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),

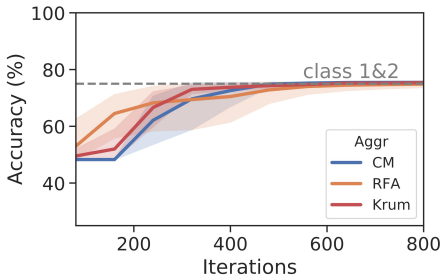

Figure 1: Failure of existing methods on imbalanced MNIST dataset. Only the head classes (class 1 and 2 here) are learnt, and the rest 8 classes are ignored. See Sec. 7.1.

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,

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

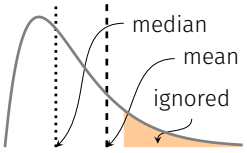

Figure 2: For fat-tailed distributions, median based aggregators ignore the tail. This bias remains even if we have infinite samples.

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 $\delta_{\max}$.

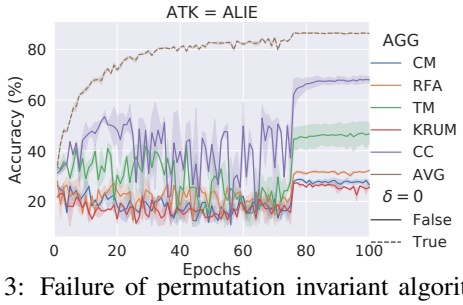

Figure 3: Failure of permutation invariant algorithms on CIFAR10 dataset with (Baruch et al., 2019) attack. Comparing to simple average with no attacker (dashed lines), all robust aggregators (including centered clip) see a significant drop in accuracy against time coupled attacks. See Sec. 7.2.

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.*

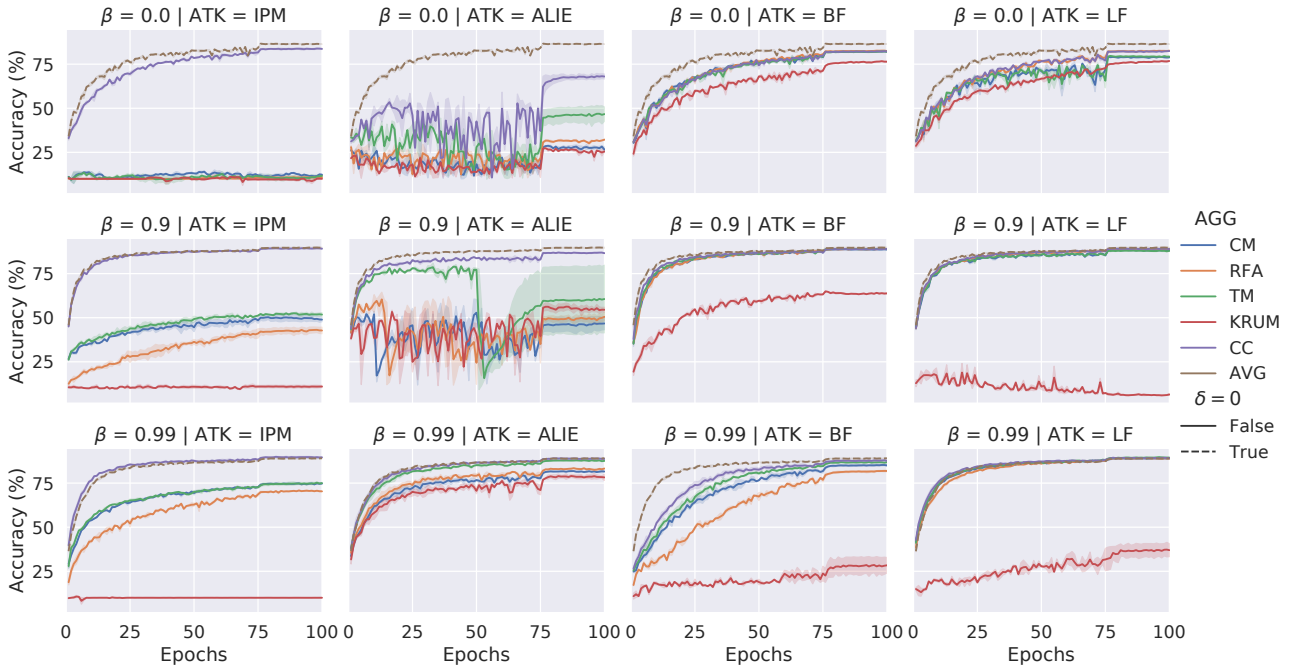

Figure 4: Coordinate median (CM), Robust Federated Aggregation (RFA), Trimmed Mean (TM), Krum, and Centered Clip (CC) are tested on Cifar10 with 25 workers. Attackers run inner-product manipulation attack (IPM) (Xie et al., 2020), "a little is enough" (ALIE) (Baruch et al., 2019), bit-flipping (BF), 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.

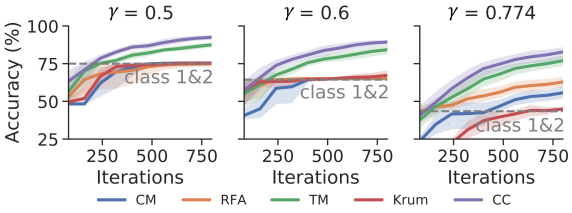

Figure 5: Robust aggregation rules on imbalanced MNIST where each successive class is a γ-fraction of the previous. Centered Clip is unaffected by imbalance where as the accuracy RFA, Krum, and CM corresponds to only learning class 1 and 2 (marked by horizontal gray dashed line).

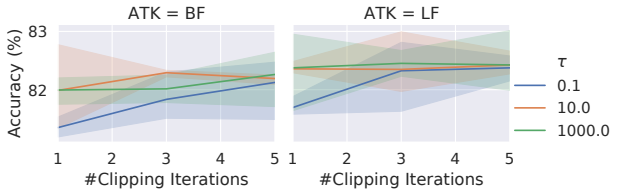

Figure 6: Final test accuracy of Centered Clip as we vary clipping iterations (l) and radius (τ). It is stable across all hyper-parameters, justifying using $l = 1$ as default.

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.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *Advances in Neural Information Processing Systems 30*, pp. 119–129, 2017.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191. ACM, 2017.

Chen, J., Pan, X., Monga, R., Bengio, S., and Jozefowicz, R. Revisiting distributed synchronous sgd. *arXiv preprint arXiv:1604.00981*, 2016.

Chen, L., Wang, H., Charles, Z., and Papailiopoulos, D. Draco: Byzantine-resilient distributed training via redundant gradients. *arXiv 1803.09877*, 2018.

Chen, X., Chen, T., Sun, H., Wu, Z. S., and Hong, M. Distributed training with heterogeneous data: Bridging median- and mean-based algorithms. *arXiv 1906.01736*, 2019.

Chen, Y., Su, L., and Xu, J. Distributed statistical machine learning in adversarial settings. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1 (2):1–25, Dec 2017. ISSN 2476-1249. doi: 10.1145/3154503. URL http://dx.doi.org/10.1145/3154503.

Cutkosky, A. and Orabona, F. Momentum-based variance reduction in non-convex sgd. In *Advances in Neural Information Processing Systems*, pp. 15236–15245, 2019.

Data, D. and Diggavi, S. Byzantine-resilient sgd in high dimensions on heterogeneous data. *arXiv 2005.07866*, 2020.

Data, D., Song, L., and Diggavi, S. Data encoding for byzantine-resilient distributed gradient descent. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 863–870. IEEE, 2018.

Data, D., Song, L., and Diggavi, S. Data encoding for byzantine-resilient distributed optimization. *ISIT - International Symposium on Information Theory*, 2019.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Steinhardt, J., and Stewart, A. Sever: A robust meta-algorithm for stochastic optimization. *arXiv 1803.02815*, 2018.

Dong, Y., Giannakis, G. B., Chen, T., Cheng, J., Hossain, M. J., and Leung, V. C. M. Communication-efficient robust federated learning over heterogeneous datasets. *arXiv 2006.09992*, 2020.

El-Mhamdi, E.-M. and Guerraoui, R. Fast and secure distributed learning in high dimension. *arXiv 1905.04374*, 2019.

El-Mhamdi, E.-M., Guerraoui, R., Guirguis, A., Hoang, L. N., and Rouault, S. Collaborative learning as an agreement problem. *arXiv 2008.00742*, 2020.

El-Mhamdi, E.-M., Guerraoui, R., and Rouault, S. Distributed momentum for byzantine-resilient learning. *ICLR*, 2021.

Feng, J., Xu, H., and Mannor, S. Distributed robust learning. *arXiv preprint arXiv:1409.5937*, 2014.

Fu, S., Xie, C., Li, B., and Chen, Q. Attack-resistant federated learning with residual-based reweighting. *arXiv 1912.11464*, 2019.

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

Gorbunov, E., Danilova, M., and Gasnikov, A. Stochastic optimization with heavy-tailed noise via accelerated gradient clipping. *NeurIPS - Advances in Neural Information Processing Systems*, 2020.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv 1706.02677*, 2017.

Gupta, N. and Vaidya, N. H. Randomized reactive redundancy for byzantine fault-tolerance in parallelized learning. *arXiv 1912.09528*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, L., Karimireddy, S. P., and Jaggi, M. Byzantine-robust learning on heterogeneous datasets via resampling. *arXiv 2006.09365*, 2020a.

He, L., Karimireddy, S. P., and Jaggi, M. Secure byzantine-robust machine learning. *arXiv 2006.04747*, 2020b.

Hoffer, E., Hubara, I., and Soudry, D. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems*, pp. 1731–1741, 2017.

Hubert, M., Rousseeuw, P. J., and Van Aelst, S. High-breakdown robust multivariate methods. *Statistical science*, pp. 92–119, 2008.

Jin, R., Huang, Y., He, X., Wu, T., and Dai, H. Stochastic-sign sgd for federated learning with theoretical guarantees. *arXiv 2002.10940*, 2020.

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., 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., Raykova, M., Qi, H., Ramage, D., Raskar, R., 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. Advances and open problems in federated learning. *arXiv 1912.04977*, 2019.

Karimireddy, S. P., Jaggi, M., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. Mime: Mimicking centralized stochastic algorithms in federated learning. *arXiv preprint arXiv:2008.03606*, 2020a.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020b.

Konstantinidis, K. and Ramamoorthy, A. Byzshield: An efficient and robust system for distributed training. *arXiv 2010.04902*, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lai, K. A., Rao, A. B., and Vempala, S. Agnostic estimation of mean and covariance. *arXiv 1604.06968*, 2016.

Lamport, L., Shostak, R., and Pease, M. The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pp. 203–226. 2019.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Li, L., Xu, W., Chen, T., Giannakis, G. B., and Ling, Q. Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1544–1551, 2019.

Liu, Y., Gao, Y., and Yin, W. An improved analysis of stochastic gradient descent with momentum. *arXiv 2007.07989*, 2020.

Mhamdi, E. M. E., Guerraoui, R., and Rouault, S. The hidden vulnerability of distributed learning in byzantium. *arXiv 1802.07927*, 2018.

Minsker, S. et al. Geometric median and robust estimation in banach spaces. *Bernoulli*, 21(4):2308–2335, 2015.

Peng, J. and Ling, Q. Byzantine-robust decentralized stochastic optimization. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5935–5939, 2020.

Peng, J., Wu, Z., and Ling, Q. Byzantine-robust variance-reduced federated learning over distributed non-i.i.d. data. *arXiv 2009.08161*, 2020.

Pillutla, K., Kakade, S. M., and Harchaoui, Z. Robust Aggregation for Federated Learning. *arXiv 1912.13445*, 2019.

Rajput, S., Wang, H., Charles, Z., and Papailiopoulos, D. Detox: A redundancy-based framework for faster and more robust gradient aggregation. *arXiv 1907.12205*, 2019.

Regatti, J. and Gupta, A. Befriending the byzantines through reputation scores. *arXiv 2006.13421*, 2020.

Rodríguez-Barroso, N., Martínez-Cámara, E., Luzón, M. V., Seco, G. G., Ángel Veganzones, M., and Herrera, F. Dynamic federated learning model for identifying adversarial clients. *arXiv 2007.15030*, 2020.

Shallue, C. J., Lee, J., Antognini, J., Sohl-Dickstein, J., Frostig, R., and Dahl, G. E. Measuring the effects of data parallelism on neural network training. *arXiv 1811.03600*, 2018.

So, J., Guler, B., and Avestimehr, A. S. Byzantine-resilient secure federated learning. *arXiv 2007.11115*, 2020a.

So, J., Guler, B., and Avestimehr, A. S. Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning. *arXiv 2002.04156*, 2020b.

Su, L. and Xu, J. Securing distributed gradient descent in high dimensional statistical learning. *arXiv 1804.10140*, 2018.

Sun, Z., Kairouz, P., Suresh, A. T., and McMahan, H. B. Can you really backdoor federated learning? *arXiv 1911.07963*, 2019.

Tran-Dinh, Q., Liu, D., and Nguyen, L. M. Hybrid variance-reduced sgd algorithms for nonconvex-concave minimax problems. *arXiv preprint arXiv:2006.15266*, 2020.

Wang, H., Sreenivasan, K., Rajput, S., Vishwakarma, H., Agarwal, S., Sohn, J.-y., Lee, K., and Papailiopoulos, D. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Xie, C., Koyejo, O., and Gupta, I. Fall of Empires: Breaking Byzantine-tolerant SGD by Inner Product Manipulation. In *UAI - Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, 2020.

Yin, D., Chen, Y., Ramchandran, K., and Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. *arXiv 1803.01498*, 2018.

Yu, H., Jin, R., and Yang, S. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. *arXiv 1905.03817*, 2019.

Zhang, J., Karimireddy, S. P., Veit, A., Kim, S., Reddi, S. J., Kumar, S., and Sra, S. Why adam beats sgd for attention models. *arXiv 1912.03194*, 2019.

Appendix

A. Convergence of momentum SGD

Here we describe the convergence proof of the naive SGD with momentum algorithm. Starting from a given $\boldsymbol{x}_0$ and with $\boldsymbol{m}_0 = 0$, we run the following updates with a sequence of momentum parameters $\alpha_t \in [0,1]$ and step-sizes $\eta_t \geq 0$

$$\begin{aligned} \boldsymbol{m}_t &= \alpha_t \boldsymbol{g}(\boldsymbol{x}_{t-1}) + (1-\alpha_t)\boldsymbol{m}_{t-1} \\ \boldsymbol{x}_t &= \boldsymbol{x}_{t-1} - \eta_t \boldsymbol{m}_t \,. \end{aligned} \tag{SGDM}$$

While there exist numerous previous analyses of SGD with momentum for smooth non-convex objectives, most of them rely on viewing the SGDm method as an approximation of an underlying SGD without momentum algorithm—see Yu et al. (2019); Liu et al. (2020) for recent examples of this viewpoint. Because they view momentum as approximating an SGD process, the rates proved are necessarily slower for momentum and further they can only handle constant values of α (i.e. the momentum parameter cannot decrease with T). In this work, we take an alternate viewpoint to momentum inspired by (Cutkosky & Orabona, 2019; Karimireddy et al., 2020a). We view the momentum update as a way to reduce the variance i.e. by using an exponential averaging over many independent stochastic gradients we get an estimate of the true full gradient which has much lesser variance (though higher bias). This way, our method can handle momentum parameter which is almost 1 ($\alpha \approx \frac{1}{\sigma\sqrt{T}}$). Thus the resulting update has very low variance which will later be crucial for deriving optimal robust methods.

Theorem VIII (Convergence of SGDm)**.** *The* SGDM *algorithm with step-size* $\eta_t = \min\{\frac{1}{4\sigma}\sqrt{\frac{f(\boldsymbol{x}_0)-f^\star}{LT}}, \frac{1}{4L}\}$ *and momentum parameter* $\alpha_1 = 1$ *and* $\alpha_t = 4L\eta_{t-1}$ *for* $t \geq 2$ *satisfies*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 \leq 80\cdot\sigma\sqrt{\frac{L(f(\boldsymbol{x}_0)-f^\star)}{T}} + \frac{4L(f(\boldsymbol{x}_0)-f^\star)}{T}$$

First, note that the rate for momentum algorithm is of the order $\frac{\sigma}{\sqrt{T}}$ which matches the optimal rate of SGD for smooth non-convex functions (Arjevani et al., 2019). Further, this rate is achieved using very high momentum with both α (and step-sizes) of the order $\frac{1}{\sigma\sqrt{T}}$. Also, when $\sigma = 0$ i.e. in the deterministic gradient case, we recover the optimal $\frac{1}{T}$ rate (but with a constant step-size and momentum). This is intuitive since we do not need to reduce the variance in the deterministic case and so large momentum is unnecessary.

Remark 6 (Large batch generalization)**.** *There is some empirical evidence that momentum is also useful when using extremely large batch sizes (i.e. nearly deterministic gradient) and helps in closing the generalization gap (Shallue et al., 2018). In contrast, current theory claims that gradient descent (without momentum) is already optimal for non-convex optimization (Arjevani et al., 2019). We believe these differences occur because even if using large batches, there remains stochasticity in the gradient due to data-augmentation. Thus* $\sigma > 0$ *in practice even when using full batches.*

We first prove some supporting lemmas before proving Theorem VIII.

Lemma 7. *For* $\alpha_1 = 1$ *and any* $\alpha_t \in [0,1]$ *for* $t \geq 2$*, and an* L*-smooth function* f *we have that* $\mathbb{E}_1[f(\boldsymbol{x}_1)] \leq f(\boldsymbol{x}_0) - \frac{\eta_1}{2}\|\nabla f(\boldsymbol{x}_0)\|^2 + \frac{\eta_1}{2}\sigma^2 - \frac{\eta_1}{2}(1-L\eta_1)\|\boldsymbol{m}_1\|^2$ *and for* $t \geq 2$

$$\mathbb{E}_t[f(\boldsymbol{x}_t)] \leq f(\boldsymbol{x}_{t-1}) + \frac{\eta_t}{2}\|\boldsymbol{m}_t - \nabla f(\boldsymbol{x}_{t-1})\|^2 - \frac{\eta_t}{2}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 - \frac{\eta_t}{2}(1-L\eta_t)\|\boldsymbol{m}_t\|^2 \,.$$

Proof. By the smoothness of the function f and the SGDm update,

$$\begin{aligned} f(\boldsymbol{x}_t) &\leq f(\boldsymbol{x}_{t-1}) - \eta_t\langle\nabla f(\boldsymbol{x}_{t-1}), \boldsymbol{m}_t\rangle + \frac{L\eta_t^2}{2}\|\boldsymbol{m}_t\|^2 \\ &= f(\boldsymbol{x}_{t-1}) + \frac{\eta_t}{2}\|\boldsymbol{m}_t - \nabla f(\boldsymbol{x}_{t-1})\|^2 - \frac{\eta_t}{2}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 - \frac{\eta_t}{2}(1-L\eta_t)\|\boldsymbol{m}_t\|^2 \,. \end{aligned}$$

Taking conditional expectation on both sides yields the second part of the lemma. The first part follows from standard descent analysis of SGD. □

Lemma 8. *Define $\boldsymbol{e}_t := \boldsymbol{m}_t - \nabla f(\boldsymbol{x}_{t-1})$. Then, using any momentum and step-sizes such that $1 \geq \alpha_t \geq 4L\eta_{t-1}$ for $t \geq 2$, we have for an L-smooth function f that $\mathbb{E}\|\boldsymbol{e}_1\|^2 \leq \alpha_1\sigma^2$ and for $t \geq 2$*

$$\mathbb{E}\|\boldsymbol{e}_t\|^2 \leq (1 - \tfrac{\alpha_t}{2})\,\mathbb{E}\|\boldsymbol{e}_{t-1}\|^2 + L^2\eta_{t-1}^2(1-\alpha_t)(1+\tfrac{2}{\alpha_t})\,\mathbb{E}\|\boldsymbol{m}_{t-1}\|^2 + \alpha_t^2\sigma^2\,.$$

Proof. Starting from the definition of $\boldsymbol{e}_t$ and $\boldsymbol{m}_t$,

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{e}_t\|^2 &= \mathbb{E}\|\boldsymbol{m}_t - \nabla f(\boldsymbol{x}_{t-1})\|^2 \\
&= \mathbb{E}\|\alpha_t\boldsymbol{g}(\boldsymbol{x}_{t-1}) + (1-\alpha_t)\boldsymbol{m}_{t-1} - \nabla f(\boldsymbol{x}_{t-1})\|^2 \\
&\leq (1-\alpha_t)^2\,\mathbb{E}\|\boldsymbol{m}_{t-1} - \nabla f(\boldsymbol{x}_{t-1})\|^2 + \alpha_t^2\sigma^2 \\
&= (1-\alpha_t)^2\,\mathbb{E}\|(\boldsymbol{m}_{t-1} - \nabla f(\boldsymbol{x}_{t-2})) + (\nabla f(\boldsymbol{x}_{t-2}) - \nabla f(\boldsymbol{x}_{t-1}))\|^2 + \alpha_t^2\sigma^2 \\
&\leq (1-\alpha_t)(1+\tfrac{\alpha_t}{2})\,\mathbb{E}\|\boldsymbol{m}_{t-1} - \nabla f(\boldsymbol{x}_{t-2})\|^2 + (1-\alpha_t)(1+\tfrac{2}{\alpha_t})\,\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-2}) - \nabla f(\boldsymbol{x}_{t-1})\|^2 + \alpha_t^2\sigma^2 \\
&\leq (1-\tfrac{\alpha_t}{2})\,\mathbb{E}\|\boldsymbol{e}_{t-1}\|^2 + L^2(1-\alpha_t)(1+\tfrac{2}{\alpha_t})\,\mathbb{E}\|\boldsymbol{x}_{t-2} - \boldsymbol{x}_{t-1}\|^2 + \alpha_t^2\sigma^2 \\
&\leq (1-\tfrac{\alpha_t}{2})\,\mathbb{E}\|\boldsymbol{e}_{t-1}\|^2 + L^2\eta_{t-1}^2(1-\alpha_t)(1+\tfrac{2}{\alpha_t})\,\mathbb{E}\|\boldsymbol{m}_{t-1}\|^2 + \alpha_t^2\sigma^2\,.
\end{aligned}$$

Here the first inequality used the fact that $\boldsymbol{g}(\boldsymbol{x}_{t-1})$ is an unbiased and independent stochastic gradient with variance bounded by σ^2. The second inequality follows from Fano's inequality i.e. $\|\boldsymbol{x}+\boldsymbol{y}\|^2 \leq (1+a)\|\boldsymbol{x}\|^2 + (1+\frac{1}{a})\|\boldsymbol{y}\|^2$ for any $a \geq 0$. □

We are now ready to prove the convergence theorem.

Proof of Theorem VIII. Scaling Lemma 7 by L and adding it to Lemma 8 we have for any $t \geq 2$

$$\begin{aligned}
\mathbb{E}\;Lf(\boldsymbol{x}_t) + \mathbb{E}\|\boldsymbol{e}_t\|^2 \leq \mathbb{E}\;Lf(\boldsymbol{x}_{t-1}) &+ \frac{L\eta_t}{2}\,\mathbb{E}\|\boldsymbol{e}_t\|^2 - \frac{L\eta_t}{2}\,\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 - \frac{L\eta_t}{2}(1-L\eta_t)\|\boldsymbol{m}_t\|^2 \\
&+ (1-\tfrac{\alpha_t}{2})\,\mathbb{E}\|\boldsymbol{e}_{t-1}\|^2 + L^2\eta_{t-1}^2(1-\alpha_t)(1+\tfrac{2}{\alpha_t})\,\mathbb{E}\|\boldsymbol{m}_{t-1}\|^2 + \alpha_t^2\sigma^2\,.
\end{aligned}$$

By taking $\eta_t = \eta_{t-1} = \eta$ and $1 \geq \alpha_t \geq 4L\eta$

$$\begin{aligned}
&\underbrace{\mathbb{E}\;L(f(\boldsymbol{x}_t) - f^\star) + \left(1 - \tfrac{L\eta_t}{2}\right)\mathbb{E}\|\boldsymbol{e}_t\|^2 + \tfrac{L\eta_t}{2}(1-L\eta_t)\|\boldsymbol{m}_t\|^2}_{=:\xi_t} + \tfrac{L\eta_t}{2}\,\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 \\
&\leq \mathbb{E}\;L(f(\boldsymbol{x}_{t-1}) - f^\star) + (1-\tfrac{\alpha_t}{2})\,\mathbb{E}\|\boldsymbol{e}_{t-1}\|^2 + L^2\eta_{t-1}^2(1-\alpha_t)(1+\tfrac{2}{\alpha_t})\,\mathbb{E}\|\boldsymbol{m}_{t-1}\|^2 + \alpha_t^2\sigma^2\,. \\
&\leq \underbrace{\mathbb{E}\;L(f(\boldsymbol{x}_{t-1}) - f^\star) + \left(1 - \tfrac{L\eta_{t-1}}{2}\right)\mathbb{E}\|\boldsymbol{e}_{t-1}\|^2 + \tfrac{L\eta_{t-1}}{2}(1-L\eta_{t-1})\,\mathbb{E}\|\boldsymbol{m}_{t-1}\|^2}_{=:\xi_{t-1}} + \alpha_t^2\sigma^2\,.
\end{aligned}$$

Note that from the first parts of Lemma 7 and Lemma 8, we have

$$\begin{aligned}
\xi_1 &\leq \mathbb{E}\;L(f(\boldsymbol{x}_1) - f^\star) + \left(1 - \tfrac{L\eta_1}{2}\right)\mathbb{E}\|\boldsymbol{e}_1\|^2 + \tfrac{L\eta_1}{2}(1-L\eta_1)\|\boldsymbol{m}_1\|^2 \\
&\leq L(f(\boldsymbol{x}_0) - f^\star) + \sigma^2 - \tfrac{L\eta_1}{2}\,\mathbb{E}\|\nabla f(\boldsymbol{x}_0)\|^2\,.
\end{aligned}$$

Summing over t and again rearranging gives

$$\sum_{t=1}^{\ell} L\eta_t\,\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 \leq L(f(\boldsymbol{x}_0) - f^\star) + \sum_{t=1}^{\ell}\alpha_t^2\sigma^2\,.$$

By taking $\eta_t = \eta_{t-1} = \eta$ and $\alpha_t = 4L\eta$, this simplifies the above inequality to

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 \leq \frac{f(\boldsymbol{x}_0) - f^\star}{\eta T} + 16L\eta\sigma^2\,.$$

By taking $\eta = \min\{\frac{1}{4\sigma}\sqrt{\frac{f(\boldsymbol{x}_0)-f^\star}{LT}}, \frac{1}{4L}\}$ we prove the theorem. □

B. Proof of Theorem II - Failure of permutation-invariant methods

Our proof builds two instances of a δ-robust optimization problem satisfying Definition A and shows that they are indistinguishable, meaning that we make a mistake on at least one of them.

For the first problem, set $f^{(1)}(x) = \frac{\mu}{2}x^2 - Gx$ with optimum at $x^\star = \frac{G}{\mu}$ for some G to be defined later. It has a gradient $\nabla f^{(1)}(x) = \mu x - G$ and we set the stochastic gradient for some $\tilde{\delta} \in [0, 1]$ to be defined later as

$$g^{(1)}(x) = \begin{cases} \mu x - \sigma\tilde{\delta}^{-1/2} & \text{with prob. } \tilde{\delta} \\ \mu x & \text{o.w.} \end{cases}$$

Defining $G := \sigma\tilde{\delta}^{1/2}$, we have that $g^{(1)}(x)$ is an unbiased stochastic gradient. Further, its variance is bounded by σ^2 since $\mathbb{E}[(g^{(1)}(x) - \nabla f^{(1)}(x))^2] \le \sigma^2$. In each round t, let each worker $i \in [n]$ draw an i.i.d. sample from the distribution $g^{(1)}(x)$ as their stochastic gradient. Define $C_t \in [n]$ to be the number of workers whose stochastic gradients is the first setting i.e.

$$C_t = \#\left\{i \in [n] \text{ s.t. } g_i^{(1)}(x_t) = \mu x_t - \sigma\tilde{\delta}^{-1/2}\right\}.$$

Now we define the second problem. Let $f_2(x) = \frac{\mu}{2}x^2$ with optimum at $x^\star = 0$. Define its stochastic gradient to always be $g^{(2)}(x) = \mu x$. Now, in round t each worker $i \in [n]$ computes $g_i^{(2)}(x_t) = x_t$. Then, $\min(n\delta, C_t)$ Byzantine workers corrupt their gradients to instead be $g_j^{(2)}(x_t) = \mu x_t - \sigma\tilde{\delta}^{-1/2}$.

Note that C_t is the sum n independent Bernoulli trials with parameter $\tilde{\delta}$. Thus, we have via Chernoff's bound that for any $\gamma \ge 2$,

$$\Pr[C_t > (1+\gamma)n\tilde{\delta}] \le \exp\left(-\frac{\gamma n\tilde{\delta}}{2}\right).$$

By picking $\gamma = \max(2, 2(1+\log(T))/(n\tilde{\delta}))$, we have that $\Pr[C_t > (1+\gamma)n\tilde{\delta}] \le \frac{1}{2T}$. By setting $\tilde{\delta} = \delta/6$ and assuming that n is large enough such that $4(1+\log T) \le \delta n$, we can simplify $(1+\gamma)n\tilde{\delta} \ge \delta n$. Taking an union bound over all values of t, we have that

$$\Pr\left[C_t \le n\delta \text{ for all } t \in [T]\right] \ge \frac{1}{2}.$$

Thus, with probability at least 0.5, we have that the stochastic gradients in problem 1 are exactly the same (up to permutation) to problem 2. This implies that with probability 0.5, no permutation-invariant algorithm can distinguish between the two settings, implying that we necessarily incur an error of the order of the difference between their minima

$$\mu\left(\frac{G}{\mu}\right)^2 = \frac{\sigma^2\tilde{\delta}}{\mu} = \frac{\sigma^2\delta}{6\mu}.$$

□

C. Proof of Theorem III (Limits of robust aggregation)

It is easy to establish the second result since if $\delta \ge \frac{1}{2}$, it is impossible to decide which of the subsets is good. E.g. if half of the inputs are a and the other are b, even if we know that $\rho = 0$, the good workers might correspond to either the a half or the b half equally likely. Assuming $\delta \le \frac{1}{2}$, define the following binomial distribution:

$$\mathcal{P} := \begin{cases} \rho\delta^{-1/2} & \text{with prob. } \delta/2 \\ 0 & \text{o.w.} \end{cases}$$

Suppose that each x_i for all $i \in [n]$ is an iid sample drawn from $\mathcal{P}$. Clearly we have that $\mathbb{E}(x_i - x_j)^2 \le \rho^2$. Define $B_n \in [n]$ to be the number of samples which are equal to $\rho\delta^{-1/2}$ (with the rest being 0). Now consider a second scenario for $\{x_i\}$: the adversary sets $\min(\delta n, B_n)$ of the variables to $\rho\delta^{-1/2}$ and the rest of the good variables are 0.

Note that $\mathbb{E}[B_n] = n\delta/2$ and so by Markov's inequality we have that $\Pr[B_n \le n\delta] \ge \frac{1}{2}$. So with at least probability $1/2$, the two cases are impossible to distinguish. However in the first case, all samples are good whereas in the second case

only the 0 samples are good. Hence, any output will necessarily have an error of the order of the difference between their respective $\bar{\boldsymbol{x}}$s:

$$(\mathbb{E}_{x\sim\mathcal{P}}[x]-0)^2=(\rho\delta^{1/2}/2)^2=\frac{\delta\rho^2}{4}\,.$$

D. Proof of Theorem IV- Robustness of iterative clipping

Our proof strategy is inspired by (Zhang et al., 2019; Gorbunov et al., 2020) who study the bias of clipped stochastic gradients in the context of heavy-tailed noise. First, suppose that $\delta=0$. In this case, we chose the clipping radius $\tau_l=\tilde{\mathcal{O}}(\rho/\sqrt{\delta})=\infty$ and our algorithm simply averages all points. Hence, we recover $\bar{\boldsymbol{x}}$ exactly with no error as required. Now if $\delta>0$, this means that at least one of the n workers is Byzantine and hence $\delta\geq 1/n$. We consider this case in the rest of the proof.

Recall that $\bar{\boldsymbol{x}}=\frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}}\boldsymbol{x}_i$ and let us define $\boldsymbol{\mu}=\mathbb{E}[\bar{\boldsymbol{x}}]$ where the expectation is over the random vectors. Now since the good random are independent, we have

$$\mathbb{E}\|\bar{\boldsymbol{x}}-\boldsymbol{\mu}\|^2\leq\frac{\rho^2}{|\mathcal{G}|}\leq\frac{2\rho^2}{n}\leq 2\delta\rho^2\,.$$

At some iteration l, suppose that $B_l^2\geq\mathbb{E}\|\boldsymbol{v}_l-\boldsymbol{\mu}\|^2$ is an upper bound on the current error with $B_0:=B$. Define indicator variables $\mathbb{1}_{i,l}:=\mathbb{1}\{\|\boldsymbol{v}_l-\boldsymbol{x}_i\|\geq\tau_l\}$ which define the event that the vector $\boldsymbol{x}_i$ is clipped, as well the resulting clipped vector

$$\boldsymbol{y}_{i,l}:=\boldsymbol{v}_l+(\boldsymbol{x}_i-\boldsymbol{v}_l)\min\left(1,\frac{\tau_l}{\|\boldsymbol{x}_i-\boldsymbol{v}_l\|}\right).$$

The output can also be written in this new notation as

$$\boldsymbol{v}_{l+1}=\frac{1}{n}\sum_{i\in[n]}\boldsymbol{y}_{i,l}=(1-\delta)\frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}}\boldsymbol{y}_{i,l}+\delta\frac{1}{|\mathcal{B}|}\sum_{j\in\mathcal{B}}\boldsymbol{y}_{j,l}\,.$$

Then the error can be decomposed as follows

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{v}_{l+1}-\boldsymbol{\mu}\|^2&=\mathbb{E}\left\|(1-\delta)\frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}}\boldsymbol{y}_{i,l}+\delta\frac{1}{|\mathcal{B}|}\sum_{j\in\mathcal{B}}\boldsymbol{y}_{j,l}-\frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}}\mathbb{E}[\boldsymbol{x}_i]\right\|^2\\
&=\mathbb{E}\left\|(1-\delta)\frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}}(\boldsymbol{y}_{i,l}-\mathbb{E}[\boldsymbol{x}_i])+\delta\frac{1}{|\mathcal{B}|}\sum_{j\in\mathcal{B}}(\boldsymbol{y}_{j,l}-\boldsymbol{\mu})\right\|^2\\
&\leq 2(1-\delta)^2\,\mathbb{E}\left\|\frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}}\boldsymbol{y}_{i,l}-\mathbb{E}[\boldsymbol{x}_i]\right\|^2+2\delta^2\frac{1}{|\mathcal{B}|}\sum_{j\in\mathcal{B}}\mathbb{E}\|\boldsymbol{y}_{j,l}-\boldsymbol{\mu}\|^2\\
&=\underbrace{2(1-\delta)^2\left\|\frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}}\mathbb{E}[\boldsymbol{y}_{i,l}]-\mathbb{E}[\boldsymbol{x}_i]\right\|^2}_{\mathcal{T}_1}+\underbrace{2(1-\delta)^2\,\mathbb{E}\left\|\frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}}\boldsymbol{y}_{i,l}-\mathbb{E}[\boldsymbol{y}_{i,l}]\right\|^2}_{\mathcal{T}_2}+\underbrace{2\delta^2\frac{1}{|\mathcal{B}|}\sum_{j\in\mathcal{B}}\mathbb{E}\|\boldsymbol{y}_{j,l}-\boldsymbol{\mu}\|^2}_{\mathcal{T}_3}\,.
\end{aligned}$$

Thus, the error can be decomposed into 3 terms: $\mathcal{T}_1$ corresponds to the bias introduced by our clipping operation in the good workers, $\mathcal{T}_2$ is the variance of the clipped good workers, and finally $\mathcal{T}_3$ is the error due to the bad workers. We will analyze each of the three errors in turn,

$\mathcal{T}_3$. For any bad index $j\in\mathcal{B}$, we can bound the error using our clipping radius as for any parameter $\gamma>0$ as

$$\mathbb{E}\|\boldsymbol{y}_{j,l}-\boldsymbol{\mu}\|^2\leq(1+\tfrac{1}{\gamma})\,\mathbb{E}\|\boldsymbol{y}_{j,l}-\boldsymbol{v}_l\|^2+(1+\gamma)\,\mathbb{E}\|\boldsymbol{v}_l-\boldsymbol{\mu}\|^2\leq(1+\gamma)\tau_l^2+(1+\tfrac{1}{\gamma})B_l^2\,.$$

The first step used Young's inequality. Further, the error due to the bad buys is also smaller if our initial estimation error B_l^2 is small.

$$\begin{aligned}\mathcal{T}_2 &= \mathbb{E}\,\frac{1}{(|\mathcal{G}|)^2}\sum_{i\in\mathcal{G}}\|\boldsymbol{y}_{i,l}-\mathbb{E}[\boldsymbol{y}_{i,l}]\|^2\\ &\leq \mathbb{E}\,\frac{1}{(|\mathcal{G}|)^2}\sum_{i\in\mathcal{G}}\|\boldsymbol{x}_{i}-\mathbb{E}[\boldsymbol{x}_{i}]\|^2\\ &\leq \frac{\rho^2}{|\mathcal{G}|}\leq\frac{2\rho^2}{n}\leq 2\delta\rho^2\,.\end{aligned}$$

The equality in the first step used the fact that the quantities were independent, and the next inequality follows because of the contractivity of a clipping (projection) step. The last used the fact that $|\mathcal{G}|\geq n/2$.

$\mathcal{T}_1$**.** We finally compute the bias in the update of a good worker $i\in\mathcal{G}$ due to the clipping operation. Let $\mathbb{1}_{i,l}$ be an indicator variable denoting if the ith worker was clipped (i.e. its distance from $\boldsymbol{v}_l$ exceeding τ_l). Note that if $\mathbb{1}_{i,l}=0$, we have that $\boldsymbol{y}_{i,l}=\boldsymbol{x}_i$. Then,

$$\begin{aligned}\mathbb{E}\|\boldsymbol{y}_{i,l}-\boldsymbol{x}_i\| &= \mathbb{E}\,\mathbb{1}_{i,l}\|\boldsymbol{y}_{i,l}-\boldsymbol{x}_i]\| \leq \mathbb{E}\,\mathbb{1}_{i,l}\|\boldsymbol{v}_l-\boldsymbol{x}_i]\| \leq \frac{\mathbb{E}\,\mathbb{1}_{i,l}\|\boldsymbol{v}_l-\boldsymbol{x}_i\|^2}{\tau}\\ &\leq \frac{\mathbb{E}\|\boldsymbol{v}_l-\boldsymbol{x}_i\|^2}{\tau}\leq\frac{(1+\frac{1}{\gamma})\,\mathbb{E}\|\boldsymbol{v}_l-\boldsymbol{\mu}\|^2+(1+\gamma)E\|\boldsymbol{x}_i-\boldsymbol{\mu}\|^2}{\tau}\\ &\leq \frac{(1+\frac{1}{\gamma})\rho^2+(1+\gamma)B_l^2}{\tau}\,.\end{aligned}$$

Using this, we can compute the error as

$$\mathcal{T}_1\leq\frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}}\|\mathbb{E}[\boldsymbol{y}_{i,l}]-\mathbb{E}[\boldsymbol{x}_i]\|^2\leq\frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}}\mathbb{E}\|\boldsymbol{y}_{i,l}-\boldsymbol{x}_i\|^2\leq\frac{((1+\frac{1}{\gamma})\rho^2+(1+\gamma)B_l^2)^2}{\tau^2}\,.$$

Combining the three error terms, we have

$$\begin{aligned}\mathbb{E}\|\boldsymbol{v}_{l+1}-\boldsymbol{\mu}\|^2 &\leq 2(1-\delta)^2\frac{((1+\frac{1}{\gamma})\rho^2+(1+\gamma)B_l^2)^2}{\tau^2}+2(1-\delta)^2 2\delta\rho^2+2\delta^2\Big((1+\gamma)\tau_l^2+(1+\tfrac{1}{\gamma})B_l^2\Big)\\ &= (4(1-\delta)\delta(1+\gamma)^{3/2}+2(1+\tfrac{1}{\gamma})\delta^2)B_l^2+4(1-\delta)^2\delta\rho^2+(4(1-\delta)(1+\tfrac{1}{\gamma})\sqrt{1+\gamma})\delta\rho^2\\ &\leq (4(1-\delta)\delta(1+\tfrac{1}{3})^{3/2}+8\delta^2)B_l^2+4\delta\rho^2+(16\sqrt{1+\tfrac{1}{3}})\delta\rho^2\\ &\leq (6.158\delta(1-\delta)+8\delta^2)B_l^2+22\delta\rho^2\\ &\leq 6.45\delta B_l^2+22\delta\rho^2\,.\end{aligned}$$

The last step assumed that $\delta\leq 0.15$, and the one before that used $\gamma=\frac{1}{3}$. The equality in the second step used a clipping radius of

$$\tau_l^2=4(1-\delta)\frac{(4\rho^2+\frac{4}{3}B_l^2)}{\sqrt{3}\delta}\,.$$

Define B_{l+1} to be the final expression derived above. Then, unrolling the recursion gives:

$$B_l\leq(6.45\delta)^l B_0^2+677\delta\rho^2\,.$$

Finally, we finish the proof of the theorem using

$$\mathbb{E}\|\boldsymbol{v}_l-\bar{\boldsymbol{x}}\|^2\leq(1+\tfrac{1}{339})\,\mathbb{E}\|\boldsymbol{v}_l-\boldsymbol{\mu}\|^2+340\,\mathbb{E}\|\bar{\boldsymbol{x}}-\boldsymbol{\mu}\|^2\leq(6.45\delta)^l 2B_0^2+679\delta\rho^2+680\delta\rho^2$$

□

E. Proof of Theorem VI - Byzantine-Robust Convergence

We state several supporting Lemmas before proving our main Theorem VI.

Lemma 9 (Aggregation error)**.** *Given that Definition C holds, and that we use momentum constant parameter with $\alpha_1 = 1$ and $\alpha_t = \alpha$ for $t \geq 2$, the error between the ideal average momentum $\bar{\boldsymbol{m}}_t$ and the output of the robust aggregation rule $\boldsymbol{m}_t$ for any $t \geq 2$ can be bounded as*

$$\mathbb{E}\|\boldsymbol{m}_t - \bar{\boldsymbol{m}}_t\|^2 \leq 2c\delta\sigma^2(\alpha + (1-\alpha)^{t-1})\,.$$

For $t = 1$ we can simplify the bound as $\mathbb{E}\|\boldsymbol{m}_1 - \bar{\boldsymbol{m}}_1\|^2 \leq 2c\delta\sigma^2$.

Proof. Expanding the definition of the worker momentum for any two good workers $i, j \in \mathcal{G}$,

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{m}_{i,t} - \boldsymbol{m}_{j,t}\|^2 &= \mathbb{E}\|\alpha_t(\boldsymbol{g}_i(\boldsymbol{x}_{t-1}) - \boldsymbol{g}_j(\boldsymbol{x}_{t-1})) + (1-\alpha_t)(\boldsymbol{m}_{i,t-1} - \boldsymbol{m}_{j,t-1})\|^2 \\
&\leq \mathbb{E}\|(1-\alpha_t)(\boldsymbol{m}_{i,t-1} - \boldsymbol{m}_{j,t-1})\|^2 + 2\alpha_t^2\sigma^2 \\
&\leq (1-\alpha_t)\,\mathbb{E}\|\boldsymbol{m}_{i,t-1} - \boldsymbol{m}_{j,t-1}\|^2 + 2\alpha_t^2\sigma^2\,.
\end{aligned}$$

Recall that we use $\alpha_1 = 1$ and a fixed momentum $\alpha_t = \alpha$ the rest of the steps. Unrolling the recursion above yields

$$\mathbb{E}\|\boldsymbol{m}_{i,t} - \boldsymbol{m}_{j,t}\|^2 \leq \left(\sum_{\ell=2}^{t}(1-\alpha)^{t-\ell}\right)2\alpha^2\sigma^2 + (1-\alpha)^{t-1}2\sigma^2 \leq 2\sigma^2(\alpha + (1-\alpha)^{t-1})\,.$$

The previous computation shows that all the good vectors given to the server are close to each other with $\rho^2 = 2\sigma^2(\alpha + (1-\alpha)^{t-1})$. Hence, by Definition C the output of the aggregation rule $\mathrm{AGG}(\boldsymbol{m}_{t,1}, \ldots, \boldsymbol{m}_{t,n})$ satisfies the lemma statement. □

Lemma 10 (Descent bound)**.** *For $\alpha_1 = 1$ and any $\alpha_t \in [0,1]$ for $t \geq 2$, $\eta_t \leq \frac{1}{L}$, and an L-smooth function f we have for any $t \geq 1$*

$$\mathbb{E}_t[f(\boldsymbol{x}_t)] \leq f(\boldsymbol{x}_{t-1}) - \frac{\eta_t}{2}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 + \eta_t\,\mathbb{E}_t\|\bar{\boldsymbol{e}}_t\|^2 + \eta_t\,\mathbb{E}_t\|\boldsymbol{m}_t - \bar{\boldsymbol{m}}_t\|^2\,.$$

where $\bar{\boldsymbol{e}}_t := \bar{\boldsymbol{m}}_t - \nabla f(\boldsymbol{x}_{t-1})$.

Proof. By the smoothness of the function f and the server update,

$$\begin{aligned}
f(\boldsymbol{x}_t) &\leq f(\boldsymbol{x}_{t-1}) - \eta_t\langle\nabla f(\boldsymbol{x}_{t-1}), \boldsymbol{m}_t\rangle + \frac{L\eta_t^2}{2}\|\boldsymbol{m}_t\|^2 \\
&\leq f(\boldsymbol{x}_{t-1}) - \eta_t\langle\nabla f(\boldsymbol{x}_{t-1}), \boldsymbol{m}_t\rangle + \frac{\eta_t}{2}\|\boldsymbol{m}_t\|^2 \\
&= f(\boldsymbol{x}_{t-1}) + \frac{\eta_t}{2}\|\boldsymbol{m}_t - \nabla f(\boldsymbol{x}_{t-1})\|^2 - \frac{\eta_t}{2}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 \\
&= f(\boldsymbol{x}_{t-1}) + \frac{\eta_t}{2}\|\boldsymbol{m}_t \pm \bar{\boldsymbol{m}}_t - \nabla f(\boldsymbol{x}_{t-1})\|^2 - \frac{\eta_t}{2}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 \\
&\leq f(\boldsymbol{x}_{t-1}) + \eta_t\|\bar{\boldsymbol{e}}_t\|^2 + \eta_t\|\boldsymbol{m}_t - \bar{\boldsymbol{m}}_t\|^2 - \frac{\eta_t}{2}\|\nabla f(\boldsymbol{x}_{t-1})\|^2\,.
\end{aligned}$$

Taking conditional expectation on both sides yields the second part of the lemma. □

Lemma 11 (Error bound)**.** *Using any constant momentum and step-sizes such that $1 \geq \alpha \geq 8L\eta$ for $t \geq 2$, we have for an L-smooth function f that $\mathbb{E}\|\bar{\boldsymbol{e}}_1\|^2 \leq \frac{2\sigma^2}{n}$ and for $t \geq 2$*

$$\mathbb{E}\|\bar{\boldsymbol{e}}_t\|^2 \leq (1 - \tfrac{2\alpha}{5})\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + \tfrac{\alpha}{10}\,\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-2})\|^2 + \tfrac{\alpha}{10}\,\mathbb{E}\|\boldsymbol{m}_{t-1} - \bar{\boldsymbol{m}}_{t-1}\|^2 + \alpha^2\tfrac{2\sigma^2}{n}\,.$$

Proof. Using the definitions (2) and proceeding as in Lemma 8, we have

$$\begin{aligned}
\mathbb{E}\|\bar{\boldsymbol{e}}_t\|^2 &= \mathbb{E}\|\bar{\boldsymbol{m}}_t - \nabla f(\boldsymbol{x}_{t-1})\|^2 \\
&= \mathbb{E}\|\alpha_t \bar{\boldsymbol{g}}(\boldsymbol{x}_{t-1}) + (1-\alpha_t)\bar{\boldsymbol{m}}_{t-1} - \nabla f(\boldsymbol{x}_{t-1})\|^2 \\
&\le \mathbb{E}\|\alpha_t \nabla f(\boldsymbol{x}_{t-1}) + (1-\alpha_t)\bar{\boldsymbol{m}}_{t-1} - \nabla f(\boldsymbol{x}_{t-1})\|^2 + \alpha_t^2 \tfrac{2\sigma^2}{n} \\
&= (1-\alpha_t)^2\, \mathbb{E}\|(\bar{\boldsymbol{m}}_{t-1} - \nabla f(\boldsymbol{x}_{t-2})) + (\nabla f(\boldsymbol{x}_{t-2}) - \nabla f(\boldsymbol{x}_{t-1}))\|^2 + \alpha_t^2 \tfrac{2\sigma^2}{n} \\
&\le (1-\alpha_t)(1+\tfrac{\alpha_t}{2})\, \mathbb{E}\|(\bar{\boldsymbol{m}}_{t-1} - \nabla f(\boldsymbol{x}_{t-2}))\|^2 + (1-\alpha_t)(1+\tfrac{2}{\alpha_t})\, \mathbb{E}\|\nabla f(\boldsymbol{x}_{t-2}) - \nabla f(\boldsymbol{x}_{t-1})\|^2 + \alpha_t^2 \tfrac{2\sigma^2}{n} \\
&\le (1-\tfrac{\alpha_t}{2})\, \mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + \tfrac{2L^2}{\alpha_t}\, \mathbb{E}\|\boldsymbol{x}_{t-2} - \boldsymbol{x}_{t-1}\|^2 + \alpha_t^2 \tfrac{2\sigma^2}{n} \\
&= (1-\tfrac{\alpha_t}{2})\, \mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + \tfrac{2L^2\eta_{t-1}^2}{\alpha_t}\, \mathbb{E}\|\boldsymbol{m}_{t-1}\|^2 + \alpha_t^2 \tfrac{2\sigma^2}{n}\,.
\end{aligned}$$

Note that we have $\frac{2\sigma^2}{n}$ instead of simply σ^2 since we average the momentums (and hence also the stochastic gradients) over all the good workers (who number at least $n/2$). Another difference is that in the last equality we have the robust aggregate $\boldsymbol{m}_{t-1}$ instead of the average momentum $\bar{\boldsymbol{m}}_{t-1}$. We can proceed as

$$\begin{aligned}
\mathbb{E}\|\bar{\boldsymbol{e}}_t\|^2 &\le (1-\tfrac{\alpha}{2})\, \mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + \tfrac{2L^2\eta^2}{\alpha}\, \mathbb{E}\|\boldsymbol{m}_{t-1}\|^2 + \alpha^2 \tfrac{2\sigma^2}{n} \\
&= (1-\tfrac{\alpha}{2})\, \mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + \tfrac{2L^2\eta^2}{\alpha}\, \mathbb{E}\|\boldsymbol{m}_{t-1} \pm \bar{\boldsymbol{m}}_{t-1} \pm \nabla f(\boldsymbol{x}_{t-2})\|^2 + \alpha^2 \tfrac{2\sigma^2}{n} \\
&\le (1-\tfrac{\alpha}{2})\, \mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + \tfrac{6L^2\eta^2}{\alpha}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + \tfrac{6L^2\eta^2}{\alpha}\, \mathbb{E}\|\boldsymbol{m}_{t-1} - \bar{\boldsymbol{m}}_{t-1}\|^2 + \tfrac{6L^2\eta^2}{\alpha}\, \mathbb{E}\|\nabla f(\boldsymbol{x}_{t-2})\|^2 + \alpha^2 \tfrac{2\sigma^2}{n}\,.
\end{aligned}$$

Our choice of the momentum parameter α implies $64L^2\eta^2 \le \alpha^2$ and yields the lemma statement.

□

Proof of Theorem VI. We will loosely follow the proof of vanilla SGDm proof in Theorem VIII. Recall that $\mathcal{G}$ denotes the good set and $\mathcal{B}$ denotes the bad Byzantine workers with $|\mathcal{G}| \le (1-\delta)n$ and $|\mathcal{B}| = n - |\mathcal{G}| \le \delta n$. Define the ideal momentum and error as

$$\bar{\boldsymbol{m}}_t := \frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} \boldsymbol{m}_{t,j}\,, \quad \bar{\boldsymbol{e}}_t := \bar{\boldsymbol{m}}_t - \nabla f(\boldsymbol{x}_{t-1})\,, \quad \text{and } \bar{\boldsymbol{g}}(\boldsymbol{x}_{t-1}) = \frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} \boldsymbol{g}_j(\boldsymbol{x}_{t-1})\,. \tag{2}$$

Now scale the modified error bound Lemma 11 by $\frac{5\eta}{2\alpha}$ and add it to the modified descent bound Lemma 10 taking expectations on both sides to get for $t \ge 2$

$$\begin{aligned}
\mathbb{E}[f(\boldsymbol{x}_t)] + \tfrac{5\eta}{2\alpha}\, \mathbb{E}\|\bar{\boldsymbol{e}}_t\|^2 \le \mathbb{E}[f(\boldsymbol{x}_{t-1})] - \tfrac{\eta}{2}\, \mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 + \eta\, \mathbb{E}\|\bar{\boldsymbol{e}}_t\|^2 + \eta\, \mathbb{E}\|\boldsymbol{m}_t - \bar{\boldsymbol{m}}_t\|^2 + \\
\tfrac{5\eta}{2\alpha}\, \mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 - \eta\, \mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + \tfrac{\eta}{4}\, \mathbb{E}\|\nabla f(\boldsymbol{x}_{t-2})\|^2 + \tfrac{\eta}{4}\, \mathbb{E}\|\boldsymbol{m}_{t-1} - \bar{\boldsymbol{m}}_{t-1}\|^2 + 5\eta\alpha\frac{\sigma^2}{n}
\end{aligned}$$

Rearranging the above terms and using the bound in the aggregation error Lemma 9 yields the recursion

$$\begin{aligned}
\underbrace{\mathbb{E}\ f(\boldsymbol{x}_t) - f^\star + (\tfrac{5\eta}{2\alpha} - \eta)\, \mathbb{E}\|\bar{\boldsymbol{e}}_t\|^2 + \frac{\eta}{4}\, \mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2}_{=:\xi_t} &\le \underbrace{\mathbb{E}\ f(\boldsymbol{x}_{t-1}) - f^\star + (\tfrac{5\eta}{2\alpha} - \eta)\, \mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + \frac{\eta}{4}\, \mathbb{E}\|\nabla f(\boldsymbol{x}_{t-2})\|^2}_{=:\xi_{t-1}} \\
&\quad - \frac{\eta}{4}\, \mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 + \frac{5\eta\alpha}{n}\sigma^2 + \frac{5\eta}{4}\, \mathbb{E}\|\boldsymbol{m}_{t-1} - \bar{\boldsymbol{m}}_{t-1}\|^2 \\
&\le \xi_{t-1} - \frac{\eta}{4}\, \mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 \\
&\quad + \frac{5\eta\alpha\sigma^2}{2}\left(\frac{2}{n} + \delta(c + \tfrac{c}{\alpha}(1-\alpha)^{t-2})\right).
\end{aligned}$$

Further, specializing the descent bound Lemma 10 and error bound Lemma 11 for $t = 1$ we have

$$\begin{aligned}\xi_1 &\leq \mathbb{E}\ f(\boldsymbol{x}_1) - f^\star + \frac{3\eta}{2}\,\mathbb{E}\|\bar{\boldsymbol{e}}_1\|^2 + \frac{\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{x}_0)\|^2 \\ &\leq f(\boldsymbol{x}_0) - f^\star + \frac{5\eta}{2}\,\mathbb{E}\|\bar{\boldsymbol{e}}_1\|^2 - \frac{\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{x}_0)\|^2 + \eta\,\mathbb{E}\|\boldsymbol{m}_1 - \bar{\boldsymbol{m}}_1\|^2 \\ &\leq f(\boldsymbol{x}_0) - f^\star - \frac{\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{x}_0)\|^2 + \frac{5\eta\sigma^2}{n} + 2c\eta\delta\sigma^2\,.\end{aligned}$$

Summing over t and again rearranging our recursion for ξ_t gives

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 &\leq \frac{4(f(\boldsymbol{x}_0)-f^\star)}{\eta T} + \frac{20\sigma^2}{nT} + \frac{8c\delta\sigma^2}{T} \\ &\quad + \frac{10\alpha\sigma^2}{T}\sum_{t=1}^{T}\left(\frac{2}{n} + \delta(c + \tfrac{c}{\alpha}(1-\alpha)^{t-2})\right) \\ &\leq \frac{4(f(\boldsymbol{x}_0)-f^\star)}{\eta T} + \frac{20\sigma^2}{nT} + \frac{8c\delta\sigma^2}{T} \\ &\quad + \frac{20\alpha\sigma^2}{n} + 10c\delta\alpha\sigma^2 + \frac{10\delta c\alpha\sigma^2}{\alpha^2 T} \\ &= \frac{4(f(\boldsymbol{x}_0)-f^\star)}{\eta T} + \frac{20\sigma^2}{nT} + \frac{8c\delta\sigma^2}{T} \\ &\quad + \frac{160L\eta\sigma^2}{n} + 80L\delta c\eta\sigma^2 + \frac{5c\delta\sigma^2}{4L\eta T} \\ &\leq 16\sqrt{\frac{5\sigma^2(2+c\delta n)}{nT}\big(L(f(\boldsymbol{x}_0)-f^\star) + \tfrac{5c\delta}{16}\sigma^2\big)} \\ &\quad + \frac{32L(f(\boldsymbol{x}_0)-f^\star)}{T} + \frac{10c\delta\sigma^2}{T} + \frac{20\sigma^2}{nT} + \frac{8c\delta\sigma^2}{T}\,.\end{aligned}$$

Substituting the appropriate step-size $\eta = \min\left(\sqrt{\frac{f(\boldsymbol{x}_0)-f^\star+\frac{5c\delta}{16L}\sigma^2}{20LT\sigma^2\left(\frac{2}{n}+c\delta\right)}}, \frac{1}{8L}\right)$ finishes the proof of the theorem. □

F. Proof of Theorem VII (Momentum based variance reduction)

We now describe how to modify the momentum method with a small correction term to improve its convergence rate (Cutkosky & Orabona, 2019). Starting from a given $\boldsymbol{x}_0$ and with $\boldsymbol{d}_0 = 0$, $\alpha_1 = 1$, we run the following updates with a sequence of momentum parameters $\alpha_t \in [0,1]$ and step-sizes $\eta_t \geq 0$ for $t \geq 2$

$$\boldsymbol{d}_{t,i} = \alpha_t \boldsymbol{g}_{t,i}(\boldsymbol{x}_{t-1}) + (1-\alpha_t)\boldsymbol{d}_{t-1,i} + (1-\alpha_t)(\boldsymbol{g}_{t,i}(\boldsymbol{x}_{t-1}) - \boldsymbol{g}_{t,i}(\boldsymbol{x}_{t-2})) \qquad \text{(MVR-WORKER)}$$

Note that both $\boldsymbol{g}_{t,i}(\boldsymbol{x}_{t-1})$ and $\boldsymbol{g}_{t,i}(\boldsymbol{x}_{t-2})$ here are computed using the same stochastic function (same batch) as indicated by the subscript. The good workers communicate $\boldsymbol{d}_{t,i}$ whereas the bad ones send arbitrary vectors. Then, the server performs

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

Define $\bar{\boldsymbol{d}}_t := \frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}}\boldsymbol{d}_{t,j}$ and $\bar{\boldsymbol{e}}_t := \bar{\boldsymbol{d}}_t - \nabla f(\boldsymbol{x}_{t-1})$. Note that since $\alpha_1 = 1$, the first step can be simplified as $\boldsymbol{d}_{1,i} = \boldsymbol{g}_{1,i}(\boldsymbol{x}_0)$. Here we assume that the stochastic gradient conditioned on all past history is unbiased $\mathbb{E}_t[\boldsymbol{g}_{t,i}(\boldsymbol{x}_{t-1})] = \nabla f(\boldsymbol{x}_{t-1})$ and has bounded variance σ^2. Further, we assume that the stochastic gradients satisfy $\mathbb{E}\|\boldsymbol{g}_{t,i}(\boldsymbol{x}_{t-1}) - \boldsymbol{g}_{t,i}(\boldsymbol{x}_{t-2})\|^2 \leq L^2\|\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2}\|^2$. This is stronger than assuming only that the full gradient ∇f is Lipschitz.

Lemma 12. *For $\alpha_1 = 1$ and any $\alpha_t \in [0,1]$ for $t \geq 2$, $\eta_t \leq \frac{1}{L}$, and an L-smooth function f we have that $E_1[f(\boldsymbol{x}_1)] \leq f(\boldsymbol{x}_0) - \frac{\eta_t}{2}\|\nabla f(\boldsymbol{x}_0)\|^2 + \frac{\eta_t^2 L}{2}\sigma^2$ and for $t \geq 2$ with $\bar{\boldsymbol{e}}_t := \bar{\boldsymbol{d}}_t - \nabla f(\boldsymbol{x}_{t-1})$ we have*

$$\mathbb{E}_t[f(\boldsymbol{x}_t)] \leq f(\boldsymbol{x}_{t-1}) - \frac{\eta_t}{2}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 + \eta\big(\mathbb{E}_t\|\bar{\boldsymbol{e}}_t\|^2 + \mathbb{E}_t\|\boldsymbol{d}_t - \bar{\boldsymbol{d}}_t\|^2\big)\,.$$

The proof is identical to that of Lemma 7.

Lemma 13. *Using any momentum and step-sizes such that $1 \geq \alpha \geq 16L^2\eta^2$ for $t \geq 2$, we have i) $\mathbb{E}[\boldsymbol{e}_t] = 0$, and ii) for an L-smooth function f that $\mathbb{E}\|\bar{\boldsymbol{e}}_1\|^2 \leq 2\sigma^2/n$ and for $t \geq 2$*

$$\mathbb{E}\|\bar{\boldsymbol{e}}_t\|^2 \leq (1-\tfrac{\alpha}{2})\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + 8L^2\eta^2\|\nabla f(\boldsymbol{x}_{t-2})\|^2 + 2\alpha^2\sigma^2/n + 4L^2\eta^2\,\mathbb{E}\|\boldsymbol{d}_{t-1} - \bar{\boldsymbol{d}}_{t-1}\|^2\,.$$

Proof. Starting from the definition of $\bar{\boldsymbol{e}}_{t+1}$ and $\bar{\boldsymbol{d}}_t$,

$$\begin{aligned}
\bar{\boldsymbol{e}}_t &= \bar{\boldsymbol{d}}_t - \nabla f(\boldsymbol{x}_{t-1}) \\
&= \alpha\bar{\boldsymbol{g}}_t(\boldsymbol{x}_{t-1}) + (1-\alpha)\bar{\boldsymbol{d}}_{t-1} + (1-\alpha)(\bar{\boldsymbol{g}}_t(\boldsymbol{x}_{t-1}) - \bar{\boldsymbol{g}}_t(\boldsymbol{x}_{t-2})) - \nabla f(\boldsymbol{x}_{t-1}) \\
&= \underbrace{(1-\alpha)(\bar{\boldsymbol{d}}_{t-1} - \nabla f(\boldsymbol{x}_{t-2}))}_{\mathcal{T}_1} + \\
&\qquad \underbrace{\alpha(\bar{\boldsymbol{g}}_t(\boldsymbol{x}_{t-1}) - \nabla f(\boldsymbol{x}_{t-1}))}_{\mathcal{T}_2} + \underbrace{(1-\alpha)(\bar{\boldsymbol{g}}_t(\boldsymbol{x}_{t-1}) - \bar{\boldsymbol{g}}_t(\boldsymbol{x}_{t-2}) - \nabla f(\boldsymbol{x}_{t-1}) + \nabla f(\boldsymbol{x}_{t-2}))}_{\mathcal{T}_3}\,.
\end{aligned}$$

Note that $\mathcal{T}_1 = (1-\alpha)\bar{\boldsymbol{e}}_{t-1}$ and that $\mathbb{E}[\mathcal{T}_2] = 0, \mathbb{E}[\mathcal{T}_3] = 0$. This proves that $\mathbb{E}[\bar{\boldsymbol{e}}_t] = 0$. Further, conditioned on all history $\mathcal{F}_t$ (i.e. everything before step t), we have $\mathbb{E}_t[\mathcal{T}_2] = 0$ and $\mathbb{E}_t[\mathcal{T}_3] = 0$ and $\mathcal{T}_1$ is deterministic. Hence, we can take squared norms on both sides as expand as

$$\begin{aligned}
\mathbb{E}\|\bar{\boldsymbol{e}}_t\|^2 &= (1-\alpha)^2\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + \\
&\qquad \mathbb{E}\|\alpha(\bar{\boldsymbol{g}}_t(\boldsymbol{x}_{t-1}) - \nabla f(\boldsymbol{x}_{t-1})) + (1-\alpha)(\bar{\boldsymbol{g}}_t(\boldsymbol{x}_{t-1}) - \bar{\boldsymbol{g}}_t(\boldsymbol{x}_{t-2}) - \nabla f(\boldsymbol{x}_{t-1}) + \nabla f(\boldsymbol{x}_{t-2}))\|^2 \\
&\leq (1-\alpha)^2\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + \\
&\qquad 2\,\mathbb{E}\|\alpha(\bar{\boldsymbol{g}}_t(\boldsymbol{x}_{t-1}) - \nabla f(\boldsymbol{x}_{t-1}))\|^2 + 2\|(1-\alpha)(\bar{\boldsymbol{g}}_t(\boldsymbol{x}_{t-1}) - \bar{\boldsymbol{g}}_t(\boldsymbol{x}_{t-2}) - \nabla f(\boldsymbol{x}_{t-1}) + \nabla f(\boldsymbol{x}_{t-2}))\|^2 \\
&\leq (1-\alpha)\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + 2\alpha^2\sigma^2/n + 2(1-\alpha)^2\,\mathbb{E}\|\bar{\boldsymbol{g}}_t(\boldsymbol{x}_{t-1}) - \bar{\boldsymbol{g}}_t(\boldsymbol{x}_{t-2})\|^2 \\
&\leq (1-\alpha)\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + 2\alpha^2\sigma^2/n + 2(1-\alpha)^2L^2\,\mathbb{E}\|\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2}\|^2 \\
&= (1-\alpha)\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + 2\alpha^2\sigma^2/n + 4(1-\alpha)^2L^2\eta^2\,\mathbb{E}\|\bar{\boldsymbol{d}}_{t-1}\|^2 + 4(1-\alpha)^2L^2\eta^2\,\mathbb{E}\|\boldsymbol{d}_{t-1} - \bar{\boldsymbol{d}}_{t-1}\|^2 \\
&\leq (1-\alpha)\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + 2\alpha^2\sigma^2/n + 8L^2\eta^2\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + 8L^2\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-2})\|^2 + 4L^2\eta^2\,\mathbb{E}\|\boldsymbol{d}_{t-1} - \bar{\boldsymbol{d}}_{t-1}\|^2\,.
\end{aligned}$$

Here the the third inequality used the expected squared Lipschitzness of $g_t(\,\cdot\,)$, whereas the rest relied on Young's inequality and that $\alpha \in [0,1]$. Now the condition on the momentum implies that $8L^2\eta^2 \leq \frac{\alpha}{2}$, yielding the second statement of the lemma for $t \geq 2$. The statement for $\boldsymbol{e}_1$ follows since $\bar{\boldsymbol{d}}_0 = 0$. □

Lemma 14 (Aggregation error)**.** *Given Definition C holds and we use a momentum constant parameter $\alpha_1 = 1$ and $\alpha_t = \alpha \geq 192L^2\eta^2(c\delta+1)$ for $t \geq 2$, the error between the ideal average momentum $\bar{\boldsymbol{d}}_t$ and the robust aggregate $\boldsymbol{d}_t$ for any $t \geq 2$ can be bounded as*

$$\begin{aligned}
\mathbb{E}\|\bar{\boldsymbol{e}}_t\|^2 + c\delta\,\mathbb{E}\|\boldsymbol{d}_{i,t} - \boldsymbol{d}_{j,t}\|^2 \leq (1-\tfrac{\alpha}{4})\big(\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + c\delta\,\mathbb{E}\|\boldsymbol{d}_{i,t-1} - \boldsymbol{d}_{j,t-1}\|^2\big) \\
+ \tfrac{\alpha}{16}\|\nabla f(\boldsymbol{x}_{t-2})\|^2 + (c\delta + 1/n)4\alpha^2\sigma^2
\end{aligned}$$

For $t = 1$, we can simplify the bound to $\mathbb{E}\|\bar{\boldsymbol{e}}_1\|^2 + c\delta\,\mathbb{E}\|\boldsymbol{d}_{i,2} - \boldsymbol{d}_{j,2}\|^2 \leq 2\sigma^2(c\delta + 1/n)$.

Proof. Expanding the definition of the worker momentum for any two good workers $i, j \in \mathcal{G}$ for $t \geq 2$,

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{d}_{i,t} - \boldsymbol{d}_{j,t}\|^2 &= \mathbb{E}\,\|\alpha(\boldsymbol{g}_i(\boldsymbol{x}_{t-1}) - \boldsymbol{g}_j(\boldsymbol{x}_{t-1})) + \\
&\qquad (1-\alpha)(\boldsymbol{d}_{i,t-1} - \boldsymbol{d}_{j,t-1}) + \\
&\qquad (1-\alpha)(\boldsymbol{g}_{t,i}(\boldsymbol{x}_{t-1}) - \boldsymbol{g}_{t,j}(\boldsymbol{x}_{t-1}) - \boldsymbol{g}_{t,i}(\boldsymbol{x}_{t-2}) + \boldsymbol{g}_{t,j}(\boldsymbol{x}_{t-2}))\|^2 \\
&\leq \mathbb{E}\|(1-\alpha)(\boldsymbol{d}_{i,t-1} - \boldsymbol{d}_{j,t-1})\|^2 + 4\alpha^2\sigma^2 + 4L^2(1-\alpha)^2\,\mathbb{E}\|\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2}\|^2 \\
&\leq (1-\alpha)\,\mathbb{E}\|\boldsymbol{d}_{i,t-1} - \boldsymbol{d}_{j,t-1}\|^2 + 4\alpha^2\sigma^2 + 4L^2\eta^2\,\mathbb{E}\|\boldsymbol{d}_{t-1}\|^2 \\
&\leq (1-\alpha)\,\mathbb{E}\|\boldsymbol{d}_{i,t-1} - \boldsymbol{d}_{j,t-1}\|^2 + 4\alpha^2\sigma^2 + 12L^2\eta^2\,\mathbb{E}\|\boldsymbol{d}_{t-1} - \bar{\boldsymbol{d}}_{t-1}\|^2 \\
&\qquad + 12L^2\eta^2\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + 12L^2\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-2})\|^2 \\
&\leq (1-\alpha+12c\delta L^2\eta^2)\,\mathbb{E}\|\boldsymbol{d}_{i,t-1} - \boldsymbol{d}_{j,t-1}\|^2 + 4\alpha^2\sigma^2 \\
&\qquad + 12L^2\eta^2\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + 12L^2\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-2})\|^2\,.
\end{aligned}$$

Scale this by $c\delta$ and then add the inequality from Lemma 13 to get

$$\begin{aligned}
\mathbb{E}\|\bar{\boldsymbol{e}}_t\|^2 + c\delta\,\mathbb{E}\|\boldsymbol{d}_{i,t} - \boldsymbol{d}_{j,t}\|^2 &\leq (1-\tfrac{\alpha}{2})\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + 8L^2\eta^2\|\nabla f(\boldsymbol{x}_{t-2})\|^2 + 2\alpha^2\sigma^2/n + 4L^2\eta^2\,\mathbb{E}\|\boldsymbol{d}_{t-1} - \bar{\boldsymbol{d}}_{t-1}\|^2 \\
&\qquad + (1-\alpha+12c\delta L^2\eta^2)c\delta\,\mathbb{E}\|\boldsymbol{d}_{i,t-1} - \boldsymbol{d}_{j,t-1}\|^2 + 4c\delta\alpha^2\sigma^2 \\
&\qquad + 12c\delta L^2\eta^2\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + 12c\delta L^2\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-2})\|^2 \\
&\leq (1-\tfrac{\alpha}{2})\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + 8L^2\eta^2\|\nabla f(\boldsymbol{x}_{t-2})\|^2 + 2\alpha^2\sigma^2/n \\
&\qquad + (1-\alpha+12c\delta L^2\eta^2 + 4L^2\eta^2)c\delta\,\mathbb{E}\|\boldsymbol{d}_{i,t-1} - \boldsymbol{d}_{j,t-1}\|^2 + 4c\delta\alpha^2\sigma^2 \\
&\qquad + 12c\delta L^2\eta^2\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + 12c\delta L^2\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-2})\|^2 \\
&= (1-\tfrac{\alpha}{2}+12c\delta L^2\eta^2)\,\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + (8L^2\eta^2 + 12c\delta L^2\eta^2)\|\nabla f(\boldsymbol{x}_{t-2})\|^2 + (4c\delta + 2/n)\alpha^2\sigma^2 \\
&\qquad + (1-\alpha+12c\delta L^2\eta^2 + 4L^2\eta^2)c\delta\,\mathbb{E}\|\boldsymbol{d}_{i,t-1} - \boldsymbol{d}_{j,t-1}\|^2 \\
&\leq (1-\tfrac{\alpha}{4})\big(\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + c\delta\,\mathbb{E}\|\boldsymbol{d}_{i,t-1} - \boldsymbol{d}_{j,t-1}\|^2\big) + \tfrac{\alpha}{16}\|\nabla f(\boldsymbol{x}_{t-2})\|^2 + (4c\delta + 2/n)\alpha^2\sigma^2\,.
\end{aligned}$$

Here we used $\alpha \geq 192L^2\eta^2(c\delta+1)$, and Definition C that $\mathbb{E}\|\boldsymbol{d}_{t-1} - \bar{\boldsymbol{d}}_{t-1}\|^2 \leq c\delta\,\mathbb{E}\|\boldsymbol{d}_{i,t-1} - \boldsymbol{d}_{j,t-1}\|^2$.

□

We are now ready to prove the convergence theorem.

Theorem IX (Byzantine robust MVR)**.** *Let us run the MVR algorithm combined with a robust aggregation rule* AGG*with step-size* $\eta = \min\left(\sqrt[3]{\frac{f(\boldsymbol{x}_0)-f^\star}{T(1536L^2\sigma^2(c\delta+1)(c\delta+1/n))}}, \frac{1}{4L}\right)$ *and momentum parameter* $\alpha = 192L^2\eta^2(1+c\delta)$*. Then,*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 \leq 120\left(\frac{L\sigma\sqrt{(c\delta+1/n)(c\delta+1)}(f(\boldsymbol{x}_0)-f^\star)}{T}\right)^{\frac{2}{3}} + \frac{16L(f(\boldsymbol{x}_0)-f^\star) + 32\sigma^2(c\delta+1/n)}{T}\,.$$

Proof. Scaling Lemma 14 by $\frac{4\eta}{\alpha}$ and adding it to Lemma 12 we have for any $t \geq 2$

$$\begin{aligned}
(\mathbb{E}\;f(\boldsymbol{x}_t) - f^\star) + \frac{4\eta}{\alpha}\big(\mathbb{E}\|\bar{\boldsymbol{e}}_t\|^2 + c\delta\,\mathbb{E}\|\boldsymbol{d}_{i,t} - \boldsymbol{d}_{j,t}\|^2\big) &\leq (\mathbb{E}\;f(\boldsymbol{x}_{t-1}) - f^\star) + \frac{4\eta}{\alpha}\big(\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + c\delta\,\mathbb{E}\|\boldsymbol{d}_{i,t-1} - \boldsymbol{d}_{j,t-1}\|^2\big) \\
&\qquad - \eta\big(\mathbb{E}\|\bar{\boldsymbol{e}}_{t-1}\|^2 + c\delta\,\mathbb{E}\|\boldsymbol{d}_{i,t-1} - \boldsymbol{d}_{j,t-1}\|^2\big) \\
&\qquad - \frac{\eta}{2}\,\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 + \eta\big(\mathbb{E}\|\bar{\boldsymbol{e}}_t\|^2 + c\delta\,\mathbb{E}\|\boldsymbol{d}_{t,i} - \boldsymbol{d}_{t,j}\|^2\big) \\
&\qquad + \tfrac{\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-2})\|^2 + (c\delta+1/n)16\eta\alpha\sigma^2\,.
\end{aligned}$$

Define the constant

$$\xi_t := (\mathbb{E}\;f(\boldsymbol{x}_t) - f^\star) + \left(\frac{4\eta}{\alpha} - \eta\right)\big(\mathbb{E}\|\bar{\boldsymbol{e}}_t\|^2 + c\delta\,\mathbb{E}\|\boldsymbol{d}_{i,t} - \boldsymbol{d}_{j,t}\|^2\big) + \frac{\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2\,.$$

Then the previously stated inequality can be rearranged as

$$\frac{\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 \leq \xi_{t-1} - \xi_t + (c\delta + 1/n)16\eta\alpha\sigma^2\,.$$

Also note that $\xi_t \geq 0$ for any t and also for $t = 1$,

$$\begin{aligned}\xi_1 &= \mathbb{E}\ f(\boldsymbol{x}_1) - f^\star + \left(\frac{4\eta}{\alpha} - \eta\right)\left(\mathbb{E}\|\bar{\boldsymbol{e}}_t\|^2 + c\delta\,\mathbb{E}\|\boldsymbol{d}_{i,t} - \boldsymbol{d}_{j,t}\|^2\right) + \frac{\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{x}_0)\|^2 \\ &\leq f(\boldsymbol{x}_0) - f^\star - \frac{\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{x}_0)\|^2 + 8\eta\sigma^2(c\delta + 1/n)\,.\end{aligned}$$

Note that here we assumed a batch size of T in the first step to simplify computations. This does not change the asymptotic rate (multiplies it by 2), similar to (Tran-Dinh et al., 2020). This is easy to work around by using changing step-sizes/momentum values as shown by (Cutkosky & Orabona, 2019). Now summing over t and again rearranging gives

$$\frac{1}{\sum_{t=1}^{\ell}\eta}\sum_{t=1}^{\ell}\eta\,\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 \leq \frac{4(f(\boldsymbol{x}_0) - f^\star)}{\sum_{t=1}^{\ell}\eta} + \frac{1}{\sum_{t=1}^{\ell}\eta}\sum_{t=1}^{\ell}32(c\delta + 1/n)\eta\alpha\sigma^2\,.$$

For simplicity, let us use a constant $\eta = \min\left(\sqrt[3]{\frac{f(\boldsymbol{x}_0)-f^\star}{T(1536L^2\sigma^2(c\delta+1)^2)}}, \frac{1}{4L}\right)$ for $t \geq 1$ and momentum parameter $\alpha_1 = 1$ and $\alpha = 192L^2\eta^2(c\delta + 1)$ for $t \geq 2$. This simplifies the above inequality to

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1})\|^2 \leq \frac{4(f(\boldsymbol{x}_0) - f^\star)}{\eta T} + 6144L^2\eta^2(c\delta + 1)(c\delta + 1/n)\sigma^2 + \frac{32\sigma^2(c\delta + 1/n)}{T}\,.$$

Substituting the appropriate η yields the desired rate. □

G. Additional Experiments

G.1. Experiment setups

G.1.1. General setup

The default experiment setup is listed in Table 1. The default hyperparameters of the aggregators are summarized as follows

Aggregators	Hyperparameters
Krum	N/A
CM	N/A
RFA	$T = 3$
TM	$b = \delta$
CC	$\tau = 100$

About Figure 4. We have the following setup

- For all aggregators except mean, there are $n = 25$ workers and $n\delta = 11$ of them are Byzantine.
- For aggregator mean, there are $n = 14$ workers and 0 Byzantine workers.
- The IPM attack has strength of $\epsilon = 0.1$.
- The ALIE Attack has a hyperparameter z which is computed according to (Baruch et al., 2019)

$$z = \max_z\left(\phi(z) < \frac{n - n\delta - s}{n - n\delta}\right)$$

where $s = \lfloor \frac{n}{2} + 1 \rfloor - n\delta$ and ϕ is the cumulative standard normal function. In our setup, the $z \approx 1.06$.

G.1.2. Constructing datasets

Long-tailness. The MNIST dataset has 10 classes each with similar amount of samples. The long-tailness is achieved by sampling class with exponentially decreasing portions $\gamma \in (0, 1]$. That is, for class $i = 1, \ldots, 10$, we only randomly sample γ^i portion of all samples in class i. Note that the same procedure has to be applied to the test dataset.

Table 1: Default experimental settings for CIFAR-10 and MNIST.

Dataset	CIFAR-10	MNIST
Architecture	ResNet-20 (He et al., 2016)	CONV-CONV-DROPOUT-FC-DROPOUT-FC
Training objective	Cross entropy loss	Negative log likelihood loss
Evaluation objective	Top-1 accuracy	Top-1 accuracy
Batch size per worker	32	1
Momentum β	0 or 0.9 or 0.99	0
Learning rate	0.1	$\frac{0.1}{256}$
LR decay	0.1 at epoch 75	No
LR warmup	No	No
# Epochs / # Iterations	100 Epochs	800 Iterations
Weight decay	No	No
Repetitions	2, with varying seeds	2, with varying seeds

About dataset on Byzantine workers. The training set is divided by the number of good workers. So the good workers has to full information of training dataset. The Byzantine worker has access to the whole training dataset.

G.1.3. Running environment

We summarize the running environment of this paper as in Table 2.

Table 2: Runtime hardwares and softwares.

CPU	
Model name	Intel (R) Xeon (R) Gold 6132 CPU @ 2.60 GHz
# CPU(s)	56
NUMA node(s)	2
GPU	
Product Name	Tesla V100-SXM2-32GB
CUDA Version	11.0
PyTorch	
Version	1.7.1

G.2. Exploring local steps between aggregations

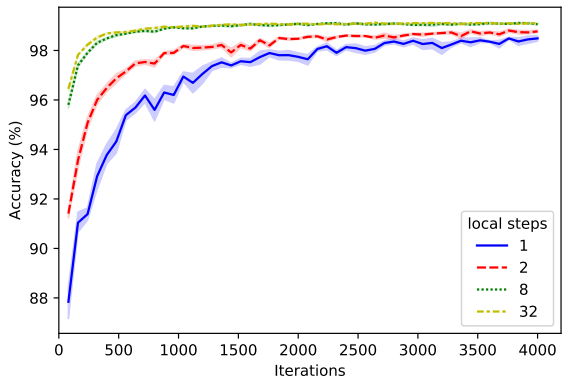

Figure 7: CC with 1, 2, 8, 32, local steps for MNIST dataset.

In this experiment, we combine CC with local SGD and bench marked on MNIST without attacker. The results in Fig. 7 shows that using higher local steps improves the accuracy and convergence rate. It supports that CC is compatible with localSGD.

G.3. Comparison with (Allen-Zhu et al., 2021)

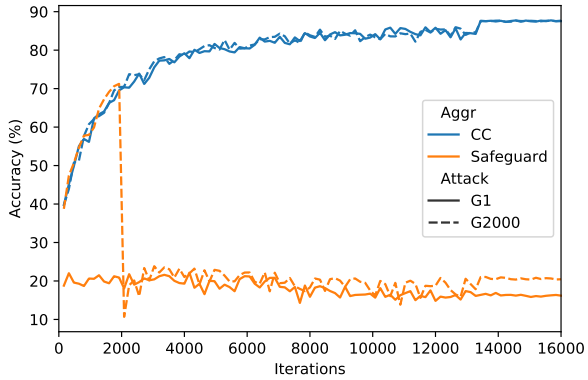

Figure 8: Comparing CC ($\tau = 100$) with Safeguard (Allen-Zhu et al., 2021) ($T_0 = 1$,$T_1 = 6$, $\mathfrak{T}_0 = 20$, $\mathfrak{T}_1 = 50$). The Byzantine workers send to the server vectors from a Gaussian distribution with standard deviation of 10^8. The "$G1$" attack inject attack at the 1st iteration while the "G2000" attack inject attack at the 2000th iteration. There are 10 nodes in total and 4 of them are Byzantine. The underlying dataset is Cifar10. We use batch size 32 and learning rate 0.1.

The recent independent work SAFEGUARD (Allen-Zhu et al., 2021) also uses historical information to detect Byzantine workers. However, as we discussed earlier, 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. In this secion, we conduct further empirical comparison of Safeguard and Centered clip CC.

First, note that Algorithm 1 in (Allen-Zhu et al., 2021) is vulnerable to simple attacks, e.g. sending an arbitrary vector of very large magnitude, while CC is not. This is because Safeguard uses information from the previous step to filter in the current step. This is necessary in order to make the algorithm amenable to analysis. This means that even if a Byzantine worker sends a very large bad update, the algorithm will apply it once and filter out the worker only from the next round onward. Thus, all Byzantine workers can ensure that their update is incorporated at least once. While theoretically this might not be problematic since the influence of a single update is limited, in practice this means that the Byzantine workers can push the training process to encounter `NaN`s, ensuring no chance of recovery.

To demonstrate the effect, we apply the Gaussian attack to SAFEGUARD and CC at $t = 1$ (G1) and $t = 2000$ (G2000). The Gaussian attacker sends to the server vectors of Gaussian distribution of standard deviation 10^8. Since the workers behave correctly until $t - 1$, they all belong to $\mathbf{good}_t$ and their updates are incorporated. While the Byzantine worker is removed from $\mathbf{good}_{t+1}$, the attack already succeeded and there is no chance of recovery. We show the experimental results in Figure 8. In contrast, CC (even without momentum) easily defends against such attacks.

Secondly, SAFEGUARD requires tuning additional parameters (e.g. $\mathfrak{T}_0$, $\mathfrak{T}_1$) for each kind of attack while CC does not. For example, SAFEGUARD uses $\mathfrak{T}_0 = 1$, $\mathfrak{T}_1 = 2$ for Bit-Flipping attack (Allen-Zhu et al., 2021, Appendix C.2.1) and $\mathfrak{T}_0 = 2$, $\mathfrak{T}_1 = 7$ for Label-Flipping attack (Allen-Zhu et al., 2021, Appendix C.2.3). However, by the definition of Byzantine attack, the attacker is allowed to adaptively change attacks *after* tuning. This makes it crucial to ensure that any Byzantine robust algorithm works without additional tuning. In contrast, CC uses $\tau = 100$ and $l = 1$ for all experiments in the paper unless otherwise clarified.