Robust Distributed Learning Against Both Distributional Shifts and Byzantine Attacks

Guanqiang Zhou, Ping Xu, *Member, IEEE*, Yue Wang, *Senior Member, IEEE*, and Zhi Tian, *Fellow, IEEE*

***Abstract*— In distributed learning systems, robustness threat may arise from two major sources. On the one hand, due to distributional shifts between training data and test data, the trained model could exhibit poor out-of-sample performance. On the other hand, a portion of working nodes might be subject to Byzantine attacks, which could invalidate the learning result. In this article, we propose a new research direction that jointly considers distributional shifts and Byzantine attacks. We illuminate the major challenges in addressing these two issues simultaneously. Accordingly, we design a new algorithm that equips distributed learning with both distributional robustness and Byzantine robustness. Our algorithm is built on recent advances in distributionally robust optimization (DRO) as well as norm-based screening (NBS), a robust aggregation scheme against Byzantine attacks. We provide convergence proofs in three cases of the learning model being nonconvex, convex, and strongly convex for the proposed algorithm, shedding light on its convergence behaviors and endurability against Byzantine attacks. In particular, we deduce that any algorithm employing NBS (including ours) cannot converge when the percentage of Byzantine nodes is (1/3) or higher, instead of (1/2), which is the common belief in current literature. The experimental results verify our theoretical findings (on the breakpoint of NBS and others) and also demonstrate the effectiveness of our algorithm against both robustness issues, justifying our choice of NBS over other widely used robust aggregation schemes. To the best of our knowledge, this is the first work to address distributional shifts and Byzantine attacks simultaneously.**

***Index Terms*— Byzantine attacks, distributed learning, distributional shifts, norm-based screening (NBS), Wasserstein distance.**

I. Introduction

DISTRIBUTED learning usually refers to the paradigm where a number of working nodes (workers) carry out the overall task of training a model in parallel, coordinated by a central node (server). It plays an increasingly important role in solving large-scale machine learning problems for several reasons, including its expandable computational/storage capacities, growing size of modern datasets, and privacy concerns [1], [2], [3], [4]. As the deployment of machine learning becomes prevalent in modern safety-critical fields (such as autonomous driving [5] and medical diagnosis [6]) where the cost of model failure is extremely high, it is crucial to equip the learning systems with some robust features such that the risk of model failure is minimized.

In distributed learning, there are two major robustness issues that may pose a threat to model safety. The first issue is distributional shifts, which exposes the vulnerability of empirical risk minimization (ERM), the de facto training paradigm in machine learning. In ERM, the model is trained to minimize the training loss and then is applied to unseen data, or test data, on the key assumption that training data and test data are drawn from the same distribution. However, this assumption rarely holds in a practical scenario due to selection biases in training data [7], nonstationarity in the environment [8], or even adversarial perturbations [12], leaving the ERM-trained models susceptible to drastically degraded performance under some minor level of distributional shifts. Note that this issue is naturally encountered in distributed and centralized settings alike.

The second issue is Byzantine attacks. In a typical distributed training iteration, each worker is supposed to send its honest and accurate local update to the server, which uses the average of these local updates to refine the model. However, due to a myriad of system glitches such as data corruption, computational error, and transmission interference, a portion of workers could send unwarranted updates to the server, thus polluting the refined model [41]. Even worse, an adversary might intentionally insert malicious workers into the system to attack the model. Such a scenario is especially likely in federated learning where the server usually does not have a chance to thoroughly verify the honesty and competency of all participating devices. Due to the difficulty in modeling each type of system error separately, as well as the concern of malicious workers, researchers in this field often model them uniformly as Byzantine attacks [27], where a malfunctional/Byzantine worker can send arbitrary messages to the server. It is well known that even a single Byzantine worker can totally invalidate the learning result and cause model failure [39].

In this article, we propose a new research direction that jointly considers distributional shifts and Byzantine attacks in distributed learning. Such an effort is worthwhile in order to attain overall model safety/robustness since a Byzantine-robust model may not necessarily be resilient to distributional shifts (as shown in our simulations) and vice versa. It is worth noting that there appears to be a growing interest in simultaneously satisfying multiple well-known constraints that are relevant in

Manuscript received 15 June 2022; revised 22 January 2023, 4 May 2023, and 1 January 2024; accepted 15 March 2024. This work was supported in part by the National Science Foundation under Grant 1939553, Grant 2003211, Grant 2128596, Grant 2231209, and Grant 2413622. *(Corresponding author: Guanqiang Zhou.)*

Guanqiang Zhou was with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030 USA. He is now with the Department of Electrical and Computer Engineering, University of Iowa, Iowa City, IA 52242 USA (e-mail: gzhou4@gmu.edu).

Ping Xu is with the Department of Electrical and Computer Engineering, University of Texas Rio Grande Valley, Edinburg, TX 78539 USA.

Yue Wang is with the Department of Computer Science, Georgia State University, Atlanta, GA 30303 USA.

Zhi Tian is with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030 USA.

Digital Object Identifier 10.1109/TNNLS.2024.3436149

federated learning, such as communication efficiency, fairness, robustness, and privacy [2], [3], [4]. Yet, when robustness is considered as one of the design goals, it almost exclusively refers to Byzantine robustness [47], [48], [49], [50], [51]. We want to point out that Byzantine robustness is not an all-around safety measure and mitigating distributional shifts is of equal importance as far as model safety is concerned.

Despite extensive efforts to address distributional shifts and Byzantine attacks separately, we observe that there has not yet been any work that claims to resolve both issues simultaneously, which may encounter two hurdles. The first hurdle is that the issue of distributional shifts has been conventionally considered in the centralized setting where a single machine has access to all the data. Consequently, the established approaches often lead to solving some form of convex programs, such as linear programs [19], semidefinite programs [13], and second-order cone programs [14], which are not directly solvable in a distributed network where data scatter across multiple local devices. The second hurdle along this path, although being less known, is that Byzantine-robust approaches only have quite limited success in providing theoretical convergence guarantees [33] since they usually require strong assumptions (such as subexponential [42] and sub-Gaussian [43]) on the distribution of local gradients. These assumptions often fall short of proper justifications, and they become even harder to justify when training a distributionally robust model as opposed to ERM.

A. Our Work

In this article, we aim to fill this gap by proposing a robust distributed learning algorithm that is resilient to both distributional shifts and Byzantine attacks. To address the aforementioned first obstacle, we utilize a recent work on Wasserstein distributionally robust optimization (DRO) [22], which leads to a reformulation that can be solved in a distributed fashion (see Section II-A1). To bypass the second obstacle, we implement norm-based screening (NBS), a simple robust aggregation scheme. We formulate a robust property of NBS, which enables us to avoid making unjustified assumptions on local gradients while providing convergence guarantees (see Section III). These two adopted techniques equip our algorithm with robust features against distributional shifts and Byzantine attacks. We further derive theoretical convergence guarantees of the proposed algorithm for nonconvex, convex, and strongly convex learning problems. We note that these convergence guarantees are built upon the theoretical robust property of NBS, which differentiates NBS from other robust aggregation measures and makes it suitable for our theoretical framework. The theoretical results offer valuable insights into the convergence behaviors of our algorithm (see Section VI-B), the considerations in selecting certain parameters effectively (see Section VI-D), and the breakpoint of NBS (see Section VI-C). In particular, we point out the common misconception that the breakpoint of NBS is $(1/2)$ (of workers being abnormal) and correct it as $(1/3)$. We empirically verify our algorithm's effectiveness against both distributional shifts and Byzantine attacks on the Spambase dataset [56], through which the empirical superiority of NBS is illuminated. In addition, it is shown that our algorithm's outstanding performance is not sensitive to the selection of hyperparameters, which is desirable in practical implementations.

Our main contributions are summarized as follows.

1) We propose a new research direction aiming to level up the overall model robustness in distributed learning in which distributional shifts and Byzantine attacks are addressed jointly. To achieve this goal, we design a distributed learning algorithm with robust features against both robustness issues, the very first of its kind.
2) We provide convergence proofs for our algorithm on nonconvex, convex, and strongly convex learning problems, respectively, giving insights into our algorithm's convergence behaviors, endurability against Byzantine attacks, and parameter selection strategies.
3) For the first time, we debunk the widely held misconception that the breakpoint of NBS is $(1/2)$, and we deduce that it should have been $(1/3)$.
4) We conduct thorough experiments to explore and identify the scenarios in which NBS outperforms other widely implemented robust aggregation schemes. Specifically, we found that NBS enjoys distinct advantages in the challenging setting with heterogeneous dataset.

B. Notations

Throughout, the norm notation $\|\cdot\|$ refers to the L_2 norm if not otherwise specified.

II. Related Work

A. Distributionally Robust Optimization

To combat distributional shifts, the conventional approach is robust optimization where the hypothetical data shifts are restricted to be within a deterministic uncertainty set [9], [10], [11], [12], and the goal is to find the optimal model for the worst case set of data. However, these works are found to be intractable except for specially structured losses [22] and they tend to promote overconservative solutions [19]. DRO, on the other hand, treats the data uncertainty in a probabilistic way and has been the more favored approach to dealing with distributional shifts, due to its appealing theoretical guarantees [20], computational tractability, and extraordinary empirical performance [21].

The goal of DRO is to find a model θ that minimizes the worst case expected loss $\sup_{Q\in\Omega} \mathbb{E}_{x\sim Q} f(\theta; x)$ over an ambiguity set Ω, which encompasses a cluster of data distributions. In practice, Ω is constructed based on the information of $\hat{P}_N$, the empirical distribution of training data. If Ω is selected judiciously such that it is able to capture the test data distribution (under reasonable levels of perturbation), then the solution θ_{DRO} is guaranteed to have robust out-of-sample performance. Meanwhile, we want to make Ω small enough to exclude irregular distributions that are not representative of the test data and incentivize overconservative results. Note that DRO reduces to ERM when Ω shrinks to a singleton $\hat{P}_N$.

Previous works have considered constructing Ω based on moment conditions [15], [16], as well as probability distance measures such as f-divergence [17], [18] and Wasserstein distance [19], [20], [21], [22]. Although many of these works demonstrate appealing theoretical guarantees and computational tractability, most of them do not admit a distributed implementation as explained previously. To this end, we resort to the Wasserstein DRO framework in [22], which not only admits a reformulation that is solvable in the distributed setting but also provides certified robustness under moderate levels of distributional shifts.

1) Wasserstein DRO: In Wasserstein DRO, the ambiguity set Ω is chosen as a Wasserstein ball $B_\rho(\hat{P}_N) = \{Q : W_c(Q, \hat{P}_N) \leq \rho\}$ with $\hat{P}_N$ at the center and ρ being the radius, and $W_c(\cdot,\cdot)$ is the Wasserstein distance between two probability distributions with $c(\cdot,\cdot)$ being the transportation cost between two data points. Following a duality result, Sinha et al. [22, Proposition 1] prove the equality $\sup_{Q\in B_\rho(\hat{P}_N)} \mathbb{E}_{x\sim Q} f(\theta; x) = \inf_{\lambda\geq 0}\{\lambda\rho + \mathbb{E}_{x\sim\hat{P}_N}\phi_\lambda(\theta; x)\}$, where $\phi_\lambda(\theta; x) = \sup_z\{f(\theta; z) - \lambda c(z, x)\}$ represents the robust surrogate of $f(\theta; x)$ and λ is the dual variable. Relaxing the original problem with a prespecified ρ, i.e.,

$$\min_\theta \sup_{Q:W_c(Q,\hat{P}_N)\leq\rho} \mathbb{E}_{x\sim Q} f(\theta; x). \tag{1}$$

Sinha et al. [22] instead seek to solve an easier problem

$$\min_\theta \mathbb{E}_{x\sim\hat{P}_N}\phi_\lambda(\theta; x) \tag{2}$$

with a fixed $\lambda \geq 0$. Note that when λ approaches infinity, (2) boils down to ERM. To justify the switch from (1) to (2), Sinha et al. [22] provide a certificate of robustness for any ρ in (1), which corresponds to a specific λ by duality. In this way, the original infinite-dimensional optimization problem (1) is transformed into a tractable ERM-like problem (2). Moreover, since the objective function in (2) is simply the average of a cluster of empirical losses each defined by a single sample x in the training set, (2) immediately admits a distributed implementation where each worker can calculate gradient-like updates based on its own local data. This attribute differentiates (2) from other centralized DRO reformulations and makes it suitable for our work.

2) Distributed Implementation of DRO: Even though most conventional solutions of DRO tend not to be parallelizable/decomposable, a few works take the initiative to implement DRO in the distributed setting. For example, Sadeghi et al. [23] also consider the Wasserstein ambiguity set and propose a framework called distributionally robust federated learning (DRFL) where the dual variable λ is simultaneously updated with the model parameter θ during the outer minimization step. Subsequently, Shen et al. [24] consider DRFL in an adversarial setting with Byzantine attacks and propose a two-stage attack strategy based on reinforcement learning to jeopardize the performance of DRFL. Our work differs from these two works in that we consider the Byzantine issue from the defender's point of view and we establish certified robustness accordingly.

In order to address concerns of data heterogeneity in federated learning, Mohri et al. [25] propose agnostic federated learning (AFL), a minimax optimization scheme whose ambiguity set is formed by a mixture of local distributions with the weight vector $\mathbf{w}$ confined to a regular simplex. Mohri et al. [25] show that AFL naturally promotes a sense of fairness by minimizing the training loss of the worst-off client among all workers. To alleviate the huge communication overhead of AFL and enhance its scalability, Deng et al. [26] propose a communication-efficient algorithm called distributionally robust federated averaging (DRFA) by infrequently updating the weight vector $\mathbf{w}$. Note that although works [25], [26] fall under the category of DRO, their motivation differs from ours since they try to prevent the model from overfitting any specific worker and thus exhibit poor generalization performance. In addition, these two works do not consider Byzantine attacks as does in this article.

B. Byzantine-Robust Distributed Learning

Under the default protocol of distributed learning, the server takes a simple average of the gradients collected from local nodes, some of which might be malicious. Since a Byzantine device can send any message containing arbitrary values, its influence on the aggregated sum, hence the deviation of the updated model, cannot be upper bounded, thus inflicting unbounded harm. In recent years, a host of works are proposed to tackle this issue, and we observe a common theme throughout these works. Intuitively, the retrievability of untainted global gradient in the presence of corrupted local gradients clearly indicates the existence of redundant information within the system. Based on different sources of redundancy, existing works aimed at achieving Byzantine robustness can be largely classified into three categories [29]: coding-based schemes, reference-based schemes, and robust aggregation schemes, which are introduced as follows.

Coding-based schemes assign each worker redundant data and rely on this redundancy to neutralize the effect of erroneous gradients [32], [33], [34]. Such an idea is originally proposed to mitigate stragglers so that the true global gradient can be computed even if some workers fail to report their local gradients to the server [31] and subsequently evolves into other stronger variants such as DRACO [32] and DETOX [33] that are able to handle Byzantine workers. Although this approach can recover the global gradient precisely, it does so at the cost of prohibitively high computational overhead [54]. Furthermore, in federated learning where data are generated locally and cannot be replicated and reassigned for user privacy, coding-based schemes are not applicable.

Reference-based schemes assume that the server has access to a set of auxiliary clean data that are similar to local data but smaller in size and use this auxiliary dataset as a reference to eliminate oddly behaving gradients that are potentially Byzantine. Typical examples in this category include Zeno [35], Cao and Lai [36], FLTrust [37], and ByGARS [38]. With the assistance of external information, this line of work can recover the global gradient even if over half of the workers are Byzantine. However, the auxiliary dataset is not always available, which restricts the applicability of this approach.

Robust aggregation schemes replace the averaging (of local gradients) step with a robust aggregation operation, such as Krum [39], Bulyan [40], geometric median [41], coordinate-wise median (CM) [42], iterative filtering [43], signSGD [44], and NBS [45], [46], [47], [48]. This line of work does not require redundantly coded local data or auxiliary data and it simply capitalizes on the redundancy (or homogeneity) within the honest gradients themselves. In the most homogeneous scenario where each worker has the same data, Byzantine gradients can be easily singled out since all the honest gradients are exactly the same. On the other hand, in case where the honest gradients all point to very different directions, offsetting Byzantine gradients becomes theoretically intractable since there is no exploitable redundancy. Due to its wide variety of aggregation rules and broad applicability, robust aggregation is commonly viewed as the mainstream approach to mitigating Byzantine attacks and is also the focus of this article. In practice, one has to select or create a feasible setting with sufficient amount of redundancy before implementing robust aggregation measures, which may require controlling the level of data heterogeneity and the percentage of Byzantine workers.

Algorithm 1 NBS

Input: $g_1, \ldots, g_m$ (m vector inputs), screening percentage β
Output: $G = \textbf{Norm\_Screen}_\beta(g_1, \ldots, g_m)$

1: generate a new set of indices $(1), \ldots, (m)$, such that $\|g_{(1)}\| \leq \cdots \leq \|g_{(m)}\|$
2: define an index set $\mathcal{U} = \{(1), \ldots, ((1-\beta)m)\}$, which specifies the unscreened inputs
3: calculate the output by averaging the unscreened inputs $G = \frac{1}{|\mathcal{U}|}\sum_{i\in\mathcal{U}} g_i$

III. Norm-Based Screening

As a robust aggregation measure, the idea of NBS is fairly simple: leave out the vector inputs (i.e., local updates) with large norms and take the average of the remaining inputs as output. In this way, the influence of an erroneous/malicious input is properly bounded. It is either filtered out for having a large norm or can only finitely impact the output with a norm comparable to some benign inputs. We formally define NBS as a function "**Norm_Screen**," as detailed in Algorithm 1.

Although NBS has previously been applied to screen Byzantine-prone local gradients [45], [46], [47] and Newton updates [48], we argue that NBS did not get its fair share of appreciation and publicity, partly because its robust property has not been formally stated and theorized. To fill this gap, we formulate an important property of NBS as explicated in Theorem 1, whose proof is deferred to Appendix A.

Theorem 1: Suppose that a percentage of $\alpha \leq (1/2)$ among m inputs $g_1, \ldots, g_m$ are Byzantine, whose indices compose a set $\mathcal{B}$ ($|\mathcal{B}| = \alpha m$), and the index set of honest inputs is denoted as $\mathcal{M}$ ($|\mathcal{M}| = (1-\alpha)m$). With $G = \textbf{Norm\_Screen}_\beta(g_1, \ldots, g_m)$ and $\beta \geq \alpha$, the following inequality holds:

$$\|G - S\| \leq \frac{2\alpha}{1-\beta}\|S\| + \max_{i\in\mathcal{M}} \|g_i - S\| \tag{3}$$

where S can be any vector with the same dimension as G.

Theorem 1 plays an essential role in the convergence analysis of our algorithm, as it properly upper bounds the distance between the robustly aggregated gradient and the true global gradient without making any unjustified assumptions on the distribution of local gradients (see Lemma 2). In addition, as shown in Section VI-C, our intuition on the breakpoint of NBS is drawn from Lemma 2, which is credited to the explicit exposition of Theorem 1.

We note that a similar result to (3) has appeared in the existing work [45, Sec. 9.1] as an intermediate step in the derivation, though being mixed with other terms. While giving [45] its due credit, we argue that a more formal statement of this property is well-deserved.

IV. Problem Statement

In this section, we formulate the problem of robust distributed learning under both distributional shifts and Byzantine attacks.

A. Basic Setting

We consider a typical distributed learning scenario with one central server and m parallel workers, among whom a total of N data points $x_1, \ldots, x_N$ are allocated/collected for training. For simplicity and clear exposition, we assume an even data-split scenario where worker i holds n samples $x_{(i-1)n+1}, \ldots, x_{(i-1)n+n}$ for $i = 1, \ldots, m$ with $mn = N$. Note that uneven data-split cases can easily fit into our framework with minor adjustment.

B. Learning Goal

Let $f(\theta; x_j)$ be the loss function contingent upon model parameter θ and sample x_j. We aim for a model that has robust performance on the test data, which may exhibit some degree of distributional shifts from the training data. According to the discussion in Section II-A1, such a model can be acquired by solving (2), whose solution enjoys theoretically proven robustness against data perturbations. Specifically, we seek to minimize the objective $F(\theta) = (1/N)\sum_{j=1}^{N} \phi_\lambda(\theta; x_j)$ in which $\phi_\lambda(\theta; x_j) = \sup_z\{f(\theta; z) - \lambda c(z, x_j)\}$ is the robustified version of $f(\theta; x_j)$.

C. Byzantine Attack

We assume that a percentage α of local workers are Byzantine and the remaining $1-\alpha$ are normal/honest. The sets of Byzantine workers and honest workers are denoted as $\mathcal{B}$ and $\mathcal{M}$, respectively, with $|\mathcal{B}| = \alpha m$ and $|\mathcal{M}| = (1-\alpha)m$. During each training iteration, the server would ask all workers to conduct certain computational task based on their respective local data and to report the result back to the server. While honest workers would follow the given instructions faithfully, Byzantine workers need not to obey the protocol and can send arbitrary messages to the server. By convention, we assume that Byzantine workers have complete knowledge of the system and learning algorithms, which allows them to generate the most damaging updates to attack the system.

V. Proposed Algorithm

On the macro level, our algorithm is based on distributed gradient descent combined with robust aggregation, through the following three key components.

A. Gradient Computation

We first consider a single unit of the objective function, i.e., $\phi_\lambda(\theta; x_j)$. To calculate its gradient on a fixed model θ_t, Sinha et al. [22] propose to first find the maximizer, i.e., $z_j^*(\theta_t) = \arg\sup_z\{f(\theta_t; z) - \lambda c(z, x_j)\}$, and then take the gradient of $f(\theta; z_j^*(\theta_t))$ before replacing θ with θ_t. The correctness of this approach is guaranteed by the following equation:

$$\begin{aligned}\nabla_\theta \phi_\lambda(\theta; x_j)|_{\theta=\theta_t} &= \nabla_\theta\big[f\big(\theta; z_j^*(\theta_t)\big) - \lambda c\big(z_j^*(\theta_t), x_j\big)\big]_{\theta=\theta_t} \\ &= \nabla_\theta f\big(\theta; z_j^*(\theta_t)\big)|_{\theta=\theta_t}.\end{aligned} \tag{4}$$

To simplify notations, we denote $\nabla_\theta f(\theta; z)|_{\theta=\theta_t}$ as $\nabla_\theta f(\theta_t; z)$ where z, sometimes taking the form of $z(\theta_t)$, is always treated as a constant in the differentiation step.

B. ε-Approximation

In most cases, the maximizer $z_j^*(\theta_t)$ does not have a closed-form solution and thus can only be solved to a certain precision via iterative methods. Therefore, we only require workers to obtain an ε-optimal maximizer $z_j^\varepsilon(\theta_t)$, satisfying $\|z_j^\varepsilon(\theta_t) - z_j^*(\theta_t)\| \leq \varepsilon$. This approximation offers a tradeoff between computational cost and model accuracy. In Section VI, we will analyze both the effects of ε-approximation on model convergence and the cost of obtaining such an ε-optimal maximizer.

Algorithm 2 Distributional and Byzantine-Robust Distributed Gradient Descent

Input: screening percentage β ($\geq \alpha$), learning rate η, model initialization θ_0, total iteration T

Output: completed model θ_T

1: **for** $t = 0, 1, \ldots, T-1$ **do**
2: **Server**: send θ_t to all workers
3: **for** $i = 1, 2, \ldots, m$ **do**
4: **Worker** i: receive model θ_t from the server
5: obtain $z_j^{\varepsilon}(\theta_t)$ for each local sample x_j by solving $\sup_z\{f(\theta_t; z) - \lambda c(z, x_j)\}$ to ε-precision
6: compute local gradient $g_i(\theta_t) = \begin{cases} \frac{1}{n}\sum_{j=(i-1)n+1}^{(i-1)n+n} \nabla_\theta f(\theta_t; z_j^{\varepsilon}(\theta_t)) & i \in \mathcal{M} \\ \star & i \in \mathcal{B} \end{cases}$
7: send $g_i(\theta_t)$ to the server
8: **end for**
9: **Server**: collect $g_1(\theta_t), \ldots, g_m(\theta_t)$ from the workers
10: compute the aggregated gradient $G(\theta_t) =$ **Norm_Screen**$_\beta(g_1(\theta_t), \ldots, g_m(\theta_t))$
11: update model $\theta_{t+1} = \theta_t - \eta \cdot G(\theta_t)$
12: **end for**

C. Robust Aggregation

After obtaining the ε-optimally perturbed samples, each honest worker computes its (approximate) local gradient before sending it to the server, while Byzantine workers would craft their own ill-intended gradients (denoted as $\star$). On the other end, the server robustly aggregates the received local gradients via NBS and uses the result to update the model. Here, we assume that the proportion α of Byzantine workers is known, and we always enforce that $\beta \geq \alpha$.

The detailed procedure of our algorithm is given in Algorithm 2.

VI. Convergence Analysis

A. Preliminaries

To delineate the convergence behavior of the proposed algorithm, we adopt some widely used assumptions as follows. Assumptions 1–3 concern the distributional shifts as in [22], which hold for tractable scenarios. Assumption 4 upper bounds the distance between the average gradient and the gradient with respect to a single sample, which is characteristic of gradient averaging methods and clearly holds.

Assumption 1: The loss function $f(\theta; z)$ satisfies the Lipschitzian smoothness conditions

$$\begin{aligned} \|\nabla_\theta f(\theta_1; z) - \nabla_\theta f(\theta_2; z)\| &\leq L_{\theta\theta}\|\theta_1 - \theta_2\| \\ \|\nabla_\theta f(\theta; z_1) - \nabla_\theta f(\theta; z_2)\| &\leq L_{\theta z}\|z_1 - z_2\| \\ \|\nabla_z f(\theta_1; z) - \nabla_z f(\theta_2; z)\| &\leq L_{z\theta}\|\theta_1 - \theta_2\| \\ \|\nabla_z f(\theta; z_1) - \nabla_z f(\theta; z_2)\| &\leq L_{zz}\|z_1 - z_2\|. \end{aligned}$$

Assumption 2: The function $c(z, x)$ defined in the Wasserstein metric is L_c-smooth and 1-strongly convex with respect to z.

Assumption 3: The dual variable λ satisfies $\lambda > L_{zz}$, where L_{zz} is defined in Assumption 1.

Assumption 4: For any specific θ_t, it holds that

$$\max_{1\leq k\leq N} \left\| \nabla_\theta f\left(\theta_t; z_k^*(\theta_t)\right) - \frac{1}{N}\sum_{j=1}^{N} \nabla_\theta f\left(\theta_t; z_j^*(\theta_t)\right) \right\| \leq \sigma. \quad (5)$$

Based on the above assumptions, we formulate two lemmas that will serve as core building blocks of the ensuing theorems on convergence. We should note that Lemma 1 is a direct result of [22, Lemma 1]. For completeness, we summarize the proof of Lemma 1, matching the notations of this article.

Lemma 1: Under Assumptions 1–3, the objective function $F(\theta) = (1/N)\sum_{j=1}^{N} \phi_\lambda(\theta; x_j)$ is L_F-smooth with $L_F = L_{\theta\theta} + (L_{\theta z}L_{z\theta}/(\lambda - L_{zz}))$.

Lemma 1 specifies the smoothness level of the objective function, thus allowing standard gradient descent to make steady progress with a proper step size, such as $(1/L_F)$. However, in our problem, the error-free gradient is unattainable due to the Byzantine nodes. To this end, we propose Lemma 2 that quantifies the deviation of our implemented gradient from the true global gradient $\nabla F(\theta_t)$. The proof of Lemma 2 is deferred to Appendix C.

Lemma 2: Under Assumptions 1–4, for any specific θ_t, it holds that

$$\|G(\theta_t) - \nabla F(\theta_t)\| \leq \frac{2\alpha}{1-\beta}\|\nabla F(\theta_t)\| + (L_{\theta z}\varepsilon + \sigma) \quad (6)$$

where $G(\theta_t)$ is the aggregated gradient in Algorithm 2 (line 10).

B. Main Theorems

1) Nonconvex Losses: We first consider the most general case of the loss function $f(\theta; z)$ being nonconvex in θ, such as in neural network training. For this case, we derive Theorem 2 that guarantees convergence of our algorithm to a stationary point of the objective function. The proof of Theorem 2 is deferred to Appendix D.

Theorem 2: Suppose that Assumptions 1–4 hold and $\alpha < (1/3)$. Taking $\eta = (1/L_F)$, Algorithm 2 satisfies

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F(\theta_t)\|^2 \leq \frac{2L_F}{(1-(1+r)C_\alpha^2)T}[F(\theta_0) - F(\theta^*)] + \frac{(1+1/r)(L_{\theta z}\varepsilon + \sigma)^2}{1-(1+r)C_\alpha^2} \quad (7)$$

where θ^* is the global minimizer of $F(\theta)$, $C_\alpha = (2\alpha/(1-\beta))$, and r should satisfy $0 < r < ((1-\beta)/2\alpha)^2 - 1$.

2) Convex Losses: Now, we consider the case where the loss function is convex as in Assumption 5. In addition, we make Assumption 6 suggesting that all the intermediate iterations would not be infinitely worse than the initialization θ_0. We propose Theorem 3 that grants our algorithm convergence guarantee in the convex regime. The proof of Theorem 3 is deferred to Appendix E.

Assumption 5: The loss function $f(\theta; z)$ is convex with respect to θ.

Assumption 6: There exists a fixed k such that $\|\theta_t - \theta^*\| \leq k\|\theta_0 - \theta^*\|$ holds for $t = 0, 1, \ldots, T-1$.

Theorem 3: Suppose that Assumptions 1–6 hold and $\alpha < (1/3)$. Taking $\eta = (1/L_F)$, Algorithm 2 satisfies

$$\begin{aligned} &F(\theta_T) - F(\theta^*) \\ &\leq \max\left\{ \frac{4L_F D^2}{(1-(1+r)C_\alpha^2)T}, \sqrt{\frac{2(1+1/r)}{1-(1+r)C_\alpha^2}}D(L_{\theta z}\varepsilon + \sigma) \right. \\ &\quad \left. + \frac{(1+1/r)(L_{\theta z}\varepsilon + \sigma)^2}{2L_F} \right\} \quad (8) \end{aligned}$$

where $D = k\|\theta_0 - \theta^*\|$ and C_α and r are the same as in Theorem 2.

3) Strongly Convex Losses: Finally, we assume strong convexity on the objective function as in Assumption 7, in which case we propose Theorem 4 that guarantees convergence of our algorithm to the optimal model θ^*. The proof of Theorem 4 is deferred to Appendix F.

Assumption 7: The objective function $F(\theta)$ is λ_F-strongly convex.

Theorem 4: Suppose that Assumptions 1–4 and 7 hold, and $\alpha < (1/(1+2L_F/\lambda_F)) < (1/3)$. Taking $\eta = (2/(L_F+\lambda_F))$, Algorithm 2 satisfies (with $C_\alpha = (2\alpha/(1-\beta))$)

$$\|\theta_T - \theta^*\| \leq \left(\frac{2L_F C_\alpha + L_F - \lambda_F}{L_F + \lambda_F}\right)^T \|\theta_0 - \theta^*\| + \frac{L_{\theta z}\varepsilon + \sigma}{\lambda_F - L_F C_\alpha}. \tag{9}$$

Observations: According to Theorems 2–4, Algorithm 2 is able to achieve some sense of convergence under all three cases. Meanwhile, we can clearly identify the effects of Byzantine percentage α and suboptimality level ε on convergence; a larger α (entailing larger C_α) not only decreases convergence speed but also increases convergence error, whereas ε only affects the convergence error and has no impact on the convergence rate.

C. Breakpoint of NBS

We define the breakpoint of a certain algorithm as the minimum Byzantine percentage at which that algorithm cannot converge. According to Theorems 2–4, the breakpoint of our algorithm is (1/3). [Although Theorem 4 requires that $\alpha < (1/(1+2L_F/\lambda_F))$, it can still converge as in (7) under $\alpha < (1/3)$ by taking $\eta = (1/L_F)$.] In fact, based on the derivations in Appendixes D and F, we assert that for any algorithm that incorporates NBS to converge, it always should hold that $\alpha < (1/3)$. This insight can be drawn from Lemma 2, where the distance between $G(\theta_t)$ and $\nabla F(\theta_t)$ is upper bounded by two terms. In the convergence proofs, we found that the coefficient of the first term must be less than 1, i.e., $(2\alpha/(1-\beta)) < 1$, which, combined with $\beta \geq \alpha$, imposes that $\alpha < (1/3)$.

We also notice that previous works implementing NBS for Byzantine robustness uniformly claimed that the breakpoint of their algorithms is (1/2) [45], [46], [47], [48]. The fallacy of this claim can be illustrated by a simple counterexample; suppose that there are four Byzantine updates g_1, g_2, g_3, and g_4 and six honest updates $g_5, g_6, \ldots, g_{10}$ (in norm-descending order). If the Byzantine updates are crafted such that $g_1 = g_2 = g_3 = g_4 = -g_9$, then the NBS output with $\beta = \alpha = 0.4$ is totally dominated by Byzantine updates as $G = ((g_1 + g_2 + g_3 + g_4 + g_9 + g_{10})/6)$. If this happens for every iteration, then the algorithm surely would not converge to the correct solution.

To the best of our knowledge, our claim that the breakpoint of NBS is (1/3) is new in the literature.

D. Discussion

1) Cost of Computing ε-Optimal Maximizer: To obtain each perturbed sample $z_j^\varepsilon(\theta_t)$, we need to maximize $g(z) = f(\theta_t, z) - \lambda c(z, x_j)$ (at fixed θ_t and x_j) to ε-optimality. It turns out that $g(z)$ is both smooth and strongly concave, which, according to optimization theory, suggests that maximizing $g(z)$ enjoys a linear convergence rate using a gradient method. Specifically, according to Assumption 1, $-L_{zz} \cdot \mathbf{I} \preceq \nabla_z^2 f(\theta; z) \preceq L_{zz} \cdot \mathbf{I}$; according to Assumption 2, $1 \cdot \mathbf{I} \preceq \nabla_z^2 c(z, x) \preceq L_c \cdot \mathbf{I}$. Therefore, $-(\lambda L_c + L_{zz}) \cdot \mathbf{I} \preceq \nabla_z^2 g(z) \preceq -(\lambda - L_{zz}) \cdot \mathbf{I}$, which means that $g(z)$ is L_g-smooth and λ_g-strongly concave with $L_g = \lambda L_c + L_{zz}$ and $\lambda_g = \lambda - L_{zz}$. According to the convergence analysis similar to that of Appendix F, by iterating $z_{t+1} = z_t + \eta_z \nabla g(z_t)$ with $\eta_z = (2/(L_g + \lambda_g)) = (2/(\lambda L_c + \lambda))$, we have $\|z_T - z^*\| \leq p^T \|z_0 - z^*\|$ with the exact maximizer z^* and the convergence factor $p = (L_g - \lambda_g)/(L_g + \lambda_g) = (2L_{zz} + \lambda L_c - \lambda)/(\lambda L_c + \lambda)$. As a result, to obtain an ε-optimal solution z^ε satisfying $\|z^\varepsilon - z^*\| \leq \varepsilon$ requires that $T_z \geq (\ln(D_z/\varepsilon)/\ln(1/p))$, where T_z is the number of gradient ascent iterations and $D_z = \|z_0 - z^*\|$. Here, we show that ε is easily adjustable by tuning T_z, and a smaller ε only entails a moderate increase in T_z.

2) Strategy of Adjusting ε: According to our analysis, small ε corresponds to low model error. On the other hand, enforcing small ε puts a relatively heavy computational workload on local workers. To achieve a good tradeoff between computational cost and model accuracy, we recommend a two-stage strategy where a big ε is adopted in the beginning stage of training and a small ε is enforced in the ending stage. This is because, in the beginning stage, the Byzantine gradients would contaminate the aggregated gradient to a large degree, and there is no need for honest workers to calculate their perturbed data/local gradients with very high precision. To make it clearer, we refer to (6), where $(2\alpha/(1-\beta))\|\nabla F(\theta_t)\|$ is the dominant term at the beginning, and therefore, a relatively big ε would have little impact on the converging process. In the ending stage where $\nabla F(\theta_t)$ approaches zero, $L_{\theta z}\varepsilon + \sigma$ becomes the dominant term, and we switch into a small ε regime to achieve high model accuracy, at the cost of concentrated computation in the end.

3) Effects of Non-Independent and Identically Distributed Data: Recall that in Section IV where the targeted problem is formulated, we did not assume that the data are independent and identically distributed (i.i.d.) across all workers, suggesting that our convergence results should hold in non-i.i.d. cases as well. This is due to the analytical approach we take in the proof of Lemma 2, where we bound the maximal distance between honest local gradients and the targeted gradient as $\max_{i\in\mathcal{M}} \|g_i(\theta_t) - \nabla F(\theta_t)\| \leq L_{\theta z}\varepsilon + \sigma$, regardless of local data distribution. In practice, this distance should increase if the distribution of local data goes from i.i.d. to highly pathological/non-i.i.d. However, for the convenience of analysis, this subtle difference is erased through the adoption of a universal upper bound $L_{\theta z}\varepsilon + \sigma$. There are two major takeaways from this observation. On the one hand, one should be aware that our theoretical results may not reflect the empirical effects of non-i.i.d. data on the convergence since our emphasis is on the effects of α and ε. On the other hand, this analytical approach we adopted, i.e., making assumptions in the spirit of Assumption 4 and imposing a universal upper bound to eliminate local updates' differences, might serve as a pathway for future works to bypass the non-i.i.d. issue theoretically in the convergence analysis.

VII. Simulations

A. Experimental Setup

1) Dataset and Allocation: In this section, we empirically evaluate the performance of our algorithm for a classification

task using a logistic regression model on the Spambase dataset [56]. We assign (2/3) of the 4601 total samples for training and the other (1/3) for testing. We consider a distributed learning setup with $m = 20$ workers, among which $\alpha m = 6$ are Byzantine. For training data allocation, we consider both the i.i.d. setting, which is most commonly seen in the literature, and the non-i.i.d. setting, which may better resemble real-world situations such as federated learning. For the i.i.d. setting, the training set is randomly permutated before being evenly split among the 20 workers. To simulate the non-i.i.d. setting, we first divide the training set into spam emails (labeled as 1) and nonspams (labeled as 0), which are then evenly split among two groups of workers numbering 8 and 12, respectively. Although in theory, there are numerous ways to create a non-i.i.d. setting, splitting data according to their labels seems to be the common practice. In our non-i.i.d. setting, we assume that half of the Byzantine workers are from each worker group. Upon the completion of training, we use the percentage of correct classifications on the test set as the performance metric of model accuracy.

2) Byzantine Model: To simulate Byzantine gradients, we experiment with four different strategies, i.e., sign-flipping attack [52], label-flipping attack [42], inner product manipulation (IPM) attack [53], and "A Little Is Enough" (ALIE) attack [54]. These four types of Byzantine model are commonly considered in the literature [29], [30].

In sign-flipping attack, each Byzantine worker flips the direction of the authentic local gradient and increases its magnitude by a constant factor, which is set as 2 in our experiments. The label-flipping attack generates Byzantine gradients using local data on each node with flipped labels, i.e., spams switching to nonspams and vice versa. It is reported in [42] that gradients computed with flipped labels have moderate values, which may make them less conspicuous to outlier filters. The IPM attack aims for the negative inner product between the mean of honest gradients and the output of certain aggregation schemes so that the iterated model is not moving toward a descending direction. To achieve this, Xie et al. [53] propose to craft Byzantine gradients as $-(\epsilon/|\mathcal{M}|)\sum_{i\in\mathcal{M}} g_i$, where $\epsilon > 0$ controls the strength of the attack and has different effects when taking small and large values. In our experiments, we consider two regimes of $\epsilon = 0.5$ and $\epsilon = 50$. The ALIE attack seeks to engineer Byzantine gradients with similar statistical features to honest gradients by exploiting the empirical variance among them. To do so, Baruch et al. [54] assume that the honest gradients follow a normal distribution for each coordinate $i \in [d]$, and the mean u_i and standard deviation σ_i are calculated empirically based on honest gradients to craft the Byzantine value as $u_i - z\cdot\sigma_i$ where $z = \phi^{-1}(((1-\alpha)m - s)/((1-\alpha)m))$ with $s = \lfloor (m/2)+1 \rfloor - \alpha m$ and $\phi^{-1}(\cdot)$ being the normal inverse cumulative distribution function. Note that both sign-flipping attack and label-flipping attack only use the information available to the local nodes controlled by the adversary. In contrast, the IPM attack and ALIE attack require access to all the honest gradients, which obviously faces greater challenges in practice.

3) Shift Model and Training Perturbation: To simulate distributional shifts, we follow [22] and perturb the test data with a controlled budget q under the common L_1, L_2, and L_∞ norms. Since in supervised learning, it is a common practice to only perturb the feature vector x, not the label y, we perturb each data point (x, y) into (z, y) satisfying $\|z - x\|_p \leq q$ ($p = 1, 2, \infty$) and the z's are chosen to maximally increase the cross-entropy loss on test data. In this way, the test stage perturbation is tailored to the iterated model θ_t, the type of shift L_p, and shift budget q.

In the training phase, the perturbed samples are obtained by approximately solving $\sup_z\{f(\theta; z) - \lambda c(z, x)\}$, in which we set $c(z, x) = (1/2)\|z - x\|^2$ as in [22] to satisfy Assumption 2. In accordance with the adversarial perturbation on test data, we only perturb the feature vector x into z without changing the label y. For logistic regression, the augmented objective is $g(z) = -y \ln a - (1-y)\ln(1-a) - (\lambda/2)\|z - x\|^2$ with $a = 1/(1 + e^{-\theta^{\mathrm{T}} z})$, which has no closed-form solution. Therefore, we calculate the approximate maximizer via gradient ascent $z_{t+1} = z_t + \eta_z \nabla g(z_t)$ using T_z iterations initialized at $z_0 = x$. Throughout, we set $\lambda = 3$, $\eta_z = 0.05$, and $T_z = 10$ if not otherwise specified. These parameter selections are not rigid and we show that a wide range of parameters allow similar stable performance.

B. Performance of NBS

We first seek to understand the characteristics of NBS, the situations in which NBS thrives or collapses, and its comparison to other state-of-the-art robust aggregation schemes. To this end, we use the most noted and compared aggregation measures Krum [39], CM, and coordinate-wise trimmed mean (CTM) [42], as benchmarks, including the plain averaging (AVG). For Krum, CTM, and NBS, the screening/trimming parameter is chosen according to the number of Byzantine nodes, which is assumed as known. Note that we do not consider a distributional shift in this part since our first goal is to explore the role of NBS. Accordingly, we omit step 5 in Algorithm 2 and also for other schemes.

In Fig. 1, we plot the performance curves of the five considered algorithms under various Byzantine attacks in the i.i.d. setting. From Fig. 1(a), we can see that in the absence of Byzantine workers, all robust schemes but Krum perform as well as AVG, despite the partial gradient information that they discard for the purpose of robustness. This indicates a relatively high level of redundancy among the local gradients in the i.i.d. setting. Due to this rich redundancy, all robust schemes perform reasonably well under all kinds of attacks, and there is no absolute distinction between NBS and other robust measures. The slightly inferior performance of Krum in Fig. 1(a) is also reported in existing literature such as [28], which reveals its inability to fully take advantage of the redundancy since it only selects a single gradient among multiple available copies without any averaging. Fig. 1 shows that AVG performs well under label-flipping attack, IPM attack with small ϵ, and ALIE attack. This indicates that in those cases, the engineered gradients may have moderate values, which makes them easily balanced out by the honest gradients (14 versus 6). On the other hand, AVG collapses under sign-flipping attack and IPM attack with large ϵ, suggesting that the engineered gradients are very aggressive and likely have large norms. As a result, NBS thrives in these two cases by outperforming other schemes since Byzantine gradients are directly eliminated by NBS due to their large norms.

Interestingly, as shown in Fig. 1(f), the performance of Krum is even enhanced under ALIE attack compared to the case of no attack. The likely explanation is that the shifting parameter z, which according to [54] equals 0.37, is too small to cause any damage. Given that the ALIE attack crafts the

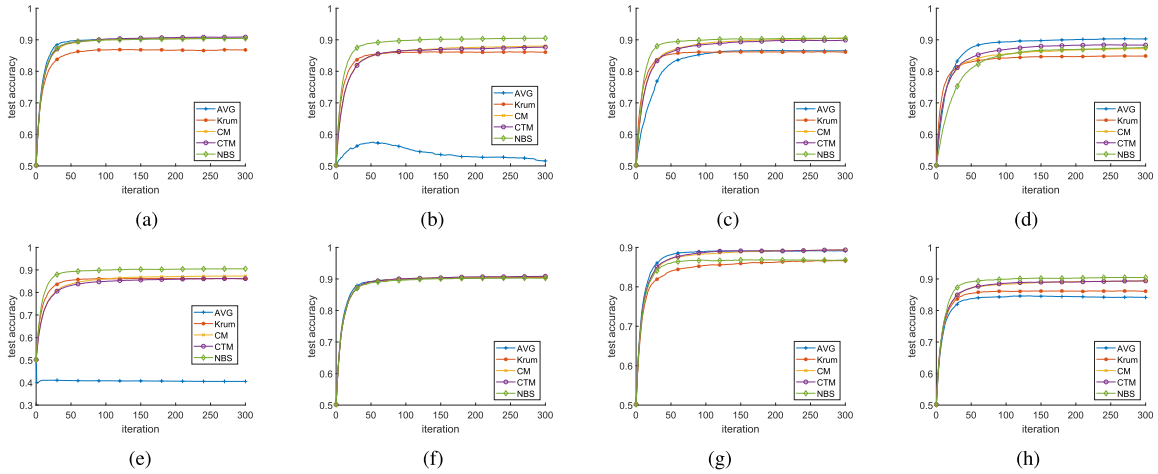


Fig. 1. Performance comparisons of different robust aggregation schemes under various attacks in the i.i.d. setting. (a) No attack. (b) Sign-flipping attack. (c) Label-flipping attack. (d) IPM attack ($\epsilon = 0.5$). (e) IPM attack ($\epsilon = 50$). (f) ALIE attack. (g) ALIE attack ($z = 1.5$). (h) ALIE attack ($z = 3$).

TABLE I

IMPACT OF VARIOUS ATTACKS ON ALGORITHM 2 AND ITS COUNTERPARTS

	DRO	Algorithm 2	DRO + CTM	DRO + CM	DRO + Krum
No attack	**90.5%**	85.5%	88.7%	61.2%	61.2%
Sign-flipping	58.5%	**89.0%**	61.1%	61.2%	61.1%
Label-flipping	**83.2%**	77.8%	62.3%	61.1%	61.1%
IPM ($\epsilon = 0.5$)	**89.7%**	65.1%	60.7%	50.8%	57.0%
IPM ($\epsilon = 50$)	34.6%	**89.1%**	66.3%	52.0%	61.1%
ALIE ($z = 3$)	71.6%	**89.7%**	84.3%	42.5%	61.1%

gradient element as $u_i - z \cdot \sigma_i$, using an overly small z may inadvertently serve as the averaging operation for Krum and thus elevate its performance. To make ALIE more effective as an attack, we increase z to 1.5 and 3. In the case of $z = 1.5$, NBS slightly underperforms AVG because the Byzantine gradients likely bypass NBS and are aggregated with fewer honest gradients than in AVG. However, if Byzantine gradients become even more aggressive as in the case of $z = 3$, they are not able to bypass NBS anymore, which exhibits a uniform superior performance against all aggressive attacks. Note that our modification on ALIE is not without precedent. In the original paper, Baruch et al. [54] draw Table I in a setting ($m = 51$ and $\alpha = 24\%$) where the value of z should have been 0.36 according to their formula. Instead, they use $z = 1.5$, which degrades the test accuracy of AVG from 96.1% to 91.1%. In comparison, we use 30% of Byzantine workers with $z = 3$ to degrade the test accuracy of AVG from 90.5% to 84.2%.

Next, we consider the non-i.i.d. setting and plot the corresponding performance curves in Fig. 2. In this setting, honest gradients are more heterogeneous than in the i.i.d. case, which leads to worse performance for robust schemes across the board. Specifically, Fig. 2(a) shows that both Krum and CM collapse even in the absence of attack. This phenomenon is previously observed in [30], and the authors conclude that the failure of Krum and CM is due to that they attempt to pick a single representative gradient, which may not exist in the non-i.i.d. setting. In comparison, CTM and NBS exhibit better overall performance because they both employ a combination of screening and averaging, making them better at exploiting the available redundancy. To compare these two, although CTM slightly outperforms NBS in a few cases (no attack, label-flipping attack, and ALIE attack with small z's), NBS enjoys more stable overall performance and it does not collapse under aggressive attacks such as sign-flipping attack and IPM attack with large ϵ. We want to point out that there is no such scheme that can outperform all other schemes under all attacks. The reliability of any aggregation scheme should hinge on its ability to prevent disastrous outcome from happening. This argument is also echoed in [54], which indicates that AVG (no defense), despite being least affected by the ALIE attack, cannot serve as the aggregation rule because of its serious vulnerabilities against aggressive attacks. In this sense, NBS appears to be the best/safest choice by a clear margin since it achieves a minimal test accuracy of 76% under all circumstances (except for IPM attack with small ϵ), compared to CTM with 56%, Krum with 46%, and CM with 39%. We exclude the case of IPM attack with small ϵ because no robust schemes can achieve a better accuracy than 65% despite AVG being unaffected, which suggests an infeasible task due to insufficient redundancy and high Byzantine percentage. For the task to be feasible in the first place, one needs to control the level of data heterogeneity and also restrict the percentage of Byzantine nodes.

Our previous experiments demonstrate the advantages of NBS over its counterparts at a fixed Byzantine percentage of 30%. To explore whether these results carry over to other cases with different Byzantine percentages, we plot the performance curves with varying numbers of Byzantine nodes (up to night out of 20) in both i.i.d. and non-i.i.d. settings in Fig. 3. Since some attacks share very similar results, such as sign-flipping attack and IPM attack with large ϵ, we only display the most

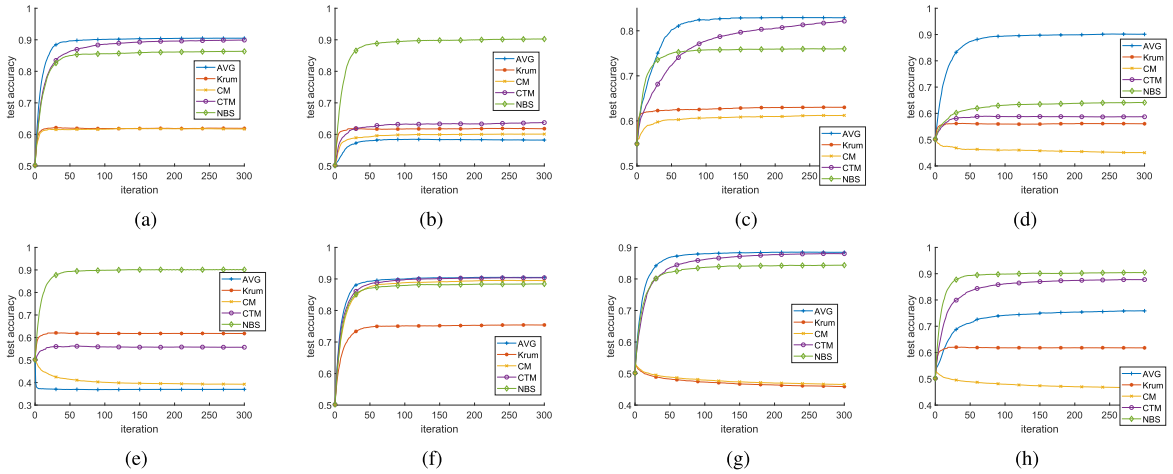


Fig. 2. Performance comparisons of different robust aggregation schemes under various attacks in the non-i.i.d. setting. (a) No attack. (b) Sign-flipping attack. (c) Label-flipping attack. (d) IPM attack ($\epsilon = 0.5$). (e) IPM attack ($\epsilon = 50$). (f) ALIE attack. (g) ALIE attack ($z = 1.5$). (h) ALIE attack ($z = 3$).

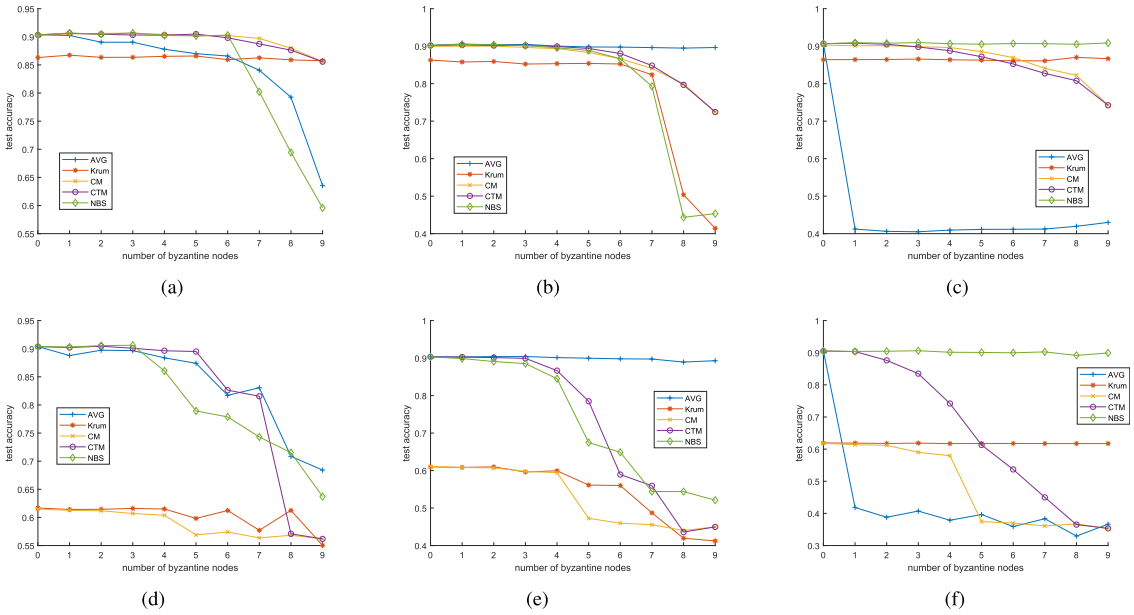


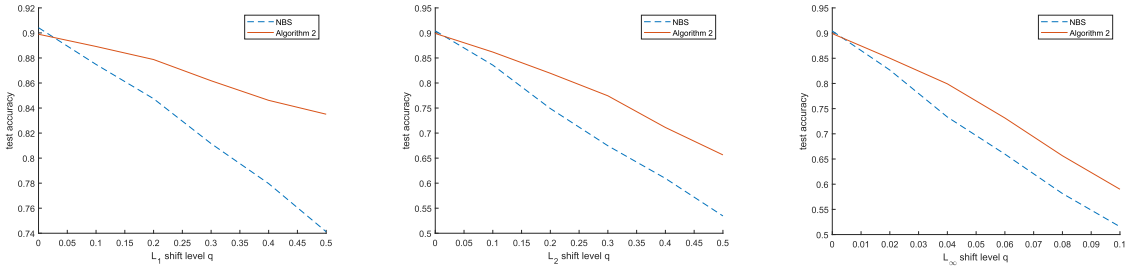
Fig. 3. Performance comparisons with varying numbers of Byzantine nodes in both i.i.d. and non-i.i.d. settings. (a) Label-flipping attack (i.i.d.). (b) IPM attack with $\epsilon = 0.5$ (i.i.d.). (c) IPM attack with $\epsilon = 50$ (i.i.d.). (d) Label-flipping attack (non-i.i.d.). (e) IPM attack with $\epsilon = 0.5$ (non-i.i.d.). (f) IPM attack with $\epsilon = 50$ (non-i.i.d.).

representative three attack regimes (label-flipping attack and IPM attacks with $\epsilon = 0.5$ and $\epsilon = 50$) to avoid redundant figures. We first consider the i.i.d. setting [Fig. 3(a)–(c)], where all robust schemes have very stable performance when $\alpha m \leq 6$. However, NBS starts to show drastically degraded performance when $\alpha m = 7$ under label-flipping attack, which corroborates our theoretical conclusion that α cannot exceed $(1/3)$ for NBS. In comparison, such a turning point for Krum is $\alpha m = 8$ under IPM attack with $\epsilon = 0.5$. For CM and CTM, no such turning point exists when less than half of workers are Byzantine. This result suggests that CM and CTM are the best robust measures when the Byzantine percentage is between $(1/3)$ and $(1/2)$, whereas NBS has the similar overall performance to other robust schemes in the case of $\alpha < (1/3)$ and enjoys marginal advantages under aggressive attacks [see Fig. 3(c)]. However, these guiding rules become invalid in the challenging non-i.i.d. setting, as shown in Fig. 3(d)–(f). Aside from the incompetence of Krum and CM, the theoretical breakpoints of CTM at $(1/2)$ and NBS at $(1/3)$ are no longer applicable due to reduced redundancy. Instead, the performance of either scheme quickly deteriorates when α exceeds 20% or so. Overall, NBS seems to be the preferred choice since it can tolerate up to four Byzantine nodes while guaranteeing 80% test accuracy in all cases. In comparison, CTM can only tolerate three Byzantine nodes to achieve the same goal.

In summary, in the i.i.d. setting with a high level of redundancy, NBS is a safe choice with comparable performance to other state-of-the-art schemes if $\alpha < (1/3)$, while median-based methods, such as CM and CTM, are preferred in the regime of $(1/3) < \alpha < (1/2)$. In the challenging non-i.i.d. setting, the theoretical breakpoints are not applicable and NBS is the safest choice over other robust aggregation schemes due to its ability to capitalize on available redundancy.

C. Combined Robustness

We now consider Algorithm 2, which is a combination of DRO and NBS, and seek to examine its robustness toward Byzantine attacks and distributional shifts, respectively. We first compare Algorithm 2 with its counterparts (by integrating DRO with other robust benchmarks) under various attacks in the non-i.i.d. setting, documenting their



(a) (b) (c)

Fig. 4. Performance curves under different levels of shifts and sign-flipping attack in the i.i.d. setting. (a) Shift defined by L_1 norm. (b) Shift defined by L_2 norm. (c) Shift defined by L_∞ norm.

performance in Table I. Comparing Table I and Fig. 2, we can see that Algorithm 2 largely inherits the Byzantine robustness of NBS. Specifically, excluding the intractable case of IPM with $\epsilon = 0.5$, the worse case accuracies for Algorithm 2 and the DRO-integrated versions of CTM/CM/Krum are 77.8% and 61.1%/42.5%/61.1%, respectively. Next, we focus on exploring the effectiveness of Algorithm 2 against various distributional shifts, which take place during the test stage. In the training stage, we choose the i.i.d. setting with six Byzantine workers conducting label-flipping attack. After training is completed, we measure the model performance on test data in the presence of distributional shifts with varying levels of budget q under L_1, L_2, and L_∞ norms. We experiment with q up to 0.5 for L_1 and L_2 shifts, and a smaller budget 0.1 for L_∞ shift since the latter is more potent and noticeable, e.g., when perturbing the pixels of test images. The performance curves of Algorithm 2 along with the benchmark NBS are plotted in Fig. 4 (note that there is no other benchmark that jointly deals with distributional shift and Byzantine failure as explained in Introduction). From Fig. 4, we observe that when there is no distributional shift ($q = 0$), Algorithm 2 and NBS have almost identical performance, which verifies that the robust features of NBS against label-flipping attack successfully transfer to Algorithm 2. As the shift budget q increases, the performance gap between Algorithm 2 and NBS gradually widens until it levels off at a certain point. This indicates that Algorithm 2 indeed inherits the distributional robustness of DRO and it is effective against different types of distributional shifts. Also, Fig. 4 shows that NBS, although being Byzantine-robust, is not immune from distributional shifts, which corroborates our previous claim that Byzantine robustness does not imply distributional robustness. However, with the proper incorporation of DRO, Algorithm 2 is able to achieve combined robustness at a negligible cost compared to NBS.

In [22], it is observed that models trained with the Euclidean cost $c(z, x) = (1/2)\|z - x\|^2$ can still provide robustness to L_∞ shift. We expand this result by showing their effectiveness against L_1 shift, which is not considered in [22].

D. Parameter Selection

Finally, we explore the influence of parameter selection on Algorithm 2. To see the impact of λ, which is the dual variable that has to be selected empirically according to [22], we run Algorithm 2 under all three shift regimes (L_1 shift with $q = 0.3$, L_2 shift with $q = 0.3$, and L_∞ shift with $q = 0.06$) with different values of λ and plot Fig. 5. Here, we set $T_z = 100$ to make the curves smooth. Fig. 5 demonstrates that the performance of Algorithm 2 is quite stable in a wide range of values for λ. However, Algorithm 2 fails when λ is too small. This is consistent with the claim in [22] that λ has to be large enough for $f(\theta; z) - \lambda c(z, x_j)$ to be strongly concave (also reflected in Assumption 3).

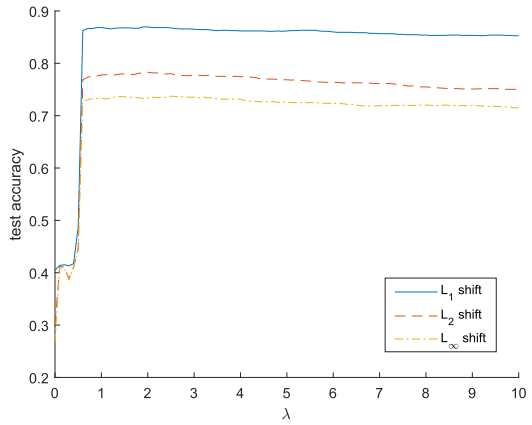


Fig. 5. Performance of Algorithm 2 with different λ's under three distributional shift regimes.

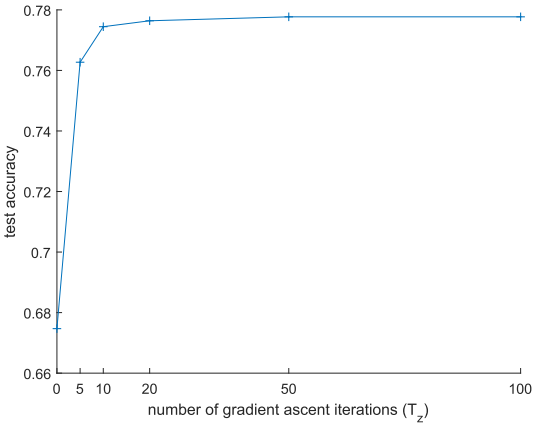


Fig. 6. Performance of Algorithm 2 with different T_z's under L_2 shift.

Moreover, we plot Fig. 6 to document the performance of Algorithm 2 under L_2 shift with $q = 0.3$ using different T_z's, the number of gradient ascent iterations to obtain the ε-optimal maximizer. From Fig. 6, we observe that achieving good performance does not require too many iterations ($T_z = 10$ suffices in this case). This result corroborates the first remark in Section VI-D that computing perturbed samples with high precision only requires a moderate increase in the computational cost. Figs. 5 and 6 suggest that our algorithm's effectiveness is not sensitive to the selection of hyperparameters, which is a desirable attribute in practice.

VIII. Conclusion

In this article, we address the uncharted problem of robust distributed learning in the presence of both distributional shifts

and Byzantine attacks. We propose a new algorithm that incorporates effective robust features to defend against both safety threats. The convergence of the proposed algorithm is theoretically guaranteed for different types of learning models. We also empirically demonstrate that our algorithm enjoys satisfying performance, matching the theoretical results.

APPENDIX A
PROOF OF THEOREM 1

For $G = (1/|\mathcal{U}|)\sum_{i\in\mathcal{U}} g_i$ with $\mathcal{U} = \{(1), \ldots, ((1-\beta)m)\}$ and any specific vector S, we have

$$\begin{aligned}\|G - S\| &= \left\| \frac{1}{|\mathcal{U}|}\sum_{i\in\mathcal{U}} g_i - S \right\| \\ &= \left\| \frac{1}{|\mathcal{U}|}\sum_{i\in\mathcal{U}} (g_i - S) \right\| \\ &= \frac{1}{|\mathcal{U}|}\left\| \sum_{i\in\mathcal{U}\cap\mathcal{M}} (g_i - S) + \sum_{i\in\mathcal{U}\cap\mathcal{B}} (g_i - S) \right\| \\ &\le \frac{1}{|\mathcal{U}|}\left(\sum_{i\in\mathcal{U}\cap\mathcal{M}} \|g_i - S\| + \sum_{i\in\mathcal{U}\cap\mathcal{B}} \|g_i - S\| \right).\end{aligned}$$

For $i \in \mathcal{U}\cap\mathcal{M}$, $\|g_i - S\| \le \Delta$ (we define $\Delta = \max_{i\in\mathcal{M}} \|g_i - S\|$). For $i \in \mathcal{U}\cap\mathcal{B}$, we bound $\|g_i - S\|$ as

$$\begin{aligned}\|g_i - S\| &\le \|g_i\| + \|S\| \\ &\le \|g_{((1-\beta)m)}\| + \|S\| \\ &\le \|g_{((1-\alpha)m)}\| + \|S\| \\ &\le \max_{i\in\mathcal{M}} \|g_i\| + \|S\| \\ &= \max_{i\in\mathcal{M}} \|g_i - S + S\| + \|S\| \\ &\le \max_{i\in\mathcal{M}} \|g_i - S\| + 2\|S\| \\ &= \Delta + 2\|S\|.\end{aligned}$$

Combining the above results, we have

$$\begin{aligned}\|G - S\| &\le \frac{1}{|\mathcal{U}|}(|\mathcal{U}\cap\mathcal{M}|\cdot\Delta + |\mathcal{U}\cap\mathcal{B}|\cdot(\Delta + 2\|S\|)) \\ &= \frac{1}{|\mathcal{U}|}(|\mathcal{U}|\cdot\Delta + 2|\mathcal{U}\cap\mathcal{B}|\cdot\|S\|) \\ &= \Delta + \frac{2|\mathcal{U}\cap\mathcal{B}|}{|\mathcal{U}|}\|S\| \\ &\le \Delta + \frac{2\alpha}{1-\beta}\|S\|\end{aligned}$$

which is exactly the conclusion in Theorem 1. Note that the last inequality only holds on condition that $|\mathcal{B}| \le |\mathcal{U}|$, i.e., $\alpha \le 1 - \beta$, which, combined with $\beta \ge \alpha$, suggests that $\alpha \le (1/2)$.

APPENDIX B
PROOF OF LEMMA 1

Define $g(\theta; z) = f(\theta; z) - \lambda c(z, x)$ (we fix λ and x and view them as constants). Since $f(\theta; z)$ is L_{zz}-smooth with respect to z (Assumption 1) and $c(z, x)$ is 1-strongly convex (Assumption 2), we have

$$\nabla_z^2 g(\theta; z) = \nabla_z^2 f(\theta; z) - \lambda\cdot\nabla_z^2 c(z, x) \preceq -(\lambda - L_{zz})\cdot\mathbf{I}$$

which shows that $g(\theta; z)$ is $(\lambda - L_{zz})$-strongly concave with respect to z.

For any θ_1 and θ_2, define $z_1^* = \arg\sup_z g(\theta_1; z)$ and $z_2^* = \arg\sup_z g(\theta_2; z)$. Apparently, we have $\nabla_z g(\theta_1; z_1^*) = \nabla_z g(\theta_2; z_2^*) = 0$. According to the strong concavity of $g(\theta; z)$, we can obtain the following two inequalities:

$$\begin{aligned}g(\theta_1; z_1^*) \le\ & g(\theta_1; z_2^*) + \langle \nabla_z g(\theta_1; z_2^*), z_1^* - z_2^* \rangle \\ & - \frac{\lambda - L_{zz}}{2}\|z_1^* - z_2^*\|^2 \qquad (10) \\ g(\theta_1; z_2^*) \le\ & g(\theta_1; z_1^*) + \langle \nabla_z g(\theta_1; z_1^*), z_2^* - z_1^* \rangle \\ & - \frac{\lambda - L_{zz}}{2}\|z_2^* - z_1^*\|^2. \qquad (11)\end{aligned}$$

Adding (10) and (11) together, we have

$$\begin{aligned}&(\lambda - L_{zz})\|z_1^* - z_2^*\|^2 \\ &\le \langle \nabla_z g(\theta_1; z_2^*), z_1^* - z_2^* \rangle \\ &= \langle \nabla_z g(\theta_1; z_2^*) - \nabla_z g(\theta_2; z_2^*), z_1^* - z_2^* \rangle \\ &= \langle \nabla_z f(\theta_1; z_2^*) - \nabla_z f(\theta_2; z_2^*), z_1^* - z_2^* \rangle \\ &\le \|\nabla_z f(\theta_1; z_2^*) - \nabla_z f(\theta_2; z_2^*)\|\cdot\|z_1^* - z_2^*\| \\ &\le L_{z\theta}\|\theta_1 - \theta_2\|\cdot\|z_1^* - z_2^*\|\end{aligned}$$

which leads to

$$\|z_1^* - z_2^*\| \le \frac{L_{z\theta}}{\lambda - L_{zz}}\|\theta_1 - \theta_2\|.$$

Recall that $\nabla_\theta \phi_\lambda(\theta; x) = \nabla_\theta f(\theta; z^*(\theta))$ [see (4)], we have

$$\begin{aligned}&\|\nabla_\theta \phi_\lambda(\theta_1; x) - \nabla_\theta \phi_\lambda(\theta_2; x)\| \\ &= \|\nabla_\theta f(\theta_1; z_1^*) - \nabla_\theta f(\theta_2; z_2^*)\| \\ &= \|\nabla_\theta f(\theta_1; z_1^*) - \nabla_\theta f(\theta_1; z_2^*) + \nabla_\theta f(\theta_1; z_2^*) \\ &\qquad - \nabla_\theta f(\theta_2; z_2^*)\| \\ &\le \|\nabla_\theta f(\theta_1; z_1^*) - \nabla_\theta f(\theta_1; z_2^*)\| \\ &\quad + \|\nabla_\theta f(\theta_1; z_2^*) - \nabla_\theta f(\theta_2; z_2^*)\| \\ &\le L_{\theta z}\|z_1^* - z_2^*\| + L_{\theta\theta}\|\theta_1 - \theta_2\| \\ &\le \left(L_{\theta\theta} + \frac{L_{\theta z}L_{z\theta}}{\lambda - L_{zz}}\right)\|\theta_1 - \theta_2\|.\end{aligned}$$

According to the definition of smoothness, $\phi_\lambda(\theta; x)$ is L_F-smooth with respect to θ with $L_F = L_{\theta\theta} + (L_{\theta z}L_{z\theta}/(\lambda - L_{zz}))$. As a result, $F(\theta) = (1/N)\sum_{j=1}^N \phi_\lambda(\theta; x_j)$ is also L_F-smooth.

APPENDIX C
PROOF OF LEMMA 2

According to Theorem 1, we have the following result by setting $S = \nabla F(\theta_t)$:

$$\begin{aligned}&\|G(\theta_t) - \nabla F(\theta_t)\| \\ &\le \frac{2\alpha}{1-\beta}\|\nabla F(\theta_t)\| + \max_{i\in\mathcal{M}} \|g_i(\theta_t) - \nabla F(\theta_t)\|. \qquad (12)\end{aligned}$$

According to Algorithm 2, for $i \in \mathcal{M}$, $g_i(\theta_t) = (1/n)\sum_{j=(i-1)n+1}^{(i-1)n+n} \nabla_\theta f(\theta_t; z_j^\varepsilon(\theta_t))$. Defining an auxiliary term $g_i^*(\theta_t) = (1/n)\sum_{j=(i-1)n+1}^{(i-1)n+n} \nabla_\theta f(\theta_t; z_j^*(\theta_t))$ (where $z_j^*(\theta_t)$ is the exact maximizer), we can bound the distance between $g_i(\theta_t)$ and $g_i^*(\theta_t)$ for $\forall i \in \mathcal{M}$ as

$$\begin{aligned}&\|g_i(\theta_t) - g_i^*(\theta_t)\| \\ &\le \max_{1\le j\le N} \|\nabla_\theta f(\theta_t; z_j^\varepsilon(\theta_t)) - \nabla_\theta f(\theta_t; z_j^*(\theta_t))\|\end{aligned}$$

$$\leq L_{\theta z} \max_{1\leq j\leq N} \left\| z_j^{\varepsilon}(\theta_t) - z_j^*(\theta_t) \right\|$$
$$\leq L_{\theta z}\varepsilon \tag{13}$$

in which the second inequality follows from Assumption 1 and the third inequality follows from the definition of $z_j^{\varepsilon}(\theta_t)$.

Next, we have

$$\begin{aligned}
&\max_{i\in\mathcal{M}} \|g_i(\theta_t) - \nabla F(\theta_t)\| \\
&= \max_{i\in\mathcal{M}} \left\| g_i(\theta_t) - g_i^*(\theta_t) + g_i^*(\theta_t) - \nabla F(\theta_t) \right\| \\
&\leq \max_{i\in\mathcal{M}} \left(\left\| g_i(\theta_t) - g_i^*(\theta_t) \right\| + \left\| g_i^*(\theta_t) - \nabla F(\theta_t) \right\| \right) \\
&\leq L_{\theta z}\varepsilon + \max_{i\in\mathcal{M}} \left\| g_i^*(\theta_t) - \nabla F(\theta_t) \right\| \\
&\leq L_{\theta z}\varepsilon \\
&\quad + \max_{1\leq k\leq N} \left\| \nabla_\theta f\left(\theta_t; z_k^*(\theta_t)\right) - \frac{1}{N}\sum_{j=1}^{N} \nabla_\theta f\left(\theta_t; z_j^*(\theta_t)\right) \right\| \\
&\leq L_{\theta z}\varepsilon + \sigma
\end{aligned} \tag{14}$$

where the second inequality follows from (13) and the last inequality follows from Assumption 4.

Finally, combining (12) and (14) leads to the conclusion in Lemma 2.

Appendix D
Proof of Theorem 2

According to Lemma 1, $F(\theta)$ is L_F-smooth. According to the property of smoothness, we have

$$\begin{aligned}
&F(\theta_{t+1}) \\
&\leq F(\theta_t) + \langle \nabla F(\theta_t), \theta_{t+1} - \theta_t \rangle + \frac{L_F}{2}\|\theta_{t+1} - \theta_t\|^2 \\
&= F(\theta_t) - \eta \langle \nabla F(\theta_t), G(\theta_t) \rangle + \frac{L_F}{2}\eta^2 \|G(\theta_t)\|^2 \\
&= F(\theta_t) - \frac{1}{L_F}\langle \nabla F(\theta_t), G(\theta_t) - \nabla F(\theta_t) + \nabla F(\theta_t) \rangle \\
&\quad + \frac{1}{2L_F}\|G(\theta_t) - \nabla F(\theta_t) + \nabla F(\theta_t)\|^2 \\
&= F(\theta_t) - \frac{1}{2L_F}\|\nabla F(\theta_t)\|^2 + \frac{1}{2L_F}\|G(\theta_t) - \nabla F(\theta_t)\|^2
\end{aligned} \tag{15}$$

where the first equality follows from $\theta_{t+1} = \theta_t - \eta \cdot G(\theta_t)$ and the second equality follows from $\eta = (1/L_F)$. Note that the derivation of (15) is a common trick in analyzing the convergence of smooth functions, which shifts the burden of proving convergence into the relatively easy task of quantifying $\|G(\theta_t) - \nabla F(\theta_t)\|$.

According to Lemma 2, we have $\|G(\theta_t) - \nabla F(\theta_t)\| \leq C_\alpha \|\nabla F(\theta_t)\| + \Delta$ with $C_\alpha = (2\alpha/(1-\beta))$ and $\Delta = L_{\theta z}\varepsilon + \sigma$, which leads to

$$\begin{aligned}
&\|G(\theta_t) - \nabla F(\theta_t)\|^2 \\
&\leq C_\alpha^2 \|\nabla F(\theta_t)\|^2 + 2C_\alpha \|\nabla F(\theta_t)\|\Delta + \Delta^2 \\
&\leq (1+r)C_\alpha^2 \|\nabla F(\theta_t)\|^2 + (1+1/r)\Delta^2
\end{aligned} \tag{16}$$

for any $r > 0$.

Combining (15) and (18), we have

$$F(\theta_{t+1}) \leq F(\theta_t) - \frac{1-(1+r)C_\alpha^2}{2L_F}\|\nabla F(\theta_t)\|^2 + \frac{1+1/r}{2L_F}\Delta^2 \tag{17}$$

which is equivalent to

$$\begin{aligned}
\|\nabla F(\theta_t)\|^2 &\leq \frac{2L_F}{1-(1+r)C_\alpha^2}[F(\theta_t) - F(\theta_{t+1})] \\
&\quad + \frac{1+1/r}{1-(1+r)C_\alpha^2}\Delta^2.
\end{aligned} \tag{18}$$

Summing up (18) for $t = 0, 1, \ldots, T-1$ before being divided by T gives

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla F(\theta_t)\|^2 \\
&\leq \frac{2L_F}{(1-(1+r)C_\alpha^2)T}[F(\theta_0) - F(\theta_T)] + \frac{1+1/r}{1-(1+r)C_\alpha^2}\Delta^2 \\
&\leq \frac{2L_F}{(1-(1+r)C_\alpha^2)T}[F(\theta_0) - F(\theta^*)] + \frac{1+1/r}{1-(1+r)C_\alpha^2}\Delta^2
\end{aligned} \tag{19}$$

which is exactly the conclusion in Theorem 2.

Note that the transition from (17) to (18) only stands under the condition that $1-(1+r)C_\alpha^2 > 0$, which constrains r to the less than $(1/C_\alpha^2)-1$. On the other hand, $r > 0$, which requires $C_\alpha = (2\alpha/(1-\beta)) < 1$, i.e., $2\alpha + \beta < 1$. Since $\beta \geq \alpha$, we can conclude that (19) holds if and only if $\alpha < (1/3)$ and $0 < r < ((1-\beta)/2\alpha)^2 - 1$.

Appendix E
Proof of Theorem 3

First, we seek to establish the convexity of $F(\theta)$. Recall that $\phi_\lambda(\theta; x) = \sup_z\{f(\theta; z) - \lambda c(z, x)\}$. For any θ_1, θ_2 and $0 \leq t \leq 1$, we have

$$\begin{aligned}
&\phi_\lambda(t\theta_1 + (1-t)\theta_2; x) \\
&= \sup_z\{f(t\theta_1 + (1-t)\theta_2; z) - \lambda c(z, x)\} \\
&\leq \sup_z\{tf(\theta_1; z) + (1-t)f(\theta_2; z) - \lambda c(z, x)\} \\
&= \sup_z\{t[f(\theta_1; z) - \lambda c(z, x)] + (1-t)[f(\theta_2; z) - \lambda c(z, x)]\} \\
&\leq t\sup_z\{f(\theta_1; z) - \lambda c(z, x)\} \\
&\quad + (1-t)\sup_z\{f(\theta_2; z) - \lambda c(z, x)\} \\
&= t\phi_\lambda(\theta_1; x) + (1-t)\phi_\lambda(\theta_2; x)
\end{aligned} \tag{20}$$

in which the first inequality follows from Assumption 5. According to (20), $\phi_\lambda(\theta; x)$ is convex with respect to θ. As a result, $F(\theta) = (1/N)\sum_{j=1}^{N} \phi_\lambda(\theta; x_j)$ is also convex.

The convexity of $F(\theta)$ suggests that $F(\theta^*) \geq F(\theta_t) + \langle \nabla F(\theta_t), \theta^* - \theta_t \rangle$, which leads to

$$\begin{aligned}
F(\theta_t) - F(\theta^*) &\leq \langle \nabla F(\theta_t), \theta_t - \theta^* \rangle \\
&\leq \|\nabla F(\theta_t)\| \cdot \|\theta_t - \theta^*\| \\
&\leq D\|\nabla F(\theta_t)\|
\end{aligned} \tag{21}$$

in which $D = k\|\theta_0 - \theta^*\|$. The third inequality of (21) follows from Assumption 6. As a result, we obtain (22) as a key property in the subsequent analysis

$$\|\nabla F(\theta_t)\| \geq \frac{1}{D}[F(\theta_t) - F(\theta^*)]. \tag{22}$$

Since Theorem 3 keeps all the assumptions made in Theorem 2, all the intermediate steps in the proof of Theorem 2 also apply here. In this regard, we borrow (17), i.e.,

$$F(\theta_{t+1}) - F(\theta_t) \leq -A\|\nabla F(\theta_t)\|^2 + B \tag{23}$$

in which we define $A = ((1-(1+r)C_\alpha^2)/2L_F)$ and $B = (((1+1/r)(L_{\theta z}\varepsilon+\sigma)^2)/2L_F)$ for convenience.

Next, we consider two cases in regard to the relationship between $A\|\nabla F(\theta_t)\|^2$ and B.

Case 1: Suppose that for all $0 \le t \le T-1$, it holds that $B \le (A/2)\|\nabla F(\theta_t)\|^2$. In this case, we have

$$F(\theta_{t+1}) - F(\theta_t) \le -\frac{A}{2}\|\nabla F(\theta_t)\|^2. \tag{24}$$

Combining (22) and (24) gives

$$[F(\theta_t) - F(\theta^*)]^2 \le \frac{2D^2}{A}([F(\theta_t) - F(\theta^*)] - [F(\theta_{t+1}) - F(\theta^*)]) \tag{25}$$

which, after divided by $[F(\theta_t) - F(\theta^*)][F(\theta_{t+1}) - F(\theta^*)]$ on both sides, leads to

$$\frac{F(\theta_t) - F(\theta^*)}{F(\theta_{t+1}) - F(\theta^*)} \le \frac{2D^2}{A}\left(\frac{1}{F(\theta_{t+1}) - F(\theta^*)} - \frac{1}{F(\theta_t) - F(\theta^*)}\right). \tag{26}$$

According to (24), we have $F(\theta_{t+1}) \le F(\theta_t)$. Therefore, $(F(\theta_t) - F(\theta^*))/(F(\theta_{t+1}) - F(\theta^*)) \ge 1$, and (26) can be simplified as

$$\frac{1}{F(\theta_{t+1}) - F(\theta^*)} - \frac{1}{F(\theta_t) - F(\theta^*)} \ge \frac{A}{2D^2}. \tag{27}$$

Summing up (27) for $t = 0, 1, \ldots, T-1$ gives

$$\frac{1}{F(\theta_T) - F(\theta^*)} \ge \frac{AT}{2D^2} + \frac{1}{F(\theta_0) - F(\theta^*)} \ge \frac{AT}{2D^2} \tag{28}$$

which leads to

$$F(\theta_T) - F(\theta^*) \le \frac{2D^2}{AT}. \tag{29}$$

Case 2: Suppose that there exists $t_0 \in \{0, 1, \ldots, T-1\}$ such that $B > (A/2)\|\nabla F(\theta_{t_0})\|^2$. In this case, we have

$$\|\nabla F(\theta_{t_0})\| < \sqrt{\frac{2B}{A}}. \tag{30}$$

Combining (22) and (30) gives

$$F(\theta_{t_0}) - F(\theta^*) < D\sqrt{\frac{2B}{A}}. \tag{31}$$

Next, we show by contradiction that for all $t \ge t_0$, it holds that

$$F(\theta_t) - F(\theta^*) \le D\sqrt{\frac{2B}{A}} + B. \tag{32}$$

Suppose that there exists $t_1 \ge t_0$ such that

$$F(\theta_{t_1}) - F(\theta^*) > D\sqrt{\frac{2B}{A}} + B. \tag{33}$$

According to (23), we have

$$F(\theta_{t_1}) - F(\theta_{t_1-1}) \le -A\|\nabla F(\theta_{t_1-1})\|^2 + B \le B. \tag{34}$$

Combining (33) and (34) gives

$$F(\theta_{t_1-1}) - F(\theta^*) > D\sqrt{\frac{2B}{A}}. \tag{35}$$

Combining (35) and (22) gives

$$\|\nabla F(\theta_{t_1-1})\| > \sqrt{\frac{2B}{A}}. \tag{36}$$

Plugging (36) into (23), we obtain $F(\theta_{t_1-1}) \ge F(\theta_{t_1}) + B$, which suggests that (33) also holds with t_1 replaced by $t_1 - 1$. By the same token, we can conclude that (33) should hold with t_1 replaced by all $t \le t_1$. This is in clear contradiction with the incident of $t = t_0$ as shown in (31). Therefore, (32) is valid for all $t \ge t_0$ as stated.

Finally, combining the results of Case 1 (29) and Case 2 (31), we achieve that

$$F(\theta_T) - F(\theta^*) \le \max\left\{\frac{2D^2}{AT}, D\sqrt{\frac{2B}{A}} + B\right\} \tag{37}$$

which completes the proof of Theorem 3.

Appendix F
Proof of Theorem 4

According to Lemma 1, $F(\theta)$ is L_F-smooth, and according to Assumption 7, $F(\theta)$ is λ_F-strongly convex. In the convex optimization theory, it is well known that smooth and strongly convex functions enjoy a linear convergence rate with gradient descent. Here, we will first establish and then use such a property with a specific convergence factor. We start with the following equality:

$$\|\theta_t - \eta\nabla F(\theta_t) - \theta^*\|^2 = \|\theta_t - \theta^*\|^2 - 2\eta\langle\nabla F(\theta_t), \theta_t - \theta^*\rangle + \eta^2\|\nabla F(\theta_t)\|^2. \tag{38}$$

According to the co-coercivity of smooth and strongly convex function, we have

$$\langle\nabla F(\theta_t) - \nabla F(\theta^*), \theta_t - \theta^*\rangle \ge \frac{1}{L_F + \lambda_F}\|\nabla F(\theta_t) - \nabla F(\theta^*)\|^2 + \frac{L_F\lambda_F}{L_F + \lambda_F}\|\theta_t - \theta^*\|^2. \tag{39}$$

Since θ^* is the global minimizer of $F(\theta)$, therefore $\nabla F(\theta^*) = 0$, (39) reduces to

$$\langle\nabla F(\theta_t), \theta_t - \theta^*\rangle \ge \frac{1}{L_F + \lambda_F}\|\nabla F(\theta_t)\|^2 + \frac{L_F\lambda_F}{L_F + \lambda_F}\|\theta_t - \theta^*\|^2. \tag{40}$$

Plugging (40) into (38), we have

$$\|\theta_t - \eta\nabla F(\theta_t) - \theta^*\|^2 \le \left(1 - 2\eta\frac{L_F\lambda_F}{L_F + \lambda_F}\right)\|\theta_t - \theta^*\|^2 + \left(\eta^2 - \frac{2\eta}{L_F + \lambda_F}\right)\|\nabla F(\theta_t)\|^2. \tag{41}$$

In order to eliminate the last term in (41), we take $\eta = (2/(L_F + \lambda_F))$ and simplify (41) as

$$\|\theta_t - \eta\nabla F(\theta_t) - \theta^*\|^2 \le \left(1 - \frac{4L_F\lambda_F}{(L_F + \lambda_F)^2}\right)\|\theta_t - \theta^*\|^2 \tag{42}$$

which is the same as

$$\|\theta_t - \eta\nabla F(\theta_t) - \theta^*\| \le \frac{L_F - \lambda_F}{L_F + \lambda_F}\|\theta_t - \theta^*\| \tag{43}$$

which verifies linear convergence with a factor of $(L_F - \lambda_F)/(L_F + \lambda_F)$. Note that (43) holds on condition that $\eta = (2/(L_F + \lambda_F))$.

Next, we try to evaluate the single-step progress made by our algorithm as follows:

$$\begin{aligned}
&\|\theta_{t+1} - \theta^*\| \\
&= \|\theta_t - \eta G(\theta_t) - \theta^*\| \\
&= \|\theta_t - \eta \nabla F(\theta_t) - \theta^* + \eta[\nabla F(\theta_t) - G(\theta_t)]\| \\
&\leq \|\theta_t - \eta \nabla F(\theta_t) - \theta^*\| + \eta \|\nabla F(\theta_t) - G(\theta_t)\| \\
&\leq \frac{L_F - \lambda_F}{L_F + \lambda_F} \|\theta_t - \theta^*\| + \frac{2}{L_F + \lambda_F} \|G(\theta_t) - \nabla F(\theta_t)\| \\
&\leq \frac{L_F - \lambda_F}{L_F + \lambda_F} \|\theta_t - \theta^*\| + \frac{2C_\alpha}{L_F + \lambda_F} \|\nabla F(\theta_t)\| + \frac{2\Delta}{L_F + \lambda_F}
\end{aligned} \tag{44}$$

in which the second inequality follows from (43) by taking $\eta = (2/(L_F + \lambda_F))$ and the third inequality follows from Lemma 2 with $C_\alpha = (2\alpha/(1-\beta))$ and $\Delta = L_{\theta z}\varepsilon + \sigma$.

According to the properties of $F(\theta)$ being L_F-smooth, we have

$$\frac{1}{2L_F} \|\nabla F(\theta_t)\|^2 \leq F(\theta_t) - F(\theta^*) \leq \frac{L_F}{2} \|\theta_t - \theta^*\|^2$$

which leads to

$$\|\nabla F(\theta_t)\| \leq L_F \|\theta_t - \theta^*\|. \tag{45}$$

Plugging (45) into (44), we have

$$\|\theta_{t+1} - \theta^*\| \leq \frac{2L_F C_\alpha + L_F - \lambda_F}{L_F + \lambda_F} \|\theta_t - \theta^*\| + \frac{2\Delta}{L_F + \lambda_F}. \tag{46}$$

By iterating (46), we obtain

$$\begin{aligned}
\|\theta_T - \theta^*\| \leq &\left(\frac{2L_F C_\alpha + L_F - \lambda_F}{L_F + \lambda_F} \right)^T \|\theta_0 - \theta^*\| \\
&+ \frac{\Delta}{\lambda_F - L_F C_\alpha}
\end{aligned} \tag{47}$$

which is exactly the conclusion in Theorem 4.

Note that the transition from (46) to (47) only stands under the condition that $(2L_F C_\alpha + L_F - \lambda_F)/(L_F + \lambda_F) < 1$, which requires that $C_\alpha = (2\alpha/(1-\beta)) < (\lambda_F/L_F)$, i.e., $2\alpha(L_F/\lambda_F) + \beta < 1$. Since $\beta \geq \alpha$, we can conclude that (47) holds if and only if $\alpha < (1/(1 + 2L_F/\lambda_F))$.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[2] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, 2021.

[3] J. Wang, A. Pal, Q. Yang, K. Kant, K. Zhu, and S. Guo, "Collaborative machine learning: Schemes, robustness, and privacy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 9625–9642, Dec. 2022.

[4] L. Lyu et al., "Privacy and robustness in federated learning: Attacks and defenses," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 8726–8746, Jul. 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9945997

[5] J. Wang, J. Liu, and N. Kato, "Networking and communications in autonomous driving: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1243–1274, 2nd Quart., 2018.

[6] I. Kononenko, "Machine learning for medical diagnosis: History, state of the art and perspective," *Artif. Intell. Med.*, vol. 23, no. 1, pp. 89–109, Aug. 2001.

[7] A. Liu and B. Ziebart, "Robust classification under sample selection bias," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 37–45.

[8] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. Cambridge, MA, USA: MIT Press, 2009.

[9] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski, *Robust Optimization* (Princeton Series in Applied Mathematics). Princeton, NJ, USA: Princeton Univ. Press, 2009.

[10] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM Rev.*, vol. 53, no. 3, pp. 464–501, 2011.

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[12] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[13] H. Xu, Y. Liu, and H. Sun, "Distributionally robust optimization with matrix moment constraints: Lagrange duality and cutting plane methods," *Math. Program.*, vol. 169, no. 2, pp. 489–529, Jun. 2018.

[14] S. Mehrotra and H. Zhang, "Models and algorithms for distributionally robust least squares problems," *Math. Program.*, vol. 146, nos. 1–2, pp. 123–141, Aug. 2014.

[15] E. Delage and Y. Ye, "Distributionally robust optimization under moment uncertainty with application to data-driven problems," *Operations Res.*, vol. 58, no. 3, pp. 595–612, Jun. 2010.

[16] J. Goh and M. Sim, "Distributionally robust optimization and its tractable approximations," *Oper. Res.*, vol. 58, no. 4, pp. 902–917, Aug. 2010.

[17] H. Namkoong and J. C. Duchi, "Stochastic gradient methods for distributionally robust optimization with f-divergences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2208–2216.

[18] J. C. Duchi and H. Namkoong, "Learning models with uniform performance via distributionally robust optimization," *Ann. Statist.*, vol. 49, no. 3, pp. 1378–1406, Jun. 2021.

[19] S. Shafieezadeh-Abadeh, P. M. Esfahani, and D. Kuhn, "Distributionally robust logistic regression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1576–1584.

[20] P. M. Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations," *Math. Program.*, vol. 171, nos. 1–2, pp. 115–166, Sep. 2018.

[21] R. Chen, "Distributionally robust learning under the Wasserstein metric," Doctoral dissertation, College Eng., Boston Univ., Boston, MA, USA, 2019.

[22] A. Sinha, H. Namkoong, and J. Duchi, "Certifying some distributional robustness with principled adversarial training," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[23] A. Sadeghi, G. Wang, M. Ma, and G. B. Giannakis, "Learning while respecting privacy and robustness to distributional uncertainties and adversarial data," 2020, *arXiv:2007.03724*.

[24] W. Shen, H. Li, and Z. Zheng, "Learning to attack distributionally robust federated learning," in *Proc. NeurIPS Workshop Scalability, Privacy, Secur. Federated Learn. (SpicyFL)*, 2020.

[25] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4615–4625.

[26] Y. Deng, M. M. Kamani, and M. Mahdavi, "Distributionally robust federated averaging," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 15111–15122.

[27] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982.

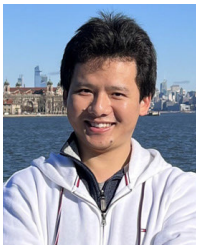
[28] Z. Yang, A. Gang, and W. U. Bajwa, "Adversary-resilient distributed and decentralized statistical inference and machine learning: An overview of recent advances under the Byzantine threat model," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 146–159, May 2020.

[29] S. Li, C. H. Ngai, and T. Voigt, "An experimental study of Byzantine-robust aggregation schemes in federated learning," *TechRxiv Preprint*, pp. 1–13, Jan. 2023, doi: 10.36227/techrxiv.19560325.v1. [Online]. Available: https://ieeexplore.ieee.org/document/10018261

[30] S. P. Karimireddy, L. He, and M. Jaggi, "Byzantine-robust learning on heterogeneous datasets via bucketing," in *Proc. Int. Conf. Learn. Represent.*, 2022.

[31] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3368–3376.

[32] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos, "DRACO: Byzantine resilient distributed training via redundant gradients," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 903–912.
[33] S. Rajput, H. Wang, Z. Charles, and D. Papailiopoulos, "DETOX: A redundancy-based framework for faster and more robust gradient aggregation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 10320–10330.
[34] D. Data, L. Song, and S. Diggavi, "Data encoding for Byzantine-resilient distributed gradient descent," in *Proc. 56th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2018, pp. 863–870.
[35] C. Xie, O. Koyejo, and I. Gupta, "Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6893–6901.
[36] X. Cao and L. Lai, "Distributed gradient descent algorithm robust to an arbitrary number of Byzantine attackers," *IEEE Trans. Signal Process.*, vol. 67, no. 22, pp. 5850–5864, Nov. 2019.
[37] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "FLTrust: Byzantine-robust federated learning via trust bootstrapping," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2021.
[38] J. Regatti, H. Chen, and A. Gupta, "ByGARS: Byzantine SGD with arbitrary number of attackers," 2020, *arXiv:2006.13421*.
[39] P. Blanchard, E. M. Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 119–129.
[40] E. M. Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in Byzantium," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3521–3530.
[41] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 2, pp. 1–25, 2017.
[42] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5650–5659.
[43] L. Su and J. Xu, "Securing distributed gradient descent in high dimensional statistical learning," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 47, no. 1, pp. 83–84, Dec. 2019.
[44] J. Bernstein, J. Zhao, K. Azizzadenesheli, and A. Anandkumar, "signSGD with majority vote is communication efficient and fault tolerant," in *Proc. Int. Conf. Learn. Represent.*, 2019.
[45] A. Ghosh, R. K. Maity, S. Kadhe, A. Mazumdar, and K. Ramchandran, "Communication-efficient and Byzantine-robust distributed learning," in *Proc. Inf. Theory Appl. Workshop (ITA)*, Feb. 2020, pp. 1–28.
[46] A. Ghosh, R. K. Maity, S. Kadhe, A. Mazumdar, and K. Ramchandran, "Communication-efficient and Byzantine-robust distributed learning with error feedback," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 3, pp. 942–953, Sep. 2021.
[47] A. Ghosh, R. K. Maity, S. Kadhe, A. Mazumdar, and K. Ramachandran, "Communication efficient and Byzantine tolerant distributed learning," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 2545–2550.
[48] A. Ghosh, R. K. Maity, and A. Mazumdar, "Distributed Newton can communicate less and resist Byzantine workers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 18028–18038.
[49] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 6357–6368.
[50] Z. Hu, K. Shaloudegi, G. Zhang, and Y. Yu, "Federated learning meets multi-objective optimization," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 4, pp. 2039–2051, Jul. 2022.
[51] X. Xu and L. Lyu, "A reputation mechanism is all you need: Collaborative fairness and adversarial robustness in federated learning," 2020, *arXiv:2011.10464*.
[52] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1544–1551.
[53] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation," in *Proc. Uncertainty Artif. Intell.*, 2020, pp. 261–270.
[54] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8632–8642.
[55] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," in *Proc. USENIX Secur. Symp.*, 2020, pp. 1605–1622.
[56] M. Hopkins. (1999). *UCI Machine Learning Repository*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/spambase



Guanqiang Zhou received the B.S. degree in electronics and information engineering and the M.S. degree in communication and information systems from Northwestern Polytechnical University, Xi'an, China, in 2015 and 2018, respectively, and the Ph.D. degree in electrical and computer engineering from George Mason University, Fairfax, VA, USA, in 2024.

He is currently an Assistant Professor of instruction with the Department of Electrical and Computer Engineering, University of Iowa, Iowa City, IA, USA. His research interests include statistical signal processing, convex optimization theory, and distributed machine learning.

Ping Xu (Member, IEEE) received the B.E. degree in electronics and information engineering from Northwestern Polytechnical University, Xi'an, China, in 2015, and the M.S. and Ph.D. degrees in electrical engineering from George Mason University, Fairfax, VA, USA, in 2018 and 2022, respectively.

She was a Post-Doctoral Researcher with the Department of Electrical and Computer Engineering, George Mason University. She is currently an Assistant Professor with the Electrical and Computer Engineering Department, The University of Texas Rio Grande Valley, Edinburg, TX, USA. Her research interests span the areas of machine learning and optimization, signal processing, dynamical systems, and cooperative control.

Dr. Xu received the Rising Star in EECS Award in 2022, the Outstanding Academic Achievement Award at GMU in 2022, and the IEEE Signal Processing Society Professional Development Grant in 2021.

Yue Wang (Senior Member, IEEE) received the Ph.D. degree in communication and information system from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, in 2011.

He was a Research Assistant Professor with the Electrical and Computer Engineering Department, George Mason University, Fairfax, VA, USA. He is currently an Assistant Professor with the Department of Computer Science, Georgia State University, Atlanta, GA, USA, since August 2023. His general research interests include the interdisciplinary areas of machine learning, signal processing, wireless communications, and their applications in cyber-physical systems. His specific research focuses on distributed optimization and machine learning, sparse signal processing, massive multi-input–multi-output (MIMO), millimeter-wave (mmWave) communications, cognitive radios, spectrum sensing, the Internet of Things, direction-of-arrival estimation, and high-dimensional data analysis.

Zhi Tian (Fellow, IEEE) was on the faculty of Michigan Technological University, Houghton, MI, USA, from 2000 to 2014. She was the Program Director of the U.S. National Science Foundation, Alexandria, VA, USA, from 2012 to 2014. She is currently a Professor with the Electrical and Computer Engineering Department, George Mason University, Fairfax, VA, USA, since 2015. Her general research interests are in the areas of signal processing, communications, detection, and estimation. Her current research focuses on decentralized optimization and learning over networks, statistical inference from distributed data, compressed sensing for random processes, cognitive radios, and millimeter-wave multi-input–multioutput (MIMO) communications.

Dr. Tian received the IEEE Communications Society TCCN Publication Award in 2018. She was the Chair of the IEEE Signal Processing Society Big Data Special Interest Group and a member of the IEEE Signal Processing for Communications and Networking Technical Committee. She served on the Board of Governors of the IEEE Signal Processing Society from 2019 to 2021. She served as an Associate Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE TRANSACTIONS ON SIGNAL PROCESSING. She is the Editor-in-Chief of IEEE TRANSACTIONS ON SIGNAL PROCESSING. She was a Distinguished Lecturer of the IEEE Communications Society and the IEEE Vehicular Technology Society.