BYZANTINE-ROBUST LEARNING ON HETEROGENEOUS DATASETS VIA BUCKETING

Sai Praneeth Karimireddy[*] sai.karimireddy@epfl.ch
Lie He[*] lie.he@epfl.ch
Martin Jaggi martin.jaggi@epfl.ch

ABSTRACT

In Byzantine robust distributed or federated learning, a central server wants to train a machine learning model over data distributed across multiple workers. However, a fraction of these workers may deviate from the prescribed algorithm and send arbitrary messages. While this problem has received significant attention recently, most current defenses assume that the workers have identical data. For realistic cases when the data across workers are heterogeneous (non-iid), we design new attacks which circumvent current defenses, leading to significant loss of performance. We then propose a simple bucketing scheme that adapts existing robust algorithms to heterogeneous datasets at a negligible computational cost. We also theoretically and experimentally validate our approach, showing that combining bucketing with existing robust algorithms is effective against challenging attacks. Our work is the first to establish guaranteed convergence for the non-iid Byzantine robust problem under realistic assumptions.

1 INTRODUCTION

Distributed or federated machine learning, where the data is distributed across multiple workers, has become an increasingly important learning paradigm both due to growing sizes of datasets, as well as data privacy concerns. In such a setting, the workers collaborate to train a single model without directly transmitting their training data (McMahan et al., 2016; Bonawitz et al., 2019; Kairouz et al., 2019). However, by decentralizing the training across a vast number of workers we potentially open ourselves to new security threats. Due to the presence of agents in the network which are actively malicious, or simply due to system and network failures, some workers may disobey the protocols and send arbitrary messages; such workers are also known as *Byzantine* workers (Lamport et al., 2019). Byzantine robust optimization algorithms attempt to combine the updates received from the workers using robust aggregation rules and ensure that the training is not impacted by the presence of a small number of malicious workers.

While this problem has received significant recent attention due to its importance, (Blanchard et al., 2017; Yin et al., 2018; Alistarh et al., 2018; Karimireddy et al., 2021), most of the current approaches assume that the data present on each different worker has identical distribution. This assumption is very unrealistic in practice and heterogeneity is inherent in distributed and federated learning (Kairouz et al., 2019). In this work, we show that existing Byzantine aggregation rules catastrophically fail with very simple attacks (or sometimes even with no attacks) in realistic settings. We carefully examine the causes of these failures, and propose a simple solution which provably solves the Byzantine resilient optimization problem under heterogeneous workers.

Concretely, our contributions in this work are summarized below

- We show that when the data across workers is heterogeneous, existing aggregation rules fail to converge, even when no Byzantine adversaries are present. We also propose a simple new attack, *mimic*, which explicitly takes advantage of data heterogeneity and circumvents median-based defenses. Together, these highlight the fragility of existing methods in real world settings.
- We then propose a simple fix - a new bucketing step which can be used before any existing aggregation rule. We introduce a formal notion of a robust aggregator (ARAGG) and prove that existing methods like KRUM, coordinate-wise median (CM), and geometric median aka

[*]equal contribution.

robust federated averaging (RFA)—though insufficient on their own—become provably robust aggregators when augmented with our bucketing.

- We combine our notion of robust aggregator (ARAGG) with worker momentum to obtain optimal rates for Byzantine robust optimization with matching lower bounds. Unfortunately, our lower bounds imply that convergence to an exact optimum may not be possible due to heterogeneity. We then circumvent this lower bound and show that when heterogeneity is mild (or when the model is overparameterized), we can in fact converge to an exact optimum. This is the first result establishing convergence to the optimum for heterogeneous Byzantine robust optimization.
- Finally, we evaluate the effect of the proposed techniques (bucketing and worker momentum) against known and new attacks showcasing drastic improvement on realistic heterogeneously distributed datasets.

Setup and notations. Suppose that of the total n workers, the set of good workers is denoted by $\mathcal{G} \subseteq \{1, \ldots, n\}$. Our objective is to minimize

$$f(\boldsymbol{x}) := \tfrac{1}{|\mathcal{G}|} \textstyle\sum_{i \in \mathcal{G}} \left\{ f_i(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{\xi}_i}[F_i(\boldsymbol{x}; \boldsymbol{\xi}_i)] \right\} \tag{1}$$

where f_i is the loss function on worker i defined over its own (heterogeneous) data distribution $\boldsymbol{\xi}_i$. The (stochastic) gradient computed by a good worker $i \in \mathcal{G}$ over minibatch $\boldsymbol{\xi}_i$ is given as $\boldsymbol{g}_i(\boldsymbol{x}, \boldsymbol{\xi}_i) := \nabla F_i(\boldsymbol{x}; \boldsymbol{\xi}_i)$. The noise in every stochastic gradient is independent, unbiased with $\mathbb{E}_{\boldsymbol{\xi}_i}[\boldsymbol{g}_i(\boldsymbol{x}, \boldsymbol{\xi}_i)] = \nabla f_i(\boldsymbol{x})$, and has bounded variance $\mathbb{E}_{\boldsymbol{\xi}_i} \|\boldsymbol{g}_i(\boldsymbol{x}, \boldsymbol{\xi}_i) - \nabla f_i(\boldsymbol{x})\|^2 \leq \sigma^2$. Further, we assume that the data heterogeneity across the workers can be bounded as

$$\mathbb{E}_{j \sim \mathcal{G}} \|\nabla f_j(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2 \leq \zeta^2 \,, \quad \forall \boldsymbol{x} \,.$$

We write $\boldsymbol{g}_i^t$ or simply $\boldsymbol{g}_i$ instead of $\boldsymbol{g}_i(\boldsymbol{x}^t, \boldsymbol{\xi}_i^t)$ when there is no ambiguity.

Byzantine attack model. The set of Byzantine workers $\mathcal{B} \subset [n]$ is fixed over time, with the remaining workers $\mathcal{G}$ being good, i.e. $[n] = \mathcal{B} \uplus \mathcal{G}$. We write δ for the fraction of Byzantine workers, $|\mathcal{B}| =: q \leq \delta n$. The Byzantine workers can deviate arbitrarily from our protocol, sending any update to the server. Further, they can collude and may even know the states of all other workers.

Our modeling assumes that the practitioner picks a value of $\delta \in [0, 0.5)$. This δ reflects the level of robustness required. A choice of a large δ (say near 0.5) would mean that the system is very robust and can tolerate a large fraction of attackers, but the algorithm becomes much more conservative and slow. On the flip side, if the practitioner knows that the the number of Byzantine agents are going to be few, they can pick a small δ (say 0.05–0.1) ensuring some robustness with almost no impact on convergence. The choice of δ can also be formulated as how expensive do we want to make an attack? To carry out a succesful attack the attacker would need to control δ fraction of all workers. We recommend implementations claiming robustness be transparent about their choice of δ.

2 Related work

IID defenses. There has been a significant amount of recent work on the case when all workers have identical data distributions. Blanchard et al. (2017) initiated the study of Byzantine robust learning and proposed a distance-based aggregation approach KRUM and extended to (Mhamdi et al., 2018; Damaskinos et al., 2019). Yin et al. (2018) propose to use and analyze the coordinate-wise median (CM), and Pillutla et al. (2019) use approximate geometric median. Bernstein et al. (2018) propose to use the signs of gradients and then aggregate them by majority vote, however, Karimireddy et al. (2019) show that it may fail to converge. Most recently, Alistarh et al. (2018); Allen-Zhu et al. (2021); Mhamdi et al. (2021); Karimireddy et al. (2021) showcase how to use past gradients to more accurately filter iid Byzantine workers and specifically *time-coupled* attacks. In particular, our work builds on top of (Karimireddy et al., 2021) and non-trivially extends to the non-iid setting.

IID vs. Non-IID attacks. For the iid setting, the state-of-the-art attacks are *time-coupled* attacks (Baruch et al., 2019; Xie et al., 2020). These attacks introduce a small but consistent bias at every step which is hard to detect in any particular round, but accumulates over time and eventually leads to divergence, breaking most prior robust methods. Our work focuses on developing attacks (and defenses) which specifically take advantages of the non-iid setting. The non-iid setting also enables targeted *backdoor* attacks which are designed to take advantage of heavy-tailed data (Bagdasaryan et al., 2020; Bhagoji et al., 2018). However, this is a challenging and open problem (Sun et al., 2019; Wang et al., 2020). Our focus is on the overall accuracy of the trained model, not on any subproblem.

Non-IID defenses. The non-iid defenses are relatively under-examined. Ghosh et al. (2019); Sattler et al. (2020) use an outlier-robust clustering method. When the server has the entire training dataset, Xie et al. (2019); Chen et al. (2018); Rajput et al. (2019) show that non-iid-ness can be overcome. Typical examples are parallel training of neural networks on public cloud, or volunteer computing (Meeds et al., 2015; Miura & Harada, 2015). However, none of these methods are applicable to the standard federated learning. This is partially tackled in (Data & Diggavi, 2020; 2021) who analyze spectral methods for robust optimization. However, these methods require $\Omega(d^2)$ time, making them infeasible for large scale optimization. Li et al. (2019) proposes an SGD variant (RSA) with additional ℓ_p penalty which only works for strongly convex objectives. In an independent recent work, Acharya et al. (2021) analyze geometric median (GM) on non-iid data using sparsified gradients. However, they do not defend against time coupled attacks, and their analysis neither proves convergence to the optimum nor recovers the standard rate of SGD when $\delta \to 0$. In contrast, our analysis of GM addresses both issues and is more general. Finally, Yang & Li (2021) propose to use bucketing for asynchronous Byzantine learning which is very similar to the bucketing trick proposed in this paper for non-iid setup. In Appendix F.1 we further compare these two methods in terms of purposes, techniques, and analysis. [1]

Strong growth condition. The assumption that $\mathbb{E}_{j\sim\mathcal{G}}\|\nabla f_j(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2 \leq B^2\|\nabla f(\boldsymbol{x})\|^2$ for some $B \geq 0$ is also referred to as the strong growth condition (Schmidt & Roux, 2013). This has been extensively used to analyze and derive optimization algorithms for deep learning (Schmidt & Roux, 2013; Ma et al., 2018; Vaswani et al., 2019a;b; Meng et al., 2020). This line of work shows that the strong growth assumption is both realistic and (perhaps more importantly) useful in understanding optimization algorithms in deep learning. However, this is stronger than the *weak* growth condition which states that $\mathbb{E}_{j\sim\mathcal{G}}\|\nabla f_j(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2 \leq B^2(f(\boldsymbol{x}) - f^\star)$ for some $B \geq 0$. For a smooth function f, the strong growth condition always implies the weak growth condition. Further, for smooth convex functions this is equivalent to assuming that all the workers functions $\{f_i\}$ share a common optimum, commonly known as interpolation. Our work uses the stronger version of the growth condition and it remains open to extend our results to the weaker version. This latter condition is strictly necessary for heterogeneous Byzantine optimization (Gupta & Vaidya, 2020).

3 Attacks against existing aggregation schemes

In this section we show that when the data across the workers is heterogeneous (non-iid), then we can design simple new attacks which take advantage of the heterogeneity, leading to the failure of existing aggregation schemes. We study three representative and widely used defenses:

Krum. For $i \neq j$, let $i \to j$ denote that $\boldsymbol{x}_j$ belongs to the $n-q-2$ closest vectors to $\boldsymbol{x}_i$. Then,

$$\textsc{Krum}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) := \operatorname*{argmin}_i \textstyle\sum_{i\to j} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 .$$

Krum is computationally expensive, requiring $\mathcal{O}(n^2)$ work by the server (Blanchard et al., 2017).
CM. Coordinate-wise median computes for the kth coordinate:

$$[\mathrm{CM}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)]_k := \mathrm{median}([\boldsymbol{x}_1]_k, \ldots, [\boldsymbol{x}_n]_k) = \operatorname*{argmin}_i \textstyle\sum_{j=1}^n |[\boldsymbol{x}_i]_k - [\boldsymbol{x}_j]_k| .$$

Coordinate-wise median is fast to implement requiring only $\mathcal{O}(n)$ time (Chen et al., 2017).

RFA. Robust federated averaging (RFA) computes the geometric median

$$\mathrm{RFA}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) := \operatorname*{argmin}_{\boldsymbol{v}} \textstyle\sum_{i=1}^n \|\boldsymbol{v} - \boldsymbol{x}_i\|_2 .$$

While the geometric median has no closed form solution, (Pillutla et al., 2019) approximate it using multiple iterations of smoothed Weiszfeld algorithm, each of which requires $\mathcal{O}(n)$ computation.

3.1 Failure on imbalanced data without Byzantine workers

We show that when the data amongst the workers is imbalanced, existing aggregation rules *fail* even in the *absence* of any Byzantine workers. Algorithms like Krum select workers who are *representative* of a majority of the workers by relying on statistics such as pairwise differences between the various worker updates. Our key insight is that when the data across the workers is heterogeneous, there is no

[1] The previous version of this work uses resampling which has identical performance as bucketing. The detailed comparison is listed in Appendix A.2.4.

Table 1: Test accuracy (%) with no Byzantine workers ($\delta = 0$) on imbalanced data.

Aggr	iid	non-iid
AVG	98.79±0.10	98.75±0.02
KRUM	97.95±0.25	89.90±4.75
CM	97.72±0.22	80.36±0.05
RFA	98.62±0.08	82.60±0.84
CCLIP	98.78±0.10	98.78±0.06

Table 2: Test accuracy (%) under mimic attack with $\delta = 0.2$ fraction of Byzantine workers.

Aggr	iid	non-iid
AVG	93.20±0.21	92.73±0.32
KRUM	90.36±0.25	37.33±6.78
CM	90.80±0.12	64.27±3.70
RFA	92.92±0.25	78.93±9.27
CCLIP	93.16±0.22	91.53±0.06

single worker who is representative of the whole dataset. This is because each worker computes their local gradient over vastly different local data.

Example. Suppose that there are $2n + 1$ workers with worker i holding $(-1)^i \in \{\pm 1\}$. This means that the true mean is ≈ 0, but KRUM, CM, and RFA will output ± 1. This motivates our next attack.

Hence, for convergence it is important to not only select a good (non-Byzantine) worker, but also ensure that each of the good workers is selected with roughly equal frequency. In Table 1, we demonstrate failures of such aggregators by training on MNIST with $n = 20$ and no attackers ($\delta = 0$). We construct an imbalanced dataset where each successive class has only a fraction of samples of the previous class. We defer details of the experiments to Appendix A. As we can see, KRUM, CM and RFA match the ideal performance of SGD in the iid case, but only attain less than 90% accuracy in the non-iid case. This corresponds to learning only the top 2–3 classes and ignoring the rest.

A similar phenomenon was observed when using batch-size 1 in the iid case by (Karimireddy et al., 2021). However, in the iid case this can be easily overcome by increasing the batch-size. In contrast, when the data across the works is non-iid (e.g. split by class), increasing the batch-size does *not* make the worker gradients any more similar and there remains a big drop in performance. Finally, note that in Table 1 a hitherto new algorithm (CCLIP) maintains its performance both in the iid and the non-iid setting. We will explore this in more detail in Section 4.

3.2 MIMIC ATTACK ON BALANCED DATA

Motivated by how data imbalance could lead to consistent errors in the aggregation rules and significant loss in accuracy, in this section, we will propose a new attack *mimic* which specifically tries to maximize the perceived data imbalance even if the original data is balanced.

Mimic attack. All Byzantine workers pick a good worker (say $i_\star$) to mimic and copy its output ($\boldsymbol{x}_{i_\star}^t$). This inserts a consistent bias towards over-emphasizing worker $i_\star$ and thus under-representing other workers. Since the attacker simply mimics a good worker, it is impossible to distinguish it from a real worker and hence it cannot be filtered out. Indeed, the target $i_\star$ can be any fixed good worker. In Appendix B, we present an empirical rule to choose $i_\star$ and include a simple example demonstrating how median based aggregators suffer from the heterogeneity under mimic attack.

Table 2 shows the effectiveness of mimic attack even when the fraction of Byzantine nodes is small (i.e. $n = 25$, $|\mathcal{B}| = 5$). Note that this attack specifically targets the non-iid nature of the data—all robust aggregators maintain their performance in the iid setting and only suffer in the non-iid setting. Their performance is in fact worse than even simply averaging. As predicted by our example, KRUM and CM have the worst performance and RFA performs slightly better. We will discuss the remarkable performance of CCLIP in the next section.

4 CONSTRUCTING AN AGNOSTIC ROBUST AGGREGATOR USING BUCKETING

In Section 3 we demonstrated how existing aggregation rules fail in realistic non-iid scenarios, with and without attackers. In this section, we show how using bucketing can provably fix such aggregation rules. The underlying reason for this failure, as we saw previously, is that the existing methods fixate on the contribution of only the most likely worker, and ignore the contributions from the rest. To overcome this issue, we propose to use bucketing which 'mixes' the data from all the workers thereby reducing the chance of any subset of the data being consistently ignored.

Algorithm 1 Robust Aggregation (ARAGG) using bucketing

1: **input** $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, $s \in \mathbb{N}$, aggregation rule AGGR
2: pick random permutation π of $[n]$
3: compute $\boldsymbol{y}_i \leftarrow \frac{1}{s}\sum_{k=(i-1)\cdot s+1}^{\min(n\,,\,i\cdot s)} \boldsymbol{x}_{\pi(k)}$ for $i = \{1, \ldots, \lceil n/s \rceil\}$
4: **output** $\hat{\boldsymbol{x}} \leftarrow \text{AGGR}(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{\lceil n/s \rceil})$ // aggregate after bucketing

4.1 BUCKETING ALGORITHM

Given n inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, we perform *$s$-bucketing* which randomly partitions them into $\lceil n/s \rceil$ buckets with each bucket having no more than s elements. Then, the contents of each bucket are averaged to construct $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{\lceil n/s \rceil}\}$ which are then input to an aggregator AGGR. The details are summarized in Algorithm 1. The key property of our approach is that after bucketing, the resulting set of averaged $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{\lceil n/s \rceil}\}$ are much more homogeneous (lower variance) than the original inputs. Thus, when fed into existing aggregation schemes, the chance of success increases. We formalize this in the following simple lemma.

Lemma 1 (Bucketing reduces variance)**.** *Suppose we are given n independent (but not identical) random vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ such that a good subset $\mathcal{G} \subseteq [n]$ of size at least $|\mathcal{G}| \geq n(1-\delta)$ satisfies:*

$$\mathbb{E}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \leq \rho^2\,, \quad \text{for any fixed } i, j \in \mathcal{G}\,.$$

Define $\bar{\boldsymbol{x}} := \frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} \boldsymbol{x}_j$. Let the outputs after s-bucketing be $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{\lceil n/s \rceil}\}$ and denote $\tilde{\mathcal{G}} \subseteq \{1, \ldots, \lceil n/s \rceil\}$ as a good bucket set where a good bucket contains only elements belonging to $\mathcal{G}$. Then $|\tilde{\mathcal{G}}| \geq \lceil n/s \rceil(1-\delta s)$ satisfies

$$\mathbb{E}[\boldsymbol{y}_i] = \mathbb{E}[\bar{\boldsymbol{x}}] \quad \text{and} \quad \mathbb{E}\|\boldsymbol{y}_i - \boldsymbol{y}_j\| \leq \rho^2/s \quad \text{for any fixed } i, j \in \tilde{\mathcal{G}}\,.$$

The expectation in the above lemma is taken both over the random vectors as well as over the randomness of the bucketing procedure.

Remark 2. *Lemma 1 proves that after our bucketing procedure, we are left with outputs $\boldsymbol{y}_i$ which have i) pairwise variance reduced by s, and ii) potentially s times more fraction of Byzantine vectors. Hence, bucketing trades off increasing influence of Byzantine inputs against having more homogeneous vectors. Using $s = 1$ simply shuffles the inputs and leaves them otherwise unchanged.*

4.2 AGNOSTIC ROBUST AGGREGATION

We now define what it means for an agnostic robust aggregator to succeed.

Definition A ($(\delta_{\max}, c)$-ARAGG)**.** *Suppose we are given input $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ of which a subset $\mathcal{G}$ of size at least $|\mathcal{G}| > (1-\delta)n$ for $\delta \leq \delta_{\max} < 0.5$ and satisfies $\mathbb{E}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \leq \rho^2$. Then, the output $\hat{\boldsymbol{x}}$ of a Byzantine robust aggregator satisfies:*

$$\mathbb{E}\|\hat{\boldsymbol{x}} - \bar{\boldsymbol{x}}\|^2 \leq c\delta\rho^2 \quad \text{where} \quad \hat{\boldsymbol{x}} = \text{ARAGG}_\delta(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)\,.$$

Further, ARAGG *does not need to know ρ^2 (only δ), and automatically adapts to any value ρ^2.*

Our robust aggregator is parameterized by $\delta_{\max} < 0.5$ which denotes the maximum amount of Byzantine inputs it can handle, and a constant c which determines its performance. If $\delta = 0$, i.e. when there are no Byzantine inputs, we are guaranteed to *exactly* recover the true average $\bar{\boldsymbol{x}}$. Exact recovery is also guaranteed when $\rho = 0$ since in that case it is easy to identify the good inputs since they are all equal and in majority. When both $\rho > 0$ and $\delta > 0$, we recover the average up to an additive error term. We also require that the robust aggregator is *agnostic* to the value of ρ^2 and automatically adjusts its output to the current ρ during training. The aggregator can take δ as an input though. This property is very useful in the context of Byzantine robust optimization since the variance ρ^2 keeps changing over the training period, whereas the fraction of Byzantine workers δ remains constant. This is a major difference from the definition used in (Karimireddy et al., 2021). Note that Definition A is defined for both homogeneous and heterogeneous data.

We next show that aggregators which we saw were not robust in Section 3, can be made to satisfy Definition A by combining with bucketing.

Theorem I. *Suppose we are given n inputs $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ satisfying properties in Lemma 1 for some $\delta \leq \delta_{\max}$, with $\delta_{\max}$ to be defined. Then, running Algorithm 1 with $s = \lfloor \delta_{\max}/\delta \rfloor$ yields the following:*

- *Krum:* $\mathbb{E}\|\text{KRUM} \circ \text{BUCKETING}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) - \bar{\boldsymbol{x}}\|^2 \leq \mathcal{O}(\delta\rho^2)$ *with* $\delta_{\max} < 1/4$.
- *Geometric median:* $\mathbb{E}\|\text{RFA} \circ \text{BUCKETING}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) - \bar{\boldsymbol{x}}\|^2 \leq \mathcal{O}(\delta\rho^2)$ *with* $\delta_{\max} < 1/2$.
- *Coordinate-wise median:* $\mathbb{E}\|\text{CM} \circ \text{BUCKETING}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) - \bar{\boldsymbol{x}}\|^2 \leq \mathcal{O}(d\delta\rho^2)$ *with* $\delta_{\max} < 1/2$.

Note that all these methods satisfy our notion of an *agnostic* Byzantine robust aggregator (Definition A). This is because both our bucketing procedures as well as the underlying aggregators are independent of ρ^2. Further, our error is $\mathcal{O}(\delta\rho^2)$ and is information theoretically optimal, unlike previous analyses (e.g. Acharya et al. (2021)) who had an error of $\mathcal{O}(\rho^2)$.

The error of CM depends on the dimension d which is problematic when $d \gg n$. However, we suspect this is because we measure stochasticity using Euclidean norms instead of coordinate-wise. In practice, we found that CM often outperforms KRUM, with RFA outperforming them both. Note that we select $s = \lfloor \delta_{\max}/\delta \rfloor$ to ensure that after bucketing, we have the maximum amount of Byzantine inputs tolerated by the method with $(s\delta) = \delta_{\max}$.

Remark 3 (1-step Centered clipping)**.** *The 1-step centered clipping aggregator (*CCLIP*) given a clipping radius τ and an initial guess $\boldsymbol{v}$ of the average $\bar{\boldsymbol{x}}$ performs*

$$\text{CCLIP}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \boldsymbol{v} + \frac{1}{n}\sum_{i\in[n]} (\boldsymbol{x}_n - \boldsymbol{v}) \min(1\,,\, \tau/\|\boldsymbol{x}_n - \boldsymbol{v}\|_2)\,.$$

Karimireddy et al. (2021) prove that CCLIP *even without bucketing satisfies Definition A with $\delta_{\max} = 0.1$, and $c = \mathcal{O}(1)$. This explains its good performance on non-iid data in Section 3. However,* CCLIP *is* not agnostic *since it requires clipping radius τ as an input which in turn depends on ρ^2. Devising a version of* CCLIP *which automatically adapts its clipping radius is an important open question. Empirically however, we observe that simple rules for setting τ work quite well—we always use $\tau = \frac{10}{1-\beta}$ in our limited experiments where β is the coefficient of momentum.*

While we have shown how to construct a robust aggregator which satisfies some notion of a robustness, we haven't yet seen how this affects the Byzantine robust *optimization* problem. We investigate this question theoretically in the next section and empirically in Section 6.

5 ROBUST NON-IID OPTIMIZATION USING A ROBUST AGGREGATOR

In this section, we study the problem of optimization in the presence of Byzantine workers and heterogeneity, given access to any robust aggregator satisfying Definition A. We then show that data heterogeneity makes Byzantine robust optimization especially challenging and prove lower bounds for the same. Finally, we see how mild heterogeneity, or sufficient overparameterization can circumvent these lower bounds, obtaining convergence to the optimum.

Algorithm 2 Robust Optimization using any Agnostic Robust Aggregator

Input: ARAGG, η, β
1: **for** $t = 1, \ldots$ **do**
2: **for** worker $i \in [n]$ **in parallel**
3: $\boldsymbol{g}_i \leftarrow \nabla F_i(\boldsymbol{x}, \boldsymbol{\xi}_i)$ and $\boldsymbol{m}_i \leftarrow (1-\beta)\boldsymbol{g}_i + \beta\boldsymbol{m}_i$ // worker momentum
4: **send** $\boldsymbol{m}_i$ if $i \in \mathcal{G}$, else send $*$ if Byzantine
5: $\hat{\boldsymbol{m}} =$ ARAGG $(\boldsymbol{m}_1, \ldots, \boldsymbol{m}_n)$ and $\boldsymbol{x} \leftarrow \boldsymbol{x} - \eta\hat{\boldsymbol{m}}$. // update params using robust aggregate

5.1 ALGORITHM DESCRIPTION

In Section 4 we saw that bucketing could tackle heterogeneity across the workers by reducing ζ^2. However, there still remains variance σ^2 in the gradients within each worker since each worker uses stochastic gradients. To reduce the effect of this variance, we rely on worker momentum. Each worker sends their local worker momentum vector $\boldsymbol{m}_i$ to be aggregated by ARAGG instead of $\boldsymbol{g}_i$ directly:

$$\boldsymbol{m}_i^t = \beta\boldsymbol{m}_i^{t-1} + (1-\beta)\boldsymbol{g}_i(\boldsymbol{x}^{t-1}) \quad \text{for every } i \in \mathcal{G}\,,$$
$$\boldsymbol{x}^t = \boldsymbol{x}^{t-1} - \eta\text{ARAGG}(\boldsymbol{m}_1^t, \ldots, \boldsymbol{m}_n^t)\,.$$

This is equivalent to the usual momentum description up to a rescaling of step-size η. Intuitively, using worker momentum $\boldsymbol{m}_i$ averages over $1/(1-\beta)$ independent stochastic gradients $\boldsymbol{g}_i$ and thus reduces the effect of the within-worker-variance σ^2 (Karimireddy et al., 2021). Note that the resulting $\{\boldsymbol{m}_i\}$ are *still heterogeneous* across the workers. This heterogeneity is the key challenge we face.

5.2 CONVERGENCE RATES

We now turn towards proving convergence rates for our bucketing aggregation method Algorithm 1 based on any existing aggregator AGGR. We will assume that for any fixed $i \in \mathcal{G}$, the stochastic gradients computed satisfy

$$\mathbb{E}_{\boldsymbol{\xi}_i}\|\boldsymbol{g}_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x})\|^2 \leq \sigma^2 \text{ and } \mathbb{E}_{j\sim\mathcal{G}}\|\nabla f_j(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2 \leq \zeta^2\,, \quad \forall \boldsymbol{x}\,. \tag{2}$$

This first condition bounds the variance of the stochastic gradient within a worker whereas the latter is a standard measure of inter-client heterogeneity in federated learning (Yu et al., 2019; Khaled et al., 2020; Karimireddy et al., 2020). Under these conditions, we can prove the following.

Theorem II. *Suppose we are given a $(\delta_{\max}, c)$-ARAGG satisfying Definition A, and n workers of which a subset $\mathcal{G}$ of size at least $|\mathcal{G}| \geq n(1-\delta)$ faithfully follow the algorithm for $\delta \leq \delta_{\max}$. Further, for any good worker $i \in \mathcal{G}$ let f_i be a possibly non-convex function with L-Lipschitz gradients, and the stochastic gradients on each worker be independent, unbiased and satisfy (2). Then, for $F^0 := f(\boldsymbol{x}^0) - f^\star$, the output of Algorithm 2 satisfies*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 \leq \mathcal{O}\Big(c\delta\zeta^2 + \sigma\sqrt{\frac{LF^0}{T}(c\delta + {}^1\!/_n)} + \frac{LF^0}{T}\Big)\,.$$

Remark 4 (Unified proofs)**.** *Remark 3 shows that* CCLIP *is a robust aggregator, and Theorem I shows* KRUM, RFA, *and* CM *on combining with sufficient bucketing are all robust aggregators satisfying Definition A. Most of these methods had no end-to-end convergence guarantees prior to our results. Thus, Theorem II gives the first unified analysis in both the iid and non-iid settings.*

When $\delta \to 0$ i.e. as we reduce the number of Byzantine workers, the above rate recovers the familiar $\mathcal{O}(\frac{\sigma}{\sqrt{Tn}})$ rate which is optimal for non-convex SGD and even has linear speed-up with respect to the n workers. In contrast, all previous algorithms for non-iid data (e.g. (Data & Diggavi, 2021; Acharya et al., 2021)) do not improve their rates for decreasing values of δ. This is also empirically reflected in Section 3.1, where we showed how these algorithms can fail even in the absence of Byzantine workers ($\delta = 0$).

Further, in the iid case when $\zeta = 0$ the rate above simplifies to $\mathcal{O}(\frac{\sigma}{\sqrt{T}} \cdot \sqrt{c\delta + {}^1\!/_n})$ which matches the iid Byzantine robust rates of (Karimireddy et al., 2021). In both these cases we converge to the optimum and can make the gradient arbitrarily small. However, when $\delta > 0$ and $\zeta > 0$, Theorem II only shows convergence to a radius of $\mathcal{O}(\sqrt{\delta}\zeta)$ and not to the actual optimum. We will next explore this limitation.

5.3 LOWER BOUNDS AND THE CHALLENGE OF HETEROGENEITY

Suppose worker j sends us an update which looks 'weird' and is very different from the updates from the rest of the workers. This may be because worker j might be malicious and their update represents an attempted attack. It may also be because worker j has highly *non-representative data*. In the former case the update should be ignored, whereas in the latter the update represents a valuable source of specialized data. However, it is impossible for the server to distinguish between the two situations. The above argument can in fact be formalized to prove the following lower bound.

Theorem III. *Given any optimization algorithm* ALG, *we can find n functions $\{f_1(x), \ldots, f_n(x)\}$ of which at least $(1-\delta)n$ are good (belong to $\mathcal{G}$), 1-smooth, μ-strongly convex functions, and satisfy $\mathbb{E}_{i\sim\mathcal{G}}\|\nabla f_i(x) - \nabla f(x)\|^2 \leq \zeta^2$ such that the output of* ALG *given access to these n function has an error at least*

$$\mathbb{E}[f(\textsc{Alg}(f_1, \ldots, f_n)) - f^\star] \geq \Omega\Big(\frac{\delta\zeta^2}{\mu}\Big) \quad \textit{and} \quad \mathbb{E}\|\nabla f(\textsc{Alg}(f_1, \ldots, f_n))\|^2 \geq \Omega\big(\delta\zeta^2\big)\,.$$

The expectation above is over the potential randomness of the algorithm. This theorem unfrotunately implies that it is impossible to converge to the true optimum in the presence of Byzantine workers. Note that the above lower bound is information theoretic in nature and is independent of how many gradients are computed or how long the algorithm is run.

Remark 5 (Matches lower bound)**.** *Suppose that we satisfy the heterogeneity condition (2) with $\zeta^2 > 0$ and $\sigma = 0$. Then, the rate in Theorem II can be simplified to $\mathcal{O}\big(\delta\zeta^2 + 1/T\big)$. While the*

Table 3: Table 1 + Bucketing (s=2).

Aggr	iid	non-iid
AVG	98.80±0.10	98.74±0.02
KRUM	98.35±0.20	93.27±0.10
CM	98.26±0.22	95.59±0.89
RFA	98.75±0.14	97.34±0.58
CCLIP	98.79±0.10	98.75±0.02

Table 4: Table 2 + Bucketing (s=2).

Aggr	iid	non-iid
AVG	93.17±0.23	92.67±0.27
KRUM	91.64±0.30	53.15±3.96
CM	91.91±0.24	78.60±3.15
RFA	93.00±0.23	91.17±0.51
CCLIP	93.17±0.23	92.56±0.21

second term in this decays to 0 with T, the first term remains, implying that we only converge to a radius of $\sqrt{\delta}\zeta$ around the optimum. However, this matches our lower bound result from Theorem III and hence is in general unimprovable.

This is a very strong negative result and seems to indicate that Byzantine robustness might be impossible to achieve in real world federated learning. This would be major stumbling block for deployment since the system would provably be vulnerable to attackers. We will next carefully examine the lower bound and will attempt to circumvent it.

5.4 CIRCUMVENTING LOWER BOUNDS USING OVERPARAMETERIZATION

We previously saw some strong impossibility results posed by heterogeneity. In this section, we show that while indeed in the worst case being robust under heterogeneity is impossible, we may still converge to the true optimum under more realistic settings. Let us consider a different heterogeneity bound in place of (2):

$$\mathbb{E}_{j\sim\mathcal{G}}\|\nabla f_j(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2 \leq B^2\|\nabla f(\boldsymbol{x})\|^2\,, \quad \forall \boldsymbol{x}\,. \tag{3}$$

Note that at the optimum $\boldsymbol{x}^\star$ we have $\nabla f(\boldsymbol{x}^\star) = 0$, and hence this assumption implies that $\nabla f_j(\boldsymbol{x}^\star) = 0$ for all $j \in \mathcal{G}$. This is satisfied if the model is *sufficiently over-parameterized* and typically holds in most realistic settings (Vaswani et al., 2019a).

Theorem IV. *Suppose we are given a $(\delta_{\max}, c)$-ARAGG and n workers with loss functions $\{f_1, \ldots, f_n\}$ satisfying the conditions in Theorem II with $\delta \leq \delta_{\max}$ and (3) for some $B^2 < \frac{1}{60c\delta}$. Then, for $F^0 := f(\boldsymbol{x}^0) - f^\star$, the output of Algorithm 2 satisfies*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 \leq \mathcal{O}\Big(\frac{1}{1-60c\delta B^2}\cdot\Big(\sigma\sqrt{\frac{LF^0}{T}(c\delta + {}^1\!/_n)} + \frac{LF^0}{T}\Big)\Big)\,.$$

Remark 6 (Overparameterization fixes convergence)**.** *The rate in Theorem IV not only goes to 0 with T, but also matches that of the optimal iid rate of $\mathcal{O}(\frac{\sigma}{\sqrt{T}}\cdot\sqrt{c\delta + {}^1\!/_n})$ (Karimireddy et al., 2021). Thus, using a stronger heterogeneity assumption allows us to circumvent lower bounds for the non-iid case and converge to a good solution even in the presence of Byzantine workers. This is the first result of its kind, and takes a major step towards realistic and practical robust algorithms.*

In the overparameterized setting, we can be sure that we will able to *simultaneously* optimize all worker's losses. Hence, over time the agreement between all worker's gradients increases. This in turn makes any attempts by the attackers to derail training stand out easily, especially towards the end of the training. To take advantage of this increasing closeness, we need an aggregator which automatically adapts the quality of its output as the good workers get closer. Thus, the *agnostic* robust aggregator is crucial to our overparameterized convergence result. We empirically demonstrate the effects of overparameterization in Appendix A.2.3.

6 EXPERIMENTS

In this section, we demonstrate the effects of bucketing on datasets distributed in a non-iid fashion. Throughout the section, we illustrate the tasks, attacks, and defenses by an example of training an MLP on a heterogeneous version of the MNIST dataset (LeCun et al., 1998). The dataset is sorted by labels and sequentially divided into equal parts among good workers; Byzantine workers have access to the entire dataset. Implementations are based on PyTorch (Paszke et al., 2019) and will be made publicly available.[2] We defer details of setup, implementation, and runtime to Appendix A.

[2]The code is available at this url.

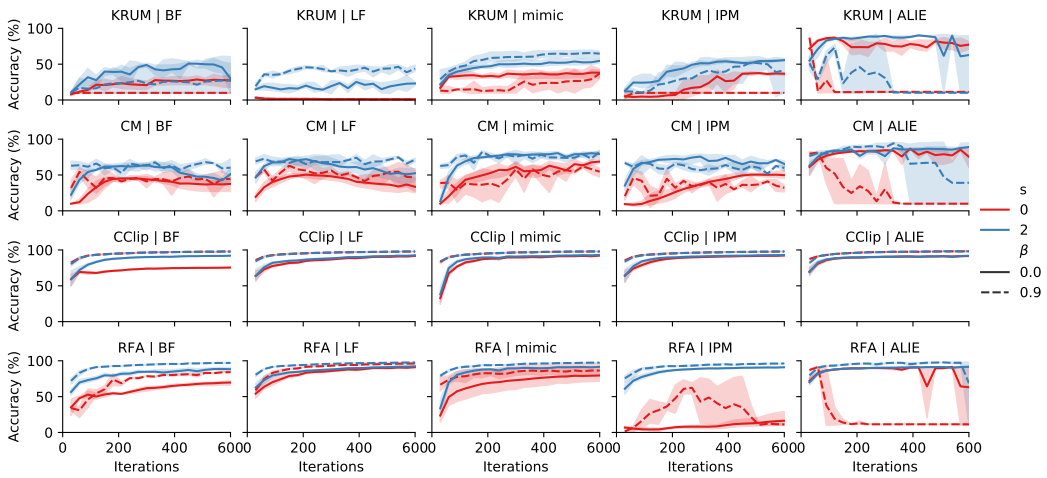


Figure 1: Top-1 test accuracies of KRUM, CM, CCLIP, RFA, under 5 attacks on non-iid datasets.

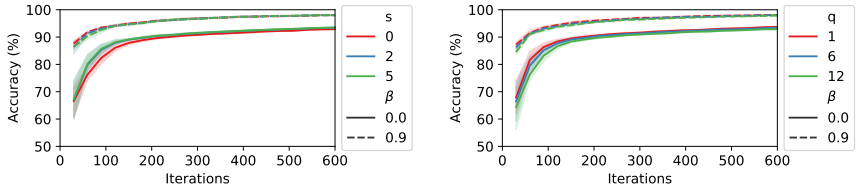
Bucketing against the attacks on non-iid data. In Section 3 we have presented how heterogeneous data can lead to failure of existing robust aggregation rules. Here we apply our proposed bucketing with $s = 2$ to the same aggregation rules, showing that bucketing overcomes the described failures. Results are presented in Table 3. Comparing Table 3 with Table 1, bucketing improves the aggregators' top-1 test accuracy on long-tail and non-iid dataset by 4% to 14% and allows them to learn classes at the tail distribution. For non-iid balanced dataset, bucketing also greatly improves the performance of KRUM and CM and makes RFA and CCLIP close to ideal performance. Similarly, combining aggregators with bucketing also performs much better on non-iid dataset under mimic attack. In Table 4, RFA and CCLIP reach accuracy similar to the iid case, and the performance of KRUM, and CM are improved by around 15%.

Bucketing against general Byzantine attacks. In Figure 1, we present thorough experiments on non-iid data over 25 workers with 5 Byzantine workers, under different attacks. In each subfigure, we compare an aggregation rule with its variant with bucketing. The aggregation rules compared are KRUM, CM, RFA, CCLIP. 5 different kinds of attacks are applied (one per column in the figure): bit flipping (BF), label flipping (LF), *mimic* attack, as well as inner product manipulation (IPM) attack (Xie et al., 2020) and the "a little is enough" (ALIE) attack (Baruch et al., 2019).

- **Bit flipping**: A Byzantine worker sends $-\nabla f(\boldsymbol{x})$ instead of $\nabla f(\boldsymbol{x})$ due to hardware failures etc.
- **Label flipping**: Corrupt MNIST dataset by transforming labels by $\mathcal{T}(y) := 9 - y$.
- **Mimic**: Explained in Section 3.2.
- **IPM**: The attackers send $-\frac{\epsilon}{|\mathcal{G}|}\sum_{i\in\mathcal{G}} \nabla f(\boldsymbol{x}_i)$ where ϵ controls the strength of the attack.
- **ALIE**: The attackers estimate the mean $\mu_{\mathcal{G}}$ and standard deviation $\sigma_{\mathcal{G}}$ of the good gradients, and send $\mu_{\mathcal{G}} - z\sigma_{\mathcal{G}}$ to the server where z is a small constant controlling the strength of the attack.

Both IPM and ALIE are the state-of-the-art attacks in the iid distributed learning setups which takes advantage of the variances among workers. These attacks are much stronger in the non-iid setup. In the last two columns of Figure 1 we show that worker momentum and bucketing reduce such variance while momentum alone is not enough. Overall, Figure 1 shows that bucketing improves the performances of almost all aggregators under all kinds of attacks. Note that τ of CCLIP is not finetuned for each attack but rather fixed to $\frac{10}{1-\beta}$ for all attacks. This scaling is required because CCLIP is *not agnostic*. We defer the discussion to Appendix A.2.1.

Bucketing hyperparameter. Finally we study the influence of s and q on the heterogeneous MNIST dataset. We use CCLIP as the base aggregator and apply IPM attack. The Figure 2a confirms that larger s gives faster convergence but $s = 2$ can be good enough. The results in Figure 2b shows that $s = 2$ still behaves well when increasing q close to 25%. The complete evaluation of the results are deferred to Appendix A.



(a) Fixed $q = 5$ IPM attackers, varying s

(b) Fixed $s = 2$, varying number q

Figure 2: Top-1 accuracies of CCLIP with varying q and s when training on a cluster of $n = 53$ nodes

Discussion. In all our experiments, we consistently observe: i) mild bucketing ($s = 2$) improves performance, ii) worker momentum further stabilizes training, and finally iii) CCLIP recovers the ideal performance. Given its ease of implementation, this leads us to strongly recommend using CCLIP in practical federated learning to safeguard against actively malicious agents or passive failures. RFA combined with bucketing and worker momentum also nearly recovers ideal performance and can instead be used when a proper radius τ is hard to find. Ideally, one would want to *adaptively* and automatically set τ so that it works in all instances without any tuning. Designing such a clipping operator as well as its large scale empirical study is left for future work.

7 Conclusion

Heterogeneity poses unique challenges for Byzantine robust optimization. The first challenge is that existing defenses attempt to pick a "representative" update, which may not exist in the non-iid setting. This, we showed, can be overcome by using bucketing. A second more fundamental challenge is that it is difficult to distinguish between a "weird" but good worker from an actually Byzantine attacker. In fact, we proved strong impossibility results in such a setting. For this we showed how overparameterization (which is prevalent in real world deep learning) provides a solution, ensuring convergence to the optimum even in the presence of attackers. Together, our results yield a practical provably Byzantine robust algorithms for the non-iid setting.

Acknowledgement. This project was supported by SNSF grant 200020_200342.

References

Anish Acharya, Abolfazl Hashemi, Prateek Jain, Sujay Sanghavi, Inderjit S Dhillon, and Ufuk Topcu. Robust training in high dimensions via block coordinate geometric median descent. *arXiv preprint arXiv:2106.08882*, 2021.

Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine stochastic gradient descent. In *NeurIPS - Advances in Neural Information Processing Systems*, pp. 4613–4623, 2018.

Zeyuan Allen-Zhu, Faeze Ebrahimian, Jerry Li, and Dan Alistarh. Byzantine-resilient non-convex stochastic gradient descent. In *ICLR 2021 - International Conference of Learning Representations*, 2021.

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *AISTATS - Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pp. 2938–2948, 2020.

Moran Baruch, Gilad Baruch, and Yoav Goldberg. A Little Is Enough: Circumventing Defenses For Distributed Learning. *NeurIPS*, 2019.

Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. signSGD with majority vote is communication efficient and fault tolerant. *arXiv preprint arXiv:1810.05291*, 2018.

Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens, 2018.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *NeurIPS - Advances in Neural Information Processing Systems 30*, pp. 119–129, 2017.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. In *SysML - Proceedings of the 2nd SysML Conference, Palo Alto, CA, USA*, 2019.

Lingjiao Chen, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos. Draco: Byzantine-resilient distributed training via redundant gradients. *arXiv preprint arXiv:1803.09877*, 2018.

Yudong Chen, Lili Su, and Jiaming Xu. 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.

Michael B Cohen, Yin Tat Lee, Gary Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 9–21, 2016.

Georgios Damaskinos, El Mahdi El Mhamdi, Rachid Guerraoui, Arsany Hany Abdelmessih Guirguis, and Sébastien Louis Alexandre Rouault. Aggregathor: Byzantine machine learning via robust gradient aggregation. *Conference on Systems and Machine Learning (SysML) 2019, Stanford, CA, USA*, pp. 19, 2019.

Deepesh Data and Suhas Diggavi. Byzantine-resilient sgd in high dimensions on heterogeneous data. *arXiv preprint arXiv:2005.07866*, 2020.

Deepesh Data and Suhas Diggavi. Byzantine-resilient high-dimensional sgd with local iterations on heterogeneous data. In *ICML 2021 - 37th International Conference on Machine Learning*, 2021.

Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*, 2019.

Nirupam Gupta and Nitin H Vaidya. Resilience in collaborative optimization: redundant and independent cost functions. *arXiv preprint arXiv:2003.09675*, 2020.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian U Stich, and Martin Jaggi. Error Feedback Fixes SignSGD and other Gradient Compression Schemes. In *ICML 2019 - Proceedings of the 36th International Conference on Machine Learning*, 2019.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *ICML 2020 - International Conference on Machine Learning*, 2020.

Sai Praneeth Karimireddy, Li He, and Martin Jaggi. Learning from history for byzantine robust optimization. In *ICML 2021 - 37th International Conference on Machine Learning*, 2021.

Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *AISTATS 2020 - International Conference on Artificial Intelligence and Statistics*, 2020.

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

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Liping Li, Wei Xu, Tianyi Chen, Georgios B Giannakis, and Qing Ling. RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1544–1551, 2019.

Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of sgd in modern over-parametrized learning. In *International Conference on Machine Learning*, pp. 3325–3334. PMLR, 2018.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

Edward Meeds, Remco Hendriks, Said Al Faraby, Magiel Bruntink, and Max Welling. Mlitb: machine learning in the browser. *PeerJ Computer Science*, 1:e11, Jul 2015. ISSN 2376-5992. doi: 10.7717/peerj-cs.11. URL http://dx.doi.org/10.7717/peerj-cs.11.

Si Yi Meng, Sharan Vaswani, Issam Hadj Laradji, Mark Schmidt, and Simon Lacoste-Julien. Fast and furious convergence: Stochastic second order methods under interpolation. In *International Conference on Artificial Intelligence and Statistics*, pp. 1375–1386. PMLR, 2020.

El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in byzantium. *arXiv preprint arXiv:1802.07927*, 2018.

El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. Distributed momentum for byzantine-resilient stochastic gradient descent. In *ICLR 2021 - International Conference on Learning Representations*, 2021.

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

Ken Miura and Tatsuya Harada. Implementation of a practical distributed calculation system with browsers and javascript, and application to distributed deep learning. *arXiv preprint arXiv:1503.05743*, 2015.

Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.

Krishna Pillutla, Sham M. Kakade, and Zaid Harchaoui. Robust Aggregation for Federated Learning. *arXiv preprint arXiv:1912.13445*, 2019.

Shashank Rajput, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos. Detox: A redundancy-based framework for faster and more robust gradient aggregation. *arXiv preprint arXiv:1907.12205*, 2019.

F. Sattler, K. Müller, T. Wiegand, and W. Samek. On the byzantine robustness of clustered federated learning. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8861–8865, 2020.

Mark Schmidt and Nicolas Le Roux. Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint arXiv:1308.6370*, 2013.

Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019.

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of sgd for over-parameterized models and an accelerated perceptron. In *AISTATS 2019 - The 22nd International Conference on Artificial Intelligence and Statistics*, 2019a.

Sharan Vaswani, Aaron Mishkin, Issam Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien. Painless stochastic gradient: Interpolation, line-search, and convergence rates. *Advances in neural information processing systems*, 32, 2019b.

Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *arXiv preprint arXiv:2007.05084*, 2020.

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

Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *ICML 2019 - 35th International Conference on Machine Learning*, 2019.

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

Yi-Rui Yang and Wu-Jun Li. Basgd: Buffered asynchronous sgd for byzantine learning. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11751–11761. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/yang21e.html.

Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. *arXiv preprint arXiv:1803.01498*, 2018.

Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *AAAI 2019 - Conference on Artificial Intelligence*, 2019.

A EXPERIMENT SETUP AND ADDITIONAL EXPERIMENTS

A.1 EXPERIMENT SETUP

A.1.1 GENERAL SETUP

The default experiment setup is listed in Table 5.

Table 5: Default experimental settings for MNIST

Dataset	MNIST
Architecture	CONV-CONV-DROPOUT-FC-DROPOUT-FC
Training objective	Negative log likelihood loss
Evaluation objective	Top-1 accuracy
Batch size	$32 \times$ number of workers
Momentum	0 or 0.9
Learning rate	0.01
LR decay	No
LR warmup	No
# Iterations	600 or 4500
Weight decay	No
Repetitions	3, with varying seeds
Reported metric	Mean test accuracy over the last 150 iterations

By default the hyperparameters of the aggregators are summarized as follows

Aggregators	Hyperparameters
KRUM	N/A
CM	N/A
RFA	$T = 8$
TM	$b = q$
CCLIP	$\tau = \frac{10}{1-\beta}$

A.1.2 CONSTRUCTING DATASETS

The MNIST dataset has 10 classes each with similar amount of samples. In this part, we discuss how to process and distribute MNIST to each workers in order to achieve long-tailness and heterogeneity.

Long-tailness. The long-tailness (*-LT) is achieved by sampling class with exponentially decreasing portions $\gamma \in (0, 1]$. That is, for class $i \in [10]$, we only randomly sample γ^i portion of all samples in class i. We define α as the ratio of the largest class over the smallest class, which can be written as $\alpha = \frac{1}{\gamma^9}$. For example, if $\gamma = 1$, then all classes have same amount of samples and thus $\alpha = 1$; if $\gamma = 0.5$ then $\alpha = 2^9 = 512$. Note that the same procedure has to be applied to the test dataset.

Heterogeneity. Steps to construct IID/non-iid dataset from MNIST dataset

1. Sort the training dataset by its labels.
2. Evenly divide the sorted training dataset into chunks of same size. The number of chunks equals the number of good workers. If the last chunk has fewer samples, we augment it with samples from itself.
3. Shuffle the samples within the same worker.

Heterogeneity + Long-tailness. First transform the training dataset into long-tail dataset, then feed it to the previous procedure to introduce heterogeneity.

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.

A.1.3 SETUP FOR EACH EXPERIMENT

In Table 6, we list the hyperparameters for the experiments.

Table 6: Setups for each experiment.

	n	q	momentum	Iters	LT	NonIID
Table 1	24	0	0	4500	$\alpha = 1$, $\alpha = 500$	iid/ non-iid
Table 2	25	5	0	600	$\alpha = 1$ (balanced)	iid/ non-iid
Table 3	24	0	0	4500	$\alpha = 1$, $\alpha = 500$	iid/ non-iid
Table 4	25	5	0	600	$\alpha = 1$ (balanced)	iid/ non-iid
Figure 1	25	5	0 / 0.9	600	$\alpha = 1$ (balanced)	non-iid
Figure 2	53	5	0 / 0.9	600	$\alpha = 1$ (balanced)	non-iid
Figure 3	25	5	0 / 0.5 / 0.9 / 0.99	600	$\alpha = 1$ (balanced)	non-iid
Figure 4	25	5	0 / 0.5 / 0.9 / 0.99	1200	$\alpha = 1$ (balanced)	non-iid
Figure 5	20	3	0	1200	$\alpha = 1$ (balanced)	non-iid
Figure 6	20	3	0	3000	$\alpha = 1$ (balanced)	non-iid
Figure 8	24	3	0	1200	$\alpha = 1$ (balanced)	non-iid

IPM Attack in Figure 1 and Figure 2. We set the strength of the attack $\epsilon = 0.1$.

ALIE Attack in in Figure 1. The hyperparameter z for ALIE is computed according to (Baruch et al., 2019)

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

where $s = \lfloor \frac{n}{2} + 1 \rfloor - q$ and ϕ is the cumulative standard normal function. In our setup, the $z \approx 0.25$.

A.1.4 RUNNING ENVIRONMENT

We summarize the running environment of this paper as in Table 7.

Table 7: 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

A.2 ADDITIONAL EXPERIMENTS

A.2.1 CLIPPING RADIUS SCALING

The radius τ of CCLIP depends on the norm of good gradients. However, PyTorch implements SGD with momentum using the following formula

$$\boldsymbol{m}_i^t = \beta \boldsymbol{m}_i^{t-1} + \boldsymbol{g}_i(\boldsymbol{x}^{t-1}) \quad \text{for every } i \in \mathcal{G}$$

which may leads to the increase in the gradient norm.

Gradient norms. In Figure 3 we present the averaged gradient norm from all good workers. Here we use CCLIP as the aggregator and $\tau = \frac{10}{1-\beta}$. The norm of gradients are computed before aggregation. Even though the dataset on workers are non-iid, the gradient norms are roughly of same order. The gradient dissimilarity ζ^2 also increases accordingly.

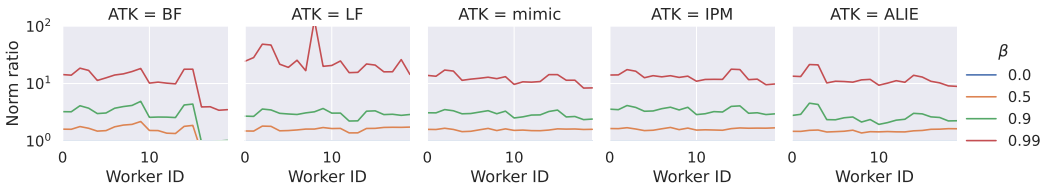


Figure 3: The ratio of norm of good gradients with momentum β over no momentum under different attacks.

Scaled clipping radius. As the gradient norm increases with momentum β, the clipping radius should increase accordingly. In Figure 4 we compare 3 schemes: 1) no scaling ($\tau = 10$, $\beta = 0$); 2) *linear* scaling $\frac{10}{1-\beta}$; 3) *sqrt* scaling $\frac{10}{\sqrt{1-\beta}}$. The no scaling scheme convergences but slower while with momentum. The linear scaling is usually better than *sqrt* scaling and with bucketing it becomes more stable. However, The scaled clipping radius fails for $\beta = 0.99$ under label flipping attack. This is because the gradient can be very large and ζ^2 dominates. So in general, a linear scaling of clipping radius with momentum $\beta = 0.9$ would be a good choice.

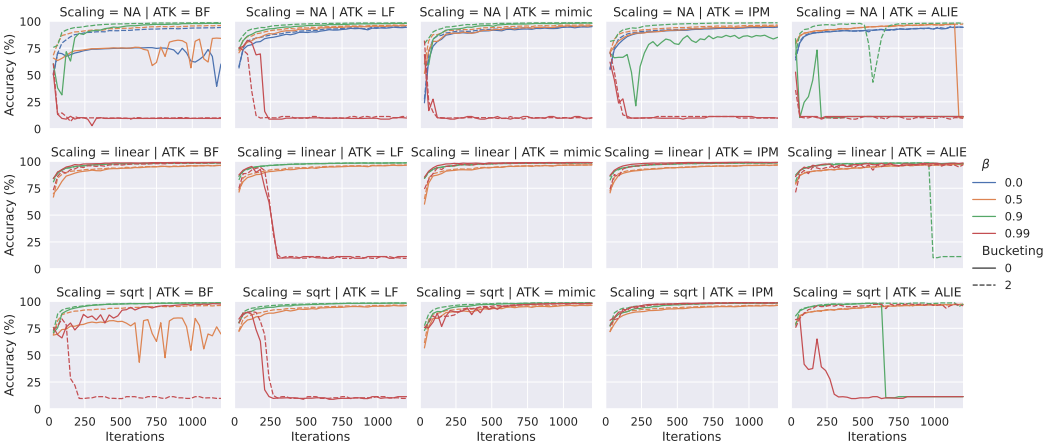


Figure 4: Convergence of CCLIP with $\tau = 10, \frac{10}{1-\beta}, \frac{10}{\sqrt{1-\beta}}$ for $\beta = 0, 0.5, 0.9, 0.99$. The s is the bucketing hyperparameter.

A.2.2 DEMONSTRATION OF EFFECTS OF BUCKETING THROUGH THE SELECTIONS OF KRUM

In the main text we have theoretically show that bucketing helps aggregators alleviate the impact of non-iid. In this section we empirically show that after bucketing aggregators can incorporate updates more evenly from good workers and therefore the problem of non-iid among good workers is less significant. Since KRUM outputs the id of the selected device, it is very convenient to record the frequency of each worker being selected. Since bucketing replicates each worker for s times, we divide their frequencies by s for normalization. From Figure 5, we can see that without bucketing KRUM basically almost always selects updates from Byzantine workers while with larger s, the selection becomes more evenly distributed.

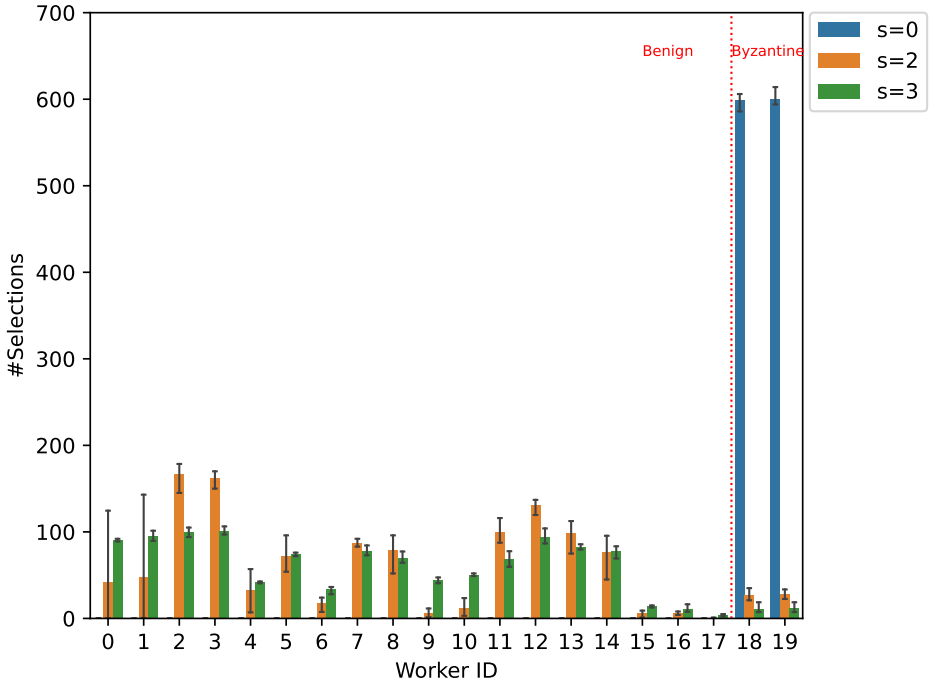


Figure 5: The selected workers of KRUM for bucketing coefficient $s = 0, 2, 3$. There are 20 workers and the last 2 workers (worker id=18,19) are Byzantine with label-flipping attack.

A.2.3 OVERPARAMETERIZATION

The architecture of the neural net used in the experiments can be scaled to make it overparameterized. We add more parameters to the model by multiplying the channels of 2D `Conv` layer and fully connected layer by a factor of 'scale'. So the original model has a scale of 1. We show the training losses decrease faster for overparameterized models in Figure 6. As we can see, the convergence behaviors are similar for different model scales with overparameterized models having smaller training loss despite the existence of Byzantine workers.

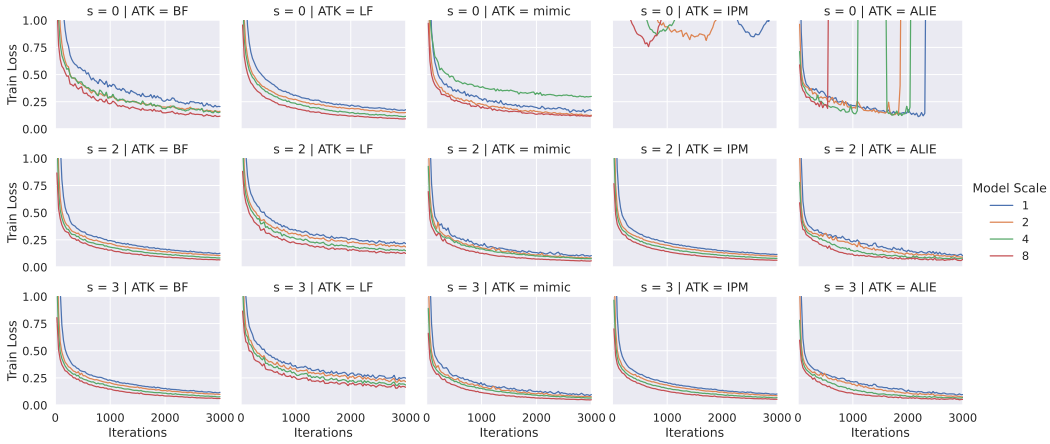


Figure 6: The training loss of models of different levels of overparameterization.

In Figure 7, we explicitly investigate the influence of overparameterization on B^2 defined in (3). As we can see, heterogeneity bound B^2 decreases with increasing level of overparameterization, showcasing how overparameterization minimizes the local objectives in the presence of Byzantine workers. It supports our theory in Section 5.4 that overparameterization can fix the convergence,

making it possible to achieve practical Byzantine-robust learning. The underlying base aggregator is RFA.

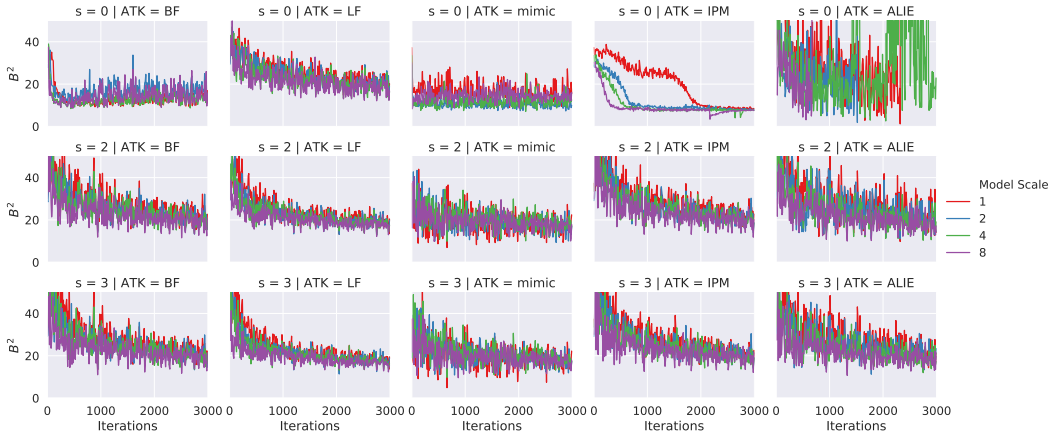


Figure 7: The B^2 in (3) for different levels of overparameterization.

A.2.4 Resampling - variant of bucketing

In the previous version of this work we repeat the gradients for s times and then put sn gradients into n buckets. The results in Figure 8 suggest that the convergence rate of bucketing and resampling is almost the same. So aggregators can benefit more from bucketing as it reduces the number of input gradients and therefore reduce the complexity.

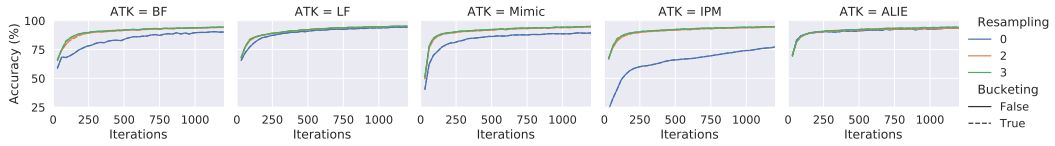


Figure 8: The convergence SGD with bucketing and resampling under different attacks. The underlying aggregator is RFA.

B Implementing the mimic attack

The Section 3.2 describes the idea and formulation of the mimic attack. In this section, we discuss how to pick $i_\star$ and implement the mimic attack efficiently. To pick $i_\star$, we use an initial phase ($\mathcal{I}^0 \approx$ 1 epoch) to compute a direction $\boldsymbol{z}$ of maximum variance of the outputs of the good workers:

$$\boldsymbol{z} = \underset{\|\boldsymbol{z}\|=1}{\operatorname{argmax}} \ \boldsymbol{z}^\top \Big(\sum_{t \in \mathcal{I}_0} \sum_{i \in \mathcal{G}} (\boldsymbol{x}_i^t - \boldsymbol{\mu})(\boldsymbol{x}_i^t - \boldsymbol{\mu})^\top \Big) \boldsymbol{z} \quad \text{where} \quad \boldsymbol{\mu} = \frac{1}{|\mathcal{G}||\mathcal{I}_0|} \sum_{i \in \mathcal{G}, t \in \mathcal{I}_0} \boldsymbol{x}_i^t \,.$$

Then we pick a worker $i^\star$ to mimic by computing

$$i_\star = \underset{i \in \mathcal{G}}{\operatorname{argmax}} \Big| \sum_{t \in \mathcal{I}_0} \boldsymbol{z}^\top \boldsymbol{x}_i^t \Big| \,.$$

In the following steps, we show how to solve the optimization problem.

First, rewrite the mimic attack in its online version at time $t \in \mathcal{I}_0$

$$\boldsymbol{z}^t = \underset{\|\boldsymbol{z}\|=1}{\operatorname{argmax}} \ h^t(\boldsymbol{z})$$

where $\boldsymbol{\mu}^t = \frac{1}{|\mathcal{G}|t} \sum_{\tau \leq t} \sum_{i \in \mathcal{G}} \boldsymbol{x}_i^\tau$ and

$$h^t(\boldsymbol{z}) = \boldsymbol{z}^\top \left(\sum_{\tau \leq t} \sum_{i \in \mathcal{G}} (\boldsymbol{x}_i^\tau - \boldsymbol{\mu}^t)(\boldsymbol{x}_i^\tau - \boldsymbol{\mu}^t)^\top \right) \boldsymbol{z}.$$

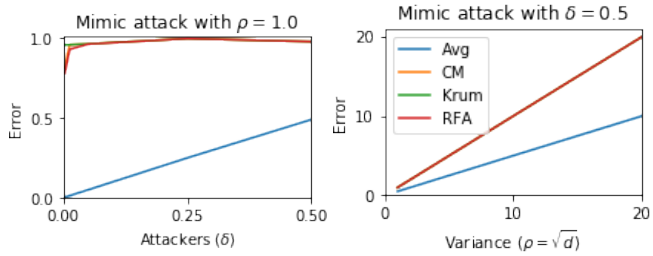


Figure 9: Error with random vectors with variance $\rho^2 = d$ and δ fraction of Byzantine workers imitating a fixed good worker (say worker $1 \in \mathcal{G}$). RFA performs slightly better than CM and KRUM, but all have *higher error* than simply averaging across various settings of δ and ρ.

Thus we can iteratively update $\boldsymbol{\mu}^t$ by

$$\boldsymbol{\mu}^{t+1} = \frac{t}{1+t}\boldsymbol{\mu}^t + \frac{1}{1+t}\frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}} \boldsymbol{x}_i^{t+1},$$

and then

$$\begin{aligned}\operatorname*{argmax}_{\|\boldsymbol{z}\|=1} h^{t+1}(\boldsymbol{z}) &\approx \frac{t}{1+t}\boldsymbol{z}^t + \frac{1}{1+t}\operatorname*{argmax}_{\|\boldsymbol{z}\|=1} \boldsymbol{z}^\top \left(\sum_{i\in\mathcal{G}} (\boldsymbol{x}_i^{t+1} - \boldsymbol{\mu}^{t+1})(\boldsymbol{x}_i^{t+1} - \boldsymbol{\mu}^{t+1})^\top\right)\boldsymbol{z} \\ &\approx \frac{t}{1+t}\boldsymbol{z}^t + \frac{1}{1+t}\left(\sum_{i\in\mathcal{G}} (\boldsymbol{x}_i^{t+1} - \boldsymbol{\mu}^{t+1})(\boldsymbol{x}_i^{t+1} - \boldsymbol{\mu}^{t+1})^\top\right)\boldsymbol{z}^t.\end{aligned}$$

The above algorithm corresponds to Oja's method for computing the top eigenvector in a streaming fashion (Oja, 1982). Then, in each subsequent iteration t, we pick

$$i_\star^t = \operatorname*{argmax}_{i\in\mathcal{G}} \boldsymbol{z}^\top \boldsymbol{x}_i^t\,.$$

Example. Each of the good workers $i \in \mathcal{G} \subseteq [n]$ has an input a $\boldsymbol{x}_i \in \{\pm 1\}^d$ where each coordinate is an independent Rademacher random variable. The inputs then have mean $\mathbf{0}$ and variance $\mathbb{E}\|\boldsymbol{x}_i\|^2 = \rho^2 = d$. Now, the Byzantine attackers $j \in \mathcal{B}$ have dual goals: i) escape detection, and ii) increase data imbalance. For this, we propose the following simple passive attack: pick some fixed worker $i_\star \in \mathcal{G}$ (say 1) and every Byzantine worker $j \in \mathcal{B}$ outputs $\boldsymbol{x}_j = \boldsymbol{x}_1$. The attackers cannot be filtered as they imitate an existing good worker, but still can cause imbalance in the data distribution. This serves as the intuition for our attack.

C CONSTRUCTING A ROBUST AGGREGATOR USING BUCKETING

C.1 SUPPORTING LEMMAS

We first start with proving the main bucketing Lemma 1 restated below.

Lemma' 1. *Suppose we are given n independent (but not identical) random vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ such that a good subset $\mathcal{G} \subseteq [n]$ of size at least $|\mathcal{G}| \geq n(1-\delta)$ satisfies:*

$$\mathbb{E}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \leq \rho^2\,, \quad \textit{for any fixed } i, j \in \mathcal{G}\,.$$

Define $\bar{\boldsymbol{x}} := \frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} \boldsymbol{x}_j$ and $m = \lceil n/s \rceil$. Let the outputs after s-bucketing be $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m\}$. Then, a subset of the outputs $\tilde{\mathcal{G}} \subseteq \{1, \ldots, m\}$ of size at least $|\tilde{\mathcal{G}}| \geq m(1-\delta s)$ satisfies

$$\mathbb{E}[\boldsymbol{y}_i] = \mathbb{E}[\bar{\boldsymbol{x}}] \quad \textit{and} \quad \mathbb{E}\|\boldsymbol{y}_i - \boldsymbol{y}_j\| \leq \rho^2/s \quad \textit{for any fixed } i, j \in \tilde{\mathcal{G}}\,.$$

Proof. Let us define the buckets used to compute $\boldsymbol{y}_i$ as

$$B_i := \{\pi(s(i-1)+1), \ldots, \pi(\min\{s\cdot i, n\})\}\,.$$

Recall that for some permutation π over $[n]$ and for every $i = \{1, \ldots, m\}$, we defined $m = \lceil n/s \rceil$ and

$$\boldsymbol{y}_i \leftarrow \frac{1}{|B_i|} \sum_{k=(i-1)\cdot s+1}^{\min(n\,,\,i\cdot s)} \boldsymbol{x}_{\pi(k)}\,.$$

Then, define the *new* good set

$$\tilde{\mathcal{G}} = \{i \in [m] \mid B_i \subseteq \mathcal{G}\}$$

$\tilde{\mathcal{G}}$ contains the set of all the resampled vectors which are made up of only good vectors i.e. are uninfluenced by any Byzantine vector. Since $|\mathcal{B}| \leq \delta n$ and each can belong to only 1 bucket, we have that $|\tilde{\mathcal{G}}| \geq (1 - \delta s)m$. Now, for any fixed $i \in \tilde{\mathcal{G}}$, let us look at the conditional expectation over the random permutation π we have

$$\mathbb{E}_\pi[\boldsymbol{y}_i | i \in \tilde{\mathcal{G}}] = \frac{1}{|B_i|} \sum_{k=(i-1)\cdot s+1}^{\min(n\,,\,i\cdot s)} \mathbb{E}_\pi[\boldsymbol{x}_{\pi(k)} | \pi(k) \in \mathcal{G}] = \frac{1}{|\mathcal{G}|} \sum_{j \in \mathcal{G}} \boldsymbol{x}_j = \bar{\boldsymbol{x}}\,.$$

This yields the first part of the lemma. Now we analyze the variance. Thus, we can write $\boldsymbol{y}_i = \frac{1}{s}\sum_{k \in B_i} \boldsymbol{x}_k$. Further, $|B_i| = s$ for any i, and $B_i \subseteq \mathcal{G}$ if $i \in \tilde{\mathcal{G}}$. With this, for any fixed $i, j \in \tilde{\mathcal{G}}$ the variance can be written as

$$\begin{aligned}\mathbb{E}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 &= \mathbb{E}\left\|\frac{1}{s}\sum_{k \in B_i} \boldsymbol{x}_k - \frac{1}{s}\sum_{l \in B_j} \boldsymbol{x}_l\right\|^2 \\ &= \frac{\rho^2}{s}\,.\end{aligned}$$

□

This additional lemma about the maximum expected distance between good workers will also be useful later.

Lemma 7 (maximum good distance)**.** *Suppose we are given the output of bucketing $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ which for $m = \lceil n/s \rceil$ satisfy for any fixed $i \in \tilde{\mathcal{G}}$, $\mathbb{E}[\boldsymbol{y}_i] = \boldsymbol{\mu}$ and $\mathbb{E}\|\boldsymbol{y}_i - \boldsymbol{\mu}\|^2 \leq \rho^2/s$. Then, we have*

$$\mathbb{E}\left[\max_{i \in \tilde{\mathcal{G}}} \|\boldsymbol{y}_i - \boldsymbol{\mu}\|^2\right] \leq n\rho^2/s^2\,.$$

Further, there exist instances where

$$\mathbb{E}\left[\max_{i \in \tilde{\mathcal{G}}} \|\boldsymbol{y}_i - \boldsymbol{\mu}\|^2\right] \geq \Omega(n\rho^2/s^2)\,.$$

Proof. For the upper bound, we simply use

$$\mathbb{E}\left[\max_{i \in \tilde{\mathcal{G}}} \|\boldsymbol{y}_i - \boldsymbol{\mu}\|^2\right] \leq \sum_{i \in \tilde{\mathcal{G}}} \mathbb{E}\|\boldsymbol{y}_i - \boldsymbol{\mu}\|^2 \leq m\rho^2/s\,.$$

For the lower bound, let $\tilde{\mathcal{G}} = [m]$ and consider $\boldsymbol{y}_i \sim \tilde{\rho}\sqrt{m}\text{Bern}(p = \frac{1}{m})$. This means $\boldsymbol{y}_i$ is either 0 or $\tilde{\rho}\sqrt{m}$. Further, its variance is clearly bounded by $\tilde{\rho}^2$. Upon drawing m samples, the probability of seeing at least 1 $\boldsymbol{y}_j = \tilde{\rho}\sqrt{m}$ is

$$1 - \Pr(\boldsymbol{y}_i = 0\ \forall i \in [m]) = 1 - (1 - \tfrac{1}{m})^m \geq 1 - 1/e \geq 0.5\,.$$

Thus, with probability at least 0.5 we have

$$\max_{i \in [n]} \|\boldsymbol{y}_i - \boldsymbol{\mu}\|^2 \geq m\tilde{\rho}^2/2\,.$$

This directly proves our lower bound by defining $\tilde{\rho}^2 := \rho^2/s$ and recalling that $m = \lceil n/s \rceil$. Note that this lemma can be tightened if we make stronger assumptions on the noise such as $\mathbb{E}\|\boldsymbol{y}_i - \boldsymbol{\mu}\|^r \leq (\rho/\sqrt{s})^r$ for some large $r \geq 2$. However, we focus on using standard stochastic assumptions ($r = 2$) in this work. □

C.2 PROOFS OF ROBUSTNESS

Let $\{\boldsymbol{y}_1 \ldots, \boldsymbol{y}_m\}$ be the resampled vectors with bucketing using $s = \frac{\delta_{\max}}{\delta}$. By Lemma 1, we have that there is a $\tilde{\mathcal{G}} \subseteq [m]$ of size $|\tilde{\mathcal{G}}| > m(1-\delta_{\max})$ which satisfies for any fixed $i, j \in \tilde{\mathcal{G}}$

$$\mathbb{E}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 \leq \frac{\delta\rho^2}{\delta_{\max}} =: \tilde{\rho}^2 \,.$$

This observation will be combined with each of the algorithms to obtain robustness guarantees.

Robustness of KRUM. We now prove that KRUM when combined with bucketing is a robust aggregator. We can rewrite the output of KRUM as the following for $\delta_{\max} = 1/4 - \nu$ for some arbitrarily small positive number $\nu \in (0, 1/4)$:

$$\text{KRUM}(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m) = \underset{\boldsymbol{y}_i}{\operatorname{argmin}} \min_{|\mathcal{S}|=3m/4} \sum_{j\in\mathcal{S}} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 \,.$$

Let $\mathcal{S}^\star$ and $k^\star$ be the quantities which minimize the optimization problem solved by KRUM.

The main difficulty of analyzing KRUM is that even if we succeed in selecting a $k^\star \in \tilde{\mathcal{G}}$, $k^\star$ depends on the sampling. Hence, we **cannot** claim that the error is bounded by $\tilde{\rho}^2$ i.e [3]

$$\mathbb{E}\|\boldsymbol{y}_{k^\star} - \boldsymbol{y}_j\|^2 \not\leq \tilde{\rho}^2 \text{ for some fixed } j \in \tilde{\mathcal{G}} \,.$$

This is because the variance is bounded by $\tilde{\rho}^2$ only for a *fixed* i, and not a data dependent $k^\star$. Instead, we will have to rely on Lemma 7 that

$$\mathbb{E}\|\boldsymbol{y}_{k^\star} - \boldsymbol{y}_j\|^2 \leq \mathbb{E}\max_{i\in\tilde{\mathcal{G}}} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 \leq m\tilde{\rho}^2 \,.$$

Lemma 7 shows that this inequality is essentially tight and hence relying on it necessarily incurs an extra factor of m which can be very large. Instead, we show an alternate analysis which works for a smaller breakdown point of $\delta_{\max} = 1/4$, but *does not* incur the extra m factor.

For any good input $i \in \tilde{\mathcal{G}}$, we have

$$\begin{aligned} \|\boldsymbol{y}_{k^\star} - \bar{\boldsymbol{x}}\|^2 &\leq 2\|\boldsymbol{y}_{k^\star} - \boldsymbol{y}_i\|^2 + 2\|\boldsymbol{y}_i - \bar{\boldsymbol{x}}\|^2 \\ \Rightarrow 2\|\boldsymbol{y}_{k^\star} - \boldsymbol{y}_i\|^2 &\geq \|\boldsymbol{y}_{k^\star} - \bar{\boldsymbol{x}}\|^2 - 2\|\boldsymbol{y}_i - \bar{\boldsymbol{x}}\|^2 \,. \end{aligned}$$

Further, for a bad worker $j \in \tilde{\mathcal{B}}$ we can write

$$2\|\boldsymbol{y}_{k^\star} - \boldsymbol{y}_j\|^2 \geq \|\boldsymbol{y}_j - \bar{\boldsymbol{x}}\|^2 - 2\|\boldsymbol{y}_{k^\star} - \bar{\boldsymbol{x}}\|^2 \,.$$

Combining both and summing over $\mathcal{S}^\star$,

$$\begin{aligned} \sum_{i\in\mathcal{S}^\star} 2\|\boldsymbol{y}_{k^\star} - \boldsymbol{y}_i\|^2 &= \sum_{i\in\tilde{\mathcal{G}}\cap\mathcal{S}^\star} 2\|\boldsymbol{y}_{k^\star} - \boldsymbol{y}_i\|^2 + \sum_{j\in\tilde{\mathcal{B}}\cap\mathcal{S}^\star} 2\|\boldsymbol{y}_{k^\star} - \boldsymbol{y}_j\|^2 \\ &\geq \sum_{j\in\tilde{\mathcal{B}}\cap\mathcal{S}^\star} \|\boldsymbol{y}_j - \bar{\boldsymbol{x}}\|^2 - 2\sum_{i\in\tilde{\mathcal{G}}\cap\mathcal{S}^\star} \|\boldsymbol{y}_i - \bar{\boldsymbol{x}}\|^2 \\ &\qquad + (|\tilde{\mathcal{G}}\cap\mathcal{S}^\star| - 2|\tilde{\mathcal{B}}\cap\mathcal{S}^\star|)\|\boldsymbol{y}_{k^\star} - \bar{\boldsymbol{x}}\|^2 \,. \end{aligned}$$

We can rearrange the above equation as

$$\begin{aligned} \|\boldsymbol{y}_{k^\star} - \bar{\boldsymbol{x}}\|^2 &\leq \frac{1}{(|\tilde{\mathcal{G}}\cap\mathcal{S}^\star| - 2|\tilde{\mathcal{B}}\cap\mathcal{S}^\star|)}\Big(\sum_{i\in\mathcal{S}^\star} 2\|\boldsymbol{y}_{k^\star} - \boldsymbol{y}_i\|^2 + \sum_{i\in\tilde{\mathcal{G}}\cap\mathcal{S}^\star} 2\|\boldsymbol{y}_i - \bar{\boldsymbol{x}}\|^2\Big) \\ &\leq \frac{1}{(|\mathcal{S}^\star| - 3|\tilde{\mathcal{B}}|)}\Big(\sum_{i\in\mathcal{S}^\star} 2\|\boldsymbol{y}_{k^\star} - \boldsymbol{y}_i\|^2 + \sum_{i\in\tilde{\mathcal{G}}\cap\mathcal{S}^\star} 2\|\boldsymbol{y}_i - \bar{\boldsymbol{x}}\|^2\Big) \\ &\leq \frac{1}{(|\mathcal{S}^\star| - 3|\tilde{\mathcal{B}}|)}\Big(2\min_{k,|\mathcal{S}|=3m/4} \sum_{i\in\mathcal{S}} \|\boldsymbol{y}_k - \boldsymbol{y}_i\|^2 + \sum_{i\in\tilde{\mathcal{G}}} 2\|\boldsymbol{y}_i - \bar{\boldsymbol{x}}\|^2\Big). \end{aligned}$$

[3] This issue was incorrectly overlooked in the original analysis of KRUM (Blanchard et al., 2017)

Taking expectation now on both sides yields

$$\mathbb{E}\|\boldsymbol{y}_{k^\star} - \bar{\boldsymbol{x}}\|^2 \leq \frac{4m\tilde{\rho}^2}{|\mathcal{S}^\star| - 3|\tilde{\mathcal{B}}|}\,.$$

Now, recall that we used a bucketing value of $s = {}^{\delta_{\max}}/\delta$ where for KRUM we have $\delta_{\max} = {}^1/_4 - \nu$. Then, the number of Byzantine workers can be bounded as $|\tilde{\mathcal{B}}| \leq m(1/4 - \nu)$. This gives the final result that

$$\mathbb{E}\|\boldsymbol{y}_{k^\star} - \bar{\boldsymbol{x}}\|^2 \leq \frac{4m\tilde{\rho}^2}{3m/4 - 3(m/4 - \nu m)} = \frac{4\tilde{\rho}^2}{3\nu} \leq \frac{4}{3\nu(1/4 - \nu)}\delta\rho^2\,.$$

Thus, KRUM with bucketing indeed satisfies Definition A with $\delta_{\max} = ({}^1/_4 - \nu)$ and $c = 4/(3\nu({}^1/_4 - \nu))$.

Robustness of Geometric median. Geometric median computes the minimum of the following optimization problem

$$\boldsymbol{y}^\star = \operatorname*{argmin}_{\boldsymbol{y}} \sum_{i\in[m]} \|\boldsymbol{y} - \boldsymbol{y}_i\|_2\,.$$

We will adapt Lemma 24 of Cohen et al. (2016), which itself is based on (Minsker et al., 2015). For a good bucket $i \in \tilde{\mathcal{G}}$ and bad bucket $j \in \tilde{\mathcal{B}}$:

$$\|\boldsymbol{y}^\star - \boldsymbol{y}_i\|_2 \geq \|\boldsymbol{y}^\star - \bar{\boldsymbol{x}}\|_2 - \|\boldsymbol{y}_i - \bar{\boldsymbol{x}}\|_2 \text{ for } i \in \tilde{\mathcal{G}}, \text{ and}$$
$$\|\boldsymbol{y}^\star - \boldsymbol{y}_j\|_2 \geq \|\boldsymbol{y}_j - \bar{\boldsymbol{x}}\|_2 - \|\boldsymbol{y}^\star - \bar{\boldsymbol{x}}\|_2 \text{ for } j \in \tilde{\mathcal{B}}\,.$$

Summing this over all buckets we have

$$\begin{aligned}
\sum_{i\in[n]} \|\boldsymbol{y}^\star - \boldsymbol{y}_i\| &\geq (|\tilde{\mathcal{G}}| - |\tilde{\mathcal{B}}|)\|\boldsymbol{y}^\star - \bar{\boldsymbol{x}}\| + \sum_{j\in\tilde{\mathcal{B}}}\|\boldsymbol{y}_j - \bar{\boldsymbol{x}}\| - \sum_{i\in\tilde{\mathcal{G}}}\|\boldsymbol{y}_i - \bar{\boldsymbol{x}}\| \\
\Rightarrow \|\boldsymbol{y}^\star - \bar{\boldsymbol{x}}\| &\leq \frac{1}{(|\tilde{\mathcal{G}}| - |\tilde{\mathcal{B}}|)}\left(\sum_{i\in[n]}\|\boldsymbol{y}^\star - \boldsymbol{y}_i\| - \sum_{j\in\tilde{\mathcal{B}}}\|\boldsymbol{y}_j - \bar{\boldsymbol{x}}\| + \sum_{i\in\tilde{\mathcal{G}}}\|\boldsymbol{y}_i - \bar{\boldsymbol{x}}\|\right) \\
&= \frac{1}{(|\tilde{\mathcal{G}}| - |\tilde{\mathcal{B}}|)}\left(\min_{\boldsymbol{y}}\sum_{i\in[n]}\|\boldsymbol{y} - \boldsymbol{y}_i\| - \sum_{j\in\tilde{\mathcal{B}}}\|\boldsymbol{y}_j - \bar{\boldsymbol{x}}\| + \sum_{i\in\tilde{\mathcal{G}}}\|\boldsymbol{y}_i - \bar{\boldsymbol{x}}\|\right) \\
&\leq \frac{2}{(|\tilde{\mathcal{G}}| - |\tilde{\mathcal{B}}|)}\left(\sum_{i\in\tilde{\mathcal{G}}}\|\boldsymbol{y}_i - \bar{\boldsymbol{x}}\|\right)\,.
\end{aligned}$$

The last step we substituted $\boldsymbol{y} = \bar{\boldsymbol{x}}$. Squaring both sides, expanding, and then taking expectation gives

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{y}^\star - \bar{\boldsymbol{x}}\|^2 &\leq \frac{4}{(|\tilde{\mathcal{G}}| - |\tilde{\mathcal{B}}|)^2}\,\mathbb{E}\left(\sum_{i\in\tilde{\mathcal{G}}}\|\boldsymbol{y}_i - \bar{\boldsymbol{x}}\|\right)^2 \\
&\leq \frac{4}{(|\tilde{\mathcal{G}}| - |\tilde{\mathcal{B}}|)^2}\left(|\tilde{\mathcal{G}}|\sum_{i\in\tilde{\mathcal{G}}}\mathbb{E}\|\boldsymbol{y}_i - \bar{\boldsymbol{x}}\|^2\right) \\
&\leq \frac{4|\tilde{\mathcal{G}}|^2}{(n - 2|\tilde{\mathcal{B}}|)^2}\tilde{\rho}^2\,.
\end{aligned}$$

Now, recall that we used a bucketing value of $s = {}^{\delta_{\max}}/\delta$ where for KRUM we have $\delta_{\max} = {}^1/_2 - \nu$. Then, the number of Byzantine workers can be bounded as $|\tilde{\mathcal{B}}| \leq n(1/2 - \nu)$. This gives the final result that

$$\mathbb{E}\|\boldsymbol{y}^\star - \bar{\boldsymbol{x}}\|^2 \leq \frac{4n^2}{4n^2\nu^2}\tilde{\rho}^2 \leq \frac{\tilde{\rho}^2}{\nu^2} \leq \frac{1}{\nu(1/2 - \nu)}\delta\rho^2\,.$$

Thus, geometric median with bucketing indeed satisfies Definition A with $\delta_{\max} = ({}^1/_2 - \nu)$ and $c = 1/(\nu({}^1/_2 - \nu))$. Note that geometric median has better theoretical performance than KRUM.

Robustness of Coordinate-wise median. The proof of coordinate-wise median largely follows that of the geometric median. First, we note that we can separate out the objective by coordinates

$$\mathbb{E}\|\mathrm{CM}(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m) - \bar{\boldsymbol{x}}\|^2 = \sum_{l=1}^{d} \mathbb{E}(\mathrm{CM}([\boldsymbol{y}_1]_l, \ldots, [\boldsymbol{y}_m]_l) - [\bar{\boldsymbol{x}}]_l)^2 \,.$$

Then note that, for any fixed coordinate $l \in [d]$ and fixed good worker $i \in \mathcal{G}$, we have $\mathbb{E}([\boldsymbol{y}_i]_l - [\bar{\boldsymbol{x}}]_l)^2 \leq \mathbb{E}\|\boldsymbol{y}_i - \bar{\boldsymbol{x}}\|^2 \leq \tilde{\rho}^2$. Thus, we can simply analyze coordinate-wise median as d separate (geometric) median problems on scalars. Thus for any fixed coordinate $l \in [d]$, we have

$$\mathbb{E}(\mathrm{CM}([\boldsymbol{y}_1]_l, \ldots, [\boldsymbol{y}_m]_l) - [\bar{\boldsymbol{x}}]_l)^2 \leq \frac{\tilde{\rho}^2}{\nu^2} \Rightarrow \mathbb{E}\|\mathrm{CM}(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m) - \bar{\boldsymbol{x}}\|^2 \leq \frac{d\tilde{\rho}^2}{\nu^2} \leq \frac{d}{\nu(1/2 - \nu)}\delta\rho^2 \,.$$

Thus, coordinate-wise median with bucketing indeed satisfies Definition A with $\delta_{\max} = (1/2 - \nu)$ and $c = d/(\nu(1/2 - \nu))$.

D Lower bounds on non-iid data (Proof of Theorem III)

Our proof builds two sets of functions $\{f_i^1(\boldsymbol{x}) \,|\, i \in \mathcal{G}^1\}$ and $\{f_i^2(\boldsymbol{x}) \,|\, i \in \mathcal{G}^2\}$ and will show that in the presence of δ-fraction of Byzantine workers, no algorithm can distinguish between them. Since the problems have different optima, this means that the algorithm necessarily has an error on at least one of them.

For the first set of functions, let there be *no* bad workers and hence $\mathcal{G}^1 = [n]$. Then, we define the following functions for any $i \in [n]$:

$$f_i^1(x) = \begin{cases} \frac{\mu}{2}x^2 - \zeta\delta^{-1/2}x & \text{for } i \in \{1, \ldots, \delta n\} \\ \frac{\mu}{2}x^2 & \text{for } i \in \{\delta n + 1, \ldots, n\} \,. \end{cases}$$

Defining $G := \zeta\delta^{1/2}$, the average function which is our objective is

$$f^1(x) = \frac{1}{n}\sum_{i=1}^{n} f_i^1(x) = \frac{\mu}{2}x^2 - Gx \,.$$

The optimum of our $f^1(x)$ is at $x = \frac{G}{\mu}$. Note that the gradient heterogeneity amongst these workers is bounded as

$$\begin{aligned} \mathbb{E}_{i\sim[n]}\|\nabla f_i^1(x) - \nabla f^1(x)\|^2 =& \delta(\zeta\delta^{-1/2} - \zeta\delta^{1/2})^2 + (1-\delta)(\zeta\delta^{1/2})^2 \\ =& \zeta^2(1-\delta)^2 + \zeta^2(1-\delta)\delta = \zeta^2(1-\delta) \leq \zeta^2. \end{aligned}$$

Now, we define the second set of functions. Here, suppose that we have δn Byzantine attackers with $\mathcal{B}^2 = \{1, \ldots, \delta n\}$. Then, the functions of the good workers are defined as

$$f_i^2(x) = \frac{\mu}{2}x^2 \text{ for } i \in \mathcal{G}^2 = \{\delta n + 1, \ldots, n\} \,.$$

We then have that the second average objective is

$$f^2(x) = \frac{1}{|\mathcal{G}^2|}\sum_{i\in\mathcal{G}^2} f_i^2(x) = \frac{\mu}{2}x^2 \,.$$

Here, we have gradient heterogeneity of 0 and hence is smaller than ζ^2. The optimum of $f^2(x)$ is at $x = 0$. The Byzantine attackers simply imitate as if they have the functions:

$$f_j^2(x) = \frac{\mu}{2}x^2 - \zeta\delta^{-1/2}x \text{ for } j \in \mathcal{B}^2 = \{1, \ldots, \delta n\} \,.$$

Note that the set of functions, $\{f_1^1(\boldsymbol{x}), \ldots, f_n^1(x)\}$ is exactly identical to the set $\{f_1^2(\boldsymbol{x}), \ldots, f_n^2(x)\}$. Only the identity of the good workers $\mathcal{G}^1$ and $\mathcal{G}^2$ are different, leading to different objective functions $f^1(x)$ and $f^2(x)$. However, since the algorithm does not have access to $\mathcal{G}$, its output on each of them is identical i.e.

$$x^{\text{out}} = \textsc{Alg}(f_1^1(\boldsymbol{x}), \ldots, f_n^1(x)) = \textsc{Alg}(f_1^2(\boldsymbol{x}), \ldots, f_n^2(x)) \,.$$

However, the leads to making a large error in least one of f^1 and f^2 since they have different optimum. This proves a lower bound error of

$$\max_{k\in\{1,2\}} f^k(x^{\text{out}}) - f^k(x^\star) \geq \mu\left(\frac{G}{2\mu}\right)^2 = \frac{\delta\zeta^2}{4\mu}\,.$$

Similarly, we can also bound the gradient norm error bound as

$$\max_{k\in\{1,2\}} \|\nabla f^k(x^{\text{out}})\|^2 \geq \mu^2\left(\frac{G}{2\mu}\right)^2 = \frac{\delta\zeta^2}{4}\,.$$

□

E Convergence of robust optimization on non-iid data (Theorems II and IV)

We will prove a more general convergence theorem which generalizes Theorems II and IV.

Theorem V. *Suppose we are given a $(\delta_{\max}, c)$-ARAgg satisfying Definition A, and n workers of which a subset $\mathcal{G}$ of size at least $|\mathcal{G}| \geq n(1-\delta)$ faithfully follow the algorithm for $\delta \leq \delta_{\max}$. Further, for any good worker $i \in \mathcal{G}$ let f_i be a possibly non-convex function with L-Lipschitz gradients, and the stochastic gradients on each worker be independent, unbiased and satisfy*

$$\mathbb{E}_{\boldsymbol{\xi}_i}\|\boldsymbol{g}_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x})\|^2 \leq \sigma^2 \text{ and } \mathbb{E}_{j\sim\mathcal{G}}\|\nabla f_j(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2 \leq \zeta^2 + B^2\|\nabla f(\boldsymbol{x})\|^2\,, \quad \forall \boldsymbol{x}\,,$$

where $\delta \leq 1/(60cB^2)$. Then, for $F^0 := f(\boldsymbol{x}^0) - f^\star$, the output of Algorithm 2 with step-size $\eta = \min\left(\mathcal{O}\left(\sqrt{\frac{LF^0 + c\delta(\zeta^2+\sigma^2)}{TL^2\sigma^2(n^{-1}+c\delta)}}\right), \frac{1}{8L}\right)$ and momentum parameter $\beta = (1-8L\eta)$ satisfies

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 \leq \mathcal{O}\Big(\frac{1}{1-60c\delta B^2}\cdot\Big(c\delta\zeta^2 + \sigma\sqrt{\frac{LF^0}{T}(c\delta + 1/n)} + \frac{LF^0}{T}\Big)\Big)\,.$$

Notes on $\delta \leq 1/(60cB^2)$. In practice the upper bound $\delta \leq 1/(60cB^2)$ does not put an extra strict constraint on δ. This is because one can always decrease B^2 and increase ζ^2 such that $\mathbb{E}_{j\sim\mathcal{G}}\|\nabla f_j(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2 \leq \zeta^2 + B^2\|\nabla f(\boldsymbol{x})\|^2$ holds for a sufficiently large domain of $\boldsymbol{x}$.

Definitions. Recall our algorithm which performs for $t \geq 2$ the following update with $(1-\beta) = \alpha$:

$$\begin{aligned}\boldsymbol{m}_i^t &= (1-\alpha)\boldsymbol{m}_i^{t-1} + \alpha\boldsymbol{g}_i(\boldsymbol{x}^{t-1}) \quad \text{for every } i \in \mathcal{G}\,,\\ \boldsymbol{x}^t &= \boldsymbol{x}^{t-1} - \eta\,\text{ARAgg}(\boldsymbol{m}_1^t, \ldots, \boldsymbol{m}_n^t)\,.\end{aligned}$$

For $t = 1$, we use $\alpha = 0$ i.e. $\boldsymbol{m}_i^1 = \boldsymbol{g}_i(\boldsymbol{x}^0)$. Let us also define the actual and ideal momentum aggregates as

$$\boldsymbol{m}^t := \text{ARAgg}(\boldsymbol{m}_1^t, \ldots, \boldsymbol{m}_n^t) \quad \text{and} \quad \bar{\boldsymbol{m}}^t := \frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}}\boldsymbol{m}_i^t\,.$$

We state several supporting lemmas before proving our main Theorem V. We will loosely follow the proof of Byzantine robustness in the iid case by Karimireddy et al. (2021), with the key difference of Lemma 8 which accounts for the non-iid error.

Lemma 8 (Aggregation error)**.** *Given that* ARAgg *satisfies Definition A holds, 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*

$$\mathbb{E}\|\boldsymbol{m}^t - \bar{\boldsymbol{m}}^t\|^2 \leq c\delta\rho_t^2\,,$$

where we define for $t \geq 2$

$$\rho_t^2 := 4(6\alpha\sigma^2 + 3\zeta^2) + 4(6\sigma^2 - 3\zeta^2)(1-\alpha)^t + 12\sum_{k=1}^{t}(1-\alpha)^{t-k}\alpha B^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{k-1})\|^2\,.$$

For $t = 1$ we can simplify the bound as $\rho_1^2 := 24c\delta\sigma^2 + 12c\delta\zeta^2 + 12c\delta B^2\|\nabla f(\boldsymbol{x}^0)\|^2$.

Proof. Let $\mathbb{E}_{\xi^t} := \mathbb{E}_{\xi_1^t,\ldots,\xi_n^t,\xi_1^{t-1},\ldots,\xi_n^{t-1},\ldots,\xi_1^0,\ldots,\xi_n^0}$ be the expectation with respect to all of the randomness until time t and let $\mathbb{E}_i := \mathbb{E}_{i\in\mathcal{G}}$ and $\mathbb{E} = \mathbb{E}_{\xi^t}\,\mathbb{E}_i$. Expanding the definition of the worker momentum for a fixed good workers $i \in \mathcal{G}$,

$$\begin{aligned}\mathbb{E}_{\xi^t}\|\boldsymbol{m}_i^t - \mathbb{E}_{\xi^t}[\boldsymbol{m}_i^t]\|^2 &= \mathbb{E}_{\xi^t}\|\alpha(\boldsymbol{g}_i(\boldsymbol{x}^{t-1}) - \nabla f_i(\boldsymbol{x}^{t-1})) + (1-\alpha)(\boldsymbol{m}_i^{t-1} - \mathbb{E}_{\xi^t}[\boldsymbol{m}_i^{t-1}])\|^2 \\ &\leq \mathbb{E}_{\xi^{t-1}}\|(1-\alpha)(\boldsymbol{m}_i^{t-1} - \mathbb{E}[\boldsymbol{m}_i^{t-1}])\|^2 + \alpha^2\sigma^2 \\ &\leq (1-\alpha)\,\mathbb{E}_{\xi^{t-1}}\|\boldsymbol{m}_i^{t-1} - \mathbb{E}[\boldsymbol{m}_i^{t-1}]\|^2 + \alpha^2\sigma^2\,.\end{aligned}$$

Unrolling the recursion above yields

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

Similar computation also shows

$$\mathbb{E}_{\xi^t}\|\bar{\boldsymbol{m}}^t - \mathbb{E}_{\xi^t}[\bar{\boldsymbol{m}}^t]\|^2 \leq \frac{\sigma^2}{n}(\alpha + (1-\alpha)^{t-1})\,.$$

So far, the expectation was only over the stochasticity of the gradients of worker i. Note that we have $\mathbb{E}_{\xi^t}[\boldsymbol{m}_i^t] = \mathbb{E}_{\xi^{t-1}}[\alpha\nabla f_i(\boldsymbol{x}^{t-1}) + (1-\alpha)\boldsymbol{m}_i^{t-1}]$. Now, suppose we sample i uniformly at random from $\mathcal{G}$. Then,

$$\begin{aligned}&\mathbb{E}_i\left\|\mathbb{E}_{\xi^t}[\boldsymbol{m}_i^t] - \mathbb{E}_{\xi^t}[\bar{\boldsymbol{m}}^t]\right\|^2 \\ =&\mathbb{E}_i\|\alpha\,\mathbb{E}_{\xi^{t-1}}[\nabla f_i(\boldsymbol{x}^{t-1}) - \nabla f(\boldsymbol{x}^{t-1})] + (1-\alpha)(\mathbb{E}_{\xi^{t-1}}[\boldsymbol{m}_i^{t-1}] - \mathbb{E}_{\xi^{t-1}}[\bar{\boldsymbol{m}}^{t-1}])\|^2 \\ \leq&(1-\alpha)\,\mathbb{E}_i\|\mathbb{E}_{\xi^{t-1}}[\boldsymbol{m}_i^{t-1}] - \mathbb{E}_{\xi^{t-1}}[\bar{\boldsymbol{m}}^{t-1}]\|^2 + \alpha\,\mathbb{E}_i\|\mathbb{E}_{\xi^{t-1}}[\nabla f_i(\boldsymbol{x}^{t-1}) - \nabla f(\boldsymbol{x}^{t-1})]\|^2 \\ \leq&(1-\alpha)\,\mathbb{E}_i\|\mathbb{E}_{\xi^{t-1}}[\boldsymbol{m}_i^{t-1}] - \mathbb{E}_{\xi^{t-1}}[\bar{\boldsymbol{m}}^{t-1}]\|^2 + \alpha\,\mathbb{E}_i\,\mathbb{E}_{\xi^{t-1}}\|\nabla f_i(\boldsymbol{x}^{t-1}) - \nabla f(\boldsymbol{x}^{t-1})\|^2 \\ \leq&(1-\alpha)\,\mathbb{E}_i\|\mathbb{E}_{\xi^{t-1}}[\boldsymbol{m}_i^{t-1}] - \mathbb{E}_{\xi^{t-1}}[\bar{\boldsymbol{m}}^{t-1}]\|^2 + \alpha\zeta^2 + \alpha B^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2\end{aligned}$$

where the second inequality uses the probabilistic Jensen's inequality. Note that here we only get α instead of α^2 as before. This is because the randomness in the sampling i of $\nabla f_i(\boldsymbol{x}^{t-1})$ is not independent of the second term $\mathbb{E}_{\xi^{t-1}}[\boldsymbol{m}_i^{t-1}] - \mathbb{E}_{\xi^{t-1}}[\bar{\boldsymbol{m}}^{t-1}]$. Expanding this we get,

$$\mathbb{E}_i\left\|\mathbb{E}_{\xi^t}[\boldsymbol{m}_i^t] - \mathbb{E}_{\xi^t}[\bar{\boldsymbol{m}}^t]\right\|^2 \leq \zeta^2(1-(1-\alpha)^t) + \sum_{k=1}^{t}(1-\alpha)^{t-k}\alpha B^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{k-1})\|^2\,.$$

We can combine all three bounds above as

$$\begin{aligned}\mathbb{E}\|\boldsymbol{m}_i^t - \bar{\boldsymbol{m}}^t\|^2 &\leq 3\,\mathbb{E}\|\boldsymbol{m}_i^t - \mathbb{E}_{\xi^t}[\boldsymbol{m}_i^t]\|^2 + 3\,\mathbb{E}\|\bar{\boldsymbol{m}}^t - \mathbb{E}_{\xi^t}[\bar{\boldsymbol{m}}^t]\|^2 + 3\,\mathbb{E}_i\|\mathbb{E}_{\xi^t}[\boldsymbol{m}_i^t] - \mathbb{E}_{\xi^t}[\bar{\boldsymbol{m}}^t]\|^2 \\ &= 3\,\mathbb{E}_i\,\mathbb{E}_{\xi^t}\|\boldsymbol{m}_i^t - \mathbb{E}_{\xi^t}[\boldsymbol{m}_i^t]\|^2 + 3\,\mathbb{E}_{\xi^t}\|\bar{\boldsymbol{m}}^t - \mathbb{E}_{\xi^t}[\bar{\boldsymbol{m}}^t]\|^2 + 3\,\mathbb{E}_i\|\mathbb{E}_{\xi^t}[\boldsymbol{m}_i^t] - \mathbb{E}_{\xi^t}[\bar{\boldsymbol{m}}^t]\|^2 \\ &\leq (6\alpha\sigma^2 + 3\zeta^2) + (6\sigma^2 - 3\zeta^2)(1-\alpha)^t + 3\sum_{k=1}^{t}(1-\alpha)^{t-k}\alpha B^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{k-1})\|^2\,.\end{aligned}$$

Therefore for $i, j \in \mathcal{G}$

$$\begin{aligned}\mathbb{E}\|\boldsymbol{m}_i^t - \boldsymbol{m}_j^t\|^2 &\leq 2\,\mathbb{E}\|\boldsymbol{m}_i^t - \bar{\boldsymbol{m}}^t\|^2 + 2\,\mathbb{E}\|\boldsymbol{m}_j^t - \bar{\boldsymbol{m}}^t\|^2 \\ &\leq 4(6\alpha\sigma^2 + 3\zeta^2) + 4(6\sigma^2 - 3\zeta^2)(1-\alpha)^t + 12\sum_{k=1}^{t}(1-\alpha)^{t-k}\alpha B^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{k-1})\|^2\,.\end{aligned}$$

Recall that the right hand side was defined to be ρ_t^2. Using Definition A, we can show that the output of the aggregation rule ARAGG satisfies the condition in the lemma. □

One major caveat in the above lemma is that here ρ^2 cannot be known to the robust aggregation since it involves $\mathbb{E}\|\nabla f(\boldsymbol{x}^{k-1})\|^2$ whose value we do not have access to. However, this does not present a hurdle to *agnostic* aggregation rules which are automatically adaptive to the value of ρ^2. Deriving a similarly provable adaptive clipping method is a very important open problem.

Lemma 9 (Descent bound)**.** *For any $\alpha \in [0,1]$ for $t \geq 2$, $\eta \leq \frac{1}{L}$, and an L-smooth function f we have for any $t \geq 1$*

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

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,

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

Here we used the identities that $-2ab = (a-b)^2 - a^2 - b^2$, and Young's inequality that $(a+b)^2 \leq (1+\gamma)a^2 + (1+\frac{1}{\gamma})b^2$ for any positive constant $\gamma \geq 0$ (here we used $\gamma = 1$). Taking conditional expectation on both sides yields the lemma. □

Lemma 10 (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. Let us define $\bar{\boldsymbol{g}}(\boldsymbol{x}) := \frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}} \boldsymbol{g}_i(\boldsymbol{x})$. This implies that

$$\mathbb{E}\|\bar{\boldsymbol{g}}(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2 \leq \frac{\sigma^2}{|\mathcal{G}|} \leq \frac{2\sigma^2}{n} \,.$$

Then by definition of $\bar{\boldsymbol{m}}$, we can expand the error as:

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

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

□

Proof of Theorem V. Scale the error bound Lemma 10 by $\frac{5\eta}{2\alpha}$ and add it to the descent bound Lemma 9 taking expectations on both sides to get for $t \geq 2$

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

Now, let use the aggregation error Lemma 8 to bound $\mathbb{E}\|\boldsymbol{m}^{t-1} - \bar{\boldsymbol{m}}^{t-1}\|^2$ and $\mathbb{E}\|\boldsymbol{m}^t - \bar{\boldsymbol{m}}^t\|^2$ in the above expression to get

$$\begin{aligned}\mathbb{E}[f(\boldsymbol{x}^t)] + \tfrac{5\eta}{2\alpha}\,\mathbb{E}\|\bar{\boldsymbol{e}}^t\|^2 \leq \mathbb{E}[f(\boldsymbol{x}^{t-1})] - \tfrac{\eta}{2}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \eta\,\mathbb{E}\|\bar{\boldsymbol{e}}^t\|^2 \\ + \tfrac{5\eta}{2\alpha}\,\mathbb{E}\|\bar{\boldsymbol{e}}^{t-1}\|^2 - \eta\,\mathbb{E}\|\bar{\boldsymbol{e}}^{t-1}\|^2 + \tfrac{\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-2})\|^2 + 5\eta\alpha\tfrac{\sigma^2}{n} \\ + 5\eta c\delta((6\alpha\sigma^2 + 3\zeta^2) + 6\sigma^2(1-\alpha)^{t-2}) \\ + \eta c\delta\left(3\sum_{k=1}^{t-1}(1-\alpha)^{t-1-k}\alpha B^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{k-1})\|^2\right) \\ + 4\eta c\delta\left(3\sum_{k=1}^{t}(1-\alpha)^{t-k}\alpha B^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{k-1})\|^2\right).\end{aligned}$$

Let us define $S_t := \sum_{k=1}^{t}(1-\alpha)^{t-k}\alpha B^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{k-1})\|^2$. Then, S_t satisfies the recursion:

$$\tfrac{1}{\alpha}S_t = (\tfrac{1}{\alpha} - 1)S_{t-1} + B^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2\,.$$

Adding $\frac{3\eta c\delta(\frac{5}{\alpha}-4)}{\alpha}S_t$ on both sides to the bound above and rearranging gives the following for $t \geq 2$

$$\begin{aligned}&\underbrace{\mathbb{E}\;f(\boldsymbol{x}^t) - f^\star + (\tfrac{5\eta}{2\alpha} - \eta)\,\mathbb{E}\|\bar{\boldsymbol{e}}^t\|^2 + \frac{\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{3\eta c\delta(\frac{5}{\alpha}-4)}{\alpha}S_t}_{=:\mathcal{E}_t} \\ &\quad\leq \underbrace{\mathbb{E}\;f(\boldsymbol{x}^{t-1}) - f^\star + (\tfrac{5\eta}{2\alpha} - \eta)\,\mathbb{E}\|\bar{\boldsymbol{e}}^{t-1}\|^2 + \frac{\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-2})\|^2 + \frac{3\eta c\delta(\frac{5}{\alpha}-4)}{\alpha}S_{t-1}}_{=:\mathcal{E}_{t-1}} \\ &\qquad(-\tfrac{\eta}{4} + 15\eta c\delta B^2)\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 \\ &\qquad+ \frac{5\eta\alpha}{n}\sigma^2 + 5\eta c\delta\big((6\alpha\sigma^2 + 3\zeta^2) + 6\sigma^2(1-\alpha)^{t-2}\big) \\ &\quad\leq \mathcal{E}_{t-1} - \frac{\eta}{4}(1 - 60c\delta B^2)\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 \\ &\qquad+ \underbrace{5\eta\alpha\sigma^2\big(\tfrac{1}{n} + 6c\delta(1 + \tfrac{1}{\alpha}(1-\alpha)^{t-2})\big) + 15\eta c\delta\zeta^2}_{=:\eta\xi_{t-1}^2}\,.\end{aligned}$$

Further, specializing the descent bound Lemma 9 and error bound Lemma 10 for $t = 1$ we have

$$\begin{aligned}\mathcal{E}_1 &= \mathbb{E}\;f(\boldsymbol{x}^1) - f^\star + \frac{3\eta}{2}\,\mathbb{E}\|\bar{e}^1\|^2 + \frac{\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^0)\|^2 + 3\eta c\delta B^2(\frac{5}{\alpha} - 4)\|\nabla f(\boldsymbol{x}^0)\|^2 \\ &\leq f(\boldsymbol{x}^0) - f^\star + \frac{5\eta}{2}\,\mathbb{E}\|\bar{e}^1\|^2 - \frac{\eta}{4}(1 - 60c\delta B^2)\,\mathbb{E}\|\nabla f(\boldsymbol{x}^0)\|^2 + \eta\,\mathbb{E}\|\boldsymbol{m}_1 - \bar{\boldsymbol{m}}_1\|^2 \\ &\leq f(\boldsymbol{x}^0) - f^\star - \frac{\eta}{4}(1 - 60c\delta B^2)\,\mathbb{E}\|\nabla f(\boldsymbol{x}^0)\|^2 + \frac{5\eta\sigma^2}{n} + 12c\delta\eta(2\sigma^2 + \zeta^2 + B^2\|\nabla f(\boldsymbol{x}^0)\|^2) \\ &= f(\boldsymbol{x}^0) - f^\star - \frac{\eta}{4}(1 - 60c\delta B^2)\,\mathbb{E}\|\nabla f(\boldsymbol{x}^0)\|^2 + \eta\xi_0^2\,.\end{aligned}$$

Above, we defined $\xi_0^2 := \frac{5\sigma^2}{n} + 12c\delta(2\sigma^2 + \zeta^2 + B^2\|\nabla f(\boldsymbol{x}^0)\|^2)$. Summing over t from 2 until T, again rearranging our recursion for $\mathcal{E}_t$, and adding $(1 - 3c\delta B^2)\,\mathbb{E}\|\nabla f(\boldsymbol{x}^0)\|^2$ on both sides gives

$$
\begin{aligned}
(1 - 60c\delta B^2)\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 &\leq \frac{4(f(\boldsymbol{x}^0) - f^\star)}{\eta T} + \frac{1}{T}\sum_{t=1}^{T} 4\xi_{t-1}^2 \\
&= \frac{4(f(\boldsymbol{x}^0) - f^\star)}{\eta T} + \frac{4\xi_0^2}{T} \\
&\qquad + \frac{1}{T}\sum_{t=2}^{T} 20\alpha\sigma^2\big(\tfrac{1}{n} + 6c\delta(1 + \tfrac{1}{\alpha}(1-\alpha)^{t-2})\big) \\
&\qquad + \frac{1}{T}\sum_{t=2}^{T} 60c\delta\zeta^2 \\
&\leq \frac{4(f(\boldsymbol{x}^0) - f^\star)}{\eta T} + \frac{4\xi_0^2}{T} + 60c\delta\zeta^2 \\
&\qquad + 20\alpha\sigma^2\big(\tfrac{1}{n} + 6c\delta\big) + \frac{120c\delta\sigma^2}{\alpha T} \\
&= \frac{4(f(\boldsymbol{x}^0) - f^\star)}{\eta T} + \frac{120c\delta\sigma^2}{\eta 8LT} + \eta 160L\sigma^2\big(\tfrac{1}{n} + 6c\delta\big) \\
&\qquad + \frac{4\xi_0^2}{T} + 60c\delta\zeta^2.
\end{aligned}
$$

The last equality substituted the value of $\alpha = 8L\eta$. Next, let us use the appropriate step-size of

$$
\eta = \min\left(\sqrt{\frac{4(f(\boldsymbol{x}_0) - f^\star) + \frac{15c\delta}{L}(\zeta^2 + 2\sigma^2)}{T(160L\sigma^2)\big(\frac{1}{n} + 6c\delta\big)}}, \frac{1}{8L}\right).
$$

This gives the following final rate of convergence:

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 \\
&\qquad \leq \frac{1}{1 - 60c\delta B^2}\cdot\Bigg(60c\delta\zeta^2 + \sqrt{\frac{160L\sigma^2\big(\frac{1}{n} + 6c\delta\big)}{T}}\cdot\sqrt{4(f(\boldsymbol{x}_0) - f^\star) + \tfrac{15c\delta}{L}(\zeta^2 + 2\sigma^2)} \\
&\qquad\qquad + \frac{32L(f(\boldsymbol{x}^0) - f^\star)}{T} + \frac{15c\delta\sigma^2}{T} \\
&\qquad\qquad + \frac{\frac{20\sigma^2}{n} + 12c\delta(2\sigma^2 + \zeta^2 + B^2\|\nabla f(\boldsymbol{x}^0)\|^2)}{T}\Bigg).
\end{aligned}
$$

□

F UPDATES WITH RESPECT TO REVIEWS.

F.1 ADDITIONAL RELATED WORK

In this section, we add comments on works which are very close to this paper.

- Li et al. (2019) propose RSA for Byzantine-resilient distributed learning on heterogeneous data. They introduce an additional ℓ_p-norm regularized term to the objective to penalize the difference between local iterates and server iterate and show convergence of RSA for strongly convex local objectives and penalized term. However, RSA cannot defend the state-of-the-art attacks like (Baruch et al., 2019; Xie et al., 2020) as they didn't utilize the temporal information. Compared to RSA, our method does not assume strongly convexity and we consider more general cost functions with no explicit regularized term. In addition, our method is shown to defend the state-of-the-art attacks.

- (Yang & Li, 2021) is a parallel work which uses buffer for asynchronous Byzantine-resilient training (BASGD). The buffer and bucketing are similar techniques with vastly different motivations. The key difference between buffer and bucketing is that buffer is only reassigned when timer exceeds a threshold while bucketing reshuffles in each iteration. Therefore, buffer does not guarantee that partial aggregated gradients are identically distributed while bucketing does. In addition, our theoretical analysis does not require bounded gradient assumption $\|\nabla f(\boldsymbol{x})\| \leq D$ for all $\boldsymbol{x}$.
- Wu et al. (2020) uses ByrdSAGA for Byzantine-resilient SAGA approach for distributed learning. The key differences between ByrdSAGA and our work are as follows. In our setting, there are two sources of variances of the gradients - intra-worker variance σ^2 and inter-client variance ζ^2. We show that simply using worker momentum suffices to tackle the former and handling the latter ζ^2 is the main challenge. ByrdSAGA assumes that each worker only has finite data points as opposed to the stochastic setting we consider. Hence they can use SAGA on the worker in place of worker momentum to reduce the intra-client variance σ^2. The effect of ζ^2 (which is our main focus) remains unaffected.

  Further, they consider the strongly convex setting whereas we analyze non-convex functions. Ignoring μ for sake of comparison, their Theorem 1 proves convergence to a radius of $\Delta_1 = O(\zeta^2)$ since always $C_\alpha \geq 2$. Thus, their rates are similar to (Acharya et al., 2021) and do not converge to the optimum even when $\delta = 0$. In contrast, our Theorem II proves convergence to a radius of $O(\delta\zeta^2)$. We believe our improved handling of ζ^2 can be combined with their usage of SAGA/variance reduction to yield even faster rates. This we leave for future work.