transmission path, i.e., $k \in \mathcal{V}(\mathsf{path}(m))$. However, faulty nodes may only tamper with $\mathsf{value}(m)$, leaving $\mathsf{path}(m)$ unchanged. This constraint is placed for ease of exposition; later in Section 2.6 we relax this constraint. Thus, the fault model considered is the general Byzantine fault model [29]. Faulty nodes are also assumed to have complete knowledge of the algorithm execution, including the states of all nodes, contents of messages that the other nodes send to each other, and the algorithm specification, so that they may potentially collaborate with each other adaptively.

Iterative approximate Byzantine consensus (IABC) algorithms The algorithms considered in this chapter proceed in iterations, and each iteration has the following structure: Each node i maintains state v_i, with $v_i[t]$ denoting the state of node i at the *end* of the t-th ($t > 0$) iteration, and $v_i[t-1]$ denoting the state of node i at the *start* of the t-th iteration. The

initial state of node i, $v_i[0]$, is equal to the initial *input* provided to node i. The IABC algorithms of interest will require each node i to perform the following three steps in iteration t. Note that the faulty nodes may deviate from this specification.

Algorithm 1: IABC: Generic code

1 *Transmit step:* Transmit messages of the form $(v_i[t-1], P)$ on each l–hop path P (including self-loops) to nodes in N_i^{l+};

2 *Receive step:* Receive messages from N_i^{l-} for which destination is i. When node i expects to receive a message from a path but does not receive the message, the message value is assumed to be equal to some default value. Let $\mathcal{M}_i[t]$ be the set of messages that node i received in this step;

3 *Update step:* Update $v_i[t]$ as

$$v_i[t] = Z_i(\mathcal{M}_i[t]). \tag{2.1}$$

In the Transmit step and Receive step of an IABC algorithm, nodes exchange messages with nodes that are up to l hops away. As noted previously, the network layer of the system forwards each message to its destination along the path specified for the message. Then in the Update step, node i updates its state using a transition function Z_i, where Z_i is a part of the specification of the algorithm, and takes as input the set $\mathcal{M}_i[t]$. Note that $v_i[t]$ only depends on $\mathcal{M}_i[t]$–the messages collected by node i at iteration t (which includes $v_i[t-1]$). No information collected/obtained during previous iterations will affect the update step in iteration t. Intuitively speaking, fault-free node i is assumed to have no memory across iterations other than its most recent state $v_i[t-1]$ (maintain minimal states across iterations). Algorithms with similar structure are considered in prior work in the distributed computing community as well [33, 34], and are also well studied in the cooperative control community [40, 41, 42] for the case when $l = 1$.

Remark 1. *Although only minimal states are carried across iterations, since the size of $\mathcal{M}_i[t]$ is exponential in l, the space complexity of each fault-free node in an IABC algorithm is also exponential in l. As can be seen later, there is a tradeoff between the space complexity and the minimal topological condition on the underlying network for asymptotic consensus to be achieved.*

Let $U[t]$ be the largest state among the fault-free nodes at the end of the t-th iteration, i.e., $U[t] = \max_{i \in \mathcal{V}-\mathcal{F}} v_i[t]$. By convention, $U[0]$ is the largest input among the fault-free nodes. Similarly, we define $\mu[t] = \min_{i \in \mathcal{V}-\mathcal{F}} v_i[t]$ and $\mu[0]$ to be the smallest input among the fault-free nodes. For an IABC algorithm to be correct, the following two conditions must be satisfied:

- *Validity:* $\forall t > 0,\ \mu[t] \geq \mu[0]$ and $U[t] \leq U[0]$;
- *Convergence:* $\lim_{t \to \infty}\ (U[t] - \mu[t]) = 0$.

The above validity condition is a canonical condition adopted in the distributed computing community [35, 31]. Without such validity condition, all fault-free nodes may trivially agree on some default value which may be independent of the system inputs. Such trivial algorithms may not be satisfactory in many applications, especially in the scenario where the convex hull of the inputs at fault-free nodes forms a safe area, and any deviation from this safe area will induce a forbiddingly high penalty.

Our goal is to identify the necessary and sufficient conditions on graph G for the existence of a *correct* IABC algorithm (i.e., an algorithm satisfying the above validity and convergence conditions) for a given l.

2.3 Necessary Condition

For a correct IABC algorithm to exist, the network G must satisfy the condition presented in this section. First, we introduce some definitions.

Definition 1. *Suppose $W \subseteq \mathcal{V}(G)$ and $x \in \mathcal{V}(G)$ such that $x \notin W$. A W, x–path is a path from some vertex $w \in W$ to vertex x. A set $S_l \subseteq \mathcal{V}(G)$ with $x \notin S_l$ is an l–*restricted vertex cut *if the deletion of S_l disconnects all W, x–paths of length at most l. The l–*restricted W, x–connectivity*, denoted by $\kappa_l(W, x)$, is defined by*

$$\kappa_l(W, x) = \min_{S_l : S_l \text{ is an } l\text{–restricted } W, x\text{–cut}} |S_l|.$$

*A set of vertices S is a W, x–*vertex cut *if the removal of set S disconnects*

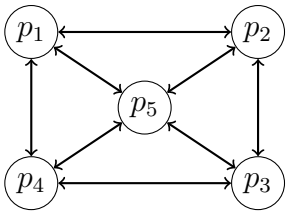

Figure 2.1: In this system, there are five nodes p_1, p_2, p_3, p_4 and p_5; all communication links are bi-directional; and at most one node can be adversarial, i.e., $f = 1$.

*all W, x–paths. The W, x–*connectivity*, denoted by $\kappa(W, x)$, is defined by*

$$\kappa(W, x) = \min_{S:S \text{ is a } W,x\text{–cut}} |S|.$$

The second part of Definition 1 is the classic definition of node connectivity in graph theory [43], which is a global notion. The first part of Definition 1 adapts node connectivity to our multi-hop communication setting by restricting the path length of interest. Note that $\kappa_l(W, x) = \kappa(W, x)$ for all $l \geq l^*$, and that $\kappa_1(W, x) = |W \cap N_x^-|$ – recalling that l^* is the length of a longest cycle-free path in the given network.

In general, $\kappa_l(W, x) \neq \kappa(W, x)$ and $\kappa_l(W, x) \leq \kappa_{l+1}(W, x)$ for all l. Consider the system depicted in Figure 2.1; via enumeration it can be seen that

$$\kappa\left(\{p_2, p_3\}, p_1\right) = 2 \geq 1 = \kappa_1\left(\{p_2, p_3\}, p_1\right).$$

Intuitively speaking, in general, the stronger the communication capability of each node (the larger l), the harder it is to prevent one node from being influenced by other nodes.

Definition 2. *For non-empty disjoint sets of nodes A and B in graph G, we say $A \Rightarrow_l B$ if and only if there exists a node $i \in B$ such that $\kappa_l(A, i) \geq f+1$; $A \not\Rightarrow_l B$ otherwise.*

Intuitively, $A \Rightarrow_l B$ implies the existence of a node i in B that can be influenced by fault-free nodes in A despite the presence of Byzantine nodes.

Let $F \subseteq \mathcal{V}(G)$ be a set of vertices in G. Denote the subgraph of G induced by vertex set $\mathcal{V}(G) - F$ by G_F.[1] We describe the necessary and sufficient

[1]Subgraph of G induced by vertex set $S \subseteq \mathcal{V}(G)$ is the subgraph H with vertex set S such that $\mathcal{E}(H) = \{(u, v) \in \mathcal{E}(G) : u, v \in S\}$. Recall that $\mathcal{V}(\cdot)$ and $\mathcal{E}(\cdot)$ are the vertex set and edge set, respectively, of a given graph.

condition below, termed *Condition NC*, whose necessity is proved in Theorem 1 and sufficiency is shown constructively in Section 2.4.

Condition NC: For any node partition L, C, R, F of G such that $L \neq \emptyset, R \neq \emptyset$ and $|F| \leq f$, at least one of the two conditions below must be true: (i) $R \cup C \Rightarrow_l L$ in G_F; (ii) $L \cup C \Rightarrow_l R$ in G_F.

Condition NC requires that, for any node partition L, C, R, F, either the nodes in $R \cup C$ are able to collectively influence a node in L in G_F or vice versa. When $l = 1$, Condition NC is equivalent to the following condition, which is shown to be both necessary and sufficient [34].
"For any node partition L, C, R, F of G such that $L \neq \emptyset, R \neq \emptyset$ and $|F| \leq f$, at least one of the two conditions below must be true: (i) there exists a node $i \in L$ such that $\left|(R \cup C) \cap N_i^-\right| \geq f + 1$; (ii) there exists a node $j \in R$ such that $\left|(L \cup C) \cap N_j^-\right| \geq f + 1$."

Our proof of the next theorem shares the structure of the proof of Theorem 1 in [34].

Theorem 1. *Suppose that a correct IABC algorithm exists over G. Then G satisfies Condition NC.*

Proof. We prove the theorem by contradiction. Let us assume that a correct IABC algorithm exists, and there exists a partition L, C, R, F of G such that $L \neq \emptyset, R \neq \emptyset$ and $|F| \leq f$, $R \cup C \not\Rightarrow_l L$ in G_F and $L \cup C \not\Rightarrow_l R$ in G_F.

Execution E Consider an execution E in which all the nodes in F are faulty, and the other nodes in sets L, C, R are fault-free. Note that the fault-free nodes are not aware of the identities of the faulty nodes. In addition, assume that (i) each node in L has initial input 0, (ii) each node in R has initial input 2ϵ, where $\epsilon > 0$, and (iii) each node in C has initial input in the interval $[0, 2\epsilon]$. The behavior of the Byzantine faulty nodes (i.e., nodes in F) in execution E is as follows. In the *Transmit step* of iteration 1, each faulty node $k \in F$ sends value 0 to the nodes in $N_k^{l+} \cap L$, sends value 2ϵ to the nodes in $N_k^{l+} \cap R$, and sends ϵ to the nodes in $N_k^{l+} \cap C$. When forwarding message m of the form $m = (w, P)$, for which $k \in F$ is an intermediate node on path P, if $\mathsf{destination}(m) \in L$, node k sets $\mathsf{value}(m) = 0$, and if $\mathsf{destination}(m) \in R$, node k sets $\mathsf{value}(m) = 2\epsilon$. In this case, all the messages received by each node in C contain values in the range $[0, 2\epsilon]$. Therefore, to satisfy the validity

condition, each node in C must choose its state at the end of iteration 1 to be also in the range $[0, 2\epsilon]$.

Consider an arbitrary node $i \in L$. Since $|F| \leq f$, we have $|N_i^{l-} \cap F| \leq f$. In addition, $C \cup R \not\twoheadrightarrow_l L$ in G_F implies that $\kappa_l(C \cup R, i) \leq f$. Let S_l be a minimum l–restricted $(C \cup R, i)$–cut in G_F. Then $|S_l| \leq f$.

Observe that all the paths of length up to l hops from nodes in $R \cup C$ to node i contain at least one node in $F \cup S_l$. Due to the above faulty behaviors in execution E, node i receives value 0 on all paths that contain at least one node in F. Thus, node i may receive values > 0 only on paths that do not contain nodes in F – such paths necessarily include at least one node in S_l.

Execution E' Now consider another execution, denoted by E', in which the nodes in S_l are faulty, while the remaining nodes are fault-free, and all the fault-free nodes have initial input 0. Then, in this execution, node i may receive values > 0 only on paths that include at least one node in S_l, and will receive value 0 on the remaining paths. Since the nodes in S_l are faulty, it is possible that all the message values > 0 should have been 0, but were tampered with by nodes in S_l. In this case, to satisfy the validity condition, node i must set its new state $v_i[1]$ as 0.

Notice that, from the perspective of node i, executions E and E' appear identical – it receives an identical set of messages in both cases. Thus, in the execution E also node i must set its new state $v_i[1]$ after iteration 1 as 0.

The above argument shows that the state of each node in L will remain 0 after iteration 1 (recall that its state was 0 before iteration 1 as well). By an analogous argument, we can show that each node in R will maintain its state equal to 2ϵ. We have already shown that after iteration 1, the state of the nodes in C remains in $[0, 2\epsilon]$, analogous to the initial state. Applying this argument inductively, it follows that the state of the nodes in L and R remains as 0 and 2ϵ, respectively. Since L and R both contain fault-free nodes, the convergence requirement is not satisfied. This contradicts the assumption that a correct iterative algorithm exists.

□

The above necessary condition is in general weaker than the necessary condition derived under single-hop message transmission model in [34], i.e.,

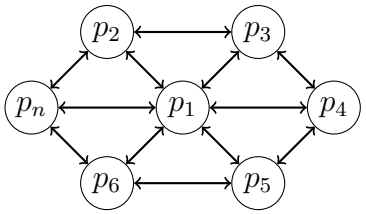

Figure 2.2: In this system, there are n nodes $p_1, \ldots, p_n$; all communication links are bi-directional; and at most one node can be adversarial, i.e., $f = 1$. Nodes $p_2, \ldots, p_n$ form a cycle of length $n-1$ and these nodes are all connected to node p_1.

when $l = 1$. Consider the system depicted in Figure 2.1. The topology of this system does not satisfy the necessary condition derived in [34] for $l = 1$ and $f = 1$. Since in the node partition $L = \{p_1, p_4\}, R = \{p_2, p_3\}, C = \emptyset$ and $F = \{p_5\}$, neither $L \cup C \Rightarrow_l R$ in G_F nor $R \cup C \Rightarrow_l L$ in G_F holds for $l = 1$ and $f = 1$. However, via enumeration it can be seen that the graph, depicted in Figure 2.1, satisfies Condition NC for $l \geq 2$ and $f = 1$.

Nevertheless, the larger the communication range l, the higher space complexity of the IABC algorithm.

It follows from the definition of Condition NC that if a graph G satisfies Condition NC for $l \in \{1, \ldots, n-1\}$, then G also satisfies Condition NC for all $l' \geq l$. Let l_0 be the smallest integer for which G satisfies Condition NC, where $l_0 = n$ by convention if G does not satisfy Condition NC for any $l \in \{1, \ldots, n-1\}$. We observe that in general given a graph G, the diameter of G can be arbitrarily smaller than l_0. For instance, the diameter of the graph depicted in Figure 2.2 is 2. However, for the depicted graph, $l_0 \geq \frac{n+1}{4}$ when $n = 4k + 3$ and $f = 1$, where k is a positive integer. So l_0 is much larger than 2 for large n. To see $l_0 \geq \frac{n+1}{4}$, consider the node partition $F = \{p_1\}, C = \emptyset, L = \{p_2, \ldots, p_{\frac{n+1}{2}}\}$ and $R = \{p_{\frac{n+3}{2}}, \ldots, p_n\}$. For $f = 1$, in order to have $L \cup C \Rightarrow_l R$ or $R \cup C \Rightarrow_l L$ hold in G_F for this particular node partition, it must be hold that $l \geq \frac{n+1}{4}$. Thus $l_0 \geq \frac{n+1}{4}$.

Similar to [34], as stated in our next corollary, Condition NC for general l also implies a lower bound on n and a lower bound on each node's incoming degree, both of which are independent of l.

Corollary 1. *For $f > 0$, if G satisfies Condition NC, then $n \geq 3f + 1$, and each node must have at least $2f + 1$ incoming neighbors other than itself, i.e., $|N_i^- - \{i\}| \geq 2f + 1$.*

Proof. The proof of Corollary 1 is similar to the proof in [34], and is presented below for completeness.

We first show the claim that $n \geq 3f + 1$. The proof is by contradiction. Suppose that $2 \leq n \leq 3f$. Since $f > 0$, we can partition $\mathcal{V}(G)$ into sets L, R, C, F such that $1 \leq |L| \leq f$, $1 \leq |R| \leq f$, $0 \leq |F| \leq f$ and $|C| = 0$, i.e., C is empty. Since $1 \leq |L \cup C| = |L| \leq f$ and $1 \leq |R \cup C| = |R| \leq f$, we have $L \cup C \not\Rightarrow_l R$ in G_F and $R \cup C \not\Rightarrow_l L$ in G_F, respectively. This contradicts the assumption that G satisfies Condition NC. Thus, $n \geq 3f + 1$.

It remains to show $|N_i^- - \{i\}| \geq 2f + 1$. Suppose that, contrary to our claim, there exists a node i such that $|N_i^- - \{i\}| \leq 2f$. Define set $L = \{i\}$ and partition $N_i^- - \{i\}$ into two sets F and H such that $|H| = \lfloor |N_i^- - \{i\}|/2 \rfloor \leq f$ and $|F| = \lceil |N_i^- - \{i\}|/2 \rceil \leq f$. Define $R = \mathcal{V}(G) - F - L = \mathcal{V}(G) - F - \{i\}$ and $C = \emptyset$. Since $|\mathcal{V}(G)| = n \geq \max(2, 3f + 1)$ and $f > 0$, it is true that R is non-empty. From the construction of R, we have $N_i^- \cap R = H$, and $|N_i^- \cap R| = |H| \leq f$. Since $L = \{i\}$, $|N_i^- \cap R| \leq f$ and $C = \emptyset$, it follows that $R \cup C \not\Rightarrow_l L$. On the other hand, as $f > 0$ and $|L| = 1 < f + 1$, we have $L \cup C \not\Rightarrow_l R$ in G_F. This violates the assumption that G satisfies Condition NC, proving the corollary. □

Note that Corollary 1 also characterizes a lower bound on the density of G, that is $|\mathcal{E}(G)| \geq n(2f + 2)$, including self-loops, which is independent of the communication range l as well.

2.3.1 Equivalent Characterization of Condition NC

Informally speaking, Condition NC describes the information propagation property in terms of four set partitions. In this subsection, an equivalent condition of Condition NC is proposed, which will be used in the sufficiency proof in Section 2.4. This alternative characterization is based on characterizing the structure of a family of special subgraphs, termed as *reduced graphs*, of the power graph G^l. The new condition suggests that all fault-free nodes will be influenced by a common collection of fault-free nodes.

Definition 3. *[44] Let $K_1, \ldots, K_k$ be the strongly connected components (SCCs) of G. The graph of SCCs, denoted by G^{SCC}, is defined as follows : (i) nodes in G^{SCC} are $K_1, \ldots, K_k$; and (ii) there is an edge (K_i, K_j) in G^{SCC}*

if there is some $u \in K_i$ *and* $v \in K_j$ *such that* (u, v) *is an edge in* G.
K_h *is a* source component *if it is* <u>not</u> *reachable from any other node in* G^{SCC}.

It is known that the G^{SCC} is a directed acyclic graph (DAG [44]). Thus, a graph G has at least one source component.

Definition 4. *[43] The* l*–th power of a graph* G*, denoted by* G^l*, is a multi-graph*[2] *with the same set of vertices as* G *and a directed edge between vertices* u, v *is defined by a path of length*[3] l *from* u *to* v *in* G.

Note that up to l–multi-hop communication can be viewed as single-hop communication with the l–powered graph. Note that a Byzantine node k can corrupt the messages whose transmission paths contain k, explained as follows. There is a one-to-one correspondence between an edge e in G^l and a path of length l in G (including self-loops). A path of length 1 between vertices u and v in G exists if (u, v) is an edge in G. A path of length 2 between vertices u and v in G exists for every vertex w such that (u, w) and (w, v) are edges in G. Then for a given graph G with self-loop at each node, the $(u, v)^{th}$ element in the square of the adjacency matrix of G counts the number of paths of length at most 2 in G. Similarly, the $(u, v)^{th}$ element in the l–th power of the adjacency matrix of G gives the number of paths of length l between vertices u and v in G.

Let e be an edge in G^l, and $P(e)$ be the corresponding path in G; we say an edge e in G^l is covered by node set S if $\mathcal{V}(P(e)) \cap S \neq \emptyset$, i.e., path $P(e)$ passes through a node in S.

Definition 5. *For a given graph* G *and* $F \subseteq \mathcal{V}(G)$*, let* E *be the set of edges in* G^l *that are covered by node set* F*, i.e.,* $E \triangleq \{e \in \mathcal{E}(G^l) : \mathcal{V}(P(e)) \cap F \neq \emptyset\}$. *For each node* $i \in \mathcal{V}(G) - F$*, choose* $C_i \subseteq N_i^{l-} - \{i\}$ *such that* $|C_i| \leq f$. *Let* E_i *be the set of incoming edges of node* i *in* G^l *that are covered by node set* C_i*, i.e.,* $E_i \triangleq \{e \in \mathcal{E}(G^l) :$ *head of* e *is node* i*, and* $\mathcal{V}(P(e)) \cap C_i \neq \emptyset\}$. *A reduced graph of* G^l*, denoted by* $\widetilde{G^l}_F$*, is a subgraph of* G^l *whose node set and edge set are defined by (i)* $\mathcal{V}(\widetilde{G_F^l}) \triangleq \mathcal{V}(G) - F$*; and (ii)* $\mathcal{E}(\widetilde{G_F^l}) \triangleq \mathcal{E}(G^l) - E - \cup_{i \in \mathcal{V}(G)-F} E_i$*, respectively.*

[2]A multigraph (or pseudograph) is a graph which is permitted to have multiple edges between each vertex pair, that is, edges that have the same end nodes. Thus two vertices may be connected by more than one edge.

[3]Recall that we assume that each node in G has a self-loop.

Note that for a given graph G and a given node set F, multiple reduced graphs may exist because for each node $i \in \mathcal{V}(G) - F$, there may be multiple choices of C_i. Let us define set R_F to be the collection of all reduced graph of G^l for a given F, i.e.,

$$R_F \;=\; \{\widetilde{G^l}_F : \widetilde{G^l}_F \text{ is a reduced graph of } G^l\}. \tag{2.2}$$

Note that G^l_F, the l–th power of G_F, itself is a reduced graph of G^l, where we choose $C_i = \emptyset$ for each $i \in \mathcal{V}(G) - F$. Thus R_F is nonempty. In addition, $|R_F|$ is finite since the graph G is finite.

Theorem 2. *Graph G satisfies Condition NC if and only if every reduced graph $\widetilde{G^l}_F$ contains exactly one* source component.

Proof. We first show that if graph G satisfies Condition NC, then every reduced graph of G^l contains exactly one source component.

For any reduced graph $\widetilde{G^l}_F$, the meta-graph $(\widetilde{G^l}_F)^{SCC}$ is a DAG and finite. Thus, at least one source component must exist in $\widetilde{G^l}_F$. We now prove that $\widetilde{G^l}_F$ cannot contain more than one source component. The proof is by contradiction. Suppose that there exists a set $F \subseteq \mathcal{V}(G)$ with $|F| \leq f$, and a reduced graph $\widetilde{G^l}_F$ corresponding to F, such that $\widetilde{G^l}_F$ contains at least two source components, say K_1 and K_2, respectively. Let $L = K_1$, $R = K_2$, and $C = \mathcal{V}(G) - F - L - R$. Then L, R, C together with the given F form a node partition of G such that $L \neq \emptyset, R \neq \emptyset$ and $|F| \leq f$. Let C_i be the sets (for each i) used to construct this reduced graph.

Since graph G satisfies Condition NC, without loss of generality, assume that $R \cup C \Rightarrow_l L$ in G_F, i.e., there exists a node $i \in L$ such that $\kappa_l(R \cup C, i) \geq f + 1$ in G_F. On the other hand, since L is a source component in $\widetilde{G^l}_F$, by the definition of reduced graph, we know $\exists\ C_i$, such that $|C_i| \leq f$ and that all paths from $R \cup C$ to node i of length at most l in G_F are covered by C_i. Thus, C_i is an l–restricted $(R \cup C, i)$–cut of G_F. This contradicts the fact that $\kappa_l\,(R \cup C, i) \geq f + 1$ in G_F.

To complete the equivalence proof it remains to show that if every reduced graph contains exactly one source component, then the graph must satisfy Condition NC.

Suppose, on the contrary, that G does not satisfy Condition NC. Then there exists a node partition L, R, C and F of G with L, R are nonempty

and $|F| \leq f$ such that $L \cup C \not\Rightarrow_l R$ in G_F and $R \cup C \not\Rightarrow_l L$ in G_F. By the definition of the relation $\not\Rightarrow_l$, we have $\kappa_l(L \cup C, i) \leq f$ in G_F, $\forall\ i \in R$, and $\kappa_l(R \cup C, j) \leq f$ in G_F,$\forall\ j \in L$. That is, all paths of length l (including self-loops) from $L \cup C$ to $i \in R$ can be covered by f nodes, and all paths of length l (including self-loops) from $R \cup C$ to $j \in L$ can be covered by f nodes. Next we construct a reduced graph with at least two source components, which contradicts the assumption that every reduced graph of G^l contains a unique source component.

For the given F, consider the reduced graph constructed as follows: choose C_i to be a minimum l–restricted $(L \cup C, i)$-cut in G_F for each $i \in R$, choose C_j to be a minimum l–restricted $(R \cup C, j)$-cut in G_F for each $j \in L$, and choose C_k to be an arbitrary set such that $|C_k| \leq f$ for each $j \in C$. Since $\kappa_l(L \cup C, i) \leq f$ in G_F, $\forall\ i \in R$, and $\kappa_l(R \cup C, j) \leq f$ in G_F,$\forall\ j \in L$, it follows that $|C_i| \leq f, \forall i \in R$, $|C_j| \leq f, \forall j \in L$. In addition, by construction, $|C_k| \leq f$, $\forall k \in C$. Thus the reduced graph defined by the given set F, and the chosen $C_i, \forall i \in R \cup L \cup C$ is a valid reduced graph of G^l. Denote the obtained reduced graph of G^l as $\widetilde{G^l}^*_F$. Since C_i is an l–restricted $(L \cup C, i)$–cut in G_F, there are no links from $L \cup C$ to node i in the reduced graph $\widetilde{G^l}^*_F$ for each $i \in R$. Thus by definition, R is a source component of $\widetilde{G^l}^*_F$. Similarly, we can show that L is also a source component of $\widetilde{G^l}^*_F$. Thus $\widetilde{G^l}^*_F$ contains at least two source components, leading to a contradiction.

The proof of Theorem 2 is complete. □

Some corollaries will be useful in the proof of sufficiency of Condition NC.

Corollary 2. *Suppose that graph G satisfies Condition NC. Then it follows that in each reduced graph $\widetilde{G^l}_F \in R_F$, there exists at least one node that has directed paths to all the nodes in $\widetilde{G^l}_F$.*

Corollary 2 follows immediately from Theorem 2.

Corollary 3. *Suppose that G satisfies Condition NC. Let $\phi \triangleq |\mathcal{F}|$. For any $\widetilde{G^l}_F \in R_F$ with $\mathbf{H}$ as the adjacency matrix, $\mathbf{H}^{n-\phi}$ has at least one non-zero column.*

Proof. By Corollary 2, in graph $\widetilde{G^l}_F$ there exists at least one node, say node k, that has a directed path in $\widetilde{G^l}_F$ to all the remaining nodes in $\mathcal{V}(G) - F$. Since the length of the path from k to any other node in $\widetilde{G^l}_F$ can contain

at most $n - \phi - 1$ directed edges, the k-th column of matrix $\mathbf{H}^{n-\phi}$ will be non-zero.[4] $\square$

2.4 Sufficiency: Algorithm TrimCov

In this section we propose an algorithm, named *Algorithm TrimCov*, and show its correctness. As can be seen later, our proposed update function works by first trimming away the received messages that contain extreme values, and then averaging the remaining message values. The extreme values are removed in order to guarantee validity condition. We first introduce our trimming strategy and show that it is well-defined.

Definition 6. *For a graph G, let $\mathcal{M}$ be a set of messages transmitted through G, and let $\mathcal{P}(\mathcal{M})$ be the set of message routes of all the messages in $\mathcal{M}$, i.e., $\mathcal{P}(\mathcal{M}) = \{\mathsf{path}(m) : m \in \mathcal{M}\}$. A message cover of $\mathcal{M}$ is a set of nodes $\mathcal{T}(\mathcal{M}) \subseteq \mathcal{V}(G)$ whose removal disconnects all messages routes, i.e., for each path $P \in \mathcal{P}(\mathcal{M})$, we have $\mathcal{V}(P) \cap \mathcal{T}(\mathcal{M}) \neq \emptyset$. In particular, a minimum message cover is defined by*

$$\mathcal{T}^*(\mathcal{M}) \in \arg \min_{\mathcal{T}(\mathcal{M}) \subseteq \mathcal{V}(G):\ \mathcal{T}(\mathcal{M})\ \textit{is a cover of}\ \mathcal{M}} |\mathcal{T}(\mathcal{M})|.$$

Conversely, given a set of messages $\mathcal{M}_0$ and a set of nodes $\mathcal{T} \subseteq \mathcal{V}(G)$, a maximal set of messages $\mathcal{M} \subseteq \mathcal{M}_0$ that are covered by $\mathcal{T}$ is defined by

$$\mathcal{M}^* \in \arg \max_{\mathcal{M} \subseteq \mathcal{M}_0:\ \mathcal{T}\ \textit{is a cover of}\ \mathcal{M}} |\mathcal{M}|.$$

Recall that $\mathcal{M}_i[t]$ is the collection of messages received by node i at iteration t. Let $\mathcal{M}'_i[t] = \mathcal{M}_i[t] - \{(v_i[t-1], (i,i))\}$. Sort messages in $\mathcal{M}'_i[t]$ in an increasing order, according to their message values, i.e., $\mathsf{value}(m)$ for $m \in \mathcal{M}'_i[t]$. Let $\mathcal{M}_{is}[t]$ be the largest sized subset of $\mathcal{M}'_i[t]$ such that (i) for all $m \in \mathcal{M}'_i[t] - \mathcal{M}_{is}[t]$ and $m' \in \mathcal{M}_{is}[t]$ we have $\mathsf{value}(m) \geq \mathsf{value}(m')$, and (ii) the cardinality of a minimum cover of $\mathcal{M}_{is}[t]$ is exactly f, i.e., $|\mathcal{T}^*(\mathcal{M}_{is}[t])| = f$. Similarly, we define $\mathcal{M}_{il}[t]$ to be the largest sized subset of $\mathcal{M}'_i[t]$ as follows: (i) for all $m \in \mathcal{M}'_i[t] - \mathcal{M}_{il}[t]$ and $m'' \in \mathcal{M}_{il}[t]$

[4] That is, all the entries of the column will be non-zero (more precisely, positive, since the entries of matrix $\mathbf{H}$ are non-negative). Also, such a non-zero column will exist in $\mathbf{H}^{n-\phi-1}$ too. We use the loose bound of $n - \phi$ to simplify the presentation.

we have $\mathsf{value}(m) \leq \mathsf{value}(m'')$, and (ii) the cardinality of a minimum cover of $\mathcal{M}_{il}[t]$ is exactly f, i.e., $|\mathcal{T}^*(\mathcal{M}_{il}[t])| = f$. In addition, define $\mathcal{M}_i^*[t] = \mathcal{M}_i'[t] - \mathcal{M}_{is}[t] - \mathcal{M}_{il}[t]$.

Intuitively speaking, from the perspective of node i, $\mathcal{M}_{is}[t]$ is the largest sized set of received messages that may be generated or tampered by faulty nodes, and contain extreme small values. Similarly, $\mathcal{M}_{il}[t]$ is the largest sized set of received messages that may be generated or tampered with by faulty nodes, and contain extreme large values.

Theorem 3. *Suppose that graph G satisfies Condition NC; then the sets of messages $\mathcal{M}_{is}[t]$, $\mathcal{M}_{il}[t]$ are well-defined and $\mathcal{M}_i^*[t]$ is nonempty for $f > 0$.*

Proof. For ease of exposition, we drop the time indices of $\mathcal{M}_i'[t]$, $\mathcal{M}_{is}[t]$, $\mathcal{M}_{il}[t]$ and $\mathcal{M}_i^*[t]$, respectively. From Corollary 1, we know $|N_i^- - \{i\}| \geq 2f + 1$. Since $|\mathcal{T}^*(\mathcal{M}_{is})| = f$ and $|\mathcal{T}^*(\mathcal{M}_{il})| = f$, the message from at least one incoming neighbor of node i is not covered by $\mathcal{T}^*(\mathcal{M}_{is}) \cup \mathcal{T}^*(\mathcal{M}_{il})$. So $\mathcal{M}_i^*$ is nonempty.

We prove the existence of $\mathcal{M}_{is}$ and $\mathcal{M}_{il}$ by construction. The set $\mathcal{M}_{is}$ can be constructed using the following algorithm, which can be easily adapted for the construction of set $\mathcal{M}_{il}$. For clarity of proof, we construct $\mathcal{M}_{is}$ and $\mathcal{M}_{il}$ sequentially, although they can be found in parallel.

Sort the messages in $\mathcal{M}_i'$ in an increasing order according to their messages values. Initialize $\mathcal{M}_{is} \leftarrow \emptyset$, $Q \leftarrow \emptyset$ and $\mathcal{M} \leftarrow \mathcal{M}_i'$. At each round, let m_s be a message with the smallest value in $\mathcal{M}$, and update Q, $\mathcal{M}$ as follows:

$$Q \leftarrow Q \cup \{m_s\}, \text{ and } \mathcal{M} \leftarrow \mathcal{M} - \{m_s\}.$$

If $|\mathcal{T}^*(Q)| \geq f + 1$, set $\mathcal{M}_{is} \leftarrow Q - m_s$ and return $\mathcal{M}_{is}$; otherwise, repeat this procedure.

If the algorithm terminates, by the code, it is easy to see that the returned $\mathcal{M}_{is}$ satisfies the following conditions: For all $m \in \mathcal{M}_i' - \mathcal{M}_{is}$ and $m' \in \mathcal{M}_{is}$ we have $\mathsf{value}(m) \geq \mathsf{value}(m')$; and the cardinality of a minimum cover of $\mathcal{M}_{is}$ is exactly f, i.e., $|\mathcal{T}^*(\mathcal{M}_{is})| = f$. It remains to show this algorithm terminates. Suppose this algorithm does not terminate. The problem of finding a minimum cover of a set of messages, i.e., computing $\mathcal{T}^*(Q)$, can be converted to the problem of finding a minimum cut of a vertex pair, by adding a new vertex y and connecting y to every vertex in $\mathcal{V}(G) - \{i\}$. The latter

problem can be solved in polynomial time. Thus, non-termination implies that $|\mathcal{T}^*(\mathcal{M}_i')| \leq f$, which further implies that the l–restricted $(\mathcal{V}(G) - \{i\}, i)$–connectivity is less than or equal to f. On the other hand, consider the node partition that $L = \{i\}$, $R = \mathcal{V}(G) - \{i\}$, and $C = F = \emptyset$, neither $L \cup C \Rightarrow_l R$ nor $R \cup C \Rightarrow_l L$ holds. This contradicts the assumption that G satisfies Condition NC. So the above algorithm terminates.

We can adapt the above procedure to construct $\mathcal{M}_{il}$ by modifying the initialization step to be $Q \leftarrow \emptyset$, $\mathcal{M} \leftarrow \mathcal{M}_i' - \mathcal{M}_{is}$. Termination can be shown similarly. Suppose this algorithm does not terminate. Non-termination implies that $|\mathcal{T}^*(\mathcal{M}_i' - \mathcal{M}_{is})| \leq f$, which further implies that in the node partition $L = \{i\}$, $F = \mathcal{T}^*(\mathcal{M}_{is})$, $R = \mathcal{V}(G) - F - L$, $C = \emptyset$, the l–restricted $(R \cup C, \{i\})$–connectivity is no more than f, i.e., $R \cup C \not\Rightarrow_l L$. In addition, since $|L| = 1$, $L \cup C \not\Rightarrow_l R$. This contradicts the assumption that G satisfies Condition NC. Therefore, $\mathcal{M}_{is}$ and $\mathcal{M}_{il}$ are well-defined.

□

From the generic code in Algorithm 1, we know that to design an IABC algorithm, it is enough to specify an update function for each fault-free node. For each $i \in \mathcal{V} - \mathcal{F}$, define

$$Z_i(\mathcal{M}_i[t]) \triangleq a_i v_i[t-1] + \sum_{m \in \mathcal{M}_i^*[t]} a_i \, w_m, \tag{2.3}$$

where $w_m = \mathsf{value}(m)$ and $a_i = \frac{1}{|\mathcal{M}_i^*[t]|+1}$. For future reference, we name the IABC algorithm with the update function (2.3) as **Algorithm TrimCov**.

Note that in (2.3), only messages in $\mathcal{M}_i^*[t]$ and the value $v_i[t-1]$ are used in updating v_i. Messages in both $\mathcal{M}_{is}[t]$ and $\mathcal{M}_{il}[t]$ are trimmed away. This trimming strategy is motivated by the observation that the messages in $\mathcal{M}_{is}[t]$ (or $\mathcal{M}_{il}[t]$) may be tampered with by nodes in $\mathcal{T}^*(\mathcal{M}_{is}[t])$ (or $\mathcal{T}^*(\mathcal{M}_{il}[t])$). These faulty behaviors are possible because of the fact that $|\mathcal{T}^*(\mathcal{M}_{is}[t])| = f$ and $|\mathcal{T}^*(\mathcal{M}_{il}[t])| = f$. Recall $\mathcal{M}_i^*[t] = \mathcal{M}_i'[t] - \mathcal{M}_{is}[t] - \mathcal{M}_{il}[t]$. The "weight" of each term on the right-hand side of (2.3) is a_i (which $a_i > 0$), and they add up to 1. For future reference, let us define α, which is used in Theorem 5, as:

$$\alpha = \min_{i \in \mathcal{V} - \mathcal{F}} a_i. \tag{2.4}$$

In *Algorithm TrimCov*, each fault-free node i's state, $v_i[t]$, is updated as a convex combination of all the *messages values* collected by node i at round t. In particular, for each message $m \in \mathcal{M}'[t]$, its coefficient is a_i if the message is in $\mathcal{M}_i^*[t]$ or the message is sent via self-loop of node i; otherwise, the coefficient of m is zero. The update step in *Algorithm TrimCov* is a generalization of the update steps proposed in [38, 45, 33, 34], where the update summation is over all the incoming neighbors of node i instead of over message routes. In [38, 45, 33, 34], only single-hop communication is allowed, i.e., $l = 1$, and the fault-free node i can receive only one message from its incoming neighbor. With multi-hop communication, a fault-free node can possibly receive messages from a node via multiple routes. Our trim function in *Algorithm TrimCov* takes the possible multi-route messages into account.

2.4.1 Correctness of *Algorithm TrimCov*

With our trim function, the iterative update of the state of a fault-free node i admits a nice matrix representation of states evolution of fault-free nodes. This representation allows us to prove the correctness of *Algorithm TrimCov*. We first briefly review some useful concepts and theorems.

Matrix Preliminaries

We use boldface upper case letters to denote matrices, rows of matrices, and their entries. For instance, $\mathbf{A}$ denotes a matrix, $\mathbf{A}_i$ denotes the i-th row of matrix $\mathbf{A}$, and $\mathbf{A}_{ij}$ denotes the element at the intersection of the i-th row and the j-th column of matrix $\mathbf{A}$.

Definition 7. *A vector is said to be* stochastic *if all the entries of the vector are* non-negative*, and the entries add up to 1. A matrix is said to be row stochastic if each row of the matrix is a stochastic vector.*

For a row stochastic matrix $\mathbf{A}$, coefficients of ergodicity $\delta(\mathbf{A})$ and $\lambda(\mathbf{A})$ are defined as [46]:

$$\delta(\mathbf{A}) = \max_j \max_{i_1, i_2} |\mathbf{A}_{i_1 j} - \mathbf{A}_{i_2 j}|, \text{ and } \lambda(\mathbf{A}) = 1 - \min_{i_1, i_2} \sum_j \min(\mathbf{A}_{i_1 j}, \mathbf{A}_{i_2 j}).$$

It is easy to see that $0 \leq \delta(\mathbf{A}) \leq 1$, $0 \leq \lambda(\mathbf{A}) \leq 1$, and that the rows are all identical if and only if $\delta(\mathbf{A}) = 0$. Additionally, $\lambda(\mathbf{A}) = 0$ if and only if $\delta(\mathbf{A}) = 0$.

The next result [47] establishes a relation between the coefficient of ergodicity $\delta(\cdot)$ of a product of row stochastic matrices, and the coefficients of ergodicity $\lambda(\cdot)$ of the individual matrices in the product.

Theorem 4. *[47] For any p square row stochastic matrices* $\mathbf{Q}(1), \mathbf{Q}(2), \ldots, \mathbf{Q}(p)$,

$$\delta(\mathbf{Q}(1)\mathbf{Q}(2)\cdots\mathbf{Q}(p)) \ \leq \ \Pi_{i=1}^{p} \ \lambda(\mathbf{Q}(i)).$$

Theorem 4 implies that if, for all i, $\lambda(\mathbf{Q}(i)) \leq 1 - \gamma$ for some $\gamma > 0$, then $\delta(\mathbf{Q}(1)\mathbf{Q}(2)\cdots\mathbf{Q}(p))$ will approach zero as p approaches ∞.

Definition 8. *[47, 46] A row stochastic matrix* $\mathbf{H}$ *is said to be a* scrambling *matrix, if* $\lambda(\mathbf{H}) < 1$.

In a scrambling matrix $\mathbf{H}$, since $\lambda(\mathbf{H}) < 1$, for each pair of rows i_1 and i_2, there exists a column j (which may depend on i_1 and i_2) such that $\mathbf{H}_{i_1 j} > 0$ and $\mathbf{H}_{i_2 j} > 0$, and vice-versa [47, 46]. As a special case, if any one column of a row stochastic matrix $\mathbf{H}$ contains only non-zero entries that are lower bounded by some constant $\gamma > 0$, then $\mathbf{H}$ must be scrambling, and $\lambda(\mathbf{H}) \leq 1 - \gamma$.

Definition 9. *For matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *of identical size, and a scalar* γ, $\mathbf{A} \leq \gamma\,\mathbf{B}$ *provided that* $\mathbf{A}_{ij} \leq \gamma\,\mathbf{B}_{ij}$ *for all* i, j.

Definition 10. *The adjacency matrix of graph G, denoted by* $\mathbf{A}$*, is a matrix with rows and columns labeled by graph vertices, and* $\mathbf{A}_{ij} = 1$ *if* $(i, j) \in \mathcal{E}$*; and* $\mathbf{A}_{ij} = 0$ *otherwise.*

Matrix Representation of *Algorithm TrimCov*

Recall that $\mathcal{F}$ is the set of faulty nodes. Let $|\mathcal{F}| = \phi$. Without loss of generality, suppose that nodes 1 through $(n - \phi)$ are fault-free, and if $\phi > 0$, nodes $(n - \phi + 1)$ through n are faulty. Denote by $\mathbf{v}[0] \in \mathbb{R}^{n-\phi}$ the column vector consisting of the initial states of all the *fault-free* nodes. Denote by $\mathbf{v}[t]$, where $t \geq 1$, the column vector consisting of the states of all the *fault-free* nodes at the end of the t-th iteration, $t \geq 1$, where the i-th element of vector $\mathbf{v}[t]$ is state $v_i[t]$.

The next theorem is our main result. Theorem 5 states that there exists a matrix representation of the states evolution of all the fault-free nodes. In addition, as will be seen later, this matrix has nice structures that guarantee the convergence condition of approximate consensus is met by *Algorithm TrimCov*.

Theorem 5. *We can express the iterative update of the state of a fault-free node i ($1 \leq i \leq n-\phi$) performed in (2.3) using the matrix form in (2.5) below, where $\mathbf{M}_i[t]$ satisfies the four conditions listed below. In addition to t, the row vector $\mathbf{M}_i[t]$ may depend on the state vector $\mathbf{v}[t-1]$ as well as the behavior of the faulty nodes in $\mathcal{F}$. For simplicity, the notation $\mathbf{M}_i[t]$ does not explicitly represent this dependence.*

$$v_i[t] \;=\; \mathbf{M}_i[t]\mathbf{v}[t-1] \tag{2.5}$$

1. $\mathbf{M}_i[t]$ *is a* stochastic *row vector of size* $(n-\phi)$. *Thus,* $\mathbf{M}_{ij}[t] \geq 0$, *where* $1 \leq j \leq n-\phi$, *and* $\sum_{1\leq j\leq n-\phi} \mathbf{M}_{ij}[t] \;=\; 1$

2. $\mathbf{M}_{ii}[t] \geq a_i \geq \alpha$.

3. $\mathbf{M}_{ij}[t]$ *is non-zero only if there exists a message* $m \in \mathcal{M}_i[t]$ *such that* $\mathsf{source}(m) = j$ *and* $\mathsf{destination}(m) = i$.

4. *For any* $t \geq 1$, *there exists a reduced graph* $\widetilde{G^l}_{\mathcal{F}} \in R_{\mathcal{F}}$ *with adjacent matrix* $\mathbf{H}[t]$ *such that* $\beta\,\mathbf{H}[t] \;\leq\; \mathbf{M}[t]$, *where* $\beta = \frac{1}{16n^{2l}}$.

The proof uses a structure similar to the proof of Claim 2 in [38].

Proof. Recall that nodes 1 through $n-\phi$ are fault-free, and the remaining ϕ nodes ($\phi \leq f$) are faulty. Consider a fault-free node i performing the *update step* in *Algorithm TrimCov*. Recall that $\mathcal{M}_{is}[t]$ and $\mathcal{M}_{il}[t]$ messages are eliminated from $\mathcal{M}_i[t]$. Let $\mathcal{S}_{ig}[t] \subseteq \mathcal{M}_{is}[t]$ and $\mathcal{L}_{ig}[t] \subseteq \mathcal{M}_{il}[t]$, respectively, be the sets of removed messages that are not covered by faulty nodes. Let $\mathcal{P}_i^*[t]$ be the set of paths corresponding to all the messages in $\mathcal{M}_i^*[t]$. With a little abuse of notation, we also use $\mathcal{P}_i^*[t]$ to denote the union of the vertex sets of all paths in $\mathcal{P}_i^*[t]$. The actual meaning of $\mathcal{P}_i^*[t]$ should be clear from the context. *Untampered message representation* of the evolution of v_i and construction of $\mathbf{M}_i[t]$ differ somewhat depending on whether sets $\mathcal{L}_{ig}[t], \mathcal{S}_{ig}[t]$ and $\mathcal{P}_i^*[t] \cap \mathcal{F}$ are empty or not, where $\mathcal{P}_i^*[t] \cap \mathcal{F} = \emptyset$ means that no message

in $\mathcal{M}_i^*[t]$ has been tampered by faulty nodes and $\mathcal{P}_i^*[t] \cap \mathcal{F} \neq \emptyset$ means that there exists a message that is covered by faulty nodes. It is possible that $\mathcal{T}^*(\mathcal{M}_{is}[t]) = \mathcal{T}^*(\mathcal{M}_{il}[t]) = \mathcal{F}$, which means all messages in $\mathcal{M}_{is}[t]$ and $\mathcal{M}_{il}[t]$ are tampered with by faulty nodes, i.e., $\mathcal{S}_{ig}[t] = \emptyset$ and $\mathcal{L}_{ig}[t] = \emptyset$. We divide the possibilities into six cases:

1. Case I: $\mathcal{S}_{ig}[t] \neq \emptyset, \mathcal{L}_{ig}[t] \neq \emptyset$ and $\mathcal{P}_i^*[t] \cap \mathcal{F} \neq \emptyset$.
2. Case II: $\mathcal{S}_{ig}[t] \neq \emptyset, \mathcal{L}_{ig}[t] \neq \emptyset$ and $\mathcal{P}_i^*[t] \cap \mathcal{F} = \emptyset$.
3. Case III: exactly one of $\mathcal{S}_{ig}[t], \mathcal{L}_{ig}[t]$ is empty and $\mathcal{P}_i^*[t] \cap \mathcal{F} \neq \emptyset$.
4. Case IV: exactly one of $\mathcal{S}_{ig}[t], \mathcal{L}_{ig}[t]$ is empty and $\mathcal{P}_i^*[t] \cap \mathcal{F} = \emptyset$.
5. Case V: $\mathcal{S}_{ig}[t] = \emptyset, \mathcal{L}_{ig}[t] = \emptyset$ and $\mathcal{P}_i^*[t] \cap \mathcal{F} \neq \emptyset$.
6. Case VI: $\mathcal{S}_{ig}[t] = \emptyset, \mathcal{L}_{ig}[t] = \emptyset$ and $\mathcal{P}_i^*[t] \cap \mathcal{F} = \emptyset$.

We first describe the construction of $\mathbf{M}_i[t]$ in case I. Recall that $w_m = \mathsf{value}(m)$. Let $\bar{w}_{is}[t]$ and $\bar{w}_{il}[t]$ be defined as shown below.

$$\bar{w}_{is}[t] = \frac{\sum_{m \in \mathcal{S}_{ig}[t]} w_m}{|\mathcal{S}_{ig}[t]|} \quad \text{and} \quad \bar{w}_{il}[t] = \frac{\sum_{m \in \mathcal{L}_{ig}[t]} w_m}{|\mathcal{L}_{ig}[t]|}.$$

By the definitions of $\mathcal{S}_{ig}[t]$ and $\mathcal{L}_{ig}[t]$, $\bar{w}_{is} \leq w_{m'} \leq \bar{w}_{il}$, for each message $m' \in \mathcal{M}_i^*[t]$. Thus, for each message m', we can find convex coefficient $\gamma_{m'}$, where $0 \leq \gamma_{m'} \leq 1$, such that

$$w_{m'} = \gamma_{m'} \bar{w}_{is} + (1 - \gamma_{m'}) \bar{w}_{il} = \frac{\gamma_{m'}}{|\mathcal{S}_{ig}[t]|} \sum_{m \in \mathcal{S}_{ig}[t]} w_m + \frac{1 - \gamma_{m'}}{|\mathcal{L}_{ig}[t]|} \sum_{m \in \mathcal{L}_{ig}[t]} w_m. \tag{2.6}$$

Recall from (2.3) that $v_i[t] = a_i v_i[t-1] + \sum_{m \in \mathcal{M}_i^*[t]} a_i w_m$, where $a_i = \frac{1}{|\mathcal{M}_i^*[t]|+1}$. In case I, since $\mathcal{P}_i^*[t] \cap \mathcal{F} \neq \emptyset$, there exist messages in $\mathcal{M}_i^*[t]$ that are tampered with by faulty nodes. We replace these "bad messages" by "good messages"

in the evolution of v_i.

$$
\begin{aligned}
v_i[t] &= a_i v_i[t-1] + \sum_{m \in \mathcal{M}_i^*[t]:\ \mathcal{V}(\mathsf{path}(m)) \cap \mathcal{F} = \emptyset} a_i w_m + \sum_{m \in \mathcal{M}_i^*[t]:\ \mathcal{V}(\mathsf{path}(m)) \cap \mathcal{F} \neq \emptyset} a_i w_m \\
&= a_i v_i[t-1] + \sum_{m \in \mathcal{M}_i^*[t]:\ \mathcal{V}(\mathsf{path}(m)) \cap \mathcal{F} = \emptyset} a_i w_m \\
&\quad + \sum_{m \in \mathcal{M}_i^*[t]:\ \mathcal{V}(\mathsf{path}(m)) \cap \mathcal{F} \neq \emptyset} a_i \Big(\frac{\gamma_m}{|\mathcal{S}_{ig}[t]|} \sum_{m' \in \mathcal{S}_{ig}[t]} w_{m'} + \frac{1-\gamma_m}{|\mathcal{L}_{ig}[t]|} \sum_{m' \in \mathcal{L}_{ig}[t]} w_{m'} \Big) \quad \text{by (2.6)} \\
&= a_i v_i[t-1] + \sum_{m \in \mathcal{M}_i^*[t]:\ \mathcal{V}(\mathsf{path}(m)) \cap \mathcal{F} = \emptyset} a_i w_m \\
&\quad + \sum_{m' \in \mathcal{S}_{ig}[t]} \Big(\sum_{m \in \mathcal{M}_i^*[t]:\ \mathcal{V}(\mathsf{path}(m)) \cap \mathcal{F} \neq \emptyset} \frac{a_i \gamma_m}{|\mathcal{S}_{ig}[t]|} \Big) w_{m'} \\
&\quad + \sum_{m' \in \mathcal{L}_{ig}[t]} \Big(\sum_{m \in \mathcal{M}_i^*[t]:\ \mathcal{V}(\mathsf{path}(m)) \cap \mathcal{F} \neq \emptyset} \frac{a_i (1-\gamma_m)}{|\mathcal{L}_{ig}[t]|} \Big) w_{m'}.
\end{aligned}
$$

That is, $v_i[t]$ can be represented as a convex combination of values of untampered messages collected at iteration t, where $v_i[t-1] = \mathsf{value}(v_i[t-1], (i,i))$. For future reference, we refer to the above convex combination as *untampered message representation of* $v_i[t]$ in case I and the convex coefficient of each message in the untampered message representation as *message weight*.

Note that if m is an untampered message in $\mathcal{M}_i^*[t]$ or $m \in \mathcal{S}_{ig}[t] \cup \mathcal{L}_{ig}[t]$, then $w_m = v_j[t-1]$ holds, where node j is the source of message m, i.e., $\mathsf{source}(m) = j$. $v_i[t]$ can be further rewritten as follows, where $\mathbb{1}\{x\} = 1$ if x is true, and $\mathbb{1}\{x\} = 0$, otherwise.

$$
\begin{aligned}
v_i[t] = \sum_{j \in \mathcal{V} - \mathcal{F}} v_j[t-1] \Big(& a_i \mathbb{1}\{j = i\} + \sum_{m \in \mathcal{M}_i^*[t]:\ \mathcal{V}(\mathsf{path}(m)) \cap \mathcal{F} = \emptyset} a_i \mathbb{1}\{\mathsf{source}(m) = j\} \\
& + \sum_{m' \in \mathcal{S}_{ig}[t]} \Big(\sum_{m \in \mathcal{M}_i^*[t]:\ \mathcal{V}(\mathsf{path}(m)) \cap \mathcal{F} \neq \emptyset} \frac{a_i \gamma_m}{|\mathcal{S}_{ig}[t]|} \mathbb{1}\{\mathsf{source}(m') = j\} \Big) \\
& + \sum_{m' \in \mathcal{L}_{ig}[t]} \Big(\sum_{m \in \mathcal{M}_i^*[t]:\ \mathcal{V}(\mathsf{path}(m)) \cap \mathcal{F} \neq \emptyset} \frac{a_i (1-\gamma_m)}{|\mathcal{L}_{ig}[t]|} \mathbb{1}\{\mathsf{source}(m') = j\} \Big) \Big),
\end{aligned}
$$

Thus, for $i, j \in \mathcal{V} - \mathcal{F}$, define the entry $\mathbf{M}_{ij}[t]$ as follows:

$$\mathbf{M}_{ij}[t] = a_i \mathbb{1}\{j = i\} + \sum_{m \in \mathcal{M}_i^*[t]:\ \mathcal{V}(\mathsf{path}(m)) \cap \mathcal{F} = \emptyset} a_i \mathbb{1}\{\mathsf{source}(m) = j\}$$
$$+ \sum_{m' \in \mathcal{S}_{ig}[t]} \Big(\sum_{m \in \mathcal{M}_i^*[t]:\ \mathcal{V}(\mathsf{path}(m)) \cap \mathcal{F} \neq \emptyset} \frac{a_i \gamma_m}{|\mathcal{S}_{ig}[t]|} \mathbb{1}\{\mathsf{source}(m') = j\} \Big)$$
$$+ \sum_{m' \in \mathcal{L}_{ig}[t]} \Big(\sum_{m \in \mathcal{M}_i^*[t]:\ \mathcal{V}(\mathsf{path}(m)) \cap \mathcal{F} \neq \emptyset} \frac{a_i (1 - \gamma_m)}{|\mathcal{L}_{ig}[t]|} \mathbb{1}\{\mathsf{source}(m') = j\} \Big). \tag{2.7}$$

Condition 3 in Theorem 5 follows trivially from (4.33). By (4.33), we have $\mathbf{M}_{ii} \geq a_i \geq \alpha$, satisfying condition 2 in Theorem 5. Now we show that $\mathbf{M}_i[t]$ satisfies condition 1 in Theorem 5, i.e., $\mathbf{M}_i[t]$ is a stochastic vector. We get

$$\begin{aligned}
\sum_{j \in \mathcal{V} - \mathcal{F}} \mathbf{M}_{ij}[t] &= a_i \sum_{j \in \mathcal{V} - \mathcal{F}} \mathbb{1}\{i = j\} + \sum_{m \in \mathcal{M}_i^*[t]:\ \mathsf{path}(m) \cap \mathcal{F} = \emptyset} a_i \sum_{j \in \mathcal{V} - \mathcal{F}} \mathbb{1}\{\mathsf{source}(m) = j\} \\
&\quad + \sum_{m \in \mathcal{M}_i^*[t]:\ \mathsf{path}(m) \cap \mathcal{F} \neq \emptyset} \Big(\frac{a_i \gamma_m}{|\mathcal{S}_{ig}[t]|} \sum_{m' \in \mathcal{S}_{ig}[t]} \sum_{j \in \mathcal{V} - \mathcal{F}} \mathbb{1}\{\mathsf{source}(m') = j\} \Big) \\
&\quad + \sum_{m \in \mathcal{M}_i^*[t]:\ \mathsf{path}(m) \cap \mathcal{F} \neq \emptyset} \Big(\frac{a_i (1 - \gamma_m)}{|\mathcal{L}_{ig}[t]|} \sum_{m' \in \mathcal{L}_{ig}[t]} \sum_{j \in \mathcal{V} - \mathcal{F}} \mathbb{1}\{\mathsf{source}(m') = j\} \Big) \\
&= a_i + \sum_{m \in \mathcal{M}_i^*[t]:\ \mathsf{path}(m) \cap \mathcal{F} = \emptyset} a_i + \sum_{m \in \mathcal{M}_i^*[t]:\ \mathsf{path}(m) \cap \mathcal{F} \neq \emptyset} \frac{a_i \gamma_m}{|\mathcal{S}_{ig}[t]|} \sum_{m' \in \mathcal{S}_{ig}[t]} 1 \\
&\quad + \sum_{m \in \mathcal{M}_i^*[t]:\ \mathsf{path}(m) \cap \mathcal{F} \neq \emptyset} \frac{a_i (1 - \gamma_m)}{|\mathcal{L}_{ig}[t]|} \sum_{m' \in \mathcal{L}_{ig}[t]} 1 \\
&= a_i + \sum_{m \in \mathcal{M}_i^*[t]:\ \mathsf{path}(m) \cap \mathcal{F} = \emptyset} a_i + \sum_{m \in \mathcal{M}_i^*[t]:\ \mathsf{path}(m) \cap \mathcal{F} \neq \emptyset} a_i \\
&= a_i(|\mathcal{M}_i^*[t]| + 1) = 1.
\end{aligned}$$

In addition, by (4.33), we know that $\mathbf{M}_{ij}[t] \geq 0$. Thus $\mathbf{M}_i[t]$ is row stochastic.

In case II, since $\mathcal{P}_i^*[t] \cap \mathcal{F} = \emptyset$, all messages in $\mathcal{M}_i^*[t]$ are untampered with by faulty nodes. Let m_0 be an arbitrary message in $\mathcal{M}_i^*[t]$, with $\mathsf{source}(m_0) =$

j^*. We rewrite $v_i[t]$ as follows:

$$
\begin{aligned}
v_i[t] &= a_i v_i[t-1] + \sum_{m\in\mathcal{M}_i^*[t]} a_i w_m \quad \text{by (2.3)} \\
&= a_i v_i[t-1] + a_i w_{m_0} + \sum_{m\in\mathcal{M}_i^*[t]-\{m_0\}} a_i w_m \\
&= a_i v_i[t-1] + \frac{1}{2}a_i w_{m_0} + \frac{1}{2}a_i w_{m_0} + \sum_{m\in\mathcal{M}_i^*[t]-\{m_0\}} a_i w_m \\
&= a_i v_i[t-1] + \frac{1}{2}a_i w_{m_0} + \frac{1}{2}a_i\Big(\frac{\gamma_{m_0}}{|\mathcal{S}_{ig}[t]|}\sum_{m'\in\mathcal{S}_{ig}[t]} w_{m'} + \frac{1-\gamma_{m_0}}{|\mathcal{L}_{ig}[t]|}\sum_{m'\in\mathcal{L}_{ig}[t]} w_{m'}\Big) \quad \text{by (2.6)} \\
&\quad + \sum_{m\in\mathcal{M}_i^*[t]-\{m_0\}} a_i w_m \\
&= a_i v_i[t-1] + \frac{1}{2}a_i w_{m_0} + \sum_{m'\in\mathcal{S}_{ig}[t]} \frac{a_i\gamma_{m_0}}{2|\mathcal{S}_{ig}[t]|} w_{m'} + \sum_{m'\in\mathcal{L}_{ig}[t]} \frac{a_i(1-\gamma_{m_0})}{2|\mathcal{L}_{ig}[t]|} w_{m'} \\
&\quad + \sum_{m\in\mathcal{M}_i^*[t]-\{m_0\}} a_i w_m.
\end{aligned}
$$

We refer to the above convex combination as the *untampered message representation of* $v_i[t]$ in case II. And we refer to the convex coefficient of each message in the above representation as *weight assigned* to that message. Combining the coefficients of messages according to message sources, it is obtained that

$$
\begin{aligned}
v_i[t] = \sum_{j\in\mathcal{V}-\mathcal{F}} v_j[t-1]\Big(& a_i\mathbb{1}\{i=j\} + \frac{1}{2}a_i\mathbb{1}\{j=j^*\} \\
&+ \sum_{m\in\mathcal{M}_i^*[t]-\{m_0\}} a_i\mathbb{1}\{\mathsf{source}(m)=j\} + \frac{a_i\gamma_{m_0}}{2|\mathcal{S}_{ig}[t]|}\sum_{m'\in\mathcal{S}_{ig}[t]}\mathbb{1}\{\mathsf{source}(m')=j\} \\
&+ \frac{a_i(1-\gamma_{m_0})}{2|\mathcal{L}_{ig}[t]|}\sum_{m'\in\mathcal{L}_{ig}[t]}\mathbb{1}\{\mathsf{source}(m')=j\}\Big).
\end{aligned}
$$

Thus, for $i,j\in\mathcal{V}-\mathcal{F}$, define $\mathbf{M}_{ij}$ by

$$
\begin{aligned}
\mathbf{M}_{ij} = a_i\mathbb{1}\{i=j\} + \frac{1}{2}a_i\mathbb{1}\{j=j^*\} &+ \sum_{m\in\mathcal{M}_i^*[t]-\{m_0\}} a_i\mathbb{1}\{\mathsf{source}(m)=j\} \\
&+ \frac{a_i\gamma_{m_0}}{2|\mathcal{S}_{ig}[t]|}\sum_{m'\in\mathcal{S}_{ig}[t]}\mathbb{1}\{\mathsf{source}(m')=j\} + \frac{a_i(1-\gamma_{m_0})}{2|\mathcal{L}_{ig}[t]|}\sum_{m'\in\mathcal{L}_{ig}[t]}\mathbb{1}\{\mathsf{source}(m')=j\}.
\end{aligned}
$$

Following the same line as in the proof of case I, it can be shown that the above $\mathbf{M}_{ij}$ satisfies conditions 1, 2 and 3 in Theorem 5.

In case III, case IV, case V and case VI, at least one of $\mathcal{S}_{ig}[t]$ and $\mathcal{L}_{ig}[t]$ is empty; without loss of generality, assume that $\mathcal{S}_{ig}[t]$ is empty. By the definition of $\mathcal{S}_{ig}[t]$, we know that the set $\mathcal{M}_{is}[t]$ is covered by $\mathcal{F}$. On the other hand, by the definition of $\mathcal{M}_{is}[t]$, a minimum cover of $\mathcal{M}_{is}[t]$ is of size f. Since $|\mathcal{F}| \leq f$, then we know $\mathcal{F}$ is a minimum cover of $\mathcal{M}_{is}[t]$ and $|\mathcal{F}| = f$. From the definition of $\mathcal{M}_{is}[t]$, we know there exists a message with the smallest value in $\mathcal{M}_i^*[t]$, denoted by m_s, that is not covered by $\mathcal{F}$. So, we can use singleton $\{m_s\}$ to mimic the role of $\mathcal{S}_{ig}[t]$ in cases I and II. Similarly, we can use the same trick when $\mathcal{L}_{ig}[t]$ is empty. The *untampered message representation of* $v_i[t]$ and *message weight* are defined similarly as that in case I and case II.

To show the above constructions satisfy the last condition in Theorem 5, we need the following two claims.

Claim 1 *For node $i \in \mathcal{V} - \mathcal{F}$, in the untampered message representation of $v_i[t]$, at most one of the sets $\mathcal{S}_{ig}[t]$ and $\mathcal{L}_{ig}[t]$ contains messages with assigned weights less than β, where $\beta = \frac{1}{16n^{2l}}$.*

Now we prove *Claim 1.* An untampered message is either in $\mathcal{M}_i^*[t]$ or in $\mathcal{S}_{ig}[t] \cup \mathcal{L}_{ig}[t]$.

For **case V** and **case VI**, both $\mathcal{S}_{ig}[t]$ and $\mathcal{L}_{ig}[t]$ are empty, all untampered messages are contained in $\mathcal{M}_i^*[t]$. For each untampered message in $\mathcal{M}_i^*[t]$, its weight in the untampered message representation is $a_i \geq \frac{1}{|\mathcal{M}_i^*[t]|+1}$. In $\mathcal{M}_i[t]$, at most n messages were transmitted via one hop, at most n^2 messages were transmitted via two hops. In general, $\mathcal{M}_i[t]$ contains at most n^d messages that were transmitted via d hops, where d is an integer in $\{1, \ldots, l\}$. Thus,

$$|\mathcal{M}_i^*[t]| + 1 \leq |\mathcal{M}_i[t]| \ \leq n + n^2 + \ldots + n^l = \frac{n(n^l - 1)}{n - 1} \overset{(a)}{\leq} \frac{n(n^l - 1)}{\frac{n}{2}} \leq 2n^l.$$

Inequality (a) is true because $n \geq 2$. Thus, $a_i \geq \frac{1}{2n^l}$. In cases V and VI, as both $\mathcal{S}_{ig}[t]$ and $\mathcal{L}_{ig}[t]$ are empty, all untampered messages have weight no less than $\frac{1}{2n^l}$.

For **case III** and **case IV**, without loss of generality, assume $\mathcal{S}_{ig}[t]$ is empty. An untampered message is either in $\mathcal{M}_i^*[t]$ or in $\mathcal{L}_{ig}[t]$. For each

untampered message in $\mathcal{M}_i^*[t]$, the weight assigned to it in the untampered message representation of $v_i[t]$ is at least $\frac{1}{2n^l}$. Thus, only $\mathcal{L}_{ig}[t]$ may contain untampered messages with assigned weights less than $\frac{1}{2n^l}$.

For **case II**, both $\mathcal{S}_{ig}[t]$ and $\mathcal{L}_{ig}[t]$ are nonempty, an untampered message is in one of $\mathcal{M}_i^*[t]$, $\mathcal{S}_{ig}[t]$ and $\mathcal{L}_{ig}[t]$. In the untampered message representation of $v_i[t]$, either $\gamma_{m_0} \geq \frac{1}{2}$ or $1-\gamma_{m_0} \geq \frac{1}{2}$. Without loss of generality, assume that $\gamma_{m_0} \geq \frac{1}{2}$, which implies that for each message in $\mathcal{S}_{ig}[t]$, the assigned weight is at least $\frac{a_i}{4|\mathcal{S}_{ig}[t]|} \geq \frac{1}{16n^{2l}}$, since $|\mathcal{S}_{ig}[t]| \leq |\mathcal{M}_i[t]| \leq 2n^l$. Letting $\beta = \frac{1}{16n^{2l}}$, then we can conclude that only $\mathcal{L}_{ig}[t]$ may contain untampered messages with assigned weights less than β–note that the $\geq \beta$ weight is assigned to messages instead of nodes.

It can be shown similarly that the above claim also holds for **case I**.

The proof of *Claim 1* is complete.

Now we are ready to show the following property is also true.

Claim 2 *For any $t \geq 1$, there exists a reduced graph $\widetilde{G^l}_{\mathcal{F}} \in R_{\mathcal{F}}$ such that $\beta\,\mathbf{H}[t] \;\leq\; \mathbf{M}[t]$.*

Now we prove *Claim 2*.

We construct the desired reduced graph $\widetilde{G^l}_{\mathcal{F}}$ as follows. Let

$$E = \{e \in \mathcal{E}(G^l) :\; \mathcal{V}(P(e)) \cap \mathcal{F} \neq \emptyset\}$$

be the set of edges in G^l that are covered by node set $\mathcal{F}$.

For a fault-free node i: (i) if both $\mathcal{S}_{ig}[t]$ and $\mathcal{L}_{ig}[t]$ are empty, then choose $C_i = \emptyset$; (ii) if one of $\mathcal{S}_{ig}[t]$ and $\mathcal{L}_{ig}[t]$ is empty, without loss of generality, assume that $\mathcal{S}_{ig}[t]$ is empty, then choose $C_i = \mathcal{T}^*(\mathcal{M}_{il}[t])$; (iii) if both $\mathcal{S}_{ig}[t]$ and $\mathcal{L}_{ig}[t]$ are nonempty, without loss of generality, assume that the weight assigned to every message in $\mathcal{S}_{ig}[t]$ is lower bounded by β, then choose $C_i = \mathcal{T}^*(\mathcal{M}_{il}[t])$. Let

$$E_i = \{e \in \mathcal{E}(G^l) :\; e \text{ is an incoming edge of node } i \text{ in } G^l \text{ and } \mathcal{V}(P(e)) \cap C_i \neq \emptyset\}$$

be the set of incoming edges of node i in G^l that are covered by node set C_i. Set $\mathcal{V}(\widetilde{G^l}_{\mathcal{F}}) = \mathcal{V}(G) - \mathcal{F}$. And let $\mathcal{E}(\widetilde{G^l}_{\mathcal{F}}) = \mathcal{E}(\widetilde{G^l}) - E - \cup_{i \in \mathcal{V}-\mathcal{F}} E_i$.

From *Claim 1*, for node i, at most one of the sets $\mathcal{S}_{ig}[t]$ and $\mathcal{L}_{ig}[t]$ contains

messages with assigned weights less than β. Then it is easy to see that $\mathbf{H}[t]$, the adjacency matrix of the obtained reduced graph $\widetilde{G^l}_{\mathcal{F}}$, has the property that $\beta\,\mathbf{H}[t] \;\leq\; \mathbf{M}[t]$.

The proof of *Claim 2* is complete.

Note that Claim 2 says that for each $t \geq 1$, the constructed matrix $\mathbf{M}[t]$ satisfies condition 4 in Theorem 5.

Therefore, the proof of Theorem 5 is complete.

□

Correctness of *Algorithm TrimCov*

With the matrix representation in Theorem 5, we are ready to show the correctness of *Algorithm TrimCov*.

By "stacking" (2.5) for different i, $1 \leq i \leq n-\phi$, we can represent the state update for all the fault-free nodes together using (2.8) below, where $\mathbf{M}[t]$ is a $(n-\phi) \times (n-\phi)$ row stochastic matrix, with its i-th row being equal to $\mathbf{M}_i[t]$ in (2.5).

$$\mathbf{v}[t] \;\;=\;\; \mathbf{M}[t]\;\mathbf{v}[t-1]. \tag{2.8}$$

By repeated application of (2.8), we obtain:

$$\mathbf{v}[t] \;\;=\;\; \left(\,\Pi_{\tau=1}^{t}\mathbf{M}[\tau]\,\right)\;\mathbf{v}[0].$$

As the backward product $\Pi_{\tau=1}^{t}\mathbf{M}[\tau]$ is a row-stochastic matrix, it holds that $\mu[0] \leq v_i[t] \leq U[0]$ for all $i = 1, \ldots, n-\phi$ and all t. Thus *Algorithm TrimCov* satisfies validity condition.

The convergence of $v_i[t]$ depends on the convergence of the backward product $\Pi_{\tau=1}^{t}\mathbf{M}[\tau]$. As a result of this, our convergence proof uses toolkit of weak-ergodic theory that is also adopted in prior work (e.g., [41, 37, 34, 33]). Recall from Theorem 5 that for any $t \geq 1$, there exists a reduced graph $\widetilde{G^l}_{\mathcal{F}} \in R_{\mathcal{F}}$ with adjacent matrix $\mathbf{H}[t]$ such that $\beta\,\mathbf{H}[t] \;\leq\; \mathbf{M}[t]$, where $\beta = \frac{1}{16n^{2l}}$.

Lemma 1. *In the product of* $\mathbf{H}[t]$ *matrices for consecutive* $\tau(n-\phi)$ *iterations, i.e.,* $\Pi_{t=z}^{z+\tau(n-\phi)-1}\,\mathbf{H}[t]$*, at least one column is non-zero.*

Proof. Since the above product consists of $\tau(n-\phi)$ matrices in R_F, at least one of the τ distinct connectivity matrices in $R_{\mathcal{F}}$, say matrix $\mathbf{H}_*$, will appear in the above product at least $n-\phi$ times.

Now observe that: (i) By Lemma 3, $\mathbf{H}_*^{n-\phi}$ contains a non-zero column, say the k-th column is non-zero, and (ii) all the $\mathbf{H}[t]$ matrices in the product contain a non-zero diagonal. These two observations together imply that the k-th column in the above product is non-zero. □

Let us now define a sequence of matrices $\mathbf{Q}(i)$ such that each of these matrices is a product of $\tau(n-\phi)$ of the $\mathbf{M}[t]$ matrices. Specifically,

$$\mathbf{Q}(i) \;=\; \Pi_{t=(i-1)\tau(n-\phi)+1}^{i\tau(n-\phi)}\, \mathbf{M}[t].$$

Observe that

$$\mathbf{v}[k\tau(n-\phi)] \;=\; \left(\Pi_{i=1}^{k}\, \mathbf{Q}(i)\right)\, \mathbf{v}[0]. \tag{2.9}$$

Lemma 2. *For $i \geq 1$, $\mathbf{Q}(i)$ is a scrambling row stochastic matrix, and*

$$\lambda(\mathbf{Q}(i)) \leq 1-\beta^{\tau(n-\phi)}.$$

Proof. Since $\mathbf{Q}(i)$ is a product of row stochastic matrices $\mathbf{M}[t]$, thus, $\mathbf{Q}(i)$ is row stochastic.

From Theorem 5, for each t, $\beta\,\mathbf{H}[t] \;\leq\; \mathbf{M}[t]$. So,

$$\beta^{\tau(n-\phi)}\; \Pi_{t=(i-1)\tau(n-\phi)+1}^{i\tau(n-\phi)}\, \mathbf{H}[t] \;\leq\; \mathbf{Q}(i).$$

By using $z=(i-1)(n-\phi)+1$ in Lemma 1, we conclude that the matrix product on the left side of the above inequality contains a non-zero column. Thus, there exists a non-zero column in $\mathbf{Q}(i)$ with each entry being $\geq \beta^{\tau(n-\phi)}$.

Therefore, $\mathbf{Q}(i)$ is a scrambling matrix, and $\lambda(\mathbf{Q}(i)) \leq 1-\beta^{\tau(n-\phi)}$. □

Theorem 6. Algorithm TrimCov *satisfies the validity and the convergence conditions.*

Proof. Since $\mathbf{v}[t] \;=\; \mathbf{M}[t]\, v[t-1]$, and $\mathbf{M}[t]$ is a row stochastic matrix, it follows that *Algorithm TrimCov* satisfies the validity condition.

By Theorem 4,

$$\begin{aligned}
\lim_{t\to\infty} \delta(\Pi_{i=1}^{t}\mathbf{M}[t]) &= \lim_{t\to\infty} \delta\left(\Pi_{i=1}^{\lfloor \frac{t}{\tau(n-\phi)}\rfloor}\mathbf{Q}(i)\Pi_{j=\lfloor \frac{t}{\tau(n-\phi)}\rfloor+1}^{t}\mathbf{M}[j]\right)\\
&\leq \lim_{t\to\infty} \left(\Pi_{i=1}^{\lfloor \frac{t}{\tau(n-\phi)}\rfloor}\lambda(\mathbf{Q}(i))\right)\lambda\left(\Pi_{j=\lfloor \frac{t}{\tau(n-\phi)}\rfloor+1}^{t}\mathbf{M}[j]\right)\\
&\leq \lim_{t\to\infty} \left(\Pi_{i=1}^{\lfloor \frac{t}{\tau(n-\phi)}\rfloor}\lambda(\mathbf{Q}(i))\right)\\
&\leq \lim_{t\to\infty} \left(1-\beta^{\tau(n-\phi)}\right)^{\lfloor \frac{t}{\tau(n-\phi)}\rfloor}\\
&= 0.
\end{aligned}$$

The above argument makes use of the facts that $\lambda(\mathbf{M}[t]) \leq 1$ and $\lambda(\mathbf{Q}(i)) \leq (1-\beta^{\tau(n-\phi)}) < 1$. Thus, the rows of $\Pi_{i=1}^{t}\mathbf{M}[t]$ become identical in the limit. This observation and the fact that $\mathbf{v}[t] = (\Pi_{i=1}^{t}\mathbf{M}[i])\mathbf{v}[t-1]$ together imply that the state of the fault-free nodes satisfies the convergence condition.

Now, the validity and convergence conditions together imply that there exists a positive scalar c such that

$$\lim_{t\to\infty} \mathbf{v}[t] \;=\; \lim_{t\to\infty} \left(\Pi_{i=1}^{t}\mathbf{M}[i]\right)\; \mathbf{v}[0] \;=\; c\,\mathbf{1},$$

where $\mathbf{1}$ denotes a column with all its entries being 1. □

2.5 Connection with Existing Work

In this section, we show that Condition NC is equivalent to the existing results on both undirected graphs and directed graphs.

2.5.1 Undirected graph with unbounded path length

If G is undirected, it has been shown in [35] that $n \geq 3f+1$ and node-connectivity $2f+1$ are both necessary and sufficient for achieving Byzantine approximate consensus. Recall that l^* is the length of a longest cycle-free path in G. We will show that when $l \geq l^*$, our Condition NC is equivalent to the above conditions, formally stated below.

Theorem 7. *When $l \geq l^*$, if G is undirected, then $n \geq 3f+1$ and node-connectivity of G is at least $2f+1$ if and only if G satisfies Condition NC.*

Proof. First we show "Condition NC implies $n \geq 3f+1$ and node connectivity at least $2f+1$".

When $f = 0$, it holds that $3f + 1 = 1$. In addition, we know $n \geq 2$. Thus, we get $n \geq 2 \geq 1 = 3f + 1$.

For $f > 0$, it has already been shown in Corollary 1 that $n \geq 3f + 1$. It remains to show the node connectivity of G is at least $2f + 1$. We prove this by contradiction. Suppose the node-connectivity is no more than $2f$. Let S be a min cut of G, then $|S| \leq 2f$. Let K_1 and K_2 be two disjoint connected components in G_S, the subgraph of G induced by node set $\mathcal{V}(G) - S$.

Construct a node partition of G as follows: Let $L = K_1, R = K_2$ and $C = \mathcal{V} - F - L - R$, where (1) if $|S| \geq f + 1$, let $F \subseteq S$ such that $|F| = f$; (2) otherwise, let $F = S$. For the latter case, $C = \emptyset$ and since $F = S$ is a cut of G disconnecting R from other nodes in G_F, then there is no path between $L \cup C$ and R in G_F, i.e., $\kappa(L \cup C, i) = 0 \leq f$ for each $i \in R$ in G_F. Similarly, $\kappa(R \cup C, j) = 0 \leq f$ for each $j \in L$. On the other hand, we know that G satisfies Condition NC. Thus, we arrive at a contradiction.

For the former case, i.e., $F \subset S$, since G satisfies Condition NC, without loss of generality, assume $R \cup C \Rightarrow_{l^*} L$ in G_F, i.e., there exists a node $i \in L$ such that there are at least $f + 1$ disjoint paths from set $R \cup C$ to node i in G_F. Add an additional node y and connect node y to all nodes in $R \cup C$. Denote the resulting graph by G'_F. From Menger's theorem we know that a minimum y, i-cut in graph G'_F has size at least $f + 1$. On the other hand, since S is a cut of G, we know that $S - F$ is a y, i–cut in G'_F. In addition, we know $|S - F| = |S| - |F| \leq 2f - f \leq f$. Thus we arrive at a contradiction.

Next we show that "$n \geq 3f + 1$ and $2f + 1$ node-connectivity imply Condition NC". Consider an arbitrary node partition L, R, C, F of G such that $L \neq \emptyset, R \neq \emptyset$ and $|F| \leq f$. Since $n \geq 3f + 1$ and $|F| \leq f$, either $|L \cup C| \geq f + 1$ or $|R \cup C| \geq f + 1$. Without loss of generality, assume that $|R \cup C| \geq f + 1$. Add a node y connecting to all nodes in $R \cup C \cup F$ and denote the newly obtained graph by G''. Since $|F| + f + 1 \leq 2f + 1$, by Expansion Lemma[5] [43], G'' is $|F| + f + 1$ connected. Fix $i \in L$. There

[5]**Expansion Lemma** If G is a k-connected graph, and G' is formed from G by adding a vertex y having at least k neighbors in G, then G' is k-connected.

are at least $|F| + f + 1$ internally disjoint y, i–paths. So there are at least $f + 1$ internally disjoint y, i–paths in G''_F. Thus $R \cup C \Rightarrow_{l^*} L$ in G_F. Since this holds for all partitions of the form L, R, C, F where $L \neq \emptyset$, $R \neq \emptyset$ and $|F| \leq f$, then we conclude that Condition NC holds. This completes the proof.

□

2.5.2 Directed graph with unbounded path length

Synchronous exact Byzantine consensus is considered in [36].

Definition 11. *[36] Given disjoint subsets A, B, where B is non-empty:*
(i) We say $A \rightarrow B$ if and only if set A contains at least $f+1$ distinct incoming neighbors of B. That is, $|\{i|\ (i,j) \in \mathcal{E}, i \in A, j \in B\}| > f$.
(ii) We say $A \not\rightarrow B$ iff $A \rightarrow B$ is not true.

A tight condition (both necessary and sufficient) over the graph structure is found in [36].

Theorem 8. *[36] Given a graph G, exact Byzantine consensus is solvable if and only if for any partition L, C, R, F of G, such that both L and R are non-empty, and $|F| \leq f$, either $L \cup C \rightarrow R$, or $R \cup C \rightarrow L$.*

We term this condition as Condition 1. Note that in order for $A \rightarrow B$ to hold, we only require that there are at least $f + 1$ incoming neighbors of set B to be in set A. As a result of this observation, our Condition NC with $l = 1$ is, in general, strictly stronger than Condition 1. However, it can be shown that our Condition NC with $l \geq l^*$ is equivalent to Condition 1. We first state an alternative version of Condition 1.

Definition 12. *[36] Given disjoint subsets A, B, F of G such that $|F| \leq f$, set A is said to propagate in G_F to set B if either (i) $B = \emptyset$, or (ii) for each node $b \in B$, there exist at least $f + 1$ disjoint (A, b)–paths in G_F.*

We will denote the fact that set A propagates in G_F to set B by the notation $A \overset{\mathcal{V}-F}{\rightsquigarrow} B$. When it is not true that $A \overset{\mathcal{V}-F}{\rightsquigarrow} B$, we will denote that fact by $A \overset{\mathcal{V}-F}{\not\rightsquigarrow} B$.

Theorem 9. *[36] Given graph G, Condition 1 holds if and only if for any node partition A, B, F of G, where A and B are both non-empty, and $|F| \leq f$, either $A \overset{\mathcal{V}-F}{\rightsquigarrow} B$ or $B \overset{\mathcal{V}-F}{\rightsquigarrow} A$ holds in G_F.*

For ease of future reference, we term the second condition in the above theorem as Condition Propagate [36]. Now we are ready to show the equivalence between Condition NC and Condition 1.

Theorem 10. *Condition NC is equivalent to Condition 1 when $l \geq l^*$.*

Proof. We will show that Condition NC implies Condition 1, and Condition Propagate implies Condition NC. By Theorem 9, Condition 1 and Condition Propagate are equivalent. Then we can conclude that Condition 1 and Condition NC are equivalent.

We first show that Condition NC implies Condition 1. Let $l \geq l^*$. For any node partition L, C, R, F of G such that $L \neq \emptyset, R \neq \emptyset$ and $|F| \leq f$, in the subgraph G_F, at least one of the two conditions below must be true: (i) $R \cup C \Rightarrow_l L$ in G_F; (ii) $L \cup C \Rightarrow_l R$ in G_F. Let $i \in L$. Without loss of generality, assume that $R \cup C \Rightarrow_l L$ in G_F and that $\kappa(R \cup C, i) \geq f + 1$, i.e., node i has at least $f + 1$ disjoint paths from $R \cup C$. For each such path, there exist at least one edge that goes from $R \cup C$ to a node in L. Since all the paths considered are disjoint, $R \cup C$ contains at least $f + 1$ incoming neighbors of L.

We next show that Condition Propagate implies Condition NC. We prove this by contradiction. Suppose, on the contrary, that Condition NC does not hold. There exists a partition L, C, R, F of G such that $L \neq \emptyset, R \neq \emptyset$ and $|F| \leq f$, in the induced subgraph G_F, (i) $R \cup C \not\Rightarrow_l L$; (ii) $L \cup C \not\Rightarrow_l R$. For each node i in L, there are at most f disjoint $(R \cup C, i)$ paths excluding F. Thus $R \cup C \overset{\mathcal{V}-\mathcal{F}}{\not\rightarrow} L$.

On the other hand, as $L \cup C \not\Rightarrow_l R$, for each node $j \in R$, there are at most f disjoint paths from $L \cup C$ to j excluding F, which further implies that there are at most f disjoint paths from L to j excluding F. Thus, $L \overset{\mathcal{V}-\mathcal{F}}{\not\rightarrow} R \cup C$. This contradicts the assumption that Condition Propagate holds. Thus we conclude that Condition Propagate implies Condition NC.

Besides, Condition Propagate is equivalent to Condition 1. Therefore, Condition NC, Condition Propagate, and Condition 1 are all equivalent. □

2.6 Summary and Discussion

In this chapter, we assume that each node knows the topology within its l–hop neighborhood, and in each iteration it can send messages to nodes that are up to l hops away, where $l \geq 1$. We prove a necessary and sufficient condition for the existence of *iterative* algorithms that achieve *approximate Byzantine consensus* in directed graphs, while maintaining minimal memory across iterations.

Throughout the presentation so far, we assumed that faulty nodes are only able to tamper with message values, leaving message paths unchanged. However, this restriction of faulty behaviors of Byzantine nodes is not necessary. In fact, the above results still hold when both message value tampering and message path tampering are allowed. Next, we sketch a proof that is also briefly discussed in [48]. Indeed, we will show that tampering with both message values and message paths is equivalent to tampering with message values only. In other words, for any faulty behavior of the faulty nodes under the more general fault model, there is an equivalent faulty behavior of the faulty nodes when only message value can be tampered with. Thus, our proposed *Algorithm TrimCov* also works under the more general fault model.

In iteration t, if multiple messages arrive at node i along the path P, then this multiplicity is caused either by message values tampering or by message paths tampering. In both cases, at least one node in path P is faulty. The former case can be seen easily. To see the message paths tampering case, suppose a fault-free node k receives or relays a message $m = (w, P)$ from node j containing a path that does not have the form $\ldots jk \ldots$. Then node k knows that node j is faulty, and will discard the message. This way, on any given path P, at least the very last faulty node will have to remain on the path (it may delete the earlier nodes on the path, but not itself). Since at least one node in path P is faulty, from the perspective of node i, the message path tampering faulty behavior is equivalent to having a faulty node in P send additional value tampered messages directly.

Similarly, in iteration t, if node i does not see any message along path P, then either a faulty node does not send/forward the message $m = (w, P)$, or it resets the message route P to be P' such that $P' \neq P$. From the perspective of node i, the message path tampering faulty behavior is equivalent to having a faulty node in P not to send/forward the message with route P.

The above two scenarios together prove that tampering with both message values and message paths is equivalent to tampering with message values only.

Throughout this chapter, we have focused on approximate Byzantine consensus, where fault-free nodes asymptotically agree with each other. We found that the tight topological condition depends on parameter l. Whether parameter l has an effect on achieving exact consensus or not is still open, and is left as further work.

CHAPTER 3

CONSENSUS-BASED MULTI-AGENT OPTIMIZATION

3.1 Introduction

In this chapter, we are interested in an optimization problem over a multi-agent network, where each agent keeps a *local* cost function that is initially known only to itself, and the networked agents want to collectively reach agreement on a global decision x such that a global objective that properly aggregates these local costs is minimized. The focus of this chapter is on the *fault-tolerant* multi-agent optimization problem, where an unknown subset of agents may be compromised by a system adversary, and be reprogrammed to behave arbitrarily under the control of the adversary.

While the failure-free version of the above problem is well-understood, we explore the case where some unknown subset of the computing agents may be adversarial. Specifically, we assume that up to f agents among the total n agents suffer Byzantine faults [49]. An agent suffering Byzantine fault may not follow the pre-specified algorithms/protocols, and *misbehave arbitrarily*. With the global objective function defined in (1.1), the global decision x identified by the non-faulty agents can be significantly biased, and may even be completely controlled by the faulty agents. Thus, a proper global objective **should not** directly aggregate the local functions kept by the faulty agents. If we denote by $\mathcal{N}$ the set of non-faulty agents *in a given execution*, then, ideally, we would like all the *non-faulty* agents to collaboratively minimize

$$\frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} h_i, \tag{3.1}$$

i.e., the average of the local cost functions associated with non-faulty agents only. The global objective in (3.1) can be viewed as a *weighted* average $\sum_{i \in \mathcal{N}} \alpha_i h_i$ with weight α_i equal to $\frac{1}{|\mathcal{N}|}$ for all $i \in \mathcal{N}$. Unfortunately, since

the non-faulty agents do not necessarily know the identity of Byzantine faulty agents, and may not be able to identify the faulty agents, the goal in (3.1) is unachievable (as proved in Theorem 11 in Section 3.4).

Observing this, we define a relaxed version of the multi-agent optimization problem. In particular, the goal of the relaxed problem is to design algorithms that enable all the non-faulty agents in the network collaboratively to reach agreement on a global decision x for which there **exists** weight vector $\boldsymbol{\alpha} \in \mathbb{R}^n$ such that the global objective

$$\sum_{i \in \mathcal{N}} \alpha_i h_i \tag{3.2}$$

is minimized, where $\sum_{i \in \mathcal{N}} \alpha_i = 1$, $\alpha_i \geq 0$ for all $i = 1, \cdots, n$, and $\alpha_i = 0$ for each $i \notin \mathcal{N}$. We note that the problem (3.2) does **not** require the non-faulty agents to learn the actual weights (α_i's) corresponding to the global cost function that was optimized.

We say that problem (3.1) (or (3.2)) is solvable if there exists an algorithm whose output minimizes the objective in (3.1) (or (3.2)) for all admissible local cost functions, and all possible behaviors of faulty nodes.

Since the (qualitative) goal of fault-tolerant multi-agent optimization is to decide on an output that takes into account the local cost functions of all the non-faulty agents, it is desired that the weights (α_i's) above be non-zero for the largest possible number of non-faulty agents, and, preferably, these non-zero weights be as close to $\frac{1}{|\mathcal{N}|}$ as possible. This would ensure that the global objective that is optimized has approximately equal representation for each agent's cost function.

We define β, γ to characterize the "goodness" of a weight vector $\boldsymbol{\alpha}$.

Definition 13. (β, γ)–**admissibility:** *For $\beta > 0$ and $\gamma \geq 1$, vector $\boldsymbol{\alpha}$ is (β, γ)-admissible if: (1) $\sum_{i \in \mathcal{N}} \alpha_i = 1$, $\alpha_i \geq 0$, for each $i \in \mathcal{N}$; (2) $\alpha_j = 0$ for each $j \notin \mathcal{N}$; and (3) at least γ elements of $\boldsymbol{\alpha}$ are lower bounded by β.*

In this chapter, we focus on the impact of Byzantine attacks on the maximal achievable γ. To characterize the fundamental limits on γ, we assume that the argument of each local cost function is a (real-valued) scalar, the network is fully-connected, and there is no restriction on the information exchange among agents.

Contributions In its general form, in (3.2) above, the argument x of the cost function h_i is a k-dimensional vector of reals (i.e., $x \in \mathbb{R}^k$), where $k \geq 1$. In this chapter, as a first step towards solving the fault-tolerant multi-agent optimization problem, we consider the special case when $k = 1$, i.e., x is a scalar. Problem (3.2) remains open for vector arguments with $k \geq 2$. Later in the chapter, we discuss the technical difficulty in solving the problem with vector inputs. We prove the following key results:

1. (Theorem 11) Problem (3.1) is not solvable when $f > 0$.
   That is, the local objectives kept by the non-faulty agents cannot be guaranteed to be utilized equally for all executions.

2. (Theorem 12) In a synchronous system, when $f > 0$, for problem (3.2), it is impossible to guarantee that $\boldsymbol{\alpha}$ is (β, γ)-admissible with $\gamma > |\mathcal{N}|-f$ for all executions.
   That is, any synchronous system cannot guarantee to utilize more than $|\mathcal{N}| - f$ local objectives of the non-faulty agents.

3. (Theorem 13) When $n > 3f$, problem (3.2) is solvable in a synchronous system with $\beta = \frac{1}{2(|\mathcal{N}|-f)}$ and $\gamma = |\mathcal{N}| - f$ for all executions.
   We prove this claim by constructing algorithms. Our algorithms are *optimal* in the sense that they match the bound in Theorem 12. By exploiting Byzantine broadcast for information exchange between agents, our proposed algorithms essentially solve a centralized problem where there are n functions among which up to f functions are injected by the system adversary. Nevertheless, these algorithms are useful for characterizing the fundamental limits that we are interested in.

4. (Section 3.6) We also propose a low-complexity suboptimal algorithm, where agents individually minimize local objectives, and run consensus over local optima. This suboptimal algorithm ensures that at least $\lceil \frac{n}{2} \rceil - \phi$ (i.e., $\gamma = \lceil \frac{n}{2} \rceil - \phi$) agents have weights that are bounded away from 0 nontrivially (indeed, $\beta = \frac{1}{|\mathcal{N}|}$), where ϕ ($\phi \leq f$) is the actual number of Byzantine agents in a given execution.

5. (Section 3.7) Finally, we present an iterative distributed algorithm that is optimal in the sense that it matches the bound in the impossibility result of Theorem 12. In particular, the proposed algorithm is

2-approximation within each index of the optimal convex combination, i.e., the achieved $\boldsymbol{\alpha}$ is $(\frac{1}{2(|\mathcal{N}|-f)},\ |\mathcal{N}|-f)$-admissible.

The iterative algorithm presented in Section 3.7 can be extended to a *constrained* version of the optimization problem, to the crash failure model, and to asynchronous environments. While much of the chapter addresses Byzantine faults in a synchronous completely connected network, Section 3.8 summarizes the above extensions, as well as a few open problems.

3.2 Related Work

The distributed optimization problem is related to Byzantine fault-tolerant consensus. In particular, the non-faulty agents are all required to produce (approximately) equal output; thus, consensus is part of the requirements satisfied by any solution for our problem. There is a significant body of work on fault-tolerant consensus, including our own prior work [32, 50, 51, 39, 33, 34, 52, 53, 54, 55, 56].

Although distributed optimization has a long history, we believe our work is the first to explore the problem of designing Byzantine fault-tolerant algorithms that achieve optimality in the sense defined earlier. Primal and dual decomposition methods that lend themselves naturally to a distributed paradigm have been known for at least fifty years, and their behavior is well understood [57]. In their seminal work, Tsitsiklis and colleagues [58, 59] analyze algorithms for minimization of a smooth function $h(x)$ by distributing the processing of the components vector $x \in \mathbb{R}^n$ among n agents assuming $h(x)$ is separable. As noted earlier, there has been significant research on problem (1.1). The need for robustness for distributed optimization problems has received some attention recently [3, 8, 60, 61]. Duchi et al. [3] and Lobel and Ozdaglar [8] study the impact of random communication link failures on the convergence of distributed variant of dual averaging algorithm and sub-gradient method, respectively. In particular, both [3] and [8] assume that each realizable link failure pattern admits a doubly-stochastic matrix which governs local estimates evolution dynamics. Byzantine agents are first considered in the context of optimization in our series of four technical reports [61, 62, 63, 64]. Subsequent to our work, [60, 65] also considered the fault-tolerant optimization problem but under a weaker model of faults. In

the weaker model, a faulty node must send identical messages to its neighbors, unlike the Byzantine faults. More importantly, the results obtained by [60, 65] do not demonstrate optimal fault-tolerance as achieved in our work. Also, since the Byzantine fault model is more general, our algorithms are also applicable under the weaker model in [60, 65].

3.3 System Model, Assumptions and Notations

The system under consideration is synchronous, and consists of $n > 3f$ agents, [1] where f is the maximum number of agents that may be Byzantine faulty. The communication network is completely connected (i.e., each agent has a communication channel to each of the agents). We discuss some extensions of our results in Section 3.8. The set of n agents is denoted $\mathcal{V} = \{1, \cdots, n\}$. In a given execution, let $\mathcal{F}$ denote the set of Byzantine faulty agents, and let $\mathcal{N} = \mathcal{V} - \mathcal{F}$ denote the set of non-faulty agents. The set $\mathcal{F}$ of faulty agents may be chosen by an adversary arbitrarily, and may be different across different executions.

We say that a function $h : \mathbb{R} \to \mathbb{R}$ is *admissible* if (i) h is convex, and continuously differentiable, (ii) the set $\arg\min_{x \in \mathbb{R}} h(x)$ containing the optima of h is non-empty and compact (i.e., bounded and closed), (iii) the magnitude of the gradients are bounded by L, i.e., $|h'(x)| \leq L, \forall x \in \mathbb{R}$. Each agent $i \in \mathcal{V}$ is initially provided with an *admissible* local cost function $h_i : \mathbb{R} \to \mathbb{R}$. Similar assumptions on the local functions are standard in past literature on failure-free distributed optimization [3, 4, 5, 6, 7, 8].

3.4 Impossibility Results

In this section, we derive an upper bound on γ.

For a given choice of $\alpha_i \geq 0$ such that $\alpha_i = 0$ for $i \in \mathcal{F}$ and $\sum_{i \in \mathcal{N}} \alpha_i = 1$,

[1] For Byzantine consensus to be reachable, $n > 3f$ is needed.

define X_i (for $i \in \mathcal{N}$) and X as follows:

$$X_i = \arg\min_{x \in \mathbb{R}} h_i(x), \forall\, i \in \mathcal{N};$$
$$X = \arg\min_{x} \sum_{i \in \mathcal{N}} \alpha_i h_i(x).$$

The connection between X_i (for $i \in \mathcal{N}$) and X is characterized in Proposition 1.

Proposition 1. *For any choice of $\alpha_i \geq 0$ such that $\alpha_i = 0$ for $i \in \mathcal{F}$ and $\sum_{i \in \mathcal{N}} \alpha_i = 1$, it holds that*

$$X \subseteq Cov\left(\cup_{i \in \mathcal{N}} X_i\right),$$

where $Cov\left(\cup_{i \in \mathcal{N}} X_i\right)$ is the convex hull of set $\cup_{i \in \mathcal{N}} X_i$.

Proposition 1 can be easily shown by contradiction.

Recall that we say that problem (3.1) or (3.2) is solvable if there exists an algorithm whose output minimizes the objective in (3.1) or (3.2), respectively, for all admissible local cost functions, and all possible behaviors of faulty nodes.

Theorem 11. *Problem* (3.1) *is not solvable when $f > 0$.*

Proof. We prove this theorem by contradiction. Suppose that there exists a correct algorithm $\mathcal{A}$ that solves problem (3.1). For each $x \in \mathbb{R}$, define the cost functions of the n agents as follows:

- $h_1(x) = (x+1)^2$,
- $h_n(x) = (x-1)^2$, and
- $h_i(x) = x^2 + i$, where $2 \leq i \leq n-1$.

Note that the functions defined above satisfy the admissibility conditions specified in Section 3.3 except for the "bounded gradient" condition. However, the "bounded gradient" condition can be easily enforced by carefully modifying the functions values (and correspondingly gradient values) for x that are far enough away from convex hull of optima of the set of functions listed above.

It is easy to see that $X_1 = \{-1\}, X_n = \{1\}$, and for $2 \leq i \leq n-1$, $X_i = \{0\}$. We consider two executions wherein $\mathcal{A}$ produces different outputs, and show that there exists a non-faulty agent that cannot distinguish these two executions.

The identities of the faulty agents in these two executions are different. In both executions, the faulty nodes follow the algorithm correctly with the above choice of cost functions.

Execution 1: In execution 1, let $\mathcal{N} = \{1, \cdots, n-1\}$ and $\mathcal{F} = \{n\}$. Since $\mathcal{A}$ is a correct algorithm, by Proposition 1 it follows that the output of the algorithm must be in $Cov\left(\cup_{j=1}^{n-1} X_j\right) = [-1, 0]$ for all agents $i \in \{1, \cdots, n-1\}$.

Execution 2: In execution 2, let $\mathcal{N} = \{2, \cdots, n\}$ and $\mathcal{F} = \{1\}$. Since $\mathcal{A}$ is a correct algorithm, by Proposition 1 it follows that, in this case, the output of the algorithm must be in $Cov\left(\cup_{j=2}^{n} X_j\right) = [0, 1]$ for all agents $i \in \{2, \cdots, n\}$.

The agents in $\{2, \cdots, n-1\}$ cannot distinguish the above two executions, and hence must produce identical output in both cases. That is, their output must be 0 since $[-1, 0] \cap [0, 1] = \{0\}$. (When $f > 0$, $n \geq 3f + 1 = 4$. Thus, the set $\{2, \cdots, n-1\}$ is non-empty.)

On the other hand, it holds that $\sum_{i=1}^{n-1} h_i'(0) \neq 0$ and $\sum_{i=2}^{n} h_i'(0) \neq 0$, contradicting the hypothesis that 0 is an optimum for either execution – note that $h_i'(x)$ is the derivative of function h_i at x for each $1 \leq i \leq n$. This contradicts the assumption that $\mathcal{A}$ is correct and the proof is complete. □

Theorem 11 implies that potential faulty behavior of the Byzantine agents can confuse the system to deviate from minimizing $\frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} h_i$. Next, we characterize this deviation.

Theorem 12. *In a synchronous system, when up to f agents may be Byzantine faulty, for problem (3.2), it is impossible to guarantee that more than $|\mathcal{N}| - f$ weights in vector $\boldsymbol{\alpha}$ are non-zero. In other words, for any $\beta > 0$, it is impossible to guarantee that $\boldsymbol{\alpha}$ is (β, γ)-admissible with $\gamma > |\mathcal{N}| - f$.*

The proof of Theorem 12 is similar to the proof of Theorem 11, and can be found in Section 3.9.1.

By Theorem 12, we know that regardless of the value of parameter β, if $\gamma > |\mathcal{N}| - f$, no algorithm can solve (3.2).

3.5 Tightness of $\gamma \leq |\mathcal{N}| - f$: Optimal Algorithms

In this section, we present two different algorithms, both of which use *Byzantine broadcast* algorithm (such as [66]) as a communication primitive. The Byzantine broadcast algorithm allows a designated sender to a message to the other agents, while satisfying the following properties when $n > 3f$:

- all the non-faulty agents decide on an identical value, and
- if the sender is non-faulty, then the received value decided by the non-faulty agents is the sender's proposed value.

In the first algorithm, by broadcasting the local functions using Byzantine broadcast, each non-faulty agent knows all the n local functions over the whole system, among which up to f functions may be faulty. We show that the non-faulty agents are essentially minimizing a global objective $\mathbf{H}$ (defined in Theorem 14) instead of the ideal objective $\frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}} h_i$.

Although broadcasting local cost functions may be costly and not always practical, it allows us to derive the mathematics basis for a more practical algorithm. Indeed, the second algorithm can be viewed as an implementation of the first algorithm using a gradient descent method.

3.5.1 Algorithm 2

Given a set of admissible functions $\{h_1, \cdots, h_n\}$, for each $x \in \mathbb{R}$, define multisets $A(x), B(x), C(x)$ below, where $h_i'(x)$ denotes the gradient of function h_i at x.

$$\begin{aligned} A\,(x) &\triangleq \{i:\ h_i'\,(x) > 0\},\\ B\,(x) &\triangleq \{i:\ h_i'\,(x) < 0\},\\ C\,(x) &\triangleq \{i:\ h_i'\,(x) = 0\}. \end{aligned} \tag{3.3}$$

Let $F_1^*(x) \subseteq A\,(x)$ and $F_2^*(x) \subseteq B\,(x)$ such that

$$\begin{aligned} F_1^*(x) &\in \arg \min_{F_1 \subseteq A(x), |F_1| \leq f} \sum_{i \in A(x) - F_1} h_i'\,(x)\,,\\ F_2^*(x) &\in \arg \max_{F_2 \subseteq B(x), |F_2| \leq f} \sum_{i \in B(x) - F_2} h_i'\,(x)\,. \end{aligned} \tag{3.4}$$

All the non-faulty agents follow Algorithm 2. Faulty agents can deviate from the following description arbitrarily.

Algorithm 2: pseudo-code for agent j

1 Perform Byzantine broadcast of local cost function.
2 **if** *there exists* $x \in \mathbb{R}$ *such that*

$$\sum_{i \in A(x) - F_1^*(x)} h_i'(x) + \sum_{i \in B(x) - F_2^*(x)} h_i'(x) \;=\; 0 \tag{3.5}$$

then
3 deterministically choose output x_o to be any one x value that satisfies (3.5);
4 **else**
5 choose output $x_o = \perp$.
6 **end**

Note that in step 1, agent j should receive from each agent $i \in \mathcal{V}$ its cost function h_i. In step 1, each agent j broadcasts a complete description of its cost function to other agents, using any Byzantine broadcast algorithm, such as [66]. For non-faulty agent $i \in \mathcal{N}$, h_i will be an admissible function (*admissible* is defined in Section 3.3). If a faulty agent $k \in \mathcal{F}$ does not correctly perform Byzantine broadcast of its cost function, or broadcasts an inadmissible cost function, then hereafter assume h_k to be a *default* admissible cost function that is known to all agents.

For the multiset $\{h_1, \cdots, h_n\}$ of n admissible cost functions gathered in step 1 of Algorithm 2, define function $F(\cdot)$ and function $G(\cdot)$ as follows. For each $x \in \mathbb{R}$,

$$F(x) \triangleq \sum_{i \in A(x) - F_1^*(x)} h_i'(x), \text{ and} \tag{3.6}$$

$$G(x) \triangleq \sum_{i \in B(x) - F_2^*(x)} h_i'(x). \tag{3.7}$$

Note that $F(x) \geq 0$ and $G(x) \leq 0$ for each $x \in \mathbb{R}$. In particular, $F(x) = 0$ if $|A(x)| \leq f$ and $F(x) > 0$ otherwise. Similarly, $G(x) = 0$ if $|B(x)| \leq f$ and $G(x) < 0$ otherwise.

Besides, the functions F and G have the following properties. The correctness of Algorithm 2 relies crucially on the following proposition.

Proposition 2. *Functions $F(\cdot)$ and $G(\cdot)$ are both non-decreasing and continuous over $\mathbb{R}$.*

The proof of Proposition 2 can be found in Section 3.9.2. The monotonicity and continuity of functions F and G imply the existence of x satisfying (3.5), formally stated next.

Lemma 3. *Algorithm 2 returns $x_o \in \mathbb{R}$ when $n > 3f$ (i.e., it never returns $\perp$).*

Proof. Consider the multiset of admissible functions $\{h_1, h_2, \cdots, h_n\}$ obtained by a non-faulty agent in step 1 of Algorithm 2. Recall that $X_i = \arg\min_{x \in \mathbb{R}} h_i(x)$. Let $\max X_i$ and $\min X_i$ denote the largest and smallest values in X_i, respectively. Sort the above n functions h_i in an *increasing* order of their $\max X_i$ values, breaking ties arbitrarily. Let i_0 denote the $f+1$-th agent in this sorted order (i.e., i_0 has the $f+1$-th smallest value in the above sorted order). Similarly, sort the functions h_i in a *decreasing* order of $\min X_i$ values, breaking ties arbitrarily. Let j_0 denote the $f+1$-th agent in this sorted order (i.e., j_0 has the $f+1$-th largest value in the above sorted order).

Define function H as

$$H(x) = F(x) + G(x), \quad \text{for each } x \in \mathbb{R}.$$

Consider $x_1 \in X_{i_0}$ and $x_2 \in X_{j_0}$. It follows that $|A(x_1)| \leq f$ and $|B(x_2)| \leq f$. Thus,

$$F(x_1) = 0 = G(x_2).$$

So, we obtain

$$\begin{aligned} H(x_1) &= F(x_1) + G(x_1) = 0 + G(x_1) \leq 0, \quad \text{and} \\ H(x_2) &= F(x_2) + G(x_2) = F(x_2) + 0 \geq 0. \end{aligned}$$

If $H(x_1) = 0$ or $H(x_2) = 0$, then x_1 or x_2, respectively, satisfy equation (3.5), proving the lemma. (Note that $H = F + G$, and the definition of F and G implies that, if $H(x_i) = 0$ then x_i satisfies equation (3.5).)

Let us now consider the case when $H(x_1) < 0$ and $H(x_2) > 0$. By Proposition 2, we know that $H(\cdot)$ is non-decreasing and continuous. Then it follows

that $x_1 \leq x_2$, and there exists $x_o \in [x_1, x_2]$ such that $H(x_o) = 0$, i.e., x_o satisfies equation (3.5), proving the lemma.

□

Next we present our main theorem. The following theorem says that the output x_o of Algorithm 2 satisfies the conditions listed in (3.2) with $\gamma = |\mathcal{N}| - f$, proving that the bound on γ stated in Theorem 12 is tight for certain values of β (as stated in the theorem below).

Theorem 13. *When $n > 3f$, the output x_o of Algorithm 2 satisfies the conditions listed in* (3.2) *with $\gamma = |\mathcal{N}| - f$ and $\beta = \frac{1}{2(|\mathcal{N}|-f)}$.*

Proof. By Lemma 3, we know that Algorithm 1 returns a value in $\mathbb{R}$. Let $\widetilde{x}$ be the output of Algorithm 1 for the set of functions $\{h_1, \cdots, h_n\}$ gathered in Step 1 of the algorithm. Let $F_1^* \subseteq A(\widetilde{x})$ and $F_2^* \subseteq B(\widetilde{x})$, with $|F_1^*| \leq f$ and $|F_2^*| \leq f$, be the sets that minimize $\sum_{i \in A(\widetilde{x}) - F_1} h_i'(\widetilde{x})$, and maximize $\sum_{i \in B(\widetilde{x}) - F_2} h_i'(\widetilde{x})$, respectively (as per equation (3.5)).

Recall that $\mathcal{V} = \{1, \ldots, n\}$. Sort the elements in the multiset

$$\{h_1'(\widetilde{x}), \ldots, h_n'(\widetilde{x})\}$$

in a non-increasing order, breaking ties in such a way that the elements corresponding to the agents in F_1^* are among the first f elements in the sorted order and the elements corresponding to the agents in F_2^* are among the last f elements in the sorted order. Such a sorted order is well-defined since $|F_1^*| \leq f$ and $|F_2^*| \leq f$. Let $\bar{F}_1 \subseteq \mathcal{V}$ be the agents corresponding to the first f elements in the sorted order, and let $\bar{F}_2 \subseteq \mathcal{V}$ be the agents corresponding to the last f elements in the sorted order. Note that $F_1^* \subseteq \bar{F}_1$ and $F_2^* \subseteq \bar{F}_2$. Since $A(\widetilde{x}), B(\widetilde{x})$ and $C(\widetilde{x})$ form a partition of $\mathcal{V}$, we have

$$\begin{aligned}
\sum_{i \in \mathcal{V} - F_1^* - F_2^*} h_i'(\widetilde{x}) &= \sum_{i \in C(\widetilde{x})} h_i'(\widetilde{x}) + \sum_{i \in A(\widetilde{x}) \cup B(\widetilde{x}) - F_1^* - F_2^*} h_i'(\widetilde{x}) \\
&\overset{(a)}{=} 0 + \sum_{i \in A(\widetilde{x}) \cup B(\widetilde{x}) - F_1^* - F_2^*} h_i'(\widetilde{x}) \\
&\overset{(b)}{=} 0 + 0 = 0. \qquad (3.8)
\end{aligned}$$

Equality (a) follows by definition of $C(\widetilde{x})$, and equality (b) is true because $\widetilde{x}$

satisfies equation (3.5). Denote $\mathcal{R}^* = \mathcal{V} - \bar{F}_1 - \bar{F}_2$. Next we show that

$$\sum_{i \in \mathcal{R}^*} h_i'(\widetilde{x}) = 0. \tag{3.9}$$

If $|A(\widetilde{x})| \geq f$, by definition of F_1^*, it holds that $|F_1^*| = f$. Thus, $\bar{F}_1 = F_1^*$. Consequently, we have

$$\sum_{i \in \bar{F}_1 - F_1^*} h_i'(\widetilde{x}) = \sum_{i \in \emptyset} h_i'(\widetilde{x}) = 0.$$

If $|A(\widetilde{x})| < f$, by definition of F_1^* and $\bar{F}_1$, and the fact that $F_1^* \subset \bar{F}_1$, it follows that $F_1^* = A(\widetilde{x})$, and $h_i'(\widetilde{x}) \leq 0$ for each $i \in \bar{F}_1 - F_1^* = \bar{F}_1 - A(\widetilde{x}) \neq \emptyset$. In addition, if there exists $i \in \bar{F}_1 - F_1^*$ such that $h_i'(\widetilde{x}) < 0$, then by definition of $\bar{F}_1$, we have $h_j'(\widetilde{x}) < 0$ for each $j \in \mathcal{V} - \bar{F}_1$. So we get

$$\begin{aligned}
0 = & \sum_{i \in \mathcal{V} - F_1^* - F_2^*} h_i'(\widetilde{x}) \quad \text{by (3.8)} \\
= & \sum_{i \in \mathcal{V} - \bar{F}_1 - F_2^*} h_i'(\widetilde{x}) + \sum_{i \in \bar{F}_1 - F_1^*} h_i'(\widetilde{x}) \quad \text{since } F_1^* \subseteq \bar{F}_1 \\
\leq & \sum_{i \in \mathcal{V} - \bar{F}_1 - F_2^*} h_i'(\widetilde{x}) \quad \text{since } h_i'(\widetilde{x}) \leq 0, \forall\ i \in \bar{F}_1 - F_1^* \\
< & 0 \quad \text{since } h_i'(\widetilde{x}) < 0,\ \forall\ i \in \mathcal{V} - \bar{F}_1,
\end{aligned}$$

proving a contradiction. Thus, there does not exist $i \in \bar{F}_1 - F_1^*$ such that $h_i'(\widetilde{x}) < 0$, i.e., $h_i'(\widetilde{x}) = 0$ for each $i \in \bar{F}_1 - F_1^*$. Consequently, we have

$$\sum_{i \in \bar{F}_1 - F_1^*} h_i'(\widetilde{x}) = \sum_{i \in \bar{F}_1 - F_1^*} 0 = 0.$$

Hence, regardless of the size of $|A(\widetilde{x})|$, the following is always true.

$$\sum_{i \in \bar{F}_1 - F_1^*} h_i'(\widetilde{x}) = 0. \tag{3.10}$$

Similarly, we can show that

$$\sum_{i \in \bar{F}_2 - F_2^*} h_i'(\widetilde{x}) = 0. \tag{3.11}$$

Therefore, we have

$$\begin{aligned}
0 &= \sum_{i\in\mathcal{V}-F_1^*-F_2^*} h_i'(\widetilde{x}) \quad \text{by (3.8)} \\
&= \sum_{i\in\mathcal{V}-\bar{F}_1-\bar{F}_2} h_i'(\widetilde{x}) + \sum_{i\in\bar{F}_1-F_1^*} h_i'(\widetilde{x}) + \sum_{i\in\bar{F}_2-F_2^*} h_i'(\widetilde{x}) \\
&= \sum_{i\in\mathcal{R}^*} h_i'(\widetilde{x}) + \sum_{i\in\bar{F}_1-F_1^*} h_i'(\widetilde{x}) + \sum_{i\in\bar{F}_2-F_2^*} h_i'(\widetilde{x}) \\
&= \sum_{i\in\mathcal{R}^*} h_i'(\widetilde{x}) + 0 + 0 \\
&= \sum_{i\in\mathcal{R}^*} h_i'(\widetilde{x}),
\end{aligned}$$

proving equation (3.9).
Let $\widetilde{F}_1 \subseteq \bar{F}_1 - \mathcal{F}$ and $\widetilde{F}_2 \subseteq \bar{F}_2 - \mathcal{F}$ such that

$$|\widetilde{F}_1| = f - \phi + |\mathcal{R}^* \cap \mathcal{F}| \text{ and } |\widetilde{F}_2| = f - \phi + |\mathcal{R}^* \cap \mathcal{F}|. \tag{3.12}$$

Since $|\mathcal{F}| = \phi \leq f$, $|\bar{F}_1| = f = |\bar{F}_2|$, and $\mathcal{R}^* \cup \bar{F}_1 \cup \bar{F}_2 = \mathcal{V}$, it holds that

$$|\bar{F}_1 - \mathcal{F}| \geq f - \phi + |\mathcal{R}^* \cap \mathcal{F}| \quad \text{and} \quad |\bar{F}_2 - \mathcal{F}| \geq f - \phi + |\mathcal{R}^* \cap \mathcal{F}|.$$

Thus, $\widetilde{F}_1$ and $\widetilde{F}_2$ are well-defined.

We now show that

$$\sum_{i\in\widetilde{F}_1} h_i'(\widetilde{x}) \geq 0 \text{ and } \sum_{i\in\widetilde{F}_2} h_i'(\widetilde{x}) \leq 0. \tag{3.13}$$

Suppose $\sum_{i\in\widetilde{F}_1} h_i'(\widetilde{x}) < 0$, then there exists $i_0 \in \widetilde{F}_1 \subseteq \bar{F}_1 - \mathcal{F}$ such that $h_{i_0}'(\widetilde{x}) < 0$. Since agents in $\bar{F}_1$ have the f largest values (including ties) in the set $\{h_1'(\widetilde{x}), \ldots, h_n'(\widetilde{x})\}$, then $h_i'(\widetilde{x}) < 0$ for each $i \in \mathcal{R}^*$, contradicting the fact that (3.9) holds. Analogously, it can be shown that $\sum_{i\in\widetilde{F}_2} h_i'(\widetilde{x}) \leq 0$.

In addition, we observe that

$$\sum_{i\in\widetilde{F}_2} h_i'(\widetilde{x}) \leq \sum_{i\in\mathcal{R}^*\cap\mathcal{F}} h_i'(\widetilde{x}) \leq \sum_{i\in\widetilde{F}_1} h_i'(\widetilde{x}). \tag{3.14}$$

To see this, consider three possibilities:
(i) $\sum_{i\in\mathcal{R}^*\cap\mathcal{F}} h_i'(\widetilde{x}) = 0$;

(ii) $\sum_{i\in\mathcal{R}^*\cap\mathcal{F}} h_i'(\widetilde{x}) > 0$;
(iii) $\sum_{i\in\mathcal{R}^*\cap\mathcal{F}} h_i'(\widetilde{x}) < 0$.

Consider the case when $\sum_{i\in\mathcal{R}^*\cap\mathcal{F}} h_i'(\widetilde{x}) = 0$ (i.e., case (i)).

Due to (3.13) and the case assumption, it holds that

$$\sum_{i\in\widetilde{F}_2} h_i'(\widetilde{x}) \le 0 = \sum_{i\in\mathcal{R}^*\cap\mathcal{F}} h_i'(\widetilde{x}) = 0 \le \sum_{i\in\widetilde{F}_1} h_i'(\widetilde{x}),$$

which is (3.14).

Now consider the case when $\sum_{i\in\mathcal{R}^*\cap\mathcal{F}} h_i'(\widetilde{x}) > 0$ (i.e., case (ii)).

Since $\sum_{i\in\mathcal{R}^*\cap\mathcal{F}} h_i'(\widetilde{x}) > 0$, it follows that $\mathcal{R}^*\cap\mathcal{F} \neq \emptyset$, and there exists $k \in \mathcal{R}^*\cap\mathcal{F}$ such that $h_k'(\widetilde{x}) > 0$. This implies that $h_i(\widetilde{x}) > 0$ for each $i \in \widetilde{F}_1$. Let $\mu = \min_{i\in\widetilde{F}_1} h_i'(\widetilde{x})$. Note that $\mu > 0$. By definition of $\widetilde{F}_1$, it follows that

$$h_i'(\widetilde{x}) \le \mu \le h_j'(\widetilde{x}),$$

for each $i \in \mathcal{R}^*$ and $j \in \widetilde{F}_1$. Thus, we obtain

$$\begin{aligned}\sum_{i\in\mathcal{R}^*\cap\mathcal{F}} h_i'(\widetilde{x}) &\le \sum_{i\in\mathcal{R}^*\cap\mathcal{F}} \mu = (|\mathcal{R}^*\cap\mathcal{F}|)\,\mu \le \left(|\widetilde{F}_1|\right)\mu \\ &= \sum_{i\in\widetilde{F}_1} \mu \le \sum_{i\in\widetilde{F}_1} h_i'(\widetilde{x}). \end{aligned} \tag{3.15}$$

Due to (3.13) and the assumption that $\sum_{i\in\mathcal{R}^*\cap\mathcal{F}} h_i'(\widetilde{x}) > 0$, we get

$$\sum_{i\in\widetilde{F}_2} h_i'(\widetilde{x}) \le 0 < \sum_{i\in\mathcal{R}^*\cap\mathcal{F}} h_i'(\widetilde{x}) \le \sum_{i\in\widetilde{F}_1} h_i'(\widetilde{x}),$$

proving relation (3.14).

Similarly, we can show the case when $\sum_{i\in\mathcal{R}^*\cap\mathcal{F}} h_i'(\widetilde{x}) < 0$ (i.e., case (iii)).

Since the relation in (3.14) holds, there exists $0 \le \zeta \le 1$ such that

$$\sum_{i\in\mathcal{R}^*\cap\mathcal{F}} h_i'(\widetilde{x}) = \zeta\cdot\sum_{i\in\widetilde{F}_1} h_i'(\widetilde{x}) + (1-\zeta)\cdot\sum_{i\in\widetilde{F}_2} h_i'(\widetilde{x}). \tag{3.16}$$

Thus, we have

$$\begin{aligned}
0 &= \sum_{i \in \mathcal{R}^*} h_i'(\widetilde{x}) \\
&= \sum_{i \in \mathcal{R}^* - \mathcal{F}} h_i'(\widetilde{x}) + \sum_{i \in \mathcal{R}^* \cap \mathcal{F}} h_i'(\widetilde{x}) \\
&= \sum_{i \in \mathcal{R}^* - \mathcal{F}} h_i'(\widetilde{x}) + \zeta \cdot \sum_{i \in \widetilde{F}_1} h_i'(\widetilde{x}) + (1-\zeta) \cdot \sum_{i \in \widetilde{F}_2} h_i'(\widetilde{x}).
\end{aligned}$$

Thus $\widetilde{x}$ is an optimum of function

$$\sum_{i \in \mathcal{R}^* - \mathcal{F}} h_i + \zeta \cdot \sum_{i \in \widetilde{F}_1} h_i + (1-\zeta) \cdot \sum_{i \in \widetilde{F}_2} h_i.$$

Since constant scaling does not change optima, it follows that $\widetilde{x}$ is an optimum of function

$$\chi \left(\sum_{i \in \mathcal{R}^* - \mathcal{F}} h_i + \zeta \sum_{i \in \widetilde{F}_1} h_i + (1-\zeta) \sum_{i \in \widetilde{F}_2} h_i \right), \tag{3.17}$$

where

$$\chi = \frac{1}{|\mathcal{R}^* - \mathcal{F}| + \zeta |\widetilde{F}_1| + (1-\zeta) |\widetilde{F}_2|}.$$

Since $|\mathcal{R}^*| = n - 2f$ and $|\widetilde{F}_1| = f - \phi + |\mathcal{R}^* \cap \mathcal{F}| = |\widetilde{F}_2|$, we have

$$\begin{aligned}
|\mathcal{R}^* - \mathcal{F}| + \zeta |\widetilde{F}_1| + (1-\zeta) |\widetilde{F}_2| &= |\mathcal{R}^* - \mathcal{F}| + |\widetilde{F}_1| \qquad \text{since } |\widetilde{F}_1| = |\widetilde{F}_2| \\
&= |\mathcal{R}^* - \mathcal{F}| + f - \phi + |\mathcal{R}^* \cap \mathcal{F}| \\
&= |\mathcal{R}^*| - |\mathcal{R}^* \cap \mathcal{F}| + f - \phi + |\mathcal{R}^* \cap \mathcal{F}| \\
&= |\mathcal{R}^*| + f - \phi = n - 2f + f - \phi \\
&= n - \phi - f = |\mathcal{N}| - f.
\end{aligned}$$

We know that either $\zeta \geq \frac{1}{2}$ or $1 - \zeta \geq \frac{1}{2}$; by symmetry, without loss of generality, assume $\zeta \geq \frac{1}{2}$. In addition, we know

$$\begin{aligned}
|(\mathcal{R}^* - \mathcal{F}) \cup \widetilde{F}_1| &= |\mathcal{R}^* - \mathcal{F}| + |\widetilde{F}_1| \\
&= |\mathcal{R}^*| - |\mathcal{R}^* \cap \mathcal{F}| + f - \phi + |\mathcal{R}^* \cap \mathcal{F}| \\
&= |\mathcal{N}| - f.
\end{aligned}$$

Recall that $\mathcal{R}^* \cup \bar{F}_1 \cup \bar{F}_2 = \mathcal{V}$. Thus, in function (3.17), which is a weighted sum of $|\mathcal{N}|$ local cost functions corresponding to agents in $\mathcal{N} = \mathcal{V} - \mathcal{F}$, at least $|\mathcal{N}| - f$ local cost functions corresponding to $i \in (\mathcal{R}^* - \mathcal{F}) \cup \widetilde{F}_1$ have weights that are lower bounded by $\frac{1}{2(|\mathcal{N}|-f)}$.

Similarly, when $1 - \zeta \geq \frac{1}{2}$, at least $|\mathcal{N}| - f$ cost functions corresponding to $i \in (\mathcal{R}^* - \mathcal{F}) \cup \widetilde{F}_2$ have weight lower bounded by $\frac{1}{2(|\mathcal{N}|-f)}$.

□

Recall that $H(\cdot) = F(\cdot) + G(\cdot)$. The global objective of the non-faulty agents in Algorithm 2 is characterized as follows. To simplify notation, let $[a, b] \triangleq Cov(\cup_{i \in \mathcal{N}} X_i)$.

Theorem 14. *For given $\mathcal{N}$ and $\mathcal{F}$, there exists a convex and differentiable function $\mathbf{H}(\cdot)$ defined over any finite interval $[c, d] \supseteq Cov(\cup_{i \in \mathcal{N}} X_i)$ such that the derivative function of $\mathbf{H}(\cdot)$ is $H(\cdot)$, i.e., $\mathbf{H}'(x) = H(x)$ for each $x \in [c, d]$ where $Cov(\cup_{i \in \mathcal{N}} X_i) \subseteq [c, d]$.*

The proof of Theorem 14 is presented in [61]. Theorem 14 says that associated with $H(\cdot)$, there is a function that is convex, differentiable, and has $H(\cdot)$ as its derivative function. The finite interval requirement in Theorem 14 is placed for detailed technical issue in calculus.

Remark 2. *The correctness of Algorithm 2 implies that $H(x_o) = 0$ and $x_o \in Cov(\cup_{i \in \mathcal{N}} X_i)$, where the latter claim follows from Proposition 1, proved in Section 3.4. Essentially, Algorithm 2 outputs an optimum of the following constrained convex optimization problem, where $Cov(\cup_{i \in \mathcal{N}} X_i) \subseteq [c, d]$:*

$$\begin{aligned} \min \quad & \mathbf{H}(x) \\ s.t. \quad & x \in [c, d]. \end{aligned} \tag{3.18}$$

3.5.2 Algorithm 3

Unlike Algorithm 2, Algorithm 3 presented below does *not* require the agents to exchange their local cost functions in their entirety. Instead, agents exchange gradients of their local cost functions via Byzantine broadcast. Indeed, Algorithm 3 can be viewed as an implementation of Algorithm 2 using (centralized) gradient descent method on the optimization problem stated

in (3.18). The main challenge here is that the Byzantine agents can behave arbitrarily – there is no restriction on the local cost functions (if any) kept by the faulty agents. To overcome this difficulty, "admissibility check" primitive is incorporated.

Recall that the gradient of each admissible function is bounded by L, i.e., $|h'(x)| \leq L$ for $x \in \mathbb{R}$. Let $\{\lambda[t]\}_{t=0}^{\infty}$ be a sequence of diminishing (non-increasing and $\lambda[t] \to 0$) stepsizes chosen beforehand such that $\lambda[t] > 0$ for each t, $\sum_{t=0}^{\infty} \lambda[t] = \infty$ and $\sum_{t=0}^{\infty} \lambda^2[t] < \infty$. In the initialization steps of Algorithm 3, it is sufficient to require that every agent has identical initial estimate.

Algorithm 3: for agent j and $t \geq 1$:

1 Initialization (i): Choose $v_j \in X_j = \text{argmin}_{x \in \mathbb{R}} h_j(x)$;
2 Initialization (ii): Perform exact Byzantine consensus (such as [49]) with v_j as the input of agent j to the consensus algorithm.
3 Initialization (iii): Set $x_j[0]$ to the output of the above consensus algorithm.

---

4 Compute $h'_j(x_j[t-1])$, and perform Byzantine broadcast (such as [66]) of $h'_j(x_j[t-1])$ to all the agents.
5 **for** $i \in \mathcal{V}$ **do**
6 | receive a gradient from agent i, denoted by $g_i[t-1]$
7 **end**
8 **for** *each* $j \in \mathcal{V}$ **do**
9 | check for *admissibility* of the sequence $(t, g_i[t-1])$
10 **end**
11 Let $\mathcal{R}[t-1]$ be the multiset of admissible gradients $\{g_1[t-1], \cdots, g_n[t-1]\}$ obtained in steps 4-6.
12 **if** *there are* $> f$ *positive gradients in* $\mathcal{R}[t-1]$ **then**
13 | remove f largest gradients from $\mathcal{R}[t-1]$
14 **else**
15 | remove all positive gradients from $\mathcal{R}[t-1]$
16 **end**
17 **if** *there are* $> f$ *negative gradients in* $\mathcal{R}[t-1]$ **then**
18 | remove f smallest gradients from $\mathcal{R}[t-1]$
19 **else**
20 | remove all negative gradients from $\mathcal{R}[t-1]$
21 **end**
22 Let $\mathcal{R}^*[t-1]$ be the set of agents corresponding to all the remaining gradients. $x_j[t] \leftarrow x_j[t-1] - \lambda[t-1] \sum_{i \in \mathcal{R}^*[t-1]} g_i[t-1]$.

In Algorithm 3, agent j keeps a record of the sequence $(t, x_j[t])$, and a record of the sequence $(t, g_i[t-1])$ for each agent i. For each $t \geq 1$, agent j checks each received gradient $g_i[t-1]$ for *admissibility* as follows:

- If no gradient is received from agent i in iteration t via a Byzantine broadcast from i, then the gradient $g_i[t-1]$ for agent i is deemed *inadmissible*.
- If there exists an iteration $1 \leq t_0 < t$ such that at least one of the following conditions is true, then the gradient received from agent i is deemed *inadmissible*.
  1. $x_j[t_0 - 1] \leq x_j[t-1]$ and $g_i[t_0 - 1] > g_i[t-1]$
  2. $x_j[t_0 - 1] \geq x_j[t-1]$ and $g_i[t_0 - 1] < g_i[t-1]$
  3. $|g_i[t-1]| > L$

If the gradient received from any agent i is deemed inadmissible, then it must be the case that agent i is faulty. In that case, agent i is isolated (i.e., removed from the system). This reduces the total number of agents n by 1, and the maximum number of faulty agents f is also reduced by 1. Algorithm 3 is *restarted* (from Step 4) using the new parameters n and f.[2] The gradients received from any non-faulty agent $i \in \mathcal{N}$ will never be found to be inadmissible.

Note that due to the restart mechanism above, the algorithm progresses to step 10 only when all the received gradients are deemed admissible. By performing initialization steps (i) and (ii), it holds that $x_j[0] \in Cov\left(\cup_{j \in \mathcal{N}} X_j\right)$ – the constraint in (3.18) is satisfied initially. The above claim follows trivially from *validity* condition imposed on a correct Byzantine consensus algorithm. Indeed, the contraint in (3.18) is satisfied throughout the execution of Algorithm 3.

Proposition 3. *In Algorithm 3,* $x_i[t] = x_j[t]$ *and*

$$x_i[t] \in [a - n\lambda[0]L,\, b + n\lambda[0]L]$$

for all $i, j \in \mathcal{N}$ *and for all* t.

[2] It is also possible to continue executing the algorithm further, but for brevity, we take the approach of eliminating the faulty agent, and restarting.

Proposition 3 is proved in [61]. Henceforth, we drop the subscript j of $x_j[t]$ for each $j \in \mathcal{N}$ and t. Similarly, we drop the time index [0] of $\lambda[0]$.

As an implementation of Algorithm 2, the admissibility check in step 1 of Algorithm 3 is necessary. Non-faulty agents in the system know that each non-faulty local function is admissible. As a result of this, in Algorithm 2, each faulty agent is forced to broadcast an admissible function. Similarly, in Algorithm 3, using admissibility check, each faulty agent is forced to behave as if its local function is admissible.

Theorem 15. *For any $i \in \mathcal{F}$, let $\{g_i[t-1])\}_{t=1}^{\infty}$ be the sequence of admissible gradients generated in Algorithm 3, where $g_i[t-1]$ is the gradient at $x[t-1]$. Then there exists a function $g(x)$ defined over $[c,d]$, which contains points $a-n\lambda L$ and $b+n\lambda L$ as interior points, such that (i) $g'(x[t-1]) = g_i[t-1]$, and (ii) $g(x)$ is convex, L–Lipschitz, and differentiable.*

The proof of Theorem 15 is presented in [61]. It is easy to see that there exists an admissible function $\bar{g}$ such that the restriction of $\bar{g}$ to $[c,d]$ equals g, i.e., $\bar{g}|_{[c,d]} = g$. For ease of further reference, we term the functions constructed in Theorem 15 as local virtual functions. Therefore, hereafter we can assume that all agents, including faulty agents, behave correctly and consistently with an admissible local cost function.

Recall that $[a,b] = Cov\left(\cup_{i\in\mathcal{N}} X_i\right)$. By Proposition 3, we know that the local estimate of each non-faulty agent i is trapped within the closed interval $[a-n\lambda L, b+n\lambda L]$ for all iterations, i.e., $x_i[t] \in [a-n\lambda L, b+n\lambda L]$ for all $i \in \mathcal{N}$ and all t. Therefore, Algorithm 3 is essentially trying to find an (exact or approximate) optimum of the following constrained convex optimization problem, which is a variant of (3.18):

$$\begin{aligned} \min \quad & \mathbf{H}(x) \\ s.t. \quad & x \in [a-nL\lambda, b+nL\lambda]. \end{aligned}$$

Theorem 14 and equation (3.5) in Algorithm 2 together imply that the x_o output of Algorithm 2 is an optimum of function $\mathbf{H}(\cdot)$ defined in Theorem 14. Also, it should be easy to see that the total gradient $\sum_{i\in\mathcal{R}^*[t-1]} g_i[t-1]$ used in computing $x_j[t]$ is identical to $F(x_j[t-1]) + G(x_j[t-1])$, which is the gradient of $\mathbf{H}$ at $x_j[t-1]$. In other words, the agents are distributedly using the gradient method for convex optimization of global cost function $\mathbf{H}$,

which is convex and continuous. Following the convergence analysis of the gradient method in Theorem 3.2.2 in [67] and Theorem 41 in [68], we can show that the limit of $\{x[t]\}_{t=0}^{\infty}$ exists and $\lim_{t\to\infty} x[t] = x^*$, where x^* is an optimum of function $\mathbf{H}$.

3.6 Suboptimal Algorithm

Algorithms 2 and 3 both use the costly Byzantine broadcast as subroutines. In contrast, in Algorithm 4, each agent optimizes its local cost function locally and exchanges the local optima, using an arbitrary Byzantine consensus algorithm. In addition, the correctness proof of Algorithm 4 *does not* require each h_i to be differentiable. However, Algorithm 4 is not an optimal algorithm. It only solves (3.2) with $\beta = \frac{1}{2|\mathcal{N}|}$ and $\gamma = \lceil \frac{n}{2} \rceil - \phi$, instead of the optimal $\gamma^* = |\mathcal{N}| - f$ achieved by Algorithms 2 and 3.

Algorithm 4: For agent $j \in \mathcal{N}$

1. Choose $v_j \in X_j = \text{argmin}_{x\in\mathbb{R}}\, h_j(x)$;
2. Send v_j to all agents, and receive messages from all agents. Agent j should receive a value from each agent $i \in \mathcal{V}$ – let us denote the value received from agent i as w_{ij}. If no value is, in fact, received from agent i, then w_{ij} is set to be a predefined default value.
3. Sort w_{ij} in a non-increasing order, breaking tie arbitrarily, and set $x_j[0]$ to be the median of this order. (We choose $x_j[0]$ to be the w_{ij} whose rank is $\lceil \frac{n}{2} \rceil$.)
4. Perform exact Byzantine consensus algorithm with $x_j[0]$ as the input of agent j to the consensus algorithm.
5. Set $\widetilde{x}$ to be the output of the above consensus algorithm, and output $\widetilde{x}$.

Theorem 16. *When $n > 3f$, Algorithm 4 solves* (3.2) *with $\beta = \frac{1}{2|\mathcal{N}|}$ and $\gamma = \lceil \frac{n}{2} \rceil - \phi$.*

Proof. Let W_j denote the multiset obtained by agent j, i.e., $W_j = \{w_{1j}, \ldots, w_{nj}\}$. For each $x \in \mathbb{R}$, define $W_j^+(x)$ and $W_j^-(x)$ as follows:

$$W_j^+(x) = \{i :\ i \in \mathcal{N} \text{ and } w_{ij} \geq x\},$$

$$W_j^-(x) = \{i :\ i \in \mathcal{N} \text{ and } w_{ij} \leq x\}.$$

Note that $W_j^+(x) \cup W_j^-(x) = \mathcal{N}$ for each $x \in \mathbb{R}$, and that $w_{ij} = v_i$ for each $i \in \mathcal{N}$. It should also be noted that $W_j^+(x)$ and $W_j^-(x)$ are not necessarily disjoint.

Recall that $\phi = |\mathcal{F}|$. For each j, since $x_j[0]$ is chosen to be the median of the non-increasing order over W_j, we have

$$\begin{aligned} |W_j^+(x_j[0])| &= |\{i : \ i \in \mathcal{N} \text{ and } w_{ij} \geq x_j[0]\}| \\ &\geq \lceil \frac{n}{2} \rceil - \phi, \end{aligned}$$

and

$$\begin{aligned} |W_j^-(x_j[0])| &= |\{i : \ i \in \mathcal{N} \text{ and } w_{ij} \leq x_j[0]\}| \\ &\geq n - \lceil \frac{n}{2} \rceil - \phi + 1 \\ &\geq \lceil \frac{n}{2} \rceil - \phi. \end{aligned}$$

Let $i_0 \in \mathcal{N}$ and $j_0 \in \mathcal{N}$ be the agents such that $x_{i_0}[0] \leq x_j[0]$ for each $j \in \mathcal{N}$ and $x_{j_0}[0] \geq x_j[0]$ for each $j \in \mathcal{N}$. Since $\widetilde{x}$ is the output of a correct exact consensus algorithm, by validity, we have $x_{i_0}[0] \leq \widetilde{x} \leq x_{j_0}[0]$. Thus

$$\begin{aligned} \{i : i \in \mathcal{N}, w_{ii_0} \leq x_{i_0}[0]\} &\subseteq \{i : i \in \mathcal{N}, w_{ii_0} \leq \widetilde{x}\} \\ &= \{i : i \in \mathcal{N}, v_i \leq \widetilde{x}\}, \end{aligned}$$

and

$$\begin{aligned} \{i : i \in \mathcal{N}, w_{ij_0} \geq x_{j_0}[0]\} &\subseteq \{i : i \in \mathcal{N}, w_{ij_0} \geq \widetilde{x}\} \\ &= \{i : i \in \mathcal{N}, v_i \geq \widetilde{x}\}. \end{aligned}$$

Consequently, we have

$$\begin{aligned} |\{i : i \in \mathcal{N}, v_i \leq \widetilde{x}\}| &\geq |\{i : i \in \mathcal{N}, w_{ii_0} \leq x_{i_0}[0]\}| \\ &= \ |W_{i_0}^-(x_{i_0}[0])| \ \geq \ \lceil \frac{n}{2} \rceil - \phi, \end{aligned} \tag{3.19}$$

and

$$|\{i : i \in \mathcal{N}, v_i \geq \widetilde{x}\}| \geq |\{i : i \in \mathcal{N}, w_{ij_0} \geq x_{j_0}[0]\}| \\ = |W_{j_0}^{+}(x_{j_0}[0])| \geq \lceil \frac{n}{2} \rceil - \phi. \tag{3.20}$$

Recall that $v_j \in X_j = \text{argmin}_{x \in \mathbb{R}} h_j(x)$. Thus, $h_i(\widetilde{x}) \geq 0$ for each $i \in \{i : i \in \mathcal{N}, v_i \leq \widetilde{x}\}$, and $h_i(\widetilde{x}) \leq 0$ for each $i \in \{i : i \in \mathcal{N}, v_i \geq \widetilde{x}\}$. Define $A(\widetilde{x}), B(\widetilde{x})$ and $C(\widetilde{x})$ as follows:

$$A(\widetilde{x}) \triangleq \{i : i \in \mathcal{N},\ h_i'(\widetilde{x}) > 0\},$$
$$B(\widetilde{x}) \triangleq \{i : i \in \mathcal{N},\ h_i'(\widetilde{x}) < 0\},$$
$$C(\widetilde{x}) \triangleq \{i : i \in \mathcal{N},\ h_i'(\widetilde{x}) = 0\}.$$

We now consider two cases: (i) $A(\widetilde{x}) = \emptyset$ or $B(\widetilde{x}) = \emptyset$, and (ii) $A(\widetilde{x}) \neq \emptyset$ and $B(\widetilde{x}) \neq \emptyset$.

Case (i): Suppose $A(\widetilde{x}) = \emptyset$ or $B(\widetilde{x}) = \emptyset$.

If $B(\widetilde{x}) = \emptyset$, then $h_i(\widetilde{x}) = 0$ for each $i \in \{i : i \in \mathcal{N}, v_i \geq \widetilde{x}\}$. Then $\widetilde{x}$ is an optimum of function

$$\frac{1}{|\{i :\ i \in \mathcal{N},\ v_i \geq \widetilde{x}\}|} \sum_{j \in \{i:\ i \in \mathcal{N},\ v_i \geq \widetilde{x}\}} h_j(x). \tag{3.21}$$

As $|\{i :\ i \in \mathcal{N},\ v_i \geq \widetilde{x}\}| \leq |\mathcal{N}|$ and by (3.19), it holds that

$$|\{i :\ i \in \mathcal{N},\ v_i \geq \widetilde{x}\}| \geq \lceil \frac{n}{2} \rceil - \phi.$$

Thus, in (3.21) at least $\lceil \frac{n}{2} \rceil - \phi$ non-faulty functions are assigned coefficients bounded below by $\frac{1}{|\mathcal{N}|}$.

Similarly, we can show the case when $A(\widetilde{x}) = \emptyset$.

Case (ii): Suppose $A(\widetilde{x}) \neq \emptyset$ and $B(\widetilde{x}) \neq \emptyset$.

When $A(\widetilde{x}) \neq \emptyset$ and $B(\widetilde{x}) \neq \emptyset$,

$$\sum_{i \in A(\widetilde{x})} h_i'(\widetilde{x}) > 0 \qquad \text{and} \qquad \sum_{i \in B(\widetilde{x})} h_i'(\widetilde{x}) < 0.$$

Then there exists $0 \leq \zeta \leq 1$ such that

$$0 = \zeta \left(\sum_{i \in A(\widetilde{x})} h_i'(\widetilde{x}) \right) + (1-\zeta) \left(\sum_{i \in B(\widetilde{x})} h_i'(\widetilde{x}) \right).$$

In addition, by definition of $C(\widetilde{x})$, we have

$$\begin{aligned} &\zeta \left(\sum_{i \in A(\widetilde{x})} h_i'(\widetilde{x}) \right) + (1-\zeta) \left(\sum_{i \in B(\widetilde{x})} h_i'(\widetilde{x}) \right) + \sum_{i \in C(\widetilde{x})} h_i'(\widetilde{x}) \\ &= 0 + \sum_{i \in C(\widetilde{x})} h_i'(\widetilde{x}) = 0 + 0 = 0. \end{aligned}$$

Thus $\widetilde{x}$ is an optimum of function

$$\chi \left(\zeta \sum_{i \in A(\widetilde{x})} h_i + (1-\zeta) \sum_{i \in B(\widetilde{x})} h_i + \sum_{i \in C(\widetilde{x})} h_i \right), \tag{3.22}$$

where

$$\chi = \frac{1}{\zeta |A(\widetilde{x})| + (1-\zeta) |B(\widetilde{x})| + |C(\widetilde{x})|}.$$

Since $0 \leq \zeta \leq 1$, either $\zeta \geq \frac{1}{2}$ or $1 - \zeta \geq \frac{1}{2}$. Without loss of generality, assume $\zeta \geq \frac{1}{2}$. We have

$$\begin{aligned} \zeta |A(\widetilde{x})| + (1-\gamma) |B(\widetilde{x})| + |C(\widetilde{x})| &\leq |A(\widetilde{x})| + |B(\widetilde{x})| + |C(\widetilde{x})| \\ &= |A(\widetilde{x}) \cup B(\widetilde{x}) \cup C(\widetilde{x})| = |\mathcal{N}|. \end{aligned}$$

In addition, since $A(\widetilde{x}) \cup C(\widetilde{x}) \supseteq \{i : i \in \mathcal{N} \text{ and } v_i \leq \widetilde{x}\}$ and $B(\widetilde{x}) \cup C(\widetilde{x}) \supseteq \{i : i \in \mathcal{N} \text{ and } v_i \geq \widetilde{x}\}$, by definition of $\widetilde{x}$, we have $|A(\widetilde{x}) \cup C(\widetilde{x})| \geq \lceil \frac{n}{2} \rceil - \phi$ and $|B(\widetilde{x}) \cup C(\widetilde{x})| \geq \lceil \frac{n}{2} \rceil - \phi$. Then in (3.22), at least $\lceil \frac{n}{2} \rceil - \phi$ non-faulty functions are assigned with weights at least $\frac{1}{2|\mathcal{N}|}$. Similar result holds when $1 - \zeta \geq \frac{1}{2}$.

Cases (i) and (ii) together prove the theorem.

□

3.7 Consensus-Based Gradient Method

In this section, we present an iterative algorithm for problem (3.2). The algorithm satisfies the requirement in (3.2) in the limit as the number of iterations $\rightarrow \infty$. The proposed iterative algorithm, named *synchronous Byzantine gradient* method (SBG), is presented below. The pseudo-code describes the steps that should be performed by each agent $j \in \mathcal{V}$. A Byzantine faulty agent may deviate from the specification arbitrarily. Algorithm SBG combines features of iterative Byzantine consensus algorithms [49, 34] with elements of gradient-based optimization [69, 67]. We will show that the SBG algorithm solves (3.2) with $\left(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}| - f\right)$–admissible weight vector $\boldsymbol{\alpha}$.

Algorithm SBG uses a trimming function Trim analogous to that used in previous Byzantine consensus algorithms.

Trim $(\mathcal{D})$

Input: Multi-set $\mathcal{D}$ of real-valued scalars, with $|\mathcal{D}| \geq 2f + 1$.

Sort the elements in $\mathcal{D}$ in a non-decreasing order (breaking ties arbitrarily), and remove the smallest f values and the largest f values. % Denote the minimum and the maximum of the remaining $|\mathcal{D}| - 2f$ values as y_s and y_l, respectively.%

Return $\frac{1}{2}(y_s + y_l)$.

Each agent j maintains a state variable x_j. We denote the value of x_j at the end of t iterations of SBG as $x_j[t]$, with the initial value being $x_j[0]$. The initial value may be chosen by each agent arbitrarily. Let $h'_j(x_j[t-1])$ denote the gradient of agent j's local cost function $h_j(\cdot)$ at $x_j[t-1]$.

The step sizes are known to all agents a priori, and satisfy the following constraints: $\lambda[t-1] \geq \lambda[t]$ for $t \geq 1$, $\sum_{t=1}^{\infty} \lambda[t-1] = \infty$ and $\sum_{t=1}^{\infty} \lambda^2[t-1] < \infty$.

As seen above, each agent j maintains minimal state (namely, x_j) across iterations. Since the Trim function is applied to the state variables and gradients separately, it is possible that the values received from different sets of agents are removed in each of those trimming operations. While the algorithm structure above resembles the prior algorithms for Byzantine consen-

Algorithm 5: SBG for agent j in iteration $t \geq 1$

1 Send the 2-tuple $(x_j[t-1], h'_j(x_j[t-1]))$ to all the other agents;

2 Receive 2-tuples from all the other agents, with the first element of each tuple being a *state variable*, and the second element being a *gradient*. If such a tuple is not received from some agent, assume a default value for the tuple. % Define: $\mathcal{D}_j^x[t-1] \triangleq$ multi-set containing $x_j[t-1]$ and state variables received from other agents; $\mathcal{D}_j^g[t-1] \triangleq$ multi-set containing $h'_j(x_j[t-1])$ and gradients received from other agents.%

3 $\widetilde{x}_j[t-1] \leftarrow \mathsf{Trim}(\mathcal{D}_j^x[t-1])$, $\widetilde{g}_j[t-1] \leftarrow \mathsf{Trim}(\mathcal{D}_j^g[t-1])$.

4 Update state as follows.

$$x_j[t] \leftarrow \widetilde{x}_j[t-1] - \lambda[t-1]\widetilde{g}_j[t-1]. \tag{3.23}$$

sus, the key difficulty in proving the desired (β, γ)-admissibility result arises due to the possibility that the Byzantine agents send different (erroneous) gradients to different non-faulty agents. Unlike the failure-free version of distributed optimization, the Byzantine faulty agents can effectively tamper with the global cost function being optimized. Thus, proving the lower bounds on β and γ requires us to show that the impact of the faulty behavior can be bounded. To delineate the impact of the faulty behavior, we now define a family $\mathcal{C}$ of "valid" global cost functions.

$$\mathcal{C} \triangleq \left\{ p : \; p = \sum_{i \in \mathcal{N}} \alpha_i h_i, \right.$$
$$\left. \text{where } \boldsymbol{\alpha} \text{ is } \left(\tfrac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}| - f\right)\text{–admissible} \right\} \tag{3.24}$$

Each $p \in \mathcal{C}$ is said to be a *valid* function (a valid global objective). Note that each $p \in \mathcal{C}$ is a convex combination of local cost functions of the non-faulty agents in $\mathcal{N}$ with $(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}| - f)$–admissible weight vector $\boldsymbol{\alpha}$.

As seen later in Lemma 5, despite the adversarial behavior of the faulty agents, it is guaranteed that the *effective gradient* $\widetilde{g}_j[t-1]$ obtained in Step 4 has an important correspondence to a *time-dependent* (i.e., varying with iteration index t) *valid* cost function in $\mathcal{C}$.

Now let us define set Y to be the union of optimal solutions for all the

valid functions in $\mathcal{C}$, i.e.,

$$Y \triangleq \bigcup_{p(\cdot)\in\mathcal{C}} \operatorname*{argmin}_{x\in\mathbb{R}} p(x). \tag{3.25}$$

Lemma 4 identifies an important property of set Y that is crucial in our convergence analysis. We will show that, as $t \to \infty$, for each agent j, its state variable $x_j[t]$ becomes trapped in set Y.

Lemma 4. *Set Y is convex and closed.*

Lemma 4 is proved in Section 3.9.3. As stated in Section 3.3, the argument x of the cost functions h_j is assumed to be a scalar in $\mathbb{R}$. In general, we would like to allow a vector argument for the cost functions (i.e., $x \in \mathbb{R}^k$, $k \geq 2$). However, set Y analogously defined for the case of vector arguments is *not necessarily convex*, making it difficult to extend our proof technique to vector arguments. In fact, the case of vector arguments remains an open problem presently.

We use the following metric for convergence analysis.

Definition 14. *For any $x \in \mathbb{R}$, the distance between x and set Y is defined as follows:*

$$Dist\,(x, Y) \triangleq \inf_{y\in Y} |x - y|\,.$$

Since Y is convex (Lemma 4), the function $Dist\,(\cdot, Y)$ is also convex. Theorem 17 states our main result, which summarizes the convergence behavior of algorithm SBG.

Theorem 17. *Algorithm SBG achieves the following properties:*
(i) Consensus: $\lim_{t\to\infty}(x_i[t] - x_j[t]) = 0$ *for all $i, j \in \mathcal{N}$, and*
(ii) Optimality: $\lim t \to \infty\ Dist\,(x_i[t], Y) = 0$ *for each $i \in \mathcal{N}$.*

Interpretation of Theorem 17:

Consensus: Property (i) in the above theorem implies consensus, since the state variables of all non-faulty agents become identical in the limit. However, property (i) does not imply that $x_j[t]$ for each $j \in \mathcal{N}$ itself has a limit. In fact, the value of the state variable $x_j[t]$ may change with t indefinitely. However, as property (i) states, in the limiting behavior, the

state variables of all the non-faulty agents change in unison, maintaining consensus. This lack of a limit for each individual $x_j[t]$ is a direct result of the simple structure of algorithm SBG, which allows a faulty agent to send different gradients to different agents. If we were to require a *Byzantine broadcast* of $(x_j[t-1], h'_j(x_j[t-1]))$ in Step 1 of algorithm SBG at each agent j, then such duplicitous behavior by faulty agents can be precluded, as we have shown elsewhere [61]. The modified algorithm has a higher cost (due to use of Byzantine broadcast); however, it can ensure that $x_j[t],\ j \in \mathcal{N}$ has a limit as $t \to \infty$, in addition to ensuring consensus. Despite the fact that the limit of $x_j[t]$ may not exist, the property (i) is useful in practice – if we were to terminate the algorithm after a sufficiently large number of iterations, then property (i) guarantees that the states of all non-faulty agents will be close to each other, thus achieving approximate consensus.

Optimality: Property (ii) in the above theorem makes guarantees about the "goodness" of the state of non-faulty agents as $t \to \infty$. In particular, observe that $Dist\,(x, Y) = 0$ if and only if $x \in Y$. Thus, property (ii) guarantees that, for sufficiently large t, state $x_j[t]$ for any non-faulty agent j approximately equals an optimum of a *valid* function in $\mathcal{C}$. That is, $x_j[t]$ approximately equals a solution of problem (3.2) with a $\left(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}|-f\right)$–admissible weight vector $\boldsymbol{\alpha}$. Thus, Theorems 12 and 17 together imply that algorithm SBG achieves *optimal fault-tolerance* in the sense that an optimal number (i.e., $|\mathcal{N}|-f$) of local cost functions of non-faulty agents are guaranteed to be represented in the global cost function that is optimized. However, as the discussion of property (i) would suggest, this global cost function is time-varying. Secondly, when $|\mathcal{N}|-f$ weights are non-zero, if the weight distribution were to be uniform, then each weight would be $\frac{1}{|\mathcal{N}|-f}$. The fact that the $\boldsymbol{\alpha}$ vector achieved by algorithm SBG is $\left(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}|-f\right)$–admissible implies that at least $|\mathcal{N}|-f$ weights are $\geq \frac{1}{2(|\mathcal{N}|-f)}$, which is within a factor of 2 of the uniform weight $\frac{1}{|\mathcal{N}|-f}$.

3.7.1 Correctness of Theorem 17

In this section, we present some key results that are useful in proving Theorem 17.

Recall that in Step 4 of algorithm SBG, each agent j applies the trimming function to compute the *effective gradient* $\widetilde{g}_j[t-1]$. Lemma 5 establishes a correspondence between this effective gradient and a valid function in $\mathcal{C}$.

Lemma 5. *For each non-faulty agent $j \in \mathcal{N}$ and each iteration $t \geq 1$, there exists a valid function $p_t^j = \sum_{i \in \mathcal{N}} b_{ji}[t]\, h_i \in \mathcal{C}$ such that the effective gradient $\widetilde{g}_j[t-1]$ computed in Step 4 of algorithm SBG can be expressed as*

$$\widetilde{g}_j[t-1] = \sum_{i \in \mathcal{N}} b_{ji}[t]\, h_i'(x_i[t-1]). \tag{3.26}$$

For agent $j \in \mathcal{N}$, the *weights* ($b_{ji}[t]$'s) in the interpolation on the right side of (3.26) correspond to the weights used to obtain a valid global cost function $p_t^j(x) \in \mathcal{C}$. Note that the vector formed by weights $b_{ji}[t]$ is $\left(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}|-f\right)$–admissible. It is also important to note that $h_i'(x_i[t-1])$ for each $i \in \mathcal{N}$ used in (3.26) is the gradient of agent i's local cost function $h_i(\cdot)$ computed at agent i's *own* state variable $x_i[t-1]$. Thus, the effective gradient $\widetilde{g}_j[t-1]$ is a linear interpolation of gradients of local cost functions at potentially *different* argument values. Despite this apparent discrepancy, algorithm SBG approximates the behavior of a gradient-based distributed optimization algorithm. Intuitively, the reason for this behavior is that the agents are guaranteed to eventually arrive at a consensus – thus, eventually, the gradients at different non-faulty agents are computed at approximately equal arguments. However, the weights $b_{ji}[t]$ in Lemma 5 are time-dependent, due to the potentially incorrect behavior by faulty agents (i.e., the weights correspond to potentially different functions in $\mathcal{C}$ in different iterations). More importantly, at different agents in $\mathcal{N}$, the weights corresponding to the effective gradients in a given iteration t can be different (i.e., for two different agents $k, j \in \mathcal{N}$, valid functions p_t^k and p_t^j may be different). Despite this difference, consensus is achieved due to the fact that set Y is convex (Lemma 4) and the decreasing step sizes $\lambda[t-1]$ used in algorithm SBG.

The next proposition will be used in proving Lemma 5.

Proposition 4. *Let $a, b, c, d \in \mathbb{R}$ such that $b < a, b \leq c \leq \frac{1}{2}(a+b), \frac{1}{2}(a+b) < a \leq d$, and there exists $0 \leq \xi \leq 1$, for which $\frac{1}{2}(a+b) = \xi c + (1-\xi)d$ holds. Then $\frac{1}{2} \leq \xi \leq 1$.*

Since $b \leq c \leq \frac{1}{2}(a+b) < a \leq d$, if the weighted average of c and d equals

$\frac{1}{2}(a+b)$, then the weight assigned to d cannot be more than $\frac{1}{2}$. The above proposition can be shown via the preceding argument. Thus the proof is omitted. Now we prove Lemma 5.

Proof of Lemma 5. Recall that in $\mathsf{Trim}(\mathcal{D}_j^g[t-1])$, the largest f values and the smallest f values were removed. Let $g_i[t-1]$ be the gradient in $\mathcal{D}_j^g[t-1]$ received from agent i at iteration t, and let $\widehat{g}_j[t-1]$ and $\check{g}_j[t-1]$ be the maximum and the minimum of the remaining $|\mathcal{D}_j^g[t-1]| - 2f = n-2f$ values. In addition, denote by $\mathcal{R}_j^2[t-1]$ the identifiers of the $n-2f$ agents from whom the remaining gradients were received.

Denote by $\mathcal{L}_j[t-1]$ and $\mathcal{S}_j[t-1]$ the set of agents from whom the largest f gradient values and the smallest f gradient values were received in iteration t. Let $i^*, j^* \in \mathcal{R}_j^2[t-1]$ such that $g_{i^*}[t-1] = \check{g}_j[t-1]$ and $g_{j^*}[t-1] = \widehat{g}_j[t-1]$. In addition, let $\phi \triangleq |\mathcal{F}|$, and let $\mathcal{L}_j^*[t-1] \subseteq \mathcal{L}_j[t-1] - \mathcal{F}$ and $\mathcal{S}_j^*[t-1] \subseteq \mathcal{S}_j[t-1] - \mathcal{F}$ such that

$$|\mathcal{L}_j^*[t-1]| = |\mathcal{S}_j^*[t-1]| = f - \phi + |\mathcal{R}_j^2[t-1] \cap \mathcal{F}|. \tag{3.27}$$

We consider two cases: (i) $\widehat{g}_j[t-1] > \check{g}_j[t-1]$ and (ii) $\widehat{g}_j[t-1] = \check{g}_j[t-1]$, separately.

Case (i): Suppose $\widehat{g}_j[t-1] > \check{g}_j[t-1]$. By definitions of $\mathcal{L}_j^*[t-1]$ and $\mathcal{S}_j^*[t-1]$, we have

$$\begin{aligned}
&\frac{1}{f-\phi+|\mathcal{R}_j^2[t-1]\cap\mathcal{F}|}\sum_{i\in\mathcal{S}_j^*[t-1]} g_i[t-1] \le \widetilde{g}_j[t-1] \\
&\le \frac{1}{f-\phi+|\mathcal{R}_j^2[t-1]\cap\mathcal{F}|}\sum_{i\in\mathcal{L}_j^*[t-1]} g_i[t-1].
\end{aligned}$$

So, there exists $0 \le \xi \le 1$ such that

$$\begin{aligned}
\widetilde{g}_j[t-1] = &\frac{\xi}{f-\phi+|\mathcal{R}_j^2[t-1]\cap\mathcal{F}|}\sum_{i\in\mathcal{S}_j^*[t-1]} g_i[t-1] \\
&+ \frac{1-\xi}{f-\phi+|\mathcal{R}_j^2[t-1]\cap\mathcal{F}|}\sum_{i\in\mathcal{L}_j^*[t-1]} g_i[t-1].
\end{aligned} \tag{3.28}$$

Without loss of generality, assume $\xi \ge \frac{1}{2}$.

Let $k \in \mathcal{R}_j^2[t-1] - \mathcal{F}$. Suppose $g_k[t-1] \le \widetilde{g}_j[t-1]$. Since $|\mathcal{L}_j[t-1] \cup$

$\{j^*\}| = f+1$, there exists a non-faulty agent $j'_k \in \mathcal{L}_j[t-1] \cup \{j^*\}$. Thus, $g_{j'_k}[t-1] \geq \widehat{g}_j[t-1] > \widetilde{g}_j[t-1]$, and there exists $0 \leq \xi_k \leq 1$ such that

$$\widetilde{g}_j[t-1] = \xi_k g_k[t-1] + (1-\xi_k) g_{j'_k}[t-1]. \tag{3.29}$$

In addition, we know $\widetilde{g}_j[t-1] = \frac{1}{2}\left(\widehat{g}_j[t-1] + \check{g}_j[t-1]\right)$. Let $a = \widehat{g}_j[t-1], b = \check{g}_j[t-1], c = g_k[t-1]$, and $d = g_{j'_k}[t-1]$. By Proposition 4, we know that $\frac{1}{2} \leq \xi_k \leq 1$.

Similarly, when $g_k[t-1] > \widetilde{g}_j[t-1]$, there also exists $\frac{1}{2} \leq \xi_k \leq 1$. In particular, since $|\mathcal{S}_j[t-1] \cup \{i^*\}| = f+1$, there exists a non-faulty agent $j'_k \in \mathcal{S}_j[t-1] \cup \{i^*\}$. Thus, $g_{j'_k}[t-1] \leq \check{g}_j[t-1] < \widetilde{g}_j[t-1]$; the existence of the desired ξ_k is implied by Proposition 4.

Since

$$\begin{aligned}
|\mathcal{N}| - f &= n - \phi - f = n - 2f + f - \phi \\
&= \left|\mathcal{R}_j^2[t-1]\right| + f - \phi \\
&= \left|\mathcal{R}_j^2[t-1] - \mathcal{F}\right| + \left|\mathcal{R}_j^2[t-1] \cap \mathcal{F}\right| + f - \phi,
\end{aligned}$$

we get

$$\begin{aligned}
&\left(|\mathcal{N}| - f\right) \widetilde{g}_j[t-1] \\
&= \left(|\mathcal{R}_j^2[t-1] - \mathcal{F}|\right) \widetilde{g}_j[t-1] \\
&\quad + \left(f - \phi + |\mathcal{R}_j^2[t-1] \cap \mathcal{F}|\right) \widetilde{g}_j[t-1] \\
&= \sum_{k \in \mathcal{R}_j^2[t-1] - \mathcal{F}} \widetilde{g}_j[t-1] \\
&\quad + \left(f - \phi + |\mathcal{R}_j^2[t-1] \cap \mathcal{F}|\right) \widetilde{g}_j[t-1] \\
&= \sum_{k \in \mathcal{R}_j^2[t-1] - \mathcal{F}} \left(\xi_k g_k[t-1] + (1-\xi_k) g_{j'_k}[t-1]\right) \quad \text{by (3.29)} \\
&\quad + \xi \sum_{i \in \mathcal{S}_j^*[t-1]} g_i[t-1] + (1-\xi) \sum_{i \in \mathcal{L}_j^*[t-1]} g_i[t-1] \quad \text{by (3.28)} \\
&= \sum_{k \in \mathcal{R}_j^2[t-1] - \mathcal{F}} \left(\xi_k\, h'_k(x_k[t-1]) + (1-\xi_k)\, h'_{j'_k}(x_{j'_k}[t-1])\right) \\
&\quad + \xi \sum_{i \in \mathcal{S}_j^*[t-1]} h'_i(x_i[t-1]) + (1-\xi) \sum_{i \in \mathcal{L}_j^*[t-1]} h'_i(x_i[t-1]),
\end{aligned} \tag{3.30}$$

where the last equality is true because for each non-faulty agent $i \in \mathcal{N}$, we

have $g_i[t-1] = h_i'(x_i[t-1])$.

Now, we define function q as follows:

$$q \triangleq \frac{1}{|\mathcal{N}|-f} \sum_{k \in \mathcal{R}_j^2[t-1]-\mathcal{F}} \left(\xi_k\, h_k + (1-\xi_k)\, h_{j_k'}\right) + \frac{\xi}{|\mathcal{N}|-f} \sum_{i \in \mathcal{S}_j^*[t-1]} h_i + \frac{1-\xi}{|\mathcal{N}|-f} \sum_{i \in \mathcal{L}_j^*[t-1]} h_i. \tag{3.31}$$

For each $k \in \mathcal{R}_j^2[t-1]-\mathcal{F}$, it holds that $\frac{\xi_k}{|\mathcal{N}|-f} \geq \frac{1}{2(|\mathcal{N}|-f)}$. For each $i \in \mathcal{S}_j^*[t-1]$, it holds that $\frac{\xi}{|\mathcal{N}|-f} \geq \frac{1}{2(|\mathcal{N}|-f)}$. In addition, we have

$$\begin{aligned}
&\left|\left(\mathcal{R}_j^2[t-1]-\mathcal{F}\right) \cup \mathcal{S}_j^*[t-1]\right| \\
&= |\mathcal{R}_j^2[t-1]| - |\mathcal{R}_j^2[t-1] \cap \mathcal{F}| + |\mathcal{S}_j^*[t-1]| \\
&= n - 2f - |\mathcal{R}_j^2[t-1] \cap \mathcal{F}| + f - \phi + |\mathcal{R}_j^2[t-1] \cap \mathcal{F}| \\
&= n - \phi - f = |\mathcal{N}| - f.
\end{aligned}$$

Thus, in (3.31), at least $|\mathcal{N}|-f$ non-faulty agents corresponding to agents $k \in \left(\mathcal{R}_j^2[t-1]-\mathcal{F}\right) \cup \mathcal{S}_j^*[t-1]$ are assigned with weights lower bounded by $\frac{1}{2(|\mathcal{N}|-f)}$.

Similarly, the case $0 \leq \xi < \frac{1}{2}$ can be proved.

Case (ii): Suppose $\widehat{g}_j[t-1] = \check{g}_j[t-1]$. Let $k \in \mathcal{R}_j^2[t-1]-\mathcal{F}$. Since $\widehat{g}_j[t-1] \geq g_k[t-1] \geq \check{g}_j[t-1]$ and $\widehat{g}_j[t-1] = \check{g}_j[t-1]$, it holds that $\widehat{g}_j[t-1] = g_k[t-1] = \check{g}_j[t-1]$. Consequently,

$$\widetilde{g}_j[t-1] = \frac{1}{2}\left(\widehat{g}_j[t-1] + \check{g}_j[t-1]\right) = g_k[t-1]. \tag{3.32}$$

Similar to (3.30), we can rewrite $\widetilde{g}_j[t-1]$ as follows:

$$\begin{aligned}
(|\mathcal{N}|-f)\, \widetilde{g}_j[t-1] = &\sum_{k \in \mathcal{R}_j^2[t-1]-\mathcal{F}} h_k'(x_k[t-1]) \\
&+ \xi \sum_{i \in \mathcal{S}_j^*[t-1]} h_i'(x_i[t-1]) + (1-\xi) \sum_{i \in \mathcal{L}_j^*[t-1]} h_i'(x_i[t-1]).
\end{aligned}$$

Define function q as follows:

$$q = \frac{1}{|\mathcal{N}| - f} \sum_{k \in \mathcal{R}_j^2[t-1] - \mathcal{F}} h_k + \frac{\xi}{|\mathcal{N}| - f} \sum_{i \in \mathcal{S}_j^*[t-1]} h_i + \frac{1 - \xi}{|\mathcal{N}| - f} \sum_{i \in \mathcal{L}_j^*[t-1]} h_i. \tag{3.33}$$

In function q defined in (3.33), for each $k \in \mathcal{R}_j^2[t-1] - \mathcal{F}$, it holds that $\frac{1}{|\mathcal{N}|-f} \geq \frac{1}{2(|\mathcal{N}|-f)}$. For each $i \in \mathcal{S}_j^*[t-1]$, it holds that $\frac{\xi}{|\mathcal{N}|-f} \geq \frac{1}{2(|\mathcal{N}|-f)}$. In addition, we have

$$\left| \left(\mathcal{R}_j^2[t-1] - \mathcal{F}\right) \cup \mathcal{S}_j^*[t-1] \right| = |\mathcal{N}| - f.$$

Thus, in (3.33), at least $|\mathcal{N}| - f$ non-faulty agents corresponding to

$$\left(\mathcal{R}_j^2[t-1] - \mathcal{F}\right) \cup \mathcal{S}_j^*[t-1]$$

are assigned with weights lower bounded by $\frac{1}{2(|\mathcal{N}|-f)}$.

Case (i) and Case (ii) together prove the lemma. □

By a similar argument as in proving Lemma 5, we can obtain the result below for $\widetilde{x}_j[t-1]$ computed in Step 4 of algorithm SBG.

Corollary 4. *For each non-faulty agent $j \in \mathcal{N}$ and $t \geq 1$, there exists a $\left(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}| - f\right)$–admissible weight vector $\mathbf{a}_j[t]$ (whose i-th element is $a_{ji}[t]$) such that $\widetilde{x}_j[t-1]$ computed in Step 4 of algorithm SBG can be expressed as*

$$\widetilde{x}_j[t-1] = \sum_{i \in \mathcal{N}} a_{ji}[t]\, x_i[t-1]. \tag{3.34}$$

Note that weights $b_{ji}[t]$ and $a_{ji}[t]$ in (3.26) and (3.34), respectively, are not necessarily identical, because the state variables and gradients are trimmed independently in Step 4 of algorithm SBG.

Asymptotic Consensus

Recall that $\lambda[t] \leq \lambda[t-1]$ and $\lim_{t\to\infty} \lambda[t] = 0$. The following proposition is used in proving consensus.

Proposition 5. *Let* $0 \leq b < 1$. *Define* $\ell(t) = \sum_{r=0}^{t-1} \lambda[r] b^{t-r}$. *Then* $\lim_{t\to\infty} \ell(t) = 0$. *Additionally, if* $\lambda[t] = \frac{1}{t}$ *for* $t \geq 1$ *and* $\lambda[0] = 1$ [3], *then* $\ell(t) = O(\frac{1}{t})$.

The results in Proposition 5 is very standard and well-known. The proof of Proposition 5 is presented in Section 3.9.4 for completeness.

Denote $M(t) \triangleq \max_{i\in\mathcal{N}} x_i[t]$ and $m(t) \triangleq \min_{i\in\mathcal{N}} x_i[t]$. Asymptotic consensus among non-faulty agents in Theorem 17 is immediately implied by the following lemma.

Lemma 6. *Under algorithm SBG,* $\lim_{t\to\infty} (M[t] - m[t]) = 0$. *Additionally, if* $\lambda[t] = \frac{1}{t}$ *for* $t \geq 1$ *and* $\lambda[0] = 1$, *then* $(M[t] - m[t]) = O(\frac{1}{t})$.

Proof. Let $i, j \in \mathcal{N}$ such that $x_i[t] = M[t]$, and $x_j[t] = m[t]$. For $t \geq 1$, by (3.23), we have

$$\begin{aligned} M[t] - m[t] &= x_i[t] - x_j[t] \\ &= (\widetilde{x}_i[t-1] - \widetilde{x}_j[t-1]) \\ &\quad + \lambda[t-1]\left(\widetilde{g}_j[t-1] - \widetilde{g}_i[t-1]\right). \end{aligned} \tag{3.35}$$

We bound the first term in the right hand side of (3.35) as follows. The second term can be bounded similarly.

By Corollary 4, we have

$$\begin{aligned} &\widetilde{x}_i[t-1] - \widetilde{x}_j[t-1] \\ &= \sum_{k\in\mathcal{N}} a_{ik}[t] x_k[t-1] - \sum_{k\in\mathcal{N}} a_{jk}[t] x_k[t-1]. \end{aligned} \tag{3.36}$$

Define $\mathcal{K}_i$ and $\mathcal{K}_j$ as follows:

$$\begin{aligned} \mathcal{K}_i &\triangleq \left\{k \in \mathcal{N} :\ a_{ik}[t] \geq \frac{1}{2\left(|\mathcal{N}| - f\right)}\right\}, \text{and} \\ \mathcal{K}_j &\triangleq \left\{k \in \mathcal{N} :\ a_{jk}[t] \geq \frac{1}{2\left(|\mathcal{N}| - f\right)}\right\}. \end{aligned} \tag{3.37}$$

By Corollary 4, both $\mathbf{a}_i[t]$ and $\mathbf{a}_j[t]$ are $\left(\frac{1}{2(|\mathcal{N}|-f)},\ |\mathcal{N}| - f\right)$–admissible. Thus,

[3] As it can be seen from the proof of Proposition 5, $\lambda[0]$ can be chosen to be any positive constant.

$|\mathcal{K}_i| \geq |\mathcal{N}| - f$ and $|\mathcal{K}_j| \geq |\mathcal{N}| - f$. Consequently,

$$\begin{aligned}|\mathcal{K}_i \cap \mathcal{K}_j| &\geq |\mathcal{N}| - f + |\mathcal{N}| - f - |\mathcal{N}| \\ &= |\mathcal{N}| - 2f \geq 2f + 1 - 2f = 1.\end{aligned}$$

Let $k^* \in \mathcal{K}_i \cap \mathcal{K}_j$. We can bound (3.36) as follows:

$$\begin{aligned}&\widetilde{x}_i[t-1] - \widetilde{x}_j[t-1] \\ &= \sum_{k\in\mathcal{N}} a_{ik}[t]x_k[t-1] - \sum_{k\in\mathcal{N}} a_{jk}[t]x_k[t-1] \\ &= \sum_{k\in\mathcal{N}, k\neq k^*} a_{ik}[t]x_k[t-1] - \sum_{k\in\mathcal{N}, k\neq k^*} a_{jk}[t]x_k[t-1] \\ &\quad + \left(a_{ik^*}[t] - \min\{a_{ik^*}[t], a_{jk^*}[t]\}\right) x_{k^*}[t-1] \\ &\quad - \left(a_{jk^*}[t] - \min\{a_{ik^*}[t], a_{jk^*}[t]\}\right) x_{k^*}[t-1] \\ &\leq \sum_{k\in\mathcal{N}, k\neq k^*} a_{ik}[t]M[t-1] - \sum_{k\in\mathcal{N}, k\neq k^*} a_{jk}[t]m[t-1] \\ &\quad + \left(a_{ik^*}[t] - \min\{a_{ik^*}[t], a_{jk^*}[t]\}\right) M[t-1] \\ &\quad - \left(a_{jk^*}[t] - \min\{a_{ik^*}[t], a_{jk^*}[t]\}\right) m[t-1] \\ &\leq \left(1 - \min\{a_{ik^*}[t], a_{jk^*}[t]\}\right)\left(M[t-1] - m[t-1]\right) \\ &\leq \left(1 - \frac{1}{2\left(|\mathcal{N}| - f\right)}\right)\left(M[t-1] - m[t-1]\right), \qquad (3.38)\end{aligned}$$

where the last inequality follows from the fact that $k^* \in \mathcal{K}_i \cap \mathcal{K}_j$ and $\min\{a_{ik^*}[t], a_{jk^*}[t]\} \geq \frac{1}{2(|\mathcal{N}|-f)}$.

Since $|h_k'(x)| \leq L$ for any x and $k \in \mathcal{N}$, it follows that

$$\max_{k\in\mathcal{N}} \max_{x\in\mathbb{R}} h_k'(x) \leq L, \quad \text{and} \quad \min_{k\in\mathcal{N}} \min_{x\in\mathbb{R}} h_k'(x) \geq -L.$$

Similar to (3.38), we get

$$\widetilde{g}_j[t-1] - \widetilde{g}_i[t-1] \leq 2L\left(1 - \frac{1}{2\left(|\mathcal{N}| - f\right)}\right). \qquad (3.39)$$

By (3.35), (3.38) and (3.39), we get

$$\begin{aligned} M[t] - m[t] &\leq \left(1 - \frac{1}{2\left(|\mathcal{N}| - f\right)}\right)\left(M[t-1] - m[t-1]\right) \\ &\quad + 2L\lambda[t-1]\left(1 - \frac{1}{2\left(|\mathcal{N}| - f\right)}\right) \\ &\leq \left(1 - \frac{1}{2\left(|\mathcal{N}| - f\right)}\right)^{t}\left(M[0] - m[0]\right) \\ &\quad + 2L\sum_{r=0}^{t-1}\lambda[r]\left(1 - \frac{1}{2\left(|\mathcal{N}| - f\right)}\right)^{t-r}. \end{aligned} \tag{3.40}$$

Therefore, by Proposition 5, we conclude that

$$\lim_{t\to\infty}\left(M[t] - m[t]\right) \;=\; 0.$$

The first term on the right-hand side of (3.40) goes to 0 exponentially fast.

For the special step sizes $\lambda[t] = \frac{1}{t}$ for $t \geq 1$ and $\lambda[0] = 1$, the second term on the right-hand side of (3.40) goes to 0 sublinearly. In particular, $M[t] - m[t] = O(\frac{1}{t})$ due to Proposition 5. □

Our next lemma (Lemma 7) says that the cumulative disagreement between two non-faulty agents is finite. Lemma 7 is proved in Section 3.9.5.

Lemma 7. *Under algorithm SBG, it holds that* $\sum_{t=0}^{\infty}\lambda[t]\left(M[t] - m[t]\right) < \infty$.

The above results establish asymptotic consensus behavior of SBG.

Convergence Analysis

In this section, we prove that $x_j[t]$, where $j \in \mathcal{N}$, is asymptotically in Y, i.e., $\lim_{t\to\infty} D\left(x_j[t], Y\right) = 0$.

Recall that lemma 6 says that $\lim_{t\to\infty}\left(x_j[t] - x_i[t]\right) = 0$ for any $j, i \in \mathcal{N}$ – asymptotic consensus among the non-faulty agents is achieved. Thus, for sufficiently large t, (3.23) approximately equals

$$x_j[t] \;\approx\; x_j[t-1] - \lambda[t-1]p_t^j\left(x_j[t-1]\right), \forall\, j \in \mathcal{N}, \tag{3.41}$$

where $p_t^j(\cdot)$ is the valid function identified in Lemma 5. (This approximation is presented and proved formally later in this section.) For a non-faulty

agent j, one typical trajectory of x_j is as follows: (1) x_j first approaches set Y (defined in (3.25)), and (2) then bounces back and forth around set Y before completely being trapped in Y.

The existence of disagreement in finite time (discussed above) complicates the convergence analysis significantly.

To show the optimality of x_j for $j \in \mathcal{N}$, we use the auxiliary sequence $\{z[t]\}_{t=0}^{\infty}$ defined as follows.

Definition 15. *Let $\{z[t]\}_{t=0}^{\infty}$ be an estimate sequence such that*

$$z[t] = x_{j_t}[t], \quad \textit{where } j_t \in \underset{j \in \mathcal{N}}{\operatorname{argmax}} \, Dist\left(x_j[t], Y\right). \tag{3.42}$$

To show $\lim_{t\to\infty} Dist\left(x_j[t], Y\right) = 0$, $\forall\, j \in \mathcal{N}$, it is enough to show

$$\lim_{t\to\infty} Dist\left(z[t], Y\right) = 0,$$

observing that $0 \leq Dist\left(x_j[t], Y\right) \leq Dist\left(z[t], Y\right)$ for each $j \in \mathcal{N}$.

Proposition 6. *If $\lim_{t\to\infty} Dist\left(z[t], Y\right) = 0$, then for each non-faulty agent i in $\mathcal{N}$, the sequence $\{Dist\left(x_i[t], Y\right)\}_{t=0}^{\infty}$ converges and*

$$\lim_{t\to\infty} Dist\left(x_i[t], Y\right) = 0.$$

Proposition 6 is proved in Section 3.9.6.

Our convergence analysis of $Dist\left(z[t], Y\right)$ uses the notion of resilient points, stated next.

Definition 16. *Given sequences $\{x[t]\}_{t=0}^{\infty}$ and $\{g[t]\}_{t=0}^{\infty}$, and a set of stepsizes $\{\lambda[t]\}_{t=0}^{\infty}$ we say $x[t]$ is a resilient point with respect to gradient $g[t]$ if one of the following items is true:*

* *$x[t] \in Y$ and $(x[t] - \lambda[t]g[t]) \notin Y$,*
* *$x[t] > \max Y$ and $(x[t] - \lambda[t]g[t]) < \min Y$,*
* *$x[t] < \min Y$ and $(x[t] - \lambda[t]g[t]) > \max Y$.*

Lemma 8. *The sequence $\{Dist\left(z[t], Y\right)\}_{t=0}^{\infty}$ converges.*

The following auxiliary lemmas and proposition are used in proving Lemma 8. Their proofs can be found in Section 3.9.6.

Corollary 5. *Under SBG,* $\lambda[t]\,(M[t]-m[t]) \to 0$ *as* $t \to \infty$*, and*

$$\sum_{\tau=t}^{\infty} \lambda[\tau]\,(M[\tau]-m[\tau]) \to 0, \qquad \text{as } t \to \infty.$$

To show $\{Dist\,(z[t], Y)\}_{t=0}^{\infty}$ is convergent, we will use the well-known "almost supermartingale" convergence theorem in [70], which can also be found as Lemma 11 in Section 2.2 [71]. We present a deterministic version of the theorem in the next lemma.

Lemma 9. *[70] Let* $\{a_t\}_{t=0}^{\infty}, \{b_t\}_{t=0}^{\infty}$*, and* $\{c_t\}_{t=0}^{\infty}$ *be non-negative sequences. Suppose that*

$$a_{t+1} \le a_t - b_t + c_t \qquad \text{for all } t \ge 0,$$

and $\sum_{t=0}^{\infty} c_t < \infty$*. Then* $\sum_{t=0}^{\infty} b_t < \infty$ *and the sequence* $\{a_t\}_{t=0}^{\infty}$ *converges to a non-negative value.*

With these auxiliary lemmas and proposition, Lemma 8 can be proved as follows.

Proof of Lemma 8. Recall that $\{z[t]\}_{t=0}^{\infty}$ is a sequence of estimates defined in Definition 15, and that there is a sequence of agents $\{j_t\}_{t=0}^{\infty}$ associated with the sequence $\{z[t]\}_{t=0}^{\infty}$. We first derive an iterative relation for $Dist\,(z[t], Y)$.

For $t \ge 0$, define

$$j'_{t+1} \in \underset{i\in\mathcal{N}}{\operatorname{argmax}}\, Dist\left(x_i[t] - \lambda[t]\widetilde{g}_{j_{t+1}}[t],\ Y\right). \tag{3.43}$$

We get

$$\begin{aligned}
&Dist\,(z[t+1], Y) = Dist\left(x_{j_{t+1}}[t+1], Y\right)\\
&= Dist\left(\widetilde{x}_{j_{t+1}}[t] - \lambda[t]\widetilde{g}_{j_{t+1}}[t],\ Y\right)\\
&\overset{(a)}{=} Dist\left(\sum_{i\in\mathcal{N}} a_{j_{t+1}i}[t+1]x_i[t] - \lambda[t]\widetilde{g}_{j_{t+1}}[t],\ Y\right)\\
&= Dist\left(\sum_{i\in\mathcal{N}} a_{j_{t+1}i}[t+1]\left(x_i[t] - \lambda[t]\widetilde{g}_{j_{t+1}}[t]\right),\ Y\right)\\
&\overset{(b)}{\le} \sum_{i\in\mathcal{N}} a_{j_{t+1}i}[t+1]\, Dist\left(x_i[t] - \lambda[t]\widetilde{g}_{j_{t+1}}[t],\ Y\right)\\
&\le \max_{i\in\mathcal{N}} Dist\left(x_i[t] - \lambda[t]\widetilde{g}_{j_{t+1}}[t],\ Y\right),
\end{aligned} \tag{3.44}$$

where equality (a) follows from Corollary 4; and inequality (b) is true because of the convexity of $Dist\,(\cdot, Y)$. By (3.43) and (3.44), we get

$$\begin{aligned}
Dist\,(z[t+1], Y) &\leq Dist\left(x_{j'_{t+1}}[t] - \lambda[t]\widetilde{g}_{j_{t+1}}[t],\ Y\right) \\
&\overset{(a)}{=} Dist\left(x_{j'_{t+1}}[t] - \lambda[t]\sum_{k\in\mathcal{N}} b_{j_{t+1}k}[t+1]h'_k(x_k[t]),\ Y\right) \\
&= \inf_{y\in Y}\left|x_{j'_{t+1}}[t] - \lambda[t]\sum_{k\in\mathcal{N}} b_{j_{t+1}k}[t+1]h'_k(x_k[t]) - y\right| \\
&= \inf_{y\in Y}\Big|x_{j'_{t+1}}[t] - \lambda[t]\sum_{k\in\mathcal{N}} b_{j_{t+1}k}[t+1]h'_k(x_{j'_{t+1}}[t]) - y \\
&+ \lambda[t]\sum_{k\in\mathcal{N}} b_{j_{t+1}k}[t+1]\left(h'_k(x_{j'_{t+1}}[t]) - h'_k(x_k[t])\right)\Big| \\
&\leq \inf_{y\in Y}\left|x_{j'_{t+1}}[t] - \lambda[t]p'_{t+1}(x_{j'_{t+1}}[t]) - y\right| \\
&+ \lambda[t]L(M[t] - m[t]),
\end{aligned} \tag{3.45}$$

where equality (a) holds due to Lemma 5; and the last inequality follows from the fact that $h'_k(\cdot)$ is L–Lipschitz continuous for each $k \in \mathcal{N}$ and

$$|x_{j'_{t+1}}[t] - x_k[t]| \leq M[t] - m[t]$$

– recalling that $p_{t+1} \triangleq p_{t+1}^{j_{t+1}}$, which is a valid global objective defined in (3.24) and Lemma 5.

Recall that j'_{t+1} is defined as (3.43). Note that for each $t \geq 0$, there exists a non-faulty agent j'_{t+1} such that (3.45) holds, and there exists a sequence of agents $\{j'_{t+1}\}_{t=0}^{\infty}$. Let $\{x[t]\}_{t=0}^{\infty}$ be a sequence of estimates such that

$$x[t] = x_{j'_{t+1}}[t]. \tag{3.46}$$

Let $\{g[t]\}_{t=0}^{\infty}$ be a sequence of gradients such that

$$g[t] = p'_t(x_{j'_{t+1}}[t]). \tag{3.47}$$

To get an iterative relation of $Dist\,(z[t], Y)$, we consider two cases: Case 1: $x[t] = x_{j'_{t+1}}[t]$ is a resilient point with respect to the gradient $g[t] = p'_t(x_{j'_{t+1}}[t])$, and Case 2: $x[t] = x_{j'_{t+1}}[t]$ is a not resilient point with respect to the gradient $g[t] = p'_t(x_{j'_{t+1}}[t])$.

Case 1: Suppose $x[t] = x_{j'_{t+1}}[t]$ is a resilient point with respect to the gradient $g[t] = p'_t(x_{j'_{t+1}}[t])$, where resilient points are defined in Definition 16. By Definition 16, we can bound (3.45) further as follows:

$$\begin{aligned} Dist\left(z[t+1], Y\right) &\leq \inf_{y \in Y} \left| x_{j'_{t+1}}[t] - \lambda[t] p'_{t+1}(x_{j'_{t+1}}[t]) - y \right| \\ &\quad + L\lambda[t]\left(M[t] - m[t]\right) \quad \text{by (3.45)} \\ &\leq L\lambda[t] + L\lambda[t]\left(M[t] - m[t]\right), \end{aligned} \tag{3.48}$$

where the last inequality holds because $x[t]$ is a resilient point and the absolute value of the gradient of the valid function $p_t(\cdot)$ is bounded above by L.

Case 2: Suppose $x[t] = x_{j'_{t+1}}[t]$ is *not* a resilient point with respect to the gradient $g[t] = p'_t(x_{j'_{t+1}}[t])$. Then from Definition 16, we know that
$B1$: if $x_{j'_{t+1}}[t] \in Y$, then $x_{j'_{t+1}}[t] - \lambda[t] p'_t(x_{j'_{t+1}}[t]) \in Y$,
$B2$: if $x_{j'_{t+1}}[t] < \min Y$, then $x_{j'_{t+1}}[t] - \lambda[t] p'_t(x_{j'_{t+1}}[t]) \leq \max Y$,
$B3$: if $x_{j'_{t+1}}[t] > \max Y$, then $x_{j'_{t+1}}[t] - \lambda[t] p'_t(x_{j'_{t+1}}[t]) \geq \min Y$.

Note that it does not impact the analysis at all whether $x[t+1]$ is resilient or not.

We consider two subcases:
Subcase 1: $x[t] - \lambda[t] g[t] = x_{j'_{t+1}}[t] - \lambda[t] p'_t(x_{j'_{t+1}}[t]) \in Y$;
Subcase 2: $x[t] - \lambda[t] g[t] = x_{j'_{t+1}}[t] - \lambda[t] p'_t(x_{j'_{t+1}}[t]) \notin Y$.

Subcase 1 can possibly appear in each of $B1, B2$, and $B3$. In contrast, Subcase 2 can only appear in $B2$ and $B3$.

Subcase 1: Suppose $x[t] - \lambda[t] g[t] = x_{j'_{t+1}}[t] - \lambda[t] p'_t(x_{j'_{t+1}}[t]) \in Y$. Then it holds that

$$\begin{aligned} &Dist\left(z[t+1], Y\right) \\ &\leq \inf_{y \in Y} \left| x_{j'_{t+1}}[t] - \lambda[t] p'_t(x_{j'_{t+1}}[t]) - y \right| + L\lambda[t]\left(M[t] - m[t]\right) \quad \text{as per (3.45)} \\ &\leq 0 + L\lambda[t]\left(M[t] - m[t]\right) \qquad (3.49) \\ &\leq Dist\left(z[t], Y\right) + L\lambda[t]\left(M[t] - m[t]\right). \qquad (3.50) \end{aligned}$$

Subcase 2: Suppose $x[t] - \lambda[t] g[t] = x_{j'_{t+1}}[t] - \lambda[t] p'_t(x_{j'_{t+1}}[t]) \notin Y =$

$[\min Y, \max Y]$. As commented earlier, either $B2$ holds or $B3$ holds. In addition, from the assumption of Subcase 2, $B2$ and $B3$ can be further refined as follows.

$B2'$: if $x_{j'_{t+1}}[t] < \min Y$, then $x_{j'_{t+1}}[t] - \lambda[t]p'_t(x_{j'_{t+1}}[t]) < \min Y$,

$B3'$: if $x_{j'_{t+1}}[t] > \max Y$, then $x_{j'_{t+1}}[t] - \lambda[t]p'_t(x_{j'_{t+1}}[t]) > \max Y$.

Suppose $B2'$ is true. As $x_{j'_{t+1}}[t] < \min Y$, and $p'_t(x_{j'_{t+1}}[t])$ is the gradient of the valid function $p_t(\cdot)$ at point $x_{j'_{t+1}}[t]$, from the definition of set Y, we know that

$$p'_t(x_{j'_{t+1}}[t]) < 0. \tag{3.51}$$

In addition, since $x_{j'_{t+1}}[t] - \lambda[t]p'_t(x_{j'_{t+1}}[t]) < \min Y$, it holds that for any $y \in Y$

$$\begin{aligned}
&\left|x_{j'_{t+1}}[t] - \lambda[t]p'_t(x_{j'_{t+1}}[t]) - y\right| \\
&= y - x_{j'_{t+1}}[t] + \lambda[t]p'_t(x_{j'_{t+1}}[t]) \\
&= \left|y - x_{j'_{t+1}}[t]\right| + \lambda[t]p'_t(x_{j'_{t+1}}[t]) \\
&= \left|y - x_{j'_{t+1}}[t]\right| - \lambda[t]\left|p'_t(x_{j'_{t+1}}[t])\right| \quad \text{by (3.51)}
\end{aligned} \tag{3.52}$$

Similarly, we can show that (3.52) still holds for the case when $B3'$ is true. Henceforth, we refer to (3.52) as the relation that holds for both $B2'$ and $B3'$, i.e., holds under Subcase 2.

Thus, under Subcase 2, we can bound (3.45) as follows:

$$\begin{aligned}
Dist\left(z[t+1], Y\right) &\leq \inf_{y \in Y} \left| x_{j'_{t+1}}[t] - \lambda[t] p'_t(x_{j'_{t+1}}[t]) - y \right| \\
&\quad + L\lambda[t]\left(M[t] - m[t]\right) \quad \text{as per (3.45)} \\
&= \inf_{y \in Y} \left| y - x_{j'_{t+1}}[t] \right| - \lambda[t] \left| p'_t(x_{j'_{t+1}}[t]) \right| \\
&\quad + L\lambda[t]\left(M[t] - m[t]\right) \quad \text{by (3.52)} \\
&= Dist\left(x_{j'_{t+1}}[t], Y\right) - \lambda[t] \left| p'_t(x_{j'_{t+1}}[t]) \right| \\
&\quad + L\lambda[t]\left(M[t] - m[t]\right) \quad \text{by Definition 14} \\
&\leq Dist\left(z[t], Y\right) - \lambda[t] \left| p'_t(x_{j'_{t+1}}[t]) \right| \\
&\quad + L\lambda[t]\left(M[t] - m[t]\right) \quad \text{by Definition 15} \qquad (3.53) \\
&\leq Dist\left(z[t], Y\right) + L\lambda[t]\left(M[t] - m[t]\right). \qquad (3.54)
\end{aligned}$$

Combining the above analysis for Subcase 1 and Subcase 2, by (3.50) and (3.54), for Case 2, we obtain the following iteration relation:

$$Dist\left(z[t+1], Y\right) \leq Dist\left(z[t], Y\right) + L\lambda[t]\left(M[t] - m[t]\right). \qquad (3.55)$$

Therefore, for both Case 1 and Case 2, by (3.48) and (3.55), we obtain the following iterative relation:

$$\begin{aligned}
Dist\left(z[t+1], Y\right) \leq \max\left\{\lambda[t]L,\ Dist\left(z[t], Y\right)\right\} \\
+ L\lambda[t]\left(M[t] - m[t]\right). \qquad (3.56)
\end{aligned}$$

With this iterative relation, we next show that $Dist\left(z[t], Y\right)$ is convergent. Recall from (4.41) and (3.47) that $x[t] = x_{j'_{t+1}}[t]$ and $g[t] = p'_t(x_{j'_{t+1}}[t])$. We consider two cases, separately: Case (i) where there are infinitely many points in $\{x[t]\}_{t=0}^{\infty}$ that are resilient with respect to $\{g[t]\}_{t=0}^{\infty}$, and Case (ii) where there are finitely many points in $\{x[t]\}_{t=0}^{\infty}$ that are resilient with respect to $\{g[t]\}_{t=0}^{\infty}$.

Case (i): Suppose there are infinitely many points in $\{x[t]\}_{t=0}^{\infty}$ that are resilient with respect to $\{g[t]\}_{t=0}^{\infty}$.

Let $\{t_i\}_{i=0}^{\infty}$ be the maximal sequence of such indices. Since $x[t_i]$ is a resilient

point with respect to $g[t_i]$ for each i, then for each t_i, by (3.48), we get

$$Dist\left(z[t_i+1], Y\right) \leq \lambda[t_i]L + \lambda[t_i]L\left(M[t_i] - m[t_i]\right), \tag{3.57}$$

and for each $t \neq t_i \, \forall i$, by (3.55), we get

$$Dist\left(z[t+1], Y\right) \leq Dist\left(z[t], Y\right) + \lambda[t]L\left(M[t] - m[t]\right). \tag{3.58}$$

Taking limit sup on both sides of (3.57), we get

$$\begin{aligned}
&\limsup_{i\to\infty} \; Dist\left(z[t_i+1], Y\right) \\
&\leq \limsup_{i\to\infty} \lambda[t_i]L + \limsup_{i\to\infty} \lambda[t_i]L\left(M[t_i] - m[t_i]\right) \\
&= \; 0 + 0 = 0 \text{ by Corollary 5.}
\end{aligned} \tag{3.59}$$

In addition, because distance is non-negative, it is true that

$$\liminf_{i\to\infty} \; Dist\left(z[t_i+1], Y\right) \geq 0.$$

Thus, the limit of $Dist\left(z[t_i+1], Y\right)$ exists, and

$$\lim_{i\to\infty} \; Dist\left(z[t_i+1], Y\right) = 0. \tag{3.60}$$

For each $\tau > t_0$ and $\tau \notin \{t_i\}_{i=0}^{\infty}$, there exists $t_{i(\tau)}$ such that $t_{i(\tau)} < \tau < t_{i(\tau)+1}$. Repeatedly applying (3.58), we get

$$\begin{aligned}
&Dist\left(z[\tau+1], Y\right) \\
&\leq \; Dist\left(z[t_{i(\tau)}+1], Y\right) + \sum_{r=t_{i(\tau)}+1}^{\tau} \lambda[r]L\left(M[r] - m[r]\right) \\
&\leq \lambda[t_{i(\tau)}]L + \lambda[t_{i(\tau)}]\left(M[t_{i(\tau)}] - m[t_{i(\tau)}]\right)L \\
&\quad + \sum_{r=t_{i(\tau)}+1}^{\tau} \lambda[r]\left(M[r] - m[r]\right)L \quad \text{by (3.57)} \\
&= \lambda[t_{i(\tau)}]L + \sum_{r=t_{i(\tau)}}^{\tau} \lambda[r]\left(M[r] - m[r]\right)L \\
&\leq \lambda[t_{i(\tau)}]L + \sum_{r=t_{i(\tau)}}^{\infty} \lambda[r]\left(M[r] - m[r]\right)L.
\end{aligned} \tag{3.61}$$

Taking limit sup on both sides of (3.61), we get

$$\begin{aligned}
&\limsup_{\tau\to\infty} Dist\left(z[\tau+1], Y\right) \\
&\leq \lim_{\tau\to\infty} \lambda[t_{i(\tau)}]L + \lim_{\tau\to\infty} \sum_{r=t_{i(\tau)}}^{\infty} \lambda[r]\left(M[r]-m[r]\right)L \\
&= 0+0=0 \quad \text{by Corollary 5.}
\end{aligned} \tag{3.62}$$

To apply Corollary 5 here we need to show that $t_{i(\tau)} \to \infty$ as $\tau \to \infty$. This is true since there are infinitely many resilient points.

From (3.60), we know that $\forall\, \epsilon > 0, \exists\, i_0$ such that for all $j \geq i_0$, the following holds:

$$\begin{aligned}
&\sup\{Dist\left(z[t_j+1], Y\right),\ t_j \in \{t_i\}_{i=0}^{\infty},\ j \geq i_0\} \\
&= |\sup\{Dist\left(z[t_j+1], Y\right),\ t_j \in \{t_i\}_{i=0}^{\infty},\ j \geq i_0\} - 0| \\
&< \epsilon.
\end{aligned} \tag{3.63}$$

From (3.62), we know that $\forall\, \epsilon > 0, \exists\, \tau^*, \tau^* \notin \{t_i\}_{i=0}^{\infty}$ such that for all $\tau \geq \tau^*, \tau \notin \{t_i\}_{i=0}^{\infty}$, the following holds:

$$\begin{aligned}
&\sup\{Dist\left(z[\tau+1], Y\right),\ \tau \geq \tau^*, \tau \notin \{t_i\}_{i=0}^{\infty}\} \\
&= |\sup\{Dist\left(z[\tau+1], Y\right),\ \tau \geq \tau^*, \tau \notin \{t_i\}_{i=0}^{\infty}\} - 0| \\
&< \epsilon.
\end{aligned} \tag{3.64}$$

Let $t^* = \max\{t_{i_0}, \tau^*\}$. Then for $\epsilon > 0$ and $t \geq t^*$, we have

$$\begin{aligned}
&\sup\{Dist\left(z[t+1], Y\right),\ t \geq t^*\} \\
&\leq \sup\Big\{\{Dist\left(z[t+1], Y\right),\ t \in \{t_i\}_{i=0}^{\infty}, t \geq t_{i_0}\} \\
&\quad \cup \{Dist\left(z[t+1], Y\right),\ t \notin \{t_i\}_{i=0}^{\infty}, t \geq \tau^*\}\Big\} \\
&= \max\Big\{\sup\{Dist\left(z[t+1], Y\right),\ t \in \{t_i\}_{i=0}^{\infty}, t \geq t_{i_0}\}, \\
&\quad \sup\{Dist\left(z[t+1], Y\right),\ t \notin \{t_i\}_{i=0}^{\infty}, t \geq \tau^*\}\Big\} \\
&< \max\{\epsilon, \epsilon\} = \epsilon \quad \text{by (3.63) and (3.64).}
\end{aligned}$$

Thus, we have

$$\limsup_{t\to\infty} Dist\left(z[t], Y\right) = \limsup_{t\to\infty} Dist\left(z[t+1], Y\right) = 0.$$

Therefore, the limit of $Dist\,(z[t], Y)$ exists, and

$$\lim_{t\to\infty} Dist\,(z[t], Y) = 0. \tag{3.65}$$

Case (ii): Suppose there are finitely many points in $\{x[t]\}_{t=0}^{\infty}$ that are resilient with respect to $\{g[t]\}_{t=0}^{\infty}$.

By the assumption in case (ii), we know that there exists a time index m_0 such that for all $t \geq m_0$, each $x[t]$ is not a resilient point with respect to $g[t]$. Thus, for $t \geq m_0$, (3.55) is applicable. Thus,

$$Dist\,(z[t+1], Y) \leq\ Dist\,(z[t], Y) + \lambda[t] L\,(M[t] - m[t])\,. \tag{3.66}$$

Define $\{a_r\}_{r=0}^{\infty}$, $\{b_r\}_{r=0}^{\infty}$, and $\{c_r\}_{r=0}^{\infty}$ as follows:

$$\begin{aligned} a_r &= Dist\,(z[m_0 + r], Y)\,, \\ b_r &= 0, \\ c_r &= \lambda[m_0 + r] L\,(M[m_0 + r] - m[m_0 + r])\,. \end{aligned}$$

By Lemmas 7 and 9, we know the limit of $Dist\,(z[t], Y)$ exists. Let $c \geq 0$ be a nonnegative constant such that

$$\lim_{t\to\infty} Dist\,(z[t], Y) = c. \tag{3.67}$$

By (3.66), for each $t \geq m_0$, we have

$$\begin{aligned} Dist\,(z[t+1], Y) &\leq\ Dist\,(z[t], Y) + \lambda[t] L\,(M[t] - m[t]) \\ &\leq\ Dist\,(z[m_0], Y) + \sum_{r=m_0}^{t} \lambda[r] L\,(M[r] - m[r]) \\ &\leq\ Dist\,(z[m_0], Y) + \sum_{r=m_0}^{\infty} \lambda[r] L\,(M[r] - m[r])\,. \end{aligned} \tag{3.68}$$

By Lemma 7, we know there exists some constant C such that

$$\sum_{r=m_0}^{\infty} \lambda[r] L\,(M[r] - m[r]) \leq \sum_{r=0}^{\infty} \lambda[r] L\,(M[r] - m[r]) \leq C.$$

In addition, $Dist\,(z[m_0], Y) \in \mathbb{R}$. Thus, by (3.68), we know for $Dist\,(z[t+1], Y)$

is bounded for each $t \geq m_0$. In particular, for each $t \geq m_0$,

$$Dist\,(z[t+1], Y) \leq Dist\,(z[m_0], Y) + C \ \in \ \mathbb{R}.$$

Thus, c, defined in (3.67), is finite, i.e., $c < \infty$. □

The convexity of Y is crucial in establishing (3.56). Up to the disagreement adjustment term (i.e., $L\lambda[t]\,(M[t] - m[t])$ in (3.56)), the iterative relation (3.56) is analogous to the basic evolution in the standard centralized gradient-based method [69]. Although the basic evolution relation has been obtained for failure-free distributed algorithms [3, 4, 5, 8], since our effective objective function is time-dependent ($p_t^j(\cdot)$ in (3.41)), their analysis does not apply to our problem.

Furthermore, we can show that the distance becomes 0 asymptotically, as stated in Lemma 10.

Lemma 10. *The sequence* $\{Dist\,(z[t], Y)\}_{t=0}^{\infty}$ *converges to* 0*, i.e.,*

$$\lim_{t\to\infty} Dist\,(z[t], Y) = 0.$$

Lemma 10 is proved by contradiction. The following proposition is used in our proof. It says that if $Dist\,(z[t], Y)$ does not converge to 0, then there exists at least one cumulative point lies outside set Y.

Proposition 7. *If there exists* $c > 0$ *such that* $\lim_{t\to\infty} Dist\,(z[t],\ Y) = c$*, then at least one of the following two statements is true.*
(A.1) There exists a subsequence $\{z[t_k]\}_{k=0}^{\infty}$ *such that* $z[t_k] < \min Y$ *for all* $k \geq 0$.
(A.2) There exists a subsequence $\{z[t'_k]\}_{k=0}^{\infty}$ *such that* $z[t'_k] > \max Y$ *for all* $k \geq 0$.
In addition, at least one of $(\min Y - c)$ *or* $(\max Y + c)$ *is an accumulation point of* $\{z[t]\}_{t=0}^{\infty}$.

Proof. Since $\lim_{t\to\infty} Dist\,(z[t], Y) = c > 0$, there exists m such that $z[t] \notin Y$ for $t \geq m$. Otherwise, there exists a subsequence $\{z[t_k]\}_{k=0}^{\infty}$ such that $z[t_k] \in Y$ for each $k \geq 0$. By definition of $Dist\,(\cdot, Y)$, we have, $Dist\,(z[t_k], Y) = 0$ for each $k \geq 0$. Then

$$c = \lim_{t\to\infty} Dist\,(z[t_k], Y) = 0,$$

contradicting the assumption that $c > 0$.

Since $z[t] \notin Y$ for $t \geq m$, at least one of the following two statements is true.

(A.1) There exists a subsequence $\{z[t_k]\}_{k=0}^{\infty}$ such that $z[t_k] < \min Y$ for all $k \geq 0$.

(A.2) There exists a subsequence $\{z[t'_k]\}_{k=0}^{\infty}$ such that $z[t'_k] > \max Y$ for all $k \geq 0$.

By symmetry, without loss of generality, assume (A.1) is true. Then, for each $y \in Y$ and each $k \geq 0$, we have

$$z[t_k] < \min Y \leq y. \tag{3.69}$$

Thus,

$$|z[t_k] - y| = y - z[t_k].$$

Minimizing over $y \in Y$, we have

$$\begin{aligned} Dist\left(z[t_k], Y\right) &= \min_{y \in Y} |z[t_k] - y| \\ &= \min_{y \in Y} (y - z[t_k]) = \min Y - z[t_k]. \end{aligned}$$

Thus,

$$z[t_k] = \min Y - Dist\left(z[t_k], Y\right). \tag{3.70}$$

Recall that the limit of $Dist\left(z[t], Y\right)$ exists and $\lim_{t\to\infty} Dist\left(z[t], Y\right) = c$, and note that $\{Dist\left(z[t_k], Y\right)\}_{k=0}^{\infty}$ is a subsequence of $\{Dist\left(z[t], Y\right)\}_{t=0}^{\infty}$. Thus, the limit of $Dist\left(z[t_k], Y\right)$ exists, and

$$\lim_{k\to\infty} Dist\left(z[t_k], Y\right) = \lim_{t\to\infty} Dist\left(z[t], Y\right) = c.$$

This, together with equation (3.69), implies that the limit of $z[t_k]$ exists, and

$$\begin{aligned} \lim_{k\to\infty} z[t_k] &= \lim_{k\to\infty} \left(\min Y - Dist\left(z[t_k], Y\right)\right) \\ &= \min Y - \lim_{k\to\infty} Dist\left(z[t_k], Y\right) \\ &= \min Y - c. \end{aligned} \tag{3.71}$$

Thus, $(\min Y - c)$ is an accumulation point of $\{z[t]\}_{t=0}^{\infty}$.

Similarly, if (A.2) is true, i.e., there exists a subsequence $\{z[t'_k]\}_{k=0}^{\infty}$ such that $z[t'_k] > \max Y$ for all $k \geq 0$, and we can show that $(\max Y + c)$ is an accumulation point of $\{z[t]\}_{t=0}^{\infty}$.

Therefore, Proposition 7 has been proved. □

Now we prove Lemma 10.

Proof of Lemma 10. Recall from (4.41) and (3.47) that $x[t] = x_{j'_{t+1}}[t]$ and $g[t] = p'_t(x_{j'_{t+1}}[t])$. In addition, we know that if there are infinitely many points in $\{x[t]\}_{t=0}^{\infty}$ that are resilient with respect to $\{g[t]\}_{t=0}^{\infty}$ (Case (i) in the proof of Lemma 8), (3.65) holds, i.e., $\lim_{t\to\infty} Dist\,(z[t], Y) = 0$. Thus, to prove Lemma 10, it is enough to consider the case when there are only finitely many points in $\{x[t]\}_{t=0}^{\infty}$ that are resilient with respect to $\{g[t]\}_{t=0}^{\infty}$ (Case (ii) in the proof of Lemma 8).

In all subcases, we assume that $t_i \neq t_j$ when $i \neq j$ – recalling that t_i and t_j are defined in Case (ii) in the proof of Lemma 8.

Case (ii.a): Suppose there are infinitely many time indices $t \geq m_0$ such that

$$x[t] - \lambda[t]g[t] \;=\; x_{j'_{t+1}}[t] - \lambda[t]p'_t(x_{j'_{t+1}}[t]) \;\in Y.$$

Let $\{u_k\}_{k=0}^{\infty}$ be the maximal sequence of such indices. By (3.49), we have

$$Dist\,(z[u_k+1], Y) \leq 0 + L\lambda[u_k]\,(M[u_k] - m[u_k])\,. \tag{3.72}$$

By Lemma 8, we know that the limit of $Dist\,(z[t], Y)$ exists. Thus, take limit on both sides of (3.72), we get

$$\begin{aligned}
&\lim_{k\to\infty} Dist\,(z[u_k+1], Y) \\
&\leq 0 + L \lim_{k\to\infty} (\lambda[u_k]\,(M[u_k] - m[u_k])) \\
&= 0 + 0 = 0 \quad \text{by Corollary 5.}
\end{aligned}$$

On the other hand, by Lemma 8, it holds that $\lim_{t\to\infty} Dist\,(z[t], Y) = c \geq 0$. Thus,

$$c = \lim_{t\to\infty} Dist\,(z[t], Y) = \lim_{k\to\infty} Dist\,(z[u_k], Y) = 0,$$

proving the theorem.

Case (ii.b): Suppose there are only finitely many time indices $t \geq m_0$ such that

$$x[t] - \lambda[t]g[t] \;=\; x_{j'_{t+1}}[t] - \lambda[t]p'_t(x_{j'_{t+1}}[t]) \;\in Y.$$

Then, there exists $t' \geq m_0$ such that for each $t \geq t' \geq m_0$, $x[t]$ is not a resilient point with respect to $g[t]$, and

$$x[t] - \lambda[t]g[t] \;=\; x_{j'_{t+1}}[t] - \lambda[t]p'_t(x_{j'_{t+1}}[t]) \;\notin Y.$$

Thus, for each $t \geq t' \geq m_0$, (3.53) holds, i.e.,

$$\begin{aligned} Dist\,(z[t+1], Y) \leq Dist\,(z[t],\; Y) - \lambda[t]\left|p'_t(x_{j'_{t+1}}[t])\right| \\ + L\lambda[t]\,(M[t] - m[t])\,. \end{aligned}$$

Recall that $0 \leq c < \infty$ is a nonnegative constant such that

$$\lim_{t\to\infty} Dist\,(z[t], Y) = c.$$

Next we show that $c = 0$. We prove this by contradiction. Suppose $c > 0$. By Proposition 7, we know that either (A.1) is true or (A.2) is true.
(A.1) There exists a subsequence $\{z[t_k]\}_{k=0}^{\infty}$ such that $z[t_k] < \min Y$ for all $k \geq 0$.
(A.2) There exists a subsequence $\{z[t'_k]\}_{k=0}^{\infty}$ such that $z[t'_k] > \max Y$ for all $k \geq 0$.
We also know that either $(\min Y - c)$ or $(\max Y + c)$ is an accumulation point of $\{z[t]\}_{t=0}^{\infty}$.

Let $a = \min Y$, $b = \max Y$ and $\epsilon = \frac{c}{2}$. It can be seen from the proof of proposition 7 that there exists m such that $z[t] \notin Y$ for $t \geq m$. We consider three scenarios: (A.1) is true but (A.2) is not true; (A.2) is true but (A.1) is not true; both (A.1) and (A.2) are true.

Suppose (A.1) holds but (A.2) does not hold. There exists a subsequence $\{z[t_k]\}_{k=0}^{\infty}$ such that $z[t_k] < \min Y$ for all $k \geq 0$; and there does not exist a subsequence $\{z[t'_k]\}_{k=0}^{\infty}$ such that $z[t'_k] > \max Y$ for all $k \geq 0$. Recall that $z[t] \notin Y$ for $t \geq m$. Then there exists $m_1 \geq m$ such that $z[t] < \min Y$

for $t \geq m_1 \geq m$. From the proof of Claim (1), we know

$$\lim_{t\to\infty} z[t] \;=\; \min Y - c. \tag{3.73}$$

Since (3.73) holds, there exists $m_1^* \geq m_1 \geq m$ such that for all $t \geq m_1^* \geq m_1 \geq m$, the following holds:

$$|z[t] - (a-c)\,| \leq \epsilon = \frac{c}{2} \iff a - \frac{3c}{2} \leq z[t] \leq a - \frac{c}{2}. \tag{3.74}$$

Since $c > 0$, we have $a - \frac{c}{2} < a$. Then, for each $p(\cdot) \in \mathcal{C}$, $p'(a - \frac{c}{2}) < 0$. Thus,

$$\rho^* \triangleq \sup_{p(\cdot)\in\mathcal{C}} p'(a - \frac{c}{2}) \leq 0.$$

Let $h'_{i_1}(a - \frac{c}{2}), \cdots, h'_{i_{|\mathcal{N}|}}(a - \frac{c}{2})$ be a non-increasing order of $h'_j(a - \frac{c}{2})$, for $j \in \mathcal{N}$. Define function q as follows:

$$q = \left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) h_{i_1} + \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} h_{i_j}.$$

It can be easily seen that $q(\cdot) \in \mathcal{C}$ is a valid function and

$$\rho^* = \sup_{p(\cdot)\in\mathcal{C}} p'(a - \frac{c}{2}) = q'(a - \frac{c}{2}) < 0.$$

By (3.53), we have, for each $t \geq t' \geq m_0$,

$$\begin{aligned} Dist\,(z[t+1], Y) &\leq Dist\,(z[t], Y) - \lambda[t] \left|p'_t(x_{j'_{t+1}}[t])\right| + L\lambda[t]\,(M[t] - m[t]) \\ &\leq Dist\,(z[t],\ Y) - \lambda[t]\,|p'_t(z[t])| + 2L\lambda[t]\,(M[t] - m[t])\,. \end{aligned}$$

Then, for each $t \geq \widetilde{t}_1 = \max\{m_1^*, t'\}$, we have

$$\begin{aligned}
Dist\left(z[t+1], Y\right) &\leq Dist\left(z[t],\ Y\right) - \lambda[t]\left|p_t'(z[t])\right| \\
&\quad + 2L\lambda[t]\left(M[t] - m[t]\right) \\
&\leq Dist\left(z[\widetilde{t}_1],\ Y\right) - \sum_{r=\widetilde{t}_1}^{t}\left(\lambda[r]\left|p_r'(z[r])\right|\right) \\
&\quad + 2L\sum_{r=\widetilde{t}_1}^{t}\lambda[r]\left(M[r] - m[r]\right) \\
&\overset{(a)}{\leq} Dist\left(z[\widetilde{t}_1],\ Y\right) - \sum_{r=\widetilde{t}_1}^{t}\left(\lambda[r]|\rho^*|\right) \\
&\quad + 2L\sum_{r=\widetilde{t}_1}^{t}\lambda[r]\left(M[r] - m[r]\right). \qquad (3.75)
\end{aligned}$$

By (3.74), we know $z[r] \leq a - \frac{c}{2}$ for each $r \geq \widetilde{t}_1 = \max\{m_1^*, t'\}$. Then,

$$p_r'(z[r]) \leq p_r'(a - \frac{c}{2}) \leq q'(a - \frac{c}{2}) = \rho^* < 0.$$

Then, $-\left|p_r'(z[r])\right| \leq -|\rho^*|$, and inequality (a) holds.

Taking limit on both sides of (3.75), we obtain

$$\begin{aligned}
\lim_{t\to\infty} Dist\left(z[t+1], Y\right) &\leq Dist\left(z[\widetilde{t}_1],\ Y\right) - \sum_{r=\widetilde{t}_1}^{\infty}\left(\lambda[r]|\rho^*|\right) \\
&\quad + 2L\sum_{r=\widetilde{t}_1}^{\infty}\lambda[r]\left(M[r] - m[r]\right). \\
&\leq Dist\left(z[\widetilde{t}_1],\ Y\right) - \left(\sum_{r=\widetilde{t}_1}^{\infty}\lambda[r]\right)|\rho^*| \\
&\quad + 2C \quad \text{by Lemma 7} \\
&\overset{(a)}{\leq} Dist\left(z[\widetilde{t}_1],\ Y\right) - \infty + 2C \\
&= -\infty,
\end{aligned}$$

where inequality (a) is true since $|\rho^*| > 0$ and $\sum_{t=0}^{\infty}\lambda[t] = \infty$. On the other hand, we know $\lim_{t\to\infty} Dist\left(z[t], Y\right) = c \in \mathbb{R}$. A contradiction is proved.

Thus,

$$\lim_{t\to\infty} \max_{j\in\mathcal{N}} Dist\left(x_j[t], Y\right) = \lim_{t\to\infty} Dist\left(z[t], Y\right) = c = 0.$$

Similarly, we can show the case when (A.2) holds but (A.1) does not hold, and the case when both (A.1) and (A.2) hold.

The proof of the Lemma 10 is complete. □

Lemma 10 is then used to establish that the distance of $x_j[t]$ from Y converges to 0 as well, proving the optimality of $x_j[t]$ (for $j \in \mathcal{N}$) stated in Theorem 17.

3.8 Discussion

In this chapter, we introduced the problem of multi-agent optimization in the presence of Byzantine agents, and characterized the fundamental limits of the output quality of any algorithms. By exploiting Byzantine broadcast, Algorithms 2 and 3 essentially solve a centralized optimization problem where there are n cost component functions, among which up to f of them are injected by the system adversary. A much simpler distributed algorithm that achieves the optimal fault-tolerance with only local communication is proposed. As a trade-off, the simpler algorithm achieves somewhat weaker convergence property than the convergence achieved by the algorithms in Section 3.5. In particular, while the algorithms in Section 3.5 ensure that the estimates at non-faulty agents have a limit, the simpler algorithm in Section 3.7 only ensures consensus among the non-faulty agents, but does not necessarily ensure that the estimates have a limit.

Many extensions of these results are possible.

When the underlying communication channel is a broadcast channel (over which all transmissions are received correctly and identically by all agents), the results presented in this report can be proved for $n \geq 2f + 1$.

We have also obtained a comparable set of results for the scenario when the cost functions are *redundant* in some manner (e.g., cost function of agent 3 may equal a convex combination of cost functions of agents 1 and 2), or the optimal sets of the local cost functions are guaranteed to overlap. These results can be found in our report [61].

We so far focus on the *unconstrained* version of the optimization problem in (3.2). However, we can also generalize our results to the *constrained* version of problem (3.2) [64]. In particular, let $\mathcal{X} \subseteq \mathbb{R}$ such that $\mathcal{X} \neq \emptyset$, and $\mathcal{X}$ is convex and closed. Then the constrained version of (3.2) is stated below in (3.76). Observe that the output is now constrained to be in set $\mathcal{X}$.

$$\begin{aligned} &\text{output } x_o \text{ such that} \\ &\qquad \text{there } \textbf{exists} \text{ weight vector } \boldsymbol{\alpha} \text{ for which} \\ &\qquad x_o \in \underset{x\in\mathcal{X}}{\text{argmin}} \ \frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}} \alpha_i h_i(x), \\ &\qquad \sum_{i\in\mathcal{N}} \alpha_i = 1, \ \text{ and } \ \forall i, \ \alpha_i \geq 0. \end{aligned} \tag{3.76}$$

The algorithm SBG can be adapted to solve (3.76) with a simple modification of the state update in (3.23), by projecting $\widetilde{x}_j[t-1] - \lambda[t-1]\widetilde{g}_j[t-1]$ on to set $\mathcal{X}$. This projection guarantees that $x_j[t]$ is within the constraint set $\mathcal{X}$. However, compared to the original algorithm, such a projection introduces a *projection error* at each iteration. Specifically, the update of x_j can be written as follows, where $e_i[t-1]$ denotes the projection error, and $Projection_{\mathcal{X}}$ denotes projection on to $\mathcal{X}$.

$$\begin{aligned} x_j[t] &= Projection_{\mathcal{X}}\left(\widetilde{x}_j[t-1] - \lambda[t-1]\widetilde{g}_j[t-1]\right) \\ &= \widetilde{x}_j[t-1] - \lambda[t-1]\widetilde{g}_j[t-1] + e_i[t-1]. \end{aligned} \tag{3.77}$$

The projection error $e_i[t]$ can be shown to approach 0 as $t \to \infty$, and Theorem 17 holds true for the modified algorithm as well [64]. A complete algorithm description and analysis is presented in [64].

When agents crash, we can improve on the $\left(\frac{1}{2(|\mathcal{N}|-f|)}, |\mathcal{N}| - f\right)$-admissibility achieved in case of Byzantine faults. The algorithm SBG is modified in this case to perform *no trimming* at all, since the agents do not tamper with messages. For the modified algorithm, we have shown [63] that all the non-faulty agents (agents in $\mathcal{N}$) produce an output that equals an optimum of a global cost function of the form

$$c\left(\sum_{i\in\mathcal{N}} h_i(x) + \sum_{i\in\mathcal{F}} \alpha_i h_i(x)\right), \tag{3.78}$$

where $\mathcal{F}$ is the set of faulty agents (that crash at some point during the execution), $0 \leq \alpha_i \leq 1$ for each $i \in \mathcal{F}$ and c is a normalization constant such that $c\left(|\mathcal{N}| + \sum_{i \in \mathcal{F}} \alpha_i\right) = 1$. Note that in (3.78), all the local functions associated with non-faulty agents have equal weights. A finite-time interpretation of the above results is also of practical interest.

We have only considered synchronous systems so far. In an asynchronous system as well, when there are up to f Byzantine faults, algorithm SBG can be modified to achieve fault-tolerant optimization. For instance, algorithm SBG may be combined with the reliable broadcast algorithm in [72]. Alternatively, we can require $n > 5f$, and combine SBG with the simpler asynchronous iterative Byzantine consensus algorithm in [73]. The two approaches will achieve a trade-off between communication cost and optimization performance.

Open Problems

Incomplete networks In this chapter, we assumed that the underlying communication network is a completely connected. We have also explored SBG-like algorithms [64] for incomplete networks. However, our present approach is not believed to be optimal in general in incomplete network topologies. In particular, as seen previously, algorithm SBG achieves optimal fault tolerance, while also ensuring weights (α_i's) are bounded below by an adequately large constant (particularly, $\frac{1}{2(|\mathcal{N}|-f)}$). Obtaining equally strong results for incomplete networks remains an open problem.

Vector arguments Algorithm SBG assumes that the domain for the argument of the cost functions is $\mathbb{R}$ (or, in case of *constrained* optimization in (3.76) with $\mathcal{X} = \mathbb{R}$). In general, we would like to solve problem (3.2) for vector (i.e., multidimensional) arguments in $\mathbb{R}^k$ for $k \geq 2$ as well. In recent work, the problem of Byzantine *vector* consensus has been solved [74, 75]. However, a solution for Byzantine vector consensus by itself is *not* adequate to be able to solve the optimization problem of interest here. The difficulty lies in the geometry of the set of optima, when the argument is a higher dimensional vector. In particular, unlike the one-dimensional case where set Y defined in (3.25) is convex, it is not necessarily convex when the argument is higher dimensional.

Additionally, Theorem 12 can be extended to d-dimensional inputs to show that no more than $|\mathcal{N}| - df$ weights can be non-zero.

Non-smooth cost functions In our work, we assumed continuously differentiable cost functions. In general, the cost functions may be non-smooth, and the optimization algorithm would need to use *subgradients* instead of *gradients*. For the *failure-free* case, distributed subgradient optimization algorithms indeed exist [3, 4]; however, design and analysis of fault-tolerant optimization algorithms for non-smooth cost functions remain open.

3.9 Proofs

3.9.1 Proof of Theorem 12

Proof. Let $\mathcal{A}$ be an arbitrary algorithm that minimizes (3.2).

Recall that we assume $n \geq 3f + 1$. Let $h_1, \ldots, h_n$ be defined as follows, where $a = f + 1$. For each $x \in \mathbb{R}$,

- $h_i(x) = (x - i)^2$, for $1 \leq i \leq f$.
  In this case, the optimum for $h_i(x)$ is at $x = i$.
- $h_i(x) = (x - a)^2$, for $f + 1 \leq i \leq n$.
  In this case, the optimum for $h_i(x)$ is at $x = a$.

Note that the functions defined above satisfy the admissibility conditions specified in Section 3.3 except for the "bounded gradient" condition. However, the "bounded gradient" condition can be easily enforced by carefully modifying the functions values (and correspondingly gradient values) for x that are far enough away the respective optima.

From a non-faulty agent j's perspective, any subset of up to f agents may be faulty. Suppose that the faulty agents, aside from choosing their cost functions as specified above, do *not* behave incorrectly. That is, all agents follow the pre-specified algorithm $\mathcal{A}$ correctly.

Let us consider any non-faulty agent j where $f + 1 \leq j \leq n - f$. Let x_o be the output of $\mathcal{A}$. Consider two possible cases:

Case 1: In this case, suppose that agents 1 through $n - \phi$ are non-faulty, and agents $n - \phi + 1$ through n are faulty. For the local cost functions (specified above) for the non-faulty agents in this case, the optima are in the interval $[1, a]$. Then by Proposition 1, for the output x_o it must be true that $x_o \in [1, a]$.

Case 2: In Case 2, suppose that agents $f + 1$ through n are non-faulty, and agents 1 through f are faulty. For the local cost functions (specified above) for the non-faulty agents in this case, the optimum must be in $\{a\}$. Then by Proposition 1, it holds that $x_o \in \{a\}$, i.e., $x_o = a$.

Since the non-faulty agent j does not know the actual faulty agents, it cannot distinguish between the above two cases, so it must choose identical output in both cases. Therefore, the output must be in $[1, a] \cap \{a\}$; that is, the output at non-faulty agent j must equal $a = f + 1$.

Now suppose that Case 1 holds, i.e., agents $n - \phi + 1$ through n are faulty. By the above argument, the output at non-faulty agent j must be a. Now, by the requirements of (3.2), there exists a collection of weights α_i's such that $x_o = a$ is an optimum of objective

$$\sum_{i=1}^{n-\phi} \alpha_i h_i. \tag{3.79}$$

Thus, $\sum_{i=1}^{n-\phi} \alpha_i \, h_i'(a) = 0$, where $h_i'(x)$ denotes the derivative of function h_i at x.

Recall that $a = f + 1$. By construction of $h_1(x), \ldots, h_{n-\phi}(x)$, we know $h_i'(a) = 0$ for $f + 1 \leq i \leq n - \phi$ and $h_i'(a) > 0$ for $1 \leq i \leq f$. Thus

$$0 = \sum_{i=1}^{n-\phi} \alpha_i h_i'(a) = \sum_{i=1}^{f} \alpha_i h_i'(a).$$

For $1 \leq i \leq f$, since $h_i'(a) > 0$ and $\alpha_i \geq 0$ it holds that $\alpha_i h_i'(a) \geq 0$, where equality holds if and only if $\alpha_i = 0$. Thus, $\sum_{i=1}^{f} \alpha_i h_i'(a) = 0$ implies that $\alpha_i h_i'(a) = 0$ for $1 \leq i \leq f$. Then $\alpha_i = 0$ for $1 \leq i \leq f$.

Since there are $|\mathcal{N}|$ non-faulty agents (1 through $n - \phi$), and weight $\alpha_i = 0$ for $1 \leq i \leq f$, at most $|\mathcal{N}| - f$ of the weights of the non-faulty agents in Case 1 are non-zero. Thus, regardless of the value of parameter β in (3.2) (where $\beta > 0$), the parameter γ cannot be larger than $|\mathcal{N}| - f$. □

3.9.2 Proof of Proposition 2

Proof. We first show that $F(x)$ is a non-decreasing function.

Choose any $x \in \mathbb{R}$, and choose any $y \geq x$. Let S_y and S_x be sets such that $\sum_{i \in A(y)-S_y} h_i'(y)$ and $\sum_{i \in A(x)-S_x} h_i'(x)$ are minimized, respectively.

Since $h_i(\cdot)$ is convex, $h_i'(\cdot)$ is non-decreasing. By definition of $A(\cdot)$ we have $A(x) \subseteq A(y)$, i.e., $A(\cdot)$ is non-decreasing. In addition, $0 \leq |A(\cdot)| \leq n$. Similarly, we can show that $B(y) \subseteq B(x)$ and $0 \leq |B(\cdot)| \leq n$.

$$
\begin{aligned}
F(y) - F(x) &= \sum_{i \in A(y)-S_y} h_i'(y) - \sum_{i \in A(x)-S_x} h_i'(x) \\
&= \sum_{i \in A(y)-S_y-S_x} h_i'(y) + \sum_{i \in S_x \cap A(y)-S_y} h_i'(y) \\
&\quad - \left(\sum_{i \in A(x)-S_x-S_y} h_i'(x) + \sum_{i \in S_y \cap A(x)-S_x} h_i'(x) \right) \\
&= \sum_{i \in A(y)-S_y-S_x} h_i'(y) - \sum_{i \in A(x)-S_x-S_y} h_i'(x) + \sum_{i \in S_x \cap A(y)-S_y} h_i'(y) - \sum_{i \in S_y \cap A(x)-S_x} h_i'(x) \\
&\overset{(a)}{\geq} \sum_{i \in A(x)-S_y-S_x} h_i'(y) - \sum_{i \in A(x)-S_x-S_y} h_i'(x) + \sum_{i \in S_x \cap A(y)-S_y} h_i'(y) - \sum_{i \in S_y \cap A(x)-S_x} h_i'(x) \\
&\overset{(b)}{=} \sum_{i \in A(x)-S_y-S_x} h_i'(y) - \sum_{i \in A(x)-S_x-S_y} h_i'(x) + \sum_{i \in S_x-S_y} h_i'(y) - \sum_{i \in S_y \cap A(x)-S_x} h_i'(x) \\
&\overset{(c)}{\geq} \sum_{i \in S_x-S_y} h_i'(y) - \sum_{i \in S_y \cap A(x)-S_x} h_i'(x). \qquad (3.80)
\end{aligned}
$$

Inequality (a) follows from the fact that $A(x) \subseteq A(y)$ and $h_i'(y) > 0$ for each $i \in A(y)$; equality (b) is true since $S_x \subseteq A(x) \subseteq A(y)$; and inequality (c) holds because that $h_i'(\cdot)$ is non-decreasing.

Now consider two cases: (i) $|S_x| < f$ and (ii) $|S_x| = f$.

Case (i): Suppose $|S_x| < f$. In this case, we have $S_x = A(x)$, and

$$
\begin{aligned}
\sum_{i \in S_x-S_y} h_i'(y) - \sum_{i \in S_y \cap A(x)-S_x} h_i'(x) &= \sum_{i \in S_x-S_y} h_i'(y) - \sum_{i \in \emptyset} h_i'(x) \\
&= \sum_{i \in S_x-S_y} h_i'(y) \geq 0. \qquad (3.81)
\end{aligned}
$$

Case (ii): Suppose $|S_x| = f$. Because $S_x \subseteq A(x) \subseteq A(y)$, if $|S_x| = f$, we

have $|A(y)| \geq f$. Then, by definition of S_y, it holds that $|S_y| = f$. Now,

$$\begin{aligned}
|S_x - S_y| &= |S_x - S_x \cap S_y| = |S_x| - |S_x \cap S_y| \\
&= f - |S_x \cap S_y| = |S_y| - |S_x \cap S_y| \\
&= |S_y - S_x \cap S_y| \geq |S_y \cap A(x) - S_x \cap S_y| \\
&\geq |S_y \cap A(x) - S_x|.
\end{aligned}$$

Thus, $|S_x - S_y| \geq |S_y \cap A(x) - S_x|$.

By definition of S_x, for each $i \in S_x - S_y$ and $j \in S_y \cap A(x) - S_x$, at point x, we have $h_i'(x) \geq h_j'(x)$, i.e., $h_i'(x) \geq \max_{j \in S_y \cap A(x) - S_x} h_j'(x)$. We have

$$\begin{aligned}
&\sum_{i \in S_x - S_y} h_i'(y) - \sum_{i \in S_y \cap A(x) - S_x} h_i'(x) \\
&\geq \sum_{i \in S_x - S_y} h_i'(x) - \sum_{i \in S_y \cap A(x) - S_x} h_i'(x) \qquad (3.82) \\
&\geq \sum_{i \in S_x - S_y} \max_{j \in S_y \cap A(x) - S_x} h_j'(x) - \sum_{i \in S_y \cap A(x) - S_x} h_i'(x) \\
&\overset{(a)}{\geq} \sum_{i \in S_y \cap A(x) - S_x} \max_{j \in S_y \cap A(x) - S_x} h_j'(x) - \sum_{i \in S_y \cap A(x) - S_x} h_i'(x) \\
&\geq 0, \qquad (3.83)
\end{aligned}$$

where inequality (3.82) holds due to the fact that $h_i'(\cdot)$ is non-decreasing and that $y \geq x$, and (a) holds because $|S_x - S_y| \geq |S_y \cap A(x) - S_x|$ and for $j \in S_y$, $h_j'(x) > 0$.

Therefore, from (3.80), (3.81) and (3.83), we have that

$$F(y) - F(x) \geq 0, \quad \text{for all } y \geq x,$$

i.e., F is non-decreasing.

The monotonicity of $G(\cdot)$ can be shown similarly [61] with the modification that S_y and S_x are the sets such that $\sum_{i \in B(y) - S_y} h_i'(y)$ and $\sum_{i \in B(x) - S_x} h_i'(x)$ are maximized, respectively, for any $y \geq x$.

Next show that $F(x)$ is continuous. We will use the previously proven fact that F is non-decreasing.

Recall that each $h_i(x)$ is continuously differentiable, i.e., $h_i'(x)$ is continu-

ous. Then, for every $\epsilon > 0$ there exists a $\delta > 0$ such that for all $x \in (c-\delta, c+\delta)$ the following holds *for all* $i \in \mathcal{N}$,

$$|h_i'(x) - h_i'(c)| < \epsilon. \tag{3.84}$$

To show $F(x)$ is continuous, we need to show that

$$|x - c| < \delta \;\Rightarrow\; |F(x) - F(c)| < \epsilon. \tag{3.85}$$

Suppose $|x - c| < \delta$ holds for some $\delta > 0$, then $c - \delta < x < c + \delta$. Let $S_{c+\delta}$ and S_c be the subsets of $A(c+\delta)$ and $A(c)$, where $|S_{c+\delta}| \leq f$ and $|S_c| \leq f$, such that $\sum_{i \in A(c+\delta) - S_{c+\delta}} h_i'(c+\delta)$ and $\sum_{i \in A(c) - S_c} h_i'(c)$ are minimized,

respectively. Note that $A(c) \subseteq A(c+\delta)$. We have

$$
\begin{aligned}
F(x) - F(c) &\overset{(a)}{\leq} F(c+\delta) - F(c) \\
&= \sum_{i \in A(c+\delta) - S_{c+\delta}} h_i'(c+\delta) - \sum_{i \in A(c) - S_c} h_i'(c) \\
&= \sum_{i \in A(c+\delta) - S_{c+\delta} - S_c} h_i'(c+\delta) + \sum_{i \in A(c+\delta) \cap S_c - S_{c+\delta}} h_i'(c+\delta) \\
&\quad - \left(\sum_{i \in A(c) - S_{c+\delta} - S_c} h_i'(c) + \sum_{i \in S_{c+\delta} \cap A(c) - S_c} h_i'(c) \right) \\
&\overset{(b)}{=} \sum_{i \in A(c+\delta) - S_{c+\delta} - S_c} h_i'(c+\delta) + \sum_{i \in S_c - S_{c+\delta}} h_i'(c+\delta) \\
&\quad - \left(\sum_{i \in A(c) - S_{c+\delta} - S_c} h_i'(c) + \sum_{i \in S_{c+\delta} \cap A(c) - S_c} h_i'(c) \right) \\
&= \sum_{i \in A(c+\delta) - S_{c+\delta} - S_c} h_i'(c+\delta) - \sum_{i \in A(c) - S_{c+\delta} - S_c} h_i'(c) \\
&\quad + \sum_{i \in S_c - S_{c+\delta}} h_i'(c+\delta) - \sum_{i \in S_{c+\delta} \cap A(c) - S_c} h_i'(c) \\
&\overset{(c)}{\leq} \sum_{i \in A(c+\delta) - S_{c+\delta} - S_c} h_i'(c+\delta) - \sum_{i \in A(c+\delta) - S_{c+\delta} - S_c} h_i'(c) \\
&\quad + \sum_{i \in S_c - S_{c+\delta}} h_i'(c+\delta) - \sum_{i \in S_{c+\delta} - S_c} h_i'(c) \\
&\overset{(d)}{\leq} \sum_{i \in A(c+\delta) - S_{c+\delta} - S_c} h_i'(c+\delta) - \sum_{i \in A(c+\delta) - S_{c+\delta} - S_c} h_i'(c) \\
&\quad + \sum_{i \in S_{c+\delta} - S_c} h_i'(c+\delta) - \sum_{i \in S_{c+\delta} - S_c} h_i'(c)\,, \qquad (3.86)
\end{aligned}
$$

where (a) holds due to monotonicity of F; equality (b) is true since $S_c \subseteq A(c) \subseteq A(c+\delta)$; inequality (c) follows from the fact that $h_i'(c) \leq 0$ for each $i \notin A(c)$ and $A(c) \subseteq A(c+\delta)$; and inequality (d) holds because, as shown next,

$$
\sum_{i \in S_c - S_{c+\delta}} h_i'(c+\delta) \leq \sum_{i \in S_{c+\delta} - S_c} h_i'(c+\delta)\,. \qquad (3.87)
$$

Now, observing that $|S_c| \leq |S_{c+\delta}|$, we get

$$\begin{aligned}|S_c - S_{c+\delta}| &= |S_c - S_c \cap S_{c+\delta}| = |S_c| - |S_c \cap S_{c+\delta}| \\ &\leq |S_{c+\delta}| - |S_c \cap S_{c+\delta}| = |S_{c+\delta} - S_c|.\end{aligned}$$

In addition, by definition of S_c, for each $i \in S_c - S_{c+\delta}$ and $j \in S_{c+\delta} - S_c$, $h_i'(c+\delta) \leq h_j'(c+\delta)$. Then,

$$\begin{aligned}\sum_{i \in S_c - S_{c+\delta}} h_i'(c+\delta) &\leq \sum_{i \in S_c - S_{c+\delta}} \min_{j \in S_{c+\delta} - S_c} h_j'(c+\delta) \\ &\overset{(a)}{\leq} \sum_{i \in S_{c+\delta} - S_c} \min_{j \in S_{c+\delta} - S_c} h_j'(c+\delta) \\ &\leq \sum_{i \in S_{c+\delta} - S_c} h_i'(c+\delta),\end{aligned}$$

where inequality (a) is true because

$$|S_c - S_{c+\delta}| \leq |S_{c+\delta} - S_c|$$

and

$$\min_{j \in S_{c+\delta} - S_c} h_j'(c+\delta) > 0.$$

This proves (3.87). Then we have

$$\begin{aligned}&F(x) - F(c) \\ &\leq \sum_{i \in A(c+\delta) - S_{c+\delta} - S_c} h_i'(c+\delta) - \sum_{i \in A(c+\delta) - S_{c+\delta} - S_c} h_i'(c) \\ &\quad + \left(\sum_{i \in S_{c+\delta} - S_c} h_i'(c+\delta) - \sum_{i \in S_{c+\delta} - S_c} h_i'(c) \right) \quad \text{by (3.86)} \\ &\overset{(a)}{=} \sum_{i \in A(c+\delta) - S_c} (h_i'(c+\delta) - h_i'(c)) \\ &\overset{(b)}{<} |A(c+\delta) - S_c| \, \epsilon \\ &< n\epsilon.\end{aligned}$$

Equality (a) follows because $(A(c+\delta) - S_{c+\delta} - S_c) \cup (S_{c+\delta} - S_c) = A(c+\delta) - S_c$ and sets $A(c+\delta) - S_{c+\delta} - S_c$ and $S_{c+\delta} - S_c$ are disjoint. Inequality (b)

follows from (3.84).

By an analogous argument, we can also show that for any $x \in (c-\delta, c+\delta)$,

$$F(x) - F(c) > -n\epsilon.$$

For completeness, we present the proof as follows.

Let $S_{c-\delta}$ and S_c be the subsets of $A(c-\delta)$ and $A(c)$, where $|S_{c-\delta}| \leq f$ and $|S_c| \leq f$, such that $\sum_{i \in A(c-\delta)-S_{c-\delta}} h_i'(c-\delta)$ and $\sum_{i \in A(c)-S_c} h_i'(c)$ are minimized, respectively.

$$\begin{aligned}
&F(x) - F(c) \geq F(c-\delta) - F(c) \\
&= \sum_{i \in A(c-\delta)-S_{c-\delta}} h_i'(c-\delta) - \sum_{i \in A(c)-S_c} h_i'(c) \\
&= \sum_{i \in A(c-\delta)-S_{c-\delta}-S_c} h_i'(c-\delta) + \sum_{i \in S_c \cap A(c-\delta)-S_{c-\delta}} h_i'(c-\delta) \\
&\quad - \left(\sum_{i \in A(c)-S_{c-\delta}-S_c} h_i'(c) + \sum_{i \in S_{c-\delta} \cap A(c)-S_c} h_i'(c) \right) \\
&= \sum_{i \in A(c-\delta)-S_{c-\delta}-S_c} h_i'(c-\delta) - \sum_{i \in A(c)-S_{c-\delta}-S_c} h_i'(c) \\
&\quad + \sum_{i \in S_c \cap A(c-\delta)-S_{c-\delta}} h_i'(c-\delta) - \sum_{i \in S_{c-\delta} \cap A(c)-S_c} h_i'(c) \\
&\overset{(a)}{\geq} \sum_{i \in A(c)-S_{c-\delta}-S_c} h_i'(c-\delta) - \sum_{i \in A(c)-S_{c-\delta}-S_c} h_i'(c) \\
&\quad + \sum_{i \in S_c - S_{c-\delta}} h_i'(c-\delta) - \sum_{i \in S_{c-\delta} \cap A(c)-S_c} h_i'(c) \\
&\overset{(b)}{=} \sum_{i \in A(c)-S_{c-\delta}-S_c} h_i'(c-\delta) - \sum_{i \in A(c)-S_{c-\delta}-S_c} h_i'(c) \\
&\quad + \sum_{i \in S_c - S_{c-\delta}} h_i'(c-\delta) - \sum_{i \in S_{c-\delta}-S_c} h_i'(c) \\
&= \sum_{i \in A(c)-S_{c-\delta}-S_c} \left(h_i'(c-\delta) - h_i'(c) \right) \\
&\quad + \sum_{i \in S_c - S_{c-\delta}} h_i'(c-\delta) - \sum_{i \in S_{c-\delta}-S_c} h_i'(c).
\end{aligned}$$

Inequality (a) follows from the fact that $h_i'(c-\delta) \leq 0$ for each $i \notin A(c-\delta)$

and $A(c-\delta) \subseteq A(c)$. Equality (b) is true because that $S_{c-\delta} \subseteq A(c-\delta) \subseteq A(c)$. Now, observing that $|S_{c-\delta}| \leq |S_c|$, we get

$$\begin{aligned} |S_c - S_{c-\delta}| &= |S_c| - |S_c \cap S_{c-\delta}| \\ &\geq |S_{c-\delta}| - |S_c \cap S_{c-\delta}| = |S_{c-\delta} - S_c|. \end{aligned} \tag{3.88}$$

In addition, we have

$$\begin{aligned} &\sum_{i \in S_c - S_{c-\delta}} h_i'(c-\delta) - \sum_{i \in S_{c-\delta} - S_c} h_i'(c) \\ &\overset{(a)}{\geq} \sum_{i \in S_c - S_{c-\delta}} h_i'(c-\delta) - \sum_{i \in S_{c-\delta} - S_c} \min_{j \in S_c - S_{c-\delta}} h_j'(c) \\ &\overset{(b)}{\geq} \sum_{i \in S_c - S_{c-\delta}} h_i'(c-\delta) - \sum_{i \in S_c - S_{c-\delta}} \min_{j \in S_c - S_{c-\delta}} h_j'(c) \\ &\geq \sum_{i \in S_c - S_{c-\delta}} h_i'(c-\delta) - \sum_{i \in S_c - S_{c-\delta}} h_i'(c) \\ &= \sum_{i \in S_c - S_{c-\delta}} \left(h_i'(c-\delta) - h_i'(c)\right). \end{aligned} \tag{3.89}$$

Inequality (a) holds due to the fact that for each $i \in S_{c-\delta} - S_c$, $h_i'(c) \leq \min_{j \in S_c - S_{c-\delta}} h_j'(c)$. Inequality (b) follows from (3.88) and the fact that $\min_{j \in S_c - S_{c-\delta}} h_j'(c) > 0$. Thus

$$\begin{aligned} F(x) - F(c) &\geq \sum_{i \in A(c) - S_{c-\delta} - S_c} \left(h_i'(c-\delta) - h_i'(c)\right) \\ &\quad + \left(\sum_{i \in S_c - S_{c-\delta}} h_i'(c-\delta) - \sum_{i \in S_{c-\delta} - S_c} h_i'(c) \right) \\ &\geq \sum_{i \in A(c) - S_{c-\delta} - S_c} \left(h_i'(c-\delta) - h_i'(c)\right) \\ &\quad + \sum_{i \in S_c - S_{c-\delta}} \left(h_i'(c-\delta) - h_i'(c)\right) \quad \text{from (3.89)} \\ &= \sum_{i \in A(c) - S_{c-\delta}} \left(h_i'(c-\delta) - h_i'(c)\right) \\ &> -n\epsilon \quad \text{from (3.84)}. \end{aligned}$$

Then we have, for any $\epsilon_0 = n\epsilon > 0$, there exists $\delta > 0$ such that

$$|x - c| < \delta \;\Rightarrow\; |F(x) - F(c)| < \epsilon_0.$$

Therefore, $F(\cdot)$ is continuous.

Continuity of $G(\cdot)$ can be proved similarly. □

3.9.3 Proof of Lemma 4

Proof that set Y is convex

We first prove that set Y is convex. Let $x_1, x_2 \in Y$ such that $x_1 \neq x_2$. By definition of Y, there exist valid functions $p_1 = \sum_{i\in\mathcal{N}} \alpha_i h_i \in \mathcal{C}$ and $p_2 = \sum_{i\in\mathcal{N}} \beta_i h_i \in \mathcal{C}$ such that $x_1 \in \text{argmin } p_1(x)$ and $x_2 \in \text{argmin } p_2(x)$, respectively. In addition, let $p = \frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}} h_i$. By definition of valid function in (3.24), it holds that $p \in \mathcal{C}$. Note that it is possible that $p_1 = p_2$, and that $p_i = p$ for $i = 1$ or $i = 2$.

Given $0 \leq \alpha \leq 1$, let $x_\alpha = \alpha x_1 + (1-\alpha)x_2$. We consider two cases: (i) $x_\alpha \in \text{argmin } p_1(x) \cup \text{argmin } p_2(x) \cup \text{argmin } p(x)$, and (ii) $x_\alpha \notin \text{argmin } p_1(x) \cup \text{argmin } p_2(x) \cup \text{argmin } p(x)$.

When $x_\alpha \in \text{argmin } p_1(x) \cup \text{argmin } p_2(x) \cup \text{argmin } p(x)$, by definition of Y, we have

$$x_\alpha \in \text{argmin } p_1(x) \cup \text{argmin } p_2(x) \cup \text{argmin } p(x) \subseteq Y.$$

Now we consider the case when $x_\alpha \notin \text{argmin } p_1(x) \cup \text{argmin } p_2(x) \cup \text{argmin } p(x)$.

Without loss of generality, assume that $x_1 < x_2$. By definition of x_α, we have $x_1 < x_\alpha < x_2$. By the fact that $\text{argmin}_{x\in\mathbb{R}} p_1(x)$ and $\text{argmin}_{x\in\mathbb{R}} p_2(x)$ are convex, it holds that $\max(\text{argmin } p_1(x)) < x_\alpha < \min(\text{argmin } p_2(x))$, which imply that $p_1'(x_\alpha) > 0$ and $p_2'(x_\alpha) < 0$.

There are two possibilities for $p'(x_\alpha)$ (the gradient of $p(x_\alpha)$): $p'(x_\alpha) < 0$ or $p'(x_\alpha) > 0$. Note that $p'(x_\alpha) \neq 0$ because $x_\alpha \notin \text{argmin } p(x)$.

When $p'(x_\alpha) < 0$, there exists $0 \leq \zeta \leq 1$ such that

$$\zeta\, p_1'(x_\alpha) + (1-\zeta)\, p'(x_\alpha) = 0.$$

By definition of functions p_1 and p, we have

$$\begin{aligned} 0 &= \zeta\ p_1'(x_\alpha) + (1-\zeta)\ p'(x_\alpha) \\ &= \sum_{i\in\mathcal{N}} \left(\alpha_i\zeta + (1-\zeta)\frac{1}{|\mathcal{N}|}\right) h_i'(x_\alpha). \end{aligned}$$

Thus, x_α is an optimum of function

$$\sum_{i\in\mathcal{N}} \left(\alpha_i\zeta + (1-\zeta)\frac{1}{|\mathcal{N}|}\right) h_i. \tag{3.90}$$

Let $\mathcal{I}$ be the collection of indices defined by

$$\mathcal{I} \triangleq \{\ i:\ i\in\mathcal{N},\ \text{and}\ \alpha_i\zeta + (1-\zeta)\frac{1}{|\mathcal{N}|}\ \geq\ \frac{1}{2(|\mathcal{N}|-f)}\ \}.$$

Next we show that $|\mathcal{I}| \geq |\mathcal{N}| - f$. Let $\mathcal{I}_1$ defined by

$$\mathcal{I}_1 \triangleq \{\ i:\ i\in\mathcal{N},\ \text{and}\ \alpha_i \geq \frac{1}{2(|\mathcal{N}|-f)}\ \}.$$

Since $p_1 \in \mathcal{C}$, then $|\mathcal{I}_1| \geq |\mathcal{N}| - f$. In addition, since $n > 3f$ and $|\mathcal{N}| = n - |\mathcal{F}| > 2f$, we have $|\mathcal{N}| < 2(|\mathcal{N}| - f)$. Thus, for each $j \in \mathcal{I}_1$, we have $\alpha_j\zeta + (1-\zeta)\frac{1}{|\mathcal{N}|} > \frac{1}{2(|\mathcal{N}|-f)}$, i.e., $j \in \mathcal{I}$. Thus, $\mathcal{I}_1 \subseteq \mathcal{I}$.

Since $|\mathcal{I}_1| \geq |\mathcal{N}| - f$, we have $|\mathcal{I}| \geq |\mathcal{N}| - f$. So function (3.90) is a valid function in $\mathcal{C}$. Thus, $x_\alpha \in Y$.

Similarly, we can show that the above result holds when $p'(x_\alpha) > 0$. Therefore, set Y is convex.

Proof that Set Y is closed

To show that Y is closed, we need the following proposition.

For each $x \in \mathbb{R}$, let $h'_{i_1(x)}(x), \cdots, h'_{i_{|\mathcal{N}|}(x)}(x)$ be a *non-increasing* order of $h'_j(x)$, for $j \in \mathcal{N}$, i.e., $h'_{i_1(x)}(x) \geq \cdots \geq h'_{i_{|\mathcal{N}|}(x)}(x)$. Note that associated with the gradient order, there is a corresponding list of non-faulty agents $\{i_1(x), i_2(x), \ldots, i_{|\mathcal{N}|}(x)\}$, in which the relative ranks of non-faulty agents

vary with x. For each x, define $r(x)$ to be

$$r(x) \triangleq \left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) h'_{i_1(x)}(x) + \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} h'_{i_j(x)}(x). \tag{3.91}$$

At each $x \in \mathbb{R}$, function $r(x)$ is the largest gradient value among all the valid functions in $\mathcal{C}$.

Proposition 8. *Function r is continuous and non-decreasing.*

Proof. Recall that $\{i_1(x), i_2(x), \ldots, i_{|\mathcal{N}|}(x)\}$ is the list of non-faulty agents that corresponds to the non-increasing gradient order $h'_{i_1(x)}(x), \cdots, h'_{i_{|\mathcal{N}|}(x)}(x)$. By definition, function

$$\left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) h_{i_1(x)}(\cdot) + \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} h_{i_j(x)}(\cdot)$$

is contained in $\mathcal{C}$. Since at each $x \in \mathbb{R}$, $r(x)$ is the largest gradient among gradients of all valid functions in $\mathcal{C}$, for any y (which may equal x) we have

$$r(y) \geq \left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) h'_{i_1(x)}(y) + \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} h'_{i_j(x)}(y). \tag{3.92}$$

Now, suppose $y \geq x \in \mathbb{R}$. Since $h'_i(\cdot)$ is non-decreasing, we have

$$\begin{aligned} r(y) &\geq \left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) h'_{i_1(x)}(y) + \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} h'_{i_j(x)}(y) \quad \text{by (3.92)} \\ &\geq \left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) h'_{i_1(x)}(x) + \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} h'_{i_j(x)}(x) \\ &= r(x) \quad \text{by (3.91)} \end{aligned}$$

Thus, function $r(\cdot)$ is non-decreasing.

Next we show that function $r(\cdot)$ is continuous.

For each $i \in \mathcal{V}$, since $h_i(\cdot)$ is differentiable, it follows that $h'_i(\cdot)$ is continu-

ous. That is, for each $i \in \mathcal{V}$ and $\forall\, \epsilon > 0,\ \exists\ \delta > 0$ such that

$$|x - c| < \delta \implies |h_i'(x) - h_i'(c)| \leq \epsilon. \tag{3.93}$$

Assume $c \leq x < c + \delta$. Then

$$\begin{aligned}
&|r(x) - r(c)| = r(x) - r(c) \quad \text{by monotonicity of } r(\cdot) \\
&= \left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) h'_{i_1(x)}(x) + \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} h'_{i_j(x)}(x) \\
&\quad - \left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) h'_{i_1(c)}(c) - \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} h'_{i_j(c)}(c) \\
&\leq \left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) h'_{i_1(x)}(x) + \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} h'_{i_j(x)}(x) \\
&\quad - \left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) h'_{i_1(x)}(c) - \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} h'_{i_j(x)}(c) \quad \text{by (3.92)} \\
&\leq \left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) \left(h'_{i_1(x)}(x) - h'_{i_1(x)}(c)\right) \\
&\quad + \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} \left(h'_{i_j(x)}(x) - h'_{i_1(x)}(c)\right) \\
&< \left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) \epsilon + \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} \epsilon \quad \text{by (3.93)} \\
&= \left(\left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) + \frac{1}{2(|\mathcal{N}| - f)} \cdot (|\mathcal{N}| - f - 1)\right) \epsilon \\
&= \epsilon.
\end{aligned} \tag{3.94}$$

Similarly, we can show that when $c - \delta < x \leq c$, $|r(x) - r(c)| < \epsilon$.

Thus, function $r(\cdot)$ is continuous.

The proof of Proposition 8 is complete.

□

Proof that Y is closed

With the auxiliary function r at hand, we can show the closedness of set Y as follows.

Recall that Y is convex. To show Y is closed, it is enough to show that Y is bounded and both $\min Y$ and $\max Y$ exist.

It can be easily seen that $r(x)$ is negative for "sufficiently" small x, and positive for "sufficiently" large x. By Proposition 8, we know that there exists $x_0 \in \mathbb{R}$ such that

$$\begin{aligned} 0 \;=\; r(x_0) = &\left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) h'_{i_1(x_0)}(x_0) \\ &+ \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} h'_{i_j(x_0)}(x_0). \end{aligned}$$

Define function q as follows:

$$q \triangleq \left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) h_{i_1(x_0)} + \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} h_{i_j(x_0)}. \tag{3.95}$$

By definition of function q, we know that $q'(x_0) = r(x_0) = 0$, and that $q \in \mathcal{C}$ is a valid function. Note that due to the possibility of existence of ties in the top $|\mathcal{N}| - f$ rankings of the order $h'_{i_1(x)}(x), \cdots, h'_{i_{|\mathcal{N}|}(x)}(x)$, for a given x, there may be multiple orders over $h'_i(x_0), \forall i \in \mathcal{N}$ of the top $|\mathcal{N}| - f$ elements. Let $\mathcal{O}$ be the collection of all such orders. Note that there is a one-to-one correspondence of an order and a valid function defined in (3.95). We denote q_o as the valid function associated with an order o. Let $a = \min_{o \in \mathcal{O}} \min\left(\operatorname{argmin} q_o(x)\right)$, which is well-defined since $\operatorname{argmin} q_o(x)$ is compact, and $|\mathcal{O}|$ is finite.

By definition $a \in Y$. Next we show that $a = \min Y$. Suppose, on the contrary, that there exists $\widetilde{a} < a$ such that $\widetilde{a} \in Y$. Since $\widetilde{a} \in Y$, there exists $\widetilde{q} = \sum_{i \in \mathcal{N}} \alpha_i h_i \in \mathcal{C}$ such that $\widetilde{a} \in \operatorname{argmin} \widetilde{q}(x)$. That is, $\widetilde{q}'(\widetilde{a}) = 0$. Then, we

have

$$\begin{aligned} 0 = q'(\widetilde{a}) &= \sum_{i\in\mathcal{N}} \alpha_i h_i'(\widetilde{a}) \\ &\le \left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) h'_{i_1(\widetilde{a})}(\widetilde{a}) \\ &\quad + \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} h'_{i_j(\widetilde{a})}(\widetilde{a}) \\ &= r(\widetilde{a}). \end{aligned}$$

From the definition of a and the assumption that $\widetilde{a} < a$, we get $\widetilde{a} < x_0$. Then, by monotonicity of $r(\cdot)$, we have

$$r(\widetilde{a}) \le r(x_0).$$

Thus, $r(\widetilde{a}) = 0 = r(x_0)$. In addition, we have

$$\begin{aligned} 0 = r(\widetilde{a}) &\le \left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) h'_{i_1(\widetilde{a})}(x_0) \\ &\quad + \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} h'_{i_j(\widetilde{a})}(x_0) \\ &\le \left(1 - \frac{|\mathcal{N}| - f - 1}{2(|\mathcal{N}| - f)}\right) h'_{i_1(x_0)}(x_0) \\ &\quad + \frac{1}{2(|\mathcal{N}| - f)} \sum_{j=2}^{|\mathcal{N}|-f} h'_{i_j(x_0)}(x_0) \le 0, \end{aligned}$$

which implies that $i_1(\widetilde{a}), \cdots, i_{|\mathcal{N}|-f}(\widetilde{a})$ is an order in $\mathcal{O}$. Thus, $\widetilde{a} \ge a = \min_{o\in\mathcal{O}} \min\left(\operatorname{argmin}\, q_o(x)\right)$, contradicting the assumption that $\widetilde{a} < a$.

Thus, $a = \min Y$, i.e., $\min\, Y$ exists. Similarly, we can show that $\max\, Y$ also exists. Therefore, set Y is closed.

3.9.4 Proof of Proposition 5

Proof of Proposition 5. For any $t \geq 1$, we have

$$\begin{aligned}
\ell(t) &= \sum_{r=0}^{t-1} \lambda[r] b^{t-r} = \sum_{r=0}^{\lceil \frac{t}{2} \rceil} \lambda[r] b^{t-r} + \sum_{r=\lceil \frac{t}{2} \rceil + 1}^{t-1} \lambda[r] b^{t-r} \\
&\leq \sum_{r=0}^{\lceil \frac{t}{2} \rceil} \lambda[0] b^{t-r} + \lambda[\lceil \frac{t}{2} \rceil] \sum_{r=\lceil \frac{t}{2} \rceil + 1}^{t-1} b^{t-r} \\
&\leq \lambda[0] \frac{b^{t-\lceil \frac{t}{2} \rceil}}{1-b} + \frac{b \lambda[\lceil \frac{t}{2} \rceil]}{1-b} \\
&\leq \lambda[0] \frac{b^{\frac{t}{2}-1}}{1-b} + \frac{b \lambda[\lceil \frac{t}{2} \rceil]}{1-b}. \qquad (3.96)
\end{aligned}$$

Thus, we get

$$\begin{aligned}
\limsup_{t \to \infty} \ell(t) &\leq \lim_{t \to \infty} \left(\lambda[0] \frac{b^{\frac{t}{2}-1}}{1-b} + \frac{b \lambda[\lceil \frac{t}{2} \rceil]}{1-b} \right) \\
&= \lambda[0] \frac{1}{1-b} \lim_{t \to \infty} b^{\frac{t}{2}-1} + \frac{b}{1-b} \lim_{t \to \infty} \lambda[\lceil \frac{t}{2} \rceil] \\
&\overset{(a)}{=} 0 + 0 = 0.
\end{aligned}$$

Equality (a) holds because $0 \leq b < 1$ and $\lim_{t \to \infty} \lambda[\lceil \frac{t}{2} \rceil] = 0$. On the other hand, by definition of $\ell(t)$ we know $\ell(t) \geq 0$ for each $t \geq 1$. Thus, $\liminf_{t \to \infty} \ell(t) \geq 0$.

Therefore, the limit of $\ell(t)$ exists and $\lim_{t \to \infty} \ell(t) = 0$.

Consider step sizes $\lambda[t] = \frac{1}{t}$ for $t \geq 1$ and $\lambda[0] = 1$. It immediately follows from (3.96) that

$$\ell(t) \leq \lambda[0] \frac{b^{\frac{t}{2}-1}}{1-b} + \frac{b \lambda[\lceil \frac{t}{2} \rceil]}{1-b} = \lambda[0] \frac{b^{\frac{t}{2}-1}}{1-b} + \frac{b}{1-b} \frac{1}{\lceil \frac{t}{2} \rceil}$$

Thus, $\ell(t) = O(\frac{1}{t})$. □

3.9.5 Proof of Lemma 7

Proof of Lemma 7. To show Lemma 7, it is enough to show that

$$\sum_{t=1}^{\infty} \lambda[t]\left(M[t]-m[t]\right) < \infty.$$

We have

$$\begin{aligned}
&\sum_{t=1}^{\infty} \lambda[t]\left(M[t]-m[t]\right) \\
&\leq \left(M[0]-m[0]\right) \sum_{t=1}^{\infty} \lambda[t] \left(1-\frac{1}{2\left(|\mathcal{N}|-f\right)}\right)^{t} \\
&\quad + 2L \sum_{t=1}^{\infty} \sum_{r=0}^{t-1} \left(\left(1-\frac{1}{2\left(|\mathcal{N}|-f\right)}\right)^{t-r} \lambda[r]\lambda[t] \right) \quad \text{by (3.40)} \\
&\overset{(a)}{\leq} \left(M[0]-m[0]\right) \sum_{t=1}^{\infty} \lambda[t] \left(1-\frac{1}{2\left(|\mathcal{N}|-f\right)}\right)^{t} \\
&\quad + L \sum_{t=1}^{\infty} \lambda^2[t] \sum_{r=0}^{t-1} \left(1-\frac{1}{2\left(|\mathcal{N}|-f\right)}\right)^{t-r} \\
&\quad + L \sum_{t=1}^{\infty} \sum_{r=0}^{t-1} \left(\left(1-\frac{1}{2\left(|\mathcal{N}|-f\right)}\right)^{t-r} \lambda^2[r] \right). \qquad (3.97)
\end{aligned}$$

Inequality (a) holds because $\lambda[t]\lambda[r] \leq \frac{\lambda^2[t]+\lambda^2[r]}{2}$. It is easy to see that

$$\sum_{t=1}^{\infty} \left(1-\frac{1}{2\left(|\mathcal{N}|-f\right)}\right)^{t} \leq 2\left(|\mathcal{N}|-f\right). \qquad (3.98)$$

We bound the terms in the right hand side of (3.97) separately.

The first term of (3.97): Since $\lambda[t] \leq \lambda[0]$ for $t \geq 1$, we have

$$\begin{aligned}
&\left(M[0]-m[0]\right) \sum_{t=1}^{\infty} \lambda[t] \left(1-\frac{1}{2\left(|\mathcal{N}|-f\right)}\right)^{t} \\
&\leq \left(M[0]-m[0]\right) \lambda[0] \sum_{t=1}^{\infty} \left(1-\frac{1}{2\left(|\mathcal{N}|-f\right)}\right)^{t} \\
&\overset{(b)}{\leq} \left(M[0]-m[0]\right) \lambda[0] 2\left(|\mathcal{N}|-f\right) < \infty, \qquad (3.99)
\end{aligned}$$

where inequality (b) follows from (3.98).

The second term of (3.97):

$$
\begin{aligned}
& L \sum_{t=1}^{\infty} \lambda^2[t] \sum_{r=0}^{t-1} \left(1 - \frac{1}{2\left(|\mathcal{N}| - f\right)}\right)^{t-r} \\
&= L \sum_{t=1}^{\infty} \lambda^2[t] \sum_{r=1}^{t} \left(1 - \frac{1}{2\left(|\mathcal{N}| - f\right)}\right)^{r} \\
&\leq L \sum_{t=1}^{\infty} \lambda^2[t] \sum_{r=0}^{\infty} \left(1 - \frac{1}{2\left(|\mathcal{N}| - f\right)}\right)^{r} \\
&= 2\left(|\mathcal{N}| - f\right) L \sum_{t=1}^{\infty} \lambda^2[t] \quad \text{by (3.98)} \\
&< \infty. \qquad (3.100)
\end{aligned}
$$

The last inequality follows from the fact that $\sum_{t=1}^{\infty} \lambda^2[t] \leq \sum_{t=0}^{\infty} \lambda^2[t] < \infty$ (by assumption about $\lambda[t]$ in Section 3.7).

The third term of (3.97): For any fixed T, we get

$$
\begin{aligned}
& L \sum_{t=1}^{T} \sum_{r=0}^{t-1} \left(\left(1 - \frac{1}{2\left(|\mathcal{N}| - f\right)}\right)^{t-r} \lambda^2[r] \right) \\
&= L \sum_{r=0}^{T-1} \lambda^2[r] \sum_{t=1}^{T-r} \left(1 - \frac{1}{2\left(|\mathcal{N}| - f\right)}\right)^{t} \\
&\leq L \sum_{r=0}^{T-1} \lambda^2[r] \sum_{t=0}^{\infty} \left(1 - \frac{1}{2\left(|\mathcal{N}| - f\right)}\right)^{t} \\
&= 2\left(|\mathcal{N}| - f\right) L \sum_{r=0}^{T-1} \lambda^2[r] \quad \text{by (3.98)}.
\end{aligned}
$$

Let $T \to \infty$, we get

$$
\begin{aligned}
& L \sum_{t=1}^{\infty} \sum_{r=0}^{t-1} \left(\left(1 - \frac{1}{2\left(|\mathcal{N}| - f\right)}\right)^{t-r} \lambda^2[r] \right) \\
&\leq 2\left(|\mathcal{N}| - f\right) L \sum_{r=0}^{\infty} \lambda^2[r] \; < \; \infty. \qquad (3.101)
\end{aligned}
$$

We get

$$\begin{aligned}
&\sum_{t=1}^{\infty} \lambda[t]\left(M[t]-m[t]\right) \\
&\leq \left(M[0]-m[0]\right) \sum_{t=1}^{\infty} \lambda[t]\left(1-\frac{1}{2\left(|\mathcal{N}|-f\right)}\right)^{t} \\
&\quad + L \sum_{t=1}^{\infty} \lambda^{2}[t] \sum_{r=0}^{t-1}\left(1-\frac{1}{2\left(|\mathcal{N}|-f\right)}\right)^{t-r} \\
&\quad + L \sum_{t=1}^{\infty} \sum_{r=0}^{t-1}\left(1-\frac{1}{2\left(|\mathcal{N}|-f\right)}\right)^{t-r} \lambda^{2}[r] \quad \text{by} \quad (3.97) \\
&< \infty+\infty+\infty=\infty \quad \text{by (3.99), (3.100) and (3.101)},
\end{aligned}$$

proving the lemma. □

3.9.6 Proof of Lemma 8

Proof of Corollary 5. By Lemma 7, we have $\sum_{t=0}^{\infty} \lambda[t]\left(M[t]-m[t]\right)<\infty$. Thus, $\lim_{t\to\infty} \lambda[t]\left(M[t]-m[t]\right)=0$ holds trivially.

Now we prove that $\lim_{t\to\infty} \sum_{\tau=t}^{\infty} \lambda[\tau]\left(M[\tau]-m[\tau]\right)=0$.

Let $F=\sum_{\tau=0}^{\infty} \lambda[\tau](M[\tau]-m[\tau])$, and let $\{F_t\}_{t=0}^{\infty}$ be a sequence such that for each t,

$$F_t=\sum_{\tau=0}^{t-1} \lambda[\tau](M[\tau]-m[\tau]).$$

Since $M[\tau]-m[\tau]\geq 0$ for each $\tau\geq 0$, by construction, it holds that $F_t\leq F_{t+1}$ and that $F_t\leq F$ for each $t\geq 1$. Thus, by monotone convergence theorem, we know that

$$\lim_{t\to\infty} F_t=F.$$

Now, let $R_t \triangleq F-F_t=\sum_{\tau=t}^{\infty} \lambda[\tau](M[\tau]-m[\tau])$. By Lemma 7, we know that $F<\infty$. Thus the sequence R_t is well-defined. In addition, since the sequence F_t converges, then the sequence R_t also converges. So, we get

$$\lim_{t\to\infty} \sum_{\tau=t}^{\infty} \lambda[\tau](M[\tau]-m[\tau])=\lim_{t\to\infty} R_t=F-\lim_{t\to\infty} F_t=0,$$

proving that $\lim_{t\to\infty} \sum_{\tau=t}^{\infty} \lambda[\tau]\left(M[\tau]-m[\tau]\right)=0$. □

Proof of Proposition 6. For each $i \in \mathcal{N}$, by Definition 15, we have

$$Dist\left(x_i[t], Y\right) \leq \max_{j \in \mathcal{N}} Dist\left(x_j[t], Y\right) = Dist\left(z[t], Y\right).$$

By proposition assumption, we get

$$0 \leq \limsup_{t \to \infty} Dist\left(x_i[t], Y\right) \leq \limsup_{t \to \infty} Dist\left(z[t], Y\right) = 0.$$

Therefore, for each $i \in \mathcal{N}$, $\{Dist\left(x_i[t], Y\right)\}_{t=0}^{\infty}$ converges and

$$\lim_{t \to \infty} Dist\left(x_i[t], Y\right) = 0.$$

□

CHAPTER 4

CONSENSUS-BASED DISTRIBUTED HYPOTHESIS TESTING

4.1 Introduction

The traditional decentralized detection framework consists of a collection of spatially distributed sensors and a fusion center [76, 77, 78]. The sensors independently collect *noisy* observations of the environment state, and send only *summary* of the private observations to the fusion center, where a final decision is made. In the case when the sensors directly send all the private observations, the detection problem can be solved using a centralized scheme. The above framework does not scale well, since each sensor needs to be connected to the fusion center and full reliability of the fusion center is required, which may not be practical as the system scales.

Distributed hypothesis testing in the *absence* of fusion center is considered in [79, 80, 81]. In particular, Gale and Kariv [79] studied the distributed hypothesis testing problem in the context of social learning, where the fully Bayesian belief update rule is studied. The Bayesian update rule is impractical in many applications due to memory and computation constraints of each agent.

To avoid the complexity of Bayesian learning, a non-Bayesian learning framework that combines local Bayesian learning with distributed consensus was proposed by Jadbabaie et al. [11], and has attracted much attention [82, 83, 84, 85, 86, 87, 88, 89]. Jadbabaie et al. [11] considered the general setting where external signals are observed during each iteration of the algorithm execution. Specifically, the "belief" of each agent is repeatedly updated as the arithmetic mean of its local Bayesian update and the "beliefs" of its neighbors – combining iterative consensus algorithm with local Bayesian update. Note that the "belief" in [11] is not the exact belief, instead, it is only an approximate. Henceforth, in this chapter, to avoid confusion, we refer to

the approximation "belief" as *score*. It is shown [11] that, under this learning rule, each agent learns the true state almost surely. The publication of [11] has inspired significant efforts in designing and analyzing non-Bayesian learning rules with a particular focus on refining the fusion strategies and analyzing the (asymptotic and/or finite time) convergence rates of the refined algorithms [82, 83, 84, 85, 86, 87, 88, 89]. In this chapter we are particularly interested in the log-linear form of the update rule, in which, essentially, each agent updates its score as the geometric average of the local Bayesian update and its neighbors' scores [84, 82, 83, 85, 86, 87, 88, 89]. The log-linear form (geometric averaging) update rule is shown to converge exponentially fast [82, 85]. Taking an axiomatic approach, the geometric averaging fusion is proved to be optimal [89]. An optimization-based interpretation of this rule is presented in [85], using dual averaging method with properly chosen proximal functions. Finite-time convergence rates are investigated independently in [83, 86, 88]. Both [83] and [87] consider time-varying networks, with slightly different network models. Specifically, [83] assumes that the union of all consecutive B networks is strongly connected, while [87] considers random networks. In this chapter, we consider static networks for ease of exposition, although we believe that our results can be easily generalized to time-varying networks.

The prior work implicitly assumes that the networked agents are reliable in the sense that they correctly follow the specified learning rules. However, in some practical multi-agent networks, this assumption may not hold. For example, in social networks, it is possible that some agents are adversarial, and try to prevent the true state from being learned by the good agents. Thus, this chapter focuses on the fault-tolerant version the non-Bayesian framework proposed in [11]. In particular, we assume that an unknown subset of agents may suffer Byzantine faults.

The existing non-Bayesian learning algorithms [82, 86, 89, 83, 84, 85, 88, 87] are not robust to Byzantine agents, since the malicious messages sent by the Byzantine agents are indiscriminately utilized in the local score updates. On the other hand, the incorporation of Byzantine consensus is non-trivial, since (i) the *effective* communication networks are *dependent* on the random local observations, making it non-trivial to adapt analysis of previous algorithms to our setting; and (ii) the problem of identifying tight topological conditions for

reaching Byzantine multi-dimensional consensus iteratively is open, making it challenging to identify the minimal detectability condition on the networked agents to learn the true environmental state.

Contributions: Our contributions are two-fold.

- We first propose an update rule wherein each agent iteratively updates its local scores as (up to normalization) the product of (1) the likelihood of the *cumulative* private signals and (2) the weighted geometric average of the scores of its incoming neighbors and itself (using iterative Byzantine multi-dimensional consensus). In contrast to the existing algorithms [83, 86], where only the *current* private signal is used in the update, our proposed algorithm relies on the *cumulative* private signals. Under reasonable assumptions on the underlying network structure and the global identifiability of the network, we show that all the non-faulty agents asymptotically agree on the true state almost surely.

- The local computation complexity per agent of the first learning rule is high due to the adoption of multi-dimensional consensus primitives. More importantly, the network identifiability condition used for that learning rule scales poorly in the number of possible states m. Thus, we propose a modification of our first learning rule, whose complexity per iteration per agent is $O(m^2 n \log n)$, where n is the number of agents in the network. We show that this improved learning rule works under a much weaker global identifiability condition, which is independent of m. We cast the general m–ary hypothesis testing problem into a collection of binary hypothesis testing sub-problems.

Outline: The rest of the chapter is organized as follows. Section 4.2 presents the problem formulation. Section 4.3 briefly reviews existing results on vector Byzantine consensus, and matrix representation of the state evolution. Our first algorithm and its correctness analysis are presented in Section 4.4. The improved learning rule and its correctness analysis are summarized in Section 4.5. Section 4.6 demonstrates the above learning rule in the special case when $f = 0$, and presents a finite-time analysis. Section 4.7 concludes the chapter and discusses possible extensions.

4.2 Problem Formulation

Network Model: Our network model is similar to the model used in [48, 34]. We consider a synchronous system. A collection of n agents (also referred as *nodes*) are connected by a *directed* network $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, n\}$ and $\mathcal{E}$ is the collection of *directed* edges. For each $i \in \mathcal{V}$, let $\mathcal{I}_i$ denote the set of incoming neighbors of agent i. In any execution, up to f agents suffer Byzantine faults. For a given execution, let $\mathcal{F}$ denote the set of Byzantine agents, and $\mathcal{N}$ denote the set of non-faulty agents. Throughout this chapter, we assume that f satisfies the condition implicitly imposed by the given topology conditions mentioned later. We assume that each non-faulty agent knows f, but does not know the *actual* number of faulty agents $|\mathcal{F}|$. Possible misbehavior of faulty agents includes sending incorrect and mismatching (or inconsistent) messages. The Byzantine agents are also assumed to have complete knowledge of the system, including the network topology, underlying running algorithm, the states or even the entire history. The faulty agents may collaborate with each other adaptively [29]. Note that $|\mathcal{F}| \leq f$ and $|\mathcal{N}| \geq n - f$ since at most f agents may fail.

Throughout this chapter, we use the terms *agent* and *node* interchangeably.

Observation Model: Our observation model is identical to the model used in [11, 86, 87]. Let $\Theta = \{\theta_1, \theta_2, \ldots, \theta_m\}$ denote a set of m environmental states, which we call *hypotheses*. In the t-th iteration, each agent *independently* obtains a private signal about the environmental state θ^*, which is initially unknown to every agent in the network. Each agent i knows the structure of its private signal, which is represented by a collection of parameterized marginal distributions $\mathcal{D}^i = \{\ell_i(w_i|\theta) | \, \theta \in \Theta, \, w_i \in \mathcal{S}_i\}$, where $\ell_i(\cdot|\theta)$ is the distribution of private signal when θ is the true state, and $\mathcal{S}_i$ is the finite private signal space. For each $\theta \in \Theta$, and each $i \in \mathcal{V}$, the support of $\ell_i(\cdot|\theta)$ is the whole signal space, i.e., $\ell_i(w_i|\theta) > 0, \, \forall \, w_i \in \mathcal{S}_i$ and $\forall \, \theta \in \Theta$. Let s_t^i be the private signal observed by agent i in iteration t, and let $\mathbf{s}_t = \{s_t^1, s_t^2, \ldots, s_t^n\}$ be the signal profile at time t (i.e., signals observed by the agents in iteration t). Given an environmental state θ, the signal profile $\mathbf{s}_t$ is generated according to the joint distribution $\ell_1(s_t^1|\theta) \times \ell_2(s_t^2|\theta) \times \cdots \times \ell_n(s_t^n|\theta)$. In addition, let $s_{1,t}^i$ be the signal history up to time t for agent $i = 1, \cdots, n$, and let $\mathbf{s}_{1,t} = \{s_{1,t}^1, s_{1,t}^2, \ldots, s_{1,t}^n\}$ be the signal profile history up to time t.

4.3 Byzantine Consensus

In this section, we briefly review relevant exsting results on Byzantine consensus. Byzantine consensus has attracted significant attention [32, 39, 75, 33, 34, 53, 74]. While the past work mostly focuses on scalar inputs, the more general vector (or multi-dimensional) inputs have been studied recently [74, 75, 53]. Complete communication networks are considered in [74, 75], where tight conditions on the number of agents are identified. Incomplete communication networks are studied in [53]. Closer to the non-Bayesian learning problem is the class of *iterative approximate Byzantine consensus algorithms*, where each agent is only allowed to exchange information about its state with its neighbors. In particular, our learning algorithms build upon the *Byz-Iter* algorithm proposed in [53] and a simple algorithm proposed in [34] for iterative Byzantine consensus with vector inputs and scalar inputs, respectively, in incomplete networks. A matrix representation of the non-faulty agents' states evolution under *Byz-Iter* algorithm is provided by [53], which also captures the dynamics of the simple algorithm with scalar inputs in [34]. To make this chapter self-contained, in this section, we briefly review the algorithm *Byz-Iter* and its matrix representation.

4.3.1 Algorithm *Byz-Iter* [53]

Algorithm *Byz-Iter* is based on Tverberg's Theorem [90].

Theorem 18. *[90] Let f be a nonnegative integer. Let Y be a multiset containing vectors from $\mathbb{R}^m$ such that $|Y| \geq (m+1)f+1$. There exists a partition $Y_1, Y_2, \cdots, Y_{f+1}$ of Y such that Y_i is nonempty for $1 \leq i \leq f+1$, and the intersection of the convex hulls of Y_i's are nonempty, i.e., $\cap_{i=1}^{f+1} \mathsf{Conv}(Y_i) \neq \emptyset$, where $\mathsf{Conv}(Y_i)$ is the convex hull of Y_i for $i = 1, \cdots, f+1$.*

The proper partition in Theorem 18, and the points in $\cap_{i=1}^{f+1} \mathsf{Conv}(Y_i)$, are referred as *Tverberg partition of Y* and *Tverberg points of Y*, respectively.

For convenience of presenting our algorithm in Section 4.4, we present *Byz-Iter* (described in Algorithm 7) below using *One-Iter* (described in Algorithm 6) as a primitive. The parameter $\mathbf{x}^i$ passed to *One-Iter* at agent i, and $\mathbf{y}^i$ returned by *One-Iter* are both m-dimensional vectors. Let $\mathbf{v}^i$ be the state of agent i that will be iteratively updated, with $\mathbf{v}_t^i$ being the state at the end

of iteration t and $\mathbf{v}_0^i$ being the input of agent i. In each iteration $t \geq 1$, a non-faulty agent performs the steps in *One-Iter*. In particular, in the message receiving step, if a message is not received from some neighbor, that neighbor must be faulty, as the system is synchronous. In this case, the missing message values are set to some default value. Faulty agents may deviate from the algorithm specification arbitrarily. In *Byz-Iter*, the value returned by *One-Iter* at agent i is assigned to $\mathbf{v}_t^i$.

Algorithm 6: Algorithm *One-Iter* with input $\mathbf{x}^i$ at agent i

$Z^i \leftarrow \emptyset$;
Transmit $\mathbf{x}^i$ on all outgoing links;
Receive messages on all incoming links. % These message values form a multiset R^i of size $|\mathcal{I}_i|$.%
for *every* $C \subseteq R^i \cup \{\mathbf{x}^i\}$ *such that* $|C| = (m+1)f + 1$ **do**
  add to Z^i a *Tverberg point* of multiset C
end
Compute $\mathbf{y}^i$ as follows: $\mathbf{y}^i \leftarrow \frac{1}{1+|Z^i|}\left(\mathbf{x}^i + \sum_{\mathbf{z} \in Z^i} \mathbf{z}\right)$;
Return $\mathbf{y}^i$;

Algorithm 7: Algorithm *Byz-Iter* [53]: t-th iteration at agent i

$\mathbf{v}_t^i \leftarrow$ *One-Iter*$(\mathbf{v}_{t-1}^i)$;

Remark 3. *Note that for each agent $i \in \mathcal{N}$, the computation complexity per iteration is*

$$\Omega\left(\binom{|R^i \cup \{\mathbf{x}^i\}|}{(m+1)f+1}\right) = \Omega\left(\binom{|\mathcal{I}_i|+1}{(m+1)f+1}\right).$$

In the worst case, $||\mathcal{I}_i| + 1| = n$, and

$$\Omega\left(\binom{|\mathcal{I}_i|+1}{(m+1)f+1}\right) = \Omega\left(\binom{n}{(m+1)f+1}\right) = \Omega\left(\left(\frac{n}{\mathrm{e}}\right)^{(m+1)f+1}\right).$$

Since our first learning rule is based on Algorithm Byz-Iter*, the computation complexity of our first proposed algorithm is also high. Nevertheless, our first learning rule contains our main algorithmic ideas. More importantly, this learning rule can be improved such that the computation complexity per*

iteration per agent is $O(m^2 n \log n)$. Specifically, the improved learning rule adopts the scalar Byzantine consensus instead of the m–dimensional consensus.

4.3.2 Correctness of Algorithm *Byz-Iter*

We briefly summarize the aspects of correctness proof of Algorithm 7 from [53] that are necessary for our subsequent discussion. By using the Tverberg points in the update of $\mathbf{v}_t^i$ above, effectively, the extreme message values (that may potentially be sent by faulty agents) are trimmed away. Informally speaking, trimming certain messages can be viewed as ignoring (or removing) incoming links that carry the outliers. [53] shows that the effective communication network thus obtained can be characterized by a "reduced graph" of $G(\mathcal{V}, \mathcal{E})$, defined below. It is important to note that the non-faulty agents **do not** know the identity of the faulty agents.

Definition 17 (m–dimensional reduced graph)**.** *An m–dimensional reduced graph $\mathcal{H}(\mathcal{N}, \mathcal{E}_{\mathcal{F}})$ of $G(\mathcal{V}, \mathcal{E})$ is obtained by (i) removing all faulty nodes $\mathcal{F}$, and all the links incident on the faulty nodes $\mathcal{F}$; and (ii) for each non-faulty node (nodes in $\mathcal{N}$), removing up to mf additional incoming links.*

Definition 18. *A source component in any given m–dimensional reduced graph is a strongly connected component (of that reduced graph), which does not have any incoming links from outside that component.*

It turns out that the effective communication network is potentially time-varying (partly) due to time-varying behavior of faulty nodes. Assumption 1 below states a condition that is sufficient for reaching approximate Byzantine vector consensus using Algorithm 6 [53].

Assumption 1. *Every m–dimensional reduced graph of $G(\mathcal{V}, \mathcal{E})$ contains a unique source component.*

Let $\mathcal{C}_m$ be the set of all the m–dimensional reduced graph of $G(\mathcal{V}, \mathcal{E})$. Define $\chi_m \triangleq |\mathcal{C}_m|$. Since $G(\mathcal{V}, \mathcal{E})$ is finite, we have $\chi_m < \infty$. Let $\mathcal{H}_m \in \mathcal{C}_m$ be an m–dimensional reduced graph of $G(\mathcal{V}, \mathcal{E})$ with source component $\mathcal{S}_{\mathcal{H}_m}$. Define

$$\gamma_m \triangleq \min_{\mathcal{H}_m \in \mathcal{C}_m} |\mathcal{S}_{\mathcal{H}_m}|, \tag{4.1}$$

i.e., γ_m is the minimum source component size among all the m–dimensional reduced graphs. Note that $\gamma_m \geq 1$ if Assumption 1 holds for a given m.

Theorem 19. *[53] Suppose Assumption 1 holds for a given $m \geq 1$. Under Algorithm* Byz-Iter*, all the non-faulty agents (agents in $\mathcal{N}$) reach consensus asymptotically, i.e.,* $\lim_{t\to\infty} |\mathbf{v}_t^i - \mathbf{v}_t^j| = 0, \forall\, i, j \in \mathcal{N}$.

The proof of Theorem 19 relies crucially on a matrix representation of the state evolution.

4.3.3 Matrix Representation [53]

Let $|\mathcal{F}| = \phi$ (thus, $0 \leq \phi \leq f$). Without loss of generality, assume that agents 1 through $n - \phi$ are non-faulty, and agents $n - \phi + 1$ to n are Byzantine.

Lemma 11. *[53] Suppose Assumption 1 holds for a given $m \geq 1$. The state updates performed by the non-faulty agents in the t–th iteration ($t \geq 1$) can be expressed as*

$$\mathbf{v}_t^i = \sum_{j=1}^{n-\phi} \mathbf{A}_{ij}[t]\mathbf{v}_{t-1}^j, \tag{4.2}$$

where $\mathbf{A}[t] \in \mathbb{R}^{(n-\phi)\times(n-\phi)}$ is a row stochastic *matrix for which there exists an m–dimensional reduced graph $\mathcal{H}_m[t]$ with adjacency matrix $\mathbf{H}_m[t]$ such that $\mathbf{A}[t] \geq \beta_m \mathbf{H}_m[t]$, where $0 < \beta_m \leq 1$ is a constant that depends only on $G(\mathcal{V}, \mathcal{E})$.*

Let $\mathbf{\Phi}(t, r) \triangleq \mathbf{A}[t] \cdots \mathbf{A}[r]$ for $1 \leq r \leq t+1$. By convention, $\mathbf{\Phi}(t, t) = \mathbf{A}[t]$ and $\mathbf{\Phi}(t, t+1) = \mathbf{I}$. Note that $\mathbf{\Phi}(t, r)$ is a backward product. Using prior work on coefficients of ergodicity [47], under Assumption 1, it has been shown [53] that

$$\lim_{t\geq r,\ t\to\infty} \mathbf{\Phi}(t, r) = \mathbf{1}\pi(r), \tag{4.3}$$

where $\pi(r) \in \mathbb{R}^{n-\phi}$ is a row stochastic vector, and $\mathbf{1}$ is the column vector with each entry being 1. Recall that χ_m is the total number of m–dimensional reduced graphs of $G(\mathcal{V}, \mathcal{E})$, and β_m is defined in Lemma 11, and $\phi \triangleq |\mathcal{F}|$. The convergence rate in (4.3) is exponential.

Theorem 20. *[53] For all $t \geq r \geq 1$, it holds that $|\mathbf{\Phi}_{ij}(t,r) - \pi_j(r)| \leq (1-\beta_m^{\nu})^{\lceil \frac{t-r+1}{\nu} \rceil}$, where $\nu \triangleq \chi_m(n-\phi)$.*

Recall that γ_m is defined in (4.1). The next lemma is a consequence of the results in [53].

Lemma 12. *[53] For any $r \geq 1$, there exists a reduced graph $\mathcal{H}[r]$ with source component $\mathcal{S}_r$ such that $\pi_i(r) \geq \beta_m^{\chi_m(n-\phi)}$ for each $i \in \mathcal{S}_r$. In addition, $|\mathcal{S}_r| \geq \gamma_m$.*

4.3.4 Tight Topological Condition for Scalar Iterative Byzantine Consensus

The above analysis shows that Assumption 1 is sufficient for achieving Byzantine consensus iteratively. For the special case when $m = 1$,(i.e., the inputs provided at individual non-faulty agents are scalars) it has been shown [34] that Assumption 1 is also necessary.

Theorem 21. *[34] For scalar inputs, iterative approximate Byzantine consensus is achievable among non-faulty agents if and only if every 1-dimensional reduced graph of $G(\mathcal{V},\mathcal{E})$ contains only one source component.*

Moreover, the following simple algorithm (Algorithm 8) works under Assumption 1 when $m = 1$.

Algorithm 8: Algorithm Scalar Byzantine Consensus: iteration $t \geq 1$ [34]

1. Transmit $v^i[t-1]$ on all outgoing links;
2. Receive messages on all incoming links. % These message values $w_j[t]$ for each $j \in \mathcal{I}_i$ form a multiset $R^i[t]$ of size $|\mathcal{I}_i|$. %
3. Sort the received values $w_j[t]$ for each $j \in \mathcal{I}_i$ in a non-decreasing order;
4. Remove the largest f values and the smallest f values. % Denote the set of indices of incoming neighbors whose values have not been removed at iteration t by $\mathcal{I}_i^*[t]$.%
5. Update v^i as follows: $v^i[t] \leftarrow \frac{\sum_{j \in \mathcal{I}_i^*[t]} w_j[t] + v^i[t-1]}{1+|\mathcal{I}_i^*[t]|}$;

In addition, it has been show that the dynamic of the non-faulty agents states admits the same matrix representation as in Subsection 4.3.3 with the

reduced graph being the 1–dimensional reduced graph defined in Definition 17.

With the above background on Byzantine vector consensus, we are now ready to present our first algorithm and its analysis.

4.4 Byzantine Fault-Tolerant Non-Bayesian Learning (BFL)

In this section, we present our first learning rule, named Byzantine Fault-Tolerant Non-Bayesian Learning (BFL). In BFL, each agent i maintains a stochastic score vector $\mu^i \in \mathbb{R}^m$. Since no signals are observed before the execution of an algorithm, the score μ^i is often initially set to be uniform over the set Θ, i.e., $(\mu_0^i(\theta_1), \mu_0^i(\theta_1), \ldots, \mu_0^i(\theta_m))^T = \left(\frac{1}{m}, \ldots, \frac{1}{m}\right)^T$. Recall that θ^* is the true environmental state. We say the networked agents collaboratively learn θ^* if for every non-faulty agent $i \in \mathcal{N}$,

$$\lim_{t\to\infty} \mu_t^i(\theta^*) \;=\; 1, \quad \text{and } \lim_{t\to\infty} \mu_t^i(\theta) \;=\; 0 \;\text{ for } \theta \neq \theta^* \;\; a.s. \tag{4.4}$$

where *a.s.* denotes *almost surely*.

BFL is a modified version of the geometric averaging update rule that has been investigated in previous work [83, 84, 86, 88]. In particular, we modify the averaging rule to take into account Byzantine faults. More importantly, in each iteration, we use the likelihood of the *cumulative* local observations (instead of the likelihood of the *current* observation only) to update the local scores.

For $t \geq 1$, the steps to be performed by agent i in the t–th iteration are listed below, where log on vector is performed element-wise. Note that faulty agents can deviate from the algorithm specification. The algorithm below uses *One-Iter* presented in the previous section as a primitive. Recall that $s_{1,t}^i$ is the cumulative local observations up to iteration t. Since the observations are *i.i.d.*, it holds that $\ell_i(s_{1,t}^i|\theta) = \prod_{r=1}^t \ell_i(s_r^i|\theta)$. So $\ell_i(s_{1,t}^i|\theta)$ can be computed iteratively in Algorithm 9.

The main differences between Algorithm 9 and the algorithms in [83, 84, 86, 88] are that (i) our algorithm uses a Byzantine consensus iteration as a primitive (in line 1), and (ii) $\ell_i(s_{1,t}^i|\theta)$ used in line 5 is the likelihood for

Algorithm 9: BFL: Iteration $t \geq 1$ at agent i

1 $\eta_t^i \leftarrow$ *One-Iter*$(\log \mu_{t-1}^i)$;
2 Observe s_t^i;
3 **for** $\theta \in \Theta$ **do**
4 $\quad \ell_i(s_{1,t}^i|\theta) \leftarrow \ell_i(s_t^i|\theta)\, \ell_i(s_{1,t-1}^i|\theta)$;
5 $\quad \mu_t^i(\theta) \leftarrow \frac{\ell_i(s_{1,t}^i|\theta) \exp\left(\eta_t^i(\theta)\right)}{\sum_{p=1}^m \ell_i(s_{1,t}^i|\theta_p) \exp\left(\eta_t^i(\theta_p)\right)}$;
6 **end**

observations from iteration 1 to t (the previous algorithms instead use $\ell_i(s_t^i|\theta)$ here). Observe that the consensus step is being performed on log of the scores, with the result being stored as η_t^i (in line 1) and used in line 4 to compute the new scores.

Recalling the matrix representation of the *Byz-Iter* algorithm as per Lemma 11, we can write the following equivalent representation of line 1 of Algorithm 9:

$$\eta_t^i(\theta) \;=\; \sum_{j=1}^{n-\phi} \mathbf{A}_{ij}[t] \log \mu_{t-1}^j(\theta) = \log \prod_{j=1}^{n-\phi} \mu_{t-1}^j(\theta)^{\mathbf{A}_{ij}[t]}, \qquad \forall \theta \in \Theta, \;\; (4.5)$$

where $\mathbf{A}[t]$ is a row stochastic matrix whose properties are specified in Lemma 11. Note that $\mu_t^i(\theta)$ is **random** for each $i \in \mathcal{N}$ and $t \geq 1$, as it is updated according to local random observations. Since the consensus is performed over $\log \mu_t^i \in \mathbb{R}^m$, the update matrix $\mathbf{A}[t]$ is also **random**. In particular, for each $t \geq 1$, matrix $\mathbf{A}[t]$ is dependent on *all the cumulative observations over the network* up to iteration t. This dependency makes it non-trivial to adapt analysis from previous algorithms to our setting. In addition, adopting the local cumulative observation likelihood makes the analysis with Byzantine faults easier.

4.4.1 Identifiability

In the absence of agent failures [11], for the networked agents to detect the true hypothesis θ^*, it is sufficient to assume that $G(\mathcal{V}, \mathcal{E})$ is strongly connected, and that θ^* is globally identifiable. That is, for any $\theta \neq \theta^*$, there exists a node $j \in \mathcal{V}$ such that the Kullback-Leiber divergence between the true marginal $\ell_j(\cdot|\theta^*)$ and the marginal $\ell_j(\cdot|\theta)$, denoted by $D\left(\ell_j(\cdot|\theta^*)||\ell_j(\cdot|\theta)\right)$, is

nonzero; equivalently,

$$\sum_{j\in\mathcal{V}} D\left(\ell_j(\cdot|\theta^*)||\ell_j(\cdot|\theta)\right) \neq 0, \tag{4.6}$$

where $D\left(\ell_j(\cdot|\theta^*)||\ell_j(\cdot|\theta)\right)$ is defined as

$$D\left(\ell_j(\cdot|\theta^*)||\ell_j(\cdot|\theta)\right) \triangleq \sum_{w_j\in\mathcal{S}_j} \ell_j(w_j|\theta^*)\log\frac{\ell_j(w_j|\theta^*)}{\ell_j(w_j|\theta)}. \tag{4.7}$$

Since θ^* may change from execution to execution, (4.6) is required to hold for any choice of θ^*. Intuitively speaking, if any pair of states θ_1 and θ_2 can be distinguished by at least one agent in the network, then sufficient exchange of local scores over strongly connected network will enable every agent distinguish θ_1 and θ_2. However, in the presence of Byzantine agents, a stronger global identifiability condition is required. The following assumption builds upon Assumption 1.

Assumption 2. *Suppose that Assumption 1 holds for $m = |\Theta|$. For any $\theta \neq \theta^*$, and for any m–dimensional reduced graph $\mathcal{H}$ of $G(\mathcal{V},\mathcal{E})$ with $\mathcal{S}_\mathcal{H}$ denoting the unique source component, the following holds:*

$$\sum_{j\in\mathcal{S}_\mathcal{H}} D\left(\ell_j(\cdot|\theta^*) \parallel \ell_j(\cdot|\theta)\right) \neq 0. \tag{4.8}$$

In contrast to (4.6), where the summation is taken over all the agents in the network, in (4.8), the summation is taken over agents in the source component only. Intuitively, the condition imposed by Assumption 2 is that all the agents in the source component can detect the true state θ^* collaboratively. If iterative consensus is achieved, the accurate score can be propagated from the source component to every other non-faulty agent in the network.

Remark 4. *We will show later that when Assumption 2 holds, the BFL algorithm enables all the non-faulty agents concentrate their scores on the true state θ^* almost surely. That is, Assumption 2 is a sufficient condition for a consensus-based non-Bayesian learning algorithm to exist. However, Assumption 2 is not necessary, observing that Assumption 1 (upon which Assumption 2 builds) is not necessary for m-dimensional Byzantine consensus algorithms to exist. As illustrated by our second learning rule (described*

later), the adoption of m-dimensional Byzantine consensus primitives is not necessary.

4.4.2 Convergence Results

Our proof parallels the structure of a proof in [83], but with some key differences to take into account our update rule for the score vector.

For any $\theta_1, \theta_2 \in \Theta$, and any $i \in \mathcal{V}$, define $\boldsymbol{\psi}_t^i(\theta_1, \theta_2)$ and $\mathcal{L}_t(\theta_1, \theta_2)$ as follows:

$$\boldsymbol{\psi}_t^i(\theta_1, \theta_2) \triangleq \log \frac{\mu_t^i(\theta_1)}{\mu_t^i(\theta_2)}, \quad \mathcal{L}_t^i(\theta_1, \theta_2) \triangleq \log \frac{\ell_i(s_t^i|\theta_1)}{\ell_i(s_t^i|\theta_2)}. \tag{4.9}$$

To show that Algorithm 9 solves (4.4), we will show that $\boldsymbol{\psi}_t^i(\theta, \theta^*) \xrightarrow{\text{a.s.}} -\infty$ for $\theta \neq \theta^*$, which implies that $\mu_t^i(\theta) \xrightarrow{\text{a.s.}} 0$ for all $\theta \neq \theta^*$ and for all $i \in \mathcal{N}$, i.e., all non-faulty agents asymptotically concentrate their scores on the true hypothesis θ^*. We do this by investigating the dynamics of scores, which is represented compactly in a matrix form.

For each $\theta \neq \theta^*$, and each $i \in \mathcal{N} = \{1, 2, \cdots, n-\phi\}$, we have

$$\begin{aligned}
\boldsymbol{\psi}_t^i(\theta, \theta^*) = \log \frac{\mu_t^i(\theta)}{\mu_t^i(\theta^*)} &\overset{(a)}{=} \log \left(\prod_{j=1}^{n-\phi} \left(\frac{\mu_{t-1}^j(\theta)}{\mu_{t-1}^j(\theta^*)} \right)^{\mathbf{A}_{ij}[t]} \times \frac{\ell_i(s_{1,t}^i|\theta)}{\ell_i(s_{1,t}^i|\theta^*)} \right) \\
&= \sum_{j=1}^{n-\phi} \mathbf{A}_{ij}[t] \log \frac{\mu_{t-1}^j(\theta)}{\mu_{t-1}^j(\theta^*)} + \log \frac{\ell_i(s_{1,t}^i|\theta)}{\ell_i(s_{1,t}^i|\theta^*)} \\
&= \sum_{j=1}^{n-\phi} \mathbf{A}_{ij}[t] \boldsymbol{\psi}_{t-1}^j(\theta, \theta^*) + \sum_{r=1}^{t} \mathcal{L}_r^i(\theta, \theta^*),
\end{aligned} \tag{4.10}$$

where equality (a) follows from (4.5) and the update of μ^i in Algorithm 9, and the last equality follows from (4.9) and the fact that the local observations are *i.i.d.* for each agent.

Let $\boldsymbol{\psi}_t(\theta, \theta^*) \in \mathbb{R}^{n-\phi}$ be the vector that stacks $\boldsymbol{\psi}_t^i(\theta, \theta^*)$, with the i–th entry being $\boldsymbol{\psi}_t^i(\theta, \theta^*)$ for all $i \in \mathcal{N}$. The evolution of $\boldsymbol{\psi}(\theta, \theta^*)$ can be compactly

written as

$$\boldsymbol{\psi}_t(\theta,\theta^*) = \mathbf{A}[t]\boldsymbol{\psi}_{t-1}(\theta,\theta^*) + \sum_{r=1}^{t} \mathcal{L}_r(\theta,\theta^*). \tag{4.11}$$

Expanding (4.11), we get

$$\boldsymbol{\psi}_t(\theta,\theta^*) = \boldsymbol{\Phi}(t,1)\boldsymbol{\psi}_0(\theta,\theta^*) + \sum_{r=1}^{t} \boldsymbol{\Phi}(t,r+1) \sum_{k=1}^{r} \mathcal{L}_k(\theta,\theta^*). \tag{4.12}$$

For each $\theta \in \Theta$ and $i \in \mathcal{V}$, define $H_i(\theta,\theta^*) \in \mathbb{R}^{n-\phi}$ as

$$\begin{aligned} H_i(\theta,\theta^*) &\triangleq \sum_{w_i \in \mathcal{S}_i} \ell_i(w_i|\theta^*) \log \frac{\ell_i(w_i \mid \theta)}{\ell_i(w_i \mid \theta^*)} \\ &= -D(\ell_i(\cdot|\theta^*) \parallel \ell_i(\cdot|\theta)) \quad \text{by (4.7)} \\ &\leq 0. \end{aligned} \tag{4.13}$$

Let $\mathcal{H} \in \mathcal{C}$ be an arbitrary reduced graph with source component $\mathcal{S}_{\mathcal{H}}$. Define C_0 and C_1 as

$$-C_0 \triangleq \min_{i \in \mathcal{V}} \min_{\theta_1,\theta_2 \in \Theta; \theta_1 \neq \theta_2} \min_{w_i \in \mathcal{S}_i} \left(\log \frac{\ell_i(w_i|\theta_1)}{\ell_i(w_i|\theta_2)} \right), \tag{4.14}$$

$$C_1 \triangleq \min_{\mathcal{H} \in \mathcal{C}} \min_{\theta,\theta^* \in \Theta; \theta \neq \theta^*} \sum_{i \in \mathcal{S}_{\mathcal{H}}} D(\ell_i(\cdot|\theta^*) \parallel \ell_i(\cdot|\theta)). \tag{4.15}$$

The constant C_0 serves as a universal upper bound on $|\log \frac{\ell_i(w_i|\theta_1)}{\ell_i(w_i|\theta_2)}|$ for all choices of θ_1 and θ_2, and for all signals. Intuitively, the constant C_1 is the minimal detection capability of the source component under Assumption 2.

Due to $|\Theta| = m < \infty$ and $|\mathcal{S}_i| < \infty$ for each $i \in \mathcal{N}$, we know that $C_0 < \infty$. Besides, it is easy to see that $-C_0 \leq 0$ (thus, $C_0 \geq 0$). In addition, under Assumption 2, we have $C_1 > 0$.

Now we present a key lemma for our main theorem.

Lemma 13. *Under Assumption 2, for any $\theta \neq \theta^*$, it holds that*

$$\frac{1}{t^2} \sum_{r=1}^{t} \left(\sum_{j=1}^{n-\phi} \left[\boldsymbol{\Phi}_{ij}(t,r+1) \sum_{k=1}^{r} \mathcal{L}_k^j(\theta,\theta^*) \right] - r \sum_{j=1}^{n-\phi} \pi_j(r+1) H_j(\theta,\theta^*) \right) \xrightarrow{\text{a.s.}} 0. \tag{4.16}$$

Proof. The proof of Lemma 13 is significantly different from the analogous lemma in [83].

By (4.9), we have

$$\left|\mathcal{L}_r^i(\theta,\theta^*)\right| = \left|\log\frac{\ell_i(s_t^i|\theta)}{\ell_i(s_t^i|\theta^*)}\right| \leq \max_{i\in\mathcal{V}}\max_{\theta_1,\theta_2\in\Theta;\theta_1\neq\theta_2}\max_{w_i\in\mathcal{S}_i}\left|\log\frac{\ell_i(w_i|\theta_1)}{\ell_i(w_i|\theta_2)}\right|.$$

Note that $\max_{i\in\mathcal{V}}\max_{\theta_1,\theta_2\in\Theta;\theta_1\neq\theta_2}\max_{w_i\in\mathcal{S}_i}\left|\log\frac{\ell_i(w_i|\theta_1)}{\ell_i(w_i|\theta_2)}\right|$ is symmetric in θ_1 and θ_2. Thus,

$$\begin{aligned}
\left|\mathcal{L}_r^i(\theta,\theta^*)\right| &\leq \max_{i\in\mathcal{V}}\max_{\theta_1,\theta_2\in\Theta;\theta_1\neq\theta_2}\max_{w_i\in\mathcal{S}_i}\left|\log\frac{\ell_i(w_i|\theta_1)}{\ell_i(w_i|\theta_2)}\right| \\
&= \max_{i\in\mathcal{V}}\max_{\theta_1,\theta_2\in\Theta;\theta_1\neq\theta_2}\max_{w_i\in\mathcal{S}_i}\log\frac{\ell_i(w_i|\theta_1)}{\ell_i(w_i|\theta_2)} \\
&= \max_{i\in\mathcal{V}}\max_{\theta_1,\theta_2\in\Theta;\theta_1\neq\theta_2}\max_{w_i\in\mathcal{S}_i}-\log\frac{\ell_i(w_i|\theta_2)}{\ell_i(w_i|\theta_1)} \\
&= -\min_{i\in\mathcal{V}}\min_{\theta_1,\theta_2\in\Theta;\theta_1\neq\theta_2}\min_{w_i\in\mathcal{S}_i}\log\frac{\ell_i(w_i|\theta_2)}{\ell_i(w_i|\theta_1)} = -(-C_0) = C_0 < \infty.
\end{aligned} \tag{4.17}$$

Thus, adding and subtracting $\frac{1}{t^2}\sum_{r=1}^{t}\sum_{j=1}^{n-\phi}\pi_j(r+1)\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*)$ from the first term on the right hand side of (4.27), we can get

$$\begin{aligned}
&\frac{1}{t^2}\sum_{r=1}^{t}\left(\sum_{j=1}^{n-\phi}\left[\mathbf{\Phi}_{ij}(t,r+1)\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*)\right] - \pi_j(r+1)r\sum_{j=1}^{n-\phi}H_j(\theta,\theta^*)\right) \\
&= \frac{1}{t^2}\sum_{r=1}^{t}\sum_{j=1}^{n-\phi}\left[(\mathbf{\Phi}_{ij}(t,r+1)-\pi_j(r+1))\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*)\right] \\
&\quad + \frac{1}{t^2}\sum_{r=1}^{t}\sum_{j=1}^{n-\phi}\left[\pi_j(r+1)\left(\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*) - rH_j(\theta,\theta^*)\right)\right].
\end{aligned} \tag{4.18}$$

For the first term of the right-hand side of (4.18), we have

$$\begin{aligned}
&\frac{1}{t^2}\left|\sum_{r=1}^{t}\sum_{j=1}^{n-\phi}(\mathbf{\Phi}_{ij}(t,r+1)-\pi_j(r+1))\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*)\right| \\
&\leq \frac{1}{t^2}\sum_{r=1}^{t}\sum_{j=1}^{n-\phi}|\mathbf{\Phi}_{ij}(t,r+1)-\pi_j(r+1)|\sum_{k=1}^{r}\left|\mathcal{L}_k^j(\theta,\theta^*)\right| \\
&\leq \frac{1}{t^2}\sum_{r=1}^{t}\sum_{j=1}^{n-\phi}|\mathbf{\Phi}_{ij}(t,r+1)-\pi_j(r+1)|\,rC_0 \quad \text{by (4.17)} \\
&\leq \frac{1}{t^2}\sum_{r=1}^{t}\sum_{j=1}^{n-\phi}(1-\beta^\nu)^{\lceil\frac{t-r}{\nu}\rceil}rC_0 \quad \text{by Theorem 20} \\
&\leq \frac{1}{t^2}\left(t(n-\phi)C_0\right)\sum_{r=1}^{t}(1-\beta^\nu)^{\lceil\frac{t-r}{\nu}\rceil} \\
&\leq \frac{(n-\phi)C_0}{(1-\beta^\nu)(1-(1-\beta^\nu)^{\frac{1}{\nu}})t},
\end{aligned} \tag{4.19}$$

where, with a bit abuse of notation, β is used to represent β_m.

Thus, for every sample path, we have

$$\frac{1}{t^2}\sum_{r=1}^{t}\sum_{j=1}^{n-\phi}(\mathbf{\Phi}_{ij}(t,r+1)-\pi_j(r+1))\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*)\to 0. \tag{4.20}$$

For the second term of the right hand side of (4.18), we will show that

$$\frac{1}{t^2}\sum_{r=1}^{t}\sum_{j=1}^{n-\phi}\pi_j(r+1)\left(\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*)-rH_j(\theta,\theta^*)\right)\xrightarrow{\text{a.s.}}0, \tag{4.21}$$

i.e., almost surely for any $\epsilon>0$ there exists sufficiently large $t(\epsilon)$ such that $\forall\, t\geq t(\epsilon)$,

$$\frac{1}{t^2}\left|\sum_{r=1}^{t}\sum_{j=1}^{n-\phi}\pi_j(r+1)\left(\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*)-rH_j(\theta,\theta^*)\right)\right| \;\leq\; \epsilon. \tag{4.22}$$

We prove this by dividing r into two ranges $r\in\{1,\cdots,\sqrt{t}\}$ and $r\in\{\sqrt{t}+$

$1, \cdots, t\}$, i.e.,

$$\begin{aligned}
&\frac{1}{t^2}\sum_{r=1}^{t}\sum_{j=1}^{n-\phi}\pi_j(r+1)\left(\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*)-rH_j(\theta,\theta^*)\right)\\
&=\frac{1}{t^2}\sum_{r=1}^{\sqrt{t}}\sum_{j=1}^{n-\phi}\pi_j(r+1)\left(\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*)-rH_j(\theta,\theta^*)\right)\\
&\quad+\frac{1}{t^2}\sum_{r=\sqrt{t}+1}^{t}\sum_{j=1}^{n-\phi}\pi_j(r+1)\left(\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*)-rH_j(\theta,\theta^*)\right). \qquad (4.23)
\end{aligned}$$

For the first term of the right hand side of (4.23), we have

$$\begin{aligned}
&\frac{1}{t^2}\left|\sum_{r=1}^{\sqrt{t}}\sum_{j=1}^{n-\phi}\pi_j(r+1)\left(\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*)-rH_j(\theta,\theta^*)\right)\right|\\
&\le\frac{1}{t^2}\sum_{r=1}^{\sqrt{t}}\sum_{j=1}^{n-\phi}\pi_j(r+1)\,(2rC_0)\quad\text{by (4.13) and (4.17)}\\
&=\frac{1}{t^2}\,(2C_0)\sum_{r=1}^{\sqrt{t}}r\\
&\le C_0\left(\frac{1}{t}+\frac{1}{t^{\frac{3}{2}}}\right).
\end{aligned}$$

Thus, there exists $t_1(\epsilon)$ such that for all $t \ge t_1(\epsilon)$, it holds that

$$\frac{1}{t^2}\left|\sum_{r=1}^{\sqrt{t}}\sum_{j=1}^{n-\phi}\pi_j(r+1)\left(\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*)-rH_j(\theta,\theta^*)\right)\right|\le\frac{\epsilon}{2}.$$

For the second term of the right hand side of (4.23), we have

$$\begin{aligned}
&\frac{1}{t^2}\sum_{r=\sqrt{t}+1}^{t}\sum_{j=1}^{n-\phi}\pi_j(r+1)\left(\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*)-rH_j(\theta,\theta^*)\right)\\
&=\frac{1}{t}\sum_{r=\sqrt{t}+1}^{t}\sum_{j=1}^{n-\phi}\pi_j(r+1)\frac{r}{t}\left(\frac{1}{r}\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*)-H_j(\theta,\theta^*)\right).
\end{aligned}$$

Since $\mathcal{L}_k^j(\theta, \theta^*)$'s are i.i.d., from Strong LLN, we know that

$$\frac{1}{r}\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*) - H_j(\theta,\theta^*) \xrightarrow{\text{a.s.}} 0. \tag{4.24}$$

That is, with probability 1, the sample path converges. Now, focus on each convergent sample path. For sufficiently large $t_1(\epsilon)$, it holds that for any $r \geq t_1(\epsilon)$,

$$\left|\frac{1}{r}\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*) - H_j(\theta,\theta^*)\right| \leq \frac{\epsilon}{2}.$$

Recall that $r \geq \sqrt{t}$. Thus, we know that there exists sufficiently large $t_2(\epsilon)$ such that $\forall\, t \geq t_2(\epsilon)$, $r \geq \sqrt{t}$ is large enough and

$$\left|\frac{1}{r}\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*) - H_j(\theta,\theta^*)\right| \leq \frac{\epsilon}{2}.$$

Then, we have $\forall\, t \geq t_2(\epsilon)$,

$$\begin{aligned}
&\frac{1}{t^2}\left|\sum_{r=\sqrt{t}+1}^{t}\sum_{j=1}^{n-\phi}\pi_j(r+1)\left(\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*) - rH_j(\theta,\theta^*)\right)\right| \\
&\leq \frac{1}{t}\sum_{r=\sqrt{t}+1}^{t}\sum_{j=1}^{n-\phi}\pi_j(r+1)\frac{r}{t}\left|\frac{1}{r}\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*) - H_j(\theta,\theta^*)\right| \\
&\leq \frac{1}{t}\sum_{r=\sqrt{t}+1}^{t}\sum_{j=1}^{n-\phi}\pi_j(r+1)\frac{r}{t}\frac{\epsilon}{2} \\
&= \frac{1}{t}\sum_{r=\sqrt{t}+1}^{t}\frac{r}{t}\frac{\epsilon}{2} = \frac{\epsilon}{2}\frac{1}{t^2}\sum_{r=\sqrt{t}+1}^{t} r \\
&= \frac{\epsilon}{2}\frac{1}{t^2}\left(\sum_{r=1}^{t} r - \sum_{r=1}^{\sqrt{t}} r\right) = \frac{\epsilon}{4}\frac{1}{t^2}\left(t^2 - \sqrt{t}\right) \leq \frac{\epsilon}{2}.
\end{aligned}$$

Therefore, for any $\epsilon > 0$, there exists $\max\{t_1(\epsilon), t_2(\epsilon)\}$, such that for any $t \geq \max\{t_1(\epsilon), t_2(\epsilon)\}$,

$$\frac{1}{t^2}\left|\sum_{r=1}^{t}\sum_{j=1}^{n-\phi}\pi_j(r+1)\left(\sum_{k=1}^{r}\mathcal{L}_k^j(\theta,\theta^*) - rH_j(\theta,\theta^*)\right)\right| \leq \epsilon,$$

for every convergent sample path. In addition, from (4.24), we know a sample path is convergent with probability 1. Thus, by the definition of convergence, (4.22) holds almost surely.

Equations (4.18), (4.20), (4.21) together prove Lemma 13.

□

Theorem 22. *When Assumption 2 holds, each non-faulty agent $i \in \mathcal{N}$ will concentrate its score on the true hypothesis θ^* almost surely, i.e., $\mu_t^i(\theta) \xrightarrow{\text{a.s.}} 0$ for all $\theta \neq \theta^*$.*

Proof. Consider any $\theta \neq \theta^*$. Recall from (4.12) that

$$\begin{aligned}\boldsymbol{\psi}_t(\theta, \theta^*) &= \boldsymbol{\Phi}(t,1)\boldsymbol{\psi}_0(\theta,\theta^*) + \sum_{r=1}^{t} \boldsymbol{\Phi}(t, r+1) \sum_{k=1}^{r} \mathcal{L}_k(\theta, \theta^*) \\ &= \sum_{r=1}^{t} \boldsymbol{\Phi}(t, r+1) \sum_{k=1}^{r} \mathcal{L}_k(\theta, \theta^*).\end{aligned}$$

The last equality holds as μ_0^i is uniform, and $\boldsymbol{\psi}_0^i(\theta, \theta^*) = 0$ for each $i \in \mathcal{N}$. Since the supports of $\ell_i(\cdot|\theta)$ and $\ell_i(\cdot|\theta^*)$ are the whole signal space $\mathcal{S}_i$ for each agent $i \in \mathcal{N}$, it holds that $\left|\frac{\ell_i(w_i|\theta)}{\ell_i(w_i|\theta^*)}\right| < \infty$ for each $w_i \in \mathcal{S}_i$, and

$$0 \geq H_i(\theta, \theta^*) \geq \min_{w_i \in \mathcal{S}_i} \left(\log \frac{\ell_i(w_i|\theta)}{\ell_i(w_i|\theta^*)} \right) \geq \ -C_0 > -\infty. \tag{4.25}$$

By (4.25), we know that $|\sum_{j=1}^{n-\phi} \pi_j(r+1) H_j(\theta, \theta^*)| \leq C_0 < \infty$. Due to the finiteness of $\sum_{j=1}^{n-\phi} \pi_j(r+1) H_j(\theta, \theta^*)$, we are able to add and subtract $r\mathbf{1} \sum_{j=1}^{n-\phi} \pi_j(r+1) H_j(\theta, \theta^*)$ from (4.12), where $\mathbf{1}$ is a $n-\phi$ dimensional vector with each entry being 1.

We get

$$\begin{aligned}\boldsymbol{\psi}_t(\theta, \theta^*) = &\sum_{r=1}^{t} \left(\boldsymbol{\Phi}(t, r+1) \sum_{k=1}^{r} \mathcal{L}_k(\theta, \theta^*) - r\mathbf{1} \sum_{j=1}^{n-\phi} \pi_j(r+1) H_j(\theta, \theta^*) \right) \\ &+ \sum_{r=1}^{t} r\mathbf{1} \sum_{j=1}^{n-\phi} \pi_j(r+1) H_j(\theta, \theta^*).\end{aligned} \tag{4.26}$$

For each $i \in \mathcal{N}$, we have

$$\boldsymbol{\psi}_t^i(\theta, \theta^*) = \sum_{r=1}^{t} \left(\sum_{j=1}^{n-\phi} \boldsymbol{\Phi}_{ij}(t, r+1) \sum_{k=1}^{r} \mathcal{L}_k^j(\theta, \theta^*) - r \sum_{j=1}^{n-\phi} \pi_j(r+1) H_j(\theta, \theta^*) \right) + \sum_{r=1}^{t} r \sum_{j=1}^{n-\phi} \pi_j(r+1) H_j(\theta, \theta^*). \tag{4.27}$$

To show $\lim_{t\to\infty} \mu_t^i(\theta) \xrightarrow{\text{a.s.}} 0$ for $\theta \neq \theta^*$, it is enough to show $\boldsymbol{\psi}_t^i(\theta, \theta^*) \xrightarrow{\text{a.s.}} -\infty$. Our convergence proof has similar structure as the analysis in [83]. From Lemma 13, we know that

$$\frac{1}{t^2} \sum_{r=1}^{t} \left(\sum_{j=1}^{n-\phi} \left[\boldsymbol{\Phi}_{ij}(t, r+1) \sum_{k=1}^{r} \mathcal{L}_k^j(\theta, \theta^*) \right] - r \sum_{j=1}^{n-\phi} \pi_j(r+1) H_j(\theta, \theta^*) \right) \xrightarrow{\text{a.s.}} 0. \tag{4.28}$$

Next we show that the second term of the right-hand side of (4.27) decreases quadratically in t.

$$\begin{aligned} \sum_{r=1}^{t} r \sum_{j=1}^{n-\phi} \pi_j(r+1) H_j(\theta, \theta^*) &\leq \sum_{r=1}^{t} r \sum_{j \in \mathcal{S}_r} \pi_j(r+1) H_j(\theta, \theta^*) \qquad \text{by (4.13)} \\ &\leq \sum_{r=1}^{t} r \beta^{\chi(n-\phi)} \sum_{j \in \mathcal{S}_r} H_j(\theta, \theta^*) \qquad \text{by Lemma 12} \\ &\leq -\sum_{r=1}^{t} r \beta^{\chi(n-\phi)} C_1 \qquad \text{by (4.15) and (4.13)} \\ &\leq -\frac{t^2}{2} \beta^{\chi(n-\phi)} C_1. \end{aligned} \tag{4.29}$$

Therefore, by (4.27), (4.28) and (4.29), almost surely, the following holds:

$$\lim_{t\to\infty} \frac{1}{t^2} \psi_t^i(\theta, \theta^*) \leq -\frac{1}{2} \beta^{\chi(n-\phi)} C_1.$$

Therefore, we have $\psi_t^i(\theta, \theta^*) \xrightarrow{\text{a.s.}} -\infty$ and $\mu_t^i(\theta) \xrightarrow{\text{a.s.}} 0$ for $i \in \mathcal{N}$ and $\theta \neq \theta^*$, proving Theorem 22.

□

4.5 Improved BFL

To reduce the computation complexity per iteration in general, and to identify an improved global identifiability of the network for any consensus-based learning rule of interest to learn the true state, we propose a modification of the above learning rule, which works under much weaker network topology and global identifiability condition.

We decompose the m-ary hypothesis testing problem into $m(m-1)$ (ordered) binary hypothesis testing problems. For each pair of hypotheses θ_1 and θ_2, each non-faulty agent updates the likelihood ratio of θ_1 over θ_2 as follows. Let $r_t^i(\theta_1, \theta_2)$ be the "log likelihood ratio" of θ_1 over θ_2 kept by agent i at the end of iteration t. Our improved learning rule applies consensus procedures to log likelihood ratio, i.e., $r_t^i(\theta_1, \theta_2)$, which is a scalar. For Algorithm 10, we only require scalar iterative Byzantine (approximate) consensus among the non-faulty agents to be achievable.

Assumption 3. *Suppose that every* 1*-dimensional reduced graph of* $G(\mathcal{V}, \mathcal{E})$ *contains only one source component. For any* $\theta \neq \theta^*$*, and for any* 1*-dimensional reduced graph* $\mathcal{H}_1$ *of* $G(\mathcal{V}, \mathcal{E})$ *with* $\mathcal{S}_{\mathcal{H}_1}$ *denoting the unique source component, the following holds:*

$$\sum_{j \in \mathcal{S}_{\mathcal{H}_1}} D\left(\ell_j(\cdot|\theta^*) \parallel \ell_j(\cdot|\theta)\right) \neq 0. \tag{4.30}$$

For each iteration, the computation complexity per agent (non-faulty) can be calculated as follows. The cost-dominant procedure in each iteration is sorting the received log likelihood ratios, which takes $O(n \log n)$ operations. In total, we have $m(m-1)$ order pairs of hypotheses. Thus, the total computation per agent per iteration is $O(m^2 n \log n)$.

Theorem 23. *Suppose Assumption 3 holds. Under Algorithm 10, for any* $\theta \neq \theta^*$*, the following holds:*

$$r_t^i(\theta^*, \theta) \xrightarrow{\text{a.s.}} +\infty, \text{ and } \quad r_t^i(\theta, \theta^*) \xrightarrow{\text{a.s.}} -\infty.$$

Proof. By [38], we know that for each pair of hypotheses θ_1 and θ_2, there

Algorithm 10: Pairwise Learning

Initialization: **for** $\theta_1, \theta_2 \in \Theta$, *and* $\theta_1 \neq \theta_2$ **do**
  $r_0^i(\theta_1, \theta_2) \leftarrow 0$;
end
while $t \geq 1$ **do**
  for $\theta_1, \theta_2 \in \Theta$, *and* $\theta_1 \neq \theta_2$ **do**
    Transmit current score vector $r_{t-1}^i(\theta_1, \theta_2)$ on all outgoing edges;
    Wait until a private signal s_t^i is observed and log likelihood ratios $\widetilde{r}_{t-1}^j(\theta_1, \theta_2)$ are received from all incoming neighbors $\mathcal{I}_i$;
    Sort the received log likelihood ratios $\widetilde{r}_{t-1}^j(\theta_1, \theta_2)$ in a non-decreasing order, and remove the smallest f values and the largest f values. % Denote the set of indices of incoming neighbors whose ratios have not been removed at iteration t by $\mathcal{I}_i^*[t]$.%
    $r_t^i(\theta_1, \theta_2) \leftarrow \frac{\sum_{j \in \mathcal{I}_i^*[t]} \widetilde{r}_{t-1}^j(\theta_1, \theta_2) + r_{t-1}^i(\theta_1, \theta_2)}{|\mathcal{I}^*[t]|+1} + \log \frac{\ell_i(s_{1,t}^i | \theta_1)}{\ell_i(s_{1,t}^i | \theta_2)}$.
  end
end

exists a row-stochastic matrix $\mathbf{M}^{1,2}[t] \in \mathbb{R}^{(n-\phi)\times(n-\phi)}$ such that

$$r_t^i(\theta_1, \theta_2) = \sum_{j=1}^{n-\phi} \mathbf{M}_{ij}^{1,2}[t] r_{t-1}^j(\theta_1, \theta_2) + \log \frac{\ell_i(s_{1,t}^i \mid \theta_1)}{\ell_i(s_{1,t}^i \mid \theta_2)}. \tag{4.31}$$

Note that matrix $\mathbf{M}^{1,2}$ depends on the choice of hypotheses θ_1 and θ_2.

For a given pair of hypotheses θ_1 and θ_2, let $\mathbf{r}_t(\theta_1, \theta_2) \in \mathbb{R}^{n-\phi}$ be the vector that stacks $r_t^i(\theta_1, \theta_2)$. The evolution of $\mathbf{r}(\theta_1, \theta_2)$ can be compactly written as

$$\begin{aligned} \mathbf{r}_t(\theta_1, \theta_2) &= \mathbf{M}^{1,2}[t] \mathbf{r}_{t-1}(\theta_1, \theta_2) + \sum_{r=1}^{t} \mathcal{L}_r(\theta_1, \theta_2) \\ &= \sum_{r=1}^{t} \mathbf{\Phi}^{1,2}(t, r+1) \sum_{k=1}^{r} \mathcal{L}_k(\theta_1, \theta_2), \end{aligned} \tag{4.32}$$

where $\mathbf{\Phi}^{1,2}(t, r+1) \triangleq \mathbf{M}^{1,2}[t]\mathbf{M}^{1,2}[t-1] \cdots \mathbf{M}^{1,2}[r+1]$ for $r \leq t$, $\mathbf{\Phi}^{1,2}(t,t) \triangleq \mathbf{M}^{1,2}[t]$ and $\mathbf{\Phi}^{1,2}(t, t+1) \triangleq \mathbf{I}$. We do the analysis for each pair of θ_1 and θ_2 separately.

The remaining proof is identical to the proof of Theorem 22, and is omitted. □

Proposition 9. *Suppose there exists* $\widetilde{\theta} \in \Theta$ *such that for any* $\theta \neq \widetilde{\theta}$, *it holds*

that $r_t^i(\widetilde{\theta}, \theta) \xrightarrow{\text{a.s.}} +\infty$*, and* $r_t^i(\theta, \widetilde{\theta}) \xrightarrow{\text{a.s.}} -\infty$*. Then* $\widetilde{\theta} = \theta^*$*.*

Proof. We prove this proposition by contradiction. Suppose there exists $\widetilde{\theta} \neq \theta^* \in \Theta$ such that for any $\theta \neq \widetilde{\theta}$, it holds that $r_t^i(\widetilde{\theta}, \theta) \xrightarrow{\text{a.s.}} +\infty$, and $r_t^i(\theta, \widetilde{\theta}) \xrightarrow{\text{a.s.}} -\infty$. Then we know that $r_t^i(\widetilde{\theta}, \theta^*) \xrightarrow{\text{a.s.}} +\infty$ and $r_t^i(\theta^*, \widetilde{\theta}) \xrightarrow{\text{a.s.}} -\infty$, contradicting Theorem 23. Thus, Proposition 9 is true.

□

4.6 BFL in the Absence of Byzantine Agents

In this section, we present BFL for the special case in the absence of Byzantine agents, i.e., $f = 0$, named Failure-free BFL. Since $f = 0$, all the agents in the network are cooperative, and no trimming is needed. Indeed, the BFL for $f = 0$ is a simple modification of the algorithm proposed in [83].

Algorithm 11: Failure-free BFL

1 Transmit current score vector μ_{t-1}^i on all outgoing edges;
2 Wait until a private signal s_t^i is observed and score vectors are received from all incoming neighbors $\mathcal{I}_i$;
3 **for** $\theta \in \Theta$ **do**
4 $\quad \mu_t^i(\theta) \leftarrow \dfrac{\ell_i(s_{1,t}^i|\theta) \prod_{j \in \mathcal{I}_i \cup \{i\}} \mu_{t-1}^j(\theta)^{\frac{1}{|\mathcal{I}_i|+1}}}{\sum_{p=1}^m \ell_i(s_{1,t}^i|\theta) \prod_{j \in \mathcal{I}_i \cup \{i\}} \mu_{t-1}^j(\theta)^{\frac{1}{|\mathcal{I}_i|+1}}}$.
5 **end**

For each time $t \geq 1$, we define a matrix that follows the structure of $G(\mathcal{V}, \mathcal{E})$ as follows:

$$\mathbf{A}_{ij} \triangleq \begin{cases} \frac{1}{|\mathcal{I}_i|+1}, & j \in \mathcal{I}_i \cup \{i\} \\ 0, & \text{otherwise.} \end{cases} \tag{4.33}$$

Thus, the dynamic of $\psi_t^i(\theta, \theta^*)$ (defined in (4.9)) under Algorithm 11 can be

written as

$$\begin{aligned}
\psi_t^i(\theta,\theta^*) &= \log \frac{\mu_t^i(\theta)}{\mu_t^i(\theta^*)} \\
&= \log \frac{\ell_i(s_{1,t}^i|\theta)\prod_{j\in\mathcal{I}_i\cup\{i\}}\mu_{t-1}^j(\theta)^{\frac{1}{|\mathcal{I}_i|+1}}}{\ell_i(s_{1,t}^i|\theta^*)\prod_{j\in\mathcal{I}_i\cup\{i\}}\mu_{t-1}^j(\theta^*)^{\frac{1}{|\mathcal{I}_i|+1}}} \\
&= \log \prod_{j\in\mathcal{I}_i\cup\{i\}}\left[\frac{\mu_{t-1}^j(\theta)}{\mu_{t-1}^j(\theta^*)}\right]^{\frac{1}{|\mathcal{I}_i|+1}} + \log\frac{\ell_i(s_{1,t}^i \mid \theta)}{\ell_i(s_{1,t}^i \mid \theta^*)} \\
&= \log \prod_{j\in\mathcal{I}_i\cup\{i\}}\left[\frac{\mu_{t-1}^j(\theta)}{\mu_{t-1}^j(\theta^*)}\right]^{\frac{1}{|\mathcal{I}_i|+1}} + \sum_{r=1}^{t}\log\frac{\ell_i(s_r^i \mid \theta)}{\ell_i(s_r^i \mid \theta^*)} \\
&= \sum_{j=1}^{n}\mathbf{A}_{ij}\psi_{t-1}^i(\theta,\theta^*) + \sum_{r=1}^{t}\mathcal{L}_r^i(\theta,\theta^*) \quad \text{by (4.9) and (4.33).}
\end{aligned}$$

Recall that $\boldsymbol{\psi}_t(\theta,\theta^*) \in \mathbb{R}^{n-\phi}$ is the vector that stacks $\psi_{t-1}^i(\theta,\theta^*)$ with the i–th entry being $\psi_{t-1}^i(\theta,\theta^*)$ for all $i \in \mathcal{N}$. Since $f = 0$, i.e., the network is free of failures, it holds that

$$0 \leq \phi = |\mathcal{F}| \leq f = 0.$$

Thus, $\boldsymbol{\psi}_t(\theta,\theta^*) \in \mathbb{R}^n$. Similar to (4.12), the evolution of $\boldsymbol{\psi}_t(\theta,\theta^*)$ can be compactly written as follows:

$$\begin{aligned}
\boldsymbol{\psi}_t(\theta,\theta^*) &= \mathbf{A}^t\boldsymbol{\psi}_0(\theta,\theta^*) + \sum_{r=1}^{t}\mathbf{A}^{t-r}\sum_{k=1}^{r}\mathcal{L}_k(\theta,\theta^*) \\
&= \sum_{r=1}^{t}\mathbf{A}^{t-r}\sum_{k=1}^{r}\mathcal{L}_k(\theta,\theta^*). \qquad (4.34)
\end{aligned}$$

The last equality holds from the fact that $\boldsymbol{\psi}_0(\theta,\theta^*) = \mathbf{0}$.

As mentioned before, the non-Bayesian learning rules [83, 84, 86, 88] are consensus-based learning algorithms, wherein agents are required to reach a common decision asymptotically.

Assumption 4. *The underlying communication network $G(\mathcal{V},\mathcal{E})$ is strongly connected.*

It is easy to see that $G(\mathcal{V},\mathcal{E})$ itself is the only reduced graph of $G(\mathcal{V},\mathcal{E})$, and that Assumption 4 is the special case of Assumption 1 when $f = 0$.

Thus,

$$\chi_m = 1, \qquad \text{and} \qquad \nu_m = \chi_m(n-\phi) = n.$$

Note that both χ_m and ν_m are independent of m when $f = 0$. Henceforth in this section, we drop the subscripts of χ_m and ν_m for ease of notation.

Similar to (4.3), for any $r \geq 1$, we get

$$\lim_{t\geq r,\ t\to\infty} \mathbf{A}^{t-r} = \mathbf{1}\boldsymbol{\pi}.$$

Since $\mathbf{A}$ is time-invariant, the product limit $\lim_{t\geq r,\ t\to\infty} \mathbf{A}^{t-r}$ is also independent of r.

It is easy to see that

$$\mathbf{A} \geq \frac{1}{n}\mathbf{H},$$

where $\mathbf{H}$ is the adjacency matrix of the communication graph $G(\mathcal{V}, \mathcal{E})$, and that

$$\pi_j \geq \frac{1}{n^n}, \quad \forall\, j = 1, \cdots, n. \tag{4.35}$$

The following corollary is a direct consequence of Theorem 20, and its proof is omitted.

Corollary 6. *For all $t \geq r \geq 1$, it holds that $|[\mathbf{A}^{t-r}]_{ij} - \pi_j| \leq (1 - \frac{1}{n^n})^{\lceil \frac{t-r}{n} \rceil}$, where $[\mathbf{A}^{t-r}]_{ij}$ is the i,j–th entry of matrix $\mathbf{A}^{t-r}$.*

In addition, when $f = 0$, Assumption 2 becomes

Assumption 5. *Suppose that Assumption 4 holds. For any $\theta \neq \theta^*$, the following holds*

$$\sum_{j=1}^{m} D\left(\ell_j(\cdot|\theta^*) \parallel \ell_j(\cdot|\theta)\right) \ \neq\ 0. \tag{4.36}$$

As an immediate consequence of Theorem 22, we have the following corollary.

Corollary 7. *When Assumption 5 holds, each agent i will concentrate its score on the true hypothesis θ^* almost surely, i.e., $\mu_t^i(\theta) \xrightarrow{\text{a.s.}} 0$ for all $\theta \neq \theta^*$.*

Since Corollary 7 is the special case of Theorem 22 for $f = 0$, the proof of Corollary 7 is omitted.

4.6.1 Finite-Time Analysis of Failure-Free BFL

In this subsection, we present the convergence rate on the score vectors that is achievable in finite time with high probability. Note that this convergence rate is not the convergence rate of the real belief vectors. Our proof is similar to the proof presented in [83, 88].

Lemma 14. *Let* $\lambda \triangleq \left(1 - (\frac{1}{n})^n\right)^{\frac{1}{n}}$*, and let* $\theta \neq \theta^*$*, and consider* $\psi_t^i(\theta, \theta^*)$ *as defined in* (4.9)*. Then, for each agent* i *we have*

$$\mathbb{E}\left[\psi_t^i(\theta, \theta^*)\right] \leq \frac{nC_0}{(1 - \frac{1}{n^n})(1 - \lambda)} t - \frac{C_1}{2n^n} t^2.$$

Proof. By (4.34), we have $\psi_t^i(\theta, \theta^*) = \sum_{r=1}^{t} \sum_{j=1}^{n} [\mathbf{A}^{t-r}]_{ij} \sum_{k=1}^{r} \mathcal{L}_k^j(\theta, \theta^*)$. Taking expectation of $\psi_t^i(\theta, \theta^*)$ with respect to $\ell^i(\cdot \mid \theta^*)$, we get

$$\begin{aligned}
\mathbb{E}^*\left[\boldsymbol{\psi}_t^i(\theta, \theta^*)\right] &= \mathbb{E}^*\left[\sum_{r=1}^{t} \sum_{j=1}^{n} [\mathbf{A}^{t-r}]_{ij} \sum_{k=1}^{r} \mathcal{L}_k^j(\theta, \theta^*)\right] \\
&= \sum_{r=1}^{t} \sum_{j=1}^{n} [\mathbf{A}^{t-r}]_{ij} \sum_{k=1}^{r} \mathbb{E}^*\left[\mathcal{L}_k^j(\theta, \theta^*)\right] \\
&= \sum_{r=1}^{t} \sum_{j=1}^{n} [\mathbf{A}^{t-r}]_{ij} r H_j(\theta, \theta^*) \quad \text{by} \quad (4.13) \\
&= \sum_{r=1}^{t} \sum_{j=1}^{n} \left([\mathbf{A}^{t-r}]_{ij} - \pi_j\right) r H_j(\theta, \theta^*) + \sum_{r=1}^{t} \sum_{j=1}^{n} \pi_j r H_j(\theta, \theta^*).
\end{aligned} \tag{4.37}$$

For the first term on the right-hand side of (4.37), we have

$$\begin{aligned}
&\sum_{r=1}^{t}\sum_{j=1}^{n}\left([\mathbf{A}^{t-r}]_{ij}-\pi_j\right) rH_j(\theta,\theta^*)\\
&\leq \sum_{r=1}^{t}\sum_{j=1}^{n}\left|[\mathbf{A}^{t-r}]_{ij}-\pi_j\right| r\left|H_j(\theta,\theta^*)\right|\\
&\leq \sum_{r=1}^{t}\sum_{j=1}^{n}\left[1-\frac{1}{n^n}\right]^{\lceil\frac{t-r}{n}\rceil} rC_0 \quad \text{by Corollary 6, and (4.14)}\\
&= nC_0\sum_{r=1}^{t}\left[1-\frac{1}{n^n}\right]^{\lceil\frac{t-r}{n}\rceil} r\\
&\leq \frac{nC_0}{(1-\frac{1}{n^n})(1-\lambda)}t. \qquad (4.38)
\end{aligned}$$

Since $G(\mathcal{V},\mathcal{E})$ is the only source component, C_1 (defined in (4.15)) becomes

$$C_1 = \min_{\theta,\theta^*\in\Theta;\theta\neq\theta^*}\sum_{i=1}^{n} D(\ell_i(\cdot|\theta^*)\parallel\ell_i(\cdot|\theta)).$$

Thus, for the second term on the right-hand side of (4.37), we get

$$\begin{aligned}
\sum_{r=1}^{t}\sum_{j=1}^{n}\pi_j rH_j(\theta,\theta^*) &\leq \sum_{r=1}^{t}\sum_{j=1}^{n}\frac{1}{n^n}rH_j(\theta,\theta^*) \quad \text{by (4.35) and (4.13)}\\
&= \frac{1}{n^n}\sum_{r=1}^{t} r\sum_{j=1}^{n}H_j(\theta,\theta^*)\\
&\leq -\frac{1}{n^n}\sum_{r=1}^{t} rC_1\\
&\leq -\frac{C_1}{2n^n}t^2. \qquad (4.39)
\end{aligned}$$

By (4.38) and (4.39), (4.37) becomes

$$\begin{aligned}
\mathbb{E}^*\left[\boldsymbol{\psi}_t^i(\theta,\theta^*)\right] &= \sum_{r=1}^{t}\sum_{j=1}^{n}\left([\mathbf{A}^{t-r}]_{ij}-\pi_j\right) rH_j(\theta,\theta^*)+\sum_{r=1}^{t}\sum_{j=1}^{n}\pi_j rH_j(\theta,\theta^*)\\
&\leq \frac{nC_0}{(1-\frac{1}{n^n})(1-\lambda)}t-\frac{C_1}{2n^n}t^2, \qquad (4.40)
\end{aligned}$$

proving the lemma.

□

Similar to [83, 88], we also use McDiarmid's inequality.

Theorem 24 (McDiarmid's inequality)**.** *Let $X_1, \cdots, X_t$ be independent random variables with sample space $\mathcal{X}$ and consider the mapping $H : \mathcal{X}^t \to \mathbb{R}$. If for $r = 1, \cdots, t$, and every sample $x_1, \cdots, x_t$, $x_r' \in \mathcal{X}$, the function H satisfies*

$$|H(x_1, \cdots, x_r, \cdots, x_t) - H(x_1, \cdots, x_r', \cdots, x_t)| \leq c_r,$$

then for all $\epsilon > 0$,

$$\mathbb{P}\left[|H(x_1, \cdots, x_t) - \mathbb{E}[H(x_1, \cdots, x_t)]| \geq \epsilon\right] \leq \exp\left\{\frac{-2\epsilon^2}{\sum_{r=1}^t c_r^2}\right\}.$$

Theorem 25. *Under Assumption 5, for any $\rho \in (0, 1)$, there exists an integer $T(\rho)$ such that with probability at least $1 - \rho$, for all $t \geq T(\rho)$ and for all $\theta \neq \theta^*$, we have*

$$\mu_t^i(\theta) \leq \exp\left(\frac{nC_0}{(1 - \frac{1}{n^n})(1 - \lambda)}t - \frac{C_1}{4n^n}t^2\right),$$

where C_0 and C_1 are defined in (4.14) *and* (4.15) *respectively, and $T(\rho) = \frac{64C_0^2 n^{2n}}{3C_1^2} \log \frac{1}{\rho}$.*

Proof. Since $\mu_t^i(\theta^*) \in (0, 1]$, we have

$$\mu_t^i(\theta) \leq \frac{\mu_t^i(\theta)}{\mu_t^i(\theta^*)} = \exp\left(\psi_t^i(\theta, \theta^*)\right). \tag{4.41}$$

Thus, we have

$$\begin{aligned}
&\mathbb{P}\left(\mu_t^i(\theta) \geq \exp\left(\frac{nC_0}{(1 - \frac{1}{n^n})(1 - \lambda)}t - \frac{C_1}{4n^n}t^2\right)\right) \qquad (4.42)\\
&\leq \mathbb{P}\left(\psi_t^i(\theta, \theta^*) \geq \frac{nC_0}{(1 - \frac{1}{n^n})(1 - \lambda)}t - \frac{C_1}{4n^n}t^2\right) \quad \text{due to (4.41)}\\
&\leq \mathbb{P}\left(\psi_t^i(\theta, \theta^*) - \mathbb{E}^*\left[\psi_t^i(\theta, \theta^*)\right] \geq \frac{C_1}{4n^n}t^2\right). \quad \text{due to (4.40)}
\end{aligned}$$

Note that $\psi_t^i(\theta, \theta^*)$ is a function of the random vector $\mathbf{s}_1, \cdots, \mathbf{s}_t$. Let $\bar{\mathcal{S}} \triangleq \mathcal{S}_1 \times \cdots \times \mathcal{S}_{n-\phi}$ be the joint signal space of all the good agents. Let $\bar{\mathbf{s}}_1, \cdots, \bar{\mathbf{s}}_t$

be a sample path of length t; and let $p \in \{1, \cdots, t\}$. We use

$$\max_{\mathbf{s}_p \in \bar{\mathcal{S}}} \psi_t^i(\theta, \theta^*)$$

to denote the maximum value of $\psi_t^i(\theta, \theta^*)$ that is obtained by maximizing $\psi_t^i(\theta, \theta^*)$ over all the possible realization of the p–th signal vector, i.e., $\bar{\mathcal{S}}$, while keeping all the other elements of the sample path fixed. Similarly, we denote

$$\min_{\mathbf{s}_p \in \bar{\mathcal{S}}} \psi_t^i(\theta, \theta^*)$$

as the minimum value of $\psi_t^i(\theta, \theta^*)$ that is obtained by minimizing $\psi_t^i(\theta, \theta^*)$ over all the possible realization of the p–th signal vector, i.e., $\bar{\mathcal{S}}$, while keeping all the other elements of the sample path fixed. We consider the difference between $\max_{\mathbf{s}_p \in \bar{\mathcal{S}}} \psi_t^i(\theta, \theta^*)$ and $\min_{\mathbf{s}_p \in \bar{\mathcal{S}}} \psi_t^i(\theta, \theta^*)$ for the given sample path $\bar{\mathbf{s}}_1, \cdots, \bar{\mathbf{s}}_t$ w. r. t. the p-th element. In particular, we have

$$\begin{aligned}
&\max_{\mathbf{s}_p \in \bar{\mathcal{S}}} \psi_t^i(\theta, \theta^*) - \min_{\mathbf{s}_p \in \bar{\mathcal{S}}} \psi_t^i(\theta, \theta^*) \\
&= \max_{\mathbf{s}_p \in \bar{\mathcal{S}}} \sum_{r=1}^{t} \sum_{j=1}^{n} [\mathbf{A}^{t-r}]_{ij} \sum_{k=1}^{r} \mathcal{L}_k^j(\theta, \theta^*) - \min_{\mathbf{s}_p \in \bar{\mathcal{S}}} \sum_{r=1}^{t} \sum_{j=1}^{n} [\mathbf{A}^{t-r}]_{ij} \sum_{k=1}^{r} \mathcal{L}_k^j(\theta, \theta^*) \text{ by (4.34)} \\
&= \max_{\mathbf{s}_p \in \bar{\mathcal{S}}} \sum_{r=p}^{t} \sum_{j=1}^{n} [\mathbf{A}^{t-r}]_{ij} \mathcal{L}_p^j(\theta, \theta^*) - \min_{\mathbf{s}_p \in \bar{\mathcal{S}}} \sum_{r=p}^{t} \sum_{j=1}^{n} [\mathbf{A}^{t-r}]_{ij} \mathcal{L}_p^j(\theta, \theta^*) \\
&\leq \sum_{r=p}^{t} \sum_{j=1}^{n} [\mathbf{A}^{t-r}]_{ij} C_0 + \sum_{r=p}^{t} \sum_{j=1}^{n} [\mathbf{A}^{t-r}]_{ij} C_0 \\
&= 2C_0(t-p+1) \triangleq c_p.
\end{aligned}$$

By McDiarmid's inequality (Theorem 24), we obtain that

$$\begin{aligned}
\mathbb{P}\left(\psi_t^*(\theta, \theta^*) - \mathbb{E}^*\left[\psi_t^*(\theta, \theta^*)\right] \geq \frac{C_1}{4n^n} t^2\right) &\leq \exp\left(-\frac{2\frac{C_1^2}{16n^{2n}} t^4}{\sum_{p=1}^{t} (2C_0(t-p+1))^2}\right) \\
&\leq \exp\left(-\frac{3C_1^2}{64C_0^2 n^{2n}} t\right), \qquad (4.43)
\end{aligned}$$

where the last inequality follows from the fact that

$$t(t+1)(2t+1) \leq 4t^3 \quad \forall\, t \geq 2,$$

which can be shown by induction.

From Equations (4.41), (4.42), and (4.43), it follows that for a given confidence level ρ, in order to have

$$\mathbb{P}\left(\mu_t^i(\theta) \geq \exp\left(\frac{nC_0}{(1-\frac{1}{n^n})(1-\lambda)}t - \frac{C_1}{4n^n}t^2\right)\right) \leq \rho,$$

we require that

$$t \geq T(\rho) = \frac{64C_0^2 n^{2n}}{3C_1^2}\log\frac{1}{\rho}.$$

□

Remark 5. *The above finite-time analysis is not directly applicable for the general case when $f > 0$, due to the fact that the local scores are dependent on all the observations collected so far* as well as *all the future observations.*

4.7 Conclusion

This chapter addresses the problem of consensus-based non-Bayesian learning over multi-agent networks when an unknown subset of agents may be adversarial (Byzantine). We propose two learning rules, and characterize the tight network identifiability condition for any consensus-based learning rule of interest to exist. In our first update rule, each agent updates its local scores as (up to normalization) the product of (1) the likelihood of the *cumulative* private signals and (2) the weighted geometric average of the scores of its incoming neighbors and itself. Under reasonable assumptions on the underlying network structure and the global identifiability of the network, we show that all the non-faulty agents asymptotically agree on the true state almost surely. In general when agents may be adversarial, the network identifiability condition specified for the above learning rule scales poorly in m. In addition, the computation complexity per agent per iteration of this learning rule is forbiddingly high. Thus, we propose a modification of our first learning rule, whose complexity per iteration per agent is $O(m^2 n \log n)$. We show that this improved learning rule works under a much weaker global identifiability condition that is independent of m.

We so far focus on a synchronous system and static network; our results may be generalizable to asynchronous as well as time varying networks.

Throughout this chapter, we assume that consensus among non-faulty agents needs to be achieved. Although this is necessary for the family of consensus-based algorithms (by definition), this is not the case for the non-faulty agents to collaboratively learn the true state in general. Indeed, there is a tradeoff between the capability of the network to reach consensus and the tight condition of the network detectability. For instance, if the network is disconnected, then information cannot be propagated across the connected components. Thus, the non-faulty agents in each connected component have to be able to learn the true state. We leave investigating the above tradeoff as future work.

CHAPTER 5

DISTRIBUTED STATISTICAL MACHINE LEARNING IN ADVERSARIAL SETTINGS: BYZANTINE GRADIENT DESCENT

5.1 Introduction

In many machine learning tasks, we are interested in efficiently training an accurate prediction model from observed data samples. As the data volume and model complexity continue to grow, such tasks consume a large and still increasing amount of computation resources. Distributed machine learning has emerged as an attractive solution to large-scale problems and received intensive attention [14, 15, 16, 17, 18, 19]. In this setting, the data samples and computation are distributed across multiple machines, which collaboratively learn a model by communicating with each other.

Many efficient distributed machine learning algorithms [14, 15] and system implementations [16, 17, 18, 19] have been proposed and studied. Prior work mostly focuses on the traditional "training within cloud" framework where the model training process is carried out within the cloud infrastructures. In this framework, distributed machine learning is secured via system architectures, hardware devices, and monitoring [20, 21, 22]. This framework faces significant privacy risk, as the data has to be collected from owners and stored within the clouds. Although a variety of privacy-preserving solutions have been developed [23, 24], privacy breaches still occur frequently, with recent examples including iCloud leaks of celebrity photos [25] and PRISM surveillance program [26].

To address privacy concerns, a new machine learning paradigm called *Federated Learning* was proposed by Google researchers [27, 28]. It aims at learning an accurate model without collecting data from owners and storing the data in the cloud. The training data is kept locally on the owners' computing devices, which are recruited to participate directly in the model training process and hence function as working machines. Google has been intensively

testing this new paradigm in their recent projects such as *Gboard* [28], the Google Keyboard. Compared to "training within cloud", Federated Learning has lower privacy risk, but inevitably becomes less secured. In particular, it faces the following three key challenges:

- Security: The devices of the recruited data owners can be easily re-programmed and completely controlled by external attackers, and thus behave adversarially.
- Small local datasets versus high model complexity: While the total number of data samples over all data owners may be large, each individual owner may keep only a small amount of data, which by itself is insufficient for learning a complex model.
- Communication constraints: Data transmission between the recruited devices and the cloud may suffer from high latency and low-throughout. Communication between them is therefore a scarce resource, whose usage should be minimized.

In this chapter, we address the above challenges by developing a new iterative distributed machine learning algorithm that is able to (1) tolerate Byzantine failures, (2) accurately learn a highly complex model with low local data volume, and (3) converge exponentially fast using logarithmic communication rounds.

5.1.1 Learning Goals

To formally study the distributed machine learning problem in adversarial settings, we consider a standard statistical learning setup, where the data is generated probabilistically from an unknown distribution and the true model is parameterized by a vector. More specifically, let $X \in \mathcal{X}$ be the input data generated according to some distribution μ. Let $\Theta \subset \mathbb{R}^d$ be the set of all choices of model parameters. We consider a loss function $f : \mathcal{X} \times \Theta \to \mathbb{R}$, where $f(x, \theta)$ measures the risk induced by a realization x of the data under the model parameter choice θ. A classical example is linear regression, where $x = (w, y) \in \mathbb{R}^d \times \mathbb{R}$ is the feature-response pair and $f(x, \theta) = \frac{1}{2}\left(\langle w, \theta \rangle - y\right)^2$ is the usual squared loss.

We are interested in learning the model choice θ^* that minimizes the *population risk*, i.e.,

$$\theta^* \in \arg\min_{\theta \in \Theta} F(\theta) \triangleq \mathbb{E}\left[f(X, \theta)\right], \tag{5.1}$$

assuming that $\mathbb{E}\left[f(X, \theta)\right]$ is well defined over Θ.[1] The model choice θ^* is optimal in the sense that it minimizes the average risk to pay if the model chosen is used for prediction in the future with a fresh random data sample.

When μ—the distribution of X—is known, which is rarely the case in practice, the population risk can be evaluated exactly, and θ^* can be computed by solving the minimization problem in (5.1). We instead assume that μ is *unknown*, in which case the population risk function $F(\cdot) = \mathbb{E}\left[f(X, \cdot)\right]$ can only be approximated using the observed data samples generated from μ. In particular, we assume that there exist N independently and identically distributed data samples $X_i \overset{\text{i.i.d.}}{\sim} \mu$ for $i = 1, \ldots, N$. Note that estimating θ^* using finitely many data samples will always have a *statistical error* due to the randomness in the data, even in the centralized, failure-free setting. Our results account for this effect.

5.1.2 System Model

We are interested in distributed solutions of the above statistical learning problem. Specifically, the system of interest consists of a parameter server[2] and m working machines. In the example of Federated Learning, the parameter server represents the cloud, and the m working machines correspond to m data owners' computing devices.

We assume that the N data samples are distributed evenly across the m working machines. In particular, each working machine i keeps a subset $\mathcal{S}_i$ of the data, where $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ and $|\mathcal{S}_i| = N/m$. We further assume that the parameter server can communicate with all working machines in

[1] For example, if $\mathbb{E}\left[|f(X, \theta)|\right]$ is finite for every $\theta \in \Theta$, the population risk $\mathbb{E}\left[f(X, \theta)\right]$ is well defined.

[2] Note that, due to communication bandwidth constraints, practical systems use multiple networked parameter servers. In this chapter, for ease of explanation, we assume there is only one parameter server in the system. Fortunately, as can be seen from our algorithm descriptions and our detailed correctness analysis, the proposed algorithm also works for the aforementioned more practical setting.

synchronous communication rounds, and leave the asynchronous setting as future directions.

Among the m working machines, we assume that up to q of them can suffer Byzantine failures and thus behave maliciously; for example, they may be reprogrammed and completely controlled by the system attacker. In a given execution, the set of Byzantine machines can even change between communication rounds. Byzantine machines are assumed to have complete knowledge of the system, including the total number of working machines m, all N data samples over the whole system, the programs that the working machines are supposed to run, and the program run by the parameter server. Moreover, Byzantine machines can collaborate with each other [91]. The only constraint is that these machines cannot corrupt the local data — but they can lie when communicating with the server. This arbitrary behavior of Byzantine machines creates unspecified dependency across communication rounds — a key challenge in our algorithm design and convergence analysis.

In this chapter, we use *rounds* and *iterations* interchangeably.

5.1.3 Existing Distributed Machine Learning Algorithms

Existing algorithms for distributed machine learning can be roughly categorized into three classes according to their communication costs.

SGD: On one end of the spectrum lies the *Stochastic Gradient Descent (SGD)* algorithm. Using this algorithm, the parameter server receives, in each iteration, a gradient computed at a single data sample from one working machine, and uses it to perform one gradient descent step. Even when F is strongly convex, the convergence rate of SGD is only $O(1/t)$ with t iterations. This is much slower than the exponential (geometric) convergence of standard gradient descent. Therefore, SGD requires a large number of communication rounds, which could be costly. Indeed, it has been demonstrated in [28] that SGD has 10-100 times higher communication cost than standard gradient descent, and is therefore inadequate for scenarios with scarce communication bandwidth.

One-Shot Aggregation: On the other end of the spectrum, using a *One-Short Aggregation* method, each working machine computes an estimate of the model parameter using only its local data and reports it to

the server, which then aggregates all the estimates reported to obtain a final estimate [92, 93]. One-short aggregation method only needs a single round of communication from the working machines to the parameter server, and thus is communication-efficient. However, it requires $N/m \gg d$ so that a coarse parameter estimate can be obtained at each machine. This algorithm is therefore not applicable in scenarios where local data is small in size but the model to learn is of high dimension.

BGD: *Batch Gradient Descent (BGD)* lies in between the above two extremes. At each iteration, the parameter server sends the current model parameter estimate to all working machines. Each working machine computes the gradient based on all locally available data, and then sends the gradient back to the parameter server. The parameter server averages the received gradients and performs a gradient descent step. When F is strongly convex, BGD converges exponentially fast, and hence requires only a few rounds of communication. BGD also works in the scenarios with limited local data, i.e., $N/m \ll d$, making it an ideal candidate in Federated Learning. However, it is prone to Byzantine failures. A single Byzantine failure at a working machine can completely skew the average value of the gradients received by the parameter server, and thus foils the algorithm.

5.1.4 Contributions

In this chapter, we propose a Byzantine gradient descent method. Specifically, the parameter server aggregates the local gradients reported by the working machines in three steps: (1) it partitions all the received local gradients into k batches and computes the mean for each batch, (2) it computes the *geometric median* of the k batch means, and (3) it performs a gradient descent step using the geometric median.

We prove that the proposed algorithm can tolerate q Byzantine failures up to $2(1+\epsilon)q \leq m$ for an arbitrarily small but fixed constant $\epsilon > 0$, is applicable even in the scarce local data regime where $N/m \ll d$, and only requires $\log(N)$ communication rounds. However, as q increases, the estimation error also increases. In particular, the error in estimating the target model parameter θ^* converges exponentially fast to $\max\{\sqrt{dq/N},\ \sqrt{d/N}\}$, whereas the idealized estimation error rate in the centralized and failure-free

Table 5.1: Comparison of our proposed Byzantine gradient descent algorithm with several existing distributed machine learning algorithms. In [94], the focus is to estimate the minimizer of the average cost over a given deterministic dataset; almost sure convergence is proved without an explicit characterization of convergence speed nor the estimation errors.

	Byzantine Failures	Convergence speed	Estimation error
One-shot	0	No iteration	$\sqrt{d/N}$
SGD	0	$1/t$	?
BGD	0	$\exp(-t)$	$\sqrt{d/N}$
Robust one-shot [95]	$2q+1 \leq m$	No iteration	$\sqrt{dm/N}$
Byzantine SGD [94]	$2q+2 < m$	*	*
Byzantine GD (This chapter)	$2(1+\epsilon)q \leq m$	$\exp(-t)$	$\sqrt{d(q \vee 1)/N}$

setting is $\sqrt{d/N}$. The total computational complexity of our algorithm is of $O((N/m)d \log N)$ at each working machine and $O(md + qd \log^3 N)$ at the parameter server, and the total communication cost is of $O(md \log N)$. We provide a comparison with existing distributed machine learning algorithms in Table 5.1, where $q \vee 1 \triangleq \max\{1, q\}$.

Notably, our algorithm does *not* assume that at each iteration fresh samples are drawn, or that the data is split into multiple chunks beforehand and the gradient is computed using a new chunk at each round. This poses a significant challenge in our convergence proof: there exists complicated probabilistic dependency among the iterates and the aggregated gradients. Even worse, such dependency cannot be specified due to the arbitrary behavior of the Byzantine machines. We overcome this challenge by proving that the geometric median of means of gradients *uniformly* converges to the true gradient function $\nabla F(\theta)$.

5.2 Related Work

The present chapter intersects with two main areas of research: statistical machine learning and distributed computing. Most related to our work is

a very interesting recent arXiv preprint [94] that we became aware of when preparing this chapter. It also studies distributed machine learning in adversarial settings, but the setup is different from ours. In particular, their focus is solving an optimization problem, where all m working machines have access to a common dataset $\{x_i\}_{i=1}^N$ and the goal is to collectively compute the minimizer $\widehat{\theta}$ of the average cost $Q(\theta) = (1/N)\sum_{i=1}^N f(x_i, \theta)$. Importantly, the dataset $\{x_i\}_{i=1}^N$ are assumed to be deterministic. In contrast, we adopt the standard statistical learning framework, where each working machine only has access to its own data samples, which are assumed to be generated from some unknown distribution μ, and the goal is to estimate the optimal model parameter θ^* that minimizes the true prediction error $\mathbb{E}_{X\sim\mu}[f(X, \theta)]$ — as mentioned, characterizing the statistical estimation accuracy is a main focus of ours. Our algorithmic approaches and main results are also significantly different; see Table 5.1 for a comparison.

Our work is also closely related to the literature on robust parameter estimation using geometric median. It is shown in [96] that geometric median has a breakdown point of 0.5, that is, given a collection of n vectors in $\mathbb{R}^d$, at least $\lfloor (n+1)/2 \rfloor / n$ number of points needs to be corrupted in order to arbitrarily perturb the geometric median. A more quantitative robustness result is recently derived in [97, Lemma 2.1]. The geometric median has been applied to distributed machine learning under the one-shot aggregation framework [95], under the restrictive assumption that the number of data available in each working machine satisfies $N/m \gg d$. While we also apply geometric median-of-mean as a sub-routine, our problem setup, overall algorithms and main results are completely different.

On the technical front, a crucial step in our convergence proof is to show the geometric median of means of n i.i.d. random gradients converges to the underlying gradient function $\nabla F(\theta)$ uniformly over θ. Our proof builds on several ideas from the empirical process theory, which guarantees uniform convergence of the empirical risk function $(1/n)\sum_{i=1}^n f(X_i, \cdot)$ to the population risk $F(\cdot)$. However, what we need is the uniform convergence of empirical *gradient* function $(1/n)\sum_{i=1}^n \nabla f(X_i, \cdot)$, as well as its *geometric median* version, to the population gradient function $\nabla F(\cdot)$. To this end, we use concentration inequalities to first establish point-wise convergence and then boost it to uniform convergence via the celebrated ϵ-net argument. Similar ideas have been used recently in the work [98], which studies the stationary

points of the empirical risk function.

5.3 Algorithms and Summary of Main Results

In this section, we present our distributed statistical machine learning algorithm, named *Byzantine Gradient Descent Method*, and briefly summarize our main results on the performance of our algorithm.

The main algorithm design idea is to exploit the statistical properties of the N training data samples. Recall that they are generated from some common but unknown distribution μ. Informally speaking, this implies that the local datasets $\mathcal{S}_j$'s share some *similarity*, and the locally computed gradients reported by these good machines may also be "similar". Based on this observation, in each iteration of our algorithm, the parameter server first groups the received gradients into batches and computes the averages in each batch in order to amplify the "similarity" of the averaged gradients in batches; and then the parameter server computes the *geometric median* of the averaged gradients to cripple the interruption of Byzantine machines.

5.3.1 Algorithms

Recall that our fundamental goal is to learn the optimal model choice θ^* defined in (5.1). We make the following standard assumption [14] so that the minimization problem in (5.1) can be solved efficiently (exponentially fast) in the ideal case when the population risk function F is known exactly, i.e., the distribution μ is known.

Assumption 6. *The population risk function $F : \Theta \to \mathbb{R}$ is L-strongly convex, and differentiable over Θ with M-Lipschitz gradient. That is, for all $\theta, \theta' \in \Theta$,*

$$F(\theta') \geq F(\theta) + \langle \nabla F(\theta), \theta' - \theta \rangle + \frac{L}{2}\|\theta' - \theta\|^2,$$

and

$$\|\nabla F(\theta) - \nabla F(\theta')\| \leq M\|\theta - \theta'\|.$$

Under Assumption 6, it is well-known [69] that using the standard gradient descent update

$$\theta_t = \theta_{t-1} - \eta \times \nabla F(\theta_{t-1}), \tag{5.2}$$

where η is some fixed stepsize, θ_t approaches the optimal θ^* exponentially fast. In particular, choosing the stepsize $\eta = L/(2M^2)$, it holds that

$$\|\theta_t - \theta^*\| \leq \left(1 - \left(\frac{L}{2M}\right)^2\right)^{\frac{t}{2}} \|\theta_0 - \theta^*\|.$$

Nevertheless, when the distribution μ is unknown, assumed in this chapter, the population gradient ∇F can only be approximated using sample gradients, if they exist. Recall that each working machine j (can possibly be Byzantine) keeps a very small set of data $\mathcal{S}_j$ with $|\mathcal{S}_j| = N/m$. Define the local empirical risk function, denoted by $\bar{f}^{(j)} : \Theta \to \mathbb{R}$, as follows:

$$\bar{f}^{(j)}(\theta) \triangleq \frac{1}{|\mathcal{S}_j|} \sum_{i \in \mathcal{S}_j} f(X_i, \theta), \quad \forall\, \theta \in \Theta. \tag{5.3}$$

Notice that $\bar{f}^{(j)}(\cdot)$ is a function of data samples $\mathcal{S}_j$ stored at machine j. Hence $\bar{f}^{(j)}(\cdot)$ is random. Although Byzantine machines can send arbitrarily malicious messages to the parameter server, they are not able to corrupt the local stored data. Thus, the local risk function $g_j(\cdot)$ is well-defined for all j, including the Byzantine machines. With a bit abuse of notation, we let

$$\bar{\boldsymbol{f}}(\theta) \triangleq \left(\bar{f}^{(1)}(\theta), \ldots, \bar{f}^{(m)}(\theta)\right)$$

be the vector that stacks the values of the m local functions evaluated at θ. For any $x \in \mathcal{X}$, we assume that $f(x, .) : \Theta \to \mathbb{R}$ is differentiable over Θ. When there is no confusion, we write $\nabla_\theta f(x, \theta)$ – the gradient of function $f(x, \cdot)$ evaluated at θ – simply as $\nabla f(x, \theta)$.

It is well-known that the average of the local gradients can be viewed as an approximation of the population gradient $\nabla F(\cdot)$. In particular, for a fixed θ,

$$\frac{1}{m} \sum_{j=1}^{m} \nabla \bar{f}^{(j)}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \nabla f(X_i, \theta) \xrightarrow{\text{a.s.}} \nabla F(\theta), \quad \text{as } N \to \infty. \tag{5.4}$$

Batch Gradient Descent relies on this observation. However, this method is sensitive to Byzantine failures as we explain in the sequel.

Batch Gradient Descent We describe the *Batch Gradient Descent (BGD)* in Algorithm 12. We initialize θ_0 to be some arbitrary value in Θ for simplicity. In practice, there are standard guides in choosing the initial point [99]. In round $t \geq 1$, the parameter server sends the current model parameter estimator θ_{t-1} to all working machines. Each working machine j computes the gradient $\nabla \bar{f}^{(j)}(\theta_{t-1})$ and sends $\nabla \bar{f}^{(j)}(\theta_{t-1})$ back to the parameter server. Note that q Byzantine machines may not follow the codes in Algorithm 12 – though their local gradients are also well-defined. Instead of the true local gradients, Byzantine machines can report arbitrarily malicious messages or no message to the server. If the server does not receive any message from a working machine, then that machine must be Byzantine faulty. In that case, the server sets $g_t^{(j)}(\theta_{t-1})$ to some arbitrary value. Precisely, let $\mathcal{B}_t$ denote the set of Byzantine machines at round t. The message received from machine j, denoted by $g_t^{(j)}(\theta_{t-1})$, can be described as

$$g_t^{(j)}(\theta_{t-1}) = \begin{cases} \nabla \bar{f}^{(j)}(\theta_{t-1}) & \text{if } j \notin \mathcal{B}_t \\ \star & \text{o.w. ,} \end{cases} \tag{5.5}$$

where, with a bit of abuse of notation, $\star$ denotes the arbitrary message whose value may be different across Byzantine machines, iterations, executions, etc. In step 3, the parameter server averages the received $g_t^{(j)}(\theta_{t-1})$ and updates θ_t using a gradient descent step.

Under Assumption 6, when there are no Byzantine machines, it is well-known that BGD converges exponentially fast. However, a single Byzantine failure can completely skew the average value of the gradients received by the parameter server, and thus foils the algorithm. This is because a Byzantine machine is assumed to have complete knowledge of the system, including the gradients reported by other machines.

Robust Gradient Aggregation Instead of taking the average of the received gradients $g_t^{(1)}(\theta_{t-1}), \cdots, g_t^{(m)}(\theta_{t-1})$, we propose a robust way to aggregate the collected gradients. Our aggregation rule is based on the notion of *geometric median*.

Geometric median, also known as spatial median or L_1 median, is a gen-

Algorithm 12: Standard Gradient Descent: Synchronous iteration $t \geq 1$

Parameter server:

Initialize: Let θ_0 be an arbitrary point in Θ;

1: Broadcast the current model parameter estimator θ_{t-1} to all working machines;
2: Wait to receive all the gradients reported by the m machines; Let $g_t^{(j)}(\theta_{t-1})$ denote the value received from machine j.
If no message from machine j is received, set $g_t^{(j)}(\theta_{t-1})$ to be some arbitrary value;
3: Update: $\theta_t \leftarrow \theta_{t-1} - \eta \times \left(\frac{1}{m}\sum_{j=1}^{m} g_t^{(j)}(\theta_{t-1})\right)$;

Working machine j:

1: Compute the gradient $\nabla \bar{f}^{(j)}(\theta_{t-1})$;
2: Send $\nabla \bar{f}^{(j)}(\theta_{t-1})$ back to the parameter server;

eralization of median in one-dimension to multiple dimensions, and has been widely used in robust statistics [100, 101, 102, 103]. Let $\{y_1, \ldots, y_n\} \subseteq \mathbb{R}^d$ be a multi-set of size n. The geometric median of $\{y_1, \ldots, y_n\}$, denoted by $\mathsf{med}\{y_1, \ldots, y_n\}$, is defined as

$$\mathsf{med}\{y_1, \ldots, y_n\} \triangleq \operatorname*{argmin}_{y \in \mathbb{R}^d} \sum_{i=1}^{n} \|y - y_i\|. \tag{5.6}$$

Geometric median is NOT required to lie in $\{y_1, \ldots, y_n\}$, and is unique unless all the points in $\{y_1, \ldots, y_n\}$ lie on a line. Note that if the ℓ_2 norm in (5.6) is replaced by the squared ℓ_2 norm, i.e., $\|\cdot\|^2$, then the minimizer is exactly the average.

In one dimension, median has the following nice robustness property: if strictly more than $\lfloor n/2 \rfloor$ points are in $[-r, r]$ for some $r \in \mathbb{R}$, then the median must be in $[-r, r]$. Likewise, in multiple dimensions, geometric median has the following robust property.

Lemma 15. *[97, Lemma 2.1] Let $z_1, z_2, \ldots, z_n$ denote n points in a Hilbert space. Let z_* denote their geometric median. For any $\alpha \in (0, 1/2)$ and given*

$r \in \mathbb{R}$, *if* $\sum_{i=1}^{n} \mathbf{1}_{\{\|z_i\|_2 \le r\}} \ge (1-\alpha)n$, *then* $\|z_*\|_2 \le C_\alpha r$, *where*

$$C_\alpha = \frac{1-\alpha}{\sqrt{1-2\alpha}}. \tag{5.7}$$

The above lemma shows that as long as there are sufficiently many points (majority in terms of fraction) inside the Euclidean ball of radius r centered at origin, then the geometric median must lie in the Euclidean ball blowed up by a constant factor only. Intuitively, geometric median can be viewed as an aggregated center of a set based on majority vote.

Let $\boldsymbol{g}_t(\theta_{t-1}) = \left(g_t^{(1)}(\theta_{t-1}), \ldots, g_t^{(m)}(\theta_{t-1})\right)$ be the m-dimensional vector that stacks the gradients received by the parameter server at iteration t. Let k be an integer which divides m and let $b = m/k$ denote the batch size. In our proposed robust gradient aggregation, the parameter server first divides m working machines into k batches, then takes the average of local gradients in each batch, and finally takes the geometric median of those k bath means. With the aggregated gradient, the parameter server performs a gradient descent update. Notice that when the number of batches $k = 1$,

Algorithm 13: Byzantine Gradient Descent: Synchronous iteration $t \ge 1$

Parameter server:

Initialize: Let θ_0 be an arbitrary point in Θ; group the m machines into k batches, with the ℓ-th batch being $\{(\ell-1)b+1, \ldots, \ell b\}$ for $1 \le \ell \le k$.

1: Broadcast the current model $\cdots$
2: Wait to receive all the gradients $\cdots$
3: *Robust Gradient Aggregation*

$$\mathcal{A}_k(\boldsymbol{g}_t(\theta_{t-1})) \leftarrow \mathsf{med}\left\{\frac{1}{b}\sum_{j=1}^{b} g_t^{(j)}(\theta_{t-1}),\ \cdots,\ \frac{1}{b}\sum_{j=n-b+1}^{n} g_t^{(j)}(\theta_{t-1})\right\}. \tag{5.8}$$

4: Update: $\theta_t \leftarrow \theta_{t-1} - \eta \times \mathcal{A}_k\big(\boldsymbol{g}_t(\theta_{t-1})\big)$;

Working machine j:

1: Compute the gradient $\nabla \bar{f}^{(j)}(\theta_{t-1})$;
2: Send $\nabla \bar{f}^{(j)}(\theta_{t-1})$ back to the parameter server;

the geometric median of means reduces to the average, i.e., $\mathcal{A}_1\{\boldsymbol{g}_t(\theta_{t-1})\} =$

$\frac{1}{m}\sum_{j=1}^{m} g_t^{(j)}(\theta_{t-1})$. When $k = m$, the median of means reduces to the geometric median $\mathcal{A}_m\{\boldsymbol{g}_t(\theta_{t-1})\} = \mathsf{med}\{g_t^{(1)}(\theta_{t-1}), \ldots, g_t^{(m)}(\theta_{t-1})\}$. Hence, the geometric median of means can be viewed as an interpolation between the mean and the geometric median. We assume q is known to the parameter server, who can choose the number of batches accordingly. We will discuss the choice of k after the statement of our main theorem. Informally, when q is small, the parameter server chooses a relatively small k – still larger than q. Small k is preferred since, based on the current analysis, it will lead to smaller estimation error guarantee. As q increases, the parameter server is forced to choose a larger k to prevent the system from being "controlled" by Byzantine machines. Since $k \leq m$, naturally there is an upper bound on q that can be tolerated.

Our correctness proof of Algorithm 13 relies on the key intermediate result that the aggregated gradient, as a function of θ defined in (5.8) for every θ, converges *uniformly* to the true gradient function $\nabla F(\theta)$.

5.3.2 Summary of Main Results

For ease of presentation, we present an informal statement of our main theorem. The precise statement and its proof are given in Section 5.4.3.

Theorem 26 (Informal)**.** *Suppose some mild technical assumptions hold and* $2(1+\epsilon)q \leq k \leq m$ *for any arbitrary but fixed constant* $\epsilon > 0$*. Fix any constant* $\alpha \in (1/2(1+\epsilon), 1/2)$ *and any* $\delta > 0$ *such that* $\delta \leq \alpha - q/k$ *and* $\log(1/\delta) = O(d)$*. There exist universal constants* $c_1, c_2 > 0$ *such that if* $N/k \geq c_1 C_\alpha^2 d \log(N/k)$*, then with probability at least*

$$1 - \exp(-kD((\alpha - q/k)\|\delta)),$$

the iterates $\{\theta_t\}$ *given by Algorithm 13 with* $\eta = L/(2M^2)$ *satisfy*

$$\|\theta_t - \theta^*\| \leq \left(\frac{1}{2} + \frac{1}{2}\sqrt{1 - \frac{L^2}{4M^2}}\right)^t \|\theta_0 - \theta^*\| + c_2 C_\alpha \sqrt{\frac{dk}{N}}, \quad \forall t \geq 1. \quad (5.9)$$

where $D(\delta'\|\delta) = \delta' \log \frac{\delta'}{\delta} + (1-\delta') \log \frac{1-\delta'}{1-\delta}$ *denotes the binary divergence. In particular,* $\limsup_{t\to\infty} \|\theta_t - \theta^*\| \leq c_2 C_\alpha \sqrt{dk/N}$*.*

Intuitively, the technical assumptions mentioned in Theorem 26 are placed

on the sample gradients $\nabla f(X_i, \theta)$ such that, with high probability, those sample gradients $\nabla f(X_i, \theta)$, as functions of θ, are good approximation of the population gradient $\nabla F(\theta)$.

As can be seen later, δ can be viewed as the expected fraction of batches that are "statistically bad"; the larger the batch sample size, i.e., N/k, the smaller δ. Since q/k is upper bounded by constant $1/2(1+\epsilon)$ – recalling that ϵ is a fixed constant, for sufficiently large N/k, we will have $\delta \leq \alpha - q/k$. In addition to the "statistically bad" batches, up to q/k fraction of the batches may contain Byzantine machines. In total, we might expect $\delta + q/k \leq \alpha$ fraction of bad batches. Intuitively speaking, the Theorem 26 says that as long as the total fraction of bad batches is less than $1/2$, we are able to show with high probability, our Byzantine Gradient Descent algorithm converges exponentially fast.

Notice that if we drop the assumption $\log(1/\delta) = O(d)$, our results still hold; however, both of the two terms on the right-hand side of (5.9) may be functions of δ. Additionally, the sample size at each batch, N/k, needed for (5.9) to hold also depend on δ. This "dependency" is characterized explicitly in the formal statement of our main theorem – Theorem 29.

Remark 6. *In this remark, we discuss the choice of k.*

- *In the failure-free case with $q = 0$, k can be chosen to be 1 and thus the geometric median of means reduces to simple averaging. The asymptotic estimation error rate is $\sqrt{d/N}$, which is the optimal estimation error rate even in the centralized setting.*

  For $q \geq 1$, we can also choose k to be $2(1+\epsilon)q$ for an arbitrarily small but fixed constant $\epsilon > 0$.

- *Based on our analysis, the number of batches k in our Byzantine gradient algorithm provides a trade-off between the statistical estimation error and the Byzantine failures: With a larger k, our algorithm can tolerate more Byzantine failures, but the estimation error gets larger. However, this trade-off may be due to our analysis only. Thus, it may not be fundamental.*

- *In terms of the probability mentioned in Theorem 26, it is not immediately clear how does this probability vary with k. To see this, consider the scenario when N, d, α and q are fixed: The smaller k, the*

larger N/k, the latter further implies a smaller δ. Because $\alpha - q/k$ is lower bounded by a constant that is independent of k, $D((\alpha - q/k)\|\delta)$ is roughly increasing in k. Thus, it is not immediately clear whether $k \cdot D((\alpha - q/k)\|\delta)$ is increasing in k or decreasing in k or neither increasing nor decreasing.

Our algorithm is both computation and communication efficient. Under the choice of k in Remark 6, the computation and communication cost of our proposed algorithm can be summarized as follows. For estimation error converging to $c_2\sqrt{dq/N}$, $O(\log N)$ communication rounds are sufficient. In each round, every working machine computes a gradient based on N/m local data samples, which takes $O(Nd/m)$ time steps. The parameter server computes the geometric median of means of gradients, which takes $O(md+qd\log^3(N))$, as the geometric median can be computed in $O(qd\log^3(N))$ [104]. In terms of communication cost, in each round, every working machine transmits a d-dimensional vector to the parameter server.

Application to Linear Regression

We illustrate our general results by applying them to the classical linear regression problem. Let $X_i = (w_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$ denote the input data and define the risk function $f(X_i, \theta) = \frac{1}{2}\left(\langle w_i, \theta\rangle - y_i\right)^2$. For simplicity, we assume that y_i is indeed generated from a linear model:

$$y_i = \langle w_i, \theta^*\rangle + \zeta_i,$$

where θ^* is an unknown true model parameter, $w_i \sim N(0, \mathbf{I})$ is the covariate vector whose covariance matrix is assumed to be identity, and $\zeta_i \sim N(0,1)$ is i.i.d. additive Gaussian noise independent of w_i's. Intuitively, the inner product $\langle w_i, \theta^*\rangle$ can be viewed as some "measurement" of θ^* – the signal; and ζ_i is the additive noise.

The population risk minimization problem (5.1) is simply

$$\min_{\theta} \quad \frac{1}{2}\|\theta - \theta^*\|_2^2 + \frac{1}{2},$$

where

$$F(\theta) \triangleq \mathbb{E}\left[f(X,\theta)\right] = \mathbb{E}\left[\frac{1}{2}\left(\langle w,\theta\rangle - y\right)^2\right]$$
$$= \mathbb{E}\left[\frac{1}{2}\left(\langle w,\theta\rangle - \langle w,\theta^*\rangle - \zeta\right)^2\right] = \frac{1}{2}\|\theta-\theta^*\|_2^2 + \frac{1}{2},$$

for which θ^* is indeed the unique minimum. If the function $F(\cdot)$ can be computed exactly, then θ^* can be read from its expression directly. The standard gradient descent method for minimizing $F(\cdot)$ is also straightforward. The population gradient is $\nabla_\theta F(\theta) = \theta - \theta^*$. It is easy to see that the population risk $\nabla F(\theta)$ is M-Lipschitz continuous with $M = 1$, and L-strongly convex with $L = 1$. Hence, Assumption 6 is satisfied with $M = L = 1$; and the stepsize $\eta = L/(2M^2) = 1/2$.

In practice, unfortunately, since we do not know exactly the distribution of the random input X, we can neither read θ^* from the expression $F(\cdot)$ nor compute the population gradient $\nabla F(\theta)$ exactly. We are only able to approximate the population risk $F(\cdot)$ or the population gradient $\nabla F(\theta)$. Our focus is the gradient approximation. In particular, for a given random sample, the associated random gradient is given by $\nabla f(X,\theta) = w\langle w,\theta-\theta^*\rangle - w\zeta$, where $w \sim \mathcal{N}(0,\mathbf{I})$ and $\zeta \sim \mathcal{N}(0,1)$ that is independent of w. We will show later that those sample gradients satisfy the "mild technical assumptions" mentioned in Theorem 26. Thus, according to Theorem 26, our Byzantine Gradient Descent method can robustly solve the linear regression problem exponentially fast with high probability – formally stated the following corollary.

Corollary 8 (Linear regression)**.** *Under the aforementioned least-squares model for linear regression, assume* $\Theta \subset \{\theta : \|\theta-\theta^*\| \le r\sqrt{d}\}$ *for* $r > 0$ *such that* $\log r = O(d\log(N/k))$*. Suppose that* $2(1+\epsilon)q \le k \le m$*. Fix any* $\alpha \in (q/k, 1/2)$ *and any* $\delta > 0$ *such that* $\delta \le \alpha - q/k$ *and* $\log(1/\delta) = O(d)$*, there exist universal constants* $c_1, c_2 > 0$ *such that if* $N/k \ge c_1 C_\alpha^2 d\log(N/k)$*. Then with probability at least* $1 - \exp(-kD((\alpha-q/k)\,\|\delta))$*, the iterates* $\{\theta_t\}$ *given by Algorithm 13 with* $\eta = 1/2$ *satisfy*

$$\|\theta_t - \theta^*\| \le \left(\frac{1}{2} + \frac{\sqrt{3}}{4}\right)^t \|\theta_0 - \theta^*\| + c_2 C_\alpha \sqrt{\frac{dk}{N}}, \quad \forall t \ge 1.$$

Note that in Corollary 8, we assume the "searching space" Θ belongs to some range, which may grow with d and N/k. This assumption is rather mild since in practice; we typically do have some prior knowledge about the range of θ^*.

5.4 Main Results and Proofs

In this section, we present our main results and their proofs.

Recall that in Algorithm 13, the machines are grouped into k batches beforehand. For each batch of machines $1 \leq \ell \leq k$, we define a function $Z_\ell : \Theta \to \mathbb{R}^d$ to be the *difference* between the average of the batch sample gradient functions and the population gradient, i.e., $\forall\, \theta \in \Theta$

$$\begin{aligned} Z_\ell(\theta) &\triangleq \frac{1}{b} \sum_{j=(\ell-1)b+1}^{\ell b} \nabla \bar{f}^{(j)}(\theta) - \nabla F(\theta) \\ &= \frac{k}{N} \sum_{j=(\ell-1)b+1}^{\ell b} \sum_{i \in \mathcal{S}_j} \nabla f(X_i, \theta) - \nabla F(\theta), \end{aligned} \tag{5.10}$$

where the last equality follows from (5.3) and the fact that batch size $b = m/k$ and local data size $|\mathcal{S}_j| = N/m$. It is easy to see that the functions $Z_\ell(\cdot)$'s are independently and identically distributed. For any given positive precision parameters ξ_1 and ξ_2 specified later, and $\alpha \in (0, 1/2)$, define a good event

$$\mathcal{E}_{\alpha,\xi_1,\xi_2} \triangleq \left\{ \sum_{\ell=1}^{k} \mathbf{1}_{\{\forall \theta:\, C_\alpha \|Z_\ell(\theta)\| \leq \xi_2 \|\theta - \theta^*\| + \xi_1\}} \geq k(1-\alpha) + q \right\}. \tag{5.11}$$

Informally speaking, on event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$, in at least $k(1-\alpha)+q$ batches, the average of the batch sample gradient functions is uniformly close to the population gradient function.

We show our convergence results of Algorithm 13 in two steps. The first step is "deterministic", showing that our Byzantine gradient descent algorithm converges exponentially fast on good event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$. The second part is "stochastic", proving that this good event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$ happens with high probability.

5.4.1 Convergence of Byzantine Gradient Descent on $\mathcal{E}_{\alpha,\xi_1,\xi_2}$

We consider a fixed execution. Let $\mathcal{B}_t$ denote the set of Byzantine machines at iteration t of the given execution, which could vary across iterations t. Define a vector of functions $\boldsymbol{g}_t(\cdot)$ with respect to $\mathcal{B}_t$ as:

$$\boldsymbol{g}_t(\theta) = (g_t^{(1)}(\theta), \ldots, g_t^{(m)}(\theta)), \ \forall\, \theta$$

such that $\forall\, \theta$,

$$g_t^{(j)}(\theta) = \begin{cases} \nabla \bar{f}^{(j)}(\theta) & \text{if } j \notin \mathcal{B}_t \\ \star & \text{o.w. ,} \end{cases}$$

where $\star$ is arbitrary[3]. That is, $g_t^{(j)}(\cdot)$ is the true gradient function $\bar{f}^{(j)}(\cdot)$ if machine j is not Byzantine at iteration t, and arbitrary otherwise. It is easy to see that the definition of $\boldsymbol{g}_t(\cdot)$ is consistent with the definition of $\boldsymbol{g}_t(\theta_{t-1})$ in (5.5). Define $\widetilde{Z}_\ell^t(\cdot)$ for each θ as

$$\widetilde{Z}_\ell(\theta) \triangleq \frac{1}{b} \sum_{j=(\ell-1)b+1}^{\ell b} g_t^{(j)}(\theta) - \nabla F(\theta). \tag{5.12}$$

By definition of $g_t^{(j)}(\cdot)$, for any ℓ-th batch such that

$$\{b(\ell-1)+1, \ldots, b\ell\} \cap \mathcal{B}_t = \emptyset,$$

i.e., it does not contain any Byzantine machine at iteration t, it holds that $\widetilde{Z}_\ell(\theta) = Z_\ell(\theta), \ \forall \theta$.

Lemma 16. *On event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$, for every iteration $t \geq 1$, we have*

$$\|\mathcal{A}_k\left(\boldsymbol{g}_t(\theta)\right) - \nabla F(\theta)\| \leq \xi_2 \|\theta - \theta^*\| + \xi_1, \quad \forall \theta \in \Theta.$$

Proof. By definition of $\mathcal{A}_k$ in (5.8), for any fixed θ,

$$\mathcal{A}_k(\boldsymbol{g}_t(\theta)) = \mathsf{med}\left\{ \frac{1}{b}\sum_{j=1}^{b} g_t^{(j)}(\theta), \ \frac{1}{b}\sum_{j=b+1}^{2b} g_t^{(j)}(\theta), \ \cdots, \ \frac{1}{b}\sum_{j=m-b+1}^{m} g_t^{(j)}(\theta) \right\}$$

[3]By "arbitrary" we mean that $g_t^{(j)}(\cdot)$ may not even be a function, and cannot be specified.

Since geometric median is invariant with translation, it follows that

$$\mathcal{A}_k(\boldsymbol{g}_t(\theta)) - \nabla F(\theta) = \mathsf{med}\left\{\widetilde{Z}_1(\theta),\ \cdots,\ \widetilde{Z}_m(\theta)\right\}.$$

On event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$, at least $k(1-\alpha)+q$ of the k batches $\{Z_\ell : 1 \le \ell \le k\}$ satisfy $C_\alpha\|Z_\ell(\theta)\| \le \xi_2\|\theta-\theta^*\| + \xi_1$ uniformly. Moreover, for Byzantine-free batch ℓ, it holds that $\widetilde{Z}_\ell(\cdot) = Z_\ell(\cdot)$. Hence, at least $k(1-\alpha)$ of the k *received* batches $\{\widetilde{Z}_\ell : 1 \le \ell \le k\}$ satisfy $C_\alpha\|\widetilde{Z}_\ell(\theta)\| \le \xi_2\|\theta-\theta^*\| + \xi_1$ uniformly. The conclusion readily follows from Lemma 15.

□

Convergence of Approximate Gradient Descent

Next, we show a convergence result of an approximate gradient descent, which might be of independent interest. For any $\theta \in \Theta$, define a new θ' as

$$\theta' = \theta - \eta \times \nabla F(\theta). \tag{5.13}$$

We remark that the above update is one step of population gradient descent given in (5.2).

Lemma 17. *Suppose Assumption 6 holds. If we choose the step size* $\eta = L/(2M^2)$*, then* θ' *defined in* (5.13) *satisfies that*

$$\|\theta' - \theta^*\| \le \sqrt{1 - L^2/(4M^2)}\|\theta - \theta^*\|. \tag{5.14}$$

The proof of Lemma 17 is rather standard, and is presented in Section 5.5.1 for completeness. Suppose that for each $t \ge 1$, we have access to gradient function $G_t(\cdot)$, which satisfy the uniform deviation bound:

$$\|G_t(\theta) - \nabla F(\theta)\| \le \xi_1 + \xi_2\|\theta - \theta^*\|, \quad \forall\theta, \tag{5.15}$$

for two positive precision parameters ξ_1, ξ_2 that are *independent* of t. Then we perform the following approximate gradient descent as a surrogate for population gradient descent:

$$\theta_t = \theta_{t-1} - \eta \times G_t(\theta_{t-1}). \tag{5.16}$$

The following lemma establishes the convergence of the approximate gradient descent.

Lemma 18. *Suppose Assumption 6 holds, and choose* $\eta = L/(2M^2)$. *If* (5.15) *holds for each* $t \geq 1$ *and*

$$\rho \triangleq 1 - \sqrt{1 - L^2/(4M^2)} - \xi_2 L/(2M^2) > 0,$$

then the iterates $\{\theta_t\}$ *in* (5.16) *satisfy*

$$\|\theta_t - \theta^*\| \leq (1-\rho)^t \|\theta_0 - \theta^*\| + \eta\xi_1/\rho.$$

Remark 7. *As it can be seen later, the precision parameter* ξ_2 *can be chosen to be a function of* N/k *such that* $\xi_2 \to 0$ *as* $N/k \to \infty$. *Thus, there exists* ξ_2 *for* ρ *defined in Lemma 18 to be positive.*

Proof of Lemma 18. Fix any $t \geq 1$, we have

$$\begin{aligned}
\|\theta_t - \theta^*\| &= \|\theta_{t-1} - \eta G_t(\theta_{t-1}) - \theta^*\| \\
&= \|\theta_{t-1} - \eta \nabla F(\theta_{t-1}) - \theta^* + \eta\left(\nabla F(\theta_{t-1}) - G_t(\theta_{t-1})\right)\| \\
&\leq \|\theta_{t-1} - \eta \nabla F(\theta_{t-1}) - \theta^*\| + \eta \left\|\nabla F(\theta_{t-1}) - G_t(\theta_{t-1})\right\|.
\end{aligned}$$

It follows from Lemma 17 that

$$\|\theta_{t-1} - \eta \nabla F(\theta_{t-1}) - \theta^*\| \leq \sqrt{1 - L^2/(4M^2)}\, \|\theta_{t-1} - \theta^*\|$$

and from (5.15) that

$$\|\nabla F(\theta_{t-1}) - G_t(\theta_{t-1})\| \leq \xi_1 + \xi_2 \|\theta_{t-1} - \theta^*\|.$$

Hence,

$$\|\theta_t - \theta^*\| \leq \left(\sqrt{1 - L^2/(4M^2)} + \eta\xi_2\right) \|\theta_{t-1} - \theta^*\|_2 + \eta\xi_1.$$

A standard telescoping argument then yields that

$$\begin{aligned}
\|\theta_t - \theta^*\| &\leq (1-\rho)^t \|\theta_0 - \theta^*\| + \eta\xi_1 \sum_{\tau=0}^{t-1} (1-\rho)^\tau \\
&\leq (1-\rho)^t \|\theta_0 - \theta^*\| + \eta\xi_1/\rho,
\end{aligned}$$

where $\rho = 1 - \sqrt{1 - L^2/(4M^2)} - \xi_2 L/(2M^2)$, and $\eta = L/(2M^2)$. □

Convergence of Byzantine Gradient Descent on Good Event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$

With Lemma 16 and the convergence of the approximate gradient descent (Lemma 18), we show that Algorithm 13 converges exponentially fast on good event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$.

Theorem 27. *Suppose event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$ holds and iterates $\{\theta_t\}$ are given by Algorithm 13 with $\eta = L/(2M^2)$. If $\rho = 1-\sqrt{1 - L^2/(4M^2)}-\xi_2 L/(2M^2) > 0$ as defined in Lemma 18, then*

$$\|\theta_t - \theta^*\| \leq (1-\rho)^t \|\theta_0 - \theta^*\| + \eta\xi_1/\rho. \tag{5.17}$$

Proof. In Algorithm 13, at iteration t, the parameter server updates the model parameter θ_{t-1} using the approximate gradient $\mathcal{A}_k\left(\boldsymbol{g}_t(\theta_{t-1})\right)$ – the value of the approximate gradient function $\mathcal{A}_k\left(\boldsymbol{g}_t(\cdot)\right)$ evaluated at θ_{t-1}. From Lemma 16, we know that on event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$

$$\|\mathcal{A}_k\left(\boldsymbol{g}_t(\theta)\right) - \nabla F(\theta)\| \leq \xi_2\|\theta - \theta^*\| + \xi_1, \quad \forall \theta \in \Theta.$$

The conclusion then follows from Lemma 18 by setting $G_t(\theta)$ to be $\mathcal{A}_k\left(\boldsymbol{g}_t(\theta)\right)$. □

5.4.2 Bound Probability of Good Event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$

Recall that for each batch ℓ for $1 \leq \ell \leq k$, Z_ℓ is defined in (5.10) w.r.t. the data samples collectively kept by the machines in this batch. Thus, function Z_ℓ is random. The following lemma gives a lower bound to the probability of good event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$.

Lemma 19. *Suppose for all $1 \leq \ell \leq k$, Z_ℓ satisfies*

$$\mathbb{P}\left\{\forall \theta : C_\alpha\|Z_\ell(\theta)\| \leq \xi_2\|\theta - \theta^*\| + \xi_1\right\} \geq 1 - \delta \tag{5.18}$$

for any $\alpha \in (q/k, 1/2)$ and $0 < \delta \leq \alpha - q/k$. Then

$$\mathbb{P}\left\{\mathcal{E}_{\alpha,\xi_1,\xi_2}\right\} \geq 1 - e^{-kD(\alpha - q/k\|\delta)}. \tag{5.19}$$

Proof. Let $T \sim \text{Binom}(k, 1-\delta)$. By assumption (5.18), the random variable

$$\sum_{\ell=1}^{k} \mathbf{1}_{\{\forall\theta: C_\alpha \|Z_\ell(\theta)\|_2 \le \xi_2 \|\theta-\theta^*\|+\xi_1\}}$$

first-order stochastically dominates T, i.e.,

$$\mathbb{P}\left\{\sum_{\ell=1}^{k} \mathbf{1}_{\{\forall\theta: C_\alpha \|Z_\ell(\theta)\|_2 \le \xi_2 \|\theta-\theta^*\|+\xi_1\}} \ge k(1-\alpha)+q\right\} \ge \mathbb{P}\left\{T \ge k(1-\alpha)+q\right\}. \tag{5.20}$$

By Chernoff's bound for binomial distributions, the following holds:

$$\mathbb{P}\left\{T \ge k(1-\alpha)+q\right\} \ge 1 - e^{-kD(\alpha-q/k\|\delta)}. \tag{5.21}$$

Combining (5.20) and (5.21) together, we conclude (5.19).

□

It remains to show the uniform convergence of Z_ℓ as required by (5.18). To this end, we need to impose a few technical assumptions that are rather standard [105]. Recall that gradient $\nabla f(X, \theta)$ is random as the input X is random. We assume gradient $\nabla f(X, \theta^*)$ is sub-exponential. The definition and some related concentration properties of sub-exponential random variables are presented in Section 5.5.3 for completeness.

Assumption 7. *There exist positive constants σ_1 and α_1 such that for any unit vector $v \in B$, $\langle \nabla f(X, \theta^*), v\rangle$ is sub-exponential with scaling parameters σ_1 and α_1, i.e.,*

$$\sup_{v\in B} \mathbb{E}\left[\exp\left(\lambda\langle \nabla f(X,\theta^*), v\rangle\right)\right] \le e^{\sigma_1^2\lambda^2/2}, \quad \forall|\lambda| \le \frac{1}{\alpha_1},$$

where B denotes the unit sphere $\{\theta : \|\theta\|_2 = 1\}$.

Intuitively speaking, Assumption 7 is placed to ensure that, with high probability, using the *true* sample gradient for individual batches, we are able to "identify" the optimal model θ^*. That is, with Assumption 7, we are able to bound the deviation of $(1/n)\sum_{i=1}^n \nabla f(X_i, \theta^*)$ from its mean $\nabla F(\theta^*) = 0$, as shown in the following lemma.

Lemma 20. *Suppose Assumption 7 holds. For any $\delta \in (0,1)$ and any positive integer n, let*

$$\Delta_1(n,d,\delta,\sigma_1) = \sqrt{2}\sigma_1\sqrt{\frac{d\log 6 + \log(3/\delta)}{n}}. \tag{5.22}$$

If $\Delta_1(n,d,\delta,\sigma_1) \le \sigma_1^2/\alpha_1$, then

$$\mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f(X_i,\theta^*) - \nabla F(\theta^*)\right\| \ge 2\Delta_1(n,d,\delta,\sigma_1)\right\} \le \frac{\delta}{3}.$$

Remark 8. *By definition of $\Delta_1(n,d,\delta,\sigma_1)$, for fixed δ and σ_1, if $d = o(n)$, $\Delta_1(n,d,\delta,\sigma_1)$ is a non-increasing function of n. In particular,*

$$\Delta_1(n,d,\delta,\sigma_1) = \sqrt{2}\sigma_1\sqrt{\frac{d\log 6 + \log(3/\delta)}{n}} \to 0 \qquad as \;\; n\to\infty.$$

Thus, there exists n for $\Delta_1(n,d,\delta,\sigma_1) \le \sigma_1^2/\alpha_1$ to hold.

With a little abuse of notation, we write $\Delta_1(n,d,\delta,\sigma_1)$ as Δ_1 or $\Delta_1(n)$ for short when its meaning is clear from the context.

Proof of Lemma 20. Let $\mathcal{V} = \{v_1,\ldots,v_{N_{1/2}}\}$ denote an $\frac{1}{2}$-cover of unit sphere B. It is shown in [105, Lemma 5.2, Lemma 5.3] that $\log N_{1/2} \le d\log 6$, and

$$\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f(X_i,\theta^*) - \nabla F(\theta^*)\right\| \le 2\sup_{v\in\mathcal{V}}\left\{\frac{1}{n}\sum_{i=1}^{n}\langle\nabla f(X_i,\theta^*) - \nabla F(\theta^*), v\rangle\right\}.$$

Note that since $\nabla F(\theta^*) = 0$, it holds that $\nabla f(X_i,\theta^*) - \nabla F(\theta^*) = \nabla f(X_i,\theta^*)$. By Assumption 7 and the condition that $\Delta_1 \le \sigma_1^2/\alpha_1$, it follows from concentration inequalities for sub-exponential random variables given in Theorem 30 that, for $v \in \mathcal{V}$

$$\mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n}\langle\nabla f(X_i,\theta^*) - \nabla F(\theta^*), v\rangle \ge \Delta_1\right\} \le \exp\left(-n\Delta_1^2/(2\sigma_1^2)\right).$$

Recall that in $\mathcal{V}$ contains at most 6^d vectors. In view of the union bound, it

further yields that

$$\begin{aligned}
&\mathbb{P}\left\{2\sup_{v\in\mathcal{V}}\left\{\frac{1}{n}\sum_{i=1}^{n}\langle\nabla f(X_i,\theta^*)-\nabla F(\theta^*),v\rangle\right\}\geq 2\Delta_1\right\}\\
&\leq\sum_{v\in\mathcal{V}}\mathbb{P}\left\{2\left\{\frac{1}{n}\sum_{i=1}^{n}\langle\nabla f(X_i,\theta^*)-\nabla F(\theta^*),v\rangle\right\}\geq 2\Delta_1\right\}\\
&\leq 6^d\exp\left(-n\Delta_1^2/(2\sigma_1^2)\right)=\exp\left(-n\Delta_1^2/(2\sigma_1^2)+d\log 6\right).
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f(X_i,\theta^*)-\nabla F(\theta^*)\right\|\geq 2\Delta_1\right\}\\
&\leq\mathbb{P}\left\{2\sup_{v\in\mathcal{V}}\left\{\frac{1}{n}\sum_{i=1}^{n}\langle\nabla f(X_i,\theta^*)-\nabla F(\theta^*),v\rangle\right\}\geq 2\Delta_1\right\}\\
&\leq\exp\left(-n\Delta_1^2/(2\sigma_1^2)+d\log 6\right).
\end{aligned}$$

The proof is complete. □

In addition to the "identifiability" of the optimal θ^* using sample gradients $\nabla f(X,\theta^*)$, similar to the smoothness requirements of the population gradient $\nabla F(\cdot)$ stated in Assumption 6, some smoothness properties (in stochastic sense) of the sample gradients $\nabla f(X,\cdot)$ are also desired. Next, we define gradient difference

$$h(x,\theta)\triangleq\nabla f(x,\theta)-\nabla f(x,\theta^*),\tag{5.23}$$

which characterizes the deviation of random gradient $\nabla f(x,\theta)$ from $\nabla f(x,\theta^*)$. Note that

$$\mathbb{E}\left[h(X,\theta)\right]=\nabla F(\theta)-\nabla F(\theta^*)\tag{5.24}$$

for each θ. The following assumptions ensure that for every θ, $h(x,\theta)$ normalized by $\|\theta-\theta^*\|$ is also sub-exponential.

Assumption 8. *There exist positive constants σ_2 and α_2 such that for any $\theta\in\Theta$ with $\theta\neq\theta^*$ and unit vector $v\in B$, $\langle h(X,\theta)-\mathbb{E}\left[h(X,\theta)\right],v\rangle/\|\theta-\theta^*\|$*

is sub-exponential with scaling parameters (σ_2, α_2)*, i.e.,*

$$\sup_{\theta\in\Theta, v\in B} \mathbb{E}\left[\exp\left(\frac{\lambda\langle h(X,\theta) - \mathbb{E}\left[h(X,\theta)\right], v\rangle}{\|\theta-\theta^*\|}\right)\right] \leq e^{\sigma_2^2\lambda^2/2}, \quad \forall|\lambda| \leq \frac{1}{\alpha_2}.$$

The following lemma bounds the deviation of $(1/n)\sum_{i=1}^n h(X_i,\theta)$ from $\mathbb{E}\left[h(X,\theta)\right]$ for every $\theta \in \Theta$ under Assumption 8. Its proof is similar to that of Lemma 20 and thus is omitted.

Lemma 21. *Suppose Assumption 8 holds and fix any* $\theta \in \Theta$*. Let*

$$\Delta_1'(n,d,\delta,\sigma_2) = \sqrt{2}\sigma_2\sqrt{\frac{d\log 6 + \log(3/\delta)}{n}}. \tag{5.25}$$

If $\Delta_1'(n,d,\delta,\sigma_2) \leq \sigma_2^2/\alpha_2$*, then*

$$\mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^n h(X_i,\theta) - \mathbb{E}\left[h(X,\theta)\right]\right\| > 2\Delta_1'(n,d,\delta,\sigma_2)\|\theta-\theta^*\|\right\} \leq \frac{\delta}{3}.$$

Remark 9. *Similar to* $\Delta_1(n,d,\delta,\sigma_2)$*, there also exists* n *for* $\Delta_1'(n,d,\delta,\sigma_2) \leq \sigma_2^2/\alpha_2$ *to hold.*

For ease of notation, we write $\Delta_1'(n,d,\delta,\sigma_2)$ as Δ_1' or $\Delta_1'(n)$ for short.

Assumption 7 and Assumption 8 can be potentially relaxed at an expense of looser concentration bounds. Note that Assumption 8, roughly speaking, only imposes some smoothness condition w. r. t. the optimal model θ^*. To mimic the Lipschitz continuity of the sample gradients (in stochastic sense), we impose the following assumption, which holds automatically if we strengthen Assumption 8 by replacing θ^* with an arbitrary θ' such that $\theta \neq \theta'$. In general, Assumption 9 is strictly weaker than the strengthened version of Assumption 8.

Assumption 9. *For any* $\delta \in (0,1)$*, there exists an* $M' = M'(n,\delta)$ *that is non-increasing in* n *such that*

$$\mathbb{P}\left\{\sup_{\theta,\theta'\in\Theta:\theta\neq\theta'} \frac{\|\frac{1}{n}\sum_{i=1}^n\left(\nabla f(X_i,\theta) - \nabla f(X_i,\theta')\right)\|}{\|\theta-\theta'\|} \leq M'\right\} \geq 1 - \frac{\delta}{3}.$$

Assumption 7–Assumption 9 of the sample gradients can be viewed as the corresponding stochastic version of Assumption 6 of the population gradient.

The following lemma verifies that Assumption 7–Assumption 9 are satisfied with appropriate parameters in the aforementioned linear regression example.

Lemma 22. *Under the linear regression model, the sample gradient function* $\nabla f(X, \cdot)$ *satisfies*
(1) Assumption 7 with $\sigma_1 = \sqrt{2}$ *and* $\alpha_1 = \sqrt{2}$,
(2) Assumption 8 with $\sigma_2 = \sqrt{8}$ *and* $\alpha_2 = 8$,
(3) and Assumption 9 with $M'(\delta) = d + 2\sqrt{d\log(4/\delta)} + 2\log(4/\delta)$.

The proof of Lemma 22 can be found in Section 5.5.2.

Define Δ_2 as follows.

$$\Delta_2(n) = \sigma_2\sqrt{\frac{2}{n}}\sqrt{d\log 18 + d\log\frac{M\vee M'}{\sigma_2} + \frac{1}{2}d\log\frac{n}{d} + \log\left(\frac{6\sigma_2^2 r\sqrt{n}}{\alpha_2\sigma_1\delta}\right)}. \tag{5.26}$$

With Assumption 7–Assumption 9, we apply the celebrated ϵ-net argument to prove the averaged random gradients $(1/n)\sum_{i=1}^n \nabla f(X_i, \theta)$ *uniformly* converges to $\nabla F(\theta)$.

Proposition 10. *Suppose Assumption 7 – Assumption 9 hold, and* $\Theta \subset \{\theta : \|\theta - \theta^*\| \le r\sqrt{d}\}$ *for some positive parameter* r. *For any* $\delta \in (0,1)$ *and any integer* n, *recall* Δ_1 *defined in* (5.22) *and define* Δ_2 *as in* (5.26). *If* $\Delta_1 \le \sigma_1^2/\alpha_1$ *and* $\Delta_2 \le \sigma_2^2/\alpha_2$, *then*

$$\mathbb{P}\left\{\forall\theta\in\Theta : \left\|\frac{1}{n}\sum_{i=1}^n \nabla f(X_i,\theta) - \nabla F(\theta)\right\| \le 8\Delta_2\|\theta-\theta^*\| + 4\Delta_1\right\} \ge 1-\delta.$$

Proof. The proof is based on the classical ϵ-net argument. Let

$$\tau = \frac{\alpha_2\sigma_1}{2\sigma_2^2}\sqrt{\frac{d}{n}} \quad \text{and} \quad \ell^* = \lceil r\sqrt{d}/\tau\rceil.$$

Henceforth, for ease of exposition, we assume ℓ^* is an integer. For integers $1 \le \ell \le \ell^*$, define

$$\Theta_\ell \triangleq \{\theta : \|\theta - \theta^*\| \le \tau\ell\}.$$

For a given ℓ, let $\theta_1, \ldots, \theta_{N_{\epsilon_\ell}}$ be an ϵ_ℓ-cover of Θ_ℓ, where ϵ_ℓ is given by

$$\epsilon_\ell = \frac{\sigma_2\tau\ell}{M\vee M'}\sqrt{\frac{d}{n}},$$

where $M \vee M' = \max\{M, M'\}$. By [105, Lemma 5.2], $\log N_{\epsilon_\ell} \leq d \log(3\tau\ell/\epsilon_\ell)$. Fix any $\theta \in \Theta_\ell$. There exists a $1 \leq j_\ell \leq N_{\epsilon_\ell}$ such that $\|\theta - \theta_{j_\ell}\|_2 \leq \epsilon_\ell$. By triangle's inequality,

$$\begin{aligned}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f(X_i,\theta) - \nabla F(\theta)\right\| \leq{}& \|\nabla F(\theta) - \nabla F(\theta_{j_\ell})\| \\ &+ \left\|\frac{1}{n}\sum_{i=1}^{n}\left(\nabla f(X_i,\theta) - \nabla f(X_i,\theta_{j_\ell})\right)\right\| \\ &+ \left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f(X_i,\theta_{j_\ell}) - \nabla F(\theta_{j_\ell})\right\|. \quad (5.27)\end{aligned}$$

In view of Assumption 6,

$$\|\nabla F(\theta) - \nabla F(\theta_{j_\ell})\| \leq M\|\theta - \theta_{j_\ell}\| \leq M\epsilon_\ell, \quad (5.28)$$

where the last inequality holds because by the construction of ϵ-net, and the fact that for a given θ, θ_{j_ℓ} is chosen in such a way that $\|\theta - \theta_{j_\ell}\| \leq \epsilon_\ell$.

Define event

$$\mathcal{E}_1 = \left\{\sup_{\theta,\theta'\in\Theta:\theta\neq\theta'} \frac{\|\frac{1}{n}\sum_{i=1}^{n}\left(\nabla f(X_i,\theta) - \nabla f(X_i,\theta')\right)\|}{\|\theta - \theta'\|} \leq M'\right\}.$$

By Assumption 9, we have $\mathbb{P}\{\mathcal{E}_1\} \geq 1 - \delta/3$. On event $\mathcal{E}_1$, it holds that

$$\sup_{\theta\in\Theta}\left\|\frac{1}{n}\sum_{i=1}^{n}\left(\nabla f(X_i,\theta) - \nabla f(X_i,\theta_{j_\ell})\right)\right\| \leq M'\epsilon_\ell. \quad (5.29)$$

By triangle's inequality again,

$$\begin{aligned}&\left\|\frac{1}{n}\sum_{i=1}^{n}\left(\nabla f(X_i,\theta_{j_\ell}) - \nabla F(\theta_{j_\ell})\right)\right\| \leq \left\|\frac{1}{n}\sum_{i=1}^{n}\left(\nabla f(X_i,\theta^*) - \nabla F(\theta^*)\right)\right\| \\ &\quad + \left\|\frac{1}{n}\sum_{i=1}^{n}\left(\nabla f(X_i,\theta_{j_\ell}) - \nabla f(X_i,\theta^*)\right) - \left(\nabla F(\theta_{j_\ell}) - \nabla F(\theta^*)\right)\right\| \\ &\leq \left\|\frac{1}{n}\sum_{i=1}^{n}\left(\nabla f(X_i,\theta^*) - \nabla F(\theta^*)\right)\right\| + \left\|\frac{1}{n}\sum_{i=1}^{n}h(X_i,\theta_{j_\ell}) - \mathbb{E}\left[h(X,\theta_{j_\ell})\right]\right\|, \quad (5.30)\end{aligned}$$

where function $h(x,\cdot)$ is defined in (5.23). Define event

$$\mathcal{E}_2 = \left\{ \left\| \frac{1}{n}\sum_{i=1}^{n} \nabla f(X_i,\theta^*) - \nabla F(\theta^*) \right\| \le 2\Delta_1 \right\}$$

and event

$$\mathcal{F}_\ell = \left\{ \sup_{1\le j\le N_\epsilon} \left\| \frac{1}{n}\sum_{i=1}^{n} h(X_i,\theta_j) - \mathbb{E}\left[h(X,\theta_j)\right] \right\| \le 2\tau\ell\Delta_2 \right\},$$

where Δ_2 is defined in (5.26) and satisfies

$$\Delta_2 = \sqrt{2}\sigma_2 \sqrt{\frac{d\log 6 + d\log(3\tau\ell/\epsilon_\ell) + \log(3\ell^*/\delta)}{n}}. \tag{5.31}$$

In (5.26), note that Δ_2 is independent of ℓ, due to the choice of ϵ_ℓ made earlier. It is easy to check that (5.26) and (5.31).

Since $\Delta_1 \le \sigma_1^2/\alpha_1$, it follows from Lemma 20 that $\mathbb{P}\{\mathcal{E}_2\} \ge 1-\delta/3$. Similarly, since $\Delta_2 \le \sigma_2^2/\alpha_2$, by Lemma 21, $\mathbb{P}\{\mathcal{F}_\ell\} \ge 1-\delta/(3\ell^*)$. In particular,

$$\begin{aligned}
\mathbb{P}\{\mathcal{F}_\ell^c\} &= \mathbb{P}\left\{ \sup_{1\le j\le N_{\epsilon_\ell}} \left\| \frac{1}{n}\sum_{i=1}^{n} h(X_i,\theta_j) - \mathbb{E}\left[h(X,\theta_j)\right] \right\| > 2\tau\ell\Delta_2 \right\} \\
&= \mathbb{P}\left\{ \exists_{1\le j\le N_{\epsilon_\ell}} \left\| \frac{1}{n}\sum_{i=1}^{n} h(X_i,\theta_j) - \mathbb{E}\left[h(X,\theta_j)\right] \right\| > 2\tau\ell\Delta_2 \right\} \\
&\le \sum_{j=1}^{N_{\epsilon_\ell}} \mathbb{P}\left\{ \left\| \frac{1}{n}\sum_{i=1}^{n} h(X_i,\theta_j) - \mathbb{E}\left[h(X,\theta_j)\right] \right\| > 2\tau\ell\Delta_2 \right\}.
\end{aligned} \tag{5.32}$$

For each $1 \le j \le N_{\epsilon_\ell}$, by Lemma 21, since $\theta_j \in \Theta_\ell$, it holds that $\|\theta_j - \theta^*\| \le \tau\ell$. Thus, we have

$$\begin{aligned}
&\mathbb{P}\left\{ \left\| \frac{1}{n}\sum_{i=1}^{n} h(X_i,\theta_j) - \mathbb{E}\left[h(X,\theta_j)\right] \right\| > 2\tau\ell\Delta_2 \right\} \\
&\le \mathbb{P}\left\{ \left\| \frac{1}{n}\sum_{i=1}^{n} h(X_i,\theta_j) - \mathbb{E}\left[h(X,\theta_j)\right] \right\| > 2\Delta_2\|\theta_j - \theta^*\| \right\} \\
&\le \frac{\delta}{3\ell^*} \frac{1}{\left(\frac{3\tau\ell}{\epsilon_\ell}\right)^d},
\end{aligned} \tag{5.33}$$

where the last inequality holds due to the choice of $\Delta_2(n)$ in (5.31). With

(5.33), we bound (5.32) as follows:

$$\begin{aligned}\mathbb{P}\left\{\mathcal{F}_\ell^c\right\} &\le \sum_{j=1}^{N_{\epsilon_\ell}} \mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^n h(X_i,\theta_j) - \mathbb{E}\left[h(X,\theta_j)\right]\right\| > 2\tau\ell\Delta_2\right\} \\ &\le \frac{\delta}{3\ell^*}\frac{1}{\left(\frac{3\tau\ell}{\epsilon_\ell}\right)^d}|N_{\epsilon_\ell}| \\ &= \frac{\delta}{3\ell^*}\frac{1}{\left(\frac{3\tau\ell}{\epsilon_\ell}\right)^d}\left(\frac{3\tau\ell}{\epsilon_\ell}\right)^d = \frac{\delta}{3\ell^*}.\end{aligned}$$

Therefore, we have $\mathbb{P}\left\{\mathcal{F}_\ell\right\} \ge 1 - \delta/(3\ell^*)$.

In conclusion, by combining (5.27), (5.28), (5.29) and (5.30), it follows that on event $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{F}_\ell$,

$$\sup_{\theta\in\Theta_\ell}\left\|\frac{1}{n}\sum_{i=1}^n \nabla f(X_i,\theta) - \nabla F(\theta)\right\| \le (M+M')\epsilon_\ell + 2\Delta_1 + 2\Delta_2\tau\ell \le 4\Delta_2\tau\ell + 2\Delta_1,$$

where the last inequality holds due to $(M \vee M')\epsilon_\ell \le \Delta_2\tau\ell$. Let

$$\mathcal{E} = \mathcal{E}_1 \cap \mathcal{E}_2 \cap \left(\cap_{\ell=1}^{\ell^*}\mathcal{F}_\ell\right).$$

It follows from the union bound, $\mathbb{P}\left\{\mathcal{E}\right\} \ge 1 - \delta$. Moreover, suppose event $\mathcal{E}$ holds. Then for all $\theta \in \Theta_{\ell^*}$, there exists an $1 \le \ell \le \ell^*$ such that $(\ell-1)\tau < \|\theta - \theta^*\| \le \ell\tau$. If $\ell \ge 2$, then

$$\left\|\frac{1}{n}\sum_{i=1}^n \nabla f(X_i,\theta) - \nabla F(\theta)\right\| \le 4\Delta_2\tau\ell + 2\Delta_1 \le 8\Delta_2\|\theta - \theta^*\| + 2\Delta_1.$$

If $\ell = 1$, then

$$\left\|\frac{1}{n}\sum_{i=1}^n \nabla f(X_i,\theta) - \nabla F(\theta)\right\| \le 4\Delta_2\tau + 2\Delta_1 \le 4\Delta_1,$$

where the last inequality follows from our choice of τ and the assumption that $\Delta_2 \le \sigma_2^2/\alpha_2$ and $\Delta_1 \ge \sigma_1\sqrt{d/n}$. In conclusion, on event $\mathcal{E}$,

$$\sup_{\theta\in\Theta_{\ell^*}}\left\|\frac{1}{n}\sum_{i=1}^n \nabla f(X_i,\theta) - \nabla F(\theta)\right\| \le 4\Delta_1 + 8\Delta_2\|\theta - \theta^*\|.$$

The proposition follows by the assumption that $\Theta \subset \Theta_{\ell^*}$.

□

Theorem 28. *Suppose Assumption 7 – Assumption 9 hold, and* $\Theta \subset \{\theta : \|\theta - \theta^*\| \le r\sqrt{d}\}$ *for some positive parameter* r*. For any* $\delta \in (0,1)$ *and any integer* n*, define* $\Delta_1(n)$ *and* $\Delta_2(n)$ *as in* (5.22) *and* (5.26)*, respectively. If* $\Delta_1(N/k) \le \sigma_1^2/\alpha_1$ *and* $\Delta_2(N/k) \le \sigma_2^2/\alpha_2$*, then for every* $1 \le \ell \le k$*,*

$$\mathbb{P}\left\{\forall \theta \in \Theta : C_\alpha \|Z_\ell(\theta)\| \le \xi_2 \|\theta - \theta^*\| + \xi_1\right\} \ge 1 - \delta,$$

where $\xi_1 = 4C_\alpha \times \Delta_1(N/k)$ *and* $\xi_2 = 8C_\alpha \times \Delta_2(N/k)$*.*

Proof. Recall that Z_ℓ is defined in (5.10). Note that for each $1 \le \ell \le k$, Z_ℓ has the same distribution as the average of N/k i.i.d. random gradients $f(X_i, \theta)$ subtracted by $\nabla F(\theta)$. Hence, Theorem 28 readily follows from Proposition 10. □

Remark 10. *Suppose* σ_1, α_1, σ_2, α_2 *are all of* $\Theta(1)$, $\log(M \vee M') = O(\log d)$, $\log(1/\delta) = O(d)$ *and* $\log r = O(d \log(N/k))$*. In this case, Theorem 28 implies that if* $N/k \gtrsim C_\alpha^2 d \log(N/k)$*, then*

$$\xi_1 \lesssim C_\alpha \sqrt{kd/N} \quad \text{and} \quad \xi_2 \lesssim C_\alpha \sqrt{kd \log(N/k)/N}.$$

In particular, those assumptions are indeed satisfied under the linear regression model as shown in Lemma 22.

5.4.3 Main Theorem

By combining Theorem 27, Lemma 19, and Theorem 28, we prove the main theorem.

Theorem 29. *Suppose Assumption 6 – Assumption 9 hold, and* $\Theta \subset \{\theta : \|\theta - \theta^*\| \le r\sqrt{d}\}$ *for some positive parameter* r*. Assume* $2(1+\epsilon)q \le k \le m$*. Fix any constant* $\alpha \in (q/k, 1/2)$ *and any* $\delta > 0$ *such that* $\delta \le \alpha - q/k$*. If*

$$\Delta_1(N/k) \le \sigma_1^2/\alpha_1, \qquad \Delta_2(N/k) \le \sigma_2^2/\alpha_2$$
$$\text{and} \quad \rho = 1 - \sqrt{1 - L^2/(4M^2)} - \xi_2 L/(2M^2) > 0$$

for $\xi_2 = 8C_\alpha \times \Delta_2(N/k)$*, then with probability at least*

$$1 - \exp(-kD(\alpha - q/k\|\delta)),$$

the iterates $\{\theta_t\}$ *given by Algorithm 13 with* $\eta = L/(2M^2)$ *satisfy*

$$\|\theta_t - \theta^*\| \le (1-\rho)^t \|\theta_0 - \theta^*\| + \eta\xi_1/\rho, \quad \forall t \ge 1,$$

where $\xi_1 = 4C_\alpha \times \Delta_1(N/k)$.

Under certain conditions, we are able to further bound ξ_1 and ξ_2. Next we present a formal statement of Theorem 26; it readily follows from Theorem 29 as a corollary.

Corollary 9. *Suppose that Assumption 6 – Assumption 9 hold such that* L*,* M*,* σ_1*,* α_1*,* σ_2*,* α_2 *are all of* $\Theta(1)$*,* $d = o(\frac{N/k}{\log N/k})$ *and* $\log M' = O(\log d)$*. Assume that* $\Theta \subset \{\theta : \|\theta - \theta^*\| \le r\sqrt{d}\}$ *for some positive parameter* r *such that* $\log(r) = O(d\log(N/k))$*, and* $2(1+\epsilon)q \le k \le m$*. Fix any* $\alpha \in (q/k, 1/2)$ *and any* $\delta > 0$ *such that* $\delta \le \alpha - q/k$ *and* $\log(1/\delta) = O(d)$*. There exist universal positive constants* c_1, c_2 *such that if* $N/k \ge c_1 C_\alpha^2 d\log(N/k)$*, then with probability at least* $1 - \exp(-kD(\alpha - q/k\|\delta))$*, the iterates* $\{\theta_t\}$ *given by Algorithm 13 with* $\eta = L/(2M^2)$ *satisfy*

$$\|\theta_t - \theta^*\| \le \left(\frac{1}{2} + \frac{1}{2}\sqrt{1 - \frac{L^2}{4M^2}}\right)^t \|\theta_0 - \theta^*\| + c_2\sqrt{\frac{dk}{N}}, \quad \forall t \ge 1.$$

Proof. Recall from (5.22) that

$$\Delta_1(N/k, d, \delta, \sigma_1) = \sqrt{2}\sigma_1\sqrt{\frac{d\log 6 + \log(3/\delta)}{N/k}}.$$

When $\sigma_1 = \Theta(1)$ and $\log(1/\delta) = O(d)$, it holds that $\Delta_1(N/k) = \Theta\left(\sqrt{kd/N}\right)$. Similarly, we have $\Delta_2(N/k) = \Theta\left(\sqrt{\frac{kd\log(N/k)}{N}}\right)$. Both Δ_1 and Δ_2 go to zero as N/k goes to infinity. Hence, there exists an universal positive constant c_1 such that for all $N/k \ge c_1 C_\alpha^2 d\log(N/k)$, it holds that $\Delta_1(N/k) \le \sigma_1^2/\alpha_1$,

$\Delta_2(N/k) \leq \sigma_2^2/\alpha_2$, and

$$\Delta_2(N/k) \leq \frac{M^2}{4C_\alpha L}\left(1 - \sqrt{1 - L^2/(4M^2)}\right). \tag{5.34}$$

So, for $\xi_2 = 4C_\alpha \times \Delta_2(N/k)$,

$$\rho = 1 - \sqrt{1 - L^2/(4M^2)} - \xi_2 L/(2M^2) \geq \frac{1}{2} - \frac{1}{2}\sqrt{1 - L^2/(4M^2)} > 0.$$

Recall that $\eta = L/(2M^2)$. The term $\eta\xi_1/\rho$ can be bounded as follows:

$$\eta\xi_1/\rho = \frac{L/(2M^2)4C_\alpha\Delta_1(N/k)}{\rho} \leq \frac{L/(2M^2)4C_\alpha\Delta_1(N/k)}{\frac{1}{2} - \frac{1}{2}\sqrt{1 - L^2/(4M^2)}} \leq c_2\sqrt{\frac{dk}{N}},$$

where c_2 is some universal constant.

Hence, the conclusion readily follows from Theorem 29. □

5.5 Additional Proofs

5.5.1 Proof of Lemma 17

Proof.

$$\begin{aligned}
\|\theta' - \theta^*\|^2 &= \|\theta - \theta^* - \eta\nabla F(\theta)\|^2 \qquad \text{by (5.13)} \\
&= \|\theta - \theta^* - \eta\left(\nabla F(\theta) - \nabla F(\theta^*)\right)\|^2 \qquad \text{since } \nabla F(\theta^*) = \mathbf{0} \\
&= \|\theta - \theta^*\|^2 + \eta^2\|\nabla F(\theta) - \nabla F(\theta^*)\|^2 \\
&\quad - 2\eta\langle\theta - \theta^*, \nabla F(\theta) - \nabla F(\theta^*)\rangle.
\end{aligned}$$

By Assumption 6, we have

$$\|\nabla F(\theta) - \nabla F(\theta^*)\| \leq M\|\theta - \theta^*\|,$$

and

$$F(\theta) \geq F(\theta^*) + \langle\nabla F(\theta^*), \theta - \theta^*\rangle + \frac{L}{2}\|\theta - \theta^*\|^2,$$

and

$$F(\theta^*) \geq F(\theta) + \langle\nabla F(\theta), \theta^* - \theta\rangle.$$

Summing up the last two displayed equations yields that

$$0 \geq \langle \nabla F(\theta) - \nabla F(\theta^*), \theta^* - \theta \rangle + \frac{L}{2}\|\theta - \theta^*\|^2.$$

Therefore,

$$\|\theta' - \theta^*\|^2 \leq \left(1 + \eta^2 M^2 - \eta L\right) \|\theta - \theta^*\|^2 .$$

The conclusion follows by the choosing $\eta = L/2M^2$. □

5.5.2 Proofs for Linear Regression Example

Proof of Lemma 22. **We first check Assumption 7**

Recall that $\nabla f(X, \theta) = w\langle w, \theta - \theta^* \rangle - w\zeta$, where $w \sim \mathcal{N}(0, \mathbf{I})$ and $\zeta \sim \mathcal{N}(0, 1)$ is independent of w. Hence, $\nabla f(X, \theta^*) = -w\zeta$. It follows that for any v in unit sphere B,

$$\langle \nabla f(X, \theta^*), v \rangle = -\zeta \langle w, v \rangle .$$

Because $w \sim \mathcal{N}(0, \mathbf{I})$ and are independent of ζ, it holds that $\langle w, v \rangle \sim \mathcal{N}(0, 1)$ and is independent of ζ. Thus, to compute $\mathbb{E}\left[\exp\left(-\lambda\zeta \langle w, v \rangle\right)\right]$, we can use the standard conditioning argument. In particular, for $\lambda^2 < 1$,

$$\begin{aligned} \mathbb{E}\left[\exp\left(\lambda \langle \nabla f(X, \theta^*), v \rangle\right)\right] &= \mathbb{E}\left[\exp\left(-\lambda\zeta \langle w, v \rangle\right)\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\exp\left(-\lambda y \langle w, v \rangle\right) | \zeta = y\right]\right], \end{aligned} \tag{5.35}$$

where the expectation of $\mathbb{E}\left[\exp\left(-\lambda y \langle w, v \rangle\right) | \zeta = y\right]$ is taken over the conditional distribution of $\langle w, v \rangle$ conditioning on ζ being y. Since $\langle w, v \rangle$ and ζ are independent of each other, the conditional distribution of $\langle w, v \rangle$ w. r. t. ζ is the same as the unconditional distribution of $\langle w, v \rangle$, which is a Gaussian distribution. Thus, we can apply the moment generating function of Gaussian distribution to get

$$\mathbb{E}\left[\exp\left(-\lambda y \langle w, v \rangle\right) | \zeta = y\right] = \exp\left(\lambda^2 y^2 / 2\right).$$

Then, the right-hand side of (5.35) becomes

$$\begin{aligned}\mathbb{E}\left[\exp\left(\lambda\left\langle\nabla f(X,\theta^*),v\right\rangle\right)\right] &= \mathbb{E}\left[\mathbb{E}\left[\exp\left(-\lambda y\left\langle w,v\right\rangle\right)|\zeta=y\right]\right] \\ &= \mathbb{E}\left[\exp\left(\lambda^2\zeta^2/2\right)\right] \\ &\overset{(a)}{=} (1-\lambda^2)^{-1/2}, \end{aligned} \tag{5.36}$$

where equality (a) follows from the moment generating function of χ^2 distribution, i.e.,

$$\mathbb{E}\left[\exp\left(t\zeta^2\right)\right] = (1-2t)^{-1/2} \quad \text{for} \;\; t < 1/2.$$

Using the fact that $1-\lambda^2 \geq e^{-2\lambda^2}$ for $\lambda^2 \leq 1/2$, it follows that

$$\mathbb{E}\left[\exp\left(\lambda\left\langle\nabla f(X,\theta),v\right\rangle\right)\right] \leq e^{\lambda^2}, \quad \forall|\lambda| \leq \frac{1}{\sqrt{2}}.$$

Thus Assumption 7 holds with $\sigma_1 = \sqrt{2}$ and $\alpha_1 = \sqrt{2}$.

Next, we verify Assumption 9. Note that $\nabla^2 f(X,\theta) = ww^\top$ and hence it suffices to show that

$$\mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^n \nabla^2 f(X_i,\theta)\right\| \leq M'\right\} = \mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^n w_i w_i^\top\right\| \leq M'\right\} \geq 1-\frac{\delta}{3},$$

for some M' depending on n, d, and δ.

Let $W = [w_1, w_2, \ldots, w_n]$ denote the $d \times n$ matrix whose columns are given by w_i's. Then $\sum_{i=1}^n w_i w_i^\top = WW^\top$. Also, the spectral norm of $WW^\top$ equals $\|W\|^2$. Therefore,

$$\mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^n w_i w_i^\top\right\| \leq M'\right\} = \mathbb{P}\left\{\|W\| \leq \sqrt{nM'}\right\}.$$

Note that W is an $d \times n$ matrix with i.i.d. standard Gaussian entries. Standard Gaussian matrix concentration inequality (see, e.g., [105, Corollary 5.35]) states that for every $t \geq 0$,

$$\mathbb{P}\left\{\|W\| \leq \sqrt{n} + \sqrt{d} + t\right\} \geq 1 - \exp(-t^2/2).$$

Plugging $t = \sqrt{2\log(4/\delta)}$ and setting

$$M' = \frac{1}{n}\left(\sqrt{n} + \sqrt{d} + \sqrt{2\log(4/\delta)}\right)^2$$

complete the proof.

Finally, we verify Assumption 8. Recall that the gradient difference $h(X,\theta)$ is given by $h(X,\theta) = w\langle w, \theta - \theta^*\rangle$, and $\mathbb{E}\left[h(X,\theta)\right] = \theta - \theta^*$. It follows that for any vector v in unit sphere B,

$$\langle h(X,\theta) - \mathbb{E}\left[h(X,\theta)\right], v\rangle = \langle w, \theta - \theta^*\rangle\langle w, v\rangle - \langle \theta - \theta^*, v\rangle.$$

For a fixed $\theta \in \Theta$ with $\theta \neq \theta^*$ and let $\tau = \|\theta - \theta^*\| > 0$. Then we have the following orthogonal decomposition: $\theta - \theta^* = \sqrt{\gamma}v + \sqrt{\eta}v_\perp$, where $\gamma + \eta = \tau^2$, and $v_\perp$ denote an vector perpendicular to v. It follows that

$$\langle w, \theta - \theta^*\rangle\langle w, v\rangle - \langle \theta - \theta^*, v\rangle = \sqrt{\gamma}\langle w, v\rangle^2 - \sqrt{\gamma} + \sqrt{\eta}\langle w, v_\perp\rangle\langle w, v\rangle.$$

It is easy to see that random variables $\langle w, v_\perp\rangle \sim \mathcal{N}(0,1)$ and $\langle w, v\rangle \sim \mathcal{N}(0,1)$ are jointly Gaussian. In addition, we have

$$\begin{aligned}\mathbb{E}\left[\langle w, v_\perp\rangle\langle w, v\rangle\right] &= \mathbb{E}\left[v_\perp^\top w w^\top v\right] \\ &= v_\perp^\top \mathbb{E}\left[ww^\top\right] v = v_\perp^\top \boldsymbol{I} v = 0.\end{aligned}$$

Thus, $\langle w, v_\perp\rangle \sim \mathcal{N}(0,1)$ and $\langle w, v\rangle \sim \mathcal{N}(0,1)$ are mutually independent.

For any λ with $\lambda\sqrt{\gamma} < 1/4$ and $\lambda^2\eta < 1/4$,

$$\begin{aligned}&\mathbb{E}\left[\exp\left(\lambda\langle h(X,\theta) - \mathbb{E}\left[h(X,\theta)\right], v\rangle\right)\right] \\ &= \mathbb{E}\left[\exp\left(\lambda\sqrt{\gamma}\left(\langle w, v\rangle^2 - 1\right) + \lambda\sqrt{\eta}\langle w, v_\perp\rangle\langle w, v\rangle\right)\right] \\ &\leq \sqrt{\mathbb{E}\left[e^{2\lambda\sqrt{\gamma}(\langle w,v\rangle^2 - 1)}\right]\mathbb{E}\left[e^{2\lambda\sqrt{\eta}\langle w,v_\perp\rangle\langle w,v\rangle}\right]} \\ &= e^{-\lambda\sqrt{\gamma}}\sqrt{\mathbb{E}\left[e^{2\lambda\sqrt{\gamma}(\langle w,v\rangle^2)}\right]}\sqrt{\mathbb{E}\left[e^{2\lambda\sqrt{\eta}\langle w,v_\perp\rangle\langle w,v\rangle}\right]} \\ &= e^{-\lambda\sqrt{\gamma}}\left(1 - 4\lambda\sqrt{\gamma}\right)^{-1/4}\left(1 - 4\lambda^2\eta\right)^{-1/4},\end{aligned}$$

where the first inequality holds due to Cauchy-Schwartz's inequality, and the last equality follows by plugging in the moment generating functions for χ^2 distributions as well as using the conditioning argument that is similar to the derivation of (5.35).

Using the fact that $e^{-t}/\sqrt{1-2t} \leq e^{2t^2}$ for $|t| \leq 1/4$ and $1-t \geq e^{-4t}$ for $0 \leq t \leq 1/2$, it follows that for $\lambda^2 \leq 1/(64\tau^2)$,

$$\mathbb{E}\left[\exp\left(\lambda\langle h(X,\theta) - \mathbb{E}\left[h(X,\theta)\right], v\rangle\right)\right] \leq \exp\left(4\lambda^2(\gamma+\eta)\right) \leq \exp\left(4\lambda^2\tau^2\right).$$

Hence, Assumption 8 holds with $\sigma_2 = \sqrt{8}$ and $\alpha_2 = 8$.

□

5.5.3 Concentration Inequality for Sub-exponential Random Variables

Definition 19 (Sub-exponential)**.** *Random variable X with mean μ is sub-exponential if $\exists\ \nu > 0$ and $\alpha > 0$ such that*

$$\mathbb{E}\left[\exp\left(\lambda(X-\mu)\right)\right] \leq \exp\left(\frac{\nu^2\lambda^2}{2}\right), \quad \forall|\lambda| \leq \frac{1}{\alpha}.$$

Theorem 30. *If $X_1, \ldots, X_n$ are independent random variables where X_i's are sub-exponential with scaling parameters (ν_i, α_i) and mean μ_i, then $\sum_{i=1}^n X_i$ is sub-exponential with scaling parameters (ν_*, α_*), where $\nu_*^2 = \sum_{i=1}^n \nu_i^2$ and $\alpha_* = max_{1\leq i\leq n}\alpha_i$. Moreover,*

$$\mathbb{P}\left\{\sum_{i=1}^n (X_i - \mu_i) \geq t\right\} \leq \begin{cases} \exp\left(-t^2/(2\nu_*^2)\right) & \text{if } 0 \leq t \leq \nu_*^2/\alpha_* \\ \exp\left(-t/(2\alpha_*)\right) & o.w. \end{cases}$$

CHAPTER 6

SUMMARY AND FUTURE DIRECTIONS

6.1 Dissertation Summary

There are many different descriptions of distributed systems, such as the swarm of drones, datacenters, manufactory plants, etc. In this dissertation, both peer to peer model (multi-agent network) and client-server model were explored. Our goal is to develop, via the concrete problems such as reaching consensus, multi-agent optimization, distributed hypothesis testing, and statistical learning, approaches for charactering the fundamental limits of the system's performance in the presence of malicious components, and to design efficient algorithms with optimal or near optimal performance.

We started the dissertation with investigating the consensus problem (Chapter 2), where a collection of networked processes/agents interact with each other using simple coordination rules to aggregate, in a distributed fashion, the scattered information.

Reaching consensus The existing work assume either local communication or full message forwarding. We addressed the impact of the number of hops allowed in a transmission on the computability of reaching consensus. Specifically, we assumed that in each iteration the processors can only communicate with other processors that are up to ℓ hops away, where ℓ is a positive integer. For a given ℓ, we identified a necessary and sufficient condition on the network structure for the existence of correct iterative algorithms that achieve asymptotic consensus in the presence of Byzantine agents. Our results bridged the above two lines of literature. In particular, our tight condition generalized the tight condition identified in existing work for $\ell = 1$, i.e., local communication. For $\ell \geq \ell^*$, where ℓ^* is the length of a longest cycle-free path in the given network, our condition is equivalent to the tight

conditions obtained for full message forwarding communication.

Following the consensus chapter, we studied two lines of research: Consensus-based multi-agent optimization (Chapter 3) and consensus-based distributed hypothesis testing (Chapter 4). In both of these chapters, we mainly focused on the family of algorithms in which the agents/processes interact with each other using the simple coordination rules that are similar to the one discussed in Chapter 2.

Consensus-Based Multi-Agent Optimization We first showed that when there exists an unknown agent that may be compromised and behave maliciously, it is impossible to minimize

$$\frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}} h_i,$$

where $\mathcal{N}$ is the unknown set of *good* agents. One important implication of the above impossibility result is, it is *impossible* to solve the "empirical risk minimization" problem exactly, when the training data samples are scattered over the entire network. Specifically, it is impossible to assign equal weights to the data collected by all the good agents.

In this dissertation, we characterized the performance degradation caused by the existence of malicious agents, and designed efficient algorithms that can achieve the optimal (the best possible) performance.

Consensus-Based Distributed Hypothesis Testing Bayesian learning approaches have been well-studied. However, these methods may only work when all the networked agents are failure-free – cooperative. This is because when some of the agents become adversarial and behave arbitrarily, the problem itself may not be described as a fully probabilistic problem. As a result of this, Bayesian approaches in the presence of Byzantine agents may not even be "well-defined".

We followed the non-Bayesian learning framework proposed by Jadbabaie et al. [11] (which is originally proposed for the failure-free setting) that combines local Bayesian learning with consensus. This dissertation addressed the problem of developing distributed learning algorithms that are robust to

adversarial attacks. We proposed the first Byzantine-resilient learning algorithm [12], and characterized tight network identifiability condition in [13] – the extended version of [12]. At first glance, our learning rule is counter-intuitive: by applying the cumulative likelihood, the "old information" contained in the previous signals is used again and again in updating local pseudo beliefs. It turns out that this learning rule enables us to deal with the dependence between the pseudo beliefs and the effective message propagation, which is rather crucial in our adversarial attacks setting.

In the subsequent chapter (Chapter 5), observing the trends in collaborative machine learning (mobile + cloud computing), we explored the problem of performing distributed machine learning in the adversarial setting. One key distinction of the distributed system assumed in Chapter 5 from the one discussed in Chapters 2, 3 and 4 is the existence of a parameter server used for the inter-agent coordination.

Distributed Statistical Machine Learning in Adversarial Settings In the distributed machine learning, we assumed the system consists of a parameter server and a collection of working clients – a typical distributed machine learning model. Due to the existence of a "centralized " server, working clients do not have to run consensus iterates for sharing the information.

We focused on the security problem faced by Google's *Federated Learning* – a new distributed machine learning paradigm initialized by Google. We developed a new iterative distributed machine learning algorithm that is able to (1) tolerate Byzantine failures, (2) accurately learn a highly complex model with low local data volume, and (3) converge exponentially fast using logarithmic communication rounds.

6.2 Future Directions

Implementation In addition to extending and generalizing the results contained in this dissertation, we also would like to explore and improve the practical performance of the adversary-resilient algorithms proposed here.

For example, for the statistical machine learning problem, we would like to test its real performance and modify our algorithms accordingly.

Game-Theoretic Model This dissertation mainly focuses on the Byzantine fault model, where some unknown subset of agents may be malicious and behave arbitrarily. The goal of the malicious agents is to create more obstacles for the system to achieve the system's goal. One direction that we would like to explore is the game-theoretic setup. For example, we might have two groups of agents located in the same multi-agent network. One group of agents mimics the roles of the Byzantine agents in the dissertation: they may be located arbitrarily in the network and can collaborate with each other. The good agents do not know the identity of the other agents, but need to collaborate with the good agents to solve some optimization problem. The bad agents may want to interrupt the collaboration to have the good agents output a bad estimator.

REFERENCES

[1] P. K. Sinha, *Distributed Operating Systems: Concepts and Design.* PHI Learning Pvt. Ltd., 1998.

[2] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *J. ACM*, vol. 27, pp. 228–234, April 1980. [Online]. Available: http://doi.acm.org/10.1145/322186.322188

[3] J. Duchi, A. Agarwal, and M. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *Automatic Control, IEEE Transactions on*, vol. 57, no. 3, pp. 592–606, March 2012.

[4] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *Automatic Control, IEEE Transactions on*, vol. 54, no. 1, pp. 48–61, Jan 2009.

[5] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Push-sum distributed dual averaging for convex optimization," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, Dec 2012, pp. 5453–5458.

[6] J. Chen and A. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *Signal Processing, IEEE Transactions on*, vol. 60, no. 8, pp. 4289–4305, August 2012.

[7] B. Johansson, "On distributed optimization in networked systems," *doctoral dissertation, KTH*, 2008.

[8] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *Automatic Control, IEEE Transactions on*, vol. 56, no. 6, pp. 1291–1306, June 2011.

[9] A. Agarwal, O. Chapelle, M. Dudík, and J. Langford, "A reliable effective terascale linear learning system," *Journal of Machine Learning Research*, vol. 15, pp. 1111–1133, 2014. [Online]. Available: http://jmlr.org/papers/v15/agarwal14a.html

[10] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 2. IEEE, 2002, pp. 1327–1332.

[11] A. Jadbabaie, P. Molavi, A. Sandroni, and A. Tahbaz-Salehi, "Non-bayesian social learning," *Games and Economic Behavior*, vol. 76, no. 1, pp. 210–225, 2012.

[12] L. Su and N. H. Vaidya, "Non-Bayesian learning in the presence of Byzantine agents," in *International Symposium on Distributed Computing*. Springer Berlin Heidelberg, 2016, pp. 414–427.

[13] L. Su and N. H. Vaidya, "Defending non-Bayesian learning against adversarial attacks," *arXiv: 1606.08883*, 2016.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[15] M. I. Jordan, J. D. Lee, and Y. Yang, "Communication-efficient distributed statistical inference," *arXiv preprint arXiv:1605.07689*, 2016.

[16] P. Moritz, R. Nishihara, I. Stoica, and M. I. Jordan, "Sparknet: Training deep networks in spark," *arXiv preprint arXiv:1511.06051*, 2015.

[17] F. J. Provost and D. N. Hennessy, "Scaling up: Distributed machine learning with cooperation," in *AAAI/IAAI, Vol. 1*. Citeseer, 1996, pp. 74–79.

[18] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[19] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein, "Distributed graphlab: a framework for machine learning and data mining in the cloud," *Proceedings of the VLDB Endowment*, vol. 5, no. 8, pp. 716–727, 2012.

[20] C. Kaufman, R. Perlman, and M. Speciner, *Network Security: Private Communication in a Public World*. Prentice Hall, Press, 2002.

[21] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*, 3rd ed. Prentice Hall Professional Technical Reference, 2002.

[22] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Infocom, 2010 Proceedings IEEE*, 2010, pp. 1–9.

[23] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *SIGMOD Rec.*, vol. 29, no. 2, pp. 439–450, May 2000.

[24] J. Duchi, M. J. Wainwright, and M. I. Jordan, "Local privacy and minimax bounds: Sharp rates for probability estimation," in *Advances in Neural Information Processing Systems*, 2013, pp. 1529–1537.

[25] "Wikipedia: icloud leaks of celebrity photos," https://en.wikipedia.org/wiki/ICloud_leaks_of_celebrity_photos, accessed: 2016-04-01.

[26] "Prism (surveillance program)," https://en.wikipedia.org/wiki/PRISM_(surveillance_program), accessed: 2016-04-01.

[27] J. Konečnỳ, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," *arXiv preprint arXiv:1511.03575*, 2015.

[28] "Federated learning: Collaborative machine learning without centralized training data," https://research.googleblog.com/2017/04/federated-learning-collaborative.html, accessed: 2017-04-10.

[29] N. A. Lynch, *Distributed Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996.

[30] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *J. ACM*, vol. 27, no. 2, pp. 228–234, Apr. 1980. [Online]. Available: http://doi.acm.org/10.1145/322186.322188

[31] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *J. ACM*, vol. 32, pp. 374–382, April 1985. [Online]. Available: http://doi.acm.org/10.1145/3149.214121

[32] D. Dolev, N. A. Lynch, S. S. Pinter, E. W. Stark, and W. E. Weihl, "Reaching approximate agreement in the presence of faults," *J. ACM*, vol. 33, no. 3, pp. 499–516, May 1986. [Online]. Available: http://doi.acm.org/10.1145/5925.5931

[33] H. J. LeBlanc, H. Zhang, S. Sundaram, and X. Koutsoukos, "Consensus of multi-agent networks in the presence of adversaries using only local information," in *Proceedings of the 1st International Conference on High Confidence Networked Systems*, ser. HiCoNS '12. New York, NY, USA: ACM, 2012. [Online]. Available: http://doi.acm.org/10.1145/2185505.2185507 pp. 1–10.

[34] 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*. ACM, 2012, pp. 365–374.

[35] M. J. Fischer, N. A. Lynch, and M. Merritt, "Easy impossibility proofs for distributed consensus problems," in *Proceedings of the Fourth Annual ACM Symposium on Principles of Distributed Computing*, ser. PODC '85. New York, NY, USA: ACM, 1985. [Online]. Available: http://doi.acm.org/10.1145/323596.323602 pp. 59–70.

[36] L. Tseng and N. H. Vaidya, "Fault-tolerant consensus in directed graphs," in *Proceedings of ACM Symposium on Principles of Distributed Computing*, 2015, pp. 451–460.

[37] F. Benezit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli, "Weighted gossip: Distributed averaging using non-doubly stochastic matrices," in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, June 2010, pp. 1753–1757.

[38] N. H. Vaidya, "Matrix representation of iterative approximate byzantine consensus in directed graphs," *CoRR*, vol. abs/1203.1888, 2012. [Online]. Available: http://arxiv.org/abs/1203.1888

[39] A. D. Fekete, "Asymptotically optimal algorithms for approximate agreement," *Distributed Computing*, vol. 4, no. 1, pp. 9–29, 1990.

[40] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[41] J. Ali, L. Jie, and A. S. Morse., "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *Automatic Control, IEEE Transactions on*, vol. 48, no. 6, pp. 988–1001, June 2003.

[42] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 2, no. 27, pp. 71–82, 2007.

[43] D. B. West et al., *Introduction to Graph Theory*. Prentice Hall, Upper Saddle River, 2001, vol. 2.

[44] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, Cambridge, 2001, vol. 2.

[45] H. Zhang and S. Sundaram, "Robustness of information diffusion algorithms to locally bounded adversaries," in *American Control Conference (ACC), 2012*, June 2012, pp. 5855–5861.

[46] J. Wolfowitz, "Products of indecomposable, aperiodic, stochastic matrices," *Proceedings of the American Mathematical Society*, vol. 14, no. 5, pp. 733–737, 1963.

[47] J. Hajnal and M. Bartlett, "Weak ergodicity in non-homogeneous markov chains," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 54, no. 02. Cambridge Univ Press, 1958, pp. 233–246.

[48] L. Su and N. Vaidya, "Reaching approximate byzantine consensus with multi-hop communication," in *Proceedings of Stabilization, Safety, and Security of Distributed Systems (SSS)*, 2015. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-21741-3_2 pp. 21–35.

[49] N. A. Lynch, *Distributed Algorithms*. Morgan Kaufmann, 1996.

[50] S. Chaudhuri, "More choices allow more faults: Set consensus problems in totally asynchronous systems," *Information and Computation*, vol. 105, pp. 132–158, 1992.

[51] A. Mostefaoui, S. Rajsbaum, and M. Raynal, "Conditions on input vectors for consensus solvability in asynchronous distributed systems," *Journal of the ACM (JACM)*, vol. 50, no. 6, pp. 922–954, 2003.

[52] R. Friedman, A. Mostefaoui, S. Rajsbaum, and M. Raynal, "Asynchronous agreement and its relation with error-correcting codes," *Computers, IEEE Transactions on*, vol. 56, no. 7, pp. 865–875, 2007.

[53] N. H. Vaidya, "Iterative Byzantine vector consensus in incomplete graphs," in *Distributed Computing and Networking*. Springer, 2014, pp. 14–28.

[54] L. Tseng and N. H. Vaidya, "Iterative approximate consensus in the presence of Byzantine link failures," in *Networked Systems*. Springer International Publishing, 2014, pp. 84–98.

[55] L. Tseng and N. H. Vaidya, "Iterative approximate Byzantine consensus under a generalized fault model," in *Distributed Computing and Networking*. Springer, 2013, pp. 72–86.

[56] D. Stolz and R. Wattenhofer, "Byzantine agreement with median validity," in *19th International Conference on Priniciples of Distributed Systems (OPODIS), Rennes, France*, 2015.

[57] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1561/2200000016

[58] J. N. Tsitsiklis, "Problems in decentralized decision making and computation." DTIC Document, Tech. Rep., 1984.

[59] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *Automatic Control, IEEE Transactions on*, vol. 31, no. 9, pp. 803–812, Sep 1986.

[60] S. Sundaram and B. Gharesifard, "Consensus-based distributed optimization with malicious nodes," in *Proceedings of the 53rd Annual Allerton Conference on Communication, Control and Computing*. IEEE, 2015.

[61] L. Su and N. H. Vaidya, "Byzantine multi-agent optimization: Part I," *arXiv preprint arXiv:1506.04681*, 2015.

[62] L. Su and N. H. Vaidya, "Byzantine multi-agent optimization: Part II," *CoRR*, vol. abs/1507.01845, 2015. [Online]. Available: http://arxiv.org/abs/1507.01845

[63] L. Su and N. Vaidya, "Fault-tolerant multi-agent optimization: Part III," *arXiv preprint arXiv:1509.01864*, 2015.

[64] L. Su and N. H. Vaidya, "Fault-tolerant distributed optimization (Part IV): Constrained optimization with arbitrary directed networks," *arXiv preprint arXiv:1511.01821*, 2015.

[65] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *CoRR*, vol. abs/1606.08939, 2016. [Online]. Available: http://arxiv.org/abs/1606.08939

[66] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. on Programming Languages and Systems*, 1982.

[67] Y. Nesterov, *Introductory Lectures on Convex Optimization*. Springer Science & Business Media, 2004, vol. 87.

[68] A. Nedic, *Lecture Notes on Optimization*. University of Illinois, 2008.

[69] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[70] H. Robbins and D. Siegmund, "A convergence theorem for non negative almost supermartingales and some applications," in *Herbert Robbins Selected Papers*, T. Lai and D. Siegmund, Eds. Springer, New York, 1985, pp. 111–135. [Online]. Available: http://dx.doi.org/10.1007/978-1-4612-5110-1_10

[71] B. T. Poljak, *Introduction to Optimization*. Optimization Software, 1987.

[72] I. Abraham, Y. Amit, and D. Dolev, "Optimal resilience asynchronous approximate agreement," in *Principles of Distributed Systems*. Springer, 2005, pp. 229–239.

[73] 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)*, vol. 33, no. 3, pp. 499–516, 1986.

[74] H. Mendes and M. Herlihy, "Multidimensional approximate agreement in byzantine asynchronous systems," in *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, ser. STOC '13. New York, NY, USA: ACM, 2013. [Online]. Available: http://doi.acm.org/10.1145/2488608.2488657 pp. 391–400.

[75] N. H. Vaidya and V. K. Garg, "Byzantine vector consensus in complete graphs," in *Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing*. ACM, 2013, pp. 65–73.

[76] J. N. Tsitsiklis, "Decentralized detection by a large number of sensors," *Mathematics of Control, Signals and Systems*, vol. 1, no. 2, pp. 167–182, 1988. [Online]. Available: http://dx.doi.org/10.1007/BF02551407

[77] J. N. Tsitsiklis et al., "Decentralized detection," *Advances in Statistical Signal Processing*, vol. 2, no. 2, pp. 297–344, 1993.

[78] P. K. Varshney, *Distributed Detection and Data Fusion*. Springer Science & Business Media, 2012.

[79] D. Gale and S. Kariv, "Bayesian learning in social networks," *Games and Economic Behavior*, vol. 45, no. 2, pp. 329–346, 2003.

[80] F. S. Cattivelli and A. H. Sayed, "Distributed detection over adaptive networks using diffusion adaptation," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 1917–1932, 2011.

[81] D. Jakovetic, J. M. Moura, and J. Xavier, "Distributed detection over noisy networks: Large deviations analysis," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4306–4320, 2012.

[82] A. Jadbabaie, P. Molavi, and A. Tahbaz-Salehi, "Information heterogeneity and the speed of learning in social networks," *Columbia Business School Research Paper*, no. 13-28, 2013.

[83] A. Nedić, A. Olshevsky, and C. A. Uribe, "Nonasymptotic convergence rates for cooperative learning over time-varying directed graphs," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 5884–5889.

[84] K. R. Rad and A. Tahbaz-Salehi, "Distributed parameter estimation in networks," in *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 5050–5055.

[85] S. Shahrampour and A. Jadbabaie, "Exponentially fast parameter estimation in networks using distributed dual averaging," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 6196–6201.

[86] A. Lalitha, A. Sarwate, and T. Javidi, "Social learning and distributed hypothesis testing," in *2014 IEEE International Symposium on Information Theory*. IEEE, 2014, pp. 551–555.

[87] S. Shahrampour, A. Rakhlin, and A. Jadbabaie, "Finite-time analysis of the distributed detection problem," in *53rd Annual Allerton Conference on Communication, Control, and Computing, 2015*, 2015, pp. 598–603.

[88] S. Shahrampour, A. Rakhlin, and A. Jadbabaie, "Distributed detection: Finite-time analysis and impact of network topology," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3256–3268, 2016.

[89] P. Molavi, A. Tahbaz-Salehi, and A. Jadbabaie, "Foundations of non-bayesian social learning," *Columbia Business School Research Paper*, 2015.

[90] H. Tverberg, "A generalization of radon's theorem," *Journal of the London Mathematical Society*, vol. s1-41, no. 1, pp. 123–128, 1966. [Online]. Available: http://jlms.oxfordjournals.org/content/s1-41/1/123.short

[91] N. A. Lynch, *Distributed Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996.

[92] Y. Zhang, J. C. Duchi, and M. J. Wainwright, "Communication-efficient algorithms for statistical optimization," *Journal of Machine Learning Research*, vol. 14, pp. 3321–3363, 2013. [Online]. Available: http://jmlr.org/papers/v14/zhang13b.html

[93] Y. Zhang, J. Duchi, and M. Wainwright, "Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates," *J. Mach. Learn. Res*, vol. 16, pp. 3299–3340, 2015.

[94] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, and J. Stainer, "Byzantine-tolerant machine learning," *arXiv preprint arXiv:1703.02757*, 2017.

[95] J. Feng, H. Xu, and S. Mannor, "Distributed robust learning," *arXiv preprint arXiv:1409.5937*, 2014.

[96] H. P. Lopuhaa and P. J. Rousseeuw, "Breakdown points of affine equivariant estimators of multivariate location and covariance matrices," *The Annals of Statistics*, pp. 229–248, 1991.

[97] S. Minsker et al., "Geometric median and robust estimation in Banach spaces," *Bernoulli*, vol. 21, no. 4, pp. 2308–2335, 2015.

[98] S. Mei, Y. Bai, and A. Montanari, "The landscape of empirical risk for non-convex losses," *arXiv preprint arXiv:1607.06534*, 2016.

[99] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Belmont, MA: Athena Scientific, 1998.

[100] J. Möttönen, K. Nordhausen, H. Oja et al., "Asymptotic theory of the spatial median," in *Nonparametrics and Robustness in Modern Statistical Inference and Time Series Analysis: A Festschrift in honor of Professor Jana Jurečková*. Institute of Mathematical Statistics, 2010, pp. 182–193.

[101] P. Milasevic, G. Ducharme et al., "Uniqueness of the spatial median," *The Annals of Statistics*, vol. 15, no. 3, pp. 1332–1333, 1987.

[102] J. Kemperman, "The median of a finite measure on a Banach space," *Statistical data analysis based on the L1-norm and related methods (Neuchâtel, 1987)*, pp. 217–230, 1987.

[103] H. Cardot, P. Cénac, P.-A. Zitt et al., "Efficient and fast estimation of the geometric median in Hilbert spaces with an averaged stochastic gradient algorithm," *Bernoulli*, vol. 19, no. 1, pp. 18–43, 2013.

[104] M. B. Cohen, Y. T. Lee, G. Miller, J. Pachocki, and A. Sidford, "Geometric median in nearly linear time," in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2016, pp. 9–21.

[105] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," *Arxiv preprint arxiv:1011.3027*, 2010.