Near-Optimal Resilient Aggregation Rules for Distributed Learning Using 1-Center and 1-Mean Clustering with Outliers

Yuhao Yi[1*], **Ronghui You**[2*], **Hong Liu**[1], **Changxin Liu**[3], **Yuan Wang**[4†], **Jiancheng Lv**[1†]

[1]College of Computer Science, Sichuan University
[2]School of Statistics and Data Science, Nankai University
[3]School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology
[4]School of Robotics, Hunan University
yuhaoyi@scu.edu.cn, yourh@nankai.edu.cn, changxin@kth.se, yuanw@hnu.edu.cn, lvjiancheng@scu.edu.cn

Abstract

Byzantine machine learning has garnered considerable attention in light of the unpredictable faults that can occur in large-scale distributed learning systems. The key to secure resilience against Byzantine machines in distributed learning is resilient aggregation mechanisms. Although abundant resilient aggregation rules have been proposed, they are designed in ad-hoc manners, imposing extra barriers on comparing, analyzing, and improving the rules across performance criteria. This paper studies near-optimal aggregation rules using clustering in the presence of outliers. Our outlier-robust clustering approach utilizes geometric properties of the update vectors provided by workers. Our analysis show that constant approximations to the 1-center and 1-mean clustering problems with outliers provide near-optimal resilient aggregators for metric-based criteria, which have been proven to be crucial in the homogeneous and heterogeneous cases respectively. In addition, we discuss two contradicting types of attacks under which no single aggregation rule is guaranteed to improve upon the naive average. Based on the discussion, we propose a two-phase resilient aggregation framework. We run experiments for image classification using a non-convex loss function. The proposed algorithms outperform previously known aggregation rules by a large margin with both homogeneous and heterogeneous data distributions among non-faulty workers. Code and appendix are available at https://github.com/jerry907/AAAI24-RASHB.

1 Introduction

Distributed machine learning (ML) that involves several collaborative computing machines has been recognized as the backbone for training large-scale ML models in the modern society (Warnat-Herresthal et al. 2021; Kairouz et al. 2021). However, although distributed ML has significantly improved the efficiency of training process, it tends to be more vulnerable to misbehaving (a.k.a., *Byzantine*) workers. It has been reported in (Baruch, Baruch, and Goldberg 2019; Karimireddy, He, and Jaggi 2022) that a few Byzantine machines can largely deteriorate the training performance by transmitting falsified information. To this end, Byzantine re-

silience in distributed ML has recently received increasing attention from both academia and industry.

In resilient distributed ML, a robust distributed optimization algorithm is designed such that the training model remains to be accurate in the presence of a subset of Byzantine workers (Farhadkhani et al. 2022). The key to achieving this objective is a robust aggregation protocol within the server to distill the information sent by the workers. Prior works in this regard are roughly categorized into two classes depending on the property of the training dataset. In the first class, known as the *homogeneous* setup, the data sampled by the workers are assumed to be identically distributed. Another class is the *heterogeneous* setup (Li et al. 2019; Data and Diggavi 2021), where the data samples among the workers may not precisely reflect the overall population. The difference in datasets induces distinct treatment of Byzantine workers and the best achievable performance (Karimireddy, He, and Jaggi 2022), and the problem of resilient distributed ML under heterogeneity is arguably more challenging (Allouah et al. 2023a).

To facilitate the analysis of the aggregation rules, a line of recent work has established the connections between the properties of aggregators and the performance of the optimization algorithms. The paper (Farhadkhani et al. 2022) studied resilient distributed ML with homogeneous data distributions and proposed the concept of (f, λ)-resilient averaging aggregators. Subsequent work study distributed ML with heterogeneous data distributions, proposing a series of concepts such as the $(\delta_{\max}, \zeta)$-agnostic robust (Karimireddy, He, and Jaggi 2022), the (f, κ)-robust (Allouah et al. 2023a), and the (f, ξ)-robust averaging (Allouah et al. 2023b) aggregators. The relationships of these concepts are also discussed in (Allouah et al. 2023a) and (Allouah et al. 2023b).

The criteria for the aggregation rules are defined using metrics on subsets of update vectors. However, most of the aggregators are not directly designed to minimize the criteria, resulting in the suboptimality of many aggregation rules. Exceptions include the MDA algorithm in (Farhadkhani et al. 2022) and the SMEA algorithm in (Allouah et al. 2023b), both of which suffer from high computational costs.

The 1-Center problem, or the minimum enclosing ball problem, is a fundamental problem in computational geom-

---

etry (Yildirim 2008), even in low dimensions (Smallwood 1965; Har-Peled 2011). Its variant with outliers also receives significant interests (Narayanan 2018; Ding 2020). 1-Mean clustering with outliers is a similar problem using the sum of squared distances as the cost function. Approximation algorithms are intensively studied for clustering problems with outliers (Friggstad et al. 2019; Agrawal et al. 2023; Banerjee, Ostrovsky, and Rabani 2021). In this paper we propose to use 1-center and 1-mean clustering with outliers as resilient aggregation rules.

Related work. In recent years, Byzantine resilient distributed ML has been intensively studied. Many algorithmic frameworks are proposed (Yin et al. 2018; Liu, Gupta, and Vaidya 2021; Allen-Zhu et al. 2021) to address complex attacks developed progressively (Xie, Koyejo, and Gupta 2020; Baruch, Baruch, and Goldberg 2019). These early discussions diverge in assumptions and overall frameworks of the training algorithms. A line of recent work based on resilient aggregating of momentum, or resilient stochastic heavy ball (Karimireddy, He, and Jaggi 2021, 2022; Farhadkhani et al. 2022; Allouah et al. 2023a,b), is the most relevant to its paper.

Under the framework, properties of some previously designed aggregators are investigated. Some provably optimal aggregators often have high computational costs while many commonly used simple aggregators turn out to be suboptimal. Pre-aggregation steps such as bucketing (Karimireddy, He, and Jaggi 2022) and nearest neighbor mixing (NNM) (Allouah et al. 2023a) are proposed to improve the performance of aggregators.

Clustering algorithms (Ghosh et al. 2019; Sattler et al. 2020) are also proposed and analyzed under more restrictive assumptions, where machines in the same cluster have the same data distribution. El Mhamdi, Guerraoui, and Rouault studied a medoid-based algorithm as an approximation to the geometric median (Chen, Su, and Xu 2017), but its optimality remains to be studied. In this paper, we reduce the problems of optimal aggregation rule design to 1-center/mean clustering problems with outliers, and apply computationally efficient approximations to construct near-optimal aggregators.

Main contributions. In this paper, we propose near-optimal aggregators for Byzantine resilient distributed ML using approximation algorithms for the problems of 1-center and 1-mean clustering with outliers. Although the problems of 1-center and 1-mean clustering with outliers are NP-hard, their approximations can be computed efficiently. We show that 2-approximations achieve near-optimal safety guarantees under existing analytical framework. Specifically, the 1-center with outliers algorithm is optimal for (f, λ)-resilience and achieves the currently best bound for $(\delta_{\max}, \zeta)$-agnostic robustness; the 1-mean with outliers approach is optimal for (f, κ)-robustness.

In addition, we discuss two types of contradicting attacks, namely the sneak attack and the siege attack, to show that no single aggregation algorithm, being agnostic about the true distribution of update vectors of normal clients, outperforms other algorithms in all circumstances. To address the

dilemma created by the indistinguishability of the two types of attacks, we propose a two-phase aggregation framework. In the framework, 1) the server proposes two candidate sets of parameters using received update vectors; 2) clients elect a set of parameters to commit by evaluating the losses using resampled data. Using clustering-based approaches, the two proposed sets of parameters are easily generated by constructing filters to address the sneak attacks and the siege attacks respectively.

In summary, this paper 1) proposes near-optimal aggregation rules with provable guarantees by approximating the problems of 1-center/mean clustering with outliers; 2) proposes a two-phase aggregating of optimization framework in which the clustering approaches are used to defend two types of contradicting attacks; 3) empirically shows the advantages of our approach over existing aggregators by performing image classification under various attacks.

Outline. The remainder of the paper is organized as follows: In Section 2 we introduce the problem setup, some basic concepts and definitions. In Section 3, we introduce the proposed aggregators and prove their robust guarantees. In Section 4, we discuss two types of contradicting attacks which motivates the two-phase aggregating framework. In Section 5 we show empirical results, followed by the section for conclusion and future work.

We use the words robustness and resilience interchangeably in this paper except for formally defined concepts.

2 Byzantine Resilient Distributed Learning

In this section, we introduce the Byzantine ML problem, which is followed by a general resilient framework for distributed learning. Then, we recall some useful robust notions of aggregation rules, under which the distributed learning algorithms are provably resilient and convergent.

2.1 Problem Setup

Consider a server-worker distributed learning system with one central server and n workers. Each worker $i \in [n]$ possesses a local dataset consisting of m data points $\mathcal{D}_i := \{z_1^{(i)}, \ldots, z_m^{(i)}\}$. The server stores sets of model parameters and update vectors received from the workers. For a given ML model parameterized by $\theta \in \mathbb{R}^d$, each worker i has a local loss function $\mathcal{L}_i(\theta) := \frac{1}{m} \sum_{k=1}^{m} l(\theta, z_k^{(i)})$, where $l(\cdot, \cdot)$ represents the loss over a single data point. We assume $l(\cdot, \cdot)$ is differentiable with respect to the first argument, and each $L_i(\cdot)$ is L-smooth, that is, $\|\nabla \mathcal{L}_i(\theta_1) - \nabla \mathcal{L}_i(\theta_2)\| \leq L\|\theta_1 - \theta_2\|$, $\forall \theta_1, \theta_2 \in \mathbb{R}^d$.

We consider a standard adversarial setting where the server is honest and f workers with unknown identities are Byzantine (Lamport, Shostak, and Pease 2019). The Byzantine workers need not follow the given learning protocol and may behave arbitrarily in the learning process. However they cannot make other workers faulty, falsify the message of any other nodes, or block message passing between the server and any honest (or non-Byzantine) workers.

In real-world ML applications, the datasets held by the honest workers are typically heterogeneous (Shi et al. 2023).

In this work, we model data heterogeneity by the following standard assumption (Karimireddy, He, and Jaggi 2022).

Assumption 1 (**Bounded heterogeneity**). *Let $\mathcal{H}$ denote the set of indices of honest workers and $\mathcal{L}_{\mathcal{H}}(\theta) := |\mathcal{H}|^{-1} \sum_{i \in \mathcal{H}} \mathcal{L}_i(\theta)$. There exists a positive value G such that $\frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \|\nabla \mathcal{L}_i(\theta) - \nabla \mathcal{L}_{\mathcal{H}}(\theta)\|^2 \leq G^2, \ \forall \theta \in \mathbb{R}^d$.*

The goal of the server is to approximate a stationary point of $\mathcal{L}_{\mathcal{H}}(\theta)$. Throughout this process, the server iteratively updates a model based on the stochastic gradients received from the workers. To proceed, we introduce the concept of Byzantine resilience as follows.

Definition 1 ((f, ε)-**Byzantine resilience**). A learning algorithm is said (f, ε)-Byzantine resilient if, even in the presence of f Byzantine workers, it outputs $\hat{\theta}$ satisfying $\|\nabla \mathcal{L}_{\mathcal{H}}(\hat{\theta})\|^2 \leq \varepsilon$.

We note that (f, ε)-Byzantine resilience is generally not possible (for any ε) when $f \geq n/2$ (Liu, Gupta, and Vaidya 2021). Therefore, we assume an upper bound for the number of Byzantine workers $f < n/2$ in this work. Furthermore, the heterogeneous datasets render the Byzantine distributed ML much more challenging, as the incorrect gradients from Byzantine workers and the correct gradients from honest workers becomes more difficult to distinguish in this case; see (Karimireddy, He, and Jaggi 2022) for a detailed discussion and a lower bound of the training error.

2.2 Resilient Distributed Learning Algorithm

We recall a class of resilient distributed learning algorithms in Algorithm 1, which we call Resilient Aggregated Stochastic Heavy Ball (RASHB) in this work. This approach aggregates stochastic momentum in a resilient manner, which can be applied in both homogeneous (Farhadkhani et al. 2022) and heterogeneous (Allouah et al. 2023b) worker settings.

To proceed, we present four useful definitions for the robustness of aggregation rules in the literature, under which the convergence result of Algorithm 1 follows.

First, the definition of (f, λ)-resilient averaging was proposed in (Farhadkhani et al. 2022) for homogeneous data.

Definition 2 ((f, λ)-**resilient averaging**). Given an integer $f < n/2$ and a real number $\lambda \geq 0$, an aggregation rule F is called (f, λ)-resilient averaging if for any set of n vectors $X := \{x_i\}_{i=1}^n$, and any subset $S \subseteq [n]$ with $|S| = n - f$,

$$\|F(X) - \overline{x}_S\| \leq \lambda \max_{i,j \in S} \|x_i - x_j\|, \tag{1}$$

where $\overline{x}_S := \frac{1}{|S|} \sum_{i \in S} x_i$.

Second, to address the heterogeneity in worker's data, Karimireddy, He, and Jaggi proposed the concept of agnostic robust aggregator (ARAgg) for a (randomized or deterministic) aggregation rule.

Definition 3 (($\delta_{\max}, \zeta$)-**ARAgg**). Given a set of n vectors $X := \{x_i\}_{i=1}^n$ and a subset $S \subseteq [n]$ with $|S| = n - f$ with $f/n \leq \delta_{\max} < 0.5$ satisfying $\mathbb{E}\left[\|x_i - x_j\|^2\right] \leq \rho^2$ for all $i, j \in S$, the output $F(X)$ of a ($\delta_{\max}, \zeta$)-ARAgg satisfies

$$\mathbb{E}\left[\|F(X) - \overline{x}_S\|^2\right] \leq \zeta \frac{f}{n} \rho^2. \tag{2}$$

Algorithm 1: RASHB

Initialization: for server and worker: Initial model θ_0, initial momentum $m_0^{(1)} = 0$, the number of rounds T; for each honest worker w_i: , robust aggregation F, batch size b, learning rates $\{\gamma_t\}_{t=1}^T$, momentum coefficient β.

1: **for** $t = 0, \ldots, T - 1$ **do**
2:     **Server** broadcasts θ_t to all workers;
3:     **for** every honest worker w_i, $i \in \mathcal{H}$, in parallel **do**
4:         Compute a local stochastic gradient $g_t^{(i)}$ using mini-batch data samples;
5:         Update local momentum:

$$m_t^{(i)} = \beta m_t^{(i)} + (1 - \beta) g_t^{(i)};$$

6:         Send $m_t^{(i)}$ to the server;
7:     **end for**
8:     Server aggregates the received momentums:

$$R_t = F(\{m_t^{(1)}, \ldots, m_t^{(n)}\});$$

9:     Server updates the model: $\theta_t = \theta_{t-1} - \gamma_t R_t$;
10: **end for**
11: **return** $\frac{1}{T} \sum_{t=0}^{T-1} \theta_t$;

Third, a stronger notion of (f, κ)-robustness for aggregation rules was then proposed by the paper (Allouah et al. 2023a).

Definition 4 ((f, κ)-**robustness**). Given an integer $f < n/2$ and a real number $\kappa \geq 0$, an aggregation rule F is (f, κ)-robust if for any set of n vectors $X = \{x_i\}_{i=1}^n$, and any subset $S \subseteq [n]$ with $|S| = n - f$,

$$\|F(X) - \overline{x}_S\|^2 \leq \frac{\kappa}{|S|} \sum_{i \in S} \|x_i - \overline{x}_S\|^2. \tag{3}$$

Lastly, a recent work (Allouah et al. 2023b) introduced the (f, ξ)-robust averaging criterion. Compared with (f, κ)-robustness, it considers the maximum eigenvalue of the covariance matrix of a set of data points instead of its trace. Therefore it controls the deviation form honest values in *all* directions.

Definition 5 ((f, ξ)-**robust averaging**). Given an integer $f < n/2$ and a real number $\xi \geq 0$, an aggregation rule F is (f, ξ)-robust averaging if for any set of n vectors $X := \{x_i\}_{i=1}^n$, and any subset $S \subseteq [n]$ with $|S| = n - f$,

$$\|F(X) - \overline{x}_S\|^2 \leq \xi \cdot \lambda_{\max}(M_S), \tag{4}$$

where $\overline{x}_S := \frac{1}{|S|} \sum_{i \in S} x_i$; $\lambda_{\max}(\cdot)$ is the eigenvalue of a matrix; and $M_S := \frac{1}{|S|} \sum_{i \in S} (x_i - \overline{x}_S)(x_i - \overline{x}_S)^\top$.

Assumption 2 (**Bounded variance**). *For each honest worker i, there holds that $\frac{1}{m} \sum_{z \in \mathcal{D}_i} \|\nabla_\theta l(\theta, z) - \nabla \mathcal{L}_i(\theta)\|^2 \leq \sigma^2, \ \forall \theta \in \mathbb{R}^d$.*

We are in a position to present the convergence results for Algorithm 1, whose proof can be found in existing works that proposed the robustness definitions.

Theorem 1. *Suppose Assumptions 2 and 1 hold, and recall that $\mathcal{L}_{\mathcal{H}}(\cdot)$ is L-smooth. Consider Algorithm 1 and define $Res_T = T^{-1} \sum_{t=1}^{T} \mathbb{E}\left[\|\nabla \mathcal{L}_{\mathcal{H}}(\theta_{t-1})\|^2 \right]$.*

i) *If F is a (f, λ)-resilient aggregation rule and $G = 0$, then $Res_T \leq \mathcal{O}\left(\sqrt{(n-f)} \cdot \lambda \sigma / \sqrt{T} \right)$;*

ii) *If F is a (δ_{max}, ζ)-ARAgg aggregation rule and $G > 0$, then $Res_T \leq \mathcal{O}\left(\zeta f G^2 / n + \sigma \sqrt{\zeta f + 1} / \sqrt{nT} \right)$;*

iii) *If F is a (f, κ)-robust aggregation rule and $G > 0$, then $Res_T \leq \mathcal{O}\left(\kappa G^2 + \sigma / \sqrt{T} \right)$;*

iv) *If the aggregation rule F satisfies the condition of (f, ξ)-robust averaging and $G > 0$, then $Res_T \leq \mathcal{O}\left(\xi G^2 + \sigma / \sqrt{T} \right)$.*

Theorem 1 highlights the crucial significance of the resilient aggregation rule in Byzantine distributed learning. This rule not only ensures resilience against Byzantine workers but also influences the overall learning performance.

3 A Framework for Resilient Aggregation

In this section we develop a resilient aggregation algorithmic framework using 1-center and 1-mean clustering with outliers. We provide analysis for the proposed aggregation rules to show their near-optimality under various criteria.

3.1 1-Center/Mean Clustering with Outliers

The 1-*center clustering* problem is also referred to as the *minimum enclosing ball* problem. The problem is to find a ball with minimum radius containing all given points. The problem with outliers is defined as

Definition 6 (1-center clustering with outliers, or minimum enclosing ball with outliers). Given a set of n points X in $\mathbb{R}^d$, and an integer $f < n$ indicating the largest number of faulty points in X, find a ball $B(c, r)$ with a center $c \in \mathbb{R}^d$ and a radius $r \in \mathbb{R}$ to cover $(n - f)$ points in X, such that r is the minimum among all possible balls.

The minimum enclosing ball with outliers problem has been shown to be strongly NP-hard (Shenmaier 2013).

In the 1-*mean clustering problem*, given a set of n points X, the aim is to find a point c in the given space so as to minimize the sum of squared distances from each point $x \in X$ to c. The centroid (also called the center of mass) of a set X is defined as $cm(X) := 1/|X| \cdot \sum_{x_i \in X} x_i$. It is known that the centroid of a given set of points is the optimal solution to the problem. Now we introduce a variant of the problem in the presence of outliers.

Definition 7 (1-mean clustering with outliers). Given a set of n points X in $\mathbb{R}$, and the largest number of faulty nodes $f < n$, the problem is to find a vector $c \in \mathbb{R}^d$ so as to minimize $\sum_{x \in K(c)} \|c - x\|^2$, where $K(c)$ is the nearest $(n - f)$ points in X to c.

In a similar manner as the proof of hardness for the problem of minimum enclosing ball with outliers (Shenmaier

---

Algorithm 2: CenterwO/MeanwO

Input: a set of n vectors X in $\mathbb{R}^d$, and an integer $f < \frac{n}{2}$.
Output: the mass center of the $n - f$ points in an approximate minimum 1-center/mean cluster with f outliers.

1: **for** $i = 1, \ldots, n$ **do**
2:     Find $K(x_i)$, the $n - f$ closest vectors in X to the vector x_i (including x_i), breaking ties arbitrarily;
3:     Let $\text{dist}_i = \text{cost}(x_i, K(x_i))$;
        /*see (5) for definition of 1-center/mean cost*/
4: **end for**
5: Let $j \in \arg\min_i \text{dist}_i$;
6: **return** $\frac{1}{n-f} \sum_{x \in K(x_j)} x$;

2013), the problem of 1-means clustering with outliers can also be shown to be strongly NP-hard, by a reduction from the k-clique problem in regular graphs.

3.2 Efficient Approximation Algorithms

The two clustering problems considered are special cases of k-center/means clustering with outliers, which are central problems in both geometry and learning. Since both problems are NP-hard, it is natural to seek for approximations.

Definition 8. A polynomial time algorithm $\mathcal{A}$ is called an α-approximation algorithm for a (minimization) optimization problem if for all instances of the problem, $\mathcal{A}$ produces a solution whose value is guaranteed to be at most α times the optimum value.

In this paper we use simple approximation algorithms that consider all data points in X as cluster center candidates instead of all points in $\mathbb{R}^d$. Mostly in the k-means setup, cluster centers chosen from data points are also called medoids. Given a cluster center x and a set of data points S, we define the cost function

$$\text{cost}(c, S) := \begin{cases} \max_{x \in S} \|x - c\| & \text{(CenterwO)} \\ \sum_{x \in S} \|x - c\|^2 & \text{(MeanwO)} \end{cases} \quad (5)$$

for the approximate 1-center/mean clustering problem. The aggregation rules using approximations to the 1-center/mean clustering with outliers are shown in Algorithm 2.

Lemma 1. *The CenterwO and MeanwO algorithms are 2-approximations to the problems of 1-center and 1-mean clustering with outliers, respectively.*

The CenterwO algorithm was mentioned in many previous work, a proof for its approximation guarantee was given in (Shenmaier 2013). We prove the 2-approximation of MeanwO in the technical appendix. The running time of the CenterwO/MeanwO algorithm is $\mathcal{O}(n^2 d)$, and the memory usage is $\mathcal{O}(nd)$.

We note that the approximation algorithms for the considered problems can be improved in approximation ratio, running time, and memory usage. Examples of such improvements for the one center with outliers problem include a deterministic $\mathcal{O}(1)$-approximation algroithm with

$\mathcal{O}(nd)$ running time (Narayanan 2018) and a deterministic $\mathcal{O}(1)$-approximation streaming algorithm using $\mathcal{O}(fd)$ memory (McCutchen and Khuller 2008). Nevertheless, we adopt Algorithm 2 in our analysis and experiments.

3.3 Analysis of the Algorithms

The robust properties of the CenterwO and MeanwO aggregation rules are summarized in Theorem 2, the proof of which is shown in the technical appendix.

Theorem 2. *For any* $f < n/2$, $f/n \le \delta_{\max} < 1/2$, *and* $\nu \overset{\text{def}}{=} 1/2 - \delta_{\max}$, *the* CenterwO *algorithm is*

- $\left(f, \frac{(2\sqrt{2}+1)f}{n-f} \right)$-*resilient averaging,*
- $\left(\delta_{\max}, \frac{(18+8\sqrt{2})}{(1+2\nu)^2} \frac{f}{n} \right)$-*agnostic robust,*
- $\left(f, \frac{8f^2+2f}{n-2f} \cdot \frac{n-f}{n-2f} \right)$-*robust,*
- $\left(f, \frac{8f^2+2f}{n-2f} \cdot \frac{n-f}{n-2f} \right)$-*robust averaging;*

the MeanwO *algorithm is*

- $\left(f, \frac{\sqrt{3f(n-f)}}{(n-2f)} \right)$-*resilient averaging,*
- $\left(\delta_{\max}, \frac{3(1+2\nu)}{8\nu^2} \right)$-*agnostic robust,*
- $\left(f, \frac{6f}{n-2f} \cdot \frac{n-f}{n-2f} \right)$-*robust,*
- $\left(f, \frac{6f \cdot \min\{n-f,d\}}{n-2f} \cdot \frac{n-f}{n-2f} \right)$-*robust averaging.*

Remark 1. The (f, λ)-resilient aggregating propoerty of CenterwO matches the lower bound $\lambda \ge f/(n-f)$ shown in (Farhadkhani et al. 2022), up to a constant factor. The only previously known aggregator that matches this bound is MDA (El Mhamdi, Guerraoui, and Rouault 2018), which runs in $\mathcal{O}\left(\binom{n}{f} + n^2 d \right)$ time. The CenterwO algorithm is much more efficient with moderate n and f.

Remark 2. CenterwO algorithm is the first known $(\delta_{\max}, \mathcal{O}(f/n))$-agnostic robust aggregator. MeanwO is a $(\delta_{\max}, \mathcal{O}(1))$-agnostic robust aggregator, which matches the previously known best $\mathcal{O}(1)$ results achieved by CClip (not agnostic), Bucketing+Krum, and Bucketing+GM (Karimireddy, He, and Jaggi 2022).

Remark 3. The (f, κ)-robust aggregating propoerty of CenterwO matches the lower bound $\kappa \ge f/(n-2f)$ shown in (Allouah et al. 2023a) up to a constant factor, given that ν is a constant. The CWTM algorithm matches this bound without pre-aggregation steps, so does the SMEA algorithm (Allouah et al. 2023b). With the mixing pre-aggregation step, NNM+(GM/CWMed/CWTM/Krum) (Allouah et al. 2023a) matches this bound.

Although CWTM also achieves near-optimal (f, κ)-robustness, it removes $2f$ values in each coordinate, and therefore utilizes less information in the data, making the training algorithm potentially inefficient. Pre-aggregation steps can be applied prior to any aggregation rule, at the cost of additional processing time at the server end. SMEA and other eigenvector-based algorithms (Allouah et al. 2023b; Data and Diggavi 2021; Zhu et al. 2023) are generally more expensive, with $\Omega\left(\binom{n}{f} n^2 d \right)$ or $\Omega(nd^2)$ running time.

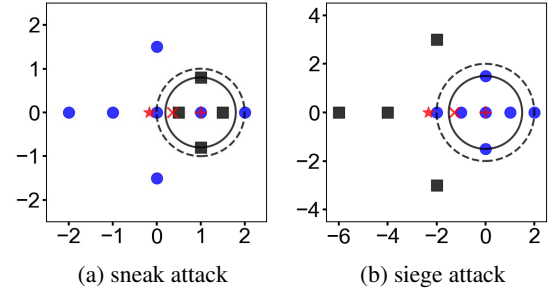(a) sneak attack   (b) siege attack

Figure 1: Two types of attacks produce a dilemma for any robust aggregation rule. Blue circles are update vectors produced by honest clients. Black squares show update vectors provided by Byzantine clients. Aggregated values of the naive averaging, inner-cluster and outer-cluster aggregator are denoted by the red symbol '×', '+', '⋆', respectively.

4 The Two-Phase Aggregation Framework

In this section, we propose a 2-phase aggregation framework motivated by a dilemma arising from a thought experiment.

4.1 Two Contradicting Types of Attacks

The Byzantine clients and the aggregation rule play a zero sum game to decide the model update in each round of the learning process. We provide a simple example to show that no fixed aggregation rule outperforms other strategies under *all* attacker strategies.

Suppose $n = 11$ and $f = 5$, and there are indeed 7 honest clients and 4 Byzantine clients. In Figure (1a) and Figure (1b) the blue circles show 7 update vectors produced by honest clients. We discuss two types of attacks to shift the average vector from the true mean $(0, 0)$. In Figure (1a), the attacker shift the average by placing 4 biased update vectors within the convex hull of the blue points. In Figure (1b), the attacker place 4 biased update vectors around the update vectors of honest clients. We call the first type of attack the *sneak attack* and the second type the *siege attack*.

If we remove all labels of the vectors, the two cases are *indistinguishable* for *all* aggregation algorithms without knowing the range of honest update values or their variance. Furthermore, any algorithm outperforms the naive average algorithm in the first case produces larger bias than the average algorithm in the second case, and vice versa. Therefore no aggregation rule dominates other rules under all attacks.

Although Figure (1a) and (1b) are artificially constructed examples, we claim that the issue exists in real update vector filtering problems. Real attacks can roughly be categorized into these two types, depending on whether the Byzantine vectors are placed within or out of the distribution of the update vectors of honest clients.

Most of the existing methods focus on preventing the siege attack by reducing the impact of outliers. This is because the siege attack can add unbounded bias to the model update, while the bias added by the sneak attack is restricted by the variance of the honest client updates. However, when the data distribution among clients are heterogeneous, the honest updates are not identically distributed. In this case the guarantees given by the optimization algorithms heavily

depend on the smoothness of the loss function, which may not hold in deep models used in practice.

Given the information of the type of attack used, we can use 1-center/means clustering with outliers algorithm to remove the faulty update vectors effectively. In Figure (1a), an outer-cluster averaging rule identifies an optimal cluster of f points, and returns the average of the vectors *outside* the cluster. In Figure (1b), an inner-cluster averaging rule identifies an optimal cluster of $(n - f)$ points, and returns the average of the vectors *inside* the cluster. As the figures show, if both aggregation rules are used, one of the rules returns a value close to the true average. It remains a problem to choose one from the two aggregated values.

4.2 The 2PRASHB Algorithm

We propose a two-phase aggregation framework to let the clients elect a model from two candidate models in each round. We note that if we let each client to directly adopt one model from the two proposed ones, it will result in multiple models in the system[1]. An instantiation of the framework is given in Algorithm 3. The algorithm is called the two-phase aggregation stochastic heavy ball algorithm (2PRASHB). In the first phase, the *prepare* phase, 1) the server broadcasts the current model; 2) the workers send their local updates to the server; 3) the server then proposes two models using the Inner and Outer aggregation algorithm[2]. In the second phase, the *voting* phase, 1) each honest client then samples a new batch of data to evaluate the loss of the two proposed models, and the honest clients then send their votes to the server[3]; 2) the server acknowledges the winning model.

We note that the server is not allowed to propose more than two candidate models for the clients to choose. This is because if there are more than two choices, vote splitting can happen between models with similar losses, and the Byzantine clients may win the election with $f < n/2$.

5 Experiments

In this section, we run simulations for an image classification task with a non-convex objective. We run four algorithms based on our study: Cent1P and Mean1P adopt the RASHB framework, using the CenterwO and MeanwO algorithm as the aggregator; Cent2P and Mean2P adopt the 2PRASHB framework. The Inner and Outer aggregators are instantiated by 1-center and 1-mean clustering with outliers. Limited by space, we leave more details about the experiments to the appendix.

Generation of datasets. The FEMNIST dataset, a standard benchmark for distributed and federated learning, is constructed by partitioning data in the EMNIST (Cohen et al. 2017) dataset. We sample 5% of the images in the original dataset to construct our datasets. The FEMNIST dataset has 805,263 images under 62 classes.For the homogeneous setting, each client sample images from a uniform distribution over 62 classes. We generate heterogeneous datasets for clients using categorical distributions q

---

Algorithm 3: 2PRASHB

Initialization: The same as Algorithm 1.

1: **for** $t = 0, \ldots, T - 1$ **do**
2:   **Server** broadcasts θ_t to all workers;
3:   Clients calculate and send momentums $\{m_t^{(i)}\}_{i=1}^n$ by executing line 2-7 in Algorithm 1;
4:   Server calculates two values from the received momentums:

$$R_t = \text{Inner}(\{m_t^{(1)}, \ldots, m_t^{(n)}\}), \quad (6)$$

$$Q_t = \text{Outer}(\{m_t^{(1)}, \ldots, m_t^{(n)}\}). \quad (7)$$

5:   Server proposes two updated models with parameters: $\widetilde{\theta}_t = \theta_{t-1} - \gamma_t R_t$ and $\widehat{\theta}_t = \theta_{t-1} - \gamma_t Q_t$, and send both $\widetilde{\theta}_t$ and $\widehat{\theta}_t$ to all clients;
6:   **for** every honest worker w_i, $i \in \mathcal{H}$, in parallel **do**
7:     Evaluate the loss of the two proposed models $\widetilde{\theta}_t$ and $\widehat{\theta}_t$ on a new mini-batch of data samples;
8:     Choose one set of parameters with smaller loss, and send its choice to the server;
9:   **end for**
10:   Server chooses the model which wins the popular vote (breaking ties arbitrarily), and sets it as θ_t;
11: **end for**
12: **return** $\frac{1}{T}\sum_{t=0}^{T-1}\theta_t$;

---

drawn from a Dirichlet distribution $q \sim \text{Dir}(\alpha p)$, where p is a prior class distribution over 62 classes (Hsu, Qi, and Brown 2019). Each client sample from a categorical distribution characterized by an independent q. In our experiment for the heterogeneous setting, we let $\alpha = 0.1$, which is described as the *extreme heterogeneity* setting in (Allouah et al. 2023a). For each worker, the training set and testing set are sampled independently from a distribution characterized by the same q. Due to space limitations, similar results for CIFAR10 (Krizhevsky 2009) are deferred to appendix.

Adversarial attacks. We run experiments for 3 levels of adversarial rates: 0.1, 0.2, and 0.4, i.e. 3, 7, and 14 out of 35 clients are corrupted. The honest workers are always honest during the learning process. The Byzantine workers send corrupted update vectors to the server, and vote for the model with larger loss in the voting phase if the two-phase framework is applied. We simulate 6 commonly studied adversarial attacks: the label flipping attack **LF**, the sign flipping attack **SF**, the random Gaussian attack **Gauss**, the omniscient attack **Omn** (Blanchard et al. 2017), the fall of empire attack **Empire** (Xie, Koyejo, and Gupta 2020), and the scaled variance attack **SV** (Baruch, Baruch, and Goldberg 2019; Allen-Zhu et al. 2021).We also customize a more sophisticated attack tailored to aggregation rules, PGA algorithm (Shejwalkar et al. 2022), to attack various algorithms. See appendix for complementary description about all attack algorithms.

Baselines. We compare the proposed algorithms with 6 baseline aggregation rules: the naive average (Avg), Geo-

---

[1]which is called a split brain state in distributed systems.
[2]See appendix for details.
[3]Byzantine workers can send any message or send nothing.

metric Median (GM) approximated by the Weiszfeld's algorithm (Pillutla, Kakade, and Harchaoui 2022) with the 1 iteration, Centered Clipping (CClip) (Karimireddy, He, and Jaggi 2021) with hyperparameters $v = 0$ and $\tau = 0.215771$, Coordinate-Wise Median (CWM) (Yin et al. 2018), Coordiante-Wise Trimmed Mean (CWTM) (Yin et al. 2018), and Krum (Blanchard et al. 2017). We apply all the baseline aggregators to the RASHB framework.

Architecture and hyperparameters. In our study, we employ a Convolutional Neural Network (CNN) comprising two convolutional layers (see appendix for details). with a learning rate of 0.1 and momentum of 0.99. The training process is carried out over 1500 rounds with a batch size of 3. We run all models with different aggregation rules under each attack five times, each with different random seeds. Finally, we report the averages of performance across these runs[4]. The implementation is based on RFA[5] and MEBwO[6].

Experimental results. Table 1 shows the performance of different aggregation algorithms and adversarial attacks on the uniform sampling dataset at an adversarial rate of 0.4. The results demonstrate the robustness of Cent2P and Mean2P, as they consistently achieved the highest worst performance across different attack scenarios. Specifically, Cent2P achieved a minimum accuracy of 0.62, while Mean2P achieved 0.61, both outperforming the third method, CClip, with a minimum accuracy of 0.20. Furthermore, only Cent2P and Mean2P achieved an accuracy above 0.75, whereas the accuracy of all the other methods remained below 0.20 and 0.32 under the Omn and SV attack scenarios respectively. These results underscore the remarkable effectiveness of Cent2P and Mean2P in maintaining strong performance, even when confronted with challenging adversarial conditions.

Aggregation	LF	SF	Gauss	Omn	Empire	SV	Worst
Avg	0.56	0.04	**0.79**	0.00	**0.79**	0.32	0.00
GM	0.64	0.46	**0.79**	0.00	0.74	0.07	0.00
CClip	0.58	0.45	0.64	0.20	0.26	0.26	0.20
CWM	0.50	0.45	0.54	0.02	0.07	0.05	0.02
CWTM	0.51	0.39	0.55	0.02	0.05	0.05	0.02
Krum	0.53	0.36	0.47	0.10	0.01	0.05	0.01
Cent2P	**0.74**	**0.62**	0.76	**0.79**	0.74	**0.76**	**0.62**
Mean2P	0.73	0.61	0.76	**0.79**	0.73	0.75	0.61

Table 1: Performance comparison on the uniform sampling datasets at an adversarial rate of 0.4.

Figure 2 illustrates the testing accuracy on the heterogeneous datasets at an adversarial rate of 0.2. The results highlight that when subjected to the SF and Omn attacks, Cent2P and Mean2P exhibit significant advantages over all other aggregation rules. When facing the SV attack, Cent2P and Mean2P display a slight advantage over Avg and much outperform the remaining baseline methods. Conversely, Avg

[4]Standard deviations across runs are shown in the appendix.
[5]https://github.com/krishnap25/tRFA
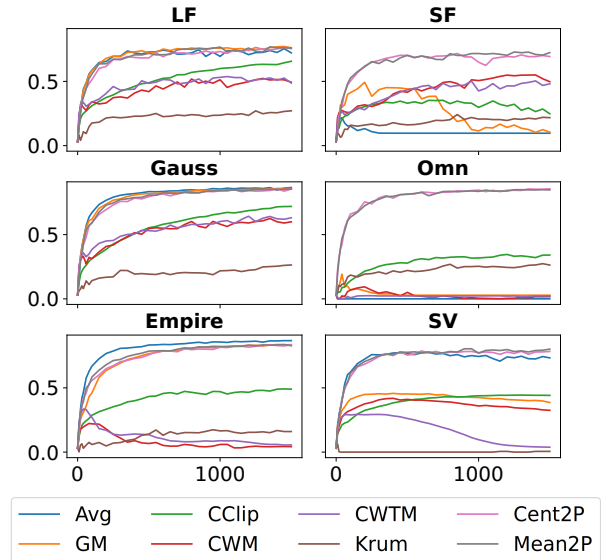[6]https://github.com/tomholmes19/Minimum-Enclosing-Balls-with-Outliers

Figure 2: Performance comparison on the heterogeneous datasets at an adversarial rate of 0.2, with the x, y axis representing testing accuracy and step number, respectively.

achieves the best performance under the Empire attack, while certain resilient aggregation rules experience significant degradation. Cent2P, Mean2P, and GM yield comparable results to Avg. In the presence of LF and Gauss attacks, Cent2P, Mean2P, Avg, and GM stand out as having the highest testing accuracy.

6 Conclusion and Future Work

We have proposed a near-optimal resilient aggregation framework based on 1-center and 1-mean clustering with outliers. Approximation algorithms have been applied to achieve both efficiency and resilience/robustness. We have proven safety guarantees provided by the proposed algorithms. We have proposed a two-phase resilient aggregation framework based on the observation that no single aggregation rule outperforms other rules against two contradicting types of attacks. We have shown the advantages of the proposed approaches by running numerical simulations for image classification with non-convex loss in the homogeneous and heterogeneous settings. Future work may study the resilience of other outlier-robust clustering methods, and the theoretical guarantee of the 2PRASHB framework.

References

Agrawal, A.; Inamdar, T.; Saurabh, S.; and Xue, J. 2023. Clustering What Matters: Optimal Approximation for Clustering with Outliers. *In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'23)*, 37(6): 6666–6674.

Allen-Zhu, Z.; Ebrahimianghazani, F.; Li, J.; and Alistarh, D. 2021. Byzantine-Resilient Non-Convex Stochastic Gradient Descent. In *International Conference on Learning Representations (ICLR'21)*.

Allouah, Y.; Farhadkhani, S.; Guerraoui, R.; Gupta, N.; Pinot, R.; and Stephan, J. 2023a. Fixing by Mixing: A Recipe for Optimal Byzantine ML under Heterogeneity. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics (AISTATS'23)*, volume 206 of *Proceedings of Machine Learning Research*, 1232–1300. PMLR.

Allouah, Y.; Guerraoui, R.; Gupta, N.; Pinot, R.; and Stephan, J. 2023b. On the Privacy-Robustness-Utility Trilemma in Distributed Learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*, volume 202 of *Proceedings of Machine Learning Research*. PMLR.

Banerjee, S.; Ostrovsky, R.; and Rabani, Y. 2021. Min-Sum Clustering (With Outliers). In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, (APPROX/RANDOM'21)*, volume 207 of *LIPIcs*, 16:1–16:16. Seattle, Washington, USA (Virtual Conference): Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Baruch, G.; Baruch, M.; and Goldberg, Y. 2019. A Little Is Enough: Circumventing Defenses For Distributed Learning. In *Advances in Neural Information Processing Systems (NeurIPS'19)*, volume 32. Curran Associates, Inc.

Blanchard, P.; El Mhamdi, E. M.; Guerraoui, R.; and Stainer, J. 2017. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *Advances in Neural Information Processing Systems (NeurIPS'17)*, volume 30. Curran Associates, Inc.

Chen, Y.; Su, L.; and Xu, J. 2017. Distributed Statistical Machine Learning in Adversarial Settings: Byzantine Gradient Descent. *Proc. ACM Meas. Anal. Comput. Syst.*, 1(2).

Cohen, G.; Afshar, S.; Tapson, J.; and van Schaik, A. 2017. EMNIST: Extending MNIST to handwritten letters. In *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN'17)*, 2921–2926.

Data, D.; and Diggavi, S. 2021. Byzantine-Resilient High-Dimensional SGD with Local Iterations on Heterogeneous Data. In *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, volume 139 of *Proceedings of Machine Learning Research*, 2478–2488. PMLR.

Ding, H. 2020. A Sub-Linear Time Framework for Geometric Optimization with Outliers in High Dimensions. In *28th Annual European Symposium on Algorithms (ESA'20)*, volume 173 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 38:1–38:21. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

El Mhamdi, E. M.; Guerraoui, R.; and Rouault, S. 2018. The Hidden Vulnerability of Distributed Learning in Byzantium. In *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, volume 80 of *Proceedings of Machine Learning Research*, 3521–3530. PMLR.

Farhadkhani, S.; Guerraoui, R.; Gupta, N.; Pinot, R.; and Stephan, J. 2022. Byzantine Machine Learning Made Easy By Resilient Averaging of Momentums. In *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, volume 162 of *Proceedings of Machine Learning Research*, 6246–6283. PMLR.

Friggstad, Z.; Khodamoradi, K.; Rezapour, M.; and Salavatipour, M. R. 2019. Approximation Schemes for Clustering with Outliers. *ACM Transactions on Algorithms*, 15(2).

Ghosh, A.; Hong, J.; Yin, D.; and Ramchandran, K. 2019. Robust federated learning in a heterogeneous environment. *arXiv:1906.06629*.

Har-Peled, S. 2011. *Geometric approximation algorithms*. 173. American Mathematical Soc.

Hsu, T.-M. H.; Qi, H.; and Brown, M. 2019. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; D'Oliveira, R. G. L.; Eichner, H.; Rouayheb, S. E.; Evans, D.; Gardner, J.; Garrett, Z.; Gascón, A.; Ghazi, B.; Gibbons, P. B.; Gruteser, M.; Harchaoui, Z.; He, C.; He, L.; Huo, Z.; Hutchinson, B.; Hsu, J.; Jaggi, M.; Javidi, T.; Joshi, G.; Khodak, M.; Konecný, J.; Korolova, A.; Koushanfar, F.; Koyejo, S.; Lepoint, T.; Liu, Y.; Mittal, P.; Mohri, M.; Nock, R.; Özgür, A.; Pagh, R.; Qi, H.; Ramage, D.; Raskar, R.; Raykova, M.; Song, D.; Song, W.; Stich, S. U.; Sun, Z.; Suresh, A. T.; Tramèr, F.; Vepakomma, P.; Wang, J.; Xiong, L.; Xu, Z.; Yang, Q.; Yu, F. X.; Yu, H.; and Zhao, S. 2021. Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning*, 14(1–2): 1–210.

Karimireddy, S. P.; He, L.; and Jaggi, M. 2021. Learning from History for Byzantine Robust Optimization. In *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, volume 139 of *Proceedings of Machine Learning Research*, 5311–5319. PMLR.

Karimireddy, S. P.; He, L.; and Jaggi, M. 2022. Byzantine-Robust Learning on Heterogeneous Datasets via Bucketing. In *International Conference on Learning Representations (ICLR'22)*.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images.

Lamport, L.; Shostak, R.; and Pease, M. 2019. *The Byzantine Generals Problem*, 203–226. New York, NY, USA: Association for Computing Machinery. ISBN 9781450372701.

Li, L.; Xu, W.; Chen, T.; Giannakis, G. B.; and Ling, Q. 2019. RSA: Byzantine-Robust Stochastic Aggregation Methods for Distributed Learning from Heterogeneous Datasets. *In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'19)*, 33(01): 1544–1551.

Liu, S.; Gupta, N.; and Vaidya, N. H. 2021. Approximate Byzantine fault-tolerance in distributed optimization. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing (PODC'21)*, 379–389. ACM.

McCutchen, M. R.; and Khuller, S. 2008. Streaming Algorithms for k-Center Clustering with Outliers and with Anonymity. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM'08)*, 165–178. Berlin, Heidelberg: Springer Berlin Heidelberg.

Narayanan, S. 2018. Deterministic O(1)-Approximation Algorithms to 1-Center Clustering with Outliers. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM'18)*, volume 116 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 21:1–21:19. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Pillutla, K.; Kakade, S. M.; and Harchaoui, Z. 2022. Robust Aggregation for Federated Learning. *IEEE Transactions on Signal Processing*, 70: 1142–1154.

Sattler, F.; Müller, K.-R.; Wiegand, T.; and Samek, W. 2020. On the Byzantine Robustness of Clustered Federated Learning. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'20)*, 8861–8865. IEEE.

Shejwalkar, V.; Houmansadr, A.; Kairouz, P.; and Ramage, D. 2022. Back to the Drawing Board: A Critical Evaluation of Poisoning Attacks on Production Federated Learning. In *Proc. IEEE Symposium on Security and Privacy (SP'22)*, 1354–1371. IEEE.

Shenmaier, V. 2013. Complexity and approximation of the smallest k-enclosing ball problem. In *Proceedings of the 7th European Conference on Combinatorics, Graph Theory and Applications (Eurocomb'13)*, 583–588. Scuola Normale Superiore.

Shi, M.; Zhou, Y.; Wang, K.; Zhang, H.; Huang, S.; Ye, Q.; and Lv, J. 2023. PRIOR: Personalized Prior for Reactivating the Information Overlooked in Federated Learning. In *Advances in Neural Information Processing Systems (NeurIPS'23)*.

Smallwood, R. D. 1965. Minimax Detection Station Placement. *Operations Research*, 13(4): 632–646.

Warnat-Herresthal, S.; Schultze, H.; Shastry, K. L.; Manamohan, S.; Mukherjee, S.; Garg, V.; Sarveswara, R.; Händler, K.; Pickkers, P.; Aziz, N. A.; et al. 2021. Swarm learning for decentralized and confidential clinical machine learning. *Nature*, 594(7862): 265–270.

Xie, C.; Koyejo, O.; and Gupta, I. 2020. Fall of Empires: Breaking Byzantine-tolerant SGD by Inner Product Manipulation. In Adams, R. P.; and Gogate, V., eds., *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference (UAI'20)*, volume 115 of *Proceedings of Machine Learning Research*, 261–270. PMLR.

Yildirim, E. A. 2008. Two Algorithms for the Minimum Enclosing Ball Problem. *SIAM Journal on Optimization*, 19(3): 1368–1391.

Yin, D.; Chen, Y.; Kannan, R.; and Bartlett, P. 2018. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, volume 80 of *Proceedings of Machine Learning Research*, 5650–5659. PMLR.

Zhu, B.; Wang, L.; Pang, Q.; Wang, S.; Jiao, J.; Song, D.; and Jordan, M. I. 2023. Byzantine-Robust Federated Learning with Optimal Statistical Rates. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics (AISTATS'23)*, volume 206 of *Proceedings of Machine Learning Research*, 3151–3178. PMLR.