Journal of Machine Learning Research 26 (2025) 1-58 Submitted 3/25; Revised 9/25; Published 10/25

Optimal Complexity in Byzantine-Robust Distributed Stochastic Optimization with Data Heterogeneity

Qiankun Shi SHIQK@MAIL2.SYSU.EDU.CN
School of Computer Science and Engineering
Sun Yat-Sen University
Guangzhou, China
Pengcheng Laboratory
Shenzhen, China

Jie Peng PENGJ95@MAIL2.SYSU.EDU.CN
School of Computer Science and Engineering
Sun Yat-Sen University
Guangzhou, China

Kun Yuan KUNYUAN@PKU.EDU.CN
Center for Machine Learning Research
Peking University
Beijing, China

Xiao Wang WANGX936@MAIL.SYSU.EDU.CN
School of Computer Science and Engineering
Sun Yat-Sen University
Guangzhou, China

Qing Ling LINGQING556@MAIL.SYSU.EDU.CN
School of Computer Science and Engineering
Sun Yat-Sen University
Guangzhou, China

Editor: Shiqian Ma

Abstract

In this paper, we establish tight lower bounds for Byzantine-robust distributed first-order stochastic methods in both strongly convex and non-convex stochastic optimization. We reveal that when the distributed nodes have heterogeneous data, the convergence error comprises two components: a non-vanishing Byzantine error and a vanishing optimization error. We establish the lower bounds on the Byzantine error and on the minimum number of queries to a stochastic gradient oracle for achieving an arbitrarily small optimization error. Nevertheless, we also identify significant discrepancies between our established lower bounds and the existing upper bounds. To fill this gap, we leverage the techniques of Nesterov's acceleration and variance reduction to develop novel Byzantine-robust distributed stochastic optimization methods that provably match these lower bounds, up to at most logarithmic factors, implying that our established lower bounds are tight.

Keywords: Distributed Optimization, Stochastic Optimization, Byzantine Robustness, Complexity Bounds

1. Introduction

Large-scale stochastic optimization has emerged as an indispensable tool in machine learning, particularly in the training of large foundation models (Brown et al., 2020; OpenAI et al., 2023). Solving such large and intricate problems poses formidable challenges, often requiring days or months to complete. Consequently, it is imperative to expedite large-scale stochastic optimization through distributed methods. The appeal of distributed stochastic optimization lies in its potential to harness the combined power of distributed computing nodes to handle the size of modern datasets. In this paper, we explore a server-based distributed architecture, where the nodes communicate with a server that coordinates their activities and manages the distribution and aggregation of computational tasks.

Nevertheless, the promise of distributed stochastic optimization is underpinned by the assumption of a trustworthy system, in which all nodes adhere to the prescribed computational protocol. The introduction of Byzantine faults/attacks to a fraction of the nodes, i.e., arbitrary deviations from expected behaviors, potentially due to node malfunctions (Zhang et al., 2020; Xiao et al., 2024), malicious manipulations (Attias et al., 2022; Liu et al., 2024), or poisoned data (Mahloujifar et al., 2019; Lewis et al., 2023), poses a significant challenge to distributed stochastic optimization methods and leads to incorrect solutions or even total failures. The complexity of defending against Byzantine attacks is further compounded in scenarios involving heterogeneous data, where the nodes may possess non-identically distributed data samples such that differentiating Byzantine attacks and honest behaviors becomes highly nontrivial (Li et al., 2019; Karimireddy et al., 2022). This paper is devoted to investigating the optimal complexity in Byzantine-robust distributed stochastic optimization with data heterogeneity.

The basic concept of Byzantine robustness originates from the seminal work of (Lamport et al., 1982), aiming at achieving consensus in a distributed system where some nodes may act maliciously or fail arbitrarily. It is then extended to the area of distributed deterministic optimization (Su and Vaidya, 2016; Chen et al., 2017). In recent years, Byzantine robustness in distributed stochastic optimization has attracted immense research interest due to the popularity of large-scale machine learning (Guerraoui et al., 2024; Ye and Ling, 2025). The pursuit of Byzantine-robust methods has led to the development of diverse strategies aimed at fortifying distributed stochastic optimization against the attacks from Byzantine nodes. The majority of these strategies rely on robust aggregators, with which the server either removes suspicious stochastic gradients prior to averaging (Chen et al., 2018; Alistarh et al., 2018; Xie et al., 2019) or uses statistically robust estimators such as trimmed mean (Yin et al., 2018), median (Yin et al., 2018), geometric median (Wu et al., 2020), to name a few.

In scenarios with heterogeneous data, the performance of the aforementioned methods shall be significantly degraded. When data heterogeneity appears among the nodes, their local stochastic gradients exhibit varying statistical properties, diminishing the effectiveness of the robust aggregators that utilize statistical similarity to distinguish the Byzantine nodes from the rest honest nodes and leading to unavoidable convergence errors (Li et al., 2019; Wu et al., 2020; El-Mhamdi et al., 2021; Karimireddy et al., 2022; Guerraoui et al., 2024; Peng et al., 2025). In light of this issue, advanced robust aggregators that are relatively insensitive to data heterogeneity, such as bucketing (Karimireddy et al., 2022) and nearest neighbor mixing (Allouah et al., 2023), has been proposed.

While the existing methods enjoy theoretical guarantees and/or empirical successes, the performance limits of Byzantine-robust distributed stochastic optimization methods have not been fully clarified. This paper aims to reveal the performance limits by establishing the optimal complexity in Byzantine-robust distributed stochastic optimization. We focus on first-order, synchronous methods; extensions to zeroth-order (Egger et al., 2025), second-order (Cao and Lai, 2020; Ghosh et al., 2020; Koushkbaghi et al., 2024) and asynchronous methods (El-Mhamdi et al., 2021; Yang and Li, 2023) will be our future works.

1.1 Problem setup

We consider a distributed system comprising a server and n nodes. The sets of the honest and Byzantine nodes are denoted by $\mathcal{H}$ and $\mathcal{B}$, respectively. Note that the identities of the honest and Byzantine nodes are unknown to the server. We assume $|\mathcal{B}| < |\mathcal{H}|$ throughout this paper. The purpose of Byzantine-robust distributed stochastic optimization is to find a minimizer to

$$\min_{x\in\mathbb{R}^d} \; f(x) = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} f_i(x), \quad \text{where} \quad f_i(x) = \mathbb{E}_{\xi\sim\mathcal{D}_i}[F(x,\xi)]. \tag{1}$$

Here, ξ represents a random variable following the local data distribution $\mathcal{D}_i$ of node i, and $F : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}$ is a Borel measurable function. Each function f_i is accessible locally by node i and is assumed to be smooth. It is important to note that data heterogeneity typically exists; that is, the local data distributions $\{\mathcal{D}_i\}_{i\in\mathcal{H}}$ differ across the honest nodes.

1.2 Fundamental open questions

The convergence error of a Byzantine-robust distributed stochastic optimization method, after taking k oracle queries of the stochastic gradients, typically comprises two components: Byzantine error and optimization error. Specifically:

$$\textbf{Convergence error} = \textbf{Byzantine error} + \textbf{Optimization error } \boldsymbol{\epsilon}. \tag{2}$$

The Byzantine error is non-vanishing; it persists throughout the entire optimization process, even as the number of oracle queries k approaches infinity. Conversely, the optimization error ϵ typically decreases with the number of oracle queries k. The interplay between these two error components characterizes the overall performance of Byzantine-robust distributed stochastic optimization methods, with the goal of simultaneously minimizing both errors to achieve certified Byzantine robustness and fast convergence rate. This dual objective presents a fundamental challenge in the design and analysis of Byzantine-robust distributed stochastic optimization methods. Tackling this challenge requires answering the following two fundamental questions:

Q1. *What is the smallest Byzantine error that any Byzantine-robust distributed stochastic optimization methods can achieve?*

Q2. *What is the optimal convergence rate at which the optimization error ϵ decreases to zero for any Byzantine-robust distributed stochastic optimization methods, or equivalently, what is the minimum number of queries to a stochastic gradient oracle required to attain an arbitrarily small ϵ?*

In this paper, we answer these two open questions via establishing tight lower bounds of the Byzantine error and the oracle query complexity. Note that several pioneering works have already shed light on these two open questions. The work of (Alistarh et al., 2018) provides valuable insights into the tight lower bound of the oracle query complexity in strongly convex optimization. However, the analysis is confined to homogeneous data distribution and does not account for the lower bound of the Byzantine error. On the other hand, the work of (Karimireddy et al., 2022) addresses the tight lower bound of the Byzantine error in non-convex optimization and heterogeneous data distribution, but does not explore the oracle query complexity. The recent work of (Farhadkhani et al., 2024) analyzes the tight lower bounds of the Byzantine error and the oracle query complexity in heterogeneous data distribution, but only under the Polyak-Łojasiewicz condition.

1.3 Main results and contributions

In this paper, we provide a comprehensive analysis that establishes the lower bounds of the Byzantine error and the oracle query complexity in Byzantine-robust distributed stochastic optimization. We also validate the tightness of these lower bounds through developing methods that can attain optimal Byzantine robustness and optimal convergence rate simultaneously. In particular, our contributions are:

- We establish the lower bounds on the Byzantine error for Byzantine-robust distributed methods in both strongly convex and non-convex stochastic optimization.

- We establish the lower bounds on the convergence rate at which the optimization error ϵ approaches zero for Byzantine-robust distributed methods in both strongly convex and non-convex stochastic optimization. Leveraging these results, we reveal the lower bounds on the minimum number of queries to a stochastic gradient oracle required to achieve an arbitrarily small optimization error ϵ.

- We identify significant discrepancies between our established lower bounds and the Byzantine robustness and convergence rates reported in the existing works. To fill this gap, we propose novel Byzantine-robust distributed stochastic optimization methods that provably match these lower bounds, up to at most logarithmic factors. This fact implies that our established lower bounds are tight, and our proposed methods attain the optimal Byzantine robustness and the optimal convergence rates simultaneously.

The bounds established in this paper, along with those from the existing state-of-the-art Byzantine-robust distributed stochastic optimization methods, are summarized in Tables 1 and 2. From the lower bound perspective, our work simultaneously explores the Byzantine error and the oracle query complexity. While previous studies either address only one of these aspects (Alistarh et al., 2018; Karimireddy et al., 2021) or consider both but under the Polyak-Łojasiewicz condition (Farhadkhani et al., 2024), our results apply to more general strongly convex and non-convex functions. From the upper bound perspective, our proposed methods match the lower bounds, achieving superior Byzantine robustness while demonstrating theoretically faster convergence rates.

Table 1: Lower and upper bounds of finding x such that $\mathbb{E}[\|\nabla f(x)\|]$ is no larger than the Byzantine error plus the optimization error ϵ in strongly convex stochastic optimization. Notations: n is the number of nodes; $\delta \in [0, \frac{1}{2})$ is the estimated fraction of Byzantine nodes that is no smaller than the true fraction of Byzantine nodes; σ^2 bounds the variance of the stochastic gradient estimates (see Assumption 3), with $\sigma^2 = 0$ corresponding to deterministic optimization; ζ^2 bounds the local gradient dissimilarity between the nodes (see Assumption 2), with $\zeta^2 = 0$ corresponding to homogeneous data distribution; $\rho \geq 0$ is the coefficient to measure the robustness of an aggregator (see Definition 2); $R = \|x^0 - \arg\min_x f(x)\|$ where x^0 stands for the initial variable; L is the Lipschitz smoothness constant; μ is the strongly convex constant; $\kappa := \frac{L}{\mu}$ is the condition number; $\tilde{\Omega}(\cdot)$ and $\tilde{O}(\cdot)$ hide constants and logarithmic factors.

	Byzantine error	Oracle query complexity	Reference
Lower bound	/	$\Omega\left(\frac{\delta^2\sigma^2}{\epsilon^2} + \frac{\sigma^2}{n\epsilon^2}\right)$	Alistarh et al. (2018)
	$\Omega\left(\delta^{1/2}\zeta\right)$	$\tilde{\Omega}\left(\frac{\delta\sigma^2}{\epsilon^2} + \frac{\sigma^2}{n\epsilon^2} + \kappa\right)$	Farhadkhani et al. (2024)$^\circ$
	$\Omega\left(\rho^{1/2}\delta^{1/2}\zeta\right)$	$\tilde{\Omega}\left(\frac{\rho\delta\sigma^2}{\epsilon^2} + \frac{\sigma^2}{(1-\delta)n\epsilon^2} + \kappa^{1/2}\right)$	Thm. 11
Upper bound	/	$\tilde{O}\left(\kappa + \frac{\kappa\delta^2\sigma^2}{\epsilon^2} + \frac{\kappa\sigma^2}{n\epsilon^2}\right)$	Alistarh et al. (2018)†
	$O\left(L\kappa\delta n\zeta\right)$	$O\left(\frac{LR^2}{\epsilon^2} + \frac{\kappa}{\mu}\frac{(\delta^2+(1-\delta)^2)n^2\zeta^2+(1-\delta)n\sigma^2}{\epsilon^2}\right)$	Li et al. (2019)†
	$O\left(\kappa^2\zeta\right)$	$\tilde{O}\left(\kappa + \frac{\kappa^5(1+\delta)\sigma^2}{(1-\delta)n\epsilon^2} + \frac{\kappa^5\delta\sigma^2}{\epsilon^2}\right)$	Data and Diggavi (2021)†
	$O\left(\frac{\zeta}{1-2\delta}\right)$	$\tilde{O}\left(\kappa + \frac{\kappa L\sigma^2}{(1-2\delta)^2\epsilon^2}\right)$	Pillutla et al. (2022)†
	$O\left(\kappa^{1/2}\rho^{1/2}\delta^{1/2}\zeta\right)$	$\tilde{O}\left(\frac{\kappa^{3/2}\rho\delta\sigma^2}{\epsilon^2} + \frac{\kappa^{3/2}\sigma^2}{(1-\delta)n\epsilon^2} + \kappa^{1/2}\right)$	Algorithm 2 (Thm. 15)

† The bounds are established for specific robust aggregators.
∘ The bounds are established under the Polyak-Łojasiewicz condition. We translate it to strongly convex optimization, and change the measure from function value to gradient norm. The oracle query complexity involves κ rather than $\kappa^{1/2}$ due to the Polyak-Łojasiewicz condition. Observe that the robust aggregator constant ρ does not appear in the bounds.

1.4 Related works

Lower bounds for Byzantine-free single-node optimization. For deterministic problems, the lower bounds on the iteration complexity of strongly convex and convex optimization are established and proved to be tight in the works of (Nemirovski and Yudin, 1983; Nesterov, 2003). That of non-convex optimization is established in (Carmon et al., 2020, 2021). For convex stochastic problems with the finite-sum and expectation-minimization structures, the tight lower bounds are derived in (Woodworth and Srebro, 2016) and (Foster et al., 2019), respectively. For non-convex stochastic problems, the works of (Fang et al., 2018) and (Li et al., 2021) investigate the tight lower bound in the finite-sum structure; the work of (Arjevani et al., 2023) considers that in the expectation-minimization structure.

Table 2: Lower and upper bounds of finding x such that $\mathbb{E}[\|\nabla f(x)\|]$ is no larger than the Byzantine error plus the optimization error ϵ in non-convex stochastic optimization. Notations not appeared in Table 1: m is the batch size; $\Delta := f(x^0) - f^*$ in which f^* stands for the minimum value of (1); $c_1 = \frac{\rho\delta\sigma^4}{(1-\delta)n\epsilon^4} + \frac{\sigma^2}{(1-\delta)n\epsilon^2}$; $c_2 = \frac{(1-\delta)^{1/3}L^{1/3}\Delta^{1/3}\sigma^{2/3}}{(1+(1-\delta)\rho\delta n)^{1/3}n^{1/3}\epsilon^{4/3}}$.

	Byzantine error	Oracle query complexity	References
Lower bound	$\Omega\left(\delta^{1/2}\zeta\right)$	/	Karimireddy et al. (2022)
	$\Omega\left(\rho^{1/2}\delta^{1/2}\zeta\right)$	$\Omega\left(\frac{L\Delta\rho\delta\sigma^2}{\epsilon^4} + \frac{L\Delta\sigma^2}{(1-\delta)n\epsilon^4} + \frac{L\Delta}{\epsilon^2}\right)$	Thm. 11
Upper bound	$O\left((1+\delta)^{1/2}\zeta\right)$	$O\left(\frac{L^2R^2}{\epsilon^2}\left(1 + \frac{(1+\delta)\sigma^2}{(1-\delta)n\epsilon^2} + \frac{\delta\sigma^2}{\epsilon^2}\right)\right)$	Data and Diggavi (2021)†
	$O\left(\rho^{1/2}\delta^{1/2}\zeta\right)$	$O\left(\frac{L\Delta\rho\delta\sigma^2}{\epsilon^4} + \frac{L\Delta\sigma^2}{(1-\delta)n\epsilon^4} + \frac{L\Delta}{\epsilon^2} + c_1\right)$	Karimireddy et al. (2022)
	$O\left(\rho^{1/2}\delta^{1/2}\zeta\right)$	$O\left(\frac{L\Delta\rho\delta\sigma^2}{\epsilon^4} + \frac{L\Delta\sigma^2}{(1-\delta)n\epsilon^4} + \frac{L\Delta}{\epsilon^2} + c_1\right)$	Algorithm 1 (Cor. 17)
	$O\left(\rho^{1/2}\delta^{1/2}\zeta\right)$	$O\left(\frac{L\Delta\rho\delta\sigma^2}{\epsilon^4} + \frac{L\Delta\sigma^2}{(1-\delta)n\epsilon^4} + \frac{L\Delta}{\epsilon^2} + c_2\right)$	Allouah et al. (2023)
	$O\left(\rho^{1/2}\delta^{1/2}\zeta\right)$	$\tilde{O}\left(\frac{L\Delta\rho\delta\sigma^2}{\epsilon^4} + \frac{L\Delta\sigma^2}{(1-\delta)n\epsilon^4} + \frac{L\Delta}{\epsilon^2}\right)$	Algorithm 3 (Thm. 18)

† The bound is established for a specific robust aggregator.

Lower bounds for Byzantine-free distributed optimization. The lower bounds on the iteration complexity of distributed strongly convex deterministic optimization is established in (Scaman et al., 2017), in which the network can be both server-based and server-less. A distributed dual accelerated method is proposed to achieve these lower bounds. For distributed server-less, non-convex, stochastic optimization, the optimal oracle query and communication complexities are obtained in (Lu and De Sa, 2021) given that the communication graphs are linear. These findings are extended to general graphs in (Yuan et al., 2022). The communication complexity of distributed server-based methods with communication compression is investigated in (Huang et al., 2022).

Lower bounds for Byzantine-robust distributed stochastic optimization. For the convex problems with homogeneous data distributions, the optimal oracle query complexity is established in (Alistarh et al., 2018). When the data distributions are heterogeneous, the non-vanishing Byzantine error emerges (Karimireddy et al., 2022). While the work of (Farhadkhani et al., 2024) also considers both the Byzantine error and the oracle query complexity, their analysis relies the Polyak-Łojasiewicz condition. In contrast, our work establishes tight lower bounds in both aspects, for general strongly convex and non-convex functions. The lower bound of the statistical learning rate for Byzantine-robust distributed stochastic mean estimation is investigated in (Yin et al., 2018). Two Byzantine-robust methods based on the trimmed mean and coordinate-wise median aggregators are proposed to achieve the order-optimal statistical learning rate. The impact of the dimensionality on the statistic learning rate is taken into account in (Zhu et al., 2023).

1.5 Organization

The rest of this paper is organized as follows. Section 2 introduces Byzantine-robust distributed stochastic optimization, including the function, stochastic gradient oracle, robust aggregator, and method classes that are necessary for the ensuing analysis. Section 3 states the lower bounds of the Byzantine error and the oracle query complexity for strongly convex and non-convex problems. Section 4 proposes novel methods to attain the established lower bounds, validating their tightness. Numerical experiments are conducted in Section 5. Section 6 summarizes this work. For clarity, we leave the proofs of the main results to Appendix A.

2. Byzantine-robust distributed stochastic optimization

This section specifies the notations, assumptions, and problem setup under which we study the optimal complexity for solving the Byzantine-robust distributed stochastic optimization problem in the form of (1).

2.1 Notation

Throughout this paper, we use $\mathbb{E}_{\xi\sim\mathcal{D}}$ to denote the expectation over ξ, which is a random variable following the distribution $\mathcal{D}$, and we refer to it as $\mathbb{E}_\xi$ or $\mathbb{E}$ if there is no confusion. We use t and k to denote the numbers of iterations and oracle queries, respectively. Accordingly, T and K represent the overall numbers of iterations and oracle queries, respectively. The Euclidean norm is denoted by $\|\cdot\|$. We use the big-O notations to describe complexity, with $O(\cdot)$ and $\Omega(\cdot)$ hiding constants while $\tilde{O}(\cdot)$ hiding both constants and logarithmic factors.

2.2 Function class $\mathcal{F}$

We let the function class $\mathcal{F}_{L,\zeta^2}$, abbreviated as $\mathcal{F}$, denote the set of all functions f satisfying Assumptions 1 and 2 for any underlying dimension $d \in \mathbb{N}_+$.

Assumption 1 *The function $f(x)$ is continuously differentiable. The functions $\{f_i(x)\}_{i\in\mathcal{H}}$ are lower bounded. In addition, the functions $\{f_i(x)\}_{i\in\mathcal{H}}$ are L-smooth,* i.e.*, there exists a constant $L > 0$ such that*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \le L\|x - y\|$$

for all $i \in \mathcal{H}$ and $x, y \in \mathbb{R}^d$.

Assumption 2 *The gradients $\{\nabla f_i(x)\}_{i\in\mathcal{H}}$ satisfy*

$$\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\nabla f_i(x) - \nabla f(x)\|^2 \le \zeta^2$$

for some $\zeta^2 \ge 0$, where $\nabla f(x) = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\nabla f_i(x)$ according to (1).

Assumption 1 is very common. Assumption 2 is widely used in distributed optimization to restrict the data heterogeneity (Lian et al., 2017; Reddi et al., 2021; Karimireddy et al., 2022; Allouah et al., 2023).

In the ensuing analysis, we shall examine the complexity bounds when $\{f_i(x)\}_{i\in\mathcal{H}}$ are either μ-strongly convex or non-convex. Below we give the definition of μ-strong convexity.

Definition 1 *The functions $\{f_i(x)\}_{i\in\mathcal{H}}$ are μ-strongly convex, if there exists a constant $\mu > 0$ such that*

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

for all $i \in \mathcal{H}$ and $x, y \in \mathbb{R}^d$.

If a function is both L-smooth and μ-strongly convex, then $\mu \leq L$.

2.3 Stochastic gradient oracle class $\mathcal{O}$

We assume that at each iteration t, each node $i \in \mathcal{H}$ can obtain its local stochastic gradient $\nabla F(x, \xi_i^t)$ through an oracle O, *i.e.*, $\nabla F(x, \xi_i^t) = \mathsf{O}(F, x, \xi_i^t)$. We let $\mathcal{O}_{\sigma^2}$, abbreviated as $\mathcal{O}$, denote the set of all oracles that satisfy the following assumption.

Assumption 3 *The function $F(x, \xi)$ is continuously differentiable with respect to x, and the stochastic gradient $\nabla F(x, \xi_i^t) = \mathsf{O}(F, x, \xi_i^t)$ obtained by node $i \in \mathcal{H}$ through the oracle $\mathsf{O} \in \mathcal{O}$ satisfies the following conditions:*

- *The random variable ξ_i^t is independently drawn across all nodes $i \in \mathcal{H}$ and all iterations $t \in \mathbb{N}$.*
- *The stochastic gradient is an unbiased estimator of the true gradient, i.e.,*

$$\mathbb{E}_{\xi_i^t}[\nabla F(x, \xi_i^t)] = \nabla f_i(x), \quad \forall i \in \mathcal{H},\ t \in \mathbb{N}.$$

- *The variance of the stochastic gradient is bounded, i.e.,*

$$\mathbb{E}_{\xi_i^t}[\|\nabla F(x, \xi_i^t) - \nabla f_i(x)\|^2] \leq \sigma^2, \quad \forall i \in \mathcal{H},\ t \in \mathbb{N}$$

  for some constant $\sigma^2 \geq 0$.

In distributed stochastic optimization, independent sampling across different nodes and iterations is common. Besides, the unbiasedness and the bounded variance of the stochastic gradient are widely used assumptions in stochastic optimization (Bottou et al., 2018). In the ensuing analysis, we will denote all honest nodes computing their stochastic gradients once as one oracle query.

2.4 Robust aggregator class $\mathcal{A}$

Robust aggregators are essential for mitigating the impact of the Byzantine nodes, which inject malicious updates in distributed stochastic optimization. A number of effective robust aggregators, such as Krum, Median, Trimmed Mean, and others, have been proposed in the literature with theoretical guarantees and empirical successes. Nevertheless, the theoretical limits of Byzantine-robust distributed stochastic optimization methods with these robust aggregators remain unknown. Investigating each individual robust aggregator would require an impractical amount of effort. For this reason, in this paper we do not study the optimal

Table 3: Comparison between different $(\delta_{\max}, \rho)$-robust aggregators.

	$\rho\delta$	References
Krum	$6 + \frac{6\delta}{1-2\delta}$	Blanchard et al. (2017)
Median (Med)	$4\left(1 + \frac{\delta}{1-2\delta}\right)^2$	Yin et al. (2018)
Trimmed Mean (TM)	$\frac{6\delta}{1-2\delta}\left(1 + \frac{\delta}{1-2\delta}\right)$	Yin et al. (2018)
FABA	$\frac{2\delta\lvert\mathcal{H}\rvert}{1-3\delta}$	Xia et al. (2019)
Geometric Median (GM)	$4\left(1 + \frac{\delta}{1-2\delta}\right)^2$	Wu et al. (2020)
Center Clipping (CC)	$18\sqrt{2}\delta\sqrt{\lvert\mathcal{H}\rvert}$	Karimireddy et al. (2021)
Lower bound	$\frac{\delta}{1-2\delta}$	Allouah et al. (2023)

† The robustness coefficients ρ of Krum, Med, TM, and GM, and the lower bound are established in (Allouah et al., 2023). That of FABA comes from (Peng et al., 2025). That of CC is given in (Shi et al., 2025).

complexity with a specific robust aggregator, but instead will focus on a class of $(\delta_{\max}, \rho)$-robust aggregators $\mathcal{A}$ (Allouah et al., 2023; Farhadkhani et al., 2022; Karimireddy et al., 2022) defined as follows.

Definition 2 ($(\delta_{\max}, \rho)$-robust aggregator) *Consider n inputs $\{w_i\}_{i=1}^n$ from all n nodes, $|\mathcal{H}|$ of them being from the honest nodes in $\mathcal{H}$ and the number of honest nodes satisfying $|\mathcal{H}| \geq (1-\delta)n$ with $0 < \delta \leq \delta_{\max} < 0.5$. Define $\bar{w} = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} w_i$. An aggregator $\mathsf{A} \in \mathcal{A}$ is called $(\delta_{\max}, \rho)$-robust if there exists a constant $\rho \geq 0$ such that the output $w = \mathsf{A}(\{w_i\}_{i=1}^n)$ satisfies*

$$\|w - \bar{w}\|^2 \leq \frac{\rho\delta}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \|w_i - \bar{w}\|^2. \tag{3}$$

Using a $(\delta_{\max}, \rho)$-robust aggregator, the deviation of the robust average w from the true average $\bar{w}$ is able to be bounded by the variation of the honest inputs $\{w_i\}_{i\in\mathcal{H}}$. These robust aggregators effectively mitigate the impact of the Byzantine nodes, preventing output divergence. In particular, a robust aggregator will recover the exact average if the honest inputs $\{w_i\}_{i\in\mathcal{H}}$ are equal and in the majority. Furthermore, it is important to note that $\delta_{\max}$ denotes the maximum fraction of the Byzantine nodes tolerated by the robust aggregator, while δ serves as a form of prior knowledge about the problem, representing the estimated fraction of the Byzantine nodes in the distributed network and being no smaller than the true fraction of the Byzantine nodes. In the analysis, we generally assume that $\delta > 0$; otherwise, we can simply use the mean aggregator to obtain $\bar{w}$, hence resulting in a trivial outcome. We also require $\delta_{\max} < 0.5$, meaning that the Byzantine nodes are not dominant. Last but not least, the robustness coefficient ρ plays a key role in characterizing the effectiveness of a robust aggregator. For most robust aggregators, ρ is a function of δ is dependent on the priori knowledge of δ. Table 3 lists the robustness coefficients ρ of various $(\delta_{\max}, \rho)$-robust aggregators.

2.5 Method class $\mathcal{M}$

In this paper, we investigate a class of server-based methods to solve the Byzantine-robust distributed stochastic optimization problem in the form of (1). With an initial variable x^0, these methods proceed with three phases.

- *Local computation.* Upon receiving the variable x^t transmitted by the server, each honest node $i \in \mathcal{H}$ computes a batch of m vectors as

  $$\mathbf{x}_i^{(t)} = \mathbf{x}^{(t)} := \left(x^{(t,1)}, \cdots, x^{(t,m)}\right) \in \mathbb{R}^{d\times m} \text{ with } x^{(t,l)} \in \operatorname{span}(x^0, \cdots, x^t) \subseteq \mathbb{R}^d,\ \forall l \in [m].$$

  Note that $\{\mathbf{x}_i^{(t)}\}_{i\in\mathcal{H}}$ are identical across all honest nodes. Given $\mathbf{x}_i^{(t)}$, each honest node $i \in \mathcal{H}$ samples m independent random variables $\{\xi_i^{(t,l)}\}_{l=1}^m$ with each $\xi_i^{(t,l)} \sim \mathcal{D}_i$, and queries a batch of m stochastic gradients from the oracle $\mathsf{O} \in \mathcal{O}$ as

  $$\left(\nabla F(x_i^{(t,1)}, \xi_i^{(t,1)}), \cdots, \nabla F(x_i^{(t,m)}, \xi_i^{(t,m)})\right) \in \mathbb{R}^{d\times m}.$$

  With the above stochastic gradients, each honest node $i \in \mathcal{H}$ computes a gradient estimator

  $$w_i^t \in \operatorname{span}\left(\left\{\nabla F\left(x_i^{(j,l)}, \xi_i^{(j,l)}\right) : j = 0, \cdots, t;\ l = 1, \cdots, m\right\}\right) \subseteq \mathbb{R}^d.$$

  In fact, the gradient estimator w_i^t can be regarded as a linear combination of the historical stochastic gradients, in the form of

  $$w_i^t = \sum_{j=1}^{t}\sum_{l=1}^{m} \alpha^{(j,l)} \nabla F(x_i^{(j,l)}, \xi_i^{(j,l)}) \in \mathbb{R}^d, \tag{4}$$

  in which $\alpha^{(j,l)} \in \mathbb{R}$ stands for the coefficient associated with $\nabla F(x_i^{(j,l)}, \xi_i^{(j,l)})$. Besides, we assume $\sum_{j=1}^{t}\sum_{l=1}^{m} \alpha^{(j,l)} \geq \alpha_{\min} > 0$ for any t and m. Such a gradient estimator w_i^t in (4) is highly versatile and reduces to various existing ones through selecting appropriate values for each $\alpha^{(j,l)}$. For instance, given that $\alpha^{(j,l)} = \frac{1}{m}$ if $j = t$ and $\alpha^{(j,l)} = 0$ otherwise, w_i^t reduces to the mini-batch stochastic gradient in the form of $\frac{1}{m}\sum_{l=1}^{m} \nabla F(x_i^{(t,l)}, \xi_i^{(t,l)})$. If we further assume $m = 1$, then w_i^t becomes the classical stochastic gradient $\nabla F(x_i^{(t,1)}, \xi_i^{(t,1)})$. Likewise, we can also recover the stochastic momentum (Polyak, 1964).

- *Communication.* Each honest node $i \in \mathcal{H}$ uploads its computed w_i^t to the server. Each Byzantine node $i \in \mathcal{B}$, however, may upload an arbitrary vector $w_i^t \in \mathbb{R}^d$.

- *Global variable update.* The server uses a $(\delta_{\max}, \rho)$-robust aggregator $\mathsf{A} \in \mathcal{A}$ to process the messages received from all nodes, yielding an aggregated gradient

  $$w^t = \mathsf{A}(w_1^t, \cdots, w_n^t).$$

  Note that, although for any $i \in \mathcal{H}$, w_i^t is restricted to a linear combination of past stochastic gradients, the aggregator A can be a nonlinear function, such as geometric

median. Subsequently, the server updates the variable x using all historical variables $x^0, \cdots, x^t$ and all historical aggregated gradients $w^0, \cdots, w^t$. Formally,

$$x^{t+1} \in \text{span}(x^0, \cdots, x^t, w^0, \cdots, w^t).$$

The server then transmits x^{t+1} to all nodes, initiating a new iteration.

Such a process is repeated. We denote the output after t iterations as $\hat{x}^t \in \text{span}(x^0, \cdots, x^t)$ for any $t \in \mathbb{N}$. In this paper, we study the set of methods that include the above processes, denoted as $\mathcal{M}$.

Remark 3 *With particular note, each honest node $i \in \mathcal{H}$ is also allowed to compute and upload multiple gradient estimators at each iteration, only bringing a constant to the overall complexity.*

3. Lower bound of Byzantine-robust distributed stochastic optimization

Having introduced the definitions of the function, stochastic gradient oracle, robust aggregator, as well as method classes, we are ready to formalize the concept of complexity. We will prove in Section 3.1 that if the data is heterogeneous, the gradient norm $\|\nabla f(\hat{x}^t)\|$ shall be always away from zero for Byzantine-robust distributed stochastic optimization – this rarely happens in analyzing the complexity lower bounds of Byzantine-free distributed stochastic optimization. We call this gap the **Byzantine error**. The Byzantine error refers to the residual that does not vanish regardless of the numbers of iterations and oracle queries, quantifying the robustness of a Byzantine-robust distributed stochastic optimization method. The Byzantine error generated by $\mathsf{M} \in \mathcal{M}$ depends on the function $f \in \mathcal{F}$, the stochastic gradient oracle $\mathsf{O} \in \mathcal{O}$, and the robust aggregator $\mathsf{A} \in \mathcal{A}$. If any of these three elements changes, the Byzantine error generated by M also varies. Therefore, we denote

$$\epsilon_{\text{bzt}}^{\mathsf{M}}(f, \mathsf{O}, \mathsf{A}) := \inf_{t \in \mathbb{N}} \left\{ \mathbb{E}[\|\nabla f(\hat{x}^t)\| \mid \mathsf{O}, \mathsf{A}] \right\}, \tag{5}$$

where $\hat{x}^t$ is the output of $\mathsf{M} \in \mathcal{M}$ after t iterations. The rest of the error is termed as the **optimization error**, which is vanishing and can be reduced to zero when increasing the number of iterations or oracle queries to infinity.

We define the oracle query complexity of the method class $\mathcal{M}$ on the function class $\mathcal{F}$, the stochastic gradient oracle class $\mathcal{O}$ and the robust aggregator class $\mathcal{A}$, to ensure that the optimization error does not exceed a given ϵ, as

$$\mathcal{K}_\epsilon(\mathcal{M}, \mathcal{A}, \mathcal{F}, \mathcal{O}) := \inf_{\mathsf{M} \in \mathcal{M}} \sup_{f, \mathsf{O}, \mathsf{A} \in \mathcal{F}, \mathcal{O}, \mathcal{A}} \inf \left\{ K \mid \mathbb{E}[\|\nabla f(\tilde{x}^K)\|] \leq \epsilon_{\text{bzt}}^{\mathsf{M}}(f, \mathsf{O}, \mathsf{A}) + \epsilon \right\}, \tag{6}$$

where $\tilde{x}^K$ is the output $\mathsf{M} \in \mathcal{M}$ after K oracle queries. Throughout this paper, given $f \in \mathcal{F}$, $\mathsf{O} \in \mathcal{O}$ and $\mathsf{A} \in \mathcal{A}$, we call $x \in \mathbb{R}^d$ an $(\epsilon_{\text{bzt}}^{\mathsf{M}}(f, \mathsf{O}, \mathsf{A}), \epsilon)$-*stationary point* if $\mathbb{E}[\|\nabla f(x)\|] \leq \epsilon_{\text{bzt}}^{\mathsf{M}}(f, \mathsf{O}, \mathsf{A}) + \epsilon$.

In this section, we are going to analyze the lower bounds of the Byzantine error in (5) and the oracle query complexity in (6). We begin with showing that there is a non-vanishing Byzantine error through an example involving Byzantine nodes and heterogeneous data

($\zeta^2 > 0$) in Section 3.1. Then, we proceed to analyze the factors influencing the oracle query complexity when the data is homogeneous ($\zeta^2 = 0$). To be specific, Section 3.2 considers $\rho = \sigma^2 = 0$ to focus on the impact of the function class $\mathcal{F}$, Section 3.3 sets $\rho = 0$ but $\sigma^2 > 0$ to highlight the impact of the stochastic gradient oracle class $\mathcal{O}$, while Section 3.4 lets $\rho > 0$ and $\sigma^2 > 0$ so as to explore the impact of the robust aggregator class $\mathcal{A}$. Finally, summing up these results in Section 3.5 yields a lower bound, whose tightness will be proved in Section 4.

3.1 Lower bound of Byzantine error

We start by analyzing the lower bound of the Byzantine error caused by data heterogeneity ($\zeta^2 > 0$), in the presence of the Byzantine nodes. We define this lower bound as

$$\epsilon_{\text{bzt}} := \inf_{\mathsf{M}\in\mathcal{M}} \sup_{f,\mathsf{O},\mathsf{A}\in\mathcal{F},\mathcal{O},\mathcal{A}} \epsilon_{\text{bzt}}^{\mathsf{M}}(f,\mathsf{O},\mathsf{A}). \tag{7}$$

The main idea of the analysis is to construct two problems with different objectives $f_1 = \frac{1}{|\mathcal{H}_1|}\sum_{i\in\mathcal{H}_1} f_{1,i}$ and $f_2 = \frac{1}{|\mathcal{H}_2|}\sum_{i\in\mathcal{H}_2} f_{2,i}$ yielding different minima, such that there exists a $(\delta_{\max},\rho)$-robust aggregator that yields the same result. Formally speaking, at any iteration t, any method $\mathsf{M}\in\mathcal{M}$, due to the same result w^t from such a $(\delta_{\max},\rho)$-robust aggregator $\mathsf{A}\in\mathcal{A}$, is going to return the same iterate x^{t+1}. Therefore, any method $\mathsf{M}\in\mathcal{M}$ must inherently incur an error on at least one of the two problems. We emphasize that the error is due to the data heterogeneity (with which we are able to construct two problems having different objectives and different minima) and the Byzantine nodes (with which we must use a robust aggregator that may yield the same result).

Lemma 4 *Given $\zeta^2 > 0$ and $\delta \in [0,\delta_{\max}]$, there exist a distributed problem in the form of (1) having at least $(1-\delta)n$ honest nodes with function $f\in\mathcal{F}$, and a $(\delta_{\max},\rho)$-robust aggregator $\mathsf{A}\in\mathcal{A}$, such that for any method $\mathsf{M}\in\mathcal{M}$, the Byzantine error is lower-bounded by*

$$\epsilon_{\text{bzt}} = \inf_{\mathsf{M}\in\mathcal{M}} \|\nabla f(\tilde{x})\| = \Omega(\rho^{1/2}\delta^{1/2}\zeta),$$

where $\tilde{x}$ is the output of M, irrelevant with the number of iterations and the number of oracle queries.

Proof See Appendix A.1. ■

Lemma 4 indicates that achieving the exact minimum of (1) is unattainable when the Byzantine nodes are present and the data is heterogeneous, consistent with the findings reported in (Karimireddy et al., 2022). The major difference between our work and (Karimireddy et al., 2022) lies in that the latter defines the lower bound of the Byzantine error as $\inf_{\mathsf{M},\mathsf{A}'\in\mathcal{M},\mathcal{A}'}\sup_{f,\mathsf{O}\in\mathcal{F},\mathcal{O}}\epsilon_{\text{bzt}}^{\mathsf{M},\mathsf{A}'}(f,\mathsf{O})$. Therein, $\mathcal{A}'$ represents the set of identity-independent robust aggregators whose outputs are independent on the identities of nodes and $\epsilon_{\text{bzt}}^{\mathsf{M},\mathsf{A}'}(f,\mathsf{O})$ $:= \inf_{t\in\mathbb{N}}\{\mathbb{E}[\|\nabla f(\hat{x}^t)\| \mid \mathsf{O}]\}$. Therefore, the lower bound $\Omega(\delta^{1/2}\zeta)$ established in (Karimireddy et al., 2022) only shows the impacts of the estimated fraction of Byzantine nodes δ and the data heterogeneity ζ^2, while our result also reveals how the robustness coefficient

of robust aggregator ρ affects the lower bound. Note that both results are irrelevant to the stochastic gradient variance σ^2. In fact, these two lower bounds are tight in their corresponding setups (see Section 4 for the tightness of our lower bounds), as the influence of σ^2 can be eliminated through proper variance reduction techniques.

3.2 Lower bound of oracle query complexity: Function

In this subsection, we investigate the lower bound of the oracle query complexity influenced by the function $f \in \mathcal{F}$. To this end, we consider the simplest case $\zeta^2 = \rho = \sigma^2 = 0$ so as to focus on the impact of $\mathcal{F}$. Observe that since we assume $\rho = 0$, the robust aggregator A is ideal and averages the inputs of the honest nodes. Besides, due to $\zeta^2 = \sigma^2 = 0$, the behaviors of the honest nodes are exactly the same. Therefore, this case reduces to single-node deterministic optimization and the classical lower bounds are applicable. For strongly convex functions, according to (Nesterov, 2003), we have the following lemma.

Lemma 5 *Given $\zeta^2 = \rho = \sigma^2 = 0$ and $\delta \in [0, \delta_{\max}]$, there exists a distributed problem in the form of (1) having at least $(1-\delta)n$ honest nodes with function $f \in \mathcal{F}$ and μ-strongly convex $\{f_i(x)\}_{i\in\mathcal{H}}$, such that for any method $\mathsf{M} \in \mathcal{M}$, to achieve a $(0,\epsilon)$-stationary point, the oracle query complexity is at least*

$$K = \Omega\left(\sqrt{\kappa}\log\frac{\mu R}{\epsilon}\right),$$

where $\kappa = \frac{L}{\mu}$ is the condition number and $R = \|x^0 - \arg\min_x f(x)\|$.

For non-convex functions, by (Carmon et al., 2021), we have the following lemma.

Lemma 6 *Given $\zeta^2 = \rho = \sigma^2 = 0$ and $\delta \in [0, \delta_{\max}]$, there exists a distributed problem in the form of (1) having at least $(1-\delta)n$ honest nodes with function $f \in \mathcal{F}$ and non-convex $\{f_i(x)\}_{i\in\mathcal{H}}$, such that for any method $\mathsf{M} \in \mathcal{M}$, to achieve a $(0,\epsilon)$-stationary point, the oracle query complexity is at least*

$$K = \Omega\left(\frac{L\Delta}{\epsilon^2}\right),$$

where $\Delta = f(x^0) - \inf_x f(x)$.

3.3 Lower bound of oracle query complexity: Stochastic gradient oracle

In this subsection, we investigate the lower bound of the oracle query complexity influenced by the stochastic gradient oracle. Now we maintain $\zeta^2 = \rho = 0$, but consider the case that $\mathsf{O} \in \mathcal{O}$ provides noisy stochastic gradients with variance $\sigma^2 > 0$. This case is essentially a distributed variant of single-node stochastic optimization, whose lower bounds have been analyzed in (Foster et al., 2019) for strongly convex functions and (Arjevani et al., 2023) for non-convex functions. The results are shown in the following lemmas.

Lemma 7 *Given $\zeta^2 = \rho = 0$, $\sigma^2 > 0$ and $\delta \in [0, \delta_{\max}]$, there exist a distributed problem in the form of (1) having at least $(1-\delta)n$ honest nodes with function $f \in \mathcal{F}$ and μ-strongly*

convex $\{f_i(x)\}_{i\in\mathcal{H}}$*, and a stochastic gradient oracle* $\mathsf{O}\in\mathcal{O}$*, such that for any method* $\mathsf{M}\in\mathcal{M}$*, to obtain a* $(0,\epsilon)$*-stationary point, the expected oracle query complexity is at least*

$$K=\Omega\left(\frac{\sigma^2}{(1-\delta)n\epsilon^2}\right).$$

Lemma 8 *Given* $\zeta^2=\rho=0$*,* $\sigma^2>0$ *and* $\delta\in[0,\delta_{\max}]$*, there exist a distributed problem in the form of* (1) *having at least* $(1-\delta)n$ *honest nodes with function* $f\in\mathcal{F}$ *and non-convex* $\{f_i(x)\}_{i\in\mathcal{H}}$*, and a stochastic gradient oracle* $\mathsf{O}\in\mathcal{O}$*, such that for any method* $\mathsf{M}\in\mathcal{M}$*, to obtain a* $(0,\epsilon)$*-stationary point, the expected oracle query complexity is at least*

$$K=\Omega\left(\frac{L\Delta\sigma^2}{(1-\delta)n\epsilon^4}\right).$$

The proofs are similar to those of single-node stochastic optimization, but each iteration involves a mini-batch of $(1-\delta)n$ stochastic gradients other than one, such that the variance is accordingly reduced. Thus, we omit the proofs.

3.4 Lower bound of oracle query complexity: Robust aggregator

In this subsection, we analyze the lower bound of the oracle query complexity influenced by the robust aggregator. We maintain $\zeta^2=0$, but investigate the case that the output of $\mathsf{A}\in\mathcal{A}$ can be different from the average of the inputs from the honest nodes with $\rho>0$ and $\mathsf{O}\in\mathcal{O}$ provides noisy stochastic gradients with variance $\sigma^2>0$.

For strongly convex $\{f_i(x)\}_{i\in\mathcal{H}}$, we construct a one-dimensional problem to analyze the lower bound of the optimization error. In this problem, the gradient at the initial point x^0 is set to $\nabla f(x^0)=2\epsilon$. We will prove that, when the number of oracle queries is insufficient, there exists a robust aggregator $\mathsf{A}\in\mathcal{A}$ that always returns 0, causing any method $\mathsf{M}\in\mathcal{M}$ to be stuck at x^0. However, our goal is to find a point x such that $|\nabla f(x)|\le\epsilon$. Clearly, x^0 does not satisfy this condition since $\nabla f(x^0)=2\epsilon$. This implies that the number of oracle queries must be sufficient to escape from such an initial point.

Lemma 9 *Given* $\zeta^2=0$*,* $\rho>0$*,* $\sigma^2>0$*, and* $\delta\in[0,\delta_{\max}]$*, there exist a distributed problem in the form of* (1) *having at least* $(1-\delta)n$ *honest nodes with function* $f\in\mathcal{F}$ *and* μ*-strongly convex* $\{f_i(x)\}_{i\in\mathcal{H}}$*, a stochastic gradient oracle* $\mathsf{O}\in\mathcal{O}$*, and a robust aggregator* $\mathsf{A}\in\mathcal{A}$*, such that for any method* $\mathsf{M}\in\mathcal{M}$*, to obtain a* $(0,\epsilon)$*-stationary point, the expected oracle query complexity is at least*

$$K=\Omega\left(\frac{\rho\delta\sigma^2}{\epsilon^2}\right).$$

Proof See Appendix A.2. ■

Lemma 9 highlights the impact of a non-ideal robust aggregator A on the oracle query complexity. Different from Lemma 7 where the robust aggregator is ideal such that $\rho=0$, the term n disappears in Lemma 9, showing that introducing a robust aggregator to defend against the Byzantine nodes affects the benefit of cooperation. Instead, the parameters ρ

and δ that characterize the performance of the robust aggregator appear, suggesting that a class of high-quality robust aggregators are beneficial to the overall complexity.

The absence of n in Lemma 9 can be explained from the definition of robust aggregators. At the right-hand side of (3) in Definition 2, the term $\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|w_i-\bar{w}\|^2$ represents the variation of the honest nodes' gradient estimators. Such a variation cannot be reduced by increasing the number of nodes n, leading to the oracle query complexity in Lemma 9.

For $\zeta^2=0$, another lower bound of the oracle query complexity has been established in (Alistarh et al., 2018). However, the definition of the lower bound in (Alistarh et al., 2018) is different from ours, but instead in the form of

$$\inf_{\mathsf{M},\mathsf{A}'\in\mathcal{M},\mathcal{A}'}\sup_{f,\mathsf{O}\in\mathcal{F},\mathcal{O}}\inf\left\{K\mid\mathbb{E}[\|\nabla f(\tilde{x}^K)\|]\leq\epsilon\right\},$$

in which $\mathcal{A}'$ denotes the set of identity-independent robust aggregators. That said, the work of (Alistarh et al., 2018) considers the best identity-independent robust aggregator while our work considers the worst robust aggregator satisfying Definition 2. Note that when $\zeta^2=0$, the Byzantine error reduces to zero and does not appear in the lower bound (see our Lemma 4). With the above definition and under the additional assumption of bounded stochastic gradients, the work of (Alistarh et al., 2018) establishes a lower bound of $\Omega\left(\frac{\delta^2\sigma^2}{\epsilon^2}\right)$ for the oracle query complexity. The differences in the definitions and assumptions result in the difference of the two lower bounds.

Now we turn to consider non-convex $\{f_i(x)\}_{i\in\mathcal{H}}$. We construct a high-dimensional problem with $d=\Omega(\epsilon^{-2})$, each node having the same non-convex function $f(x)$. This function, proposed by (Carmon et al., 2020), exhibits a chain-like structure such that any noiseless oracle query can only discover the index of the next coordinate. Besides, for every $x\in\mathbb{R}^d$ with $[x]_d=0$, $\|\nabla f(x)\|>\epsilon$. Thus, for deterministic non-convex optimization, to reach the ϵ accuracy, any method $\mathsf{M}\in\mathcal{M}$ initialized by $x^0=\mathbf{0}$ has to discover the d-th coordinate, which requires at least d oracle queries due to the chain-like structure of $f(x)$. This yields a lower bound of $\Omega(\epsilon^{-2})$ for oracle query complexity.

For non-convex stochastic optimization, a noisy oracle query is designed in (Arjevani et al., 2023) to amplify the lower bound. It discovers the next coordinate with a probability of $\Theta\left(\epsilon^2\sigma^{-2}\right)$, meaning that $\Omega\left(\sigma^2\epsilon^{-2}\right)$ oracle queries are required to discover the next coordinate in expectation. Hence, the total oracle query complexity is $\Omega\left(\sigma^2\epsilon^{-4}\right)$. Similarly, for Byzantine-robust non-convex stochastic optimization, we prove that there is a $(\delta_{\max},\rho)$-robust aggregator such that $\Omega(\rho\delta\sigma^2\epsilon^{-2})$ oracle queries are required to discover the next coordinate in expectation, leading to the total oracle query complexity of $\Omega(\rho\delta\sigma^2\epsilon^{-4})$ as stated in the following lemma.

Lemma 10 *Given $\zeta^2=0$, $\rho>0$, $\sigma^2>0$, and $\delta\in[0,\delta_{\max}]$, there exist a distributed problem in the form of (1) having at least $(1-\delta)n$ honest nodes with function $f\in\mathcal{F}$ and non-convex $\{f_i(x)\}_{i\in\mathcal{H}}$, a stochastic gradient oracle $\mathsf{O}\in\mathcal{O}$, and a robust aggregator $\mathsf{A}\in\mathcal{A}$, such that for any method $\mathsf{M}\in\mathcal{M}$, to obtain a $(0,\epsilon)$-stationary point, the expected oracle query complexity is at least*

$$K=\Omega\left(\frac{L\Delta\rho\delta\sigma^2}{\epsilon^4}\right),$$

where $\Delta=f(x^0)-\inf_x f(x)$.

Proof See Appendix A.3. ■

3.5 Final lower bound

Now, we sum up the four lower bounds on the Byzantine error and the oracle query complexity established above to yield the final complexity lower bound. The main results are given in Theorem 11.

Theorem 11 *Given $\zeta^2 \geq 0$, $\rho \geq 0$, $\sigma^2 \geq 0$, and $\delta \in [0, \delta_{\max}]$, there exist a distributed problem in the form of (1) having at least $(1-\delta)n$ honest nodes with function $f \in \mathcal{F}$ and μ-strongly convex $\{f_i(x)\}_{i\in\mathcal{H}}$, a stochastic gradient oracle $\mathsf{O} \in \mathcal{O}$ and a robust aggregator $\mathsf{A} \in \mathcal{A}$, such that for any method $\mathsf{M} \in \mathcal{M}$, to obtain an $(\epsilon_{\text{bzt}}, \epsilon)$-stationary point with $\epsilon_{\text{bzt}} = \Omega(\rho^{1/2}\delta^{1/2}\zeta)$, the expected gradient query complexity is at least*

$$K = \Omega\left(\frac{\rho\delta\sigma^2}{\epsilon^2} + \frac{\sigma^2}{(1-\delta)n\epsilon^2} + \kappa^{-1/2}\log\frac{\mu R}{\epsilon}\right). \tag{8}$$

For non-convex $\{f_i(x)\}_{i\in\mathcal{H}}$, the expected gradient query complexity is at least

$$K = \Omega\left(\frac{L\Delta\rho\delta\sigma^2}{\epsilon^4} + \frac{L\Delta\sigma^2}{(1-\delta)n\epsilon^4} + \frac{L\Delta}{\epsilon^2}\right). \tag{9}$$

4. Upper bound of Byzantine-robust distributed stochastic optimization

In this section, we will verify the tightness of the lower bound established in Section 3, via designing methods $\mathsf{M} \in \mathcal{M}$ with any robust aggregator $\mathsf{A} \in \mathcal{A}$ and any stochastic gradient oracle $\mathsf{O} \in \mathcal{O}$ to solve (1) with any function $f \in \mathcal{F}$ and to reach the lower bound. This is a nontrivial task and calls for elaborate integration of several fundamental tools. First, for strongly convex functions the traditional stochastic gradient descent method is not optimal, while for non-convex functions, to reach the corresponding lower bound we need to solve a series of strongly convex subproblems as the inner loop (see Algorithm 3 for reference). These facts necessitate the use of **Nesterov's acceleration** to attain the lower bounds for both strongly convex and non-convex functions. Second, the Byzantine nodes can utilize the stochastic gradient noise of the honest nodes to cover their attacks and maximize the error of the robust aggregator. Consequently, **variance reduction** within each honest node is essential for the performance improvement (Wu et al., 2020; Karimireddy et al., 2021; Gorbunov et al., 2022).

Below, we propose a Byzantine-robust distributed stochastic Nesterov's accelerated method with variance reduction (Byrd-Nester) to serve as the cornerstone of the subsequent optimal method design. Byrd-Nester applies **Nesterov's acceleration** in a distributed and stochastic manner, and utilizes the mini-batch technique for **variance reduction**; see Algorithm 1.

At the t-th iteration, each honest node $i \in \mathcal{H}$ queries a mini-batch of m stochastic gradients at the auxiliary point y^{t-1}, and averages them to calculate the mini-batch stochastic gradient g_i^{t-1} as in (10). The purpose of this step is to reduce the variance of the stochastic gradient noise. Then, each honest node $i \in \mathcal{H}$ calculates s_i^t, a weighted combination of the

Algorithm 1 Byzantine-robust distributed stochastic Nesterov's accelerated method with variance reduction (Byrd-Nester)

Input: initial point x^0, auxiliary point $y^0 = x^0$, maximum number of iterations T, batch size m_0, m, step size η, $\theta \in (0,1]$, $\beta \in [0,1)$, $\alpha \in [0,1]$, $\hat{s}^0 = s_i^0 = \frac{1}{m_0}\sum_{l=1}^{m_0} \nabla F(y^0, \xi_i^{(0,l)})$.
for $t = 1, \cdots, T$ **do**
  for node $i \in \mathcal{H}$ **do**
    Independently sample $\{\xi_i^{(t-1,1)}, \cdots, \xi_i^{(t-1,m)}\}$, obtain stochastic gradients from oracle $\mathsf{O} \in \mathcal{O}$ and calculate

$$g_i^{t-1} = \frac{1}{m}\sum_{l=1}^{m} \nabla F(y^{t-1}; \xi_i^{(t-1,l)}), \tag{10}$$

$$s_i^t = \beta s_i^{t-1} + \theta g_i^{t-1}. \tag{11}$$

    Send g_i^{t-1} and s_i^t to server.
  end for
  for node $i \in \mathcal{B}$ **do**
    Send arbitrary vector $g_i^{t-1} \in \mathbb{R}^d$ and $s_i^t \in \mathbb{R}^d$ to server.
  end for
  Server receives $\{g_i^{t-1}\}_{i=1}^n$ and $\{s_i^t\}_{i=1}^n$, and updates

$$s^t = \beta \hat{s}^{t-1} + \theta \mathsf{A}(\{g_i^{t-1}\}_{i=1}^n), \tag{12}$$

$$\hat{s}^t = (1-\alpha) s^t + \alpha \mathsf{A}(\{s_i^t\}_{i=1}^n), \tag{13}$$

$$x^t = x^{t-1} - \eta \hat{s}^t, \tag{14}$$

$$y^t = x^t + \beta(x^t - x^{t-1}). \tag{15}$$

  Server sends y^t to all nodes.
end for
return $\tilde{x}^K = x^T$ for strongly convex optimization; $\tilde{x}^K = y^{t'}$ where t' is randomly chosen from $0, \cdots, T-1$ for non-convex optimization. Here K is the number of oracle queries.

historical and current mini-batch stochastic gradients, for the sake of node-level acceleration as in (11). It is worth noting that $\beta + \theta$ may exceed 1, representing the aggressive usage of the mini-batch stochastic gradients. After that, each honest node $i \in \mathcal{H}$ sends g_i^{t-1} and s_i^t to the server. In contrast, each Byzantine node $i \in \mathcal{B}$ may send two arbitrary d-dimensional vectors g_i^{t-1} and s_i^t to the server.

Then, the server aggregates the received $\{g_i^{t-1}\}_{i=1}^n$ and $\{s_i^t\}_{i=1}^n$ via a robust aggregator $\mathsf{A} \in \mathcal{A}$. Therein, (12) uses $\mathsf{A}(\{g_i^{t-1}\}_{i=1}^n)$ to calculate s^t for server-level acceleration. Note that the update involves $\hat{s}^{t-1}$ instead of s^{t-1}. Meanwhile, as shown in (13), $\hat{s}^t$ is a linear combination of s^t and $\mathsf{A}(\{s_i^t\}_{i=1}^n)$, parameterized by $\alpha \in [0,1]$ to adjust the balance between node-level and server-level accelerations. In particular, $\alpha = 0$ voids node-level acceleration, while $\alpha = 1$ renders server-level acceleration ineffective. Such a design offers more flexibility to the proposed method. Using $\hat{s}^t$, (14) runs a descent step to update x^t, while (15) runs an

extrapolation step to y^t, differentiating the adopted Nesterov's acceleration from the momentum acceleration. Finally, the server sends y^t to all nodes.

Note that the node-level momentum acceleration has also been utilized in stochastic optimization and its Byzantine-robust distributed variant (Liu et al., 2020; Karimireddy et al., 2021) for variance reduction, in the form of $s_i^t = \beta s_i^{t-1} + (1-\beta) g_i^{t-1}$ that is similar to (11). However, its variance reduction effect relies on setting β sufficiently close to 1 and $1-\beta$ to 0. Our choices of β and θ do not satisfy these requirements; see the lemmas, theorems and corollaries in the following subsections. This fact explains why we still need the mini-batch technique for variance reduction, on top of the node-level Nesterov's acceleration.

To better understand the behavior of Algorithm 1 and facilitate the subsequent analysis, we provide a deeper examination of (14). Observe that (14) is equivalent to

$$\begin{aligned} x^t =& x^{t-1} - \eta \hat{s}^t = x^{t-1} - \eta \bar{s}^t + \eta \bar{s}^t - \eta \hat{s}^t \\ =& x^{t-1} - \eta\beta \bar{s}^{t-1} - \frac{\eta\theta}{|\mathcal{H}|m} \sum_{i\in\mathcal{H}} \sum_{l=1}^{m} \nabla F(y^{t-1}, \xi_i^{(t-1,l)}) + \eta \bar{s}^t - \eta \hat{s}^t \\ =& x^{t-1} - \eta\beta \hat{s}^{t-1} - \eta\theta \nabla f(y^{t-1}) + \eta\theta \nabla f(y^{t-1}) - \frac{\eta\theta}{|\mathcal{H}|m} \sum_{i\in\mathcal{H}} \sum_{l=1}^{m} \nabla F(y^{t-1}, \xi_i^{(t-1,l)}) \\ & + \eta \bar{s}^t - \eta \hat{s}^t - \eta\beta(\bar{s}^{t-1} - \hat{s}^{t-1}) \\ =& \underbrace{y^{t-1} - \eta\theta \nabla f(y^{t-1})}_{\text{Nesterov's acceleration}} + \underbrace{\eta \bar{s}^t - \eta \hat{s}^t - \eta\beta(\bar{s}^{t-1} - \hat{s}^{t-1})}_{\Delta_1^t\text{: aggregation bias}} \\ & \underbrace{+ \eta\theta \nabla f(y^{t-1}) - \frac{\eta\theta}{|\mathcal{H}|m} \sum_{i\in\mathcal{H}} \sum_{l=1}^{m} \nabla F(y^{t-1}; \xi_i^{(t-1,l)}),}_{\Delta_2^t\text{: stochasticity bias}} \end{aligned} \tag{16}$$

where $\bar{s}^t = \frac{1}{|\mathcal{H}|} \sum_{i\in\mathcal{H}} s_i^t$. According to (16), the update of x^t consists of three parts: Nesterov's acceleration, aggregation bias Δ_1^t and stochasticity bias Δ_2^t. If the robust aggregator A is ideal such that $\rho = 0$, Algorithm 1 reduces to the distributed stochastic Nesterov's accelerated method. If further the stochastic gradient variance $\sigma^2 = 0$, it turns to the distributed deterministic Nesterov's accelerated method.

4.1 Strongly convex optimization

For strongly convex optimization, we first analyze the oracle query complexity of Algorithm 1 and show that it has an $O(\log \epsilon^{-1})$ gap to the $\Omega(\epsilon^{-2})$ lower bound in Theorem 11. We set β in Algorithm 1 as

$$\beta = \frac{\sqrt{q} - 1}{\sqrt{q} + 1}, \tag{17}$$

where $q \geq 1$ is a constant. We will set $q = \frac{L}{\mu\theta} = \frac{\kappa}{\theta}$ in the ensuing analysis. The following lemma provides an effective tool to establish the convergence of Algorithm 1.

Lemma 12 *Given $\zeta^2 \geq 0$, $\rho \geq 0$, $\sigma^2 \geq 0$, $L > 0$ and $\delta \in [0, \delta_{\max}]$, for any distributed problem in the form of (1) having at least $(1-\delta)n$ honest nodes with any function $f \in \mathcal{F}$*

and μ-strongly convex $\{f_i(x)\}_{i\in\mathcal{H}}$, any stochastic gradient oracle $\mathsf{O}\in\mathcal{O}$ and any $(\delta_{\max},\rho)$-robust aggregator $\mathsf{A}\in\mathcal{A}$, if there exist an $\frac{L}{\theta}$-strongly convex function $h^t(x)$, and parameters υ^t and ε^t at each iteration t such that

(i) $\mathbb{E}[h^t(x)]\le f(x)+\frac{L-\mu\theta}{2\theta}\|x-y^{t-1}\|^2$ *for* $x=q^{-1/2}x^*+(1-q^{-1/2})x^{t-1}$,

(ii) $\mathbb{E}[f(x^t)]\le\mathbb{E}[h^t(x^{t*})]+\upsilon^t$,

(iii) $\mathbb{E}[h^t(x^t)]\le\mathbb{E}[h^t(x^{t*})]+\varepsilon^t$,

then with $\beta=\frac{\sqrt{q}-1}{\sqrt{q}+1}$ and $q=\frac{L}{\mu\theta}=\frac{\kappa}{\theta}$, the iterate x^t generated by Algorithm 1 satisfies

$$\mathbb{E}[f(x^t)-f^*]\le\left(1-\frac{1}{2\sqrt{q}}\right)^t\left(2(f(x^0)-f^*)+4\sum_{\tau=1}^{t}\left(1-\frac{1}{2\sqrt{q}}\right)^{-\tau}(\upsilon^\tau+\sqrt{q}\varepsilon^\tau)\right). \quad (18)$$

Therein, $x^*=\arg\min_x f(x)$, $x^{t*}=\arg\min_x h^t(x)$ and $f^*=\inf_x f(x)$.

Proof See Appendix A.4. ■

The result of Lemma 12 relies on the existence of a proper surrogate function $h^t(x)$ and we will discuss later. In (18), the term of $1-1/(2\sqrt{q})$ implies the accelerated convergence. Nevertheless, the convergence is negatively affected by the residual $\upsilon_\tau+\sqrt{q}\varepsilon_\tau$, in which υ^t measures the appropriateness of $h^t(x)$ as a surrogate function (see Lemma 2.2.1 in (Nesterov, 2003)), while ε^t quantifies the gap between x^t and the minimizer of $h^t(x)$.

Now, we design a set of $\{h^t(x),\upsilon^t,\varepsilon^t\}_{t=1}^T$ that satisfy the three conditions in Lemma 12. First, the surrogate function $h^t(x)$ is given by

$$h^t(x):=f(y^{t-1})+\langle\nabla f(y^{t-1}),x-y^{t-1}\rangle+\frac{L}{2\theta}\|x-y^{t-1}\|^2, \quad (19)$$

where θ has been introduced in (11) and (12). For such a surrogate function, $x^{t*}=y^{t-1}-\frac{\theta}{L}\nabla f(y^{t-1})$. Second, we set $\upsilon^t=\varepsilon^t=\frac{L}{\theta}\mathbb{E}[\|\Delta_1^t\|^2+\|\Delta_2^t\|^2]$. Below, we verify the three conditions in Lemma 12 one by one.

(i) It obviously holds from the strong convexity of f.

(ii) Consider

$$f(x^t)\le h^t(x^t)\le h^t(x^{t*})+\frac{L}{2\theta}\|x^t-x^{t*}\|^2,$$

where the first inequality follows from the L-smoothness of f and the second inequality follows from the $\frac{L}{\theta}$-smoothness of h^t. Taking expectations and letting the step size $\eta=\frac{1}{L}$, we have

$$\begin{aligned}\mathbb{E}[f(x^t)]\le&\mathbb{E}[h^t(x^{t*})]+\frac{L}{2\theta}\mathbb{E}[\|x^t-x^{t*}\|^2]=\mathbb{E}[h^t(x^{t*})]+\frac{L}{2\theta}\mathbb{E}[\|\Delta_1^t+\Delta_2^t\|^2]\\ \le&\mathbb{E}[h^t(x^{t*})]+\upsilon^t.\end{aligned}$$

where the equality is due to (16) and the fact of $x^{t*}=y^{t-1}-\frac{\theta}{L}\nabla f(y^{t-1})$.

(iii) Again, consider

$$h^t(x^t) \le h^t(x^{t*}) + \frac{L}{2\theta}\|x^t - x^{t*}\|^2,$$

that we have derived from the $\frac{L}{\theta}$-smoothness of h^t. Following the similar derivation as in (ii), we have

$$\begin{aligned}\mathbb{E}[h^t(x^t)] \le& \mathbb{E}[h^t(x^{t*})] + \frac{L}{2\theta}\mathbb{E}[\|x^t - x^{t*}\|^2] = \mathbb{E}[h^t(x^{t*})] + \frac{L}{2\theta}\mathbb{E}[\|\Delta_1^t + \Delta_2^t\|^2] \\ \le& \mathbb{E}[h^t(x^{t*})] + \varepsilon^t.\end{aligned}$$

To establish the convergence of Algorithm 1, it remains to bound υ^t and ε^t; that is, to bound $\frac{L}{\theta}\mathbb{E}[\|\Delta_1^t\|^2 + \|\Delta_2^t\|^2]$. In Appendices A.5 and A.6, we respectively bound the aggregation bias $\mathbb{E}[\|\Delta_1^t\|^2]$ and the stochasticity bias $\mathbb{E}[\|\Delta_2^t\|^2]$. With them, we have

$$\begin{aligned}\upsilon^t = \varepsilon^t &= \frac{L}{\theta}\mathbb{E}[\|\Delta_1^t\|^2 + \|\Delta_2^t\|^2] \\ &\le \frac{1}{L\theta}\left(\frac{3\chi_4\rho\delta\sigma^2}{m}\left(1 + \frac{1}{(1-\delta)n}\right) + \frac{\theta^2\sigma^2}{(1-\delta)nm} + 3\chi_5\rho\delta\zeta^2\right),\end{aligned} \tag{20}$$

where $\chi_4, \chi_5 \ge 0$ are some constants. It is worth noting that the first two terms at the right-hand side of (20) are controlled by the batch size m, implying that we can reduce the effect of the variance σ^2 by increasing m. The last term is a constant error, causing the Byzantine error in Section 3.1.

Hence, we establish the convergence result of Algorithm 1 in terms of both the function value and the gradient norm as follows.

Theorem 13 *Given $\zeta^2 \ge 0$, $\rho \ge 0$, $\sigma^2 \ge 0$, $L > 0$ and $\delta \in [0, \delta_{\max}]$, for any distributed problem in the form of (1) having at least $(1-\delta)n$ honest nodes with any function $f \in \mathcal{F}$ and μ-strongly convex $\{f_i(x)\}_{i\in\mathcal{H}}$, any stochastic gradient oracle $\mathsf{O} \in \mathcal{O}$ and any $(\delta_{\max}, \rho)$-robust aggregator $\mathsf{A} \in \mathcal{A}$, consider Algorithm 1 with the step size $\eta = \frac{1}{L}$. With $\beta = \frac{\sqrt{q}-1}{\sqrt{q}+1}$ and $q = \frac{L}{\mu\theta} = \frac{\kappa}{\theta}$ where $\kappa = \frac{L}{\mu}$, if the parameters α and θ meet the requirements in Lemma 20, then under Assumptions 1–3, the iterate x^t generated by Algorithm 1 satisfies*

$$\mathbb{E}[f(x^t) - f^*] \le 2\left(1 - \frac{1}{2\sqrt{q}}\right)^t \Delta + \frac{16}{\mu\theta^2}\left(\frac{6\chi_4\rho\delta\sigma^2}{m} + \frac{\theta^2\sigma^2}{(1-\delta)nm} + 3\chi_5\rho\delta\zeta^2\right), \tag{21}$$

$$\mathbb{E}[\|\nabla f(x^t)\|^2] \le 2\left(1 - \frac{1}{2\sqrt{q}}\right)^t L^2R^2 + \frac{32\kappa}{\theta^2}\left(\frac{6\chi_4\rho\delta\sigma^2}{m} + \frac{\theta^2\sigma^2}{(1-\delta)nm} + 3\chi_5\rho\delta\zeta^2\right), \tag{22}$$

where $\Delta = f(x^0) - f^*$ and $R = \|x^0 - x^*\|$.

Proof Combining Lemmas 12, 20 and 21, as well as utilizing the closed form formula for the sum of a geometric series, we obtain (21). Further from $\|\nabla f(x^t)\|^2 \le 2L\left(f(x^t) - f^*\right)$ and $f(x^0) - f^* \le \frac{1}{2}LR^2$, we obtain (22). ■

We specify the values of α and θ to finalize the oracle query complexity of Algorithm 1.

Corollary 14 *Under the conditions in Theorem 13, we define* $\kappa = \frac{L}{\mu}$ *and* $q = \frac{L}{\mu\theta} = \frac{\kappa}{\theta}$*, set* $\alpha = 0$*,* $\theta = 1$ *and* $\beta = \frac{\sqrt{q}-1}{\sqrt{q}+1} = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$*, as well as set the parameters* T *and* m *as*

$$T = 2\sqrt{\kappa}\log\frac{4L^2R^2}{\epsilon^2}, \quad m = m_0 = 64\kappa\left(3\rho\delta(1+\frac{1}{(1-\delta)n}) + \frac{1}{(1-\delta)n}\right)\frac{\sigma^2}{\epsilon^2}.$$

Then, the output of Algorithm 1 defined by $\tilde{x}^K = x^T$ *satisfies*

$$\mathbb{E}[\|\nabla f(\tilde{x}^K)\|] \leq \epsilon + 4\sqrt{6}\kappa^{1/2}\rho^{1/2}\delta^{1/2}\zeta,$$

with the oracle query complexity

$$K = m_0 + mT = O\left(\kappa^{3/2}\left(\rho\delta + \frac{1}{(1-\delta)n}\right)\frac{\sigma^2}{\epsilon^2}\cdot\log\frac{LR}{\epsilon}\right). \tag{23}$$

Proof When $\alpha = 0$ and $\theta = 1$, we have $\chi_4 = \chi_5 = 1$. Substituting the values of T and m into (22) yields the above result. ■

As shown in Corollary 14, the oracle query complexity of Algorithm 1 in terms of ϵ is $K = mT = O(\epsilon^{-2}\log\epsilon^{-1})$, with an $O(\log\epsilon^{-1})$ gap to the $\Omega(\epsilon^{-2})$ lower bound in Theorem 11. To close this gap, we apply the restart technique and utilize an increasing batch size, yielding the optimal Algorithm 2.

Algorithm 2 Byrd-Nester with restart (Byrd-reNester)

Input: initial point $z(0)$, $T(1)$ in (57) and $T(p) = \lceil\frac{2L^{1/2}}{\mu^{1/2}}\log 8\rceil$ for $p \geq 2$, $m(p) = 2^{p-1}$, and $P = \max\{\lceil\log_2\frac{4L(\epsilon(1))^2}{\epsilon^2}\rceil, 1\}$ with $(\epsilon(1))^2$ in (56).
for $p = 1, \cdots, P$ **do**
  Output $z(p)$ from Byrd-Nester with $x^0 = z(p-1)$, $T = T(p)$ and $m = m(p)$.
end for
return $\tilde{x}^K = z(P)$.

Here, $T(p)$, $m(p)$ and $\epsilon(p)$ respectively represent the maximum number of Byrd-Nester calls, the batch size, the expected optimization error for the p-th call of Byrd-Nester in Algorithm 2. The parameters α, θ and β are the same as those within Corollary 14. The optimal oracle query complexity of Algorithm 2 is established in the following Theorem.

Theorem 15 *Under the conditions in Theorem 13, we define* $\kappa = \frac{L}{\mu}$ *and* $q = \frac{L}{\mu\theta} = \frac{\kappa}{\theta}$*, set* $\alpha = 0$*,* $\theta = 1$ *and* $\beta = \frac{\sqrt{q}-1}{\sqrt{q}+1} = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$*. Then, the output of Algorithm 2 defined by* $\tilde{x}^K$ *satisfies*

$$\mathbb{E}[\|\nabla f(\tilde{x}^K)\|] \leq 8\sqrt{2}\kappa^{1/2}\rho^{1/2}\delta^{1/2}\zeta + \epsilon, \tag{24}$$

with the oracle query complexity

$$K = \sum_{p=1}^{P} m(p)T(p) = O\left(\kappa^{1/2}\log\frac{LR}{\epsilon} + \kappa^{3/2}\left(\rho\delta + \frac{1}{(1-\delta)n}\right)\frac{\sigma^2}{\epsilon^2}\right). \tag{25}$$

Proof See Appendix A.7. ■

The oracle query complexity established in Theorem 15 is optimal, exactly matching the lower bound for strongly convex optimization in Theorem 11. Below, we demonstrate that Algorithm 2 is also optimal in the following special cases.

- **Specialization to $\zeta^2 = 0$.** Setting $\zeta^2 = 0$ yields $\mathbb{E}[\|\nabla f(\tilde{x}^K)\|] \leq \epsilon$ and the oracle query complexity is

$$O\left(\frac{\rho\delta\sigma^2}{\epsilon^2} + \frac{\sigma^2}{(1-\delta)n\epsilon^2} + \kappa^{1/2}\log\frac{LR}{\epsilon}\right).$$

  This complexity matches the lower bound for strongly convex optimization with data homogeneity ($\zeta^2 = 0$), as shown in Theorem 11.

- **Specialization to $\delta = 0$.** Setting $\delta = 0$ yields $\mathbb{E}[\|\nabla f(\tilde{x}^K)\|] \leq \epsilon$ and the oracle query complexity is

$$O\left(\frac{\kappa^{3/2}}{n}\frac{\sigma^2}{\epsilon^2} + \kappa^{1/2}\log\frac{LR}{\epsilon}\right).$$

  This complexity matches the lower bound for strongly convex optimization in the absence of Byzantine nodes ($\delta = 0$), as shown in Theorem 11.

- **Specialization to $\sigma^2 = 0$.** Setting $\sigma^2 = 0$, $P = 1$, $m(1) = 1$, and $T(1) = 2\kappa^{1/2}\log\frac{2L^2R^2}{\epsilon^2}$ yields $\mathbb{E}[\|\nabla f(\tilde{x}^K)\|] \leq 8\sqrt{2}\kappa^{1/2}\rho^{1/2}\delta^{1/2}\zeta + \epsilon$ and the oracle query complexity is

$$O\left(\kappa^{1/2}\log\frac{LR}{\epsilon}\right).$$

  This complexity matches the lower bound for strongly convex optimization without stochasticity ($\sigma^2 = 0$), as shown in Theorem 11.

- **Specialization to $\sigma^2 = \zeta^2 = 0$ or $\sigma^2 = \delta = 0$.** Setting $\sigma^2 = \zeta^2$ (or δ) $= 0$, $K = 1$, $m(1) = 1$ and $T(1) = 2\sqrt{\kappa}\log\frac{2L^2R^2}{\epsilon^2}$ yields $\mathbb{E}[\|\nabla f(\tilde{x}^K)\|] \leq \epsilon$ and the oracle query complexity is

$$O\left(\kappa^{1/2}\log\frac{LR}{\epsilon}\right).$$

  This complexity matches the lower bound for strongly convex optimization without stochasticity but with data homogeneity ($\sigma^2 = \zeta^2 = 0$) or without stochasticity but in the absence of Byzantine nodes ($\sigma^2 = \delta = 0$), as shown in Theorem 11.

4.2 Non-convex optimization

For non-convex optimization, we begin with establishing the convergence of Algorithm 1.

Theorem 16 *Given $\zeta^2 \geq 0$, $\rho \geq 0$, $\sigma^2 \geq 0$, $L > 0$ and $\delta \in [0, \delta_{\max}]$, for any distributed problem in the form of (1) having at least $(1-\delta)n$ honest nodes with any function $f \in \mathcal{F}$,*

any stochastic gradient oracle $\mathsf{O} \in \mathcal{O}$ *and any* $(\delta_{\max}, \rho)$*-robust aggregator* $\mathsf{A} \in \mathcal{A}$*, consider Algorithm 1 with the step size* η *set in* (68)*. With* $\beta = 1 - 12L\eta$*, if batch size* $m = O(1)$*,* $m_0 = m/(L^2\eta^2)$ *and the parameters* α*,* θ *meet the requirements in Lemma 20 with* $\chi_1 + \chi_2 = O(L\eta)$ *and* $\chi_3 = O(1) \geq 0$*, as well as*

$$(1-\theta-\beta)^2 \leq \chi_6(1-\beta)^2,$$
$$\chi_7 \leq \frac{1}{3} - 6\chi_6 - (\theta + \beta^2 + \theta\beta - 1)^2,$$

for some $\chi_6 \geq 0$ *and* $\chi_7 = \Theta(1) > 0$*, then the iterate* y^t *generated by Algorithm 1 satisfies*

$$\frac{1}{T}\sum_{t=1}^{T} \|\nabla f(y^{t-1})\|^2 \leq O\left(\sqrt{\frac{L\Delta + \sigma^2/n}{T}}\sqrt{\left(\frac{1}{(1-\delta)n} + \rho\delta\right)\sigma^2} + \frac{L\Delta}{T} + \frac{\sigma^2}{Tn} + \rho\delta\zeta^2\right),$$

where $\Delta = f(x^0) - f^*$*.*

Proof See Appendix A.8. ■

Based on Theorem 16, we specify the values of α and θ to determine the oracle query complexity of Algorithm 1 in the following corollary.

Corollary 17 *Under conditions of Theorem 16, let* $\theta = 1 - \beta$ *and* $\alpha = 1$ *so that* $\chi_1 = 1 - \beta = 12L\eta$*,* $\chi_2 = \chi_6 = 0$*,* $\chi_3 = 1$ *and* $\chi_7 = \frac{1}{3}$*. Consider Algorithm 1 with the step size* η *set in* (68)*. The output of Algorithm 1 defined by* $\tilde{x}^K$ *satisfies*

$$\mathbb{E}[\|\nabla f(\tilde{x}^K)\|] \leq \sqrt{210}\rho^{1/2}\delta^{1/2}\zeta + \epsilon,$$

with the oracle query complexity

$$K = m_0 + mT = O\left(\frac{L\Delta\rho\delta\sigma^2}{\epsilon^4} + \frac{L\Delta\sigma^2}{(1-\delta)n\epsilon^4} + \frac{\rho\delta\sigma^4}{(1-\delta)n\epsilon^4} + \frac{L\Delta}{\epsilon^2} + \frac{\sigma^2}{(1-\delta)n\epsilon^2}\right).$$

In Corollary 17, $\theta = 1 - \beta = 12L\eta = O(\frac{1}{\sqrt{T}})$, vanishing as T goes to infinity. Therefore, the node-level Nesterov's acceleration in (11) is also effective for variance reduction (Liu et al., 2020; Karimireddy et al., 2021), as shown in Lemma 23. For this reason, Algorithm 1 no longer requires a large batch size m. However, there is also an $O(\frac{\rho\delta\sigma^4}{(1-\delta)n\epsilon^4} + \frac{\sigma^2}{(1-\delta)n\epsilon^2})$ gap between the oracle query complexity in Corollary 17 and the lower bound in Theorem 11. The key idea to close this gap is to approximately solve a series of strongly convex surrogates, each calling Byrd-reNester once, via an inexact proximal point algorithm outlined in Algorithm 3. We establish the oracle query complexity of Algorithm 3 in the following theorem.

Theorem 18 *Given* $\zeta^2 \geq 0$*,* $\rho \geq 0$*,* $\sigma^2 \geq 0$*,* $L > 0$*, and* $\delta \in [0, \delta_{\max}]$*, for any distributed problem in the form of* (1) *having at least* $(1-\delta)n$ *honest nodes with any function* $f \in \mathcal{F}$*, any stochastic gradient oracle* $\mathsf{O} \in \mathcal{O}$ *and any* $(\delta_{\max}, \rho)$*-robust aggregator* $\mathsf{A} \in \mathcal{A}$*, consider*

Algorithm 3 Inexact Proximal Point Algorithm with Byrd-reNester

Input: initial point $\varkappa^0$, maximum number of Byrd-reNester calls Γ.
for $\gamma = 1, \cdots, \Gamma$ **do**
  Set $f_i^\gamma(z) = f_i(z) + L\|z - \varkappa^{\gamma-1}\|^2$ for all $i \in \mathcal{H}$.
  Output $\varkappa^\gamma$ by applying Byrd-reNester to $f^\gamma(z) = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} f_i^\gamma(z)$ with $z(0) = \varkappa^{\gamma-1}$.
end for
return $\tilde{x}^K = \varkappa^{\gamma'}$, where γ' is randomly chosen from $1, \cdots, \Gamma$.

Algorithm 3 with $\Gamma = \lceil 32L\Delta\epsilon^{-2}\rceil$ *where* $\Delta = f(\varkappa^0) - f^*$, *the step size* $\eta = \frac{1}{3L}$, $\alpha = 0$, $\beta = \frac{\sqrt{3}-1}{\sqrt{3}+1}$, *and* $\theta = 1$. *Then the output of Algorithm 3 defined by* $\tilde{x}^K$ *satisfies*

$$\mathbb{E}[\|\nabla f(\tilde{x}^K)\|] \le 16\sqrt{5}\rho^{1/2}\delta^{1/2}\zeta + \epsilon, \tag{26}$$

with oracle query complexity

$$K = O\left(\frac{L\Delta\rho\delta\sigma^2}{\epsilon^4} + \frac{L\Delta\sigma^2}{(1-\delta)n\epsilon^4} + \frac{L\Delta}{\epsilon^2}\log\frac{L\Delta(1+\rho\delta\zeta^2)}{\epsilon^2}\right). \tag{27}$$

Proof See Appendix A.9. ■

Comparing the upper bound of oracle query complexity in Theorem 18 with the lower bound in Theorem 11, we notice that there is only a logarithmic factor gap between their third terms. The source of the third term in (27) is that Algorithm 3 solves $O(\epsilon^{-2})$ strongly convex surrogates, each surrogate calls Byrd-reNester once, each Byrd-reNester calls Byrd-Nester once, and the complexity of Byrd-Nester is $O(\log\epsilon^{-2})$. Nevertheless, whenever ϵ is small, the third term is not dominant compared to the first and second terms, such that the logarithmic factor gap is negligible.

For the following special cases, the established oracle query complexity of Algorithm 3 remains optimal (up to logarithmic factors).

- **Specialization to** $\zeta^2 = 0$. Setting $\zeta^2 = 0$ yields $\mathbb{E}[\|\nabla f(\tilde{x}^K)\|] \le \epsilon$ and the oracle query complexity is

  $$O\left(\frac{L\Delta\rho\delta\sigma^2}{\epsilon^4} + \frac{L\Delta\sigma^2}{(1-\delta)n\epsilon^4} + \frac{L\Delta}{\epsilon^2}\log\frac{L\Delta}{\epsilon^2}\right).$$

  This complexity matches the lower bound for non-convex optimization with data homogeneity ($\zeta^2 = 0$), as shown in Theorem 11.

- **Specialization to** $\delta = 0$. Setting $\delta = 0$ demonstrates that to achieve $\mathbb{E}[\|\nabla f(\tilde{x}^K)\|] \le \epsilon$, the oracle query complexity is

  $$O\left(\frac{L\Delta\sigma^2}{n\epsilon^4} + \frac{L\Delta}{\epsilon^2}\log\frac{L\Delta}{\epsilon^2}\right).$$

  This complexity matches the lower bound for non-convex optimization in the absence of Byzantine nodes ($\delta = 0$), as shown in Theorem 11.

- **Specialization to $\sigma^2 = 0$.** Setting $\sigma^2 = 0$ yields $\mathbb{E}[\|\nabla f(\tilde{x}^K)\|] \leq 16\sqrt{5}\rho^{1/2}\delta^{1/2}\zeta + \epsilon$ and the oracle query complexity is

$$O\left(\frac{L\Delta}{\epsilon^2}\log\frac{L\Delta(1+\rho\delta\zeta^2)}{\epsilon^2}\right).$$

  This complexity matches the lower bound for non-convex optimization without stochasticity ($\sigma^2 = 0$), as shown in Theorem 11.

- **Specialization to $\sigma^2 = \zeta^2 = 0$ or $\sigma^2 = \delta = 0$.** Setting $\sigma^2 = \zeta^2$(or δ) $= 0$ yields $\mathbb{E}[\|\nabla f(\tilde{x}^K)\|] \leq \epsilon$ and the oracle query complexity is

$$O\left(\frac{L\Delta}{\epsilon^2}\log\frac{L\Delta}{\epsilon^2}\right).$$

  This complexity exactly matches the lower bound for non-convex optimization without stochasticity but with data homogeneity ($\sigma^2 = \zeta^2 = 0$) or without stochasticity but in the absence of Byzantine nodes ($\sigma^2 = \delta = 0$), as shown in Theorem 11.

Remark 19 *For Byzantine-free single-node stochastic non-convex optimization, a number of variance reduction techniques, such as SARAH (Nguyen et al., 2017; Horváth et al., 2022) and SPIDER (Fang et al., 2018), can improve the oracle query complexity from $O(\epsilon^{-4})$ to $O(\epsilon^{-3})$. We do not investigate these techniques because this paper considers a general stochastic gradient oracle class that satisfy the independence, unbiasedness and bounded variance assumptions (see Section 2.3), but incorporating SARAH or SPIDER requires an additional mean-squared smoothness assumption.*

If we further restrict the considered stochastic gradient oracle class to also satisfy the mean-squared smoothness assumption, it is possible to reduce the oracle query complexity to $O(L\Delta\rho\delta\sigma\epsilon^{-3})$ for non-convex optimization. To establish the lower bound, the analysis still follows the argument of Lemma 10, but the constructed function f should be adapted to the one in Section 3.3 of (Arjevani et al., 2023) so that the dimension is reduced from $d = \frac{L\Delta}{\epsilon^2}$ to $d = 1 + \frac{L\Delta}{\sigma\epsilon}$. Such a reduced dimension eventually yields the reduced lower bound. On the other hand, if we replace the current stochastic gradient estimator with SARAH or SPIDER, an $O(\epsilon^{-3})$ upper bound can be established.

5. Numerical experiments

In this section, we conduct extensive numerical experiments to evaluate the performance of Algorithm 1. Here we do not consider Algorithms 2 and 3, which exhibit strong theoretical guarantees at the cost of complicated hyperparameter tuning.

Experimental setup. We consider two tasks, logistic regression and convolutional neural network training. For the first task, we consider a distributed network of 10 nodes within which 2 are Byzantine. For the second task, we consider a distributed network of 30 nodes within which 5 are Byzantine. The training dataset is MNIST with 10 classes, each having 6,000 training samples. By default, the data distribution is heterogeneous (non-iid): the entire training dataset is sorted by labels, divided into chunks equal to the number of honest nodes, allocated to different honest nodes, and then shuffled within each honest node.

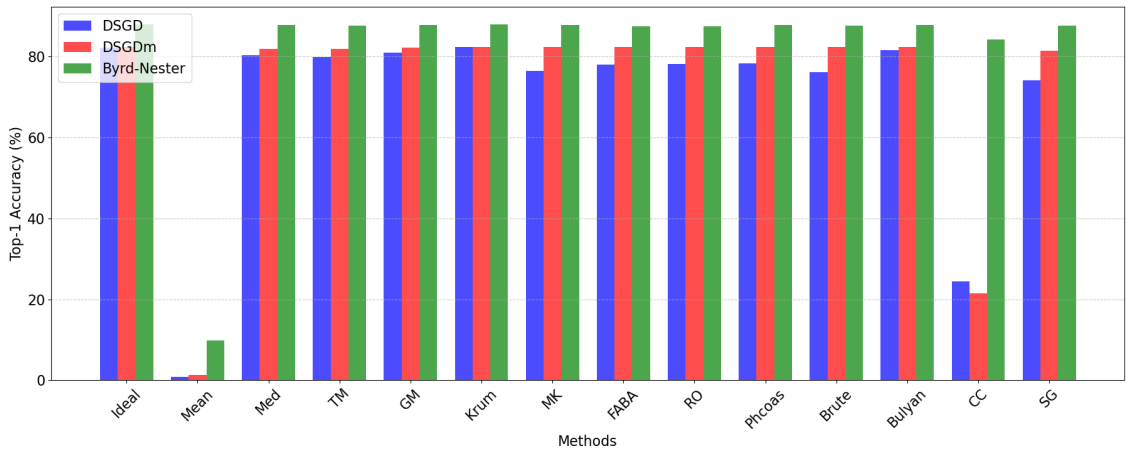

Figure 1: Worst-case maximum top-1 accuracy of DSGD, DSGDm and Algorithm 1.

Byzantine attacks. We implement nine Byzantine attacks, including "Gaussian Attack (GA) (Ye and Ling, 2024)", "Sign Flipping (SF) (Li et al., 2019)", "Label Flipping (LF) (Xiao et al., 2012)", "Sample Duplicating (Prakash and Avestimehr, 2020)", "Zero Value", "Isolation (Song et al., 2020)", "A Little is Enough (ALIE) (Baruch et al., 2019)", "Inner Product Manipulation (IPM) (Xie et al., 2020)", and "Bit Flipping (BF) (Rakin et al., 2019)".

Robust aggregation rules. We implement fourteen robust aggregation rules, including "Ideal", "Mean", "Median (Med) (Yin et al., 2018)", "Trimmed Mean (TM) (Yin et al., 2018)", "Krum (Blanchard et al., 2017)", "Multi Krum (MK) (Blanchard et al., 2017)", "FABA (Xia et al., 2019)", "Remove Outliers (RO) (Xia et al., 2020)", "Phocas (Xie et al., 2018)", "Brute (El Mhamdi et al., 2018)", "Bulyan (El Mhamdi et al., 2018)", "Centered Clipping(CC) (Karimireddy et al., 2021)", "Geometric Median (GM) (Wu et al., 2020)", and "Sign Guard (SG) (Xu et al., 2022)".

Baselines. We choose Byzantine-robust distributed mini-batch SGD (DSGD) and its momentum variant (DSGDm, Karimireddy et al. (2022)) as the baselines. In the three algorithms, the step size is set to 0.1, the batch size is 32, and the total number of epoches is 45. Because the combinations of the compared algorithms, Byzantine attacks and robust aggregation rules are immense, below we only demonstrate some of the results. More results can be found via running our source code at `https://github.com/sqkkk/Byrd-Nester`.

5.1 Logistic regression

First, we consider logistic regression with squared l_2 regularization. For each combination of the three compared algorithms and the fourteen robust aggregation rules, under each of the nine Byzantine attacks, we record the maximum accuracy obtained within the total number of epoches. Then, we depict the minimum of the nine maximum accuracies (termed as the worst-case maximum accuracy) in Figure 1. This performance metric is of practical importance, as it reflects the ability of each combination under the worst-case attack. We observe that Algorithm 1 outperforms DSGD and DSGDm when combined with most of the robust aggregation rules.

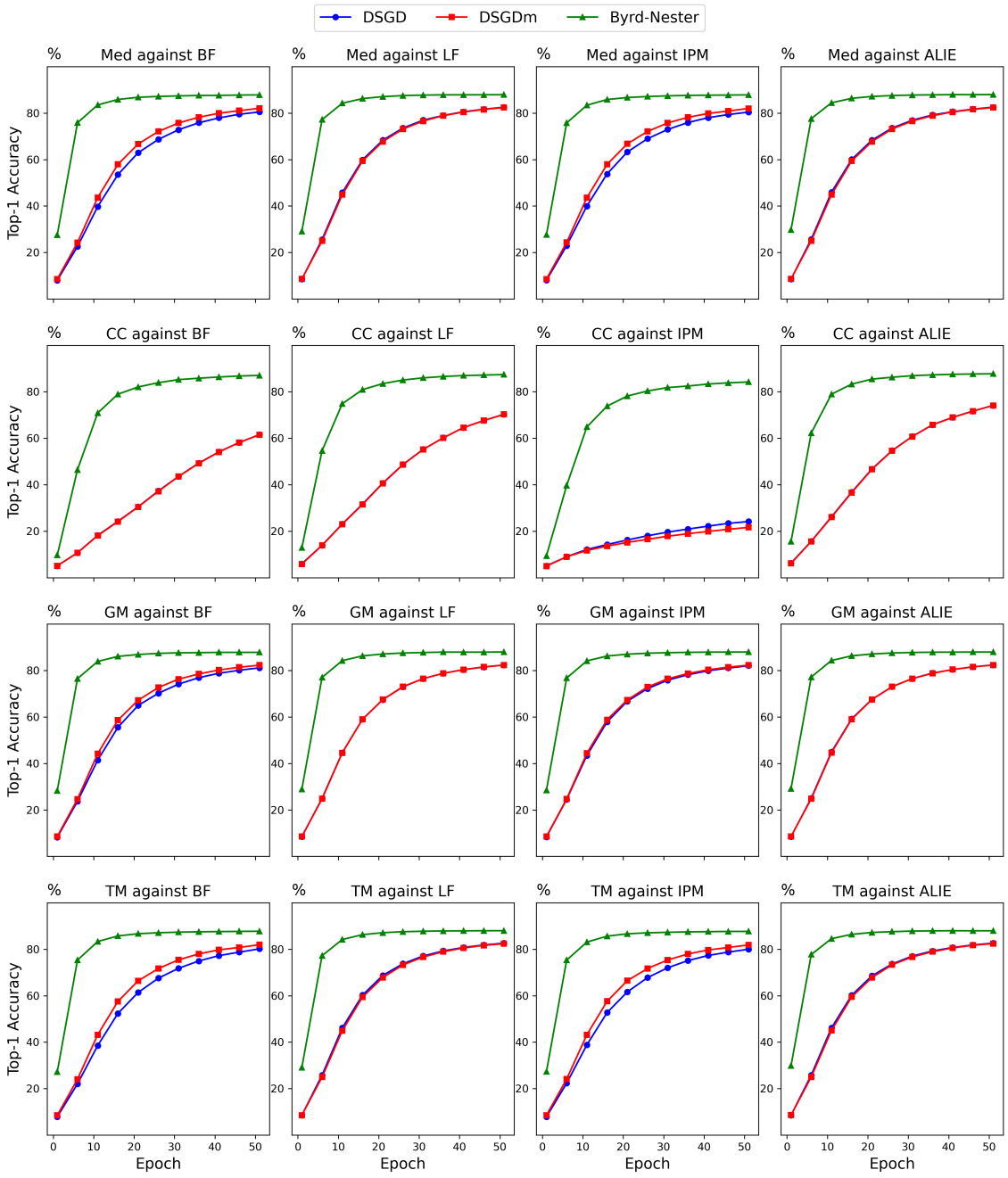

Figure 2: Top-1 test accuracies of DSGD, DSGDm and Algorithm 1 with Med, CC, GM and TM for logistic regression, under BF, LF, IPM and ALIE attacks.

Figure 2 depicts the convergence of the three compared algorithms under BF, LF, IPM and ALIE attacks when combined with Med, CC, GM and TM aggregation rules. DSGD and DSGDm perform similarly, whereas Algorithm 1 enjoys faster convergence and higher accuracy thanks to its effective usage of the historical stochastic gradients.

5.2 Convolutional neural network training

Second, we consider training a convolutional neural network that consists of two convolutional layers, followed by two fully connected layers. Figure 3 depicts the top-1 test accuracies of the three compared algorithms under BF, LF, IPM and ALIE attacks when combined with Med, CC, GM and TM aggregation rules. Algorithm 1 gains the best performance in most cases, while DSGDm is also competitive.

We also check the impact of data heterogeneity. To do so, we use three levels of data heterogeneity: non-iid, iid and semi-iid. The non-iid data distribution has been discussed before. For iid, the entire training dataset is shuffled and then evenly distributed among the honest nodes. For semi-iid, half of the data samples on the honest nodes are iid, while the other half are non-iid. Figure 4 illustrates the performance of Algorithm 1 under varying levels of data heterogeneity, showing that, in general, lower data heterogeneity corresponds to better accuracy.

Further, for the non-iid data distribution, we test the effectiveness of robust aggregator enhancement techniques, such as Bucketing (Karimireddy et al., 2022) and NNM (Allouah et al., 2023), as shown in Figure 5. The results demonstrate that these techniques are generally helpful. This is theoretically explainable from two perspectives: (1) these techniques reduces the data heterogeneity parameter ζ through data mixing; (2) these techniques, when combined with original robust aggregators, yield new robust aggregators with smaller ρ.

6. Conclusions

We established tight lower bounds for Byzantine-robust distributed first-order stochastic methods in both strongly convex and non-convex stochastic optimization. Our key observation was that with data heterogeneity, the convergence error contains a non-vanishing Byzantine error and a vanishing optimization error. Therefore, we respectively established the lower bounds on the Byzantine error and the oracle query complexity to achieve an arbitrarily small optimization error. In contrast, the analysis in (Alistarh et al., 2018) was confined to homogeneous data distribution and did not account for the Byzantine error, while the work of (Karimireddy et al., 2022) did not explore the oracle query complexity. Although the work of (Farhadkhani et al., 2024) considers both aspects, it is restricted to Polyak-Łojasiewicz functions. In contrast, our analysis applies to general strongly convex and non-convex functions. We also observed significant discrepancies between our established lower bounds and the existing upper bounds. To fill this gap, we leveraged the techniques of Nesterov's acceleration and variance reduction to develop novel Byzantine-robust distributed stochastic optimization methods that provably match these lower bounds, up to at most logarithmic factors. This fact implies that our established lower bounds are tight, and our proposed methods can simultaneously attain the optimal Byzantine robustness and the optimal oracle query complexity. Our future work is to explore the extension to Byzantine-robust decentralized stochastic optimization without the aid of any server.

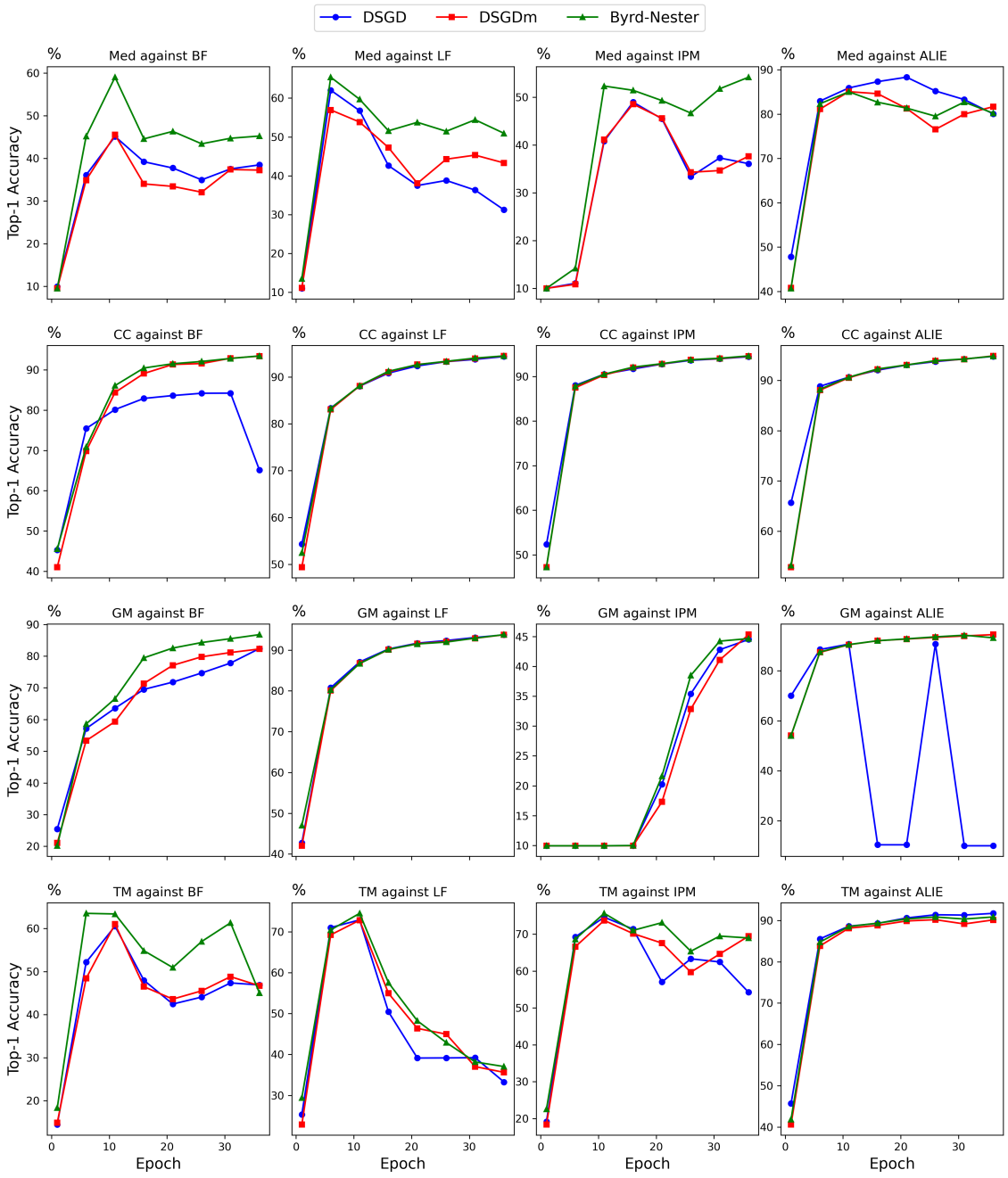

Figure 3: Top-1 test accuracies of DSGD, DSGDm and Algorithm 1 with Med, CC, GM and TM for convolutional neural network training, under BF, LF, IPM and ALIE attacks.

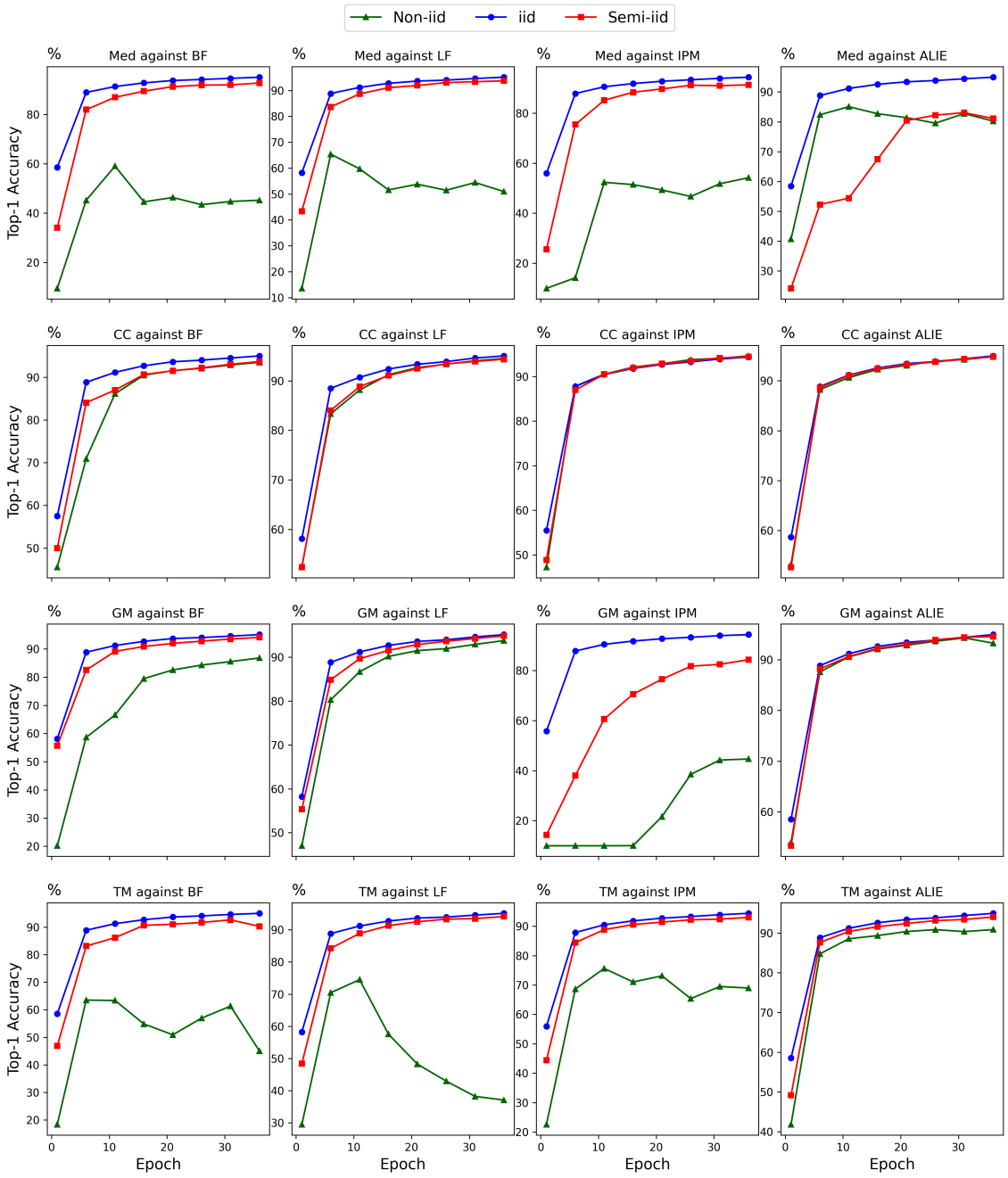

Figure 4: Top-1 test accuracies of Algorithm 1 for convolutional neural network training under different levels of data heterogeneity.

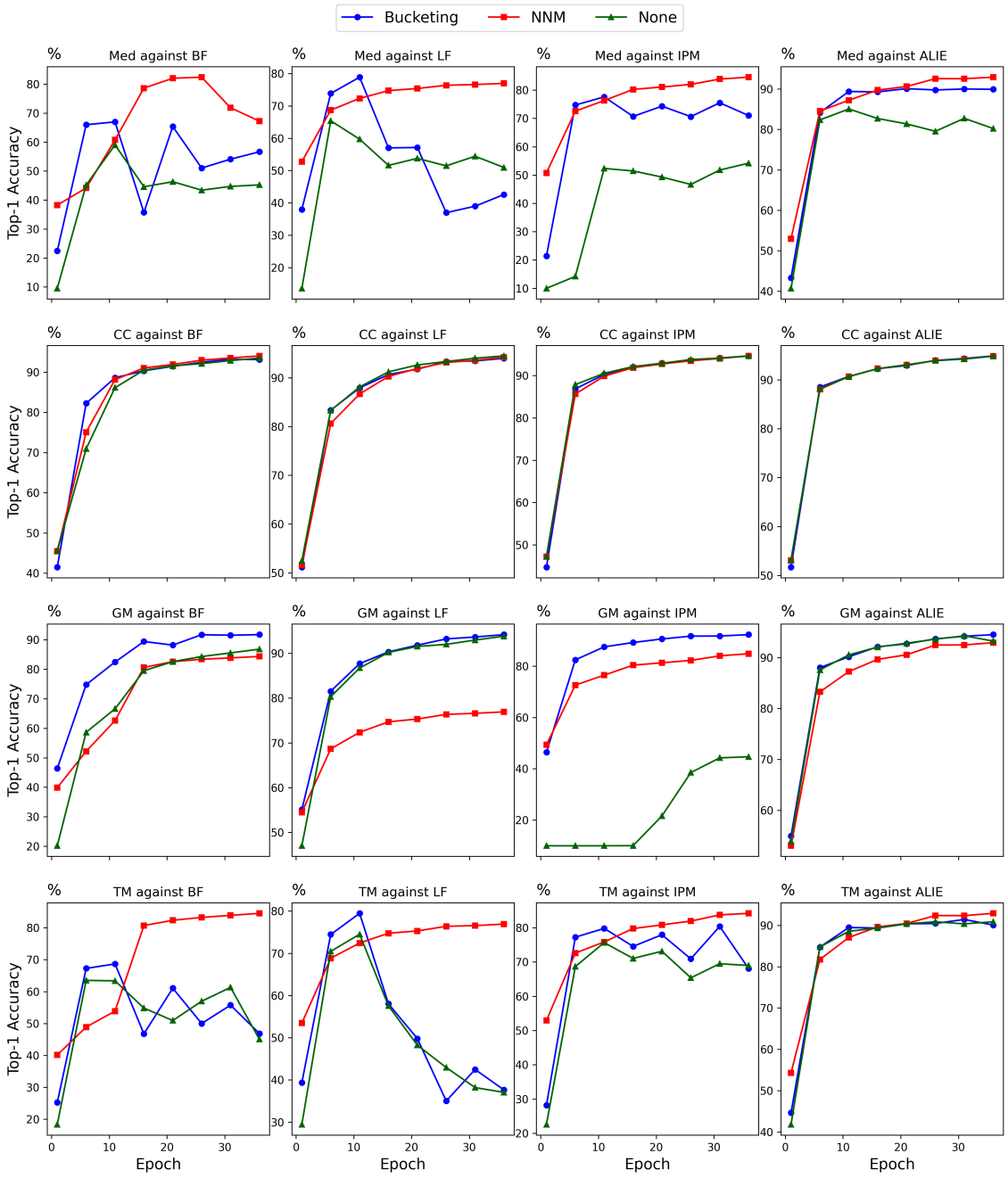

Figure 5: Top-1 test accuracies of Algorithm 1 for convolutional neural network training with robust aggregator enhancement techniques.

Acknowledgments

Qiankun Shi, Jie Peng and Qing Ling (corresponding author) are supported by the National Key R&D Program of China grant 2024YFA1014002, the NSF China grant 62373388, the Guangdong Basic and Applied Basic Research Foundation grant 2023B1515040025, and the Guangdong Provincial Key Laboratory of Mathematical Foundations for Artificial Intelligence grant 2023B1212010001. Kun Yuan is supported by the NSF China grant 12301392. Xiao Wang is supported by the NSF China grant 12271278.

References

Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 2018.

Youssef Allouah, Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafaël Pinot, and John Stephan. Fixing by mixing: A recipe for optimal Byzantine ML under heterogeneity. In *International Conference on Artificial Intelligence and Statistics*, 2023.

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, 199(1–2):165–214, 2023.

Idan Attias, Aryeh Kontorovich, and Yishay Mansour. Improved generalization bounds for adversarially robust learning. *Journal of Machine Learning Research*, 23(175):1–31, 2022.

Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In *Advances in Neural Information Processing Systems*, 2019.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, 2017.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

Xinyang Cao and Lifeng Lai. Distributed approximate Newton's method robust to Byzantine attackers. *IEEE Transactions on Signal Processing*, 68:6011–6025, 2020.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points I. *Mathematical Programming*, 184(1–2):71–120, 2020.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points II: First-order methods. *Mathematical Programming*, 185(1–2):315–355, 2021.

Lingjiao Chen, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos. Draco: Byzantine-resilient distributed training via redundant gradients. In *International Conference on Machine Learning*, 2018.

Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25, 2017.

Deepesh Data and Suhas Diggavi. Byzantine-resilient SGD in high dimensions on heterogeneous data. In *IEEE International Symposium on Information Theory*, 2021.

Maximilian Egger, Mayank Bakshi, and Rawad Bitar. Byzantine-resilient zero-order optimization for communication-efficient heterogeneous federated learning. *arXiv preprint arXiv:2502.00193*, 2025.

El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in Byzantium. In *International Conference on Machine Learning*, 2018.

El Mahdi El-Mhamdi, Sadegh Farhadkhani, Rachid Guerraoui, Arsany Guirguis, Lê-Nguyên Hoang, and Sébastien Rouault. Collaborative learning in the jungle (decentralized, Byzantine, heterogeneous, asynchronous and nonconvex learning). In *Advances in Neural Information Processing Systems*, 2021.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal nonconvex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, 2018.

Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafael Pinot, and John Stephan. Byzantine machine learning made easy by resilient averaging of momentums. In *International Conference on Machine Learning*, 2022.

Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, and Rafael Pinot. Brief announcement: a case for byzantine machine learning. In *Proceedings of the 43rd ACM Symposium on Principles of Distributed Computing*, pages 131–134, 2024.

Dylan J Foster, Ayush Sekhari, Ohad Shamir, Nathan Srebro, Karthik Sridharan, and Blake Woodworth. The complexity of making the gradient small in stochastic convex optimization. In *Conference on Learning Theory*, 2019.

Avishek Ghosh, Raj Kumar Maity, and Arya Mazumdar. Distributed Newton can communicate less and resist Byzantine workers. In *Advances in Neural Information Processing Systems*, 2020.

Eduard Gorbunov, Samuel Horváth, Peter Richtárik, and Gauthier Gidel. Variance reduction is an antidote to Byzantines: Better rates, weaker assumptions and communication compression as a cherry on the top. In *International Conference on Learning Representations*, 2022.

Rachid Guerraoui, Nirupam Gupta, and Rafael Pinot. Byzantine machine learning: A primer. *ACM Computing Surveys*, 56(7):1–39, 2024.

Samuel Horváth, Lihua Lei, Peter Richtárik, and Michael I Jordan. Adaptivity of stochastic gradient methods for nonconvex optimization. *SIAM Journal on Mathematics of Data Science*, 4(2):634–648, 2022.

Xinmeng Huang, Yiming Chen, Wotao Yin, and Kun Yuan. Lower bounds and nearly optimal algorithms in distributed learning with communication compression. In *Advances in Neural Information Processing Systems*, 2022.

Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for Byzantine robust optimization. In *International Conference on Machine Learning*, 2021.

Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. *International Conference on Learning Representations*, 2022.

Sajad Koushkbaghi, Mostafa Safi, Ali Moradi Amani, Mahdi Jalili, and Xinghuo Yu. Byzantine-resilient second-order consensus in networked systems. *IEEE Transactions on Cybernetics*, 54(9):4915–4927, 2024.

Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

Cody Lewis, Vijay Varadharajan, and Nasimul Noman. Attacks against federated learning defense systems and their mitigation. *Journal of Machine Learning Research*, 24(30): 1–50, 2023.

Liping Li, Wei Xu, Tianyi Chen, Georgios B Giannakis, and Qing Ling. RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtárik. Page: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *International Conference on Machine Learning*, 2021.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 2017.

Yanli Liu, Yuan Gao, and Wotao Yin. An improved analysis of stochastic gradient descent with momentum. In *Advances in Neural Information Processing Systems*, 2020.

Yi Liu, Cong Wang, and Xingliang Yuan. Badsampler: Harnessing the power of catastrophic forgetting to poison Byzantine-robust federated learning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024.

Yucheng Lu and Christopher De Sa. Optimal complexity in decentralized training. In *International Conference on Machine Learning*, 2021.

Saeed Mahloujifar, Mohammad Mahmoody, and Ameer Mohammed. Universal multi-party poisoning attacks. In *International Conference on Machine Learning*, 2019.

Arkadi Semenovich Nemirovski and David Borisovich Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience, 1983.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, 2017.

OpenAI et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Jie Peng, Weiyu Li, Stefan Vlaski, and Qing Ling. Mean aggregator is more robust than robust aggregators under label poisoning attacks on distributed heterogeneous data. *Journal of Machine Learning Research*, 26(27):1–51, 2025.

Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Saurav Prakash and Amir Salman Avestimehr. Mitigating Byzantine attacks in federated learning. *arXiv preprint arXiv:2010.07541*, 2020.

Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. Bit-flip attack: Crushing neural network with progressive bit search. In *International Conference on Computer Vision*, 2019.

Sashank J Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

Sheldon M Ross. *Introduction to probability models*. Academic Press, 2014.

Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *International Conference on Machine Learning*, 2017.

Qiankun Shi, Jie Peng, Kun Yuan, Xiao Wang, and Qing Ling. Optimal complexity in Byzantine-robust distributed stochastic optimization with data heterogeneity. *arXiv preprint arXiv:2503.16337*, 2025.

Mengkai Song, Zhibo Wang, Zhifei Zhang, Yang Song, Qian Wang, Ju Ren, and Hairong Qi. Analyzing user-level privacy attack against federated learning. *IEEE Journal on Selected Areas in Communications*, 38(10):2430–2444, 2020.

Lili Su and Nitin H Vaidya. Fault-tolerant multi-agent optimization: optimal iterative distributed algorithms. In *Proceedings of ACM Symposium on Principles of Distributed Computing*, 2016.

Blake E Woodworth and Nati Srebro. Tight complexity bounds for optimizing composite objectives. In *Advances in Neural Information Processing Systems*, 2016.

Zhaoxian Wu, Qing Ling, Tianyi Chen, and Georgios B Giannakis. Federated variance-reduced stochastic gradient descent with robustness to Byzantine attacks. *IEEE Transactions on Signal Processing*, 68:4583–4596, 2020.

Qi Xia, Zeyi Tao, Zijiang Hao, and Qun Li. FABA: an algorithm for fast aggregation against Byzantine attacks in distributed neural networks. In *International Joint Conference on Artificial Intelligence*, 2019.

Qi Xia, Zeyi Tao, and Qun Li. Defenses against Byzantine attacks in distributed deep neural networks. *IEEE Transactions on Network Science and Engineering*, 8(3):2025–2035, 2020.

Han Xiao, Huang Xiao, and Claudia Eckert. Adversarial label flips attack on support vector machines. In *European Conference on Artificial Intelligence*, 2012.

Yiming Xiao, Haidong Shao, Jian Lin, Zhiqiang Huo, and Bin Liu. Bce-fl: A secure and privacy-preserving federated learning system for device fault diagnosis under non-iid condition in IIoT. *IEEE Internet of Things Journal*, 11(8):14241–14252, 2024.

Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Phocas: Dimensional Byzantine-resilient stochastic gradient descent. *arXiv preprint arXiv:1805.09682*, 2018.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *International Conference on Machine Learning*, 2019.

Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation. In *Uncertainty in Artificial Intelligence*, 2020.

Jian Xu, Shao-Lun Huang, Linqi Song, and Tian Lan. Signguard: Byzantine-robust federated learning through collaborative malicious gradient filtering. In *International Conference on Distributed Computing Systems*, 2022.

Yi-Rui Yang and Wu-Jun Li. Buffered asynchronous SGD for Byzantine learning. *Journal of Machine Learning Research*, 24(204):1–62, 2023.

Haoxiang Ye and Qing Ling. On the generalization error of Byzantine-resilient decentralized learning. In *International Conference on Acoustics, Speech and Signal Processing*, 2024.

Haoxiang Ye and Qing Ling. Generalization error matters in decentralized learning under byzantine attacks. *IEEE Transactions on Signal Processing*, (73):843–857, 2025.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, 2018.

Kun Yuan, Xinmeng Huang, Yiming Chen, Xiaohan Zhang, Yingya Zhang, and Pan Pan. Revisiting optimal convergence rate for smooth and non-convex stochastic decentralized optimization. In *Advances in Neural Information Processing Systems*, 2022.

Weishan Zhang, Qinghua Lu, Qiuyu Yu, Zhaotong Li, Yue Liu, Sin Kit Lo, Shiping Chen, Xiwei Xu, and Liming Zhu. Blockchain-based federated learning for device failure detection in industrial IoT. *IEEE Internet of Things Journal*, 8(7):5926–5937, 2020.

Banghua Zhu, Lun Wang, Qi Pang, Shuai Wang, Jiantao Jiao, Dawn Song, and Michael I Jordan. Byzantine-robust federated learning with optimal statistical rates. In *International Conference on Artificial Intelligence and Statistics*, 2023.

Appendix A. Proofs of main results

A.1 Proof of Lemma 4

Proof We prove Lemma 4 through constructing two one-dimensional deterministic problems without any Byzantine nodes, such that any method $\mathsf{M} \in \mathcal{M}$, equipped with a certain $(\delta_{\max}, \rho)$-robust aggregator $\mathsf{A} \in \mathcal{A}$, cannot distinguish the two. Recall that the estimated fraction of the Byzantine nodes δ satisfies $0 < \delta \leq \delta_{\max} < 0.5$. In the following proof, we assume that δn is an integer. Otherwise, the conclusion still holds true if we round down δn in the derivation.

In the first problem, the function and the gradient of node i are respectively defined as

$$f_{1,i}(x) = \begin{cases} \frac{1}{2}x^2 - \delta^{-1/2}\zeta x, & i = 1, \cdots, \delta n, \\ \frac{1}{2}x^2, & i = \delta n + 1, \cdots, n, \end{cases}$$

$$\nabla f_{1,i}(x) = \begin{cases} x - \delta^{-1/2}\zeta, & i = 1, \cdots, \delta n, \\ x, & i = \delta n + 1, \cdots, n, \end{cases}$$

where $x \in \mathbb{R}$. Therefore, we have

$$f_1(x) = \frac{1}{n}\sum_{i=1}^{n} f_{1,i}(x) = \frac{1}{2}x^2 - \delta^{1/2}\zeta x \quad \text{and} \quad \nabla f_1(x) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_{1,i}(x) = x - \delta^{1/2}\zeta.$$

It is easy for us to verify that $f_1 \in \mathcal{F}$. Assumption 1 is obviously satisfied, and Assumption 2 holds from

$$\frac{1}{n}\sum_{i=1}^{n} \|\nabla f_{1,i}(x) - \nabla f_1(x)\|^2 = \delta\zeta^2(\delta^{-1/2} - \delta^{1/2})^2 + (1-\delta)\zeta^2\delta = (1-\delta)\zeta^2 \leq \zeta^2.$$

In the second problem, the function and gradient of node i are respectively defined as

$$f_{2,i}(x) = f_{1,i}(x) + \alpha_{\min}\rho^{1/2}\delta^{1/2}\zeta x, \quad i = 1, \cdots, n,$$

$$\nabla f_{2,i}(x) = \nabla f_{1,i}(x) + \alpha_{\min}\rho^{1/2}\delta^{1/2}\zeta, \quad i = 1, \cdots, n,$$

where $0 < \alpha_{\min} \leq \sum_{j=1}^{t}\sum_{l=1}^{m} \alpha^{(j,l)}$ and $\{\alpha^{(j,l)}\}$ are the weights in (4). Therefore, we have

$$f_2(x) = f_1(x) + \alpha_{\min}\rho^{1/2}\delta^{1/2}\zeta x \quad \text{and} \quad \nabla f_2(x) = \nabla f_1(x) + \alpha_{\min}\rho^{1/2}\delta^{1/2}\zeta.$$

Again, Assumptions 1 and 2 are both satisfied. Observe that the minimizers of f_1 and f_2 are different.

For any method $\mathsf{M} \in \mathcal{M}$, according to the update rule in (4), we obtain $w_{2,i}^t = w_{1,i}^t + \alpha_{\min}\rho^{1/2}\delta^{1/2}\zeta$ and $\bar{w}_2^t = \bar{w}_1^t + \alpha_{\min}\rho^{1/2}\delta^{1/2}\zeta$.

Now, we construct a certain $(\delta_{\max}, \rho)$-robust aggregator $\mathsf{A} \in \mathcal{A}$, which always outputs

$$w^t = \bar{w}_1^t + \frac{\alpha_{\min}}{2}\rho^{1/2}\delta^{1/2}\zeta = \bar{w}_2^t - \frac{\alpha_{\min}}{2}\rho^{1/2}\delta^{1/2}\zeta,$$

given the inputs of either $\{w_{1,i}^t\}$ or $\{w_{2,i}^t\}$. To prove that such an aggregator A is indeed $(\delta_{\max}, \rho)$-robust, we refer to

$$\|w^t - \bar{w}_1^t\|^2 = \frac{\alpha_{\min}^2}{4}\rho\delta\zeta^2 \leq \alpha_{\min}^2\rho\delta\zeta^2(1-\delta) \leq \frac{\rho\delta}{n}\sum_{i=1}^n \|w_{1,i}^t - \bar{w}_1^t\|^2, \tag{28}$$

$$\|w^t - \bar{w}_2^t\|^2 = \frac{\alpha_{\min}^2}{4}\rho\delta\zeta^2 \leq \alpha_{\min}^2\rho\delta\zeta^2(1-\delta) \leq \frac{\rho\delta}{n}\sum_{i=1}^n \|w_{2,i}^t - \bar{w}_2^t\|^2. \tag{29}$$

For the first inequalities, we use the fact that $\delta \leq \delta_{\max} < 0.5$. For the last inequality in (28), we recall that

$$w_{1,i}^t = \sum_{j=1}^t \sum_{l=1}^m \alpha^{(j,l)} \nabla f_{1,i}(x^{(j,l)})$$

and

$$\nabla f_{1,i}(x) - \nabla f_1(x) = \begin{cases} \zeta\delta^{1/2} - \zeta\delta^{-1/2}, & i = 1, \ldots, \delta n, \\ \zeta\delta^{1/2}, & i = \delta n + 1, \ldots, n. \end{cases}$$

Then letting $\alpha = \sum_{j=1}^t \sum_{l=1}^m \alpha^{(j,l)}$, we have

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^n \|w_{1,i}^t - \bar{w}_1^t\|^2 &= \frac{1}{n}\sum_{i=1}^{\delta n} \alpha^2\zeta^2\|\delta^{1/2} - \delta^{-1/2}\|^2 + \frac{1}{n}\sum_{i=\delta n+1}^n \alpha^2\|\zeta\delta^{1/2}\|^2 \\ &= \alpha^2\delta\zeta^2\left(\delta^{1/2} - \delta^{-1/2}\right)^2 + \alpha^2(1-\delta)\zeta^2\delta \\ &= \alpha^2\zeta^2\left[(1-\delta)^2 + (1-\delta)\delta\right] = \alpha^2\zeta^2(1-\delta) \geq \alpha_{\min}^2\zeta^2(1-\delta). \end{aligned}$$

The same derivation holds for the last inequality in (29).

Since the $(\delta_{\max}, \rho)$-robust aggregator $\mathsf{A} \in \mathcal{A}$ yields the same w^t for the two problems, any method $\mathsf{M} \in \mathcal{M}$ shall return the same x^{t+1}. In consequence, any method $\mathsf{M} \in \mathcal{M}$ is unable to distinguish the two problems and the Byzantine error occurs on at least one of them. For any output $\tilde{x}$ that is irrelevant with the number of iterations and the number of oracle queries, we have

$$\begin{aligned} \max_{j\in\{1,2\}} \|\nabla f_j(\tilde{x})\|^2 &\geq \frac{1}{2}\|\nabla f_1(\tilde{x})\|^2 + \frac{1}{2}\|\nabla f_2(\tilde{x})\|^2 \\ &= \frac{1}{2}(\tilde{x} - \zeta\delta^{1/2})^2 + \frac{1}{2}(\tilde{x} - \zeta\delta^{1/2} + \alpha_{\min}\rho^{1/2}\zeta\delta^{1/2})^2 \\ &= (\tilde{x} - \zeta\delta^{1/2})^2 + (\tilde{x} - \zeta\delta^{1/2})\alpha_{\min}\rho^{1/2}\zeta\delta^{1/2} + \frac{\alpha_{\min}^2}{2}\rho\delta\zeta^2 \\ &= (\tilde{x} - \zeta\delta^{1/2} + \frac{\alpha_{\min}}{2}\rho^{1/2}\zeta\delta^{1/2})^2 + \frac{\alpha_{\min}^2}{4}\rho\delta\zeta^2 \geq \frac{\alpha_{\min}^2}{4}\rho\delta\zeta^2. \end{aligned}$$

which, together with $\alpha_{\min} > 0$, establishes the lower bound. ■

A.2 Proof of Lemma 9

Proof We prove Lemma 9 via constructing a one-dimensional stochastic problem without any Byzantine nodes, such that any method $\mathsf{M} \in \mathcal{M}$, equipped with a certain $(\delta_{\max}, \rho)$-robust aggregator $\mathsf{A} \in \mathcal{A}$, will be stuck at x^0 with $\nabla f(x^0) = 2\epsilon$, as long as the number of oracle queries is insufficient. Recall that the estimated fraction of the Byzantine nodes δ satisfies $0 < \delta \leq \delta_{\max} < 0.5$. For simplicity we assume $x^0 = 0$, but the conclusion remains the same for arbitrary x^0.

In the constructed problem, all nodes have identical functions and gradients, given by

$$f_i(x) = f(x) = \frac{L}{2}x^2 + 2\epsilon x = \mathbb{E}_\xi[F(x,\xi) := \frac{L}{2}x^2 + 2\xi x], \quad \forall i = 1, \cdots, n,$$
$$\nabla f_i(x) = \nabla f(x) = Lx + 2\epsilon, \quad \forall i = 1, \cdots, n,$$

where $x \in \mathbb{R}$ and $\xi \in \Xi \subset \mathbb{R}$ is a random vector satisfying $\mathbb{E}_\xi[\xi] = \epsilon$ and $\mathbb{E}_\xi[\|\xi - \epsilon\|^2] = \frac{1}{4}\sigma^2$. In any method $\mathsf{M} \in \mathcal{M}$, each node only has access to the stochastic gradient $\nabla F(x,\xi) = Lx + 2\xi$ from a given oracle. Note that $\|\nabla f(0)\| = 2\epsilon > \epsilon$ and $\nabla F(0,\xi_i) = 2\xi_i$. It is easy to verify that Assumptions 1, 2 and 3 are all satisfied.

Given the initial point $x^0 = 0$, if there exists a $(\delta_{\max}, \rho)$-robust aggregator $\mathsf{A} \in \mathcal{A}$ always returning 0, then the output $\tilde{x}$ of any method $\mathsf{M} \in \mathcal{M}$ is also 0 such that $\|\nabla f(\tilde{x})\| = 2\epsilon > \epsilon$. To avoid such an undesired circumstance, the inequality (3) in Definition 2 should not hold, or equivalently we must have

$$\|0 - \bar{w}\|^2 > \frac{\rho\delta}{n}\sum_{i=1}^n \|w_i - \bar{w}\|^2, \tag{30}$$

where $\bar{w} = \frac{1}{n}\sum_{i=1}^n \sum_{l=1}^m \alpha^l \nabla F(0,\xi_i^l)$ and $w_i = \sum_{l=1}^m \alpha^l \nabla F(0,\xi_i^l)$. Here we omit the iteration index t for simplicity. Taking expectations on both sides of (30) yields

$$\begin{aligned}
\mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^n\sum_{l=1}^m \alpha^l \nabla F(0,\xi_i^l)\right)^2\right] &> \mathbb{E}\left[\frac{\rho\delta}{n}\sum_{i=1}^n\left(\sum_{l=1}^m \alpha^l \nabla F(0,\xi_i^l) - \frac{1}{n}\sum_{i=1}^n\sum_{l=1}^m \alpha^l \nabla F(0,\xi_i^l)\right)^2\right] \\
&= \frac{\rho\delta(n-1)}{n}\mathbb{E}\left[\left(2\sum_{l=1}^m \alpha^l \xi_i^l - 2\epsilon\sum_{l=1}^m \alpha^l\right)^2\right] \\
&= \frac{\rho\delta(n-1)\sigma^2}{n}\sum_{l=1}^m (\alpha^l)^2,
\end{aligned} \tag{31}$$

where the first equality comes from the relation between total variance and sample variance (see Chapter 2.6.1 in (Ross, 2014)) and the second is due to the independence of ξ_i^l.

For the L.H.S. of (31), we have

$$\begin{aligned}
&\mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^n\sum_{l=1}^m \alpha^l \nabla F(0,\xi_i^l)\right)^2\right] \\
&= \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^n\sum_{l=1}^m \alpha^l \nabla F(0,\xi_i^l) - \frac{1}{n}\sum_{i=1}^n\sum_{l=1}^m \alpha^l \nabla f(0) + \sum_{l=1}^m \alpha^l \nabla f(0)\right)^2\right]
\end{aligned} \tag{32}$$

$$
\begin{aligned}
&= \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{m}\alpha^l\nabla F(0,\xi_i^l) - \frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{m}\alpha^l\nabla f(0)\right)^2\right] + \left(\sum_{l=1}^{m}\alpha^l\nabla f(0)\right)^2 \\
&= \frac{1}{n^2}\mathbb{E}\left[\sum_{i=1}^{n}\sum_{l=1}^{m}(\alpha^l)^2\left(\nabla F(0,\xi_i^l) - \nabla f(0)\right)^2\right] + \left(\sum_{l=1}^{m}\alpha^l\nabla f(0)\right)^2 \\
&\leq \frac{\sigma^2}{n}\sum_{l=1}^{m}(\alpha^l)^2 + 4\epsilon^2\left(\sum_{l=1}^{m}\alpha^l\right)^2.
\end{aligned}
$$

Substituting (32) into (31) and reorganizing the terms, we obtain

$$
\frac{\left(\sum_{l=1}^{m}\alpha^l\right)^2}{\sum_{l=1}^{m}(\alpha^l)^2} > \frac{\rho\delta(n-1)\sigma^2}{4\epsilon^2 n} - \frac{\sigma^2}{4\epsilon^2 n}.
$$

Then from the Cauchy-Schwarz inequality, we have

$$
m \geq \frac{\left(\sum_{l=1}^{m}\alpha^l\right)^2}{\sum_{l=1}^{m}(\alpha^l)^2} > \frac{\rho\delta(n-1)\sigma^2}{4\epsilon^2 n} - \frac{\sigma^2}{4\epsilon^2 n} = \Omega\left(\frac{\rho\delta\sigma^2}{\epsilon^2}\right),
$$

which implies that for any method $\mathsf{M} \in \mathcal{M}$, the number of oracle queries must be at least $\Omega(\frac{\rho\delta\sigma^2}{\epsilon^2})$ to obtain a $(0,\epsilon)$-stationary point. This completes the proof. ■

A.3 Proof of Lemma 10

Proof We prove Lemma 10 through constructing a d-dimensional stochastic problem without any Byzantine nodes, where $d = \Theta(\epsilon^{-2})$. The function has a chain-like structure, and $\|\nabla f(x)\| > \epsilon$ if $[x]_d = 0$. Given this problem, any method $\mathsf{M} \in \mathcal{M}$, equipped with a certain $(\delta_{\max}, \rho)$-robust aggregator $\mathsf{A} \in \mathcal{A}$, must query $\Omega(\frac{\rho\delta\sigma^2}{\epsilon^2})$ times in expectation to identify the next coordinate. Thus, given $x^0 = \mathbf{0}$, the overall oracle query complexity is $\Omega(\frac{\rho\delta\sigma^2}{\epsilon^4})$. Recall that the estimated fraction of the Byzantine nodes δ satisfies $0 < \delta \leq \delta_{\max} < 0.5$. For simplicity we assume $x^0 = \mathbf{0}$, but the conclusion remains the same for arbitrary x^0.

In the constructed problem, all nodes have identical functions, given by

$$
f(x) = f_i(x) = \frac{L\nu^2}{152}h\left(\frac{x}{\nu}\right) \quad \text{with} \quad \nu = \frac{152}{L}\cdot 2\epsilon, \quad \forall i = 1, \cdots, n,
$$

Therein, $h : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$
h(x) := -\Psi(1)\Phi([x]_1) + \sum_{j=2}^{d}\left[\Psi(-[x]_{j-1})\Phi(-[x]_j) - \Psi([x]_{j-1})\Phi([x]_j)\right] \quad \text{with} \quad d = \left\lfloor\frac{L\Delta}{7296\epsilon^2}\right\rfloor,
$$

while $\Psi : \mathbb{R} \to \mathbb{R}$ and $\Phi : \mathbb{R} \to \mathbb{R}$ are defined as

$$
\Psi(a) = \begin{cases} 0, & a \leq 1/2, \\ \exp\left(1 - \frac{1}{(2a-1)^2}\right), & a > 1/2, \end{cases} \quad \text{and} \quad \Phi(a) = \sqrt{e}\int_{-\infty}^{a} e^{-\frac{\tau^2}{2}}d\tau.
$$

According to (Arjevani et al., 2023), f_i is L-smooth and satisfies Assumption 1. In addition, Assumption 2 obviously holds.

Below, we construct a stochastic gradient oracle $\nabla F(x,\xi)$ that satisfies Assumption 3 for $f(x)$, in the form of

$$\nabla F(x,\xi) = \frac{L\nu}{152} \cdot \nabla H\left(\frac{x}{\nu}, \xi\right) = 2\epsilon \cdot \nabla H\left(\frac{x}{\nu}, \xi\right), \tag{33}$$

where

$$\nabla_j H\left(\frac{x}{\nu}, \xi\right) := \nabla_j h(\frac{x}{\nu}) \cdot \left(1 + \mathbb{1}\{j > \mathrm{prog}_{\frac{1}{2}}(\frac{x}{\nu})\}\left(\frac{\xi}{p} - 1\right)\right). \tag{34}$$

Here, $\mathbb{1}$ is the indicator function that returns 1 if the argument holds true and 0 otherwise; the random variable $\xi \sim \text{Bernoulli}(p)$; $\mathrm{prog}_{\frac{1}{2}}(\frac{x}{\nu}) := \max\{j = 0, 1, \cdots, d \mid |\frac{[x]_j}{\nu}| > \frac{1}{2}\}$ denotes the largest index of $|\frac{x}{\nu}|$ whose element is larger than $\frac{1}{2}$ – we additionally define $|\frac{[x]_0}{\nu}| = 1$ so as to return index 0 if no other elements are qualified. Note that the constant $\frac{1}{2}$ can be replaced by any other constant larger than 0. According to Lemma 3 in (Arjevani et al., 2023), we can show that such a stochastic gradient oracle satisfies Assumption 3, as

$$\mathbb{E}\left[\nabla F\left(x,\xi\right)\right] = \nabla f(x) \quad \text{and} \quad \mathbb{E}\left[\|\nabla F\left(x,\xi\right) - \nabla f(x)\|^2\right] \le \sigma^2,$$

when $\frac{1}{p} = \frac{\sigma^2}{2116\epsilon^2} + 1$.

Recall that x is initialized as $\mathbf{0}$. Now we construct a $(\delta_{\max}, \rho)$-robust aggregator $\mathsf{A} \in \mathcal{A}$ with which the coordinates of x are updated sequentially, from 1 to d. When $x = \mathbf{0}$, the stochastic gradient oracle defined in (33) returns 0 except for the first coordinate. Thus, for any method $\mathsf{M} \in \mathcal{M}$, the elements of all $\{w_i\}$ are 0 except for the first coordinate according to (4). Therefore, if there exists a $(\delta_{\max}, \rho)$-robust aggregator $\mathsf{A} \in \mathcal{A}$ returning 0 for the first coordinate but the true averages for the other coordinates, the aggregated w remain $\mathbf{0}$. In consequence, the variable x and the output $\tilde{x}$ are also $\mathbf{0}$. This is an undesired circumstance since $\|\nabla f(x)\| > \epsilon$ if $[x]_d = 0$. To avoid this circumstance and discover the first coordinate, we must have

$$\|0 - [\bar{w}]_1\|^2 > \frac{\rho\delta}{n}\sum_{i=1}^{n}\|[w_i]_1 - [\bar{w}]_1\|^2.$$

Likewise, given the currently undiscovered coordinate $j \in \{1, \cdots, d\}$, such that $[x]_{j'} = 0$ for $j' \ge j$, the stochastic gradient oracle defined in (33) returns 0 for the coordinates $j' > j$. Thus, for any method $\mathsf{M} \in \mathcal{M}$, the elements of all $\{w_i\}$ are 0 for the coordinates $j' > j$ according to (4). Therefore, if there exists a $(\delta_{\max}, \rho)$-robust aggregator $\mathsf{A} \in \mathcal{A}$ returning 0 for the j-th coordinate but the true averages for the coordinates $j' \ne j$, the aggregated $[w]_j$ remain 0 for the coordinates $j' \ge j$. In consequence, $[x]_{j'}$ and $[\tilde{x}]_{j'}$ are also 0 for the coordinates $j' \ge j$. As discussed before, this is an undesired circumstance since $\|\nabla f(x)\| > \epsilon$ if $[x]_d = 0$. To avoid this circumstance and discover the j-th coordinate, we must have

$$\|0 - [\bar{w}]_j\|^2 > \frac{\rho\delta}{n}\sum_{i=1}^{n}\|[w_i]_j - [\bar{w}]_j\|^2, \quad j = 1, \cdots, d. \tag{35}$$

Otherwise, the method $\mathsf{M} \in \mathcal{M}$ shall get stuck in the j-th coordinate.

By definition $[\bar{w}]_j = \frac{1}{n}\sum_{i=1}^n \sum_{l=1}^m \alpha^l \nabla_j F(x^l, \xi_i^l)$ and $[w_i]_j = \sum_{l=1}^m \alpha^l \nabla_j F(x^l, \xi_i^l)$ in (35). Here and thereafter, we omit the iteration index t for simplicity. Observe that $\nabla_j F(x^l, \xi_i^l) \neq 0$ only if $[x^l]_{j-1} \neq 0$. We denote the number of the sampled $\{x^l\}_{l=1}^m$ satisfying $[x^l]_{j-1} \neq 0$ as m_j and suppose that they are $\{x^l\}_{l=m-m_j}^m$. Thus, $[\bar{w}]_j = \frac{1}{n}\sum_{i=1}^n \sum_{l=m-m_j}^m \alpha^l \nabla_j F(x^l, \xi_i^l)$ and $[w_i]_j = \sum_{l=m-m_j}^m \alpha^l \nabla_j F(x^l, \xi_i^l)$. Note that because $\sum_{j=1}^t \sum_{l=1}^m \alpha^{(j,l)} \geq \alpha_{\min} > 0$, there exists at least one $\alpha_l \neq 0$ for $l = m - m_j, \cdots, m$. Substituting $[\bar{w}]_j$ and $[w_i]_j$ into (35) and taking expectations, we have

$$\begin{aligned} &\mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^n \sum_{l=m-m_j}^m \alpha^l \nabla_j F(x^l, \xi_i^l)\right)^2\right] \qquad (36) \\ >\ &\mathbb{E}\left[\frac{\rho\delta}{n}\sum_{i=1}^n \left(\sum_{l=m-m_j}^m \alpha^l \nabla_j F(x^l, \xi_i^l) - \frac{1}{n}\sum_{i=1}^n \sum_{l=m-m_j}^m \alpha^l \nabla_j F(x^l, \xi_i^l)\right)^2\right]. \end{aligned}$$

For the L.H.S. of (36), we have

$$\begin{aligned} &\mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^n \sum_{l=m-m_j}^m \alpha^l \nabla_j F(x^l, \xi_i^l)\right)^2\right] \\ =\ &\mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^n \sum_{l=m-m_j}^m \alpha^l \nabla_j F(x^l, \xi_i^l) - \frac{1}{n}\sum_{i=1}^n \sum_{l=m-m_j}^m \alpha^l \nabla_j f(x^l) + \sum_{l=m-m_j}^m \alpha^l \nabla_j f(x^l)\right)^2\right] \\ =\ &\mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^n \sum_{l=m-m_j}^m \alpha^l \nabla_j F(x^l, \xi_i^l) - \frac{1}{n}\sum_{i=1}^n \sum_{l=m-m_j}^m \alpha^l \nabla_j f(x^l)\right)^2\right] + \left(\sum_{l=m-m_j}^m \alpha^l \nabla_j f(x^l)\right)^2 \\ =\ &\frac{1}{n^2}\mathbb{E}\left[\sum_{i=1}^n \sum_{l=m-m_j}^m (\alpha^l)^2 \left(\nabla_j F(x^l, \xi_i^l) - \nabla_j f(x^l)\right)^2\right] + \left(\sum_{l=m-m_j}^m \alpha^l \nabla_j f(x^l)\right)^2 \\ =\ &\frac{\sum_{l=m-m_j}^m \left(\alpha^l \nabla_j f(x^l)\right)^2}{n}\frac{1-p}{p} + \left(\sum_{l=m-m_j}^m \alpha^l \nabla_j f(x^l)\right)^2, \end{aligned}$$

where the second equality comes from $\mathbb{E}[\nabla F(x, \xi)] = \nabla f(x)$, the third equality is due to the independence of ξ_i^l and the last equality follows from

$$\mathbb{E}\left[\left(\nabla_j F(x^l, \xi_i^l) - \nabla_j f(x^l)\right)^2\right] = (\nabla_j f(x^l))^2 \frac{\mathbb{E}[(\xi_i^l - p)^2]}{p^2} = (\nabla_j f(x^l))^2 \frac{1-p}{p}.$$

For the R.H.S. of (36), we have

$$\mathbb{E}\left[\frac{\rho\delta}{n}\sum_{i=1}^n \left(\sum_{l=m-m_j}^m \alpha^l \nabla_j F(x^l, \xi_i^l) - \frac{1}{n}\sum_{i=1}^n \sum_{l=m-m_j}^m \alpha^l \nabla_j F(x^l, \xi_i^l)\right)^2\right]$$

$$
\begin{aligned}
&= \frac{\rho\delta(n-1)}{n}\mathbb{E}\left[\left(\sum_{l=m-m_j}^{m}\alpha^l\nabla_j F(x^l,\xi_i^l)-\sum_{l=m-m_j}^{m}\alpha^l\nabla_j f(x^l)\right)^2\right]\\
&= \frac{\rho\delta(n-1)}{n}\mathbb{E}\left[\sum_{l=m-m_j}^{m}(\alpha^l)^2\left(\nabla_j F(x^l,\xi_i^l)-\nabla_j f(x^l)\right)^2\right]\\
&= \frac{\rho\delta(n-1)\sum_{l=m-m_j}^{m}\left(\alpha^l\nabla_j f(x^l)\right)^2}{n}\frac{1-p}{p},
\end{aligned}
$$

where the first equality uses the relation between total variance and sample variance and the second is also due to the independence of ξ_i^l.

Therefore, (36) becomes

$$
\frac{\left(\sum_{l=m-m_j}^{m}\alpha^l\nabla_j f(x^l)\right)^2}{\sum_{l=m-m_j}^{m}\left(\alpha^l\nabla_j f(x^l)\right)^2} > \frac{\rho\delta(n-1)}{n}\frac{1-p}{p}-\frac{1}{n}\frac{1-p}{p}.
$$

Since $\frac{1-p}{p}=\frac{\sigma^2}{2116\epsilon^2}$, we have

$$
m_j > \frac{\rho\delta\sigma^2}{2116\epsilon^2}-\frac{\rho\delta\sigma^2}{2116\epsilon^2 n}-\frac{\sigma^2}{2116\epsilon^2 n}=\Omega(\frac{\rho\delta\sigma^2}{\epsilon^2}),\ \forall j=1,\cdots,d.
$$

Recall that we need to sequentially update to the d-th coordinate with $d=\Omega(L\Delta\epsilon^{-2})$, the overall oracle query complexity is $\sum_{j=1}^{d}m_j=\Omega(\frac{L\Delta\rho\delta\sigma^2}{\epsilon^4})$. ■

A.4 Proof of Lemma 12

Proof We begin with introducing several auxiliary sequences. The first sequence is

$$
\bar{x}^t := x^t+(\sqrt{q}-1)(x^t-x^{t-1}),\quad \forall t=1,2,\cdots, \tag{37}
$$

with $\bar{x}^0=x^0$. According to the definition of β in (17) and the definition of $\bar{x}^t$ in (37), we further introduce

$$
\begin{aligned}
y^t :=& x^t+\beta(x^t-x^{t-1})\\
=& x^t+\frac{\sqrt{q}-1}{\sqrt{q}+1}(x^t-x^{t-1})\\
=& x^t+\frac{\sqrt{q}-1}{q-1}\cdot(\sqrt{q}-1)(x^t-x^{t-1})\\
=& \frac{\sqrt{q}-1}{q-1}\bar{x}^t+(1-\frac{\sqrt{q}-1}{q-1})x^t,\quad \forall t=1,2,\cdots.
\end{aligned}
$$

We also introduce

$$
z^{t-1} := \frac{x^*}{\sqrt{q}}+(1-\frac{1}{\sqrt{q}})x^{t-1},\quad \forall t=1,2,\cdots.
$$

From the $\frac{L}{\theta}$-strong convexity of h^t, we have

$$\begin{aligned} h^t(z^{t-1}) \geq & h^{t*} + \frac{L}{2\theta}\|z^{t-1} - x^{t*}\|^2 \\ = & h^{t*} + \frac{L}{2\theta}\|z^{t-1} - x^t\|^2 + \underbrace{\frac{L}{2\theta}\|x^t - x^{t*}\|^2 + \frac{L}{\theta}\langle z^{t-1} - x^t, x^t - x^{t*}\rangle}_{-\Pi^t}. \end{aligned} \tag{38}$$

Therefore, we have

$$\begin{aligned} \mathbb{E}[f(x^t)] &\leq \mathbb{E}[h^{t*}] + \upsilon^t \\ &\leq \mathbb{E}[h^t(z^{t-1})] - \mathbb{E}\left[\frac{L}{2\theta}\|z^{t-1} - x^t\|^2\right] + \mathbb{E}[\Pi^t] + \upsilon^t \\ &\leq \mathbb{E}[f(z^{t-1})] + \mathbb{E}\left[\frac{L - \mu\theta}{2\theta}\|z^{t-1} - y^{t-1}\|^2\right] - \mathbb{E}\left[\frac{L}{2\theta}\|z^{t-1} - x^t\|^2\right] + \mathbb{E}[\Pi^t] + \upsilon^t, \end{aligned} \tag{39}$$

where the first and last inequalities are respectively due to (ii) and (i) in the hypothesis, while the second inequality comes from taking expectation on (38). Note that the expectation is conditioned on all random variables prior to $t-1$.

Now we bound the first three terms at the R.H.S. of (39). We have

$$\begin{aligned} \mathbb{E}[f(z^{t-1})] &\leq \frac{1}{\sqrt{q}} f^* + (1 - \frac{1}{\sqrt{q}})\mathbb{E}[f(x^{t-1})] - \frac{\mu(\sqrt{q}-1)}{2q} R^{t-1}, \\ \mathbb{E}[\|z^{t-1} - y^{t-1}\|^2] &\leq \frac{1}{\sqrt{q}} \cdot \frac{R^{t-1}}{q + \sqrt{q}} + \frac{1}{\sqrt{q}} \cdot \frac{\bar{R}^{t-1}}{\sqrt{q}+1}, \\ \mathbb{E}[\|z^{t-1} - x^t\|^2] &= \frac{\bar{R}^t}{q}, \end{aligned} \tag{40}$$

where $R^t = \mathbb{E}[\|x^* - x^t\|^2]$ and $\bar{R}^t = \mathbb{E}[\|x^* - \bar{x}^t\|^2]$. The first inequality in (40) is due to the strong convexity of f, the last one can be obtained from the definition of $\bar{x}^t$ in (37) after simple derivation, and the second one can be derived through

$$\begin{aligned} \|z^{t-1} - y^{t-1}\|^2 &= \left\| \frac{x^* - x^{t-1}}{q + \sqrt{q}} + \frac{x^* - \bar{x}^{t-1}}{\sqrt{q}+1} \right\|^2 \\ &= \frac{1}{q}\left\| \left(1 - \frac{\sqrt{q}}{\sqrt{q}+1}\right)(x^* - x^{t-1}) + \frac{\sqrt{q}}{\sqrt{q}+1}(x^* - \bar{x}^{t-1}) \right\|^2 \\ &\leq \frac{1}{q} \cdot \left(1 - \frac{\sqrt{q}}{\sqrt{q}+1}\right)\|x^* - x^{t-1}\|^2 + \frac{1}{q} \cdot \frac{\sqrt{q}}{\sqrt{q}+1}\|x^* - \bar{x}^{t-1}\|^2 \\ &= \frac{1}{\sqrt{q}} \cdot \frac{1}{q + \sqrt{q}}\|x^* - x^{t-1}\|^2 + \frac{1}{\sqrt{q}} \cdot \frac{1}{\sqrt{q}+1}\|x^* - \bar{x}^{t-1}\|^2. \end{aligned}$$

Substituting (40) into (39) and using the fact that $q = \frac{L}{\theta\mu}$, we have

$$\mathbb{E}[f(x^t) - f^*] + \frac{\mu}{2}\bar{R}^t \leq (1 - \frac{1}{\sqrt{q}})\mathbb{E}[f(x^{t-1}) - f^*] - \frac{\mu(\sqrt{q}-1)}{2q} R^{t-1} \tag{41}$$

$$+\frac{L-\mu\theta}{2\theta}\frac{1}{\sqrt{q}}\cdot\frac{R^{t-1}}{q+\sqrt{q}}+\frac{L-\mu\theta}{2\theta}\frac{1}{\sqrt{q}}\cdot\frac{\bar{R}^{t-1}}{\sqrt{q}+1}+\mathbb{E}[\Pi^t]+\upsilon^t$$
$$=(1-\frac{1}{\sqrt{q}})\mathbb{E}[f(x^{t-1})-f^*]+\frac{L-\mu\theta}{2\theta}\frac{1}{\sqrt{q}}\cdot\frac{\bar{R}^{t-1}}{\sqrt{q}+1}+\mathbb{E}[\Pi^t]+\upsilon^t.$$

Further defining a Lyapunov function

$$S^t=(1-\frac{1}{\sqrt{q}})\mathbb{E}[f(x^t)-f^*]+\frac{L-\mu\theta}{2\theta}\frac{1}{\sqrt{q}}\cdot\frac{\bar{R}^t}{\sqrt{q}+1},\quad t=1,2,\cdots, \tag{42}$$

from (41) we have

$$(1-\frac{1}{\sqrt{q}})^{-1}S^t\le S^{t-1}+\mathbb{E}[\Pi^t]+\upsilon^t. \tag{43}$$

For the term $\mathbb{E}[\Pi^t]$ in (43), we know that

$$\begin{aligned}\Pi^t&=-\frac{L}{2\theta}\|x^t-x^{t*}\|^2-\frac{L}{\theta}\langle z^{t-1}-x^t,x^t-x^{t*}\rangle \\ &=-\frac{L}{2\theta}\|x^t-x^{t*}\|^2-\frac{L}{\theta\sqrt{q}}\langle x^*-\bar{x}^t,x^t-x^{t*}\rangle \\ &\le-\frac{L}{2\theta}\|x^t-x^{t*}\|^2+\frac{L}{\theta\sqrt{q}}\|x^*-\bar{x}^t\|\|x^t-x^{t*}\| \\ &\le(2\sqrt{q}-1)\,\frac{L}{2\theta}\|x^t-x^{t*}\|^2+\frac{L}{4\theta q\sqrt{q}}\|x^*-\bar{x}^t\|^2\ \ \text{(using Young's inequality)} \\ &\le(2\sqrt{q}-1)\,(h^t(x^t)-h^{t*})+\frac{L}{4\theta q\sqrt{q}}\|x^*-\bar{x}^t\|^2\ \ \text{(using } 2\sqrt{q}\ge1\text{ and strong convexity)} \\ &=(2\sqrt{q}-1)\,(h^t(x^t)-h^{t*})+\frac{L-\mu\theta}{4\theta\sqrt{q}(q-1)}\|x^*-\bar{x}^t\|^2.\end{aligned} \tag{44}$$

Taking expectation on (44), noticing that the quadratic term involving $\|x^*-\bar{x}^t\|^2$ is smaller than $S^t/2(\sqrt{q}-1)$ in expectation (from the definition of S^t in (42)) and using (iii) in the hypothesis, we obtain

$$\mathbb{E}[\Pi^t]\le(2\sqrt{q}-1)\,\varepsilon^t+\frac{S^t}{2(\sqrt{q}-1)}. \tag{45}$$

Substituting (45) into (43), we have

$$S^t\le\frac{2\sqrt{q}-2}{2\sqrt{q}-1}\left(S^{t-1}+\upsilon^t+(2\sqrt{q}-1)\,\varepsilon^t\right).$$

Unrolling the recursion yields

$$\begin{aligned}S^t&\le\left(\frac{2\sqrt{q}-2}{2\sqrt{q}-1}\right)^t\left(S^0+\sum_{\tau=1}^t\left(\frac{2\sqrt{q}-1}{2\sqrt{q}-2}\right)^{\tau-1}(\upsilon^\tau-\varepsilon^\tau+2\sqrt{q}\varepsilon^\tau)\right) \\ &=\left(1-\frac{1}{2\sqrt{q}-1}\right)^tS^0+\sum_{\tau=1}^t\left(1-\frac{1}{2\sqrt{q}-1}\right)^{t-\tau+1}(\upsilon^\tau-\varepsilon^\tau+2\sqrt{q}\varepsilon^\tau)\end{aligned} \tag{46}$$

$$\leq (1-\frac{1}{\sqrt{q}})\left(2\left(1-\frac{1}{2\sqrt{q}}\right)^t(f(x^0)-f^*)+2\sum_{\tau=1}^{t}\left(1-\frac{1}{2\sqrt{q}}\right)^{t-\tau}(\upsilon^\tau-\varepsilon^\tau+2\sqrt{q}\varepsilon^\tau)\right)$$
$$\leq (1-\frac{1}{\sqrt{q}})\left(2\left(1-\frac{1}{2\sqrt{q}}\right)^t(f(x^0)-f^*)+4\sum_{\tau=1}^{t}\left(1-\frac{1}{2\sqrt{q}}\right)^{t-\tau}(\upsilon^\tau+\sqrt{q}\varepsilon^\tau)\right),$$

where the second inequality uses the fact that $1-\frac{1}{2\sqrt{q}-1}\leq 2(1-\frac{1}{\sqrt{q}})$ and

$$\begin{aligned}S^0&=(1-\frac{1}{\sqrt{q}})(f(x^0)-f^*)+\frac{L-\mu\theta}{2\theta\sqrt{q}(\sqrt{q}+1)}\|x^0-x^*\|^2\\&=(1-\frac{1}{\sqrt{q}})(f(x^0)-f^*)+\frac{\mu}{2}(1-\frac{1}{\sqrt{q}})\|x^0-x^*\|^2\\&\leq 2(1-\frac{1}{\sqrt{q}})(f(x^0)-f^*).\end{aligned}$$

From the definition of S^t in (42), we have $(1-\frac{1}{\sqrt{q}})\mathbb{E}[f(x^t)-f^*]\leq S^t$. Combining this inequality with (46), we obtain (18) and complete the proof of Lemma 12. ■

A.5 Bound of aggregation bias $\mathbb{E}[\|\Delta_1^t\|^2]$

Lemma 20 *Consider Algorithm 1 with the step size $\eta=\frac{1}{L}$. If Assumptions 2 and 3 hold, while $\alpha\in[0,1]$, $\beta\in[0,1)$ and $\theta\in(0,1]$ satisfy*

$$\begin{aligned}&\chi_0\beta^2=(1-\alpha)\beta(\theta+\beta),\\&\chi_1\geq\frac{1}{1-\chi_0\beta^2}\frac{\alpha\theta^2}{1-\beta^2}\geq 0,\\&\chi_2\geq\frac{(1-\alpha)\theta(\theta+\beta)}{1-\chi_0\beta^2}\geq 0,\\&\chi_3\geq\frac{1}{1-\chi_0\beta^2}\frac{\alpha\theta(\theta+\beta)}{1-\beta(\theta+\beta)}\geq 0,\\&\chi_4\geq 2\alpha\beta^2(\chi_1+\chi_2+\frac{\alpha\beta^2 m}{(1-\beta^2)m_0})+\frac{2\alpha\theta^2}{1-\beta^2}+(1-\alpha)\theta^2,\\&\chi_5\geq 2\alpha\beta^2(\chi_2+\chi_3+1)+\frac{2\alpha\theta(\theta+\beta)}{1-\beta(\theta+\beta)}+(1-\alpha)\theta^2,\end{aligned}\tag{47}$$

with some χ_1, χ_2, χ_3, χ_4, $\chi_5\geq 0$, then for any robust aggregator $\mathsf{A}\in\mathcal{A}$, we have

$$\mathbb{E}[\|\Delta_1^t\|^2]\leq\frac{1}{L^2}\left(\frac{3\chi_4\rho\delta\sigma^2}{m}\left(1+\frac{1}{(1-\delta)n}\right)+3\chi_5\rho\delta\zeta^2\right).\tag{48}$$

Proof Using Young's inequality and $\eta=\frac{1}{L}$, we have

$$\|\Delta_1^t\|^2=\frac{1}{L^2}\left\|\beta(\hat{s}^{t-1}-\bar{s}^{t-1})+(\bar{s}^t-\hat{s}^t)\right\|^2\tag{49}$$

$$= \frac{1}{L^2}\left\|\alpha\beta\hat{s}^{t-1} + \frac{\theta}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} g_i^{t-1} - \alpha\mathsf{A}(\{s_i^t\}_{i=1}^n) - (1-\alpha)\theta\mathsf{A}(\{g_i^{t-1}\}_{i=1}^n)\right\|^2$$
$$\leq \frac{\alpha}{L^2}\left\|\beta\hat{s}^{t-1} + \frac{\theta}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} g_i^{t-1} - \mathsf{A}(\{s_i^t\}_{i=1}^n)\right\|^2 + \frac{(1-\alpha)\theta^2}{L^2}\left\|\mathsf{A}(\{g_i^{t-1}\}_{i=1}^n) - \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} g_i^{t-1}\right\|^2$$
$$\leq \frac{2\alpha\beta^2}{L^2}\left\|\hat{s}^{t-1} - \bar{s}^{t-1}\right\|^2 + \frac{2\alpha}{L^2}\left\|\bar{s}^t - \mathsf{A}(\{s_i^t\}_{i=1}^n)\right\|^2 + \frac{(1-\alpha)\theta^2}{L^2}\left\|\mathsf{A}(\{g_i^{t-1}\}_{i=1}^n) - \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} g_i^{t-1}\right\|^2.$$

Taking expectation on the first term, we have

$$\begin{aligned}\mathbb{E}[\left\|\hat{s}^t - \bar{s}^t\right\|^2] &= \mathbb{E}[\| - \alpha(\bar{s}^t - \mathsf{A}(\{s_i^t\}_{i=1}^n)) + (1-\alpha)(s^t - \bar{s}^t)\|^2] \\ &\leq \mathbb{E}[\alpha\|\bar{s}^t - \mathsf{A}(\{s_i^t\}_{i=1}^n)\|^2 + (1-\alpha)\|s^t - \bar{s}^t\|^2] \\ &\leq \mathbb{E}[\alpha\|\bar{s}^t - \mathsf{A}(\{s_i^t\}_{i=1}^n)\|^2 + (1-\alpha)\beta(\theta+\beta)\|\hat{s}^{t-1} - \bar{s}^{t-1}\|^2 \\ &\quad + (1-\alpha)\theta(\theta+\beta)\|\mathsf{A}(\{g_i^{t-1}\}_{i=1}^n) - \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} g_i^{t-1}\|^2].\end{aligned} \tag{50}$$

We first consider the second term at the R.H.S. of (49), which is also the first term at the R.H.S. of (50). From Definition 2 of the $(\delta_{\max}, \rho)$-robust aggregator A, we have

$$\mathbb{E}[\left\|\bar{s}^t - \mathsf{A}(\{s_i^t\}_{i=1}^n)\right\|^2] \leq \frac{\rho\delta}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\mathbb{E}[\left\|\bar{s}^t - s_i^t\right\|^2]. \tag{51}$$

We proceed to bound $\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\mathbb{E}[\|\bar{s}^t - s_i^t\|^2]$. Letting $\mathbb{E}_{\xi^{[t]}} := \mathbb{E}_{\xi^t,\xi^{t-1},\cdots,\xi^0}$, from the definition of s_i^t in (11), we obtain

$$\begin{aligned}\mathbb{E}_{\xi^{[t-1]}}[\|s_i^t - \mathbb{E}_{\xi^{[t-1]}}[s_i^t]\|^2] &= \mathbb{E}_{\xi^{[t-2]}}\mathbb{E}_{\xi^{t-1}}[\|\beta(s_i^{t-1} - \mathbb{E}_{\xi^{[t-2]}}[s_i^{t-1}]) + \theta(g_i^{t-1} - \nabla f_i(y^{t-1})\|^2] \\ &\leq \mathbb{E}_{\xi^{[t-2]}}[\|\beta(s_i^{t-1} - \mathbb{E}_{\xi^{[t-2]}}[s_i^{t-1}])\|^2] + \frac{\theta^2}{m}\sigma^2,\end{aligned}$$

where the cross term $\mathbb{E}_{\xi^{t-1}}[\beta\theta\langle s_i^{t-1} - \mathbb{E}_{\xi^{[t-2]}}[s_i^{t-1}], g_i^{t-1} - \nabla f_i(y^{t-1}\rangle] = 0$ due to the unbiasedness of g_i^{t-1}. Unrolling the above recursion yields

$$\mathbb{E}[\|s_i^t - \mathbb{E}[s_i^t]\|^2] \leq \frac{\theta^2}{(1-\beta^2)m}\sigma^2 + \beta^{2t}\mathbb{E}[\|s_i^0 - \mathbb{E}[s_i^0]\|^2] \leq \frac{\theta^2}{(1-\beta^2)m}\sigma^2 + \frac{\beta^{2t}}{m_0}\sigma^2,$$

where the last inequality is due to $s_i^0 = \frac{1}{m_0}\sum_{l=1}^{m_0}\nabla F(y^0, \xi_i^{(0,l)})$. Hence, for $\bar{s}^t = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} s_i^t$, we have

$$\begin{aligned}\mathbb{E}[\|\bar{s}^t - \mathbb{E}[\bar{s}^t]\|^2] &\leq \frac{\theta^2}{(1-\beta^2)(1-\delta)nm}\sigma^2 + \frac{\beta^{2t}}{(1-\delta)n}\mathbb{E}[\|s_i^0 - \mathbb{E}[s_i^0]\|^2] \\ &\leq \frac{\theta^2}{(1-\beta^2)(1-\delta)nm}\sigma^2 + \frac{\beta^{2t}}{(1-\delta)nm_0}\sigma^2.\end{aligned}$$

Then for $\mathbb{E}[\|\mathbb{E}[\bar{s}^t] - \mathbb{E}[s_i^t]\|^2]$, it holds that

$$\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}[\left\|\mathbb{E}[\bar{s}^t] - \mathbb{E}[s_i^t]\right\|^2] = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}[\|\beta(\mathbb{E}[\bar{s}^{t-1}] - \mathbb{E}[s_i^{t-1}]) + \theta(\nabla f(y^{t-1}) - \nabla f_i(y^{t-1}))\|^2]$$

$$\leq \frac{\beta(\theta+\beta)}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}[\|(\mathbb{E}[\bar{s}^{t-1}]-\mathbb{E}[s_i^{t-1}])\|^2]+\theta(\theta+\beta)\zeta^2.$$

Unrolling the above recursion yields

$$\begin{aligned}\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}[\left\|\mathbb{E}[\bar{s}^t]-\mathbb{E}[s_i^t]\right\|^2] &\leq \frac{\theta(\theta+\beta)}{1-\beta(\theta+\beta)}\zeta^2+\frac{\beta^t(\theta+\beta)^t}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}[\|(\mathbb{E}[\bar{s}^0]-\mathbb{E}[s_i^0])\|^2]\\ &\leq \frac{\theta(\theta+\beta)}{1-\beta(\theta+\beta)}\zeta^2+\beta^t(\theta+\beta)^t\zeta^2.\end{aligned}$$

Therefore, we have

$$\begin{aligned}\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\mathbb{E}[\|\bar{s}^t-s_i^t\|^2] &\leq \frac{3}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\mathbb{E}[\|s_i^t-\mathbb{E}[s_i^t]\|^2]+\frac{3}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\mathbb{E}[\|\mathbb{E}[\bar{s}^t]-\mathbb{E}[s_i^t]\|^2]+3\mathbb{E}[\|\bar{s}^t-\mathbb{E}[\bar{s}^t]\|^2]\\ &\leq \frac{3}{m}\left(1+\frac{1}{(1-\delta)n}\right)\frac{\theta^2}{1-\beta^2}\sigma^2+3\frac{\theta(\theta+\beta)}{1-\beta(\theta+\beta)}\zeta^2\\ &\quad+\frac{3\beta^{2t}}{(1-\delta)nm_0}\sigma^2+3\beta^t(\theta+\beta)^t\zeta^2. \qquad (52)\end{aligned}$$

Next, consider the third term at the R.H.S. of (49), which is also the third term at the R.H.S. of (50). Using Definition 2, we have

$$\begin{aligned}&\mathbb{E}[\|\mathsf{A}(\{g_i^{t-1}\})-\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}g_i^{t-1}\|^2] \leq \frac{\rho\delta}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\mathbb{E}[\|g_i^{t-1}-\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}g_i^{t-1}\|^2] \qquad (53)\\ &\leq \frac{\rho\delta}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\mathbb{E}[\|g_i^{t-1}-\nabla f_i(y^{t-1})+\nabla f_i(y^{t-1})-\nabla f(y^{t-1})+\nabla f(y^{t-1})-\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}g_i^{t-1}\|^2]\\ &\leq \frac{3\rho\delta\sigma^2}{m}\left(1+\frac{1}{(1-\delta)n}\right)+3\rho\delta\zeta^2.\end{aligned}$$

Substituting (51)–(53) into (50) yields

$$\begin{aligned}&\mathbb{E}[\left\|\hat{s}^t-\bar{s}^t\right\|^2] \qquad (54)\\ &\leq (1-\alpha)\beta(\theta+\beta)\mathbb{E}[\|\hat{s}^{t-1}-\bar{s}^{t-1}\|^2]\\ &\quad+\frac{3\rho\delta\sigma^2}{m}\left(1+\frac{1}{(1-\delta)n}\right)\left(\frac{\alpha\theta^2}{1-\beta^2}+(1-\alpha)\theta(\theta+\beta)+\frac{\alpha\beta^{2t}m}{m_0}\right)\\ &\quad+3\rho\delta\zeta^2\left(\frac{\alpha\theta(\theta+\beta)}{1-\beta(\theta+\beta)}+(1-\alpha)\theta(\theta+\beta)+\alpha\beta^t(\theta+\beta)^tm\right)\\ &\leq \frac{3\rho\delta\sigma^2}{m}\left(1+\frac{1}{(1-\delta)n}\right)\left(\frac{1}{1-(1-\alpha)\beta(\theta+\beta)}\frac{\alpha\theta^2}{1-\beta^2}+\frac{(1-\alpha)\theta(\theta+\beta)}{1-(1-\alpha)\beta(\theta+\beta)}+\frac{\alpha\beta^2m}{(1-\beta^2)m_0}\right)\\ &\quad+3\rho\delta\zeta^2\left(\frac{1}{1-(1-\alpha)\beta(\theta+\beta)}\frac{\alpha\theta(\theta+\beta)}{1-\beta(\theta+\beta)}+\frac{(1-\alpha)\theta(\theta+\beta)}{1-(1-\alpha)\beta(\theta+\beta)}+1\right)\\ &\leq \frac{3\rho\delta\sigma^2}{m}\left(1+\frac{1}{(1-\delta)n}\right)\left(\chi_1+\chi_2+\frac{\alpha\beta^2m}{(1-\beta^2)m_0}\right)+3\rho\delta\zeta^2\left(\chi_3+\chi_2+1\right).\end{aligned}$$

Here, the second inequality uses

$$\sum_{\tau=1}^{t-1}(1-\alpha)^\tau\beta^\tau(\theta+\beta)^\tau \leq \frac{1}{1-(1-\alpha)\beta(\theta+\beta)},$$

$$\begin{aligned}\sum_{\tau=1}^{t}\beta^{2(t-\tau)}(1-\alpha)^\tau\beta^\tau(\theta+\beta)^\tau &\leq \beta^{2t}\sum_{\tau=1}^{t}(1-\alpha)^\tau\beta^{-\tau}(\theta+\beta)^\tau \\ &= \beta^{2t}\sum_{\tau=1}^{t}\chi_0^\tau \leq \beta^{2t}\sum_{\tau=1}^{t}\beta^{-2\tau} \leq \frac{\beta^2}{1-\beta^2},\end{aligned}$$

and

$$\sum_{\tau=1}^{t}\beta^{t-\tau}(\theta+\beta)^{t-\tau}(1-\alpha)^\tau\beta^\tau(\theta+\beta)^\tau \leq \beta^t(\theta+\beta)^t\sum_{\tau=1}^{t}(1-\alpha)^\tau \leq \frac{1}{\alpha}.$$

Substituting (51)-(55) into (49) and thanks to (47), we obtain the desired result. ■

A.6 Bound of stochastic bias $\mathbb{E}[\|\Delta_2^t\|^2]$

Lemma 21 *Consider Algorithm 1 with the step size $\eta = \frac{1}{L}$. If Assumption 3 holds, then we have*

$$\mathbb{E}[\|\Delta_2^t\|^2] \leq \frac{\theta^2\sigma^2}{L^2|\mathcal{H}|m}. \tag{55}$$

Proof It follows from $\eta = \frac{1}{L}$ that

$$\begin{aligned}\mathbb{E}[\|\Delta_2^t\|^2] =& \frac{1}{\tilde{L}^2}\mathbb{E}\left[\left\|\nabla f(y^{t-1}) - \frac{1}{|\mathcal{H}|m}\sum_{i\in\mathcal{H}}\sum_{l=1}^{m}\nabla F(y^{t-1};\xi_i^{(t-1,l)})\right\|^2\right] \\ =& \frac{1}{\tilde{L}^2}\mathbb{E}\left[\left\|\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\nabla f_i(y^{t-1}) - \frac{1}{|\mathcal{H}|m}\sum_{i\in\mathcal{H}}\sum_{j=1}^{m}\nabla F(y^{t-1};\xi_i^{(t-1,l)})\right\|^2\right] \\ =& \frac{1}{\tilde{L}^2|\mathcal{H}|^2m^2}\sum_{i\in\mathcal{H}}\sum_{j=1}^{m}\mathbb{E}\left[\left\|\nabla f_i(y^{t-1}) - \nabla F(y^{t-1};\xi_i^{(t-1,l)})\right\|^2\right] \leq \frac{\sigma^2}{\tilde{L}^2|\mathcal{H}|m},\end{aligned}$$

which completes the proof. ■

A.7 Proof of Theorem 15

Proof For notational convenience, in this proof we define $z_p = z(p)$, $\epsilon_p = \epsilon(p)$, $T_p = T(p)$, and $m_p = m(p)$ for all $p \geq 1$. We consider two cases, $\sigma^2 = 0$ and $\sigma^2 \neq 0$.

The proof sketch is as follows: Algorithm 2 performs P calls of Algorithm 1 to achieve $\mathbb{E}[f(z_P) - f^*] \leq \epsilon_P^2 + O(\rho\delta\zeta^2) \leq \frac{\epsilon^2}{2L} + O(\rho\delta\zeta^2)$. In the p-th call, Algorithm 1 starts from a point z_{p-1} satisfying $\mathbb{E}[f(z_{p-1}) - f^*] \leq \epsilon_{p-1}^2 + O(\rho\delta\zeta^2)$ and then generates a new point z_p satisfying $\mathbb{E}[f(z_p) - f^*] \leq \epsilon_p^2 + O(\rho\delta\zeta^2)$ with $\epsilon_p^2 = \frac{1}{2}\epsilon_{p-1}^2$, ensuring the desired convergence.

To be specific, in the first call, we set

$$\epsilon_1^2 = \frac{32}{\mu}\left(3\rho\delta(1 + \frac{1}{(1-\delta)n}) + \frac{1}{(1-\delta)n}\right)\sigma^2. \tag{56}$$

We run Algorithm 1 with $m_1 = 1$ for T_1 iterations, where

$$T_1 = \min\left\{\lceil 2\sqrt{\kappa}\log\frac{2LR^2}{\epsilon_1^2}\rceil, \lceil 2\sqrt{\kappa}\log\frac{4L^2R^2}{\epsilon^2}\rceil\right\}. \tag{57}$$

When $\epsilon_1^2 \leq \frac{\epsilon^2}{2L}$, we have

$$T_1 = \lceil 2\sqrt{\kappa}\log\frac{4L^2R^2}{\epsilon^2}\rceil \quad \text{and} \quad P = \max\left\{\left\lceil \log_2\frac{4L\epsilon_1^2}{\epsilon^2}\right\rceil, 1\right\} = 1.$$

Therefore, by Theorem 13 with $\alpha = 0$ and $\theta = 1$, we obtain

$$\mathbb{E}[\|\nabla f(z_1)\|] \leq \mathbb{E}[\sqrt{2L(f(z_1) - f^*)}] \leq 2\sqrt{6}\kappa^{1/2}\rho^{1/2}\delta^{1/2}\zeta + \epsilon.$$

It means that z_1 is the desired $\tilde{x}^K$ that satisfies (24), and the required oracle query complexity is m_1T_1.

Otherwise, when $\epsilon_1^2 > \frac{\epsilon^2}{2L}$, by Theorem 13 with $\alpha = 0$ and $\theta = 1$, we have $\chi_4 = \chi_5 = 1$ and hence $\mathbb{E}[f(z_1) - f^*] \leq \epsilon_1^2 + 48\mu^{-1}\rho\delta\zeta^2$. Now we set $\epsilon_2^2 = \frac{1}{2}\epsilon_1^2$ and increase the batch size to $m_2 = 2m_1$. Then using Theorem 13 again, we have

$$\begin{aligned}\mathbb{E}[f(z_2) - f^*] &\leq 2\left(1 - \frac{1}{2\sqrt{\kappa}}\right)^{T_2}\mathbb{E}[f(z_1) - f^*] \\ &\quad + 16\mu^{-1}\left(\frac{3\rho\delta\sigma^2}{m_2}(1 + \frac{1}{(1-\delta)n}) + \frac{\sigma^2}{(1-\delta)nm_2} + 3\rho\delta\zeta^2\right) \\ &\leq 2\left(1 - \frac{1}{2\sqrt{\kappa}}\right)^{T_2}(\epsilon_1^2 + 48\mu^{-1}\rho\delta\zeta^2) + \frac{\epsilon_2^2}{2} + 48\mu^{-1}\rho\delta\zeta^2.\end{aligned}$$

With $T_2 = \lceil 2\sqrt{\kappa}\log 8\rceil$, we have $\mathbb{E}[f(z_2) - f^*] \leq \epsilon_2^2 + 48(1 + \frac{1}{4})\mu^{-1}\rho\delta\zeta^2$. Similarly, in the p-th call, by setting $\epsilon_p^2 = \frac{1}{2}\epsilon_{p-1}^2$, $m_p = 2m_{p-1} = 2^{p-1}$ and $T_p = \lceil 2\sqrt{\kappa}\log 8\rceil$, we have

$$\mathbb{E}[f(z_p) - f^*] \leq \epsilon_p^2 + 48\sum_{j=1}^{p} 4^{1-p}\mu^{-1}\rho\delta\zeta^2 \leq \epsilon_p^2 + 64\mu^{-1}\rho\delta\zeta^2. \tag{58}$$

Letting $p = P$ in (58), we have

$$\begin{aligned}\mathbb{E}[\|\nabla f(z_P)\|] &\leq \mathbb{E}[\sqrt{2L(f(z_P) - f^*)}] \\ &\leq \sqrt{128\kappa\rho\delta\zeta^2 + \epsilon^2}\end{aligned}$$

$$\leq 8\sqrt{2}\kappa^{1/2}\rho^{1/2}\delta^{1/2}\zeta + \epsilon,$$

Moreover, the oracle query complexity of the p-th ($p \geq 2$) call is upper bounded by $m_p T_p = 2^{p-1}\lceil 2\sqrt{\kappa}\log 8\rceil$, which results in the overall oracle query complexity

$$O\left(2\sqrt{\kappa}\log\frac{4L^2R^2}{\epsilon^2} + \sum_{p=2}^{P} 2^{p-1}\lceil 2\sqrt{\kappa}\log 8\rceil\right) \quad \text{with} \quad P = \left\lceil \log_2 \frac{4L\epsilon_1^2}{\epsilon^2}\right\rceil$$

to reach $\epsilon_P^2 \leq \frac{\epsilon^2}{2L}$ thanks to $\epsilon^2 < \epsilon_1^2 L$. With elementary calculations, the overall oracle query complexity can be rewritten as

$$O\left(2\sqrt{\kappa}\log\frac{4L^2R^2}{\epsilon^2} + 128\kappa^{3/2}\left(3\rho\delta(1+\frac{1}{(1-\delta)n}) + \frac{1}{(1-\delta)n}\right)\frac{\sigma^2}{\epsilon^2}\right), \tag{59}$$

which completes the proof. ■

A.8 Proof of Theorem 16

To prove Theorem 16, we summarize the update rules in Algorithm 1 as

$$\begin{aligned}
&\hat{s}^t = \alpha \mathsf{A}(\{\beta s_i^{t-1} + \theta g_i^{t-1}\}_{i=1}^n) + (1-\alpha)\beta s^{t-1} + (1-\alpha)\theta \mathsf{A}(\{g_i^{t-1}\}_{i=1}^n),\\
&x^t = x^{t-1} - \eta\hat{s}^t,\\
&y^t = x^t + \beta(x^t - x^{t-1}),
\end{aligned}$$

which further indicate that

$$\begin{aligned}
y^t &= x^t + \beta(x^t - x^{t-1}) = x^t - \eta\beta\hat{s}^t\\
&= x^{t-1} - \eta\hat{s}^t - \eta\beta\hat{s}^t\\
&= x^{t-1} - \eta\beta\hat{s}^{t-1} + \eta\beta\hat{s}^{t-1} - \eta\hat{s}^t - \eta\beta\hat{s}^t\\
&= y^{t-1} - \eta((1+\beta)\hat{s}^t - \beta\hat{s}^{t-1}) = y^{t-1} - \eta p^t,
\end{aligned}$$

with $p^t := (1+\beta)\hat{s}^t - \beta\hat{s}^{t-1}$. Before proving Theorem 16, we give two auxiliary lemmas.

Lemma 22 *Consider Algorithm 1 and suppose that Assumption 1 holds. Setting $\beta \in [0,1]$ and $\eta \leq \frac{1}{L}$, for any $t \geq 1$ we have*

$$\mathbb{E}[f(y^t)] \leq f(y^{t-1}) - \frac{\eta}{2}\|\nabla f(y^{t-1})\|^2 + \eta\mathbb{E}[\|\bar{e}^t\|^2] + \eta\mathbb{E}[\|p^t - \bar{p}^t\|^2]. \tag{60}$$

where $\bar{e}^t := \bar{p}^t - \nabla f(y^{t-1})$ and $\bar{p}^t := (1+\beta)\bar{s}^t - \beta\bar{s}^{t-1}$.

Proof By the smoothness of the function f and the server update, we have

$$\begin{aligned}
f(y^t) &\leq f(y^{t-1}) - \eta\langle\nabla f(y^{t-1}), p^t\rangle + \frac{L\eta^2}{2}\|p^t\|^2\\
&\leq f(y^{t-1}) - \eta\langle\nabla f(y^{t-1}), p^t\rangle + \frac{\eta}{2}\|p^t\|^2
\end{aligned}$$

$$
\begin{aligned}
&= f(y^{t-1}) + \frac{\eta}{2}\|p^t - \nabla f(y^{t-1})\|^2 - \frac{\eta}{2}\|\nabla f(y^{t-1})\|^2 \\
&= f(y^{t-1}) + \frac{\eta}{2}\|p^t - \bar{p}^t + \bar{p}^t - \nabla f(y^{t-1})\|^2 - \frac{\eta}{2}\|\nabla f(y^{t-1})\|^2 \\
&\leq f(y^{t-1}) + \eta\|\bar{e}^t\|^2 + \eta\|p^t - \bar{p}^t\|^2 - \frac{\eta}{2}\|\nabla f(y^{t-1})\|^2.
\end{aligned}
$$

Taking conditional expectations on both sides yields (60). ■

Lemma 23 *Consider Algorithm 1 and suppose that Assumptions 1–3 hold. Setting β, θ and η such that $\beta = 1 - 12L\eta \geq \frac{1}{2}$, $0 \leq \theta \leq 2$ and $(1-\theta-\beta)^2 \leq \chi_6(1-\beta)^2$, for any $t \geq 2$ we have*

$$
\begin{aligned}
\mathbb{E}\|\bar{e}^t\|^2 \leq& \beta\mathbb{E}\|\bar{e}^{t-1}\|^2 + \frac{3(1-\beta)}{8}\|p^{t-1} - \bar{p}^{t-1}\|^2 \\
&+ (12\chi_6 + \frac{3}{8})(1-\beta)\|\nabla f(y^{t-2})\|^2 + \frac{5\theta^2\sigma^2}{(1-\delta)nm}.
\end{aligned} \tag{61}
$$

Proof According to (11) and the definition of $\bar{p}^t := (1+\beta)\bar{s}^t - \beta\bar{s}^{t-1}$, we have

$$
\begin{aligned}
\bar{p}^t - \beta\bar{p}^{t-1} &= (1+\beta)\bar{s}^t - \beta\bar{s}^{t-1} - \beta\bar{p}^{t-1} \\
&= \beta\bar{s}^t + \theta\bar{g}^{t-1} - \beta\bar{p}^{t-1} \\
&= \theta\bar{g}^{t-1} + \beta(\beta\bar{s}^{t-1} + \theta\bar{g}^{t-1}) - \beta\bar{p}^{t-1} \\
&= \theta\bar{g}^{t-1} + \beta(\beta\bar{s}^{t-1} + \theta\bar{g}^{t-1}) - \beta(\beta\bar{s}^{t-1} + \theta\bar{g}^{t-2}) \\
&= \theta\left(\bar{g}^{t-1} + \beta(\bar{g}^{t-1} - \bar{g}^{t-2})\right),
\end{aligned}
$$

where $\bar{g}^t = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} g_i^t$. Therefore, according to the definition of $\bar{s}$, we have

$$
\begin{aligned}
\mathbb{E}\|\bar{e}^t\|^2 &= \mathbb{E}\|\bar{p}^t - \nabla f(y^{t-1})\|^2 \\
&= \mathbb{E}\|\beta\bar{p}^{t-1} + \theta\left(\bar{g}^{t-1} + \beta(\bar{g}^{t-1} - \bar{g}^{t-2})\right) - \nabla f(y^{t-1})\|^2 \\
&= \mathbb{E}\|\beta(\bar{p}^{t-1} - \nabla f(y^{t-2})) + \theta(1+\beta)(\bar{g}^{t-1} - \nabla f(y^{t-1})) - \theta\beta(\bar{g}^{t-2} - \nabla f(y^{t-2})) \\
&\quad - \beta(1-\theta)(\nabla f(y^{t-1}) - \nabla f(y^{t-2})) + (\theta+\beta-1)\nabla f(y^{t-1})\|^2 \\
&\leq \beta^2(1 + \frac{1-\beta}{2})\|\bar{p}^{t-1} - \nabla f(y^{t-2})\|^2 + \theta^2(1+\beta)^2\|\bar{g}^{t-1} - \nabla f(y^{t-1})\|^2 \\
&\quad + \theta^2\beta^2\|\bar{g}^{t-2} - \nabla f(y^{t-2})\|^2 + 2\beta^2(1-\theta)^2(1 + \frac{2}{1-\beta})\|\nabla f(y^{t-1}) - \nabla f(y^{t-2})\|^2 \\
&\quad + 2(1-\theta-\beta)^2(1 + \frac{2}{1-\beta})\|\nabla f(y^{t-1}) \pm \nabla f(y^{t-2})\|^2 \\
&\leq \beta^2(1 + \frac{1-\beta}{2})\|\bar{p}^{t-1} - \nabla f(y^{t-2})\|^2 + \theta^2(1+\beta)^2\|\bar{g}^{t-1} - \nabla f(y^{t-1})\|^2 \\
&\quad + \theta^2\beta^2\|\bar{g}^{t-2} - \nabla f(y^{t-2})\|^2 + \frac{12}{1-\beta}(1-\theta-\beta)^2\|\nabla f(y^{t-2})\|^2 \\
&\quad + \frac{6}{1-\beta}(\beta^2(1-\theta)^2 + 2(1-\theta-\beta)^2)\|\nabla f(y^{t-1}) - \nabla f(y^{t-2})\|^2
\end{aligned}
$$

$$
\begin{aligned}
&\leq \beta^2(1+\frac{1-\beta}{2})\mathbb{E}\|\bar{e}^{t-1}\|^2+\frac{5\theta^2\sigma^2}{(1-\delta)nm}+\frac{12}{1-\beta}(1-\theta-\beta)^2\|\nabla f(y^{t-2})\|^2\\
&\quad+\frac{6L^2\eta^2}{1-\beta}(\beta^2(1-\theta)^2+2(1-\theta-\beta)^2)\|p^{t-1}\pm\bar{p}^{t-1}\pm\nabla f(y^{t-2})\|^2\\
&\leq \beta^2(1+\frac{1-\beta}{2})\mathbb{E}\|\bar{e}^{t-1}\|^2+\frac{5\theta^2\sigma^2}{(1-\delta)nm}+\frac{12}{1-\beta}(1-\theta-\beta)^2\|\nabla f(y^{t-2})\|^2\\
&\quad+\frac{6L^2\eta^2}{1-\beta}(\beta^2(1-\theta)^2+2(1-\theta-\beta)^2)(3\|p^{t-1}-\bar{p}^{t-1}\|^2+3\|\bar{e}^{t-1}\|^2+3\|\nabla f(y^{t-2})\|^2)\\
&\leq \beta^2\left(\frac{3-\beta}{2}+\frac{54L^2\eta^2}{1-\beta}\right)\mathbb{E}\|\bar{e}^{t-1}\|^2+\frac{5\theta^2\sigma^2}{(1-\delta)nm}+\frac{54L^2\eta^2}{1-\beta}\|p^{t-1}-\bar{p}^{t-1}\|^2\\
&\quad+\frac{6}{1-\beta}\left(2\chi_6(1-\beta)^2+9L^2\eta^2\right)\|\nabla f(y^{t-2})\|^2\\
&\leq \beta\mathbb{E}\|\bar{e}^{t-1}\|^2+\frac{3(1-\beta)}{8}\|p^{t-1}-\bar{p}^{t-1}\|^2+(12\chi_6+\frac{3}{8})(1-\beta)\|\nabla f(y^{t-2})\|^2+\frac{5\theta^2\sigma^2}{(1-\delta)nm}\,.
\end{aligned}
$$

Here, the second to last inequality uses

$$
(1-\theta)^2\leq 1,\quad (1-\theta-\beta)^2\leq\min\{\beta^2,\chi_6(1-\beta)^2\},\quad \frac{1}{2}\leq\beta\leq 1.
$$

The last inequality is due to $\beta=1-12L\eta$. This completes the proof. ■

Proof of Theorem 16 Multiplying (61) by $\frac{\eta}{1-\beta}$ and adding it to (60) yield

$$
\begin{aligned}
\mathbb{E}[f(y^t)]+\frac{\eta\beta}{1-\beta}\mathbb{E}[\|\bar{e}^t\|^2]&\leq f(y^{t-1})-\frac{\eta}{2}\|\nabla f(y^{t-1})\|^2+\eta\mathbb{E}[\|p^t-\bar{p}^t\|^2]\\
&\quad+\frac{\eta\beta}{1-\beta}\mathbb{E}[\|\bar{e}^{t-1}\|^2]+\frac{3\eta}{8}\mathbb{E}[\|p^{t-1}-\bar{p}^{t-1}\|^2]\\
&\quad+(12\chi_6+\frac{3}{8})\eta\|\nabla f(y^{t-2})\|^2+\frac{5\theta^2\eta}{1-\beta}\frac{\sigma^2}{(1-\delta)nm}.
\end{aligned}
\tag{62}
$$

For the term of $\mathbb{E}[\|p^t-\bar{p}^t\|^2]$ in (62), applying (55), we have

$$
\begin{aligned}
\mathbb{E}[\|p^t-\bar{p}^t\|^2]&=\mathbb{E}[\|(1+\beta)\hat{s}^t-\beta\hat{s}^{t-1}-(1+\beta)\bar{s}^t+\beta\bar{s}^{t-1}\|^2]\\
&\leq 2(1+\beta)^2\mathbb{E}[\|\hat{s}^t-\bar{s}^t\|^2]+2\beta^2\mathbb{E}[\|\hat{s}^{t-1}-\bar{s}^{t-1}\|^2]\\
&\leq \frac{30\rho\delta\sigma^2}{m}\left(1+\frac{1}{(1-\delta)n}\right)\left(\chi_1+\chi_2+\frac{\alpha\beta^2m}{(1-\beta^2)m_0}\right)+30\rho\delta\zeta^2\left(\chi_3+\chi_2+1\right).
\end{aligned}
\tag{63}
$$

Substituting (63) into (62) and rearranging the terms yield

$$
\begin{aligned}
&\mathbb{E}[f(y^t)-f^*]+\frac{\eta\beta}{1-\beta}\mathbb{E}[\|\bar{e}^t\|^2]+(12\chi_6+\frac{3}{8})\eta\mathbb{E}[\|\nabla f(y^{t-1})\|^2]\\
&\leq \mathbb{E}[f(y^{t-1})-f^*]+\frac{\eta\beta}{1-\beta}\mathbb{E}[\|\bar{e}^{t-1}\|^2]+(12\chi_6+\frac{3}{8})\eta\mathbb{E}[\|\nabla f(y^{t-2})\|^2]\\
&\quad-\left(\frac{1}{8}-12\chi_6\right)\eta\|\nabla f(y^{t-1})\|^2+\frac{5\theta^2\eta}{1-\beta}\frac{\sigma^2}{(1-\delta)nm}
\end{aligned}
\tag{64}
$$

$$+\frac{42\eta\rho\delta\sigma^2}{m}\left(1+\frac{1}{(1-\delta)n}\right)\left(\chi_1+\chi_2+\frac{\alpha\beta^2 m}{(1-\beta^2)m_0}\right)+42\eta\rho\delta\zeta^2\left(\chi_3+\chi_2+1\right).$$

Summing over t from 2 to T and also rearranging the terms, we have

$$\begin{aligned}
&\sum_{t=2}^{T}\left(\frac{1}{8}-12\chi_6\right)\|\nabla f(y^{t-1})\|^2 \qquad (65)\\
\le\ &\frac{1}{\eta}\mathbb{E}[f(y^1)-f^*]+\frac{\beta}{1-\beta}\mathbb{E}[\|\bar{e}^1\|^2]+(12\chi_6+\frac{3}{8})\mathbb{E}[\|\nabla f(y^0)\|^2]\\
&+\sum_{t=2}^{T}\left(\frac{5\theta^2}{(1-\beta)(1-\delta)n}+42\rho\delta\left(1+\frac{1}{(1-\delta)n}\right)\left(\chi_1+\chi_2+\frac{\alpha\beta^2 m}{(1-\beta^2)m_0}\right)\right)\frac{\sigma^2}{m}\\
&+\sum_{t=2}^{T}42\rho\delta\zeta^2\left(\chi_3+\chi_2+1\right).
\end{aligned}$$

For the second line in (65), using Lemmas 22 and 23 with $t=1$ yields

$$\begin{aligned}
&\frac{1}{\eta}\mathbb{E}[f(y^1)-f^*]+\frac{\beta}{1-\beta}\mathbb{E}[\|\bar{e}^1\|^2]+(12\chi_6+\frac{3}{8})\mathbb{E}[\|\nabla f(y^0)\|^2] \qquad (66)\\
\le\ &\frac{1}{\eta}\left(f(y^0)-f^*\right)+\left(1+\frac{\beta}{1-\beta}\right)\mathbb{E}[\|\bar{e}^1\|^2]+\mathbb{E}[\|p^1-\bar{p}^1\|^2]+(12\chi_6-\frac{1}{8})\mathbb{E}[\|\nabla f(y^0)\|^2]\\
\le\ &\frac{1}{\eta}\Delta+\frac{(\theta+\beta^2+\theta\beta)^2}{1-\beta}\frac{\sigma^2}{(1-\delta)nm}+\mathbb{E}[\|p^1-\bar{p}^1\|^2]\\
&+(12\chi_6-\frac{1}{8}+(\theta+\beta^2+\theta\beta-1)^2)\mathbb{E}[\|\nabla f(y^0)\|^2].
\end{aligned}$$

Substituting (66) into (65), with $\frac{1}{8}-12\chi_6-(\theta+\beta^2+\theta\beta-1)^2>\chi_7$, we have

$$\begin{aligned}
&\sum_{t=1}^{T}\chi_7\|\nabla f(y^{t-1})\|^2\\
\le\ &\sum_{t=1}^{T}\left(\frac{1}{8}-12\chi_6-(\theta+\beta^2+\theta\beta-1)^2\right)\|\nabla f(y^{t-1})\|^2\\
\le\ &\frac{1}{\eta}\Delta+\frac{(\theta+\beta^2+\theta\beta)^2}{1-\beta}\frac{\sigma^2}{(1-\delta)nm}+\sum_{t=1}^{T}42\rho\delta\zeta^2\left(\chi_3+\chi_2+1\right)\\
&+\sum_{t=1}^{T}\left(\frac{5\theta^2}{(1-\beta)(1-\delta)n}+42\rho\delta\left(1+\frac{1}{(1-\delta)n}\right)\left(\chi_1+\chi_2+\frac{\alpha\beta^2 m}{(1-\beta^2)m_0}\right)\right)\frac{\sigma^2}{m},
\end{aligned}$$

where the last inequality comes from (65) and (66). Consequently, it holds that

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\|\nabla f(y^{t-1})\|^2 \qquad (67)\\
\le\ &\frac{1}{\chi_7}\Bigg(\frac{1}{\eta T}\Delta+\frac{1}{T}\frac{(\theta+\beta^2+\theta\beta)^2}{1-\beta}\frac{\sigma^2}{(1-\delta)nm}+42\rho\delta\zeta^2\left(\chi_3+\chi_2+1\right)
\end{aligned}$$

$$+\left(\frac{5\theta^2}{(1-\beta)(1-\delta)n}+42\rho\delta\left(1+\frac{1}{(1-\delta)n}\right)\left(\chi_1+\chi_2+\frac{\alpha\beta^2 m}{(1-\beta^2)m_0}\right)\right)\frac{\sigma^2}{m}\Bigg).$$

Setting $1-\beta=12L\eta$, since $\theta^2\le 2(1-\theta-\beta)^2+2(1-\beta)^2\le(2\chi_6+2)(1-\beta)^2$, $\theta+\beta^2+\theta\beta\le 2$, $\chi_1+\chi_2=O(L\eta)$, $\chi_3=O(1)\ge 0$, $m_0=\frac{m}{L^2\eta^2}$, and $\chi_7=\Theta(1)>0$, we have

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\|\nabla f(y^{t-1})\|^2\le O\Bigg(&\frac{1}{\eta T}(\Delta+\frac{\sigma^2}{L(1-\delta)nm})\\&+\eta\left(\frac{1}{(1-\delta)n}+\rho\delta\left(1+\frac{1}{(1-\delta)n}\right)\right)\frac{L\sigma^2}{m}+\rho\delta\zeta^2\Bigg).\end{aligned}$$

Setting the step size

$$\eta=\min\left(\sqrt{\frac{\Delta+\frac{\sigma^2}{L(1-\delta)nm}}{T\left(\frac{1}{(1-\delta)n}+\rho\delta\left(1+\frac{1}{(1-\delta)n}\right)\right)\frac{L\sigma^2}{m}}},\frac{1}{24L}\right),\tag{68}$$

with $m=O(1)$, we obtain

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\|\nabla f(y^{t-1})\|^2\le O\Bigg(&\sqrt{\frac{L\Delta+\sigma^2/n}{T}}\sqrt{\left(\frac{1}{(1-\delta)n}+\rho\delta\right)\sigma^2}\\&+\frac{L\Delta}{T}+\frac{\sigma^2}{Tn}+\rho\delta\zeta^2\Bigg),\end{aligned}$$

which completes the proof. ■

A.9 Proof of Theorem 18

Proof For any $\gamma=1,\cdots,\Gamma$, define an auxiliary variable $\varkappa^{\gamma*}=\arg\min f^\gamma(\varkappa)$. In view of the strong convexity of f^γ, we have

$$\begin{aligned}f^\gamma(\varkappa)&\ge f^\gamma(\varkappa^{\gamma*})+\frac{L}{2}\|\varkappa-\varkappa^{\gamma*}\|^2\\&=f^\gamma(\varkappa^\gamma)+f^\gamma(\varkappa^{\gamma*})-f^\gamma(\varkappa^\gamma)+\frac{L}{2}\|x-\varkappa^{\gamma*}\|^2\\&=f(\varkappa^\gamma)+L\|\varkappa^\gamma-\varkappa^{\gamma-1}\|^2+f^\gamma(\varkappa^{\gamma*})-f^\gamma(\varkappa^\gamma)+\frac{L}{2}\|x-\varkappa^{\gamma*}\|^2.\end{aligned}\tag{69}$$

Setting $x=\varkappa^\gamma$ in (69) and summing it from $\gamma=1,\cdots,\Gamma$, we obtain

$$\frac{L}{2}\sum_{\gamma=1}^{\Gamma}\|\varkappa^\gamma-\varkappa^{\gamma*}\|^2\le\sum_{\gamma=1}^{\Gamma}\left(f^\gamma(\varkappa^\gamma)-f^\gamma(\varkappa^{\gamma*})\right).\tag{70}$$

Similarly, setting $x = \varkappa^{\gamma-1}$ in (69) and using $f^\gamma(\varkappa^{\gamma-1}) = f(\varkappa^{\gamma-1})$ yield

$$f(\varkappa^{\gamma-1}) \geq f(\varkappa^\gamma) + L\|\varkappa^\gamma - \varkappa^{\gamma-1}\|^2 + f^\gamma(\varkappa^{\gamma*}) - f^\gamma(\varkappa^\gamma) + \frac{L}{2}\|\varkappa^{\gamma-1} - \varkappa^{\gamma*}\|^2.$$

Summing the above inequality from $\gamma = 1, \cdots, \Gamma$, we obtain

$$L\sum_{\gamma=1}^{\Gamma}\left(\|\varkappa^\gamma - \varkappa^{\gamma-1}\|^2 + \frac{1}{2}\|\varkappa^{\gamma-1} - \varkappa^{\gamma*}\|^2\right) \leq f(\varkappa^0) - f(\varkappa^\gamma) + \sum_{\gamma=1}^{\Gamma}\left(f^\gamma(\varkappa^\gamma) - f^\gamma(\varkappa^{\gamma*})\right). \quad (71)$$

Due to the optimality of $\varkappa^{\gamma*}$, we have

$$0 = \nabla f^\gamma(\varkappa^{\gamma*}) = \nabla f(\varkappa^{\gamma*}) + 2L(\varkappa^{\gamma*} - \varkappa^{\gamma-1}),$$

which, together with (70) and (71), further implies

$$\begin{aligned}
\sum_{\gamma=1}^{\Gamma}\|\nabla f(\varkappa^\gamma)\|^2 &= \sum_{\gamma=1}^{\Gamma}\|\nabla f(\varkappa^\gamma) - \nabla f(\varkappa^{\gamma*}) + \nabla f(\varkappa^{\gamma*})\|^2 \\
&\leq 2\sum_{\gamma=1}^{\Gamma}\|\nabla f(\varkappa^\gamma) - \nabla f(\varkappa^{\gamma*})\|^2 + 2\sum_{\gamma=1}^{\Gamma}\|\nabla f(\varkappa^{\gamma*})\|^2 \\
&\leq 2L^2\sum_{\gamma=1}^{\Gamma}\|\varkappa^\gamma - \varkappa^{\gamma*}\|^2 + 8L^2\sum_{\gamma=1}^{\Gamma}\|\varkappa^{\gamma*} - \varkappa^{\gamma-1}\|^2 \\
&\leq 20L\sum_{\gamma=1}^{\Gamma}\left(f^\gamma(\varkappa^\gamma) - f^\gamma(\varkappa^{\gamma*})\right) + 16L\left(f(\varkappa^0) - f(\varkappa^\gamma)\right).
\end{aligned} \quad (72)$$

Now we bound the term $f^\gamma(\varkappa^\gamma) - f^\gamma(\varkappa^{\gamma*})$ in (72) by induction. Following the analysis from (58) to (59), we know that to achieve

$$\mathbb{E}[f^1(\varkappa^1) - f^1(\varkappa^{1*})] \leq \frac{\epsilon^2}{40L} + \frac{64\rho\delta\zeta^2}{L},$$

the oracle query complexity of Byrd-reNester is at least

$$S_1 \leq 2\sqrt{3}\log\frac{36L\Delta}{\epsilon^2} + 7680\sqrt{3}\left(3\rho\delta(1 + \frac{1}{(1-\delta)n}) + \frac{1}{(1-\delta)n}\right)\frac{\sigma^2}{\epsilon^2},$$

because $\kappa = 3$ and Lipschitz smoothness constant of f^1 is $3L$. Suppose that $\mathbb{E}[f^{\gamma-1}(\varkappa^{\gamma-1}) - f^{\gamma-1}(\varkappa^{(\gamma-1)*})] \leq \frac{\epsilon^2}{40L} + \frac{64\rho\delta\zeta^2}{L}$, we have

$$\begin{aligned}
\mathbb{E}[f(\varkappa^{\gamma-1})] \leq& \mathbb{E}[f^{\gamma-1}(\varkappa^{\gamma-1})] \leq \mathbb{E}[f^{\gamma-1}(\varkappa^{(\gamma-1)*}) + \frac{\epsilon^2}{40L} + \frac{64\rho\delta\zeta^2}{L}] \\
\leq& \mathbb{E}[f(\varkappa^{\gamma-2}) + \frac{\epsilon^2}{40L} + \frac{64\rho\delta\zeta^2}{L}],
\end{aligned}$$

and hence

$$\mathbb{E}[f(\varkappa^{\gamma-1})] \leq f(\varkappa^0) + (\gamma-1)\frac{\epsilon^2}{40L} + (\gamma-1)\frac{64\rho\delta\zeta^2}{L}.$$

Using the analysis from (58) to (59) again, for any $\gamma = 1, \cdots, \Gamma$ we obtain that

$$\mathbb{E}[f^\gamma(\varkappa^\gamma) - f^\gamma(\varkappa^{\gamma*})] \leq \frac{\epsilon^2}{40L} + \frac{64\rho\delta\zeta^2}{L} \tag{73}$$

with oracle query complexity

$$\begin{aligned}
S_\gamma &\leq 2\sqrt{3}\log\frac{36L(f^\gamma(\varkappa^{\gamma-1}) - f^\gamma(\varkappa^{\gamma*}))}{\epsilon^2} + 7680\sqrt{3}\left(3\rho\delta(1+\frac{1}{(1-\delta)n}) + \frac{1}{(1-\delta)n}\right)\frac{\sigma^2}{\epsilon^2} \\
&\leq 2\sqrt{3}\log\frac{36L(f^\gamma(\varkappa^{\gamma-1}) - f^*)}{\epsilon^2} + 7680\sqrt{3}\left(3\rho\delta(1+\frac{1}{(1-\delta)n}) + \frac{1}{(1-\delta)n}\right)\frac{\sigma^2}{\epsilon^2} \\
&\leq 2\sqrt{3}\log\frac{36L(f(\varkappa^0) - f^* + (\gamma-1)\frac{\epsilon^2}{40L} + (\gamma-1)\frac{64\rho\delta\zeta^2}{L})}{\epsilon^2} \\
&\quad + 7680\sqrt{3}\left(3\rho\delta(1+\frac{1}{(1-\delta)n}) + \frac{1}{(1-\delta)n}\right)\frac{\sigma^2}{\epsilon^2}.
\end{aligned}$$

Combining (72) and (73), we have

$$\begin{aligned}
\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\mathbb{E}[\|\nabla f(\varkappa^\gamma)\|^2] &\leq \frac{20L}{\Gamma}\sum_{\gamma=1}^{\Gamma}\mathbb{E}[f^\gamma(\varkappa^\gamma) - f^\gamma(\varkappa^{\gamma*})] + \frac{16L}{\Gamma}\left(f(\varkappa^0) - f(\varkappa^\gamma)\right) \\
&\leq \frac{\epsilon^2}{2} + 1280\rho\delta\zeta^2 + \frac{16L}{\Gamma}\left(f(\varkappa^0) - f^*\right).
\end{aligned}$$

With $\Gamma = \lceil 32L\left(f(\varkappa^0) - f^*\right)\epsilon^{-2}\rceil$, the oracle query complexity is

$$\sum_{\gamma=1}^{\Gamma} S_\gamma \leq \Gamma S_\Gamma = O\left(\frac{L\Delta\rho\delta\sigma^2}{\epsilon^4} + \frac{L\Delta\sigma^2}{(1-\delta)n\epsilon^4} + \frac{L\Delta}{\epsilon^2}\log\frac{L\Delta(1+\rho\delta\zeta^2)}{\epsilon^2}\right),$$

which completes the proof. ■