Learning from History for Byzantine Robust Optimization

Sai Praneeth Karimireddy [1] **Lie He** [1] **Martin Jaggi** [1]

Abstract

Byzantine robustness has received significant attention recently given its importance for distributed and federated learning. In spite of this, we identify severe flaws in existing algorithms even when the data across the participants is identically distributed. First, we show realistic examples where current state of the art robust aggregation rules fail to converge even in the absence of any Byzantine attackers. Secondly, we prove that even if the aggregation rules may succeed in limiting the influence of the attackers in a single round, the attackers can couple their attacks across time eventually leading to divergence. To address these issues, we present two surprisingly simple strategies: a new robust *iterative clipping* procedure, and incorporating *worker momentum* to overcome time-coupled attacks. This is the first provably robust method for the standard stochastic optimization setting. Our code is open sourced at this link[2].

1. Introduction

> *"Those who cannot remember the past are condemned to repeat it."* – George Santayana.

Growing sizes of datasets as well as concerns over data ownership, security, and privacy have lead to emergence of new machine learning paradigms such as distributed and federated learning (Kairouz et al., 2019). In both of these settings, a central coordinator orchestrates many worker nodes in order to train a model over data which remains decentralized across the workers. While this decentralization improves scalability security and privacy, it also opens up the training process to manipulation by the workers (Lamport et al., 2019). These workers may be actively malicious trying to derail the process, or might simply be malfunctioning and hence sending arbitrary messages. Ensuring that our training procedure is robust to a small fraction of such potentially malicious agents is termed Byzantine robust learning and is the focus of the current work.

Given the importance of this problem, it has received significant attention from the community with early works including (Feng et al., 2014; Blanchard et al., 2017; Chen et al., 2017; Yin et al., 2018). Most of these approaches replace the averaging step of distributed or federated SGD with a robust aggregation rule such as the median. However, a closer inspection reveals that these procedures are quite brittle: we show that there exist realistic scenarios where they fail to converge, even if there are *no Byzantine attackers* and the data distribution is identical across the workers (i.i.d.). This turns out to be because on their excessive sensitivity to the distribution of the noise in the gradients. The impractical assumptions made by these methods are often violated in practice, and lead to the failure of these aggregation rules.

Further, there have been recent state of the art attacks (Baruch et al., 2019; Xie et al., 2020) which empirically demonstrate a second source of failure. They show that even when current aggregation rules may succeed in limiting the influence of the attackers in any single round, they may still diverge when run for multiple rounds. We prove that this is inevitable for a wide class of methods—any aggregation rule which ignores history can be made to eventually diverge. This is accomplished by using the inherent noise in the gradients to mask small perturbations which are undetectable in a single round, but accumulate over time.

Finally, we show how to circumvent both the issues outlines above. We first describe a simple new aggregator based on iterative *centered clipping* which is much more robust to the distribution of the gradient noise. This aggregator is especially interesting since, unlike most preceding methods, it is very scalable requiring only $\mathcal{O}(n)$ computation and communication per round. Further, it is also compatible with other strategies such as asynchronous updates (Chen et al., 2016) and secure aggregation (Bonawitz et al., 2017), both of which are crucial for real world applications. Secondly, we show that the time coupled attacks can easily be overcome by using *worker momentum*.

[1]EPFL, Switzerland. Correspondence to: Sai Praneeth Karimireddy <sai.karimireddy@epfl.ch>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[2]https://github.com/epfml/byzantine-robust-optimizer

Momentum averages the updates of each worker over time, reducing the variance of the good workers and exposing the time-coupled perturbations. We prove that our methods obtain optimal rates, and our theory also sheds light on the role of momentum in decreasing variance and building resilience to Byzantine workers.

Contributions. Our main results are summarized below.

- We show that most state of the art robust aggregators require strong assumptions and can fail in real settings even in the complete absence of Byzantine workers.
- We prove a strong lower bound showing that any optimization procedure which does not use history will diverge in the presence of time coupled attacks.
- We propose a simple and efficient aggregation rule based on iterative clipping and prove its performance under standard assumptions.
- We show that using momentum successfully defends against time-coupled attacks and provably converges when combined with any Byzantine robust aggregator.
- We incorporate the recent momentum based variance reduction (MVR) with Byzantine aggregators to obtain optimal rates for robust non-convex optimization.
- We perform extensive numerical experiments validating our techniques and results.

Setup. Let us formalize the robust non-convex stochastic optimization problem in the presence of a δ fraction of Byzantine workers.

Definition A (δ-robust non-convex optimization)**.** *Given some loss function $f(\boldsymbol{x})$, $\epsilon > 0$, and access to n workers we want to find a stationary point $\boldsymbol{x}$ such that $\mathbb{E}\|\nabla f(\boldsymbol{x})\|^2 \leq \epsilon$. The optimization proceeds in rounds where in every round, each worker $i \in [n]$ can compute a stochastic gradient $g_i(\boldsymbol{y})$ at any parameter $\boldsymbol{y}$ in parallel. Then, each worker $i \in [n]$ sends some message $\mathcal{M}_{i,t}$ to the server. The server utilizes these messages to update the parameters and proceeds to the next round. During this process, we will assume that*

- *The function f is L-smooth i.e. it satisfies $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$ for any $\boldsymbol{x}, \boldsymbol{y}$, and is bounded from below by $f^\star$.*
- *Each worker i has access to an independent and unbiased stochastic gradient with $\mathbb{E}[g_i(\boldsymbol{x})|\boldsymbol{x}] = \nabla f(\boldsymbol{x})$ and variance bounded by σ^2, $\mathbb{E}\|g_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2 \leq \sigma^2$.*
- *Of the n workers, at least $(1-\delta)n$ workers are* good *(denoted by $\mathcal{G}$) and will follow the protocol faithfully. The rest of the* bad *or Byzantine workers (denoted by $\mathcal{B}$) may act maliciously and can communicate arbitrary messages to the server.*
- *These Byzantine workers are assumed to omniscient i.e. they have access to the computations made by the rest of the good workers. However, we assume that this set of Byzantine workers $\mathcal{B}$* remains fixed *throughout the optimization process.*

2. Related work

Robust aggregators. Distributed algorithms in the presence of Byzantine agents has a long history (Lamport et al., 2019) and is becoming increasingly important in modern distribution and federated machine learning (Kairouz et al., 2019). Most solutions involve replacing the averaging of the updates from the different machines with more robust aggregation rules such as coordinate-wise median method (Yin et al., 2018), geometric median methods (Blanchard et al., 2017; Chen et al., 2017; Pillutla et al., 2019), majority voting (Bernstein et al., 2018; Jin et al., 2020) etc. There have also been attempts to use recent breakthroughs in robust high-dimensional aggregators (Diakonikolas et al., 2018; Su & Xu, 2018; El-Mhamdi & Guerraoui, 2019; Data et al., 2019; Data & Diggavi, 2020). However, these latter procedures are computationally expensive (quadratic in dimensions per round) and further it is unclear if the improved guarantees for mean estimation translate to improved performance in the distributed machine learning settings. Finally, for most of the above approaches, convergence guarantees when provided rely on using an extremely large batch size or strong unrealistic assumptions making them practically irrelevant.

Other more heuristic approaches propose to use a penalization or reweighting of the updates based on reputations (Peng & Ling, 2020; Li et al., 2019; Fu et al., 2019; Regatti & Gupta, 2020; Rodríguez-Barroso et al., 2020). These schemes however need to trust that all workers report correct statistics. In such settings where we have full control over the workers (e.g. within a datacenter) coding theory based solutions which can correct for the mistakes have also been proposed (Chen et al., 2018; Rajput et al., 2019; Gupta & Vaidya, 2019; Konstantinidis & Ramamoorthy, 2020; Data et al., 2018; 2019). These however are not applicable in federated learning where the data is decentralized across untrusted workers.

Time coupled attacks and defenses. Recently, two state-of-the-art attacks have been proposed which show that the state of the art Byzantine aggregation rules can be easily circumvented (Baruch et al., 2019; Xie et al., 2020). The key insight is that while the robust aggregation rules may ensure that the influence of the Byzantine workers in any single round is limited, the attackers can couple their attacks across the rounds. This way, over many training rounds the attacker is able to move weights significantly away from the desired direction and thus achieve the goal of lowering the model quality. Defending against time-coupled attacks and showing provable guarantees is one of

the main concerns of this work.

It is clear that time-coupled attacks need time-coupled defenses. Closest to our work is that of Alistarh et al. (2018) who use martingale concentration across the rounds to give optimal Byzantine robust algorithms for convex functions. However, this algorithm is inherently not applicable to more general non-convex functions. The recent independent work of Allen-Zhu et al. (2021) extend the method of Alistarh et al. (2018) to non-convex functions as well. However, they assume that the noise in stochastic gradients is bounded almost surely instead of the more standard assumption that only the variance is bounded. Theoretically, such strong assumptions are unlikely to hold (Zhang et al., 2019) and even Gaussian noise is excluded. Further, the lower-bounds of (Arjevani et al., 2019) no longer apply, and thus their algorithm may be sub-optimal. Practically, their algorithm removes suspected workers either permanently (a decision of high risk), or resets the list of suspects at each window boundary (which is sensitive to the choice of hyperparameters). Having said that, (Allen-Zhu et al., 2021) prove convergence to a local minimum instead of to a saddle point as we do here. Finally, in another independent work El-Mhamdi et al. (2021) empirically observe that using momentum may be beneficial, though they provide no theoretical guarantees.

Other concerns. To deploy robust learning for real world applications, many other issues such as data heterogeneity become important (Kairouz et al., 2019; Karimireddy et al., 2020b). Robust learning algorithms which assume worker data are i.i.d. may fail in the federated learning setting (He et al., 2020a). Numerous variations have been proposed which can handle non-iid data with varying degrees of success (Li et al., 2019; Ghosh et al., 2019; Chen et al., 2019; Peng et al., 2020; Data & Diggavi, 2020; He et al., 2020a; El-Mhamdi et al., 2020; Dong et al., 2020). Further, combining robustness with notions of privacy and security is also a crucial and challenging problem (He et al., 2020b; So et al., 2020a;b; Jin et al., 2020). Such heterogeneity is especially challenging and can lead to backdoor attacks (which are orthogonal to the training attacks discussed here) (Bagdasaryan et al., 2019; Sun et al., 2019; Wang et al., 2020) and remains an open challenge.

3. Brittleness of existing aggregation rules

In this section, we study the robustness of existing popular Byzantine aggregation rules. Unfortunately, we come to a surprising conclusion—most state of the art aggregators require strong non-realistic restrictions on the noise distribution. We show this frequently does not hold in practice, and present counter-examples where these aggregators fail even in the complete *absence* of Byzantine workers. State of the art aggregators such as Krum (Blanchard et al., 2017), coordinate-wise median (CW) (Yin et al., 2018),

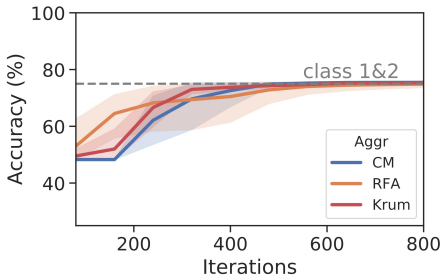

Figure 1: Failure of existing methods on imbalanced MNIST dataset. Only the head classes (class 1 and 2 here) are learnt, and the rest 8 classes are ignored. See Sec. 7.1.

RFA (Pillutla et al., 2019), Bulyan (Mhamdi et al., 2018), etc. all generalize the scalar notion of the median to higher dimensions and are hence exhibit different ways of 'middle-seeking'. At a high level, these schemes require the noise distribution to be unimodal and highly concentrated, discarding any gradients from the tail of the distribution too aggressively as 'outliers'. We give a brief summary of these rules below. We use $[\boldsymbol{v}]_j$ to indicate the jth coordinate of vector $\boldsymbol{v}$.

Coordinate-wise median:

$$[\text{CM}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)]_j = \text{median}([\boldsymbol{x}_1]_j, \dots, [\boldsymbol{x}_n]_j)\,.$$

RFA (robust federated averaging) aka geometric median:

$$\text{RFA}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \arg\min_{\boldsymbol{v}} \sum_{i=1}^{n} \|\boldsymbol{v} - \boldsymbol{x}_i\|_2\,.$$

Trimmed Mean: For each coordinate j, compute sorting Π_j which sorts the coordinate values. Compute the average after excluding ('trimming') δn largest and smallest values.

$$[\text{TM}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)]_j = \frac{1}{n - 2\delta n} \sum_{i=\delta n}^{n-\delta n} [\boldsymbol{x}_{\Pi_j(i)}]_j\,.$$

Krum: Krum tries to select a point $\boldsymbol{x}_i$ which is closest to the mean after excluding $\delta n + 2$ furthest away points. Suppose that $\mathcal{S} \subset [n]$ of size at least $(n - \delta n - 2)$. Then,

$$\text{Krum}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \arg\min_{\boldsymbol{x}_i} \min_{\mathcal{S}} \sum_{j \in \mathcal{S}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2\,.$$

Counterexample 1. Let us pick n random variables ± 1 with uniform probability for some odd n. These variables have mean 0. Since n is odd, Krum, CW, Bulyan all will necessarily return either of ± 1. This remains true even if we have infinite samples (large n), and if there are no corruptions. This simple examples illustrates the fragility of such 'middle-seekers' to bimodal noise.

Counterexample 2. Fig. 1 illustrates a more realistic example where imbalanced MNIST dataset causes a similar problem. Here, 0.5 fraction of data corresponds to class 1, 0.25 to class 2, and so on. The gradients over data of the same class are much closer than those of a different class. Hence, when we pick n i.i.d. gradients, most them will belong to class 1 or 2 with very few belonging to the rest. Thus, coordinate-wise median, geometric median and Krum always select the gradient corresponding to classes 1 or 2, ensuring that we only optimize over these classes ignoring the rest.

Counterexample 3. Middle-seekers can also fail on continuous uni-modal distributions. Consider,

$$p(x) = \begin{cases} 3x^{-4} & \text{for } x \geq 1 \\ 0 & \text{o.w.} \end{cases}$$

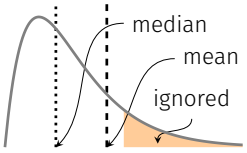

Figure 2: For fat-tailed distributions, median based aggregators ignore the tail. This bias remains even if we have infinite samples.

This power-law distribution has mean 1.5 and variance 0.75. However, since the distribution is skewed, its median is $2^{1/3} \approx 1.26$ and is smaller than the mean. This difference persists even with *infinite* samples showing that with imbalanced (i.e. skewed) distributions, coordinate-wise median, geometric median and Krum do not obtain the true optimum. Empirical evidence suggests that such heavy-tailed distributions abound in deep learning, making this setting very relevant to practice (Zhang et al., 2019).

Theorem I (Failure of 'middle-seekers')**.** *There exist simple convex stochastic optimization settings with bounded variance where traditional distributed SGD converges but coordinate-wise median, RFA, and Krum do not converge to the optimum almost surely for any number of workers and even if none of them are Byzantine.*

Remark 1 (Practical usage)**.** *Theorem I notes that one must be cautious while using median or Krum as aggregation rules when we suspect that our data is multi-modal (typically occurs when using small batch sizes), or if we believe our data to be heavy-tailed (typically occurs in imbalanced datasets or language tasks). These aggregators may suffice for standard image recognition tasks with large batch sizes since the noise is nearly Gaussian (Zhang et al., 2019).*

Median based aggregators have a long and rich history in the field of robust statistics (Minsker et al., 2015). However, classically the focus of robust statistics has been to design methods which can withstand a large fraction of Byzantine workers (high *break down* point $\delta_{\max}$) and not result in infinities (Hubert et al., 2008). It was sufficient for the output to be bounded, but the quality of the result was not a concern. The counter examples in this section exactly stem from this issue. We will later define a finer notion of a robust statistic which accounts for both the quality of the output as well as the breakdown point $\delta_{\max}$.

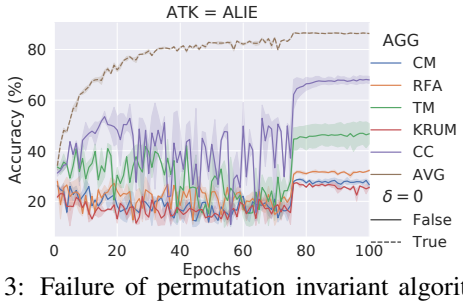

Figure 3: Failure of permutation invariant algorithms on CIFAR10 dataset with (Baruch et al., 2019) attack. Comparing to simple average with no attacker (dashed lines), all robust aggregators (including centered clip) see a significant drop in accuracy against time coupled attacks. See Sec. 7.2.

4. Necessity of using history

Recent work (Baruch et al., 2019; Xie et al., 2020) has shown a surprising second source vulnerability for most currently popular robust aggregators. In this section we take a closer look at their attack and use our observations to make an even stronger claim—any aggregation rule which is oblivious of the past cannot converge to the optimum and retains a non-zero error even after infinite time.

The inner-product manipulation attack as defined by (Baruch et al., 2019; Xie et al., 2020) is deceptively simple. Their attacks works by hiding small Byzantine perturbations within the variance of the good gradients. Since we only have access to noisy stochastic gradients, the aggregators fail to identify these perturbations. While this perturbation is small in any single round, these can accumulate over time. We formalize this argument into a lower bound in Theorem III. We show that the key reason why this attack works on algorithms such as CM, RFA, or Krum is that they are *oblivious* and do not track information from previous rounds. Thus, an attacker can couple the perturbations across time eventually leading to divergence. This is also demonstrated experimentally in Fig. 3.

Definition B (Permutation invariant algorithm)**.** *Suppose we are given an instance of δ-robust optimization problem satisfying Definition A. Define the set of stochastic gradients computed by each of the n workers at some round t to be $[\tilde{\boldsymbol{g}}_{1,t}, \ldots, \tilde{\boldsymbol{g}}_{n,t}]$. For a good worker $i \in \mathcal{G}$, these represent the true stochastic gradients whereas for a bad worker $j \in \mathcal{B}$, these represent arbitrary vectors. The output of any optimization algorithm* ALG *is a function of these gradients. A permutation-invariant algorithm is one which for any set of permutations over t rounds $\{\pi_1, \ldots, \pi_t\}$, its out-*

put remains unchanged if we permute the gradients.

$$\text{ALG}\begin{pmatrix}[\tilde{\boldsymbol{g}}_{1,1},\ldots,\tilde{\boldsymbol{g}}_{n,1}],\\ \ldots\\ [\tilde{\boldsymbol{g}}_{1,t},\ldots,\tilde{\boldsymbol{g}}_{n,t}]\end{pmatrix} = \text{ALG}\begin{pmatrix}[\tilde{\boldsymbol{g}}_{\pi_1(1),1},\ldots,\tilde{\boldsymbol{g}}_{\pi_1(n),1}],\\ \ldots\\ [\tilde{\boldsymbol{g}}_{\pi_t(1),t},\ldots,\tilde{\boldsymbol{g}}_{\pi_t(n),t}]\end{pmatrix}$$

Remark 2 (Memoryless methods are permutation invariant)**.** *Any algorithm which is 'memoryless' i.e. uses only the computations resulting from current round is necessarily permutation-invariant since the indices corresponding to the stochastic gradient are meaningless. It is only when these stochastic gradients are tracked over multiple rounds (i.e. we use memory) do the indices carry information.*

Theorem II (Failure of permutation-invariant methods)**.** *Suppose we are given any permutation invariant algorithm* AGG *as in Definition B, $\mu \geq 0$, $\delta \in [0, 1]$, and n large enough that $\delta n \geq 4(1 + \log t)$. Then, there exists a δ-robust μ strongly-convex optimization problem satisfying Definition A, such that the output $\tilde{\boldsymbol{x}}_t$ of* ALG *after t rounds necessarily has error*

$$\mathbb{E}[f(\tilde{\boldsymbol{x}}_t)] - f(\boldsymbol{x}^\star) \geq \Omega\left(\frac{\delta\sigma^2}{\mu}\right).$$

Nearly all currently popular aggregation rules, including coordinate-wise median, trimmed mean (Yin et al., 2018), Krum (Blanchard et al., 2017), Bulyan (Mhamdi et al., 2018), RFA, geometric median (Ghosh et al., 2019), etc. are permutation invariant and satisfy Definition B. Theorem II proves a very startling result—all of them *fail to converge* to the optimum even for strongly-convex problems. Further, as μ decreases (the problem becomes less strongly-convex), the error becomes unbounded.

Remark 3 (Fixed Byzantine workers)**.** *The failure of permutation-invariant algorithms also illustrates the importance of assuming that the indices of Byzantine workers are fixed across rounds. If a different fraction of workers are allowed to be Byzantine each round, then the lower bound in Theorem II applies to all algorithms and convergence is impossible. While it is indeed a valid concern that Byzantine workers may pretend to be someone else (or more generally perform Sybil attacks where they pretend to be multiple workers), simple mechanisms such as pre-registering all participants (perhaps using some identification) can circumvent such attacks.*

There are very few methods which are not permutation invariant and are not subject to our lower bound. Examples include Byzantine SGD (Alistarh et al., 2018) which only works for convex problems, and some heuristic scoring rules such as (Regatti & Gupta, 2020). There has also been a recent independent work (Allen-Zhu et al., 2021) which utilizes history, but they have strong requirements on the noise (see Section 3 for why this might be an issue) and are not compatible with our problem setting. See Appendix G.3 for a more detailed comparison.

5. *Robust* robust aggregation

Past work on Byzantine robust methods have had wildly varying assumptions making an unified comparison difficult. Perhaps more importantly, this lead to unanticipated failures as we saw in Sec. 3. In this section, we attempt to provide a standardized specification for an robust aggregator which we believe captures a wide variety of real world behavior i.e. a robust aggregator which is robust to its assumptions. We then design a simple and efficient clipping based aggregator which satisfies this notion.

5.1. Anatomy of a robust aggregator

Suppose that we are given an aggregation rule $\text{AGG}(\cdots)$ and n vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. Among the given n vectors, let $\mathcal{G} \subseteq [n]$ be *good* (i.e. satisfy some closeness property), and the rest are Byzantine (and hence can be arbitrary). The ideal aggregator would return $\frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} \boldsymbol{x}_j$ but this requires exactly identifying the good workers, and hence may not be possible. We will instead be satisfied if our aggregation rule approximates the ideal update up to some error.

Our notion of a robust aggregator is characterized by two quantities: $\delta_{\max}$ which denotes the breakdown point, and a constant c which determines the quality of the solution. We want an aggregator which has as large $\delta_{\max}$ and a small c.

Definition C ($(\delta_{\max}, c)$-robust aggregator)**.** *Suppose that for some $\delta \leq \delta_{\max} \leq 0.5$ we are given n random vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ such that a good subset $\mathcal{G} \subseteq [n]$ of size at least $|\mathcal{G}| > (1-\delta)n$ are independent with distance bounded as*

$$\mathbb{E}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \leq \rho^2,$$

for any fixed $i, j \in \mathcal{G}$. Then, define $\bar{\boldsymbol{x}} := \frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} \boldsymbol{x}_j$. The, the robust aggregation rule $\text{AGG}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ *outputs $\hat{\boldsymbol{x}}$ such that,*

$$\mathbb{E}\|\hat{\boldsymbol{x}} - \bar{\boldsymbol{x}}\|^2 \leq c\delta\rho^2,$$

where the expectation is over the random variables $\{\boldsymbol{x}_i\}_{i\in[n]}$ and randomness in the aggregation rule AGG.

The error in Definition C is of the order $\delta\rho^2$. Thus, if $\delta = 0$ (no Byzantine workers), we recover the ideal average of the workers exactly. Further, we recover the exact average $\bar{\boldsymbol{x}}$ if $\rho = 0$ (no variance) since in this case all the good points are identical and are trivial to identify if they are in the majority ($\delta \leq \delta_{\max} \leq 0.5$). We demand that when the fraction of Byzantine workers is less than the breakdown point $\delta_{\max}$, the error of the output degrades gracefully with δ.

However, the error remains positive ($\delta\rho^2$) even with infinite n and seems to indicate that having additional workers

Algorithm 1 AGG - Centered Clipping

input: $(\boldsymbol{m}_1, \dots, \boldsymbol{m}_n)$, τ, $\boldsymbol{v}$, L
default: $L = 1$ and $\boldsymbol{v} = \hat{\boldsymbol{m}}$ (previous round aggreg.)
for each iteration $l = 1, \dots, L$ **do**
  $\boldsymbol{c}_i \leftarrow (\boldsymbol{m}_i - \boldsymbol{v}) \min\left(1, \frac{\tau}{\|\boldsymbol{m}_i - \boldsymbol{v}\|}\right)$
  $\boldsymbol{v} \leftarrow \boldsymbol{v} + \frac{1}{n} \sum_{i \in [n]} \boldsymbol{c}_i$
end for
output: $\boldsymbol{v}$

may not help. It turns out that this is unfortunately the price to pay for not knowing the good subset and is unavoidable. The following theorem is adapted from standard robust estimation lower bounds (e.g. see Lai et al. (2016)).

Theorem III (Limits of robustness)**.** *There exist a set of n random vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ such that a good subset $\mathcal{G} \subseteq [n]$ of size at least $|\mathcal{G}| \geq (1-\delta)n$ is i.i.d. satisfying $\mathbb{E}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \leq \rho^2$, for any apriori fixed $i, j \in \mathcal{G}$. For these vectors,* any *aggregation rule $\hat{\boldsymbol{x}} = \text{AGG}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$ necessarily has an error*

$$\mathbb{E}\|\hat{\boldsymbol{x}} - \boldsymbol{\mu}\|^2 \geq \delta\rho^2 .$$

Further, the error can be unbounded (∞) if $\delta \geq \frac{1}{2}$.

This establishes Definition C as the tightest notion of a robust aggregation oracle possible.

5.2. Robust aggregation via centered clipping

Given that most existing aggregation rules fail to satisfy Definition C, one may wonder if any such rule exists. We propose the following iterative *centered clipping* (CC) rule: starting from some point $\boldsymbol{v}_0$, for $l \geq 0$ compute

$$\boldsymbol{v}_{l+1} = \boldsymbol{v}_l + \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{v}_l) \min\left(1, \frac{\tau_l}{\|\boldsymbol{x}_i - \boldsymbol{v}_l\|}\right) \quad \text{(CC)}$$

Remark 4 (Ease of implementation)**.** *The centered clipping update is extremely simple to implement requiring $\mathcal{O}(n)$ computation and communication per step similar to coordinate-wise median. This is unlike more complicated mechanisms such as Krum or Bulyan which require $\mathcal{O}(n^2)$ computation and are hence less scalable. Further, as we will see later empirically, a single iteration of* CC *is often sufficient in practice. This means that the update can be implemented in an asynchronous manner (Chen et al., 2016), and is compatible with secure aggregation for federated learning (Bonawitz et al., 2017).*

We can formalize the convergence of this procedure.

Theorem IV (Robustness of centered clipping)**.** *Suppose that for $\delta \leq 0.15$ we are given n random vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ such that a good subset $\mathcal{G} \subseteq [n]$ of size at least $|\mathcal{G}| \geq (1-\delta)n$ are independent with bounded as $\mathbb{E}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \leq \rho^2$ for any fixed $i, j \in \mathcal{G}$. Then, running* CC *starting from any $\boldsymbol{v}_0$ for l steps with $\tau_l^2 = \mathcal{O}(\rho^2/\delta)$ satisfies*

$$\mathbb{E}\|\boldsymbol{v}_l - \bar{\boldsymbol{x}}\|^2 \leq (6.45\delta)^l 2\, \mathbb{E}\|\boldsymbol{v}_0 - \bar{\boldsymbol{x}}\|^2 + 1360\delta\rho^2 .$$

Proof Sketch. Suppose that we are given $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ with a subset of size at most δn are bad (denoted by $\mathcal{B}$), and the rest are good ($\mathcal{G}$). Consider the following simple scenario where $\|\boldsymbol{x}_i\|^2 \leq \rho^2$ almost surely for any $i \in \mathcal{G}$. In such a case, a very simple aggregation rule exists: clip all values to a radius ρ and then compute the average. All the good vectors remain unchanged. The magnitude of a clipped bad vector is at most ρ and since only a δ of the vectors are bad, they can move the center by at most $\rho\delta$ ensuring that our error is $\delta^2\rho^2$. This is even better than Definition C, which only requires the error to be smaller than $\delta\rho^2$. Of course there were two aspects which oversimplified our computations in the above discussion: i) we measure the pair-wise distance $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ between good workers instead of absolute norms, and ii) we do not have an almost sure bound, but only in expectation. □

Corollary V. *Starting from any $\boldsymbol{v}_0$ with an initial error estimate of $\mathbb{E}\|\boldsymbol{v}_0 - \bar{\boldsymbol{x}}\|^2 \leq B^2$, running* CC *for $l = 31 \log\left(2B^2/\delta\rho^2\right)$ is a $(\delta_{\max}, c)$-robust aggregator as per Definition C with $c = 1360$ and $\delta_{\max} = 0.15$.*
Further, if $\mathbb{E}\|\boldsymbol{v}_0 - \bar{\boldsymbol{x}}\|^2 \leq \rho^2$ then a single *step of* CC *is a $(\delta_{\max}, c)$-robust aggregator with the same values.*

The above corollary proves that starting from *any* point $\boldsymbol{v}_0$ and running enough iterations of CC is guaranteed to provide a robust estimate. However, if we have a good starting point, we can prove a much stronger statement—that a *single* clipping step is sufficient to provide robustness. We will use this latter part in designing an efficient robust optimization scheme in the next section.

Note that we have not tried to optimize for the constants in the theorem above—there is room for improvement in bringing $\delta_{\max}$ closer to 0.5, as well as in reducing the value of c. This may need a more careful analysis, or perhaps even a new oracle. We leave such improvements for future.

With this, we have addressed the first stumbling block and now have a robust aggregator. Next, we see how using momentum can defend against time-coupled attacks.

6. Robust optimization using momentum

In this section we will show that any Byzantine robust aggregator satisfying Definition C can be combined with (local) worker momentum, to obtain a Byzantine robust optimization algorithm which successfully defends against time coupled attacks. Every time step $t \geq 1$, the server sends the workers parameters $\boldsymbol{x}_{t-1}$ and each good worker

Algorithm 2 Robustness using Momentum

input: $\boldsymbol{x}$, η, β, AGG
initialize: $\boldsymbol{m}_i \leftarrow \mathbf{0}\ \forall i \in [n]$
for each round $t = 1, \dots$ **do**
  server communicates $\boldsymbol{x}$ to workers
  on worker $i \in \mathcal{G}$ **in parallel do**
    compute mini-batch gradient $\boldsymbol{g}_i(\boldsymbol{x})$
    compute $\boldsymbol{m}_i \leftarrow (1-\beta)\boldsymbol{g}_i(\boldsymbol{x}) + \beta \boldsymbol{m}_i$
    communicate $\boldsymbol{m}_i$ to server
  end on worker
  aggregate $\hat{\boldsymbol{m}} = \text{AGG}(\boldsymbol{m}_1, \dots, \boldsymbol{m}_n)$
  update $\boldsymbol{x} \leftarrow \boldsymbol{x} - \eta \hat{\boldsymbol{m}}$
end for

$i \in \mathcal{G}$ sends back $\boldsymbol{m}_{t,i}$ computed recursively as below starting from $\boldsymbol{m}_{0,i} = 0$

$$\boldsymbol{m}_{t,i} = (1-\beta_t)\boldsymbol{g}_i(\boldsymbol{x}_{t-1}) + \beta_t \boldsymbol{m}_{t-1,i}\,. \qquad \text{(WORKER)}$$

The workers communicate their momentum vector to the server instead of the stochastic gradients directly since they have a much smaller variance. Byzantine workers may send arbitrary vectors to the server. The server then uses a Byzantine-resilient aggregation rule AGG such as (CC) and computes the update

$$\begin{aligned} \boldsymbol{m}_t &= \text{AGG}(\boldsymbol{m}_{t,1}\,, \dots,\, \boldsymbol{m}_{t,n}) \\ \boldsymbol{x}_t &= \boldsymbol{x}_{t-1} - \eta_t \boldsymbol{m}_t\,. \end{aligned} \qquad \text{(SERVER)}$$

Intuitively, using momentum with $\beta = (1-\alpha)$ averages the stochastic gradients of the workers over their past $1/\alpha$ gradients. This results in a reduction of the variance of the good workers by a factor α since their noise is uncoupled. However, the variance of the time-coupled Byzantine perturbations does not reduce and becomes easy to detect.

6.1. Rate of convergence

Now we prove a rate of convergence of our Byzantine aggregation algorithm.

Theorem VI (Byzantine robust SGDm)**.** *Suppose that we are given a δ-robust problem satisfying Def. A and a $(\delta_{\max}, c)$-robust aggregation rule satisfying Def. C for $\delta_{\max} \geq \delta$. Then, running* WORKER *update with step-sizes $\eta_t = \min\Big(\sqrt{\frac{(f(\boldsymbol{x}_0)-f^\star)+\frac{5c\delta}{16L}\sigma^2}{20LT\sigma^2\left(\frac{2}{n}+c\delta\right)}}, \frac{1}{8L}\Big)$ and momentum parameter $\alpha_1 = 1$ and $\alpha_t = 8L\eta_{t-1}$ for $t \geq 2$ satisfies*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 \leq 16\sqrt{\frac{\sigma^2(1+c\delta n)}{nT}(10L(f(\boldsymbol{x}_0)-f^\star)+3c\delta\sigma^2)} + \frac{32L(f(\boldsymbol{x}_0)-f^\star)}{T} + \frac{20\sigma^2(1+c\delta n)}{nT}\,.$$

Remark 5 (Convergence rate)**.** *The rate of convergence in Theorem VI is asymptotically (ignoring constants and higher order terms) of the order:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 \lesssim \sqrt{\frac{\sigma^2}{T}\Big(\frac{1}{n}+\delta\Big)}\,.$$

First note that when $\delta = 0$ i.e. when there are no Byzantine adversaries, we recover the optimal rate of $\frac{\sigma}{\sqrt{nT}}$ which linearly scales with the number of workers n. In the presence of a δ fraction of adversaries, the rate has two terms: the first term $\frac{\sigma}{\sqrt{nT}}$ which linearly scales with the number of workers n, and a second $\frac{\sigma\sqrt{\delta}}{\sqrt{T}}$ which depends on the fraction of adversaries δ but does not improve with increasing workers. Similar phenomenon occurs in the classical robust mean estimation setting (Lai et al., 2016) and is unfortunately not possible to improve.

Our algorithm uses step-size η and momentum parameter $\alpha = (1-\beta)$ of the order of $\sqrt{\frac{1}{nT\sigma^2} + \frac{\delta}{T\sigma^2}}$. Here δ represents the fraction of adversarial workers. When there are very few bad workers with $\delta = \mathcal{O}(\frac{1}{n})$, the momentum and the step-size parameters can remain as in the non-Byzantine case. As the number of adversaries increases, δ increases meaning we should use smaller learning rate and larger momentum. Either when using linear scaling (Goyal et al., 2017) or square-root scaling (Hoffer et al., 2017), we need to scale both the learning-rate and momentum parameters as $\left(\frac{1}{n}+\delta\right)$ instead of the traditional $\frac{1}{n}$ in the presence of a δ fraction of adversaries.

The above algorithm and convergence analysis crucially relied on the low variance of the update from the workers using worker momentum. The very high momentum ensures that the variance of the updates from the workers to the server have a variance of the order $\sqrt{\frac{\sigma^2}{nT} + \frac{\delta\sigma^2}{T}}$. Note that this variance asymptotically goes to 0 with T and is significantly smaller than the variance of the stochastic gradient σ^2. This way, the Byzantine adversaries have very little lee-way to fool the aggregator.

6.2. Improved convergence using MVR

Recently, a variation of the standard momentum, called momentum based variance reduction or MVR, was proposed by Tran-Dinh et al. (2020); Cutkosky & Orabona (2019). They show that by adding a small correction to correct for bias, we can improve SGD's $\mathcal{O}(T^{-\frac{1}{2}})$ rate of convergence to $\mathcal{O}(T^{-\frac{2}{3}})$. By combining worker momentum based variance reduction with a Byzantine robust aggregator, we can obtain a faster Byzantine robust algorithm.

Theorem VII (Byzantine robust MVR)**.** *Suppose we are given a δ-robust Byzantine optimization problem Def. A.*

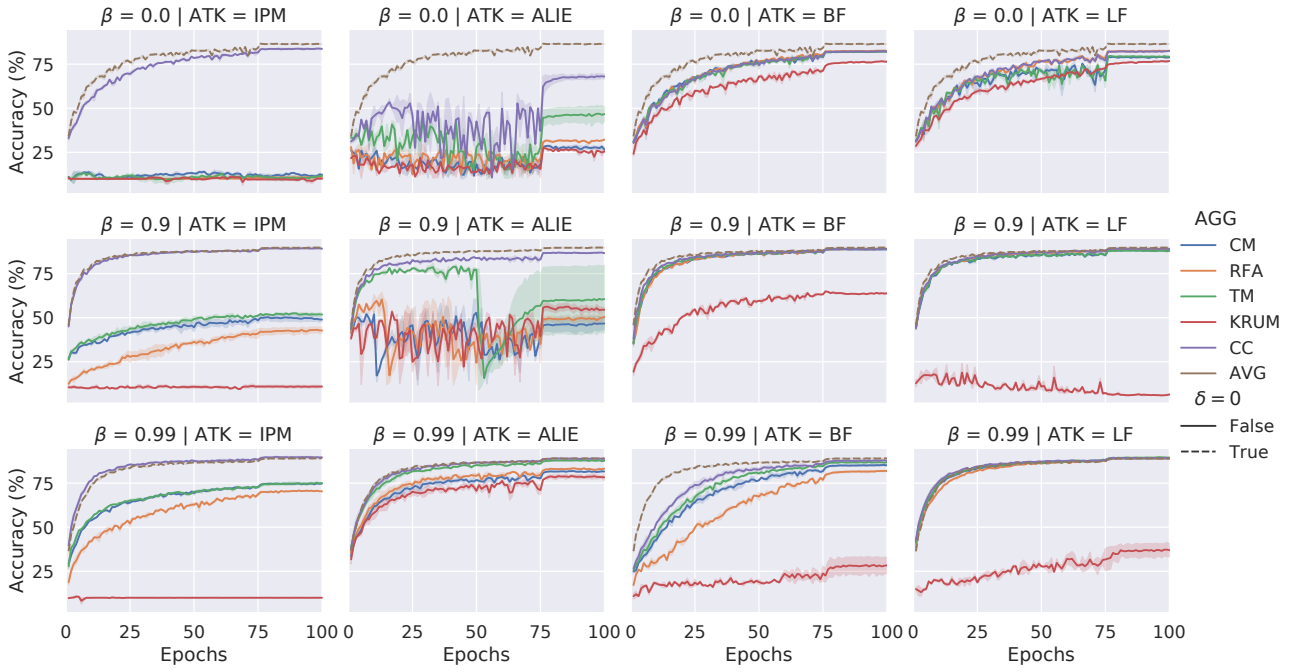

Figure 4: Coordinate median (CM), Robust Federated Aggregation (RFA), Trimmed Mean (TM), Krum, and Centered Clip (CC) are tested on Cifar10 with 25 workers. Attackers run inner-product manipulation attack (IPM) (Xie et al., 2020), "a little is enough" (ALIE) (Baruch et al., 2019), bit-flipping (BF), and label-flipping (LF). IPM uses 11 Byzantine workers while others use 5. The dashed brown line is average aggregator under no attacks ($\delta = 0$). Momentum generally improves all methods, with larger momentum adding stability. Centered Clip (CC) consistently has the best performance.

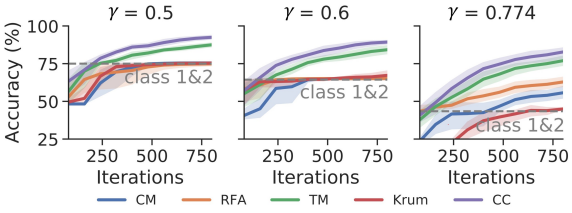

Figure 5: Robust aggregation rules on imbalanced MNIST where each successive class is a γ-fraction of the previous. Centered Clip is unaffected by imbalance where as the accuracy RFA, Krum, and CM corresponds to only learning class 1 and 2 (marked by horizontal gray dashed line).

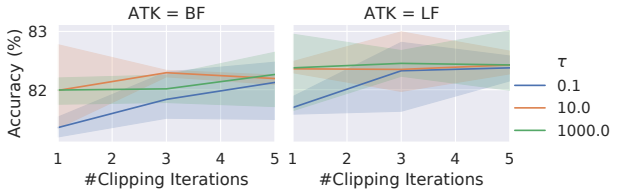

Figure 6: Final test accuracy of Centered Clip as we vary clipping iterations (l) and radius (τ). It is stable across all hyper-parameters, justifying using $l = 1$ as default.

Let us run the MVR algorithm combined with a $(\delta_{\max}, c)$*-robust aggregation rule* AGG *with* $\delta \leq \delta_{\max}$*, step-size* $\eta = \min \mathcal{O}\left(\sqrt[3]{\frac{f(\boldsymbol{x}_0)-f^\star}{T}}, \frac{1}{4L}\right)$*, and momentum parameter* $\alpha = \mathcal{O}(L^2\eta^2)$*. Then,*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 \lesssim \left(\frac{L\sigma\sqrt{c\delta+1/n}}{T}\right)^{2/3}.$$

Note that Theorem VII provides a significant asymptotic speedup over the traditional momentum used in Theorem VI and matches the lower bound of (Arjevani et al., 2019) when $\delta = 0$. This result highlights the versatility of our approach and the ease with which our notion of a Byzantine oracle can be combined with any state of the art optimization methods.

7. Experiments

In this section, we empirically demonstrate the effectiveness of CC and SGDM for Byzantine-robust learning. We refer to the baseline robust aggregation rules as RFA (Pillutla et al., 2019), coordinate-wise median (CM), trimmed mean (TM) (Yin et al., 2018), and Krum (Blanchard et al., 2017). The inner iteration (T) of RFA is fixed to 3 as suggested in (Pillutla et al., 2019). Throughout the section, we consider the distributed training for two image classification tasks, namely MNIST (LeCun & Cortes, 2010) on 16

nodes and CIFAR-10 (Krizhevsky et al., 2009) on 25 nodes. All experiments are repeated at least 2 times. The detailed setups are deferred to Appendix G.1.

7.1. Failure of "middle seekers"

In this experiment, we demonstrate the challenge stated in Section 3 by comparing robust aggregation rules on imbalanced datasets without attackers. Imbalanced training and test MNIST dataset are created by sampling classes with exponential decay, that is $1, \gamma, \gamma^2, \ldots, \gamma^{K-1}$ for classes 1 to K ($\gamma \in (0, 1]$). Then we shuffle the dataset and divide it equally into 16 nodes. The mini-batch for each node is 1.

The experimental results are presented in Fig. 5. For drastic decay $\gamma = 0.5$, the median and geometric median based rules can only achieve 75% accuracy which is the portion of class 1 and 2 in the data. This is a practical example of how "middle-seekers" fail. On the other hand, centered clip CC and trimmed mean have no such bound as they incorporate the gradients from tail distributions.

7.2. Impact of momentum on robust aggregation rules

The traditional implementation of momentum slightly differs from (WORKER) update and uses

$$\boldsymbol{m}_{t,i} = \boldsymbol{g}_i(\boldsymbol{x}_{t-1}) + \beta \boldsymbol{m}_{t-1,i}\,. \tag{1}$$

This version is equivalent to running (WORKER) update with a re-scaled learning rate of $\eta/(1-\beta)$. Further, note that our theory predicts that the clipping radius τ should be proportional to the variance of the updates which in turn depends on the momentum parameter β. We scale τ by a factor of $(1-\beta)$ if using (WORKER) update, and leave it constant if using update of the form (1).

In this experiment, we study the the influence of momentum on robust aggregation rules against various attacks, including bit-flipping (BF), label-flipping (LF), little is enough (Baruch et al., 2019), and inner product manipulation (Xie et al., 2020). We train ResNet-20 (He et al., 2016) on CIFAR-10 for 100 epochs on 25 workers where 5 of them are adversaries. For (Xie et al., 2020) we use 11 Byzantine workers to amplify the attack. The batch size per worker is set to 32 and the learning rate is 0.1 before 75th epoch and 0.01 afterwards. Note that the smaller batch size, e.g. 32, leads to larger variance among good gradients which makes the attacks in (Baruch et al., 2019; Xie et al., 2020) more challenging.

The results are presented in Fig. 4. Momentum generally makes the convergence faster and better for all aggregators, especially against SOTA attacks (Baruch et al., 2019; Xie et al., 2020). CC achieves best performance in almost all experiments. More specifically, it performs especially well on (Baruch et al., 2019; Xie et al., 2020) which is very close to training without attackers ($\delta = 0$).

7.3. Stability of Centered Clip

To demonstrate the impact of two hyperparameters τ, l of centered clip CC, we grid search τ in $[0.1, 10, 1000]$ and l in $[1, 3, 5]$. The setup is the same as in Sec. 7.2 and momentum is 0 to exclude its effect. The final accuracies are presented in Fig. 6. Centered clipping is very stable to the choice of hyperparameters, and can achieve good accuracy even without momentum.

8. Conclusion

The wildly disparate assumptions made in Byzantine robust learning not only makes comparison between different results impossible, but can also mask unexpected sources of failure. In this work, we strongly advocated for providing end to end convergence guarantees under realistic assumptions. We provided well-justified notions of a Byzantine robust aggregator and formalized the Byzantine robust stochastic optimization problem. Our theoretical lens led us to a surprisingly simple yet highly effective pair of strategies: using centered clipping and worker momentum. These strategies were thoroughly tested on a variety of attacks and shown to consistently outperform all baselines.

Acknowledgment. We thank Eduard Gorbunov and Dan Alistarh for comments on our earlier drafts. We are partly supported by a Google Focused Research Award.

References

Alistarh, D., Allen-Zhu, Z., and Li, J. Byzantine stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pp. 4613–4623, 2018.

Allen-Zhu, Z., Ebrahimian, F., Li, J., and Alistarh, D. Byzantine-resilient non-convex stochastic gradient descent. *ICLR*, 2021.

Arjevani, Y., Carmon, Y., Duchi, J. C., Foster, D. J., Srebro, N., and Woodworth, B. Lower bounds for non-convex stochastic optimization. *arXiv 1912.02365*, 2019.

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. How to backdoor federated learning. *arXiv 1807.00459*, 2019.

Baruch, G., Baruch, M., and Goldberg, Y. A little is enough: Circumventing defenses for distributed learning. In *Advances in Neural Information Processing Systems*, pp. 8635–8645, 2019.

Bernstein, J., Zhao, J., Azizzadenesheli, K., and Anandkumar, A. signSGD with majority vote is communication efficient and fault tolerant. *arXiv 1810.05291*, 2018.