Byzantine-Robust Optimization under (L_0, L_1)-Smoothness

Arman Bolatov[1], Samuel Horváth[1], Martin Takáč[1], Eduard Gorbunov[1]
[1]Mohamed bin Zayed University of Artificial Intelligence, UAE
arman.bolatov@mbzuai.ac.ae, samuel.horvath@mbzuai.ac.ae,
martin.takac@mbzuai.ac.ae, eduard.gorbunov@mbzuai.ac.ae

We consider distributed optimization under Byzantine attacks in the presence of (L_0, L_1)-smoothness, a generalization of standard L-smoothness that captures functions with state-dependent gradient Lipschitz constants. We propose Byz-NSGDM[1], a normalized stochastic gradient descent method with momentum that achieves robustness against Byzantine workers while maintaining convergence guarantees. Our algorithm combines momentum normalization with Byzantine-robust aggregation enhanced by Nearest Neighbor Mixing (NNM) to handle both the challenges posed by (L_0, L_1)-smoothness and Byzantine adversaries. We prove that Byz-NSGDM achieves a convergence rate of $O(K^{-1/4})$ up to a Byzantine bias floor proportional to the robustness coefficient and gradient heterogeneity. Experimental validation on heterogeneous MNIST classification, synthetic (L_0, L_1)-smooth optimization, and character-level language modeling with a small GPT model demonstrates the effectiveness of our approach against various Byzantine attack strategies. An ablation study further shows that Byz-NSGDM is robust across a wide range of momentum and learning rate choices.

1. Introduction

Modern machine learning systems increasingly rely on distributed optimization to train models on data distributed across multiple workers [2–10]. However, such systems face two critical challenges: Byzantine failures [11–13], where some workers may send arbitrary or malicious updates, and data heterogeneity, where different workers possess non-identically distributed data [14–16]. These challenges are particularly acute in federated learning settings where data cannot be centralized due to privacy constraints or communication costs.

Traditional Byzantine-robust optimization algorithms assume standard L-smoothness, where the gradient Lipschitz constant is globally bounded, and rely on robust aggregation rules such as Krum, geometric median (RFA), coordinate-wise median, or trimmed mean [17–19]. However, many modern machine learning objectives, including neural networks with certain activation functions and high-degree polynomial losses, exhibit (L_0, L_1)-smoothness [20] – a generalized smoothness condition where the Lipschitz constant depends on the current iterate. This state-dependent behavior can cause traditional Byzantine-robust methods to diverge or converge slowly.

Recent Byzantine-robust methods have moved beyond simple robust aggregation to incorporate advanced algorithmic techniques. Variance reduction methods [21–23] use control variates to suppress stochastic noise that Byzantine workers exploit, achieving tighter complexity bounds and removing restrictive assumptions such as gradient boundedness. Client momentum techniques [24–26] accumulate updates over time at the worker level, reducing variance of honest clients and

[1]During the preparation of the camera-ready version, we became aware of concurrent work by Yang et al. [1] that studies a closely related algorithmic framework and also derives an $O(K^{-1/4})$ convergence rate. We note that all theoretical and experimental results in our paper were obtained independently and prior to our awareness of this work. Their analysis is conducted under the L-smoothness assumption and assumes homogeneous data across workers, whereas our results hold under the more general (L_0, L_1)-smoothness condition and accommodate bounded gradient heterogeneity among workers.

Third Conference on Parsimony and Learning (CPAL 2026).

exposing Byzantine perturbations that are imperceptible in single rounds but accumulate across iterations. Meanwhile, the (L_0, L_1)-smoothness assumption has received significant attention in *non-Byzantine* settings, with specialized algorithms including gradient clipping [27, 28], normalized gradient descent [29–32], and adaptive methods [33–42].

Contributions. We study Byz-NSGDM (Byzantine-Robust Normalized Stochastic Gradient Descent with Momentum) in the regime of (L_0, L_1)-smooth objectives. While normalized momentum combined with Byzantine-robust aggregation has been considered previously under classical L-smoothness assumptions [1], its behavior under generalized (L_0, L_1)-smoothness and gradient heterogeneity has not been theoretically understood. We fill this gap by providing a convergence analysis of this algorithmic framework in the presence of state-dependent smoothness and heterogeneous data across workers.

Our analysis builds on recent advances in optimization under (L_0, L_1)-smoothness, where controlling the step magnitude via normalization is crucial for stability [29–32]. We show that, when combined with a (δ, κ)-robust aggregation rule and optionally enhanced by Nearest Neighbor Mixing (NNM) [43] to mitigate heterogeneity, Byz-NSGDM achieves non-asymptotic convergence guarantees under Byzantine attacks. In particular, we establish an $O(K^{-1/4})$ convergence rate up to a Byzantine bias floor that depends explicitly on the robustness coefficient and the level of gradient heterogeneity.

Empirically, we evaluate Byz-NSGDM on three complementary tasks: heterogeneous MNIST classification with non-IID data partitioning, a synthetic (L_0, L_1)-smooth quartic optimization problem, and character-level language modeling with a small GPT architecture. Across multiple Byzantine attack strategies (Bit Flipping, Label Flipping, Mimic, ALIE) and aggregation rules (Krum [17], RFA [18], and Coordinate-wise Median/Trimmed Mean [19]), the method demonstrates strong robustness compared to momentum-based baselines without normalization. We additionally provide an ablation study showing stability across a broad range of learning rates and momentum parameters.

2. Assumptions and Problem Setup

Problem. We consider a distributed optimization problem of the form

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{G} \sum_{i \in \mathcal{G}} f_i(x) \right\},$$

where $\mathcal{G}$ is a group of regular workers of size $G := |\mathcal{G}|$ connected with a server in a centralized way, and $f_i : \mathbb{R}^d \to \mathbb{R}$ is the objective/loss function on worker i. However, in addition to the regular workers, some Byzantine workers take part in the training. More precisely, we assume that the total number of workers is n and $[n] := \{1, \ldots, n\} = \mathcal{G} \cup \mathcal{B}$, $\mathcal{G} \cap \mathcal{B} = \varnothing$, where $\mathcal{B}$ represents the set of Byzantine workers. We assume that $B := |\mathcal{B}| \leq \delta n$ for some $\delta \leq 1/2$. We do not make any further assumptions on the clients from $\mathcal{B}$, i.e., they can arbitrarily deviate from the optimization protocol, collude, and even know the updates that other clients send to the server.

Assumptions. We state the standing assumptions and the robustness notion used throughout.

Assumption 1 (Lower boundedness)**.** *We assume that f is lower bounded by $f^* > -\infty$ and each local objective f_i is lower bounded by $f_i^* > -\infty$ for any $i \in \mathcal{G}$.*

Assumption 2 ((L_0, L_1)-smoothness [20, 27, 30])**.** *For all $i \in \mathcal{G}$, the function f_i is (L_0, L_1)-smooth with $L_0, L_1 > 0$: for all $x, y \in \mathbb{R}^d$,*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \;\leq\; \Big(L_0 + L_1 \sup_{u \in [x,y]} \|\nabla f_i(u)\|\Big) \|x - y\|.$$

Assumption 3 (Unbiased oracle with bounded variance)**.** *Each good worker $i \in \mathcal{G}$ has access to an unbiased stochastic first-order oracle: for any $x \in \mathbb{R}^d$, worker i can compute $\nabla f_{\xi_i}(x)$ such that*

$$\mathbb{E}[\nabla f_{\xi_i}(x)] \;=\; \nabla f_i(x), \qquad \mathbb{E}\big[\|\nabla f_{\xi_i}(x) - \nabla f_i(x)\|^2\big] \;\leq\; \sigma^2.$$

Assumption 4 (Gradient heterogeneity)**.** *The gradient heterogeneity across good workers is bounded for all $x \in \mathbb{R}^d$:*

$$\mathbb{E}_{i\sim\mathcal{G}}\left[\|\nabla f_i(x) - \nabla f(x)\|^2\right] \;\le\; \zeta^2. \tag{1}$$

Definition 1 (Byzantine-robust aggregator)**.** *Let $\kappa \ge 0$. An aggregation rule $\texttt{ARAgg} : \mathbb{R}^{d\times n} \to \mathbb{R}^d$ is (δ,κ)-robust if for any vectors $v_1,\dots,v_n \in \mathbb{R}^d$ with at most $B \le \delta n$ Byzantine workers (so $G = n - B \ge (1-\delta)n > n/2$ good workers) and $\bar{v}_{\mathcal{G}} := \frac{1}{G}\sum_{i\in\mathcal{G}} v_i$,*

$$\left\|\texttt{ARAgg}(v_1,\dots,v_n) - \bar{v}_{\mathcal{G}}\right\| \;\le\; \frac{\kappa}{G}\sum_{i\in\mathcal{G}}\left\|v_i - \bar{v}_{\mathcal{G}}\right\|.$$

We refer to κ as the robustness coefficient.

The above definition is a modification of the one proposed by Allouah et al. [43]. The key difference is in the usage of a linear (non-squared) bound on the right-hand side, which leads to different values of κ but is more natural for our analysis under (L_0, L_1)-smoothness. Many standard Byzantine-robust aggregation rules satisfy this definition, including geometric median (RFA) and coordinate-wise median, as well as their composition with NNM. We provide explicit formulas for the robustness coefficient κ for these aggregators in Appendix C.

3. Related Work

Our work lies at the intersection of Byzantine-robust distributed optimization and optimization under generalized smoothness. We review the relevant literature in both areas.

Byzantine-robust distributed optimization. The problem of Byzantine-robust distributed learning has received considerable attention. Early works established fundamental aggregation rules: Krum and Multi-Krum [17, 44] select gradients based on distance metrics, geometric median (RFA) [18] provides a robust centroid estimator, and coordinate-wise median/trimmed mean [19, 45] offer dimension-wise robustness. These methods provide statistical guarantees under i.i.d. data but suffer from bias terms that grow with data heterogeneity [19, 45]. However, simply applying robust aggregation is insufficient: carefully crafted attacks defeat many standard defenses [46, 47], and Karimireddy et al. [25] show that permutation-invariant schemes cannot guarantee convergence against arbitrary Byzantines without additional structure.

To address heterogeneity and improve robustness, several complementary approaches have emerged. Structural methods include *bucketing* [24], which partitions workers into homogeneous groups before aggregation, and *nearest neighbor mixing* (*NNM*) [43], which averages updates with nearest neighbors to contract heterogeneity. Recent theoretical analyses [43, 48] characterize optimal error bounds and breakdown points under data heterogeneity. Crucially, achieving tight convergence guarantees requires refined algorithmic techniques beyond robust aggregation alone. Variance reduction methods [21–23] incorporate control variates to suppress stochastic noise that Byzantine workers exploit, achieving faster rates under weaker assumptions. Client momentum techniques [24–26] accumulate worker updates over time, reducing variance of honest clients and exposing Byzantine perturbations that are undetectable in single rounds but accumulate over iterations. These works demonstrate that stabilizing local estimators before applying robust aggregation is crucial for Byzantine resilience.

(L_0, L_1)-smoothness in optimization. The (L_0, L_1)-smoothness assumption was introduced by Zhang et al. [49] to capture the empirical observation that the local smoothness constant along training trajectories of deep learning models often scales linearly with the gradient norm. Formally, for twice differentiable functions, this condition requires $\|\nabla^2 f(x)\|_2 \le L_0 + L_1\|\nabla f(x)\|$, which strictly generalizes classical L-smoothness (recovered when $L_1 = 0$). Functions satisfying (L_0, L_1)-smoothness but not L-smoothness include $f(x) = \|x\|^{2n}$ for $n \ge 2$, $f(x) = \exp(\langle a, x\rangle)$, and more generally, objectives with polynomially growing gradients [30, 49].

In the nonconvex setting, Zhang et al. [49] showed that gradient clipping achieves complexity $O(\max\{L_0\Delta/\varepsilon^2, (1+L_1^2)\Delta/L_0\})$ for finding ε-stationary points, where the dominant term is inde-

pendent of L_1. This result was extended to momentum-based clipping [27], normalized gradient descent [29, 30], SignGD [50], and adaptive methods like AdaGrad and Adam [33, 51, 52]. In the convex case, Koloskova et al. [28] and Gorbunov et al. [34] derived convergence rates for clipped gradient descent, accelerated methods, and adaptive stepsizes. Notably, recent works [34, 53] provide bounds that avoid exponentially large factors in $L_1 R_0$ that plagued earlier analyses in the convex case. Normalized gradient descent with momentum [31, 32] achieves stable convergence by controlling step magnitudes, making it well-suited for (L_0, L_1)-smooth problems.

However, all existing work on (L_0, L_1)-smooth optimization assumes benign (non-Byzantine) settings. Our work is the first to address Byzantine-robust optimization under (L_0, L_1)-smoothness, combining normalized gradient descent with momentum and robust aggregation to handle both challenges simultaneously.

4. New Method: Byz-NSGDM

We introduce Byz-NSGDM (Byzantine-Robust Normalized Stochastic Gradient Descent with Momentum), a novel algorithm designed for (L_0, L_1)-smooth optimization under Byzantine attacks.

Algorithm 1 Byz-NSGDM

Input: starting point $x^0 \in \mathbb{R}^d$, initial momentum vectors $v_1^0, \ldots, v_n^0 \in \mathbb{R}^d$, stepsizes $\gamma_k > 0$, momentum parameters $\eta_k \in (0, 1]$

1: **for** $k = 1, 2, \ldots, K$ **do**
2: **for** $i \in [n]$ **do**
3: Worker i sends:

$$v_i^k = \begin{cases} (1-\eta_{k-1}) v_i^{k-1} + \eta_{k-1} \nabla f_{\xi_i^{k-1}}(x^{k-1}), & i \in \mathcal{G} \\ \text{arbitrary } \tilde{v}_i^k \in \mathbb{R}^d, & i \in \mathcal{B} \end{cases}$$

4: **end for**
5: Server computes $v^k = \texttt{ARAgg}(v_1^k, \ldots, v_n^k)$
6: Server updates $x^k = x^{k-1} - \gamma_{k-1} \frac{v^k}{\|v^k\|}$ and broadcasts x^k
7: **end for**

The central component of the method is the normalization step in line 6:

$$x^k = x^{k-1} - \gamma_{k-1} \frac{v^k}{\|v^k\|}.$$

By rescaling the aggregated momentum vector to unit norm, the update ensures that the step magnitude is always bounded by γ_{k-1}. Such boundedness is crucial under (L_0, L_1)-smoothness, where the local Lipschitz constant depends on the gradient norm and can otherwise lead to instability.

In Algorithm 1, the server employs an aggregation rule ARAgg assumed to be (δ, κ)-robust in the sense of Definition 1. Our convergence analysis is agnostic to the specific aggregation mechanism and applies to any rule satisfying this robustness property. In the Appendix, we show that several standard Byzantine-robust aggregation schemes — such as coordinate-wise median, geometric median, and their composition with NNM — meet Definition 1, and we provide the corresponding robustness coefficients κ. Consequently, the theoretical guarantees hold for a broad class of (δ, κ)-robust aggregators, independently of whether additional preprocessing steps, such as NNM, are employed.

Remark 1. Byz-NSGDM *naturally extends to a broader class of optimizers based on a Linear Minimization Oracle (LMO) [54, 55]. Specifically, line 6 of Algorithm 1 can be replaced by*

$$d_k = \operatorname*{argmin}_{d \in \mathbb{R}^d : \|d\| \le 1} \langle v^k, d \rangle, \qquad x^k = x^{k-1} + \gamma_{k-1} d_k, \tag{Byz-LMO}$$

where $\|\cdot\|$ denotes an arbitrary (not necessarily Euclidean) norm. For the standard Euclidean norm, the LMO solution satisfies $d_k = -v^k/\|v^k\|$, and the method reduces to Byz-NSGDM. *If $\|\cdot\|$ is chosen as the operator norm (when x^k is a matrix), the resulting algorithm corresponds to a Byzantine-robust variant of* Muon *[56], which has recently gained significant attention. Moreover, recent analyses of LMO-based methods [54, 55], which are closely related to the analysis of normalized momentum methods [57], suggest that our convergence guarantees can be extended to general norms using similar arguments.*

5. Convergence Analysis

Our main theoretical result establishes the convergence rate of Byz-NSGDM under Byzantine attacks. Full proofs are deferred to the supplementary materials.

Theorem 1 (Convergence of Byz-NSGDM)**.** *Let Assumptions 1–4 hold and let the server use a (δ, κ)-robust aggregator. Run Algorithm 1 with*

$$\gamma_k \equiv \gamma = \frac{\gamma_0}{(K+1)^{3/4}}, \qquad \eta_k \equiv \eta = \frac{1}{(K+1)^{1/2}},$$

where

$$\gamma_0 \leq \min\left\{\frac{1}{2L_1}, \frac{(K+1)^{1/4}}{\sqrt{32}L_1}, \frac{1}{\sqrt{128(1+2\kappa)}L_1}\right\}. \tag{2}$$

Define $\Delta^* := \frac{1}{G}\sum_{i\in\mathcal{G}}(f^* - f_i^*)$ and $V_0 := \frac{1}{G}\sum_{i\in\mathcal{G}}\mathbb{E}\big[\|v_i^0 - \nabla f_i(x^0)\|\big]$. Then

$$\begin{aligned}\min_{0\leq k\leq K}\mathbb{E}\big[\|\nabla f(x^k)\|\big] \leq{}& \frac{e^{1/2}\big(f(x^0)-f^*\big)}{\gamma_0(K+1)^{1/4}} + \frac{2e^{1/2}(1+2\kappa)V_0}{(K+1)^{1/2}} + 4\kappa\zeta \\ &+ \frac{2L_0\gamma_0}{(K+1)^{3/4}} + \frac{8(1+2\kappa)L_0\gamma_0}{(K+1)^{1/4}} \\ &+ \frac{8L_1^2\Delta^*\gamma_0}{(K+1)^{3/4}} + \frac{32(1+2\kappa)L_1^2\Delta^*\gamma_0}{(K+1)^{1/4}} \\ &+ \frac{2\sigma}{\sqrt{G}(K+1)^{1/4}} + \frac{4\kappa\sigma}{(K+1)^{1/4}}.\end{aligned}$$

Remark 2 (Fixed-horizon schedule)**.** *The parameters γ and η depend on the total number of iterations K but remain constant across iterations. This is a standard "fixed-horizon" choice common in convergence analyses. It can be converted to a diminishing schedule that does not require knowledge of K in advance via standard doubling tricks [58].*

Corollary 1. *Under the conditions of Theorem 1, all terms except $4\kappa\zeta$ decay polynomially with K. Hence,*

$$\min_{0\leq k\leq K}\mathbb{E}\big[\|\nabla f(x^k)\|\big] \;\leq\; O\left(\frac{1}{K^{1/4}}\right) + 4\kappa\zeta.$$

The term $4\kappa\zeta$ is the irreducible Byzantine bias floor—a fundamental limit arising from the interaction between Byzantine fraction (through κ) and data heterogeneity (ζ). In homogeneous settings ($\zeta = 0$), this bias vanishes entirely.

Proof roadmap. Our analysis proceeds in three steps:

1. **Descent under normalization (Lemma 5).** We show that the normalized update $x^{k+1} = x^k - \gamma_k \frac{v^k}{\|v^k\|}$ yields a descent inequality:

   $$f(x^{k+1}) \leq f(x^k) - \gamma_k\|\nabla f(x^k)\| + 2\gamma_k\|v^k - \nabla f(x^k)\| + O(\gamma_k^2),$$

   where progress depends on $\|\nabla f(x^k)\|$ minus an error term from the momentum-to-gradient gap $\|v^k - \nabla f(x^k)\|$. The normalization ensures the step size is bounded by γ_k, preventing the (L_0, L_1)-smoothness from causing divergence.

2. **Bounding the momentum error (Lemmas 4 and 7).** We decompose this gap into two parts: (i) the averaging error $\|\bar{v}^k - \nabla f(x^k)\|$ from honest workers' momentum tracking (Lemma 4), and (ii) the Byzantine-induced error $\|v^k - \bar{v}^k\|$ controlled by the (δ, κ)-robust aggregator (Lemma 7).

3. **Telescoping.** Summing the descent inequalities with carefully chosen exponential weights absorbs the coupling between function values and momentum errors, yielding the final rate.

The key insight is that normalization step $x^{k+1} = x^k - \gamma_k \frac{v^k}{\|v^k\|}$ bounds the step size by γ_k, preventing the (L_0, L_1)-smoothness from causing divergence, while the robust aggregator limits Byzantine influence to a bias floor of $O(\kappa\zeta)$.

Remark 3. *Following the analysis of* NSGDM *by Hübler et al.* [59]*, our convergence proof can be extended to the setting of heavy-tailed stochastic noise* [49, 60]*. In particular, Assumption 3 can be relaxed to the weaker p-th moment condition*

$$\mathbb{E}[\nabla f_{\xi_i}(x)] = \nabla f_i(x), \qquad \mathbb{E}\big[\|\nabla f_{\xi_i}(x) - \nabla f_i(x)\|^p\big] \le \sigma^p,$$

for some $p \in (1, 2]$.

Under this assumption, Lemma 3 must be adapted using arguments analogous to those in Lemma 16 of Hübler et al. [59]*. The remainder of the convergence analysis then proceeds along the same lines as in the current proof, with appropriate modifications to the stepsize γ and momentum parameter η to account for the heavier-tailed noise.*

6. Experiments

We evaluate Byz-NSGDM on three distinct problems: a heterogeneous distributed learning task on MNIST, character-level language modeling, and a synthetic (L_0, L_1)-smooth optimization problem designed to validate our theoretical framework.

6.1. Attack Strategies

We evaluate our algorithm against several Byzantine attack strategies that represent different threat models in distributed learning, following common setups in prior work on non-IID Byzantine robustness [24, 43].

Bit/Sign Flipping (BF). Byzantine workers send the negation of their computed gradient: $g_i^k = -\nabla f_i(x^k)$ for $i \in \mathcal{B}$. This attack directly opposes the optimization direction and tests the algorithm's ability to handle adversarial gradients [17, 19].

Label Flipping (LF). For classification tasks, Byzantine workers flip the labels of their training data before computing gradients. Each sample with true label y is trained with label $(y + c) \bmod C$, where C is the number of classes. For MNIST with 10 classes, we use a shift of $c = 5$. This creates consistent but misaligned gradient directions and is widely used to evaluate robustness under data poisoning and non-IID heterogeneity [24].

Mimic Attack. Byzantine workers exploit data heterogeneity by initially behaving honestly for a warmup period of 50 iterations, then strategically deviating. After warmup, they send $g_i^k = -2\bar{g}^k$, where $\bar{g}^k$ is the average of honest gradients, aiming to reverse progress while avoiding early detection; see [43] for analysis of mixing defenses against such strategies.

ALIE (A Little Is Enough). Byzantine workers craft gradients to maximize damage while avoiding outlier detection. They send $v_i^k = \bar{v}^k + z \cdot \sigma^k$, where $\bar{v}^k = \frac{1}{G}\sum_{i\in\mathcal{G}} v_i^k$ and σ^k is the vector of coordinate-wise standard deviations of $\{v_i^k\}_{i\in\mathcal{G}}$ [46]. We use $z = 1$ in our experiments.

6.2. Experimental Setup

Heterogeneous MNIST Classification. We train a three-layer MLP with architecture (784, 128, 64, 10) on MNIST distributed across $n = 20$ workers with $B = 3$ Byzantine workers ($\delta = 0.15$). We create strong data heterogeneity using a non-IID partitioning scheme where data is first sorted by label and then divided sequentially among workers, ensuring each worker receives a highly skewed label distribution. Training uses cross-entropy loss with batch size 64 over 50 epochs, with each epoch limited to 30 batches, yielding 1,500 total iterations. We tune learning rates individually for each configuration and use momentum parameter $\beta_k := 1 - \eta_k = 0.9$ for all methods.

We compare three optimizer variants:

- **Baseline**: Standard momentum SGD with constant learning rate
- **Baseline-Decay**: Momentum SGD with learning rate decay $\gamma_k = \gamma_0/k^{0.5}$
- **Byz-NSGDM**: Our normalized gradient method with learning rate decay $\gamma_k = \gamma_0/k^{0.5}$

Remark 4 (Theory vs. practice step-schedules)**.** *Our theoretical analysis (Theorem 1) uses horizon-dependent schedules $\gamma \sim K^{-3/4}$ and $\eta \sim K^{-1/2}$ to derive clean convergence rates. In experiments, we use practical iteration-dependent schedules $\gamma_k \sim k^{-1/2}$ and constant $\beta = 1 - \eta = 0.9$, which are more common in deep learning. While the theoretical and practical schedules differ, our ablation study (Figure 1) confirms that performance is robust across momentum values in $[0.9, 0.99]$, and both schedules yield polynomial decay of the learning rate. Extending our analysis to iteration-dependent schedules with constant momentum is an interesting direction for future work.*

Each optimizer is combined with NNM pre-aggregation [43] followed by one of three robust aggregators: Krum [17], RFA (geometric median via smoothed Weiszfeld algorithm) [18], and Coordinate-wise Median/Trimmed Mean (CM) [19]. We test against three attacks: Bit Flipping (BF), Label Flipping (LF), and Mimic [43]. Each configuration is run with 3 random seeds, and we report mean accuracy with standard deviation bands.

Synthetic (L_0, L_1)-Smooth Optimization. We minimize the quartic objective:

$$f(x) = \|x\|^4,$$

where $x \in \mathbb{R}^{10}$ is initialized at $x^0 = [1, 1, \ldots, 1]^\top$. This function is (L_0, L_1)-smooth with $\nabla f(x) = 4x\|x\|^2$, exhibiting a gradient Lipschitz constant $L(x) = O(\|x\|^2)$ that grows with the squared iterate norm. We distribute the optimization across $n = 20$ workers with $B = 3$ Byzantine workers.

Each honest worker computes the true gradient with added Gaussian noise and a fixed worker-specific shift: $g_i = \nabla f(x) + \xi + s_i$ where $\xi \sim \mathcal{N}(0, 10^{-5}I)$ represents stochastic noise and $s_i \sim \mathcal{N}(0, 10^{-3}I)$ is a fixed shift sampled once at initialization for each honest worker $i \in \mathcal{G}$. To ensure unbiased gradients, the shifts are constructed so that $\sum_{i \in \mathcal{G}} s_i = 0$. The fixed shifts s_i model persistent heterogeneity across workers, ensuring each worker contributes a slightly different gradient direction even for the same iterate. We apply NNM pre-aggregation [43] with $n - B = 17$ nearest neighbors before robust aggregation. Training runs for 3,000 iterations. We test against three attacks: Bit Flipping (BF), Mimic, and ALIE. Each configuration uses 3 random seeds, and we report mean gradient norm evolution with standard deviation bands.

We compare the same three optimizer variants (Baseline, Baseline-Decay, Byz-NSGDM) combined with three aggregators (Krum [17], RFA [18], CM [19]). Learning rates are tuned per configuration by evaluating candidates on 1,000 iterations and selecting the rate that minimizes final gradient norm.

6.3. Results on Heterogeneous MNIST

Table 1 reports final test accuracy (mean $\pm$ std over 3 seeds) for each attack-aggregator-optimizer combination. Byz-NSGDM consistently achieves the highest accuracy under Bit Flipping (BF), reaching 86.0–86.1% compared to 81.8% for Baseline-Decay and 76.5–78.8% for Baseline. Under

Label Flipping (LF) and Mimic attacks, all methods perform comparably with accuracies in the 88–96% range. The low standard deviations for Byz-NSGDM under BF ($\leq 0.2\%$) indicate stable convergence across seeds. Full training curves are provided in Appendix D.

Table 1: Experimental results across all tasks. MNIST: test accuracy (%, ↑); Synthetic: gradient norm ($\times 10^{-6}$, ↓); LM: validation perplexity (↓). Mean ± std over 3 seeds. Best in **bold**.

Attack	Agg.	MNIST (Acc. %)			Synthetic ($\|\nabla f\| \times 10^{-6}$)			LM (Perplexity)		
		Base	Decay	Ours	Base	Decay	Ours	Base	Decay	Ours
BF	RFA	$78.6_{\pm 4.0}$	$84.1_{\pm 0.2}$	$\mathbf{86.0_{\pm 0.2}}$	$12.8_{\pm 1.2}$	$38.2_{\pm 32.5}$	$\mathbf{7.3_{\pm 0.4}}$	$11.64_{\pm 0.02}$	$12.28_{\pm 0.09}$	$\mathbf{10.22_{\pm 0.13}}$
	Krum	$76.5_{\pm 11.7}$	$81.8_{\pm 0.3}$	$\mathbf{86.0_{\pm 0.1}}$	$14.0_{\pm 2.0}$	$84.5_{\pm 9.8}$	$\mathbf{7.0_{\pm 0.5}}$	$11.68_{\pm 0.04}$	$12.31_{\pm 0.08}$	$\mathbf{10.34_{\pm 0.05}}$
	CM	$78.8_{\pm 4.2}$	$81.8_{\pm 0.3}$	$\mathbf{86.1_{\pm 0.1}}$	$13.0_{\pm 1.0}$	$64.1_{\pm 10.6}$	$\mathbf{7.7_{\pm 0.6}}$	$11.66_{\pm 0.02}$	$12.35_{\pm 0.14}$	$\mathbf{10.34_{\pm 0.06}}$
LF	RFA	$91.9_{\pm 4.6}$	$\mathbf{95.2_{\pm 4.6}}$	$92.2_{\pm 4.5}$	–	–	–	–	–	–
	Krum	$\mathbf{91.9_{\pm 4.6}}$	$88.8_{\pm 0.0}$	$89.1_{\pm 0.0}$	–	–	–	–	–	–
	CM	$\mathbf{92.0_{\pm 4.6}}$	$91.9_{\pm 4.5}$	$89.1_{\pm 0.1}$	–	–	–	–	–	–
Mimic	RFA	$90.9_{\pm 4.0}$	$94.0_{\pm 4.0}$	$\mathbf{95.6_{\pm 0.1}}$	$13.1_{\pm 1.4}$	$193_{\pm 2}$	$\mathbf{6.3_{\pm 0.5}}$	$11.40_{\pm 1.22}$	$12.57_{\pm 0.08}$	$\mathbf{10.09_{\pm 0.08}}$
	Krum	$91.7_{\pm 4.6}$	$\mathbf{92.2_{\pm 4.5}}$	$91.1_{\pm 4.6}$	$12.7_{\pm 0.9}$	$93.7_{\pm 7.0}$	$\mathbf{5.8_{\pm 0.3}}$	$11.06_{\pm 0.10}$	$12.59_{\pm 0.04}$	$\mathbf{10.65_{\pm 0.11}}$
	CM	$\mathbf{92.6_{\pm 3.9}}$	$92.2_{\pm 4.4}$	$92.3_{\pm 4.1}$	$13.1_{\pm 1.4}$	$64.1_{\pm 10.5}$	$\mathbf{5.9_{\pm 0.5}}$	$11.76_{\pm 0.66}$	$12.57_{\pm 0.10}$	$\mathbf{10.08_{\pm 0.03}}$
ALIE	RFA	–	–	–	$12.8_{\pm 1.7}$	$82.6_{\pm 12.6}$	$\mathbf{7.7_{\pm 0.8}}$	$12.20_{\pm 1.44}$	$12.43_{\pm 0.05}$	$\mathbf{10.59_{\pm 0.03}}$
	Krum	–	–	–	$12.7_{\pm 0.9}$	$28.6_{\pm 11.3}$	$\mathbf{7.6_{\pm 0.5}}$	$11.55_{\pm 0.90}$	$12.01_{\pm 0.05}$	$\mathbf{10.73_{\pm 0.11}}$
	CM	–	–	–	$12.7_{\pm 1.7}$	$28.3_{\pm 10.3}$	$\mathbf{7.8_{\pm 0.8}}$	$12.06_{\pm 0.63}$	$12.01_{\pm 0.01}$	$\mathbf{10.66_{\pm 0.07}}$

6.4. Results on Synthetic (L_0, L_1)-Smooth Optimization

Table 1 (middle columns) reports final gradient norms for the quartic optimization task. Lower values indicate better convergence toward $x^* = 0$. Byz-NSGDM consistently achieves the lowest final gradient norms across all configurations, with values around 6–7 $\times 10^{-6}$ compared to 1.3–1.4 $\times 10^{-5}$ for Baseline and 4–9 $\times 10^{-5}$ for Baseline-Decay. This represents a 2–10$\times$ improvement over baselines. Full convergence curves are provided in Appendix D.

These results validate our theoretical contributions: normalized gradient descent with momentum provides superior Byzantine resilience under (L_0, L_1)-smoothness.

6.5. Larger-Scale Experiment: Character-Level Language Modeling

To demonstrate scalability beyond simple classification, we evaluate Byz-NSGDM on character-level language modeling using Shakespeare (≈1M chars, vocab=65). We train a small GPT (4 layers, 128-dim, 4 heads, ≈0.4M params) with $n = 20$ workers and $B = 3$ Byzantine workers. Non-IID partitioning assigns contiguous text chunks to each worker.

Table 1 (right columns) shows final validation perplexity. All methods reduce perplexity from > 60 (random) to ≈10–12, demonstrating that Byzantine-robust training is viable for language modeling. Byz-NSGDM achieves competitive or best perplexity in most configurations. Training curves are in Appendix D.

6.6. Ablation Study: Sensitivity to Hyperparameters

We conduct an ablation study examining how Byz-NSGDM performs across a grid of momentum values $\beta \in \{0.0, 0.3, 0.5, 0.7, 0.9, 0.95, 0.99\}$ and learning rates $\gamma_0 \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$ on the heterogeneous MNIST task with Bit Flipping attack and RFA aggregator. Results are averaged over 3 seeds.

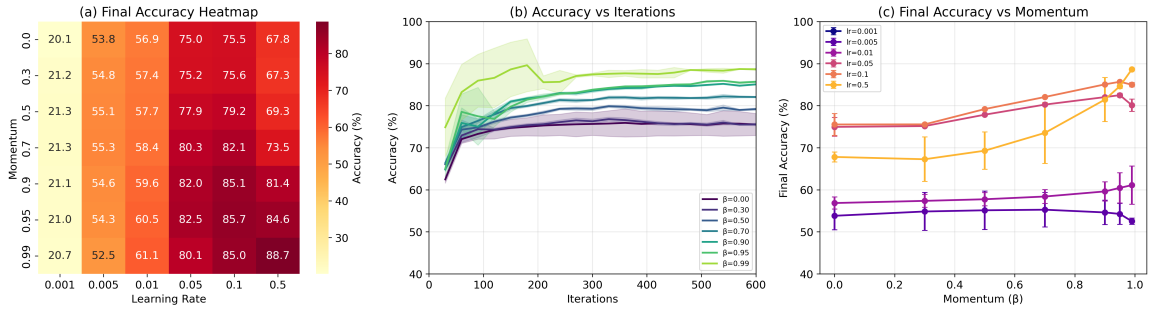

Figure 1: Ablation study on heterogeneous MNIST under Bit Flipping attack with RFA aggregation. (a) Final test accuracy heatmap across momentum (β) and learning rate (γ_0). (b) Training curves for each momentum value using its best learning rate. (c) Final accuracy vs momentum for different learning rates.

Figure 1 shows the results. The key findings are:

- **Momentum:** Any momentum parameter in the range $[0.9, 0.99]$ consistently outperforms momentum values less than 0.9. Based on this observation, we adopt $\beta = 0.9$ as a standard choice throughout our experiments.
- **Learning rate:** The optimal learning rate varies across all momentum values and experimental configurations, requiring careful tuning. We observed that optimal learning rates for Baseline-Decay and Byz-NSGDM are typically larger and in the range $[0.1, 1.0]$, while the Baseline method requires smaller learning rates to achieve stable convergence.

7. Conclusion

In this work, we analyzed the behavior of Byzantine-robust normalized stochastic gradient descent with momentum (Byz-NSGDM) under the generalized (L_0, L_1)-smoothness assumption. While normalized momentum combined with robust aggregation has been studied under classical L-smoothness and homogeneous settings, its theoretical guarantees under state-dependent smoothness and gradient heterogeneity were previously not understood. Our results provide a convergence analysis in this more general regime.

We established a non-asymptotic convergence rate of $O(K^{-1/4})$ up to an explicit Byzantine bias floor that depends on the robustness coefficient of the aggregation rule and the level of gradient heterogeneity. This characterization clarifies how generalized smoothness, heterogeneity, and adversarial corruption jointly affect the attainable accuracy.

Our empirical results on heterogeneous MNIST classification, synthetic (L_0, L_1)-smooth optimization, and character-level language modeling demonstrate that normalization combined with robust aggregation provides stable and competitive performance across diverse Byzantine attack strategies and aggregation rules. The experiments further highlight the practical robustness of the method across a wide range of hyperparameter choices.

Overall, our study bridges Byzantine-robust distributed optimization and optimization under generalized smoothness, and provides theoretical and empirical evidence that normalized momentum remains effective in adversarial and heterogeneous (L_0, L_1)-smooth settings. Future work includes extending the analysis to adaptive step-size schemes, tightening the bias floor characterization, and understanding optimal complexity limits under generalized smoothness and adversarial noise.

Code Availability

Our code and experiments are publicly available at `https://github.com/armanbolatov/byz_nsgdm`.

LLM Usage Statement

We used large language models (LLMs) to assist with code generation and visualization for experiments, to verify theorems and lemmas, to guide the proving process, and to polish the manuscript. The authors are fully responsible for the entire content and methodology of this paper.

References

[1] Yi-Rui Yang, Chang-Wei Shi, and Wu-Jun Li. On the effect of batch size in byzantine-robust distributed learning. In *International Conference on Learning Representations*, 2024.

[2] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. *Advances in neural information processing systems*, 24, 2011.

[3] Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015.

[4] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[5] Chenxin Ma, Jakub Konečnỳ, Martin Jaggi, Virginia Smith, Michael I Jordan, Peter Richtárik, and Martin Takáč. Distributed optimization with arbitrary local solvers. *optimization Methods and Software*, 32(4):813–848, 2017.

[6] Majid Jahani, Xi He, Chenxin Ma, Aryan Mokhtari, Dheevatsa Mudigere, Alejandro Ribeiro, and Martin Takác. Efficient distributed hessian free algorithm for large-scale empirical risk minimization via accumulating sample strategy. In *International Conference on Artificial Intelligence and Statistics*, pages 2634–2644. PMLR, 2020.

[7] Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed learning with compressed gradient differences. *Optimization Methods and Software*, 40(5):1181–1196, 2025.

[8] Majid Jahani, Mohammadreza Nazari, Sergey Rusakov, Albert S Berahas, and Martin Takáč. Scaling up quasi-newton algorithms: Communication efficient distributed sr1. In *International Conference on Machine Learning, Optimization, and Data Science*, pages 41–54. Springer International Publishing Cham, 2020.

[9] Yicheng Chen, Rick S Blum, Martin Takáč, and Brian M Sadler. Distributed learning with sparsified gradient differences. *IEEE Journal of Selected Topics in Signal Processing*, 16(3):585–600, 2022.

[10] Aleksandr Beznosikov, Martin Takác, and Alexander Gasnikov. Similarity, compression and local steps: three pillars of efficient communications for distributed variational inequalities. *Advances in Neural Information Processing Systems*, 36:28663–28677, 2023.

[11] Junyu Shi, Wei Wan, Shengshan Hu, Jianrong Lu, and Leo Yu Zhang. Challenges and approaches for mitigating byzantine attacks in federated learning. In *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications* (*TrustCom*), pages 139–146. IEEE, 2022.

[12] Eduard Gorbunov, Alexander Borzunov, Michael Diskin, and Max Ryabinin. Secure distributed training at scale. In *International Conference on Machine Learning*, pages 7679–7739. PMLR, 2022.

[13] Gleb Molodtsov, Daniil Medyakov, Sergey Skorik, Nikolas Khachaturov, Shahane Tigranyan, Vladimir Aletov, Aram Avetisyan, Martin Takáč, and Aleksandr Beznosikov. Bant: Byzantine antidote via trial function and trust scores. In *The 40th Annual AAAI Conference on Artificial Intelligence*, 2026.

[14] Kai Yi, Nidham Gazagnadou, Peter Richtárik, and Lingjuan Lyu. Fedp3: Federated personalized and privacy-friendly network pruning under model heterogeneity. *arXiv preprint arXiv:2404.09816*, 2024.

[15] Nazarii Tupitsa, Abdulla Jasem Almansoori, Yanlin Wu, Martin Takác, Karthik Nandakumar, Samuel Horváth, and Eduard Gorbunov. Byzantine-tolerant methods for distributed variational inequalities. *Advances in Neural Information Processing Systems*, 36:28393–28461, 2023.

[16] Nurbek Tastan, Samuel Horváth, Martin Takac, and Karthik Nandakumar. Fedpews: Personalized warmup via subnetworks for enhanced heterogeneous federated learning. *arXiv preprint arXiv:2410.03042*, 2024.

[17] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[18] Krishna Pillutla, Sham M. Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022. doi: 10.1109/TSP.2022.3153135.

[19] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659, 2018.

[20] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020. (arXiv preprint arXiv:1905.11881, 2019).

[21] Eduard Gorbunov, Samuel Horváth, Peter Richtárik, and Gauthier Gidel. Variance reduction is an antidote to Byzantines: Better rates, weaker assumptions and communication compression as a cherry on the top. *International Conference on Learning Representations*, 2023.

[22] 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.

[23] Nikita Fedin and Eduard Gorbunov. Byzantine-robust loopless stochastic variance-reduced gradient. In *International Conference on Mathematical Optimization Theory and Operations Research*, pages 39–53. Springer, 2023.

[24] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. *International Conference on Learning Representations*, 2022.

[25] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for Byzantine robust optimization. In *International Conference on Machine Learning*, pages 5311–5319, 2021.

[26] Xiaolan Gu, Ming Li, and Li Xiong. Dp-brem: Differentially-private and byzantine-robust federated learning with client momentum. In *Proceedings of the 34th USENIX Security Symposium (USENIX Security 25)*. USENIX Association, 2025.

[27] Bohang Zhang, Jikai Jin, Cong Fang, and Liwei Wang. Improved analysis of clipping algorithms for non-convex optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 17411–17422, 2020.

[28] Anastasia Koloskova, Hadrien Hendrikx, and Sebastian U Stich. Revisiting gradient clipping: Stochastic bias and tight convergence guarantees. In *International Conference on Machine Learning*, pages 17343–17363. PMLR, 2023.

[29] Renbo Zhao, Rong Jin, and Francesco Orabona. Convergence of a normalized gradient method for nonsmooth nonconvex minimization. In *International Conference on Machine Learning*, 2021.

[30] Ziyi Chen, Yi Zhou, Yingbin Liang, and Zhaosong Lu. Generalized-smooth nonconvex optimization is as efficient as smooth nonconvex optimization. In *International Conference on Machine Learning*, pages 4778–4811. PMLR, 2023.

[31] Florian Hübler, Junchi Yang, Xiang Li, and Niao He. Parameter-agnostic optimization under relaxed smoothness. In *International Conference on Artificial Intelligence and Statistics*, pages 4861–4869. PMLR, 2024.

[32] Sarit Khirirat, Abdurakhmon Sadiev, Artem Riabinin, Eduard Gorbunov, and Peter Richtárik. Error feedback under (l_0, l_1)-smoothness: Normalization and momentum. *arXiv preprint arXiv:2410.16871*, 2024.

[33] Matthew Faw, Litu Rout, Constantine Caramanis, and Sanjay Shakkottai. Beyond uniform smoothness: A stopped analysis of adaptive SGD. In *Conference on Learning Theory*, pages 89–160. PMLR, 2023.

[34] Eduard Gorbunov, Nazarii Tupitsa, Sayantan Choudhury, Alen Aliev, Peter Richtárik, Samuel Horváth, and Martin Takáč. Methods for convex (l_0, l_1)-smooth optimization: Clipping, acceleration, and adaptivity. *International Conference on Learning Representations*, 2025.

[35] Farshed Abdukhakimov, Chulu Xiang, Dmitry Kamzolov, and Martin Takáč. Stochastic gradient descent with preconditioned polyak step-size. *arXiv preprint arXiv:2310.02093*, 2023.

[36] Farshed Abdukhakimov, Cuong Anh Pham, Samuel Horváth, Martin Takáč, and Slavomır Hanzely. Polyak stepsize: Estimating optimal functional values without parameters or prior knowledge. *arXiv preprint arXiv:2508.17288*, 2025.

[37] Nicolas Loizou, Sharan Vaswani, Issam Hadj Laradji, and Simon Lacoste-Julien. Stochastic polyak step-size for sgd: An adaptive learning rate for fast convergence. In *International conference on artificial intelligence and statistics*, pages 1306–1314. PMLR, 2021.

[38] Yunxiang Li, Rui Yuan, Chen Fan, Mark Schmidt, Samuel Horváth, Robert M Gower, and Martin Takáč. Enhancing policy gradient with the polyak step-size adaption. *arXiv preprint arXiv:2404.07525*, 2024.

[39] Farshed Abdukhakimov, Chulu Xiang, Dmitry Kamzolov, Robert Gower, and Martin Takáč. Sania: Polyak-type optimization framework leads to scale invariant stochastic algorithms. *arXiv preprint arXiv:2312.17369*, 2023.

[40] Shuang Li, William J Swartworth, Martin Takáč, Deanna Needell, and Robert M Gower. Sp2: A second order stochastic polyak method. *ICLR 2023*, 2022.

[41] Farshed Abdukhakimov, Chulu Xiang, Dmitry Kamzolov, and Martin Takáč. Stochastic gradient descent with preconditioned polyak step-size. *Computational Mathematics and Mathematical Physics*, 64(4):621–634, 2024.

[42] Fabian Schaipp, Ruben Ohana, Michael Eickenberg, Aaron Defazio, and Robert M Gower. Momo: Momentum models for adaptive learning rates. *arXiv preprint arXiv:2305.07583*, 2023.

[43] Youssef Allouah, Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafael Pinot, and John Stephan. Fixing by mixing: A recipe for optimal byzantine ml under heterogeneity. *arXiv preprint arXiv:2302.01772*, 2023.

[44] Georgios Damaskinos, El-Mahdi El-Mhamdi, Rachid Guerraoui, Arsany Guirguis, and Sébastien Rouault. Aggregathor: Byzantine machine learning via robust gradient aggregation. *Proceedings of Machine Learning and Systems*, 1:81–106, 2019.

[45] Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25, 2017.

[46] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/ec1c59141046cd1866bbbcdfb6ae31d4-Paper.pdf`.

[47] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *Uncertainty in Artificial Intelligence*, pages 261–270, 2020.

[48] Youssef Allouah, Rachid Guerraoui, Nirupam Gupta, Rafaël Pinot, and Geovani Rizk. Robust distributed learning: tight error bounds and breakdown point under data heterogeneity. *Advances in Neural Information Processing Systems*, 36, 2024.

[49] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020.

[50] Michael Crawshaw, Mingrui Liu, Francesco Orabona, Wei Zhang, and Zhenxun Zhuang. Robustness to unbounded smoothness of generalized SignSGD. In *Advances in Neural Information Processing Systems*, volume 35, pages 9955–9968, 2022.

[51] Haochuan Wang, Shuo Ma, Jinlan Lu, and Lijun Zhang. Convergence of Adam under relaxed assumptions. *Advances in Neural Information Processing Systems*, 36, 2023.

[52] Bohan Wang, Huishuai Zhang, Zhiming Ma, and Wei Chen. Convergence of AdaGrad for non-convex objectives: Simple proofs and relaxed assumptions. In *Proceedings of the Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 161–190. PMLR, 2023.

[53] Daniil Vankov, Anton Rodomanov, Angelia Nedich, Lalitha Sankar, and Sebastian U. Stich. Optimizing (l_0, l_1)-smooth functions by gradient methods. *International Conference on Learning Representations*, 2025.

[54] Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained lmos. In *International Conference on Machine Learning*, pages 49069–49104. PMLR, 2025.

[55] Dmitry Kovalev. Understanding gradient orthogonalization for deep learning via non-euclidean trust-region optimization. *arXiv preprint arXiv:2503.12645*, 2025.

[56] Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL `https://kellerjordan.github.io/posts/muon/`.

[57] Ashok Cutkosky and Harsh Mehta. Momentum improves normalized sgd. In *International conference on machine learning*, pages 2260–2268. PMLR, 2020.

[58] Elad Hazan. *Introduction to Online Convex Optimization*, volume 2. Foundations and Trends in Optimization, 2016.

[59] Florian Hübler, Ilyas Fatkhullin, and Niao He. From gradient clipping to normalization for heavy tailed sgd. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan, editors, *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 2413–2421. PMLR, 03–05 May 2025. URL `https://proceedings.mlr.press/v258/hubler25a.html`.

[60] Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning*, pages 5827–5837. PMLR, 2019.

A. Technical Lemmas

Lemma 1 (Lemma 1 from Khirirat et al. [32]). *Let f_i be (L_0, L_1)-smooth for $i \in \mathcal{G}$. Then the following properties hold:*

1. ***Gradient Lipschitz bound:***
$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq (L_0 + L_1\|\nabla f_i(y)\|)\exp(L_1\|x-y\|)\|x-y\|. \tag{3}$$

2. ***Function value bound:***
$$f_i(y) \leq f_i(x) + \langle \nabla f_i(x), y-x\rangle + \frac{L_0 + L_1\|\nabla f_i(x)\|}{2}\exp(L_1\|x-y\|)\|y-x\|^2.$$

3. ***Gradient norm bound:***
$$\frac{\|\nabla f_i(x)\|^2}{4(L_0 + L_1\|\nabla f_i(x)\|)} \leq f_i(x) - f_i^*. \tag{4}$$

4. ***Average function smoothness:***
$$f(y) \leq f(x) + \langle \nabla f(x), y-x\rangle + \frac{L_0 + \frac{L_1}{G}\sum_{i\in\mathcal{G}}\|\nabla f_i(x)\|}{2}\exp(L_1\|x-y\|)\|y-x\|^2. \tag{5}$$

Lemma 2 (Lemma 2 from Khirirat et al. [32]). *Under Assumptions 1 and 2, for any $x \in \mathbb{R}^d$:*
$$\frac{1}{G}\sum_{i\in\mathcal{G}}\|\nabla f_i(x)\| \leq 8L_1(f(x) - f^*) + \frac{8L_1}{G}\sum_{i\in\mathcal{G}}(f^* - f_i^*) + \frac{L_0}{L_1}. \tag{6}$$

Lemma 3. *Let Assumptions 2, 3, 4 hold. For each good worker $i \in \mathcal{G}$ with momentum update*
$$v_i^k = (1-\eta)v_i^{k-1} + \eta\nabla f_{\xi_i^{k-1}}(x^{k-1}),$$
consider normalized steps $\|x^k - x^{k-1}\| = \gamma$ (i.e., $\gamma_k \equiv \gamma > 0$ in Algorithm 1) and a constant momentum parameter $\eta \in (0, 1]$. Then:

1. ***Bias.*** *For all $k \geq 1$,*
$$\mathbb{E}\left[v_i^k \middle| x^{k-1}\right] = (1-\eta)v_i^{k-1} + \eta\nabla f_i(x^{k-1}).$$

2. ***Individual momentum tracking error.*** *For all $k \geq 1$,*
$$\begin{aligned}\mathbb{E}\big[\|v_i^k - \nabla f_i(x^{k-1})\|\big] \leq{}& (1-\eta)^k\mathbb{E}\|v_i^0 - \nabla f_i(x^0)\| + \frac{2L_0\gamma e^{\gamma L_1}}{\eta}\\ &+ 8L_1^2\gamma e^{\gamma L_1}\sum_{t=0}^{k-1}(1-\eta)^{k-1-t}\mathbb{E}\big[f(x^t) - f^*\big]\\ &+ \frac{8L_1^2\gamma e^{\gamma L_1}}{\eta}\big(f^* - f_i^*\big) + \sqrt{\eta}\sigma.\end{aligned} \tag{7}$$

3. ***Cross-worker momentum difference.*** *Let the expectation be over the stochastic gradients and an independent uniform draw of $i, j \in \mathcal{G}$. Then, for all $k \geq 1$,*
$$\begin{aligned}\mathbb{E}\big[\|v_i^k - v_j^k\|\big] \leq{}& 2\zeta + 2(1-\eta)^k V_0 + 2\sqrt{\eta}\sigma + \frac{4L_0\gamma e^{\gamma L_1}}{\eta}\\ &+ 16L_1^2\gamma e^{\gamma L_1}\sum_{t=0}^{k-1}(1-\eta)^{k-1-t}\mathbb{E}\big[f(x^t) - f^*\big]\\ &+ \frac{16L_1^2\gamma e^{\gamma L_1}}{\eta}\Delta^*,\end{aligned} \tag{8}$$
where $V_0 := \frac{1}{G}\sum_{i\in\mathcal{G}}\mathbb{E}\|v_i^0 - \nabla f_i(x^0)\|$ and $\Delta^* := \frac{1}{G}\sum_{i\in\mathcal{G}}(f^* - f_i^*)$.

Proof. The proof of these lemmas follows a very similar structure to that of the lemmas in [32].

Part 1. This follows directly from the unbiased oracle in Assumption 3: $\mathbb{E}[\nabla f_{\xi_i^{k-1}}(x^{k-1}) \mid x^{k-1}] = \nabla f_i(x^{k-1})$.

Part 2. Define the tracking error $H_i^k := v_i^k - \nabla f_i(x^{k-1})$ and the auxiliary terms

$$U_i^{k-1} := \nabla f_{\xi_i^{k-1}}(x^{k-1}) - \nabla f_i(x^{k-1}), \qquad G_i^k := \nabla f_i(x^{k-1}) - \nabla f_i(x^k),$$

where we also set $x^{-1} := 0$, yielding $G_i^0 = 0$. From the momentum update $v_i^k = (1-\eta)v_i^{k-1} + \eta \nabla f_{\xi_i^{k-1}}(x^{k-1})$, we have

$$H_i^k = (1-\eta)H_i^{k-1} + (1-\eta)G_i^{k-1} + \eta U_i^{k-1}. \tag{9}$$

Unrolling gives

$$H_i^k = (1-\eta)^{k+1}H_i^0 + \sum_{t=0}^{k}(1-\eta)^{k+1-t}G_i^t + \eta\sum_{t=0}^{k}(1-\eta)^{k-t}U_i^t.$$

Taking norms, triangle inequality, and expectations,

$$\mathbb{E}\|H_i^k\| \le (1-\eta)^{k+1}\mathbb{E}\|H_i^0\| + \sum_{t=0}^{k}(1-\eta)^{k+1-t}\mathbb{E}\|G_i^t\| + \eta\mathbb{E}\Big\|\sum_{t=0}^{k}(1-\eta)^{k-t}U_i^t\Big\|. \tag{10}$$

By Lemma 1 (see (3)) with $\|x^t - x^{t-1}\| = \gamma$,

$$\|\nabla f_i(x^{t-1}) - \nabla f_i(x^t)\| \le \big(L_0 + L_1\|\nabla f_i(x^{t-1})\|\big)e^{\gamma L_1}\gamma.$$

From Lemma 1 (see (4)) and the same algebra as Lemma 2 in Khirirat et al. [32] applied to each f_i, $\|\nabla f_i(x)\| \le 8L_1\big(f_i(x) - f_i^*\big) + \frac{L_0}{L_1} \le 8L_1\big(f(x) - f^*\big) + 8L_1\big(f^* - f_i^*\big) + \frac{L_0}{L_1}$. We obtain

$$\mathbb{E}\|G_i^t\| \le e^{\gamma L_1}\gamma\left(2L_0 + 8L_1^2\mathbb{E}[f(x^t) - f^*] + 8L_1^2(f^* - f_i^*)\right).$$

Summing the geometric weights $\sum_{t=0}^{k}(1-\eta)^{k+1-t} \le \eta^{-1}$ yields the three deterministic terms in (7).

By Jensen and Cauchy–Schwarz,

$$\mathbb{E}\Big\|\sum_{t=0}^{k}(1-\eta)^{k-t}U_i^t\Big\| \le \Big(\sum_{t=0}^{k}(1-\eta)^{2(k-t)}\mathbb{E}\|U_i^t\|^2\Big)^{1/2}.$$

The cross terms vanish since the oracle is unbiased and adapted to the filtration $\{\mathcal{F}_t\}$: for $s < t$,

$$\mathbb{E}\big[\langle U_i^s, U_i^t\rangle\big] = \mathbb{E}\big[\langle U_i^s, \mathbb{E}[U_i^t \mid \mathcal{F}_t]\rangle\big] = 0.$$

Using $\mathbb{E}\|U_i^t\|^2 \le \sigma^2$ and summing the geometric series gives $\le \sigma/\sqrt{2\eta - \eta^2} \le \sigma/\sqrt{\eta}$, so the prefactor η in (10) produces the term $\sqrt{\eta}\sigma$. This finishes Part 2.

Part 3. By the triangle inequality, $\|v_i^k - v_j^k\| \le \|H_i^k\| + \|\nabla f_i(x^k) - \nabla f_j(x^k)\| + \|H_j^k\|$. Taking expectations, using (1) and Jensen gives $\mathbb{E}\|\nabla f_i(x^k) - \nabla f_j(x^k)\| \le 2\zeta$; applying the bound of Part 2 to H_i^k and H_j^k and averaging over i, j replaces $(f^* - f_i^*)$ by Δ^* and $\mathbb{E}\|H_i^0\|$ by V_0, yielding (8). □

Lemma 4 (Average momentum error)**.** *Let $\bar{v}^k := \frac{1}{G}\sum_{i\in\mathcal{G}} v_i^k$ and assume the setting of Lemma 3 with constant η and normalized steps of size γ. Then for $k \geq 1$,*

$$\begin{aligned} A_k := \mathbb{E}\big[\|\bar{v}^k - \nabla f(x^{k-1})\|\big] \leq (1-\eta)^k V_0 + \frac{\sqrt{\eta}\sigma}{\sqrt{G}} + \frac{2L_0\gamma e^{\gamma L_1}}{\eta} \\ + 8L_1^2\gamma e^{\gamma L_1}\sum_{t=0}^{k-1}(1-\eta)^{k-1-t}\mathbb{E}[f(x^t) - f^*] \\ + \frac{8L_1^2\gamma e^{\gamma L_1}}{\eta}\Delta^*. \end{aligned} \tag{11}$$

Here $V_0 := \frac{1}{G}\sum_{i\in\mathcal{G}}\mathbb{E}\|v_i^0 - \nabla f_i(x^0)\|$ and $\Delta^* := \frac{1}{G}\sum_{i\in\mathcal{G}}(f^* - f_i^*)$.

Proof. Let $H_i^k := v_i^k - \nabla f_i(x^{k-1})$ and define the averages

$$\bar{H}^k := \frac{1}{G}\sum_{i\in\mathcal{G}} H_i^k,\ \bar{G}^t := \frac{1}{G}\sum_{i\in\mathcal{G}}(\nabla f_i(x^{t-1}) - \nabla f_i(x^t)),\ \bar{U}^t := \frac{1}{G}\sum_{i\in\mathcal{G}}\big(\nabla f_{\xi_i^t}(x^t) - \nabla f_i(x^t)\big).$$

Averaging the per-worker recursion (9) gives

$$\bar{H}^k = (1-\eta)\bar{H}^{k-1} + (1-\eta)\bar{G}^{k-1} + \eta\bar{U}^{k-1}.$$

Unrolling, taking norms, and expectations as in Lemma 3 (Part 2),

$$\mathbb{E}\|\bar{H}^k\| \leq (1-\eta)^k\mathbb{E}\|\bar{H}^0\| + \sum_{t=0}^{k-1}(1-\eta)^{k-t}\mathbb{E}\|\bar{G}^t\| + \eta\mathbb{E}\Big\|\sum_{t=0}^{k-1}(1-\eta)^{k-1-t}\bar{U}^t\Big\|.$$

For $\mathbb{E}\|\bar{G}^t\|$, apply Lemma 1 (see (3)) and the same one-point bound used in Lemma 3 to get

$$\mathbb{E}\|\bar{G}^t\| \leq e^{\gamma L_1}\gamma\left(2L_0 + 8L_1^2\mathbb{E}[f(x^t) - f^*] + 8L_1^2\Delta^*\right).$$

For the noise term, independence across workers and unbiasedness give

$$\mathbb{E}\|\bar{U}^t\|^2 = \mathbb{E}\Big\|\frac{1}{G}\sum_i U_i^t\Big\|^2 = \frac{1}{G^2}\sum_i \mathbb{E}\|U_i^t\|^2 \leq \frac{\sigma^2}{G}.$$

As in Part 2, cross terms in time vanish by the short argument

$$\mathbb{E}\big[\langle\bar{U}^s, \bar{U}^t\rangle\big] = \mathbb{E}\big[\langle\bar{U}^s, \mathbb{E}[\bar{U}^t \mid \mathcal{F}_t]\rangle\big] = 0 \quad (s < t),$$

so

$$\mathbb{E}\Big\|\sum_{t=0}^{k}(1-\eta)^{k-t}\bar{U}^t\Big\| \leq \frac{\sigma}{\sqrt{G}}\Big(\sum_{t=0}^{\infty}(1-\eta)^{2t}\Big)^{1/2} \leq \frac{\sigma}{\sqrt{G\eta}}.$$

Putting these together and using $\sum_{t=0}^{k}(1-\eta)^{k+1-t} \leq \eta^{-1}$ yields

$$\mathbb{E}\|\bar{H}^k\| \leq (1-\eta)^k V_0 + \frac{\sqrt{\eta}\sigma}{\sqrt{G}} + \frac{2L_0\gamma e^{\gamma L_1}}{\eta} + 8L_1^2\gamma e^{\gamma L_1}\sum_{t=0}^{k-1}(1-\eta)^{k-1-t}\mathbb{E}[f(x^t) - f^*] + \frac{8L_1^2\gamma e^{\gamma L_1}}{\eta}\Delta^*.$$

Finally, note that $\bar{H}^k = \bar{v}^k - \nabla f(x^{k-1})$, which gives (11). □

Lemma 5 (Descent Lemma for Byz-NSGDM)**.** *Let $f(x) = \frac{1}{G}\sum_{i\in\mathcal{G}} f_i(x)$ with each f_i (L_0, L_1)-smooth, and let the server update*

$$x^k = x^{k-1} - \gamma_{k-1}\frac{v^k}{\|v^k\|} \qquad (\gamma_{k-1} > 0).$$

Then

$$\begin{aligned} f(x^k) \le f(x^{k-1}) &- \gamma_{k-1}\|\nabla f(x^{k-1})\| + 2\gamma_{k-1}\|\nabla f(x^{k-1}) - v^k\| \\ &+ \frac{\gamma_{k-1}^2}{2}\exp(\gamma_{k-1}L_1)\left(L_0 + \frac{L_1}{G}\sum_{i\in\mathcal{G}}\|\nabla f_i(x^{k-1})\|\right). \end{aligned}$$

Proof. Apply Lemma 1 (average smoothness, (5)) with $x = x^{k-1}$ and $y = x^k = x^{k-1} - \gamma_{k-1}\frac{v^k}{\|v^k\|}$:

$$f(x^k) \le f(x^{k-1}) + \left\langle \nabla f(x^{k-1}), -\gamma_{k-1}\tfrac{v^k}{\|v^k\|}\right\rangle + \frac{\gamma_{k-1}^2}{2}\exp(\gamma_{k-1}L_1)\left(L_0 + \frac{L_1}{G}\sum_{i\in\mathcal{G}}\|\nabla f_i(x^{k-1})\|\right). \tag{12}$$

It remains to lower bound the linear term $\left\langle \nabla f(x^{k-1}), \frac{v^k}{\|v^k\|}\right\rangle$. Write

$$\begin{aligned} \left\langle \nabla f(x^{k-1}), \tfrac{v^k}{\|v^k\|}\right\rangle &= \left\langle \nabla f(x^{k-1}) - v^k, \tfrac{v^k}{\|v^k\|}\right\rangle + \left\langle v^k, \tfrac{v^k}{\|v^k\|}\right\rangle \\ &\ge -\|\nabla f(x^{k-1}) - v^k\| + \|v^k\| \qquad \text{(Cauchy–Schwarz)} \\ &\ge -\|\nabla f(x^{k-1}) - v^k\| + \|\nabla f(x^{k-1})\| - \|\nabla f(x^{k-1}) - v^k\| \quad \text{(triangle ineq.)} \\ &= \|\nabla f(x^{k-1})\| - 2\|\nabla f(x^{k-1}) - v^k\|. \end{aligned}$$

Therefore

$$-\gamma_{k-1}\left\langle \nabla f(x^{k-1}), \tfrac{v^k}{\|v^k\|}\right\rangle \le -\gamma_{k-1}\|\nabla f(x^{k-1})\| + 2\gamma_{k-1}\|\nabla f(x^{k-1}) - v^k\|.$$

Substituting this bound into (12) gives the claim. □

Lemma 6 (Descent with curvature bound)**.** *Let $\{x^k\}$ be generated by Algorithm 1 and Assumption 2 hold. Then for $k \ge 1$,*

$$\begin{aligned} f(x^k) \le f(x^{k-1}) &- \gamma_{k-1}\|\nabla f(x^{k-1})\| + 2\gamma_{k-1}\|\nabla f(x^{k-1}) - v^k\| \\ &+ L_0\gamma_{k-1}^2 e^{\gamma_{k-1}L_1} + 4L_1^2\gamma_{k-1}^2 e^{\gamma_{k-1}L_1}\big(f(x^{k-1}) - f^*\big) + 4L_1^2\gamma_{k-1}^2 e^{\gamma_{k-1}L_1}\Delta^*. \end{aligned}$$

Proof. Applying Lemma 5, we have

$$\begin{aligned} f(x^k) \le f(x^{k-1}) &- \gamma_{k-1}\|\nabla f(x^{k-1})\| + 2\gamma_{k-1}\|\nabla f(x^{k-1}) - v^k\| \\ &+ \frac{\gamma_{k-1}^2}{2}e^{\gamma_{k-1}L_1}\left(L_0 + \frac{L_1}{G}\sum_{i\in\mathcal{G}}\|\nabla f_i(x^{k-1})\|\right). \end{aligned}$$

Using the sum-of-gradient-norms bound (6) from Lemma 2, and substituting this into the curvature term yields

$$\begin{aligned} &\frac{\gamma_{k-1}^2}{2}e^{\gamma_{k-1}L_1}\left(L_0 + \frac{L_1}{G}\sum_{i\in\mathcal{G}}\|\nabla f_i(x^{k-1})\|\right) \\ &\quad \le L_0\gamma_{k-1}^2 e^{\gamma_{k-1}L_1} + 4L_1^2\gamma_{k-1}^2 e^{\gamma_{k-1}L_1}\big(f(x^{k-1}) - f^*\big) \\ &\qquad + 4L_1^2\gamma_{k-1}^2 e^{\gamma_{k-1}L_1}\Delta^*. \end{aligned}$$

Combining gives the claim. □

Lemma 7 (Byzantine Aggregation Error Decomposition)**.** *Let $\bar{v}^k = \frac{1}{G}\sum_{i\in\mathcal{G}} v_i^k$ be the average momentum of good workers and $v^k = \texttt{ARAgg}(v_1^k, \ldots, v_n^k)$ be the output of a (δ, κ)-robust aggregator. Then for $k \geq 1$:*

$$\mathbb{E}[\|v^k - \nabla f(x^{k-1})\|] \leq \mathbb{E}[\|\bar{v}^k - \nabla f(x^{k-1})\|] + \kappa \mathbb{E}_{i,j\in\mathcal{G}}[\|v_i^k - v_j^k\|],$$

where $\mathbb{E}_{i,j\in\mathcal{G}}[\|v_i^k - v_j^k\|] = \frac{1}{G^2}\sum_{i,j\in\mathcal{G}} \mathbb{E}[\|v_i^k - v_j^k\|]$.

Proof. Using the triangle inequality:

$$\|v^k - \nabla f(x^{k-1})\| \leq \|v^k - \bar{v}^k\| + \|\bar{v}^k - \nabla f(x^{k-1})\|.$$

Applying the (δ, κ)-robust aggregator property:

$$\|v^k - \bar{v}^k\| \leq \frac{\kappa}{G}\sum_{i\in\mathcal{G}} \|v_i^k - \bar{v}^k\|.$$

For each $i \in \mathcal{G}$:

$$\|v_i^k - \bar{v}^k\| = \left\| v_i^k - \frac{1}{G}\sum_{j\in\mathcal{G}} v_j^k \right\| = \left\| \frac{1}{G}\sum_{j\in\mathcal{G}} (v_i^k - v_j^k) \right\| \leq \frac{1}{G}\sum_{j\in\mathcal{G}} \|v_i^k - v_j^k\|.$$

Therefore:

$$\|v^k - \bar{v}^k\| \leq \frac{\kappa}{G}\sum_{i\in\mathcal{G}} \frac{1}{G}\sum_{j\in\mathcal{G}} \|v_i^k - v_j^k\| = \frac{\kappa}{G^2}\sum_{i,j\in\mathcal{G}} \|v_i^k - v_j^k\|.$$

Taking expectation and combining with the triangle inequality:

$$\begin{aligned}
\mathbb{E}[\|v^k - \nabla f(x^{k-1})\|] &\leq \mathbb{E}[\|v^k - \bar{v}^k\|] + \mathbb{E}[\|\bar{v}^k - \nabla f(x^{k-1})\|] \\
&\leq \frac{\kappa}{G^2}\sum_{i,j\in\mathcal{G}} \mathbb{E}[\|v_i^k - v_j^k\|] + \mathbb{E}[\|\bar{v}^k - \nabla f(x^{k-1})\|] \\
&= \kappa \mathbb{E}_{i,j\in\mathcal{G}}[\|v_i^k - v_j^k\|] + \mathbb{E}[\|\bar{v}^k - \nabla f(x^{k-1})\|].
\end{aligned}$$

□

B. Proof of Theorem 1

Proof. We begin by establishing the descent inequality and then perform a weighted telescoping analysis. Define the following quantities for notational convenience:

$$\begin{aligned}
\Delta_k &= \mathbb{E}[f(x^k) - f^*], \quad A_k = \mathbb{E}[\|\bar{v}^k - \nabla f(x^{k-1})\|], \\
B_k &= \mathbb{E}[\|v^k - \nabla f(x^{k-1})\|], \quad D_k = \mathbb{E}_{i,j\in\mathcal{G}}[\|v_i^k - v_j^k\|].
\end{aligned}$$

Since $\gamma \leq \frac{1}{2L_1}$ from our choice in (2), we have $\exp(L_1\gamma) \leq \exp(1/2) \leq 2$. Applying Lemma 6, we obtain the one-step descent inequality:

$$\Delta_k \leq (1 + 8L_1^2\gamma^2)\Delta_{k-1} - \gamma\mathbb{E}[\|\nabla f(x^{k-1})\|] + 2\gamma B_k + 2L_0\gamma^2 + 8L_1^2\gamma^2\Delta^*. \tag{13}$$

We now bound B_k using Lemma 7: $B_k \leq A_k + \kappa D_k$. Therefore:

$$2\gamma B_k \leq 2\gamma A_k + 2\gamma\kappa D_k.$$

Next, we bound A_k and D_k using Lemma 4 and equation (8). Since $\exp(L_1\gamma) \leq 2$, we have:

$$A_k \leq (1-\eta)^k V_0 + \frac{\sqrt{\eta}\sigma}{\sqrt{G}} + \frac{4L_0\gamma}{\eta} + 16L_1^2\gamma\sum_{t=0}^{k-1}(1-\eta)^{k-1-t}\Delta_t + \frac{16L_1^2\gamma}{\eta}\Delta^*,$$

$$D_k \leq 2\zeta + 2(1-\eta)^k V_0 + 2\sqrt{\eta}\sigma + \frac{8L_0\gamma}{\eta} + 32L_1^2\gamma\sum_{t=0}^{k-1}(1-\eta)^{k-1-t}\Delta_t + \frac{32L_1^2\gamma}{\eta}\Delta^*.$$

Substituting these bounds into (13) and collecting terms yields:

$$\begin{aligned}\Delta_k &\le (1+8L_1^2\gamma^2)\Delta_{k-1} - \gamma\mathbb{E}[\|\nabla f(x^{k-1})\|] + 2\gamma A_k + 2\gamma\kappa D_k + 2L_0\gamma^2 + 8L_1^2\gamma^2\Delta^* \\ &\le (1+8L_1^2\gamma^2)\Delta_{k-1} - \gamma\mathbb{E}[\|\nabla f(x^{k-1})\|] + 32L_1^2\gamma^2(1+2\kappa)\sum_{t=0}^{k-1}(1-\eta)^{k-1-t}\Delta_t \\ &\quad + 2\gamma(1+2\kappa)(1-\eta)^k V_0 + 4\gamma\kappa\zeta + \frac{2\gamma\sqrt{\eta}\sigma}{\sqrt{G}} + 4\gamma\kappa\sqrt{\eta}\sigma \\ &\quad + 2L_0\gamma^2 + \frac{8(1+2\kappa)L_0\gamma^2}{\eta} + 8L_1^2\gamma^2\Delta^* + \frac{32(1+2\kappa)L_1^2\gamma^2}{\eta}\Delta^*. \end{aligned} \tag{14}$$

To analyze the convergence over K iterations, we introduce a weighted telescoping argument. Define the weights:

$$w_k = \left(1 + 8L_1^2\gamma^2 + \frac{32(1+2\kappa)L_1^2\gamma^2}{\eta}\right)^{-k}, \quad S_K = \sum_{k=1}^{K} w_k.$$

Note that w_k decreases geometrically with k. The key observation is that:

$$w_{k-1} = w_k \cdot \left(1 + 8L_1^2\gamma^2 + \frac{32(1+2\kappa)L_1^2\gamma^2}{\eta}\right).$$

Now we multiply (14) by w_k and sum from $k = 1$ to K. Let's handle each term carefully. For the first term on the right-hand side:

$$\sum_{k=1}^{K} w_k(1+8L_1^2\gamma^2)\Delta_{k-1} = \sum_{k=1}^{K} w_k\Delta_{k-1} + 8L_1^2\gamma^2\sum_{k=1}^{K} w_k\Delta_{k-1}.$$

For the convolution term, we need to bound $\sum_{k=1}^{K} w_k \sum_{t=0}^{k-1}(1-\eta)^{k-1-t}\Delta_t$. We can rewrite this double sum by changing the order of summation. For a fixed t, the inner sum runs over all k such that $t \le k-1$, which means $k \ge t+1$. Therefore:

$$\sum_{k=1}^{K} w_k \sum_{t=0}^{k-1}(1-\eta)^{k-1-t}\Delta_t = \sum_{t=0}^{K-1}\Delta_t \sum_{k=t+1}^{K} w_k(1-\eta)^{k-1-t}.$$

Since w_k is decreasing in k (i.e., $w_{k+1} \le w_k$), we can bound:

$$\begin{aligned}\sum_{k=t+1}^{K} w_k(1-\eta)^{k-1-t} &\le w_{t+1}\sum_{k=t+1}^{K}(1-\eta)^{k-1-t} = w_{t+1}\sum_{j=0}^{K-t-1}(1-\eta)^j \\ &\le w_{t+1}\sum_{j=0}^{\infty}(1-\eta)^j = \frac{w_{t+1}}{\eta}.\end{aligned}$$

Therefore:

$$\sum_{k=1}^{K} w_k \sum_{t=0}^{k-1}(1-\eta)^{k-1-t}\Delta_t \le \frac{1}{\eta}\sum_{t=0}^{K-1} w_{t+1}\Delta_t \le \frac{1}{\eta}\sum_{k=1}^{K} w_k\Delta_{k-1}.$$

Now, summing (14) multiplied by w_k from $k = 1$ to K:

$$\begin{aligned}\sum_{k=1}^{K} w_k\Delta_k &\le \sum_{k=1}^{K} w_k(1+8L_1^2\gamma^2)\Delta_{k-1} + \frac{32L_1^2\gamma^2(1+2\kappa)}{\eta}\sum_{k=1}^{K} w_k\Delta_{k-1} \\ &\quad - \gamma\sum_{k=1}^{K} w_k\mathbb{E}[\|\nabla f(x^{k-1})\|] + (\text{other terms}).\end{aligned}$$

Combining the terms with Δ_{k-1}:

$$\sum_{k=1}^{K} w_k \Delta_k \leq \sum_{k=1}^{K} w_k \left(1 + 8L_1^2\gamma^2 + \frac{32(1+2\kappa)L_1^2\gamma^2}{\eta}\right) \Delta_{k-1} - \gamma \sum_{k=1}^{K} w_k \mathbb{E}[\|\nabla f(x^{k-1})\|] + (\text{other terms}).$$

Using the relation $w_{k-1} = w_k(1 + 8L_1^2\gamma^2 + \frac{32(1+2\kappa)L_1^2\gamma^2}{\eta})$, we have:

$$\sum_{k=1}^{K} w_k \Delta_k \leq \sum_{k=1}^{K} w_{k-1} \Delta_{k-1} - \gamma \sum_{k=1}^{K} w_k \mathbb{E}[\|\nabla f(x^{k-1})\|] + (\text{other terms}),$$

implying

$$\gamma \sum_{k=0}^{K} w_k \mathbb{E}[\|\nabla f(x^k)\|] \leq \Delta_0 + (\text{other terms}),$$

since $w_0 = 1$ and both $w_K \geq 0$ and $\Delta_{K+1} \geq 0$ (by Assumption 1).

Let's now collect all the "other terms" from the weighted sum of (14). For the terms involving $(1-\eta)^k$:

$$\sum_{k=0}^{K} w_k \cdot 2\gamma(1+2\kappa)(1-\eta)^k V_0 \leq 2\gamma(1+2\kappa)V_0 \sum_{k=0}^{K} w_k (1-\eta)^k \leq 2\gamma(1+2\kappa)V_0 \cdot w_0 \sum_{k=0}^{\infty} (1-\eta)^k = \frac{2\gamma(1+2\kappa)V_0}{\eta}.$$

The remaining terms are constants multiplied by $\sum_{k=0}^{K} w_k = S_K$. Collecting everything:

$$\gamma \sum_{k=0}^{K} w_k \mathbb{E}[\|\nabla f(x^k)\|] \leq \Delta_0 + \frac{2\gamma(1+2\kappa)V_0}{\eta} + 4\gamma\kappa\zeta S_K + \frac{2\gamma\sqrt{\eta}\sigma}{\sqrt{G}} S_K + 4\gamma\kappa\sqrt{\eta}\sigma S_K + \left(2L_0\gamma^2 + \frac{8(1+2\kappa)L_0\gamma^2}{\eta}\right) S_K + \left(8L_1^2\gamma^2 + \frac{32(1+2\kappa)L_1^2\gamma^2}{\eta}\right) \Delta^* S_K.$$

Since $\sum_{k=0}^{K} w_k \mathbb{E}[\|\nabla f(x^k)\|] \geq S_K \cdot \min_{0\leq k\leq K} \mathbb{E}[\|\nabla f(x^k)\|]$, we obtain:

$$\gamma S_K \min_{0\leq k\leq K} \mathbb{E}[\|\nabla f(x^k)\|] \leq \Delta_0 + \frac{2\gamma(1+2\kappa)V_0}{\eta} + 4\gamma\kappa\zeta S_K + S_K \left(2L_0\gamma^2 + \frac{8(1+2\kappa)L_0\gamma^2}{\eta}\right) + S_K \left(8L_1^2\gamma^2 + \frac{32(1+2\kappa)L_1^2\gamma^2}{\eta}\right) \Delta^* + \frac{2\gamma\sqrt{\eta}\sigma S_K}{\sqrt{G}} + 4\gamma\kappa\sqrt{\eta}\sigma S_K. \tag{15}$$

To complete the proof, we need a lower bound on S_K. Using the inequality $(1+x)^n \leq e^{nx}$ for $x \geq 0$:

$$\begin{aligned} S_K = \sum_{k=0}^{K} w_k &= \sum_{k=0}^{K} \left(1 + 8L_1^2\gamma^2 + \frac{32(1+2\kappa)L_1^2\gamma^2}{\eta}\right)^{-(k+1)} \\ &\geq \frac{K+1}{\left(1 + 8L_1^2\gamma^2 + \frac{32(1+2\kappa)L_1^2\gamma^2}{\eta}\right)^{K+1}} \\ &\geq \frac{K+1}{\exp\left(\left(8L_1^2\gamma^2 + \frac{32(1+2\kappa)L_1^2\gamma^2}{\eta}\right)(K+1)\right)}. \end{aligned}$$

Substituting $\gamma = \gamma_0/(K+1)^{3/4}$ and $\eta = (K+1)^{-1/2}$:

$$\begin{aligned} 8L_1^2\gamma^2 + \frac{32(1+2\kappa)L_1^2\gamma^2}{\eta} &= \frac{8L_1^2\gamma_0^2}{(K+1)^{3/2}} + \frac{32(1+2\kappa)L_1^2\gamma_0^2}{(K+1)^{3/2}} \cdot (K+1)^{1/2} \\ &= \frac{8L_1^2\gamma_0^2}{(K+1)^{3/2}} + \frac{32(1+2\kappa)L_1^2\gamma_0^2}{K+1}. \end{aligned}$$

Therefore:

$$S_K \geq \frac{K+1}{\exp\left(\frac{8L_1^2\gamma_0^2}{(K+1)^{1/2}} + 32(1+2\kappa)L_1^2\gamma_0^2\right)}.$$

The choice (2) ensures that $32(1+2\kappa)L_1^2\gamma_0^2 \leq 1/4$ and $\frac{8L_1^2\gamma_0^2}{(K+1)^{1/2}} \leq 1/4$ for all $K \geq 0$. Hence:

$$S_K \geq (K+1)e^{-1/2}.$$

Finally, dividing (15) by γS_K and substituting the parameter choices:

$$\begin{aligned} \min_{0\leq k\leq K} \mathbb{E}[\|\nabla f(x^k)\|] \leq \frac{\Delta_0}{\gamma S_K} &+ \frac{2(1+2\kappa)V_0}{\eta S_K} + 4\kappa\zeta \\ &+ 2L_0\gamma + \frac{8(1+2\kappa)L_0\gamma}{\eta} + 8L_1^2\gamma\Delta^* + \frac{32(1+2\kappa)L_1^2\gamma\Delta^*}{\eta} \\ &+ \frac{2\sqrt{\eta}\sigma}{\sqrt{G}} + 4\kappa\sqrt{\eta}\sigma. \end{aligned}$$

Substituting $\gamma = \gamma_0/(K+1)^{3/4}$, $\eta = (K+1)^{-1/2}$, $S_K \geq (K+1)e^{-1/2}$, and $\Delta_0 = f(x^0) - f^*$:

$$\begin{aligned} \min_{0\leq k\leq K} \mathbb{E}[\|\nabla f(x^k)\|] \leq \frac{e^{1/2}(f(x^0)-f^*)(K+1)^{3/4}}{\gamma_0(K+1)} &+ \frac{2e^{1/2}(1+2\kappa)V_0(K+1)^{1/2}}{K+1} + 4\kappa\zeta \\ &+ \frac{2L_0\gamma_0}{(K+1)^{3/4}} + \frac{8(1+2\kappa)L_0\gamma_0(K+1)^{1/2}}{(K+1)^{3/4}} \\ &+ \frac{8L_1^2\Delta^*\gamma_0}{(K+1)^{3/4}} + \frac{32(1+2\kappa)L_1^2\Delta^*\gamma_0(K+1)^{1/2}}{(K+1)^{3/4}} \\ &+ \frac{2\sigma}{\sqrt{G}(K+1)^{1/4}} + \frac{4\kappa\sigma}{(K+1)^{1/4}}. \end{aligned}$$

Simplifying the powers of $(K+1)$ yields:

$$\begin{aligned}\min_{0\le k\le K}\mathbb{E}[\|\nabla f(x^k)\|] \le\ & \frac{e^{1/2}(f(x^0)-f^*)}{\gamma_0(K+1)^{1/4}} + \frac{2e^{1/2}(1+2\kappa)V_0}{(K+1)^{1/2}} + 4\kappa\zeta \\ & + \frac{2L_0\gamma_0}{(K+1)^{3/4}} + \frac{8(1+2\kappa)L_0\gamma_0}{(K+1)^{1/4}} \\ & + \frac{8L_1^2\Delta^*\gamma_0}{(K+1)^{3/4}} + \frac{32(1+2\kappa)L_1^2\Delta^*\gamma_0}{(K+1)^{1/4}} \\ & + \frac{2\sigma}{\sqrt{G}(K+1)^{1/4}} + \frac{4\kappa\sigma}{(K+1)^{1/4}}.\end{aligned}$$

□

C. Byzantine-Robust Aggregators under New Robustness Definition

This section provides proofs that several aggregation rules from [43] satisfy our modified (δ,κ)-robustness definition (Definition 1), which removes the squares compared to the original definition in [43]. We prove robustness for Geometric Median (GM) and Coordinate-wise Median (CWMed). The proofs follow a similar structure to those in [43] but are adapted for the linear (non-squared) robustness condition.

C.1. Nearest Neighbor Mixing (NNM) and Robustness

NNM operator. Given $x_1,\dots,x_n\in\mathbb{R}^d$ and an integer $B<n/2$, let $\mathcal{G}$ be the set of good indices with $|\mathcal{G}|=G=n-B$. For a pivot $\mu\in\mathbb{R}^d$, denote by $\mathcal{N}_\mu$ the indices of its G nearest neighbors among $\{1,\dots,n\}$ and set

$$y_\mu := \frac{1}{G}\sum_{j\in\mathcal{N}_\mu} x_j.$$

For each $i\in[n]$, define $y_i := y_{x_i}$. For the good set $\mathcal{G}$ let

$$\bar{x}_{\mathcal{G}} := \frac{1}{G}\sum_{j\in\mathcal{G}} x_j, \qquad \bar{y}_{\mathcal{G}} := \frac{1}{G}\sum_{i\in\mathcal{G}} y_i.$$

Assumption 5 (Bounded leverage of the good cluster)**.** *There exists a constant $C>1$ such that*

$$\max_{j\in\mathcal{G}}\|x_j-\bar{x}_{\mathcal{G}}\| \le \frac{C}{G}\sum_{t\in\mathcal{G}}\|x_t-\bar{x}_{\mathcal{G}}\|.$$

Remark 5. *Assumption 5 always holds with $C=G$ since $\max_{j\in\mathcal{G}}\|x_j-\bar{x}_{\mathcal{G}}\|\le\sum_{t\in\mathcal{G}}\|x_t-\bar{x}_{\mathcal{G}}\|$. However, such a choice leads to a large robustness coefficient κ' in the subsequent analysis. In practice, when the good workers are well-clustered, C can be much smaller than G.*

For a good pivot $i\in\mathcal{G}$ we write

$$\alpha_i := \frac{|\mathcal{N}_{x_i}\setminus\mathcal{G}|}{G} = \frac{|\mathcal{G}\setminus\mathcal{N}_{x_i}|}{G}\in[0,1/2).$$

Since at most B Byzantine points exist overall, we always have

$$\alpha_i \le \frac{B}{G} = \frac{B}{n-B}.$$

Lemma 8 (NNM Pairing Property)**.** *For any pivot $\mu \in \mathbb{R}^d$, let $\mathcal{N}_\mu$ be the set of indices of the G nearest neighbors to μ. Then*

$$\sum_{b \in \mathcal{N}_\mu \setminus \mathcal{G}} \|x_b - \mu\| \leq \sum_{g \in \mathcal{G} \setminus \mathcal{N}_\mu} \|x_g - \mu\|. \tag{16}$$

Proof. Since $|\mathcal{N}_\mu| = |\mathcal{G}| = G$, we have $|\mathcal{N}_\mu \setminus \mathcal{G}| = |\mathcal{G} \setminus \mathcal{N}_\mu|$. Let $m = |\mathcal{N}_\mu \setminus \mathcal{G}|$.

Order the points in $\mathcal{N}_\mu \setminus \mathcal{G}$ as $b_1, \ldots, b_m$ and points in $\mathcal{G} \setminus \mathcal{N}_\mu$ as $g_1, \ldots, g_m$. By the definition of NNM, every point x_{b_j} in $\mathcal{N}_\mu$ is at least as close to μ as every point x_{g_j} not in $\mathcal{N}_\mu$. Therefore, for each $j \in [m]$:

$$\|x_{b_j} - \mu\| \leq \|x_{g_j} - \mu\|.$$

Summing over all j yields (16). □

Lemma 9 (General Pivot Bound)**.** *For any pivot $\mu \in \mathbb{R}^d$,*

$$\|y_\mu - \bar{x}_\mathcal{G}\| \leq \frac{2}{G} \sum_{j \in \mathcal{G} \setminus \mathcal{N}_\mu} \|x_j - \mu\|.$$

Proof. Write

$$y_\mu - \bar{x}_\mathcal{G} = \frac{1}{G} \left(\sum_{j \in \mathcal{N}_\mu} x_j - \sum_{j \in \mathcal{G}} x_j \right) = \frac{1}{G} \left(\sum_{j \in \mathcal{N}_\mu \setminus \mathcal{G}} x_j - \sum_{j \in \mathcal{G} \setminus \mathcal{N}_\mu} x_j \right).$$

Since $|\mathcal{N}_\mu \setminus \mathcal{G}| = |\mathcal{G} \setminus \mathcal{N}_\mu|$, we can subtract and add μ to obtain

$$y_\mu - \bar{x}_\mathcal{G} = \frac{1}{G} \left(\sum_{j \in \mathcal{N}_\mu \setminus \mathcal{G}} (x_j - \mu) - \sum_{j \in \mathcal{G} \setminus \mathcal{N}_\mu} (x_j - \mu) \right).$$

Taking norms and applying the triangle inequality:

$$\|y_\mu - \bar{x}_\mathcal{G}\| \leq \frac{1}{G} \left(\sum_{j \in \mathcal{N}_\mu \setminus \mathcal{G}} \|x_j - \mu\| + \sum_{j \in \mathcal{G} \setminus \mathcal{N}_\mu} \|x_j - \mu\| \right).$$

Applying Lemma 8:

$$\|y_\mu - \bar{x}_\mathcal{G}\| \leq \frac{2}{G} \sum_{j \in \mathcal{G} \setminus \mathcal{N}_\mu} \|x_j - \mu\|.$$

□

Lemma 10. *Under Assumption 5, for any good pivot $i \in \mathcal{G}$, letting $\alpha := \max_{i \in \mathcal{G}} \alpha_i \leq B/(n-B)$,*

$$\frac{1}{G} \sum_{i \in \mathcal{G}} \|y_i - \bar{x}_\mathcal{G}\| \leq \frac{4\alpha C}{G} \sum_{t \in \mathcal{G}} \|x_t - \bar{x}_\mathcal{G}\|.$$

Proof. By Lemma 9 with $\mu = x_i$:

$$\|y_i - \bar{x}_\mathcal{G}\| \leq \frac{2}{G} \sum_{j \in \mathcal{G} \setminus \mathcal{N}_{x_i}} \|x_j - x_i\|.$$

Applying the triangle inequality $\|x_j - x_i\| \leq \|x_j - \bar{x}_\mathcal{G}\| + \|x_i - \bar{x}_\mathcal{G}\|$, $|\mathcal{G} \setminus \mathcal{N}_{x_i}| = \alpha_i G$ and by Assumption 5:

$$\|y_i - \bar{x}_{\mathcal{G}}\| \leq \frac{2}{G} \sum_{j \in \mathcal{G} \setminus \mathcal{N}_{x_i}} (\|x_j - \bar{x}_{\mathcal{G}}\| + \|x_i - \bar{x}_{\mathcal{G}}\|)$$
$$\leq \frac{4|\mathcal{G} \setminus \mathcal{N}_{x_i}|}{G} \max_{j \in \mathcal{G} \setminus \mathcal{N}_{x_i}} \|x_j - \bar{x}_{\mathcal{G}}\| \leq \frac{4\alpha_i C}{G} \sum_{j \in \mathcal{G}} \|x_j - \bar{x}_{\mathcal{G}}\|.$$

Averaging over $i \in \mathcal{G}$ and applying $\alpha_i \leq \alpha$ we have the desired lemma. □

Lemma 11. *Under Assumption 5, letting $\alpha := \max_{i \in \mathcal{G}} \alpha_i \leq B/(n-B)$, we have*

$$\|\bar{y}_{\mathcal{G}} - \bar{x}_{\mathcal{G}}\| \leq \frac{4\alpha C}{G} \sum_{t \in \mathcal{G}} \|x_t - \bar{x}_{\mathcal{G}}\|, \tag{17}$$

$$\frac{1}{G} \sum_{i \in \mathcal{G}} \|y_i - \bar{y}_{\mathcal{G}}\| \leq \frac{8\alpha C}{G} \sum_{t \in \mathcal{G}} \|x_t - \bar{x}_{\mathcal{G}}\|. \tag{18}$$

Proof. For (17), since $\bar{y}_{\mathcal{G}} = \frac{1}{G} \sum_{i \in \mathcal{G}} y_i$, by triangle inequality:

$$\|\bar{y}_{\mathcal{G}} - \bar{x}_{\mathcal{G}}\| = \left\| \frac{1}{G} \sum_{i \in \mathcal{G}} (y_i - \bar{x}_{\mathcal{G}}) \right\| \leq \frac{1}{G} \sum_{i \in \mathcal{G}} \|y_i - \bar{x}_{\mathcal{G}}\|.$$

Applying Lemma 10 yields (17).

For (18), by the triangle inequality:

$$\|y_i - \bar{y}_{\mathcal{G}}\| \leq \|y_i - \bar{x}_{\mathcal{G}}\| + \|\bar{y}_{\mathcal{G}} - \bar{x}_{\mathcal{G}}\|.$$

Averaging over $i \in \mathcal{G}$ and applying Lemma 10 and (17):

$$\frac{1}{G} \sum_{i \in \mathcal{G}} \|y_i - \bar{y}_{\mathcal{G}}\| \leq \frac{4\alpha C}{G} \sum_{t \in \mathcal{G}} \|x_t - \bar{x}_{\mathcal{G}}\| + \frac{4\alpha C}{G} \sum_{t \in \mathcal{G}} \|x_t - \bar{x}_{\mathcal{G}}\| = \frac{8\alpha C}{G} \sum_{t \in \mathcal{G}} \|x_t - \bar{x}_{\mathcal{G}}\|.$$

□

Lemma 12 (Robustness after NNM)**.** *Let $F : \mathbb{R}^{d \times n} \to \mathbb{R}^d$ be (δ, κ)-robust in the sense of Definition 1. Under Assumption 5, for any $x_1, \ldots, x_n$,*

$$\left\| F\big(\texttt{NNM}(x_1, \ldots, x_n)\big) - \bar{x}_{\mathcal{G}} \right\| \leq \frac{\kappa'}{G} \sum_{t \in \mathcal{G}} \|x_t - \bar{x}_{\mathcal{G}}\|,$$

where $\kappa' = (8\kappa + 4)\alpha C$ and $\alpha \leq B/(n-B)$.

Proof. Let $(y_1, \ldots, y_n) = \texttt{NNM}(x_1, \ldots, x_n)$. By the triangle inequality:

$$\|F(y_1, \ldots, y_n) - \bar{x}_{\mathcal{G}}\| \leq \|F(y_1, \ldots, y_n) - \bar{y}_{\mathcal{G}}\| + \|\bar{y}_{\mathcal{G}} - \bar{x}_{\mathcal{G}}\|.$$

By the (δ, κ)-robustness of F (Definition 1):

$$\|F(y_1, \ldots, y_n) - \bar{y}_{\mathcal{G}}\| \leq \frac{\kappa}{G} \sum_{i \in \mathcal{G}} \|y_i - \bar{y}_{\mathcal{G}}\|.$$

Applying Lemma 11:

$$\begin{aligned}\|F(y_1,\dots,y_n)-\bar{x}_{\mathcal{G}}\| &\le \frac{\kappa}{G}\sum_{i\in\mathcal{G}}\|y_i-\bar{y}_{\mathcal{G}}\|+\|\bar{y}_{\mathcal{G}}-\bar{x}_{\mathcal{G}}\| \\ &\le \frac{8\kappa\alpha C}{G}\sum_{t\in\mathcal{G}}\|x_t-\bar{x}_{\mathcal{G}}\|+\frac{4\alpha C}{G}\sum_{t\in\mathcal{G}}\|x_t-\bar{x}_{\mathcal{G}}\| \\ &= \frac{(8\kappa+4)\alpha C}{G}\sum_{t\in\mathcal{G}}\|x_t-\bar{x}_{\mathcal{G}}\|.\end{aligned}$$

□

Remark 6. *The composition of NNM with a (δ,κ)-robust aggregator F yields a robust aggregator with constant $\kappa' = (8\kappa+4)\alpha C$, where $\alpha \le B/(n-B)$ measures the fraction of Byzantine contamination. When $B \ll n$, this additional factor is small, showing that NNM preprocessing preserves robustness up to a multiplicative constant depending on the Byzantine fraction and the leverage constant C.*

C.2. Geometric Median

The Geometric Median of $v_1,\dots,v_n \in \mathbb{R}^d$, denoted by $\mathrm{GM}(v_1,\dots,v_n)$, is defined as a vector that minimizes the sum of the ℓ_2-distances to these vectors:

$$\mathrm{GM}(v_1,\dots,v_n) \in \operatorname*{argmin}_{y\in\mathbb{R}^d}\sum_{i=1}^{n}\|y-v_i\|.$$

Proposition 1 (Geometric Median Robustness)**.** *Let $n\in\mathbb{N}$ and $\delta<\frac{1}{2}$ such that $B=\lfloor\delta n\rfloor<\frac{n}{2}$. The Geometric Median is (δ,κ)-robust with*

$$\kappa = 2\left(1+\frac{B}{n-2B}\right).$$

Proof. The proof follows the same approach as in [43] but stops before squaring both sides to match our linear robustness definition.

Let $v^* := \mathrm{GM}(v_1,\dots,v_n)$ be the geometric median of the input vectors. Consider any subset $S\subseteq[n]$ of size $|S|=\lfloor(1-\delta)n\rfloor = n-B$. By the reverse triangle inequality, for any $i\in S$, we have

$$\|v^*-v_i\| \ge |\|v^*-\bar{v}_S\|-\|v_i-\bar{v}_S\||.$$

Similarly, for any $i\in[n]\setminus S$, we obtain

$$\|v^*-v_i\| \ge |\|v_i-\bar{v}_S\|-\|v^*-\bar{v}_S\||.$$

Summing over all input vectors and using the fact that $|[n]\setminus S|\le B$:

$$\begin{aligned}\sum_{i\in[n]}\|v^*-v_i\| &\ge \sum_{i\in S}(|\|v^*-\bar{v}_S\|-\|v_i-\bar{v}_S\||)+\sum_{i\in[n]\setminus S}(|\|v_i-\bar{v}_S\|-\|v^*-\bar{v}_S\||) \\ &\ge (n-2B)\|v^*-\bar{v}_S\|-\sum_{i\in S}\|v_i-\bar{v}_S\|+\sum_{i\in[n]\setminus S}\|v_i-\bar{v}_S\|.\end{aligned}$$

Rearranging terms:

$$\|v^*-\bar{v}_S\| \le \frac{1}{n-2B}\left(\sum_{i\in[n]}\|v^*-v_i\|+\sum_{i\in S}\|v_i-\bar{v}_S\|-\sum_{i\in[n]\setminus S}\|v_i-\bar{v}_S\|\right).$$

By the definition of the geometric median:

$$\sum_{i\in[n]}\|v^*-v_i\| \le \sum_{i\in[n]}\|\bar{v}_S-v_i\|.$$

Therefore:

$$\begin{aligned}
\|v^* - \bar{v}_S\| &\leq \frac{1}{n-2B}\left(\sum_{i\in[n]}\|v_i - \bar{v}_S\| + \sum_{i\in S}\|v_i - \bar{v}_S\| - \sum_{i\in[n]\setminus S}\|v_i - \bar{v}_S\|\right) \\
&= \frac{1}{n-2B}\left(2\sum_{i\in S}\|v_i - \bar{v}_S\|\right) = \frac{2}{n-2B}\sum_{i\in S}\|v_i - \bar{v}_S\| \\
&= \frac{2(n-B)}{(n-2B)|S|}\sum_{i\in S}\|v_i - \bar{v}_S\| = 2\left(1 + \frac{B}{n-2B}\right)\frac{1}{|S|}\sum_{i\in S}\|v_i - \bar{v}_S\|.
\end{aligned}$$

This establishes the desired robustness property with $\kappa = 2\left(1 + \frac{B}{n-2B}\right)$. □

C.3. Coordinate-wise Median

For input vectors $v_1, \ldots, v_n \in \mathbb{R}^d$, their coordinate-wise median, denoted by $\mathrm{CWMed}(v_1, \ldots, v_n)$, is defined as a vector whose k-th coordinate is:

$$[\mathrm{CWMed}(v_1, \ldots, v_n)]_k := \mathrm{Median}\left([v_1]_k, \ldots, [v_n]_k\right).$$

Lemma 13 (Coordinate-wise Robustness)**.** *Assume that $F : \mathbb{R}^{d\times n} \to \mathbb{R}^d$ is a coordinate-wise aggregation function, i.e., there exist d real-valued functions $F_1, \ldots, F_d : \mathbb{R}^n \to \mathbb{R}$ such that for all $v_1, \ldots, v_n \in \mathbb{R}^d$, $[F(v_1, \ldots, v_n)]_k = F_k([v_1]_k, \ldots, [v_n]_k)$. If for each $k \in [d]$, F_k is (δ, κ)-robust (in the sense of Definition 1 applied to scalars), then F is $(\delta, \kappa\sqrt{d})$-robust.*

Proof. Since each F_k is (δ, κ)-robust, for any subset $\mathcal{G}$ of good workers with $|\mathcal{G}| = G \geq (1-\delta)n$:

$$|[F(v_1, \ldots, v_n)]_k - [\bar{v}_\mathcal{G}]_k| \leq \frac{\kappa}{G}\sum_{i\in\mathcal{G}}|[v_i]_k - [\bar{v}_\mathcal{G}]_k|.$$

Summing over all coordinates:

$$\sum_{k=1}^{d}|[F(v)]_k - [\bar{v}_\mathcal{G}]_k| \leq \frac{\kappa}{G}\sum_{i\in\mathcal{G}}\sum_{k=1}^{d}|[v_i]_k - [\bar{v}_\mathcal{G}]_k| = \frac{\kappa}{G}\sum_{i\in\mathcal{G}}\|v_i - \bar{v}_\mathcal{G}\|_1.$$

Using $\|a\|_2 \leq \|a\|_1$ on the left and $\|a\|_1 \leq \sqrt{d}\|a\|_2$ on the right:

$$\|F(v) - \bar{v}_\mathcal{G}\| \leq \frac{\kappa\sqrt{d}}{G}\sum_{i\in\mathcal{G}}\|v_i - \bar{v}_\mathcal{G}\|.$$

□

Proposition 2 (Coordinate-wise Median Robustness)**.** *Let $n \in \mathbb{N}$ and $\delta < \frac{1}{2}$ such that $B = \lfloor \delta n \rfloor < \frac{n}{2}$. The Coordinate-wise Median is (δ, κ)-robust with*

$$\kappa = 2\sqrt{d}\left(1 + \frac{B}{n-2B}\right),$$

where d is the dimension of the input vectors.

Proof. Since geometric median coincides with median for one-dimensional inputs, it follows from Proposition 1 that for one-dimensional inputs, median is (δ, κ_0)-robust with $\kappa_0 = 2\left(1 + \frac{B}{n-2B}\right)$. Since coordinate-wise median is a coordinate-wise aggregation function, we apply Lemma 13 to obtain that the d-dimensional coordinate-wise median is $(\delta, \kappa_0\sqrt{d})$-robust, giving the stated bound. □

Remark 7. *The $\sqrt{d}$ factor arises from converting between ℓ_1 and ℓ_2 norms when combining per-coordinate bounds into a vector bound. This dimension dependence is a known limitation of coordinate-wise aggregation rules under linear robustness definitions. In contrast, the geometric median (Proposition 1) achieves dimension-independent robustness because it operates directly on the full vectors.*

D. Training Curves

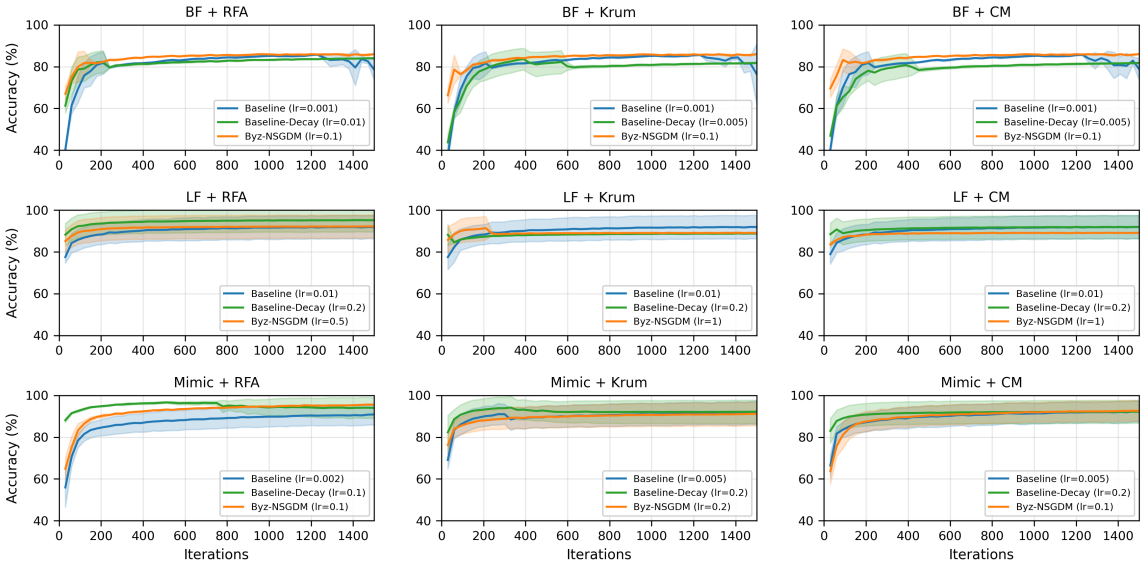

Figure 2: Test accuracy curves on heterogeneous MNIST under Byzantine attacks. Each row corresponds to a different attack (BF, LF, Mimic), and each column to a different aggregator (RFA, Krum, CM). Lines show mean accuracy over 3 seeds with shaded regions indicating ± 1 standard deviation.

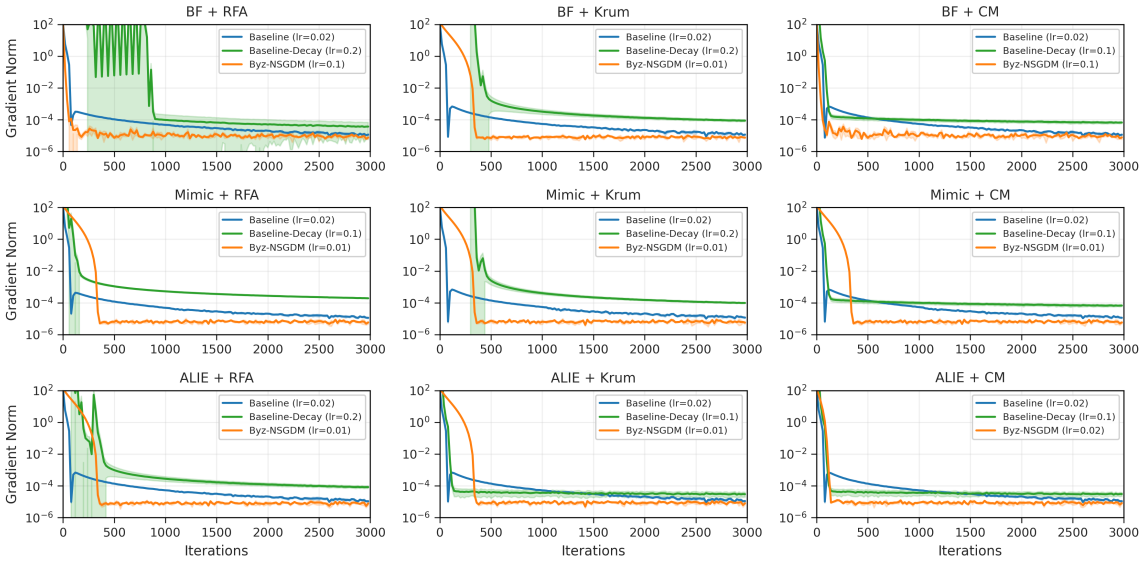

Figure 3: Gradient norm evolution (log scale) for synthetic (L_0, L_1)-smooth optimization under Byzantine attacks. Each row shows a different attack (BF, Mimic, ALIE), and each column a different aggregator (RFA, Krum, CM). Lines show mean gradient norm over 3 seeds with shaded regions indicating ± 1 standard deviation. Legend includes tuned learning rates.

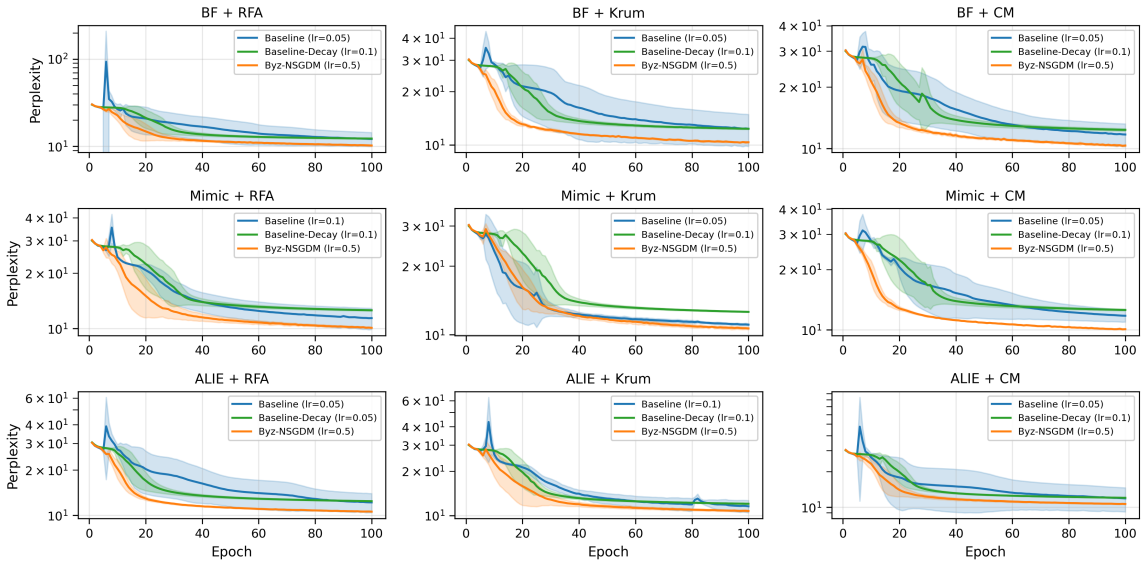

Figure 4: Validation perplexity (log scale) on Shakespeare character-level language modeling under Byzantine attacks. Each row shows a different attack (BF, Mimic, ALIE), and each column a different aggregator (RFA, Krum, CM).