Byzantine-Resilient Federated Learning under Heterogeneity and Heavy Tails

Youming Tao, Zuyuan Zhang, Di Wang, *Member, IEEE*, Dongxiao Yu, *Senior Member, IEEE*, Xiuzhen Cheng, *Fellow, IEEE*, Falko Dressler *Fellow, IEEE*

***Abstract*—Byzantine resilience is essential in federated learning (FL) to safeguard model training from malicious or faulty participants. However, existing Byzantine-resilient methods struggle when faced with heavy-tailed gradient noise, a common challenge in heterogeneous environments. In this work, we propose a Byzantine-resilient FL framework specifically designed to handle both heterogeneity and heavy-tailed noise. Our approach builds on robust distributed stochastic heavy-ball optimization, incorporating update normalization and gradient/momentum clipping to mitigate the effects of heavy-tailed noise. We establish the first high-probability convergence guarantees for Byzantine-resilient FL under these conditions, showing that our algorithms achieve optimal Byzantine resilience and align with known lower bounds. Additionally, we introduce an efficient variant of the nearest neighbor mixing technique, leveraging random projections to significantly reduce computational costs in high-dimensional settings. Through rigorous theoretical analysis and extensive empirical evaluations, we demonstrate that our methods outperform existing approaches in robustness against both Byzantine failures and heavy-tailed noise.**

***Index Terms*—Federated Learning, Byzantine resilience, heterogeneity, heavy tails**

I. Introduction

FEDERATED learning (FL) [1] has emerged as a key paradigm for distributed machine learning, enabling multiple computing devices to collaboratively train models without directly sharing their private data. By decentralizing the learning process across a large-scale network, FL is particularly suited for applications involving mobile devices [2], IoT networks [3], and edge computing environments [4].

Despite its advantages, FL faces significant challenges in terms of reliability and robustness. A major concern is the presence of faulty or adversarial devices that deviate from the expected protocol due to hardware/software failures, communication disruptions, or malicious attacks. These unpredictable behaviors are often modeled as Byzantine faults [5], where a fraction of participating devices may send arbitrary, misleading updates. Even a small number of Byzantine devices, or in some cases, a single one, can severely degrade model accuracy, potentially causing the entire FL process to fail [6]. As a result, achieving Byzantine resilience, the ability to maintain high model accuracy despite adversarial failures, has become a critical area of research in FL.

Significant progress has been made in developing Byzantine-resilient FL algorithms, particularly in settings where data across clients are assumed to be independent and identically distributed (i.i.d.). In such cases, various robust aggregation rules and Byzantine-resilient optimization techniques have been proposed and demonstrated strong convergence properties (see Section II). However, the i.i.d. assumption rarely holds in practical FL applications [7]. In real-world scenarios, each client typically collects data locally, leading to significant variations across participants. This inherent heterogeneity complicates Byzantine resilience, making it difficult to differentiate legitimate statistical variations from adversarial manipulations. Recent theoretical findings have shown that Byzantine resilience in heterogeneous FL is fundamentally constrained, with inevitable accuracy degradation due to data heterogeneity [8], [9], [10]. Recent work [11] achieved optimal Byzantine resilience by integrating the pre-aggregation technique, *Nearest Neighbor Mixing (NNM)*, into robust distributed stochastic heavy-ball (D-SHB) based algorithms. This approach aligns with the lower bounds established in prior studies [10]. However, these solutions rely on a common assumption in FL literature, the stochastic gradient noise follows a light-tailed distribution with *finite* variance.

While this finite variance assumption holds in many cases, it fails to capture scenarios where gradient noise follows a heavy-tailed distribution. Recent empirical studies [12], [13] have shown that heavy-tailed noise is prevalent in FL, especially in heterogeneous data cases. In such cases, stochastic gradient noise may have infinite variance, causing instability and performance degradation in existing Byzantine-resilient FL algorithms. Specifically, the challenges posed by heavy-tailed noise in FL are twofold: (1) *On the empirical side*, heavy-tailed noise can degrade the learning accuracy, and cause dramatic accuracy fluctuations during training even in the absence of Byzantine failures, making convergence unpredictable; (2) *From a theoretical standpoint*, existing FL convergence analyses rely on gradient variance, which are not applicable under heavy tails. Their in-expectation guarantees are also insufficient, as individual training runs may produce extreme outliers. Given the high computational cost of training modern deep learning models, it is crucial to establish *high-*

Youming Tao is with the School of Electrical Engineering and Computer Science, TU Berlin, 10623 Berlin, Germany (email: tao@ccs-labs.org).

Zuyuan Zhang is with the School of Engineering and Applied Science, George Washington University, D.C. 20052, U.S. (email: zuyuan.zhang@gwu.edu).

Di Wang is with the Division of Computer, Electrical and Mathematical Sciences and Engineering, King Abdullah University of Science and Technology, Saudi Arabia (email: di.wang@kaust.edu.sa).

Dongxiao Yu is with the School of Computer Science and Technology, Shandong University, Qingdao 266237, China (email: dxyu@sdu.edu.cn).

Xiuzhen Cheng is with the School of Computer Science and Technology, Shandong University, Qingdao 266237, China (email: xzcheng@sdu.edu.cn).

Falko Dressler is with the School of Electrical Engineering and Computer Science, TU Berlin, 10623 Berlin, Germany. (email: dressler@ccs-labs.org).

probability convergence guarantees to ensure reliable performance across all training runs.

A. Our Contributions

Motivated by the challenges above, we study this fundamental question: *Can we achieve Byzantine resilience in heterogeneous FL under heavy-tailed gradient noise with infinite variance? If so, how to establish high-probability convergence guarantees?* This paper provides affirmative answers to both questions. Our main contributions are summarized as follows:

1. Byzantine-Resilient Optimization under Heavy-Tailed Gradient Noise: We introduce a new Byzantine-resilient FL framework based on the robust distributed stochastic heavy ball (D-SHB) algorithm. Our approach leverages *normalized momentum* along with *clipping techniques* to mitigate the impact of heavy-tailed noise. Two specific clipping variants *gradient clipping* and *momentum clipping* are explored.

2. High-Probability Convergence Guarantees: We establish the first high-probability convergence guarantees for Byzantine-resilient heterogeneous FL under heavy-tailed gradient noise. By introducing a *surely one-step descent bound*, we characterize the dynamics of our proposed algorithms. Specifically, we prove that, with high probability, (gradient or momentum) clipping-based approaches can reach a neighborhood of a stationary point with an error of $O(\sqrt{\delta}H)$ at a rate of $\widetilde{O}(((1/n+\sqrt{\delta})T^{-1})^{\frac{h-1}{3h-2}})$, where $h \in (1, 2]$ denotes the tail index (the maximum finite moment of the stochastic gradient), H denotes the data heterogeneity level, δ is the fraction of Byzantine clients, and n is the total client number. Notably, the convergence error is optimal and the convergence rate is optimal when $\delta = 0$.

3. Computationally Efficient Byzantine-Resilient Aggregation: We identify a computational bottleneck in the state-of-the-art Byzantine-resilient pre-aggregation technique known as *Nearest Neighbor Mixing (*`NNM`*)*, which incurs significant computational costs in high-dimensional settings. To address this, we propose a computationally efficient variant called *Approximate Nearest Neighbor Mixing via Random Projection (*`ANNM-RP`*)*. By leveraging random projection techniques, we reduce the computational complexity from $O(dn^2)$ in `NNM` to $O(n^3 \log n)$ at the server while keeping the computational complexity at the client side unchanged, significantly improving scalability for high-dimensional models.

B. Technical Challenges

A central challenge in this work is to establish time-uniform high-probability gradient estimation guarantees for Byzantine-resilient FL under data heterogeneity and heavy-tailed noise. While recent work [14] provides high-probability analysis for normalized methods with clipping in the non-federated setting, and [11] introduces nearest neighbor mixing to amplify Byzantine robustness, these results cannot be directly combined. In Byzantine FL, the server update is produced via a robust aggregation rule applied to multiple client updates, whose guarantees control the aggregation error only indirectly through the deviation among honest clients, rather than via direct stochastic concentration. Moreover, existing analyses of robust aggregation rely on expectation-based arguments, which are insufficient in heavy-tailed settings where rare but large deviations must be controlled with high probability. Showing that such robustly aggregated and normalized updates still admit a high-probability descent guarantee therefore requires new analysis. In addition, we study momentum clipping under heavy-tailed noise, which has not been considered in prior works: in this case, bounded h-th moments of stochastic gradients do not directly imply sharp bounds on the h-th moments of the momentum sequence due to temporal accumulation, introducing an additional technical challenge.

C. Notations

Throughout the paper, we use the following notations. We use $\|\cdot\|$ to denote the ℓ_2 norm for any vector $v \in \mathbb{R}^p$, i.e., $\|v\| \coloneqq (\sum_{i=1}^p v_i^2)^{\frac{1}{2}}$. For any two real numbers x and y, the notation $x \wedge y \coloneqq \min(x, y)$ and $x \vee y \coloneqq \max(x, y)$. We use standard O-notation to hide absolute constants and $\widetilde{O}$ notation to further omit poly-logarithmic factors.

II. Related Work

Byzantine Resilience in Federated Learning: Byzantine resilience has become a critical aspect of FL due to the increasing vulnerability of distributed systems. Various Byzantine attacks have been proposed in the FL setting [15], [16], [17], [18], [10], underscoring the need for robust defense mechanisms. Classical approaches address Byzantine resilience by replacing vanilla averaging with robust aggregation rules, such as coordinate-wise median [6], coordinate-wise mean [6], geometric median [19], [20], and majority voting [21], [22]. Some methods further enhance robustness by combining these estimators; for example, Krum [23] integrates majority voting and geometric median, while Bulyan [24] refines Krum with coordinate-wise trimmed mean to ensure majority agreement. While these techniques are effective in homogeneous FL, they struggle in heterogeneous FL. Recent studies [10], [11] introduced a new robustness criterion for aggregation rules, showing that most prior defenses fail under heterogeneity. To address this, pre-aggregation techniques such as bucketing [10] and nearest neighbor mixing (`NNM`) [11] have been proposed. However, these methods assume gradient noise has finite variance, making them ineffective for heavy-tailed noise. This limitation motivates our study on Byzantine-resilient FL that accounts for both data heterogeneity and heavy-tailed noise.

Heavy-Tailed Noise in Federated Learning: Heavy-tailed noise is widely observed in deep learning optimization [25], [26], [27], [28], where the commonly assumed finite variance condition often fails. This leads to drastically different learning dynamics, as analyzed in prior works [29], [30], [31], [32]. Recent empirical studies [13] demonstrated that heavy-tailed gradient noise frequently arises due to data heterogeneity in FL. Our work takes this further by developing Byzantine-resilient FL algorithms that explicitly handle heavy-tailed noise. A very recent study [33] also explored Byzantine resilience under heavy-tailed noise but in a rather different setting. Specifically, their approach assumes each gradient component has bounded variance, which extends previous

assumptions but remains restrictive. In contrast, our work considers *a more general setting*, requiring only a bounded h-th moment with the tail index $h \in (1, 2]$, allowing cases where gradient noise has *infinite variance*.

III. Preliminaries

A. Federated Learning Setup

We consider a federated learning system with n clients coordinated by a central server. The clients can be partitioned into two groups: $[n] = \mathcal{H} \uplus \mathcal{B}$, where $\mathcal{H}$ represents the set of honest clients and $\mathcal{B}$ denotes the set of Byzantine clients. The objective is to solve the following optimization problem:

$$\min_{x \in \mathbb{R}^d} \mathcal{L}_{\mathcal{H}}(x) \coloneqq \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \mathcal{L}_i(x), \tag{1}$$

where $\mathcal{L}_i(x)$ is the (non-convex) local objective function of client i. Let $\mathcal{Z}$ be the data space, and let $\ell : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}$ represent the point-wise loss function. Each client i has a local data distribution $\mathcal{D}_i$ over $\mathcal{Z}$, and the corresponding local objective function is defined as the *expected* loss:

$$\mathcal{L}_i(x) \coloneqq \mathbb{E}_{z \sim \mathcal{D}_i}[\ell(x; z)]. \tag{2}$$

Assumption 1 (L-Smooth Function)**.** Each local objective function $\mathcal{L}_i$ satisfies L-smoothness, i.e., for all $x, x' \in \mathbb{R}^d$, it holds that $\|\nabla \mathcal{L}_i(x) - \nabla \mathcal{L}_i(x')\| \leq L \cdot \|x - x'\|$.

Since the loss function is generally non-convex (e.g., in neural networks), finding a global minimizer of $\mathcal{L}_{\mathcal{H}}$ is NP-hard [34]. Instead, we aim to approximate a stationary point x such that $\|\nabla \mathcal{L}_{\mathcal{H}}(x)\| \approx 0$, assuming $\mathcal{L}_{\mathcal{H}}$ admits such a stationary point.

B. Byzantine Fault Model

We consider an adversarial fault model where f clients, whose identities are unknown, act as *Byzantine* clients. These Byzantine clients are **unrestricted** in behavior and are assumed to be *omniscient*, meaning they have full knowledge of the learning algorithm and the states of all other clients. Byzantine clients may send arbitrary updates to the server and can collude to disrupt training.

Following standard Byzantine FL assumptions [10], [11], [35], the set of Byzantine clients $\mathcal{B} \subseteq [n]$ remains fixed over time, with size f bounded such that the fraction of Byzantine clients does not exceed δ, i.e., $|\mathcal{B}| = f = \delta \cdot n$.

C. Byzantine Resilience and Heterogeneity Challenge

An FL algorithm is *Byzantine resilient* if it can achieve an ϵ-approximate stationary point for $\mathcal{L}_{\mathcal{H}}$ despite the presence of a Byzantine fraction δ. We formalize this through the concept of (ϵ, δ)-Byzantine resilience:

Definition 1 ((ϵ, δ)-Byzantine Resilience)**.** An FL algorithm satisfies (ϵ, δ)-Byzantine resilience if, even in the presence of a Byzantine fraction δ, it produces a sequence of model parameters $\{x_0, \cdots, x_{T-1}\}$ such that $\frac{1}{T} \sum_{t=1}^{T} \|\nabla \mathcal{L}_{\mathcal{H}}(x_{t-1})\| \leq \epsilon$.

We focus on cases where $\delta \in [0, 0.5)$, as achieving (ϵ, δ)-Byzantine resilience is impossible for any $\epsilon > 0$ when $\delta \geq 0.5$, demonstrated in [9].

In practical distributed systems, data across clients are often *non-i.i.d.*, leading to variations in local objective functions. This heterogeneity complicates Byzantine resilience, as distinguishing Byzantine-injected gradients from naturally outlying gradients becomes more difficult. To formally characterize data heterogeneity, we adopt a standard assumption from previous Byzantine FL studies [10], [11], [35].

Assumption 2 (Bounded Heterogeneity)**.** There exists a constant $H \geq 0$ such that, for any model $x \in \mathbb{R}^d$,

$$\frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \|\nabla \mathcal{L}_i(x) - \nabla \mathcal{L}(x)\|^2 \leq H. \tag{3}$$

Here, H quantifies the level of data heterogeneity, with $H = 0$ corresponding to a fully homogeneous setting where all local data distributions—and thus local objective functions—are identical. In Byzantine-resilient non-convex optimization, heterogeneity significantly complicates convergence. As established in [10], any (ϵ, δ)-Byzantine resilient FL algorithm must satisfy the following fundamental accuracy limit.

Proposition 1 (Error Lower Bound for Byzantine FL with Heterogeneity [10], Theorem 3)**.** For any (ϵ, δ)-Byzantine resilient FL algorithm operating under a data heterogeneity level H, $\epsilon \geq \Omega(\sqrt{\delta} H)$.

This bound highlights an inherent trade-off: as data heterogeneity increases, the best achievable accuracy in a Byzantine-resilient FL system degrades accordingly.

D. Phenomenon of Heavy Tails in FL

Existing performance analyses of Byzantine-resilient FL algorithms typically assume that stochastic gradients (associated with point-wise losses) have uniformly bounded variance:

$$\mathbb{E}_{z \sim \mathcal{D}_i}[\|\nabla \ell(x; z) - \nabla \mathcal{L}_i(x)\|^2] \leq \sigma^2, \forall i \in [n], x \in \mathbb{R}^d. \tag{4}$$

This assumption holds under light-tailed noise distributions, such as sub-Gaussian or sub-exponential distributions. However, recent studies [13] have shown that this assumption fails in FL settings, particularly under heterogeneous data, where gradient noise often follows heavy-tailed distributions (e.g., Pareto, Student's t, and power-law distributions).

To account for the impact of heavy-tailed gradient noise, we adopt the bounded h-th moment assumption introduced in [13].

Assumption 3 (Bounded h-th moment)**.** There exists a real number $h \in (1, 2]$ and a constant $G \geq 0$ such that,

$$\begin{cases} \mathbb{E}_{z \sim \mathcal{D}_i}[\|\nabla \ell(x; z) - \nabla \mathcal{L}_i(x)\|^h] \leq G^h \\ \mathbb{E}_{z \sim \mathcal{D}_i}[\|\nabla \ell(x; z)\|^h] \leq G^h \end{cases}, \forall i \in [n], x \in \mathbb{R}^d. \tag{5}$$

Here, h is the *tail index*, which quantifies the extent of heavy-tailed behavior in the gradient noise distribution. A smaller h value indicates a heavier-tailed distribution, meaning the gradient noise exhibits more extreme fluctuations.

TABLE I
ROBUSTNESS COEFFICIENT κ FOR DIFFERENT (δ, κ)-ROBUST AGGREGATION RULES, IGNORING NUMERICAL CONSTANTS.

Robust Aggregation Rule	Robustness Coefficient κ
`GMed`	$O((1+\frac{\delta}{1-2\delta})^2)$
`CWTM`	$O(\frac{\delta}{1-2\delta}(1+\frac{\delta}{1-2\delta}))$
`CWMed`	$O((1+\frac{\delta}{1-2\delta})^2)$
`Krum`	$O(1+\frac{\delta}{1-2\delta})$

Algorithm 1: `NNM`: Nearest Neighbor Mixing

1 **Input:** number of inputs n, vectors $v_1, \ldots, v_n \in \mathbb{R}^d$, fraction of Byzantine inputs $\delta < 1/2$.
2 **for** $i = 1 \ldots n$ **do**
3 Sort inputs to get $(v_{i:1}, \ldots, v_{i:n})$ such that

$$\|v_{i:1} - v_i\| \leq \ldots \leq \|v_{i:n} - v_i\| \; ;$$

4 Average the $(1-\delta)n$ nearest neighbors of v_i,

$$v_i' = \frac{1}{(1-\delta)n} \sum_{j=1}^{(1-\delta)n} v_{i:j} \; ;$$

5 **return** $v_1', \ldots, v_n'$;

E. Robust Aggregation Rule

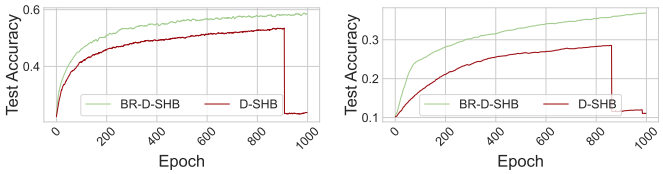
A core component of Byzantine-resilient FL methods is the use of a robust aggregation rule, denoted as `Agg`. To formally assess its robustness, we adopt the concept of (δ, κ)-robustness, adapted from [11].

Definition 2 ((δ, κ)-Robustness)**.** Let $\delta < 0.5$ and $\kappa \geq 0$. An aggregation rule $\texttt{Agg} : \mathbb{R}^{d \times n} \to \mathbb{R}^d$ is said to be (δ, κ)-robust if, for any input vectors $v_1, \cdots, v_n \in \mathbb{R}^d$ and any subset $S \subseteq [n]$ of size $(1-\delta)n$,

$$\|\texttt{Agg}(v_1, \cdots, v_n) - \overline{v}_S\|^2 \leq \frac{\kappa}{|S|} \sum_{i \in S} \|v_i - \overline{v}_S\|^2, \quad (6)$$

where $\overline{v}_S = \frac{1}{|S|} \sum_{i \in S} v_i$ is the average of the honest inputs.

Here, the parameter κ is referred to as the robustness coefficient, quantifying how well the aggregation rule estimates the mean of honest inputs. This definition ensures that the aggregation rule's error in approximating the honest input mean is bounded by κ times the empirical variance of the honest inputs. It generalizes various prior robustness notions, such as $(\delta_{\max}, c)$-ARAGG [35], [10] and (f, λ)-resilient averaging [36]. According to [11], several existing aggregation rules, such as coordinate-wise trimmed mean (`CWTM`) [6], `Krum` [23], geometric median (`GMed`) [19], and coordinate-wise median (`CWMed`) [20], satisfy (δ, κ)-robustness with different robustness coefficients κ, as summarized in Table I. Furthermore, it is established that for $\forall \delta < \frac{1}{2}$, no robust aggregation rule can achieve (δ, κ)-robustness with $\kappa < O(\frac{\delta}{1-2\delta})$.



(a) Byzantine-free case (b) Under ALIE Attack

Fig. 1. Catastrophic failures occur during training on CIFAR-10 in the **(a)** Byzantine-free setting and **(b)** under ALIE attack when using standard D-SHB. The use of $\texttt{GMed} \circ \texttt{NNM}$ (BR-D-SHB) effectively mitigates these failures.

Proposition 2 (Robust Coefficient Lower Bound, Essentially from [11, Proposition 6])**.** For any (δ, κ)-robust aggregation rule with $\delta < 1/2$, we have $\kappa \geq \Omega(\frac{\delta}{1-2\delta})$.

The lower bound implies that, in general, a robust aggregation rule cannot provide an estimate arbitrarily close to the average of honest inputs. Additionally, existing rules, such as `Krum`, `GMed`, and `CWMed`, remain suboptimal as their robustness coefficients exceed this lower bound. To enhance robustness, Nearest Neighbor Mixing (`NNM`) [11] has been introduced as a pre-aggregation technique that reduces the impact of Byzantine clients before applying `Agg`. The `NNM` procedure is summarized in Algorithm1, where the mixing step in line 4 smooths out Byzantine outliers before applying an aggregation rule, improving robustness.

Proposition 3. For any aggregation rule $\texttt{Agg} : \mathbb{R}^{d \times n} \to \mathbb{R}^d$ that is (δ, κ)-robust with $\delta < \frac{1}{2}$, then applying `NNM` before aggregation, denoted as $\texttt{Agg} \circ \texttt{NNM}$, yields an improved robustness coefficient: $\kappa' \leq 16 \frac{\delta}{1-\delta}(\kappa + 1)$.

Following [11], [10], assume there exists an arbitrarily small constant $\nu > 0$ such that $n \geq (2+\nu)f$, equivalently $\delta \leq \frac{1}{2+\nu}$. Under this condition, all robust aggregation rules `Agg` listed in Table I are (δ, κ)-robust with $\kappa = O(1)$. Moreover, it holds that $\delta \leq \frac{\delta}{1-\delta} \leq \frac{2+\nu}{1+\nu}\delta$ and $\delta \leq \frac{\delta}{1-2\delta} \leq \frac{2+\nu}{\nu}\delta$, which implies, $\frac{\delta}{1-\delta} = \Theta(\delta)$ and $\frac{\delta}{1-2\delta} = \Theta(\delta)$. Thus, $\texttt{Agg} \circ \texttt{NNM}$ ensures optimal robustness. We note that `CWTM` achieves an order-optimal robustness coefficient of $O(\delta)$ even without the use of `NNM`, matching the known lower bound. While `CWTM` is theoretically optimal, its empirical performance can be further improved when combined with `NNM`, as demonstrated in [11].

IV. BRITTLENESS OF EXISTING SOLUTIONS UNDER HETEROGENEITY AND HEAVY-TAILED NOISE

We demonstrate that existing Byzantine-resilient FL algorithms struggle under heavy-tailed gradient noise, leading to degraded learning performance. We thus conduct empirical studies on benchmark datasets. Previous work [13] observed catastrophic model failure in standard non-Byzantine FL algorithms that rely purely on stochastic gradients. This phenomenon is characterized by a sudden and severe drop in learning accuracy due to heavy-tailed noise. Here, we extend this analysis to the distributed stochastic heavy-ball (D-SHB) algorithm, which incorporates momentum in addition to stochastic gradients. Our findings confirm that D-SHB also suffers from catastrophic failures in training under heavy-tailed noise. However, our experiments reveal that applying

TABLE II
TEST ACCURACIES OF STATE-OF-THE-ART BYZANTINE-RESILIENT FL ALGORITHMS ON CIFAR-10 UNDER BOTH HEAVY-TAILED AND LIGHT-TAILED NOISE CONDITIONS.

Robust Aggregation Rule	Noise Condition	
	Heavy-Tailed	Light-Tailed
NNM ∘ GMed	39.12	46.41
NNM ∘ Krum	46.88	56.52
NNM ∘ CWTM	40.34	48.21

Byzantine-robust techniques mitigates these failures. Specifically, these methods help filter out outlier gradients caused by heavy-tailed noise, thereby reducing its adverse impact on convergence. The effectiveness of such techniques is illustrated in Figure 1, which shows that while standard D-SHB fails catastrophically, the use of NNM and Byzantine-robust aggregation rules (BR-D-SHB) significantly improves stability.

While Byzantine-robust techniques help mitigate the effects of heavy-tailed noise, they remain insufficient for fully addressing its challenges. The key limitation is that these techniques are designed to handle adversarial clients—not naturally occurring heavy-tailed noise, which affects both honest and Byzantine clients alike. In particular, Byzantine-robust aggregation rules ensure robustness only when the fraction of adversarial clients remains below a half. However, when heavy-tailed noise is present across all clients, its effects cannot be entirely removed using existing Byzantine-robust techniques. To further analyze this issue, we compare the performance of state-of-the-art Byzantine-resilient FL algorithms under both light-tailed and heavy-tailed noise conditions. The results, summarized in Table II, reveal that while these algorithms perform well in light-tailed scenarios, their accuracy significantly degrades in heavy-tailed scenarios. This highlights the brittleness of existing Byzantine-resilient FL algorithms when exposed to heavy-tailed noise. Developing robust solutions that can effectively handle both Byzantine faults and heavy-tailed noise remains an important open gap.

V. Robust Distributed Stochastic Heavy Ball with Clipping and Normalization

A. Algorithm Design

We introduce a Byzantine-resilient FL framework designed to address both data heterogeneity and heavy-tailed gradient noise. This framework extends the conventional distributed stochastic heavy ball (D-SHB) algorithm, which has demonstrated effectiveness in Byzantine-resilient FL [37], [38], [36]. However, unlike existing methods, our approach explicitly mitigates the challenges posed by heavy-tailed noise and heterogeneity by incorporating two key components:

1) **Clipping:** Reduces the impact of outliers caused by heavy-tailed noise. We explore two variants: (1) *Gradient Clipping* which Limits individual stochastic gradient magnitudes and (2) *Momentum Clipping* which Applies clipping to the local momentum updates. We focus on gradient clipping in this section and will discuss momentum clipping in the next one.

Algorithm 2: Robust D-SHB with Gradient Clipping and Momentum Normalization

Input: Initial model x_0, initial momentum $m_0^{(i)} = 0$ for honest clients, robust aggregation $\mathtt{Agg}$, learning rate η, momentum coefficient β, clipping threshold λ and number of steps T.
for $t = 1 \dots T$ **do**
  Server broadcasts x_{t-1} to all clients;
  for every *honest client* i, **in parallel do**
    Compute a stochastic gradient: $g_t^{(i)} = \nabla \ell(x_{t-1}, z_i^t)$;
    Gradient Clipping: $\widetilde{g}_t^{(i)} = \left(1 \wedge \frac{\lambda}{\|g_t^{(i)}\|}\right) \cdot g_t^{(i)}$;
    Update local momentum: $m_t^{(i)} = \beta m_{t-1}^{(i)} + (1-\beta)\widetilde{g}_t^{(i)}$;
    Send $m_t^{(i)}$ to the server;
  Server aggregates the momenta: $\widehat{m}_t = \mathtt{Agg}(m_t^{(1)}, \dots, m_t^{(n)})$;
  Server updates the model: $x_t = x_{t-1} - \eta \cdot \frac{\widehat{m}_t}{\|\widehat{m}_t\|}$;
return $x_0, \dots, x_T$;

2) **Normalization:** Normalizing the model updates, which is the global momentum in our framework, ensures that the descent direction, rather than raw gradient magnitude, guides model updates. This stabilization simplifies the convergence analysis and improves robustness.

To maximize the synergy between clipping and normalization, careful tuning of the clipping threshold λ is essential. Overly aggressive clipping may distort optimization dynamics, while insufficient clipping may fail to filter out extreme values. Striking a balance is key to achieving both robustness and effective convergence.

The algorithm proceeds over T iterations, where in each iteration the server and clients interact as follows:

1) **Model Broadcast:** At the start of iteration t, the server shares the current model x_{t-1} with all clients.
2) **Local Gradient Computation and Clipping:** Each *honest* client samples data z_i^t and computes its stochastic gradient $g_t^{(i)} = \nabla \ell(x_{t-1}, z_i^t)$. To control extreme gradient values, gradient clipping is then applied: $\widetilde{g}_t^{(i)} = (1 \wedge \frac{\lambda}{\|g_t^{(i)}\|}) \cdot g_t^{(i)}$.
3) **Local Momentum Update:** Each *honest* client uses the clipped gradient to update the local momentum as $m_t^{(i)} = \beta m_{t-1}^{(i)} + (1-\beta)\widetilde{g}_t^{(i)}$
4) **Robust Aggregation:** The server aggregates received local momenta by any (δ, κ)-robust aggregation rule $\mathtt{Agg}$ with $\kappa = O(\frac{\delta}{1-2\delta})$: $\widehat{m}_t = \mathtt{Agg}(m_t^{(1)}, \dots, m_t^{(n)})$.
5) **Model Update with Momentum Normalization:** The server updates the model parameter with normalized momentum, which ensures that only the descent direction influences updates: $x_t = x_{t-1} - \eta \frac{\widehat{m}_t}{\|\widehat{m}_t\|}$.

B. Theoretical Results and Analysis

We begin our convergence analysis by introducing a key tool: a dimension-free concentration bound for heavy-tailed random vectors under clipping, adapted from [14].

Lemma 1 (Concentration Bound for clipped heavy-tailed random vectors, essentially from [14, Lemma 8])**.** Suppose $X_1, X_2, \cdots, X_N$ are random vectors in a Hilbert space adapted to a filtration $\mathcal{F}_1, \mathcal{F}_2, \cdots$, with means $\mathbb{E}_{k-1}[X_k] = \mu_k$. Suppose there is some constant $h \in (1, 2]$ such that for all $k \in [N]$, a constant $G < \infty$ exists such that $\mathbb{E}[\|X_k\|^h] \leq G^h$. Let $b_1, b_2, \cdots, b_N$ be fixed constants with $B = \max_k b_k$ such that $B \leq 1$. For any given clipping threshold $\lambda > 0$, the clipped vector is defined as $\widetilde{X}_k \coloneqq X_k \cdot (\frac{\lambda}{\|X_k\|} \wedge 1)$. Then with probability at least $1 - \omega$,

$$\begin{aligned}&\left\|\sum_{k=1}^{N} b_k(\widetilde{X}_k - \mu_k)\right\| \\ &\leq 4B\lambda \log\frac{3}{\omega} + \sum_{k=1}^{N}\frac{b_k G^h}{\lambda^{h-1}} + 2\sqrt{\sum_{k=1}^{N} b_k^2 G^h \lambda^{2-h}\log\frac{3}{\omega}}.\end{aligned}$$

To establish the convergence guarantee of our proposed algorithm, we analyze the error in estimating the true gradient $\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})$ using the aggregated momentum $\widehat{m}_t$, i.e., $\|\widehat{m}_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\|$. For ease of analysis, we define the aggregated honest gradients and momentum terms as:

$$\widetilde{g}_t = \frac{1}{n-f}\sum_{i\in\mathcal{H}}\widetilde{g}_t^{(i)}, \quad m_t \coloneqq \frac{1}{n-f}\sum_{i\in\mathcal{H}} m_t^{(i)}. \tag{7}$$

Thus, we decompose the error bound as follows:

$$\|\widehat{m}_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\| \leq \underbrace{\|\widehat{m}_t - m_t\|}_{\text{Aggregation Error}} + \underbrace{\|m_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\|}_{\text{Momentum Deviation}}.$$

To bound $\|\widehat{m}_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\|$, we separately analyze the *aggregation error* and *momentum deviation*. Notably, due to the (δ, κ)-robustness, the aggregation error reduces to analyzing the *momentum drift* defined as $\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|m_t^{(i)} - m_t\|^2$. The following Lemma 2, Lemma 3 and Lemma 4, provide upper bounds for *momentum drift*, *aggregation error*, and *momentum deviation*, respectively. For convenience, let $\alpha \coloneqq 1 - \beta$.

Lemma 2 (Momentum Drift)**.** With probability at least $1-\omega$, for each $t \in [T]$, we have

$$\begin{aligned}&\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|m_t^{(i)} - m_t\|^2 \leq 48\alpha^2\lambda^2(1 + \frac{1}{(n-f)^2})\log^2\frac{3T}{\omega} \\ &+ \frac{3G^{2h}}{\lambda^{2h-2}} + 12\frac{\alpha}{2-\alpha}(1 + \frac{1}{n-f})G^h\lambda^{2-h}\log\frac{3T}{\omega} + 3H^2.\end{aligned}$$

Proof of Lemma 2.

$$\begin{aligned}&\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|m_t^{(i)} - m_t\|^2 \\ &= (1-\beta)^2\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\left\|\sum_{k=1}^{t}\beta^{t-k}(\widetilde{g}_k^{(i)} - \widetilde{g}_k)\right\|^2 \\ &\leq 3(1-\beta)^2\Bigg(\underbrace{\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\widetilde{g}_k^{(i)} - \nabla\mathcal{L}_i(x_{k-1})\right)\right\|^2}_{A_t} \\ &+ \underbrace{\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\widetilde{g}_k - \nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\right)\right\|^2}_{B_t} \\ &+ \underbrace{\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\nabla\mathcal{L}_i(x_{k-1}) - \nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\right)\right\|^2}_{C_t}\Bigg).\end{aligned}$$

By Lemma 1, with probability at least $1 - \frac{\omega}{T}$, A_t, B_t and C_t can be bounded as follows,

$$\begin{aligned}A_t &\leq 3\left[\left(4\lambda\log\frac{3T}{\omega}\right)^2 + \left(\frac{G^h}{\alpha\lambda^{h-1}}\right)^2 + \left(2\sqrt{\frac{G^h\lambda^{2-h}}{1-\beta^2}\log\frac{3T}{\omega}}\right)^2\right] \\ &= 48\lambda^2\log^2\frac{3T}{\omega} + 3\frac{G^{2h}}{\alpha^2\lambda^{2h-2}} + 12\frac{G^h\lambda^{2-h}}{(1-\beta^2)}\log\frac{3T}{\omega}. \\ B_t &\leq 3\left(\frac{4\lambda}{n-f}\log\frac{3T}{\omega}\right)^2 + 3\left(\frac{G^h}{\alpha\lambda^{h-1}}\right)^2 \\ &\quad + 3\left(\sqrt{\frac{4G^h\lambda^{2-h}}{(1-\beta^2)(n-f)}\log\frac{3T}{\omega}}\right)^2 \\ &= \frac{48\lambda^2}{(n-f)^2}\log^2\frac{3T}{\omega} + \frac{3G^{2h}}{\alpha^2\lambda^{2h-2}} + \frac{12G^h\lambda^{2-h}}{(1-\beta^2)(n-f)}\log\frac{3T}{\omega}. \\ C_t &\leq \frac{\sum_{k=1}^{t}\beta^{t-k}}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\sum_{k=1}^{t}\left(\beta^{t-k}\|\nabla\mathcal{L}_i(x_{k-1}) - \nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\|^2\right) \\ &\leq \frac{H^2}{(1-\beta)^2}.\end{aligned}$$

□

Lemma 3 (Aggregation Error)**.** Let $\texttt{Agg}$ be (δ, κ)-robust. Define $\epsilon_t \coloneqq m_t - \widehat{m}_t$. With probability at least $1 - \omega$, for all $t \in [T]$,

$$\begin{aligned}&\|\epsilon_t\| \leq \\ &4\sqrt{6\kappa}\alpha\lambda\log\frac{3T}{\omega} + \frac{\sqrt{3\kappa}G^h}{\lambda^{h-1}} + \sqrt{\frac{24\kappa\alpha}{2-\alpha}\log\frac{3T}{\omega}}G^{\frac{h}{2}}\lambda^{1-\frac{h}{2}} + \sqrt{3\kappa}H.\end{aligned}$$

Proof of Lemma 3. The proof follows directly from Lemma 2 and the (δ, κ)-robustness of $\texttt{Agg}$. □

Lemma 4 (Momentum Deviation)**.** Define $\Delta_t \coloneqq m_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})$. With probability at least $1-\omega$, for all $t \in [T]$,

$$\|\Delta_t\| \leq \beta^t G + \frac{4\alpha\lambda}{n-f}\log\frac{3T}{\omega} + \frac{G^h}{\lambda^{h-1}} + \sqrt{\frac{4\alpha G^h \lambda^{2-h}\log\frac{3T}{\omega}}{n-f}} + \frac{1-\alpha}{\alpha}\eta L.$$

Proof of Lemma 4. First note that

$$\Delta_t = m_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1}) = \beta m_{t-1} + (1-\beta)\widetilde{g}_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1}).$$

Then by adding and subtracting $\beta\nabla\mathcal{L}_{\mathcal{H}}(x_{t-2})$ and $\beta\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})$, we obtain that, for $t \geq 2$,

$$\begin{aligned}\Delta_t &= \beta m_{t-1} - \beta\nabla\mathcal{L}_{\mathcal{H}}(x_{t-2}) + (1-\beta)\widetilde{g}_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1}) \\ &\quad + \beta\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1}) + \beta\nabla\mathcal{L}_{\mathcal{H}}(x_{t-2}) - \beta\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1}) \\ &= \beta\Delta_{t-1} + (1-\beta)(\widetilde{g}_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})) \\ &\quad + \beta(\nabla\mathcal{L}_{\mathcal{H}}(x_{t-2}) - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})).\end{aligned}$$

Defining $m_0 = 0$, $x_{-1} = x_0$ and $\Delta_0 = -\nabla\mathcal{L}_{\mathcal{H}}(x_0)$, then $\Delta_t = \beta\Delta_{t-1} + (1-\beta)(\widetilde{g}_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})) + \beta(\nabla\mathcal{L}_{\mathcal{H}}(x_{t-2}) - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1}))$ holds also for $t = 1$. Finally, unravel the recursion for all iterations, and we have

$$\begin{aligned}\|\Delta_t\| &\leq \beta^t\|\Delta_0\| + 4(1-\beta)\frac{\lambda}{n-f}\log\frac{3T}{\omega} + \frac{G^h}{\lambda^{h-1}} \\ &\quad + 2\sqrt{\frac{1-\beta}{1+\beta}}\frac{G^{\frac{h}{2}}\lambda^{1-\frac{h}{2}}}{\sqrt{n-f}}\sqrt{\log\frac{3T}{\omega}} + \frac{\beta}{1-\beta}\eta L \\ &\leq \beta^t G + 4(1-\beta)\frac{\lambda}{n-f}\log\frac{3T}{\omega} + \frac{G^h}{\lambda^{h-1}} \\ &\quad + 2\sqrt{\frac{1-\beta}{1+\beta}}\frac{G^{\frac{h}{2}}\lambda^{1-\frac{h}{2}}}{\sqrt{n-f}}\sqrt{\log\frac{3T}{\omega}} + \frac{\beta}{1-\beta}\eta L,\end{aligned}$$

where in the second inequality we used $\|\Delta_0\| = \|\nabla\mathcal{L}_{\mathcal{H}}(x_0)\| \leq \mathbb{E}_z[\|\nabla\ell(x_0,z)\|^h]^{\frac{1}{h}} \leq G$. □

A critical component of the convergence proof is the one-step descent bound. Unlike prior works [11], [10], [13], which provide *in-expectation* descent bounds, we establish a *surely* one-step descent bound that is specific to our normalized momentum descent update.

Lemma 5 (Descent Bound)**.** Under Assumption 1, for each update from x_{t-1} to x_t, the following descent bound holds:

$$\mathcal{L}_{\mathcal{H}}(x_t) - \mathcal{L}_{\mathcal{H}}(x_{t-1}) \leq -\eta\|\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\| + 2\eta\|\epsilon_t + \Delta_t\| + \frac{L}{2}\eta^2.$$

Proof of Lemma 5.

$$\begin{aligned}&\mathcal{L}_{\mathcal{H}}(x_t) - \mathcal{L}_{\mathcal{H}}(x_{t-1}) \\ \leq&\langle\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1}), x_t - x_{t-1}\rangle + \frac{L}{2}\|x_t - x_{t-1}\|^2 \\ =&-\eta\langle\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1}), \frac{\widehat{m}_t}{\|\widehat{m}_t\|}\rangle + \frac{L}{2}\eta^2 \\ =&-\eta\|\widehat{m}_t\| + \eta\langle\epsilon_t + \Delta_t, \frac{\widehat{m}_t}{\|\widehat{m}_t\|}\rangle + \frac{L}{2}\eta^2 \\ \leq&-\eta\|\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1}) + \Delta_t + \epsilon_t\| + \eta\|\epsilon_t + \Delta_t\| + \frac{L}{2}\eta^2 \\ \leq&-\eta\|\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\| + 2\eta\|\epsilon_t + \Delta_t\| + \frac{L}{2}\eta^2\end{aligned}$$

□

Theorem 1 (Convergence Guarantee for Algorithm 2)**.** Suppose Assumptions 1, 2, and 3 hold. Define $C \coloneqq \left(\frac{L(\mathcal{L}_{\mathcal{H}}(x_0)-\mathcal{L}_{\mathcal{H}}(x_T))}{G^2}\right)^{\frac{h}{3h-2}} \vee 1$ and parameter $\alpha \coloneqq C \cdot \left(\frac{1}{T^{\frac{h}{2}}\left(1+\sqrt{\kappa}\right)\left(\frac{1}{n-f}+\sqrt{\kappa}\right)^{h-1}\left(\log\frac{3T}{\omega}\right)^{h-1}}\right)^{\frac{2}{3h-2}}$. Set $\beta = 1-\alpha$, $\eta = \left(\frac{\mathcal{L}_{\mathcal{H}}(x_0)-\mathcal{L}_{\mathcal{H}}(x_T)}{L}\cdot\frac{\alpha}{T}\right)^{\frac{1}{2}}$, and $\lambda = \left(\frac{1+\sqrt{\kappa}}{\frac{1}{n(1-\delta)}+\sqrt{\kappa}}\right)^{\frac{1}{h}}\frac{G}{\alpha^{\frac{1}{h}}\left(\log\frac{3T}{\omega}\right)^{\frac{1}{h}}}$ in Algorithm 2. Then, with probability at least $1-\omega$,

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\|\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\| \\ &\leq\widetilde{O}\left(\left[\left(1+\sqrt{\kappa}\right)^{\frac{1}{h-1}}\left(\frac{1}{n}+\sqrt{\kappa}\right)\frac{1}{T}\right]^{\frac{h-1}{3h-2}} + \sqrt{\kappa}H\right).\end{aligned}$$

Proof of Theorem 1. From Lemma 5, for each iteration $t \in [T]$, we have that,

$$\|\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\| \leq \frac{\mathcal{L}_{\mathcal{H}}(x_{t-1})-\mathcal{L}_{\mathcal{H}}(x_t)}{\eta} + 2\|\epsilon_t\| + 2\|\Delta_t\| + \frac{L}{2}\eta.$$

Take average from $t=1$ to T, and we have

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\|\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\| \\ \leq&\frac{\mathcal{L}_{\mathcal{H}}(x_0)-\mathcal{L}_{\mathcal{H}}(x_T)}{\eta T} + \frac{2}{T}\sum_{t=1}^{T}\|\epsilon_t\| + \frac{2}{T}\sum_{t=1}^{T}\|\Delta_t\| + \frac{L\eta}{2} \\ \leq&O\Bigg[\underbrace{\frac{\mathcal{L}_{\mathcal{H}}(x_0)-\mathcal{L}_{\mathcal{H}}(x_T)}{\eta T}}_{\mathscr{A}} + \underbrace{\left(\sqrt{\kappa}+1\right)\frac{G^h}{\lambda^{h-1}}}_{\mathscr{C}} \\ &+ \underbrace{\left(\sqrt{\kappa}+\frac{1}{n-f}\right)\alpha\lambda\log\frac{3T}{\omega}}_{\mathscr{B}} \\ &+ \underbrace{\left(\sqrt{\kappa}+\frac{1}{\sqrt{n-f}}\right)\sqrt{\frac{\alpha}{2-\alpha}}G^{\frac{h}{2}}\lambda^{1-\frac{h}{2}}\sqrt{\log\frac{3T}{\omega}}}_{\mathscr{D}} \\ &+ \underbrace{\frac{1}{\alpha}\eta L}_{\mathscr{E}} + \frac{1}{T}\sum_{t=1}^{T}(1-\alpha)^t G + \sqrt{\kappa}H\Bigg] \\ \leq&O\left[\mathscr{A}+\mathscr{C}+\mathscr{B}+\mathscr{D}+\mathscr{E}+\frac{G}{\alpha T}+\sqrt{\kappa}H\right],\end{aligned}$$

where the last inequality is due to $\sum_{t=1}^{T}(1-\alpha)^t \leq \frac{1}{\alpha}$. Next, we minimize the upper bound by appropriately setting the parameters η, λ and α. Firstly, balancing the trade-off between terms $\mathscr{A}$ and $\mathscr{E}$ in η gives us $\eta = \left(\frac{\mathcal{L}_{\mathcal{H}}(x_0)-\mathcal{L}_{\mathcal{H}}(x_T)}{L}\cdot\frac{\alpha}{T}\right)^{\frac{1}{2}}$. Next we balance the two trade-offs in λ: the first between terms $\mathscr{C}$ and $\mathscr{B}$, and the second between terms $\mathscr{C}$ and $\mathscr{D}$, which gives us $\lambda_1 = G\left(\frac{1+\sqrt{\kappa}}{\frac{1}{n-f}+\sqrt{\kappa}}\right)^{\frac{1}{h}}\frac{1}{\alpha^{\frac{1}{h}}\left(\log\frac{3T}{\omega}\right)^{\frac{1}{h}}}$

and $\lambda_2 = G\left(\frac{1+\sqrt{\kappa}}{\frac{1}{\sqrt{n-f}}+\sqrt{\kappa}}\right)^{\frac{2}{h}} \frac{1}{\left(\frac{\alpha}{2-\alpha}\right)^{\frac{1}{h}}\left(\log\frac{3T}{\omega}\right)^{\frac{1}{h}}}$, respectively. Note that $\frac{\lambda_2}{\lambda_1} = \left(\frac{(1+\sqrt{\kappa})(\frac{1}{n-f}+\sqrt{\kappa})}{(\frac{1}{\sqrt{n-f}}+\sqrt{\kappa})^2}(2-\alpha)\right)^{\frac{1}{h}} = \left(\frac{\frac{1}{n-f}+\kappa+\sqrt{\kappa}+\frac{\sqrt{\kappa}}{n-f}}{\frac{1}{n-f}+\kappa+2\sqrt{\frac{\kappa}{n-f}}}(2-\alpha)\right)^{\frac{1}{h}} \geq 1$. Then by setting $\lambda = \lambda_1 \wedge \lambda_2 = \lambda_1$, we get the convergence upper bound of

$$\frac{1}{T}\sum_{t=1}^{T}\|\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\| \leq O\Bigg(\underbrace{\sqrt{\frac{(1-\alpha)L(\mathcal{L}_{\mathcal{H}}(x_0)-\mathcal{L}_{\mathcal{H}}(x_T))}{\alpha T}}}_{\mathscr{F}} + \underbrace{G\left(1+\sqrt{\kappa}\right)^{\frac{1}{h}}\left(\frac{1}{n-f}+\sqrt{\kappa}\right)^{\frac{h-1}{h}}\left(\alpha\log\frac{3T}{\omega}\right)^{\frac{h-1}{h}}}_{\mathscr{G}} + \underbrace{\frac{G}{\alpha T}}_{\mathscr{H}} + \sqrt{\kappa}H\Bigg). \tag{8}$$

Finally, we balance the two trade-offs in α: one between terms $\mathscr{G}$ and $\mathscr{F}$, the other between $\mathscr{G}$ and $\mathscr{H}$, which gives us

$$\alpha_1 = \left(\frac{(L(\mathcal{L}_{\mathcal{H}}(x_0)-\mathcal{L}_{\mathcal{H}}(x_T)))^h}{G^{2h}T^h(1+\sqrt{\kappa})^2\left(\frac{1}{n-f}+\sqrt{\kappa}\right)^{2h-2}\left(\log\frac{3T}{\omega}\right)^{2h-2}}\right)^{\frac{1}{3h-2}}$$

and

$$\alpha_2 = \left(\frac{1}{T^h\left(1+\sqrt{\kappa}\right)\left(\frac{1}{n-f}+\sqrt{\kappa}\right)^{h-1}\left(\log\frac{3T}{\omega}\right)^{h-1}}\right)^{\frac{1}{2h-1}},$$

respectively. Denote $C = \left(\frac{L(\mathcal{L}_{\mathcal{H}}(x_0)-\mathcal{L}_{\mathcal{H}}(x_T))}{G^2}\right)^{\frac{h}{3h-2}} \vee 1$ and $C' = (L(\mathcal{L}_{\mathcal{H}}(x_0)-\mathcal{L}_{\mathcal{H}}(x_T)))^{\frac{h-1}{3h-2}} G^{\frac{h}{3h-2}} \vee G$. Setting $\alpha = C\cdot\left(\frac{1}{T^{\frac{h}{2}}\left(1+\sqrt{\kappa}\right)\left(\frac{1}{n-f}+\sqrt{\kappa}\right)^{h-1}\left(\log\frac{3T}{\omega}\right)^{h-1}}\right)^{\frac{2}{3h-2}}$ gives us that:

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\|\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\|\\ \leq&\widetilde{O}\left(\left[\left(1+\sqrt{\kappa}\right)^{\frac{1}{h-1}}\left(\frac{1}{n}+\sqrt{\kappa}\right)\frac{1}{T}\right]^{\frac{h-1}{3h-2}}+\sqrt{\kappa}H\right)\end{aligned}$$

□

Remark 1 (First High-Probability Bounds for Byzantine FL)**.** To the best of our knowledge, Theorem 1 establishes the first high-probability convergence guarantee for Byzantine-resilient FL under heavy-tailed gradient noise. When $h = 2$, our result recovers the light-tailed setting with bounded gradient variance and matches the in-expectation convergence rate and error of [10]. Unlike in-expectation guarantees which can mask large deviations, high-probability guarantees ensure stable and reliable performance across individual training runs. This distinction is particularly important in federated training of large models, where each training run incurs substantial computation and communication costs.

Remark 2 (Optimal Convergence Error and non-Byzantine Convergence Rate)**.** Our convergence guarantee is tight with respect to the non-vanishing error induced by data heterogeneity and Byzantine robustness, and it recovers the optimal (n, T)-dependent convergence rate in the non-Byzantine regime, up to universal constant and logarithmic factors. In particular, the non-vanishing error term scales as $O(\sqrt{\kappa}\,H)$. Under the standard assumption that the Byzantine fraction δ is bounded away from $\frac{1}{2}$ by a constant margin ν, as stated in Section III-E (i.e., $\delta \leq \frac{1}{2+\nu}$), applying NNM with standard robust aggregation rules yields $\kappa = \Theta(\delta)$. In this regime, the error term matches the lower bound $\Omega(\sqrt{\delta}\,H)$ in Proposition 1. Moreover, when $\delta = 0$, the (n, T)-dependent convergence rate matches the general lower bound for non-convex federated optimization under heavy-tailed noise, $\Omega\Big((nT)^{\frac{1-h}{3h-2}}\Big)$, established in [13]. While the lower bound in [13] is stated in expectation, our result provides a high-probability upper bound. Finally, keeping κ explicit in the bound also clarifies the asymptotic behavior as $\delta \to \frac{1}{2}$. When the margin parameter ν is no longer treated as a constant, the robustness coefficient scales as $\kappa = O(1/\nu)$, which diverges as $\nu \to 0$. Consequently, the convergence bound becomes unbounded, reflecting the impossibility of Byzantine-resilient learning at or beyond the $\frac{1}{2}$ Byzantine threshold.

Remark 3 (Speedup and Client Synergy)**.** Our convergence rate quantifies how client collaboration accelerates learning in the presence of Byzantine failures through the factor $\frac{1}{n}+\sqrt{\delta}$. When $\delta = 0$, this reduces to $1/n$, yielding the standard linear speedup with the number of clients. As δ increases, the speedup degrades, consistent with prior results [11], [10]. Compared to the in-expectation dependence $\frac{1}{n}+\delta$ in prior work, our high-probability bound exhibits a $\sqrt{\delta}$ dependence. This discrepancy is inherent to high-probability analysis under heavy-tailed noise. In particular, obtaining uniform guarantees requires controlling the maximum magnitude of truncated stochastic increments, which introduces an additional $O(B\lambda\log(1/\omega))$ term in the concentration bound (Lemma 1). Such a term is unnecessary in expectation, where rare but large deviations are averaged out. For completeness, we additionally provide an in-expectation convergence analysis for heterogeneous Byzantine FL under heavy-tailed noise in the supplementary Appendix, where the dependence on δ recovers the form $\frac{1}{n}+\delta$.

VI. Further Discussions and Extensions

A. Gradient Clipping versus Momentum Clipping

In Algorithm 2, clipping is applied directly to each stochastic gradient before it is incorporated into the momentum update. An intuitive alternative is to apply clipping to the momentum updates themselves, rather than the individual gradients. This raises a key question: Does momentum clipping offer advantages over gradient clipping? To investigate this, we analyze the momentum clipping-based algorithm, Algorithm 3, and compare its convergence behavior with the gradient clipping-based algorithm. Algorithm 3 follows the same general procedure as Algorithm 2:

Algorithm 3: Robust D-SHB with Momentum Clipping and Normalization

Input: Initial model x_0, initial momentum $m_0^{(i)} = 0$ for honest clients, robust aggregation $\mathtt{Agg}$, learning rate η, momentum coefficient β, clipping threshold λ and number of steps T.
for $t = 1 \dots T$ **do**
  Server broadcasts x_{t-1} to all clients;
  for every ***honest client*** i, **in parallel do**
    Compute a stochastic gradient: $g_t^{(i)} = \nabla \ell(x_{t-1}, z_i^t)$;
    Update local momentum: $m_t^{(i)} = \beta m_{t-1}^{(i)} + (1-\beta) g_t^{(i)}$;
    Momentum clipping: $\widetilde{m}_t^{(i)} = \left(1 \wedge \frac{\lambda}{\|m_t^{(i)}\|}\right) \cdot m_t^{(i)}$;
    Send $\widetilde{m}_t^{(i)}$ to the server;
  Server aggregates the momenta: $\widehat{m}_t = \mathtt{Agg}(\widetilde{m}_t^{(1)}, \dots, \widetilde{m}_t^{(n)})$;
  Server updates the model: $x_t = x_{t-1} - \eta \cdot \frac{\widehat{m}_t}{\|\widehat{m}_t\|}$;
return $x_0, \dots, x_T$;

1) The server broadcasts the latest model to all clients.
2) Each client computes a local stochastic gradient.
3) Unlike Algorithm 2, where gradients are clipped before updating the momentum, here, clients first update their local momentum using the raw gradients and then the momentum is clipped before being sent to the server.
4) The server aggregates the clipped local momenta and updates the model accordingly.

The key difference between the gradient clipping approach (Algorithm 2) and the momentum clipping approach (Algorithm 3) lies in the stage when clipping is applied. This distinction is highlighted in gray boxes within each algorithm.

We now establish the convergence guarantee for Algorithm 3. As in the case of gradient clipping, the key is to analyze the error of the aggregated clipped momenta $\widehat{m}_t$ in estimating the true gradient $\nabla \mathcal{L}_{\mathcal{H}}(x_{t-1})$. However, due to the different clipping strategy, the previous error decomposition no longer applies. To facilitate the analysis, we introduce the following notation:

$$\widetilde{m}_t \coloneqq \frac{1}{n-f} \sum_{i \in \mathcal{H}} \widetilde{m}_t^{(i)},$$

which represents the average clipped momentum over the honest clients. Using this, the error decomposition is:

$$\begin{aligned} &\|\widehat{m}_t - \nabla \mathcal{L}_{\mathcal{H}}(x_{t-1})\| \\ &\leq \underbrace{\|\widehat{m}_t - \widetilde{m}_t\|}_{\text{Aggregation Error}} + \underbrace{\|\widetilde{m}_t - \mathbb{E}[m_t]\|}_{\text{Clipping Bias}} + \underbrace{\|\mathbb{E}[m_t] - \nabla \mathcal{L}_{\mathcal{H}}(x_{t-1})\|}_{\text{Momentum Deviation}}. \end{aligned}$$

This decomposition differs from that in the gradient clipping case, where the h-th moment of the gradients is explicitly assumed and allows for the direct application of Lemma 1. In contrast, under momentum clipping, a tight bound on the h-th moment of the momenta is not directly implied by Assumption 3 due to temporal accumulation, and thus requires additional analysis. To control the aggregation error in this setting, we first establish a sharp upper bound on the h-th moment of the local momentum $m_t^{(i)}$ in Lemma 6.

Lemma 6. For $\forall i \in \mathcal{H}, t \in [T]$, $\mathbb{E}[\|m_t^{(i)}\|^h] \leq \alpha^{h-1} G^h$.

Proof. Using the definition of momentum updates $m_t^{(i)} = \alpha \sum_{k=1}^t \beta^{t-k} g_k^{(i)}$, we compute its h-th moment: $\mathbb{E}[\|m_t^{(i)}\|^h] = \alpha^h \mathbb{E}[\|\sum_{k=1}^t \beta^{t-k} g_k^{(i)}\|^h]$. Applying Minkowski's inequality and Jensen's inequality, we obtain: $\mathbb{E}[\|m_t^{(i)}\|^h] \leq \alpha^h \mathbb{E}[\sum_{k=1}^t \|\beta^{t-k} g_k^{(i)}\|^h] \leq \alpha^h \mathbb{E}[\sum_{k=1}^t \beta^{h(t-k)} \|g_k^{(i)}\|^h]$. Since $\mathbb{E}[\|g_k^{(i)}\|^h] \leq G^h$, it follows that $\mathbb{E}[\|m_t^{(i)}\|^h] \leq G^h \alpha^h \sum_{k=1}^t \beta^{h(t-k)}$. Using the geometric series sum bound, $\sum_{k=1}^t \beta^{h(t-k)} \leq \frac{(1-\beta^{th})}{1-\beta^h} \leq \frac{1}{\alpha}$, we obtain the final bound $\mathbb{E}[\|m_t^{(i)}\|^h] \leq \alpha^{h-1} G^h$. □

Remark 4 (Tail Index of Momenta)**.** Lemma 6 implies that the h-th moment of the momentum depends on the h-th moment bound of the gradients G^h and the momentum coefficient β, with the bound decreasing monotonically as β increases. In particular, as $\beta \to 0$, the momentum reduces to the current gradient and its h-th moment approaches G^h. In contrast, as $\beta \to 1$, the momentum increasingly remains close to its initial value due to strong accumulation, and consequently its h-th moment tends to zero.

We now provide upper bounds for each term in our error decomposition.

Lemma 7 (Aggregation Error)**.** Define $\epsilon_t \coloneqq \widehat{m}_t - \widetilde{m}_t$. With probability at least $1 - \omega$, for $\forall t \in [T]$,

$$\|\epsilon_t\| \leq O\left(\sqrt{\kappa}\left(\lambda \log\frac{T}{\omega} + \frac{\alpha^{h-1} G^h}{\lambda^{h-1}} + \sqrt{\alpha^{h-1} G^h \lambda^{2-h} \log\frac{T}{\omega}} + H\right)\right)$$

Proof. Firstly, we derive the upper bound for $\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \left\|\widetilde{m}_t^{(i)} - \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \widetilde{m}_t^{(i)}\right\|^2$.

$$\begin{aligned} &\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \left\|\widetilde{m}_t^{(i)} - \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \widetilde{m}_t^{(i)}\right\|^2 \\ \leq &\frac{3}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \underbrace{\left\|\widetilde{m}_t^{(i)} - \alpha\sum_{k=1}^t \beta^{t-k} \nabla \mathcal{L}_i(x_{k-1})\right\|^2}_{A_t} \\ &+ \frac{3}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \underbrace{\left\|\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \widetilde{m}_t^{(i)} - \alpha\sum_{k=1}^t \beta^{t-k} \nabla \mathcal{L}_{\mathcal{H}}(x_{k-1})\right\|^2}_{B_t} \\ &+ \underbrace{\frac{3}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \left\|\alpha\sum_{k=1}^t \beta^{t-k}\left(\nabla \mathcal{L}_i(x_{k-1}) - \nabla \mathcal{L}_{\mathcal{H}}(x_{k-1})\right)\right\|^2}_{C_t}. \end{aligned}$$

By using the concentration result established in Lemma 1, with probability at least $1 - \frac{\omega}{T}$, we have the following bounds for each A_t and B_t simultaneously.

$$A_t = \left\| \widetilde{m}_t^{(i)} - \alpha \sum_{k=1}^{t} \beta^{t-k} \nabla \mathcal{L}_i(x_{k-1}) \right\|^2$$
$$\leq \left(4\lambda \log \frac{3T}{\omega} + \frac{\alpha^{h-1} G^h}{\lambda^{h-1}} + 2\alpha^{\frac{h-1}{2}} G^{\frac{h}{2}} \lambda^{1-\frac{h}{2}} \log \frac{3T}{\omega} \right)^2$$
$$\leq O\left(\lambda^2 \log^2 \frac{T}{\omega} + \frac{\alpha^{2(h-1)} G^{2h}}{\lambda^{2(h-1)}} + \alpha^{h-1} G^h \lambda^{2-h} \log \frac{T}{\omega} \right).$$

$$B_t = \left\| \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \widetilde{m}_t^{(i)} - \alpha \sum_{k=1}^{t} \beta^{t-k} \nabla \mathcal{L}_{\mathcal{H}}(x_{k-1}) \right\|^2$$
$$\leq O\left(\frac{\lambda^2}{(n-f)^2} \log^2 \frac{T}{\omega} + \frac{\alpha^{2(h-1)} G^{2h}}{\lambda^{2(h-1)}} + \frac{\alpha^{h-1} G^h \lambda^{2-h}}{n-f} \log \frac{T}{\omega} \right)$$

By the heterogeneity of local objective functions, we have the follow bound for C_t.

$$C_t = \frac{3}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \left\| \alpha \sum_{k=1}^{t} \beta^{t-k} \left(\nabla \mathcal{L}_i(x_{k-1}) - \nabla \mathcal{L}_{\mathcal{H}}(x_{k-1}) \right) \right\|^2$$
$$\leq \frac{3\alpha^2}{n-f} \sum_{i \in \mathcal{H}} \left(\sum_{k=1}^{t} \beta^{t-k} \sum_{k=1}^{t} \beta^{t-k} \| \nabla \mathcal{L}_i(x_{k-1}) - \nabla \mathcal{L}_{\mathcal{H}}(x_{k-1}) \|^2 \right)$$
$$\leq 3\alpha \left(\sum_{k=1}^{t} \beta^{t-k} \frac{1}{n-f} \sum_{i \in \mathcal{H}} \| \nabla \mathcal{L}_i(x_{k-1}) - \nabla \mathcal{L}_{\mathcal{H}}(x_{k-1}) \|^2 \right)$$
$$\leq 3\alpha \left(\sum_{k=1}^{t} \beta^{t-k} \cdot H^2 \right) \leq 3H^2.$$

Combining the upper bounds on A_t, B_t and C_t, we have

$$\frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \left\| \widetilde{m}_t^{(i)} - \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \widetilde{m}_t^{(i)} \right\|^2 \leq$$
$$O\left(\lambda^2 \log^2 \frac{T}{\omega} + \frac{\alpha^{2(h-1)} G^{2h}}{\lambda^{2(h-1)}} + \alpha^{h-1} G^h \lambda^{2-h} \log \frac{T}{\omega} + H^2 \right).$$

Finally, by the definition of (δ, κ)-Robust aggregation, we can get the desired upper bound of $\|\epsilon_t\|$. □

Lemma 8 (Clipping Bias)**.** Define $\gamma_t \coloneqq \widetilde{m}_t - \mathbb{E}[m_t]$. With probability at least $1 - \omega$, for $\forall t \in [T]$,

$$\|\gamma_t\| \leq \frac{4\lambda}{n-f} \log \frac{3T}{\omega} + \frac{\alpha^{h-1} G^h}{\lambda^{h-1}} + \frac{2\alpha^{\frac{h-1}{2}} G^{\frac{h}{2}} \lambda^{1-\frac{h}{2}}}{\sqrt{n-f}} \sqrt{\log \frac{3T}{\omega}}.$$

Proof. The proof then follows directly from the fact that each $m_t^{(i)}$ is heavy-tailed with $\mathbb{E}\left[\left\| m_t^{(i)} \right\|^h \right] \leq (1-\beta)^{h-1} G^h$ shown in Lemma 6 and the concentration bound for clipped heavy-tailed random vector provided in Lemma 1. □

Lemma 9 (Momentum Deviation)**.** Define $\Delta_t \coloneqq \mathbb{E}[m_t] - \nabla \mathcal{L}_{\mathcal{H}}(x_{t-1})$. With probability at least $1 - \omega$, for $\forall t \in [T]$,

$$\|\Delta_t\| \leq (1-\alpha)^t G + \frac{1-\alpha}{\alpha} \eta L$$

Proof. Note that

$$\Delta_t = \mathbb{E}[m_t] - \nabla \mathcal{L}_{\mathcal{H}}(x_{t-1})$$
$$= \beta \mathbb{E}[m_{t-1}] + (1-\beta) \nabla \mathcal{L}_{\mathcal{H}}(x_{t-1}) - \nabla \mathcal{L}_{\mathcal{H}}(x_{t-1})$$
$$= \beta \mathbb{E}[m_{t-1}] - \beta \nabla \mathcal{L}_{\mathcal{H}}(x_{t-1}).$$

By adding and subtracting $\beta \nabla \mathcal{L}_{\mathcal{H}}(x_{t-2})$, for $t \geq 2$, we have

$$\Delta_t = \beta(\mathbb{E}[m_{t-1}] - \nabla \mathcal{L}_{\mathcal{H}}(x_{t-2})) + \beta(\nabla \mathcal{L}_{\mathcal{H}}(x_{t-2}) - \nabla \mathcal{L}_{\mathcal{H}}(x_{t-1}))$$
$$= \beta \Delta_{t-1} + \beta(\nabla \mathcal{L}_{\mathcal{H}}(x_{t-2}) - \nabla \mathcal{L}_{\mathcal{H}}(x_{t-1})).$$

For $t = 1$, let $m_0 = 0$ and $x_{-1} = x_0$, then $\Delta_0 = -\nabla \mathcal{L}_{\mathcal{H}}(x_0)$ and the recursion of $\Delta_t = \beta \Delta_{t-1} + \beta(\nabla \mathcal{L}_{\mathcal{H}}(x_{t-2}) - \nabla \mathcal{L}_{\mathcal{H}}(x_{t-1}))$ also holds for all $t \geq 1$. Thus, we have

$$\|\Delta_t\| \leq \beta^t \|\Delta_0\| + \beta \left(\sum_{k=1}^{t} \beta^{t-k} \right) \eta L \leq \beta^t G + \frac{\beta}{1-\beta} \eta L,$$

where in the last inequality we used $\|\Delta_0\| = \|\nabla \mathcal{L}_{\mathcal{H}}(x_0)\| \leq \mathbb{E}_z[\|\nabla \ell(x_0, z)\|^h]^{\frac{1}{h}} \leq G$. □

Theorem 2 (Convergence Guarantee for Algorithm 3)**.** Suppose Assumptions 1, 2, and 3 hold. Define $C \coloneqq \left(\frac{L(\mathcal{L}_{\mathcal{H}}(x_0) - \mathcal{L}_{\mathcal{H}}(x_T))}{G^2} \right)^{\frac{h}{3h-2}} \vee 1$ and parameter $\alpha \coloneqq C \cdot \left(\frac{1}{T^{\frac{h}{2}} \left(1+\sqrt{\kappa}\right) \left(\frac{1}{n-f} + \sqrt{\kappa}\right)^{h-1} \left(\log \frac{3T}{\omega}\right)^{h-1}} \right)^{\frac{2}{3h-2}}$. Set $\beta = 1 - \alpha$, $\eta = \left(\frac{\mathcal{L}_{\mathcal{H}}(x_0) - \mathcal{L}_{\mathcal{H}}(x_T)}{L} \cdot \frac{\alpha}{T} \right)^{\frac{1}{2}}$, and $\lambda = \left(\frac{1+\sqrt{\kappa}}{\frac{1}{n-f} + \sqrt{\kappa}} \right)^{\frac{1}{h}} \frac{\alpha^{\frac{h-1}{h}} G}{\left(\log \frac{3T}{\omega}\right)^{\frac{1}{h}}}$ in Algorithm 3. Then, with probability at least $1 - \omega$,

$$\frac{1}{T} \sum_{t=1}^{T} \| \nabla \mathcal{L}_{\mathcal{H}}(x_{t-1}) \|$$
$$\leq \widetilde{O}\left(\left[\left(1 + \sqrt{\kappa}\right)^{\frac{1}{h-1}} \left(\frac{1}{n} + \sqrt{\kappa} \right) \frac{1}{T} \right]^{\frac{h-1}{3h-2}} + \sqrt{\kappa} H \right).$$

Proof. Based on the descent bound shown in Lemma 5,

$$\frac{1}{T} \sum_{t=1}^{T} \| \nabla \mathcal{L}_{\mathcal{H}}(x_{t-1}) \|$$
$$\leq \frac{\mathcal{L}_{\mathcal{H}}(x_0) - \mathcal{L}_{\mathcal{H}}(w_T)}{\eta T} + \frac{2}{T} \sum_{t=1}^{T} (\|\epsilon_t\| + \|\gamma_t\| + \|\Delta_t\|) + \frac{L\eta}{2}$$
$$\leq O\Bigg[\frac{\mathcal{L}_{\mathcal{H}}(x_0) - \mathcal{L}_{\mathcal{H}}(x_T)}{\eta T} + \left(\frac{1}{n-f} + \sqrt{\kappa} \right) \lambda \log \frac{3T}{\omega}$$
$$+ \left(\frac{1}{\sqrt{n-f}} + \sqrt{\kappa} \right) \alpha^{\frac{h-1}{2}} G^{\frac{h}{2}} \lambda^{1-\frac{h}{2}} \sqrt{\log \frac{3T}{\omega}}$$
$$+ \left(1 + \sqrt{\kappa}\right) \frac{\alpha^{h-1} G^h}{\lambda^{h-1}} + \frac{\eta L}{\alpha} + \frac{G}{\alpha T} + \sqrt{\kappa} H \Bigg].$$

Balancing the trade-off in η gives the set of $\eta = \left(\frac{\mathcal{L}_{\mathcal{H}}(x_0)-\mathcal{L}_{\mathcal{H}}(x_T)}{L}\cdot\frac{\alpha}{T}\right)^{\frac{1}{2}}$. Then we have

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\|\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\| \\
&\leq O\Bigg[\sqrt{\frac{L(\mathcal{L}_{\mathcal{H}}(x_0)-\mathcal{L}_{\mathcal{H}}(x_T))}{\alpha T}}+\frac{G}{\alpha T}+\sqrt{\kappa}H \\
&+\underbrace{\left(\frac{1}{n-f}+\sqrt{\kappa}\right)\lambda\log\frac{3T}{\omega}}_{\mathscr{I}}+\underbrace{\left(1+\sqrt{\kappa}\right)\frac{\alpha^{h-1}G^h}{\lambda^{h-1}}}_{\mathscr{J}} \\
&+\underbrace{\left(\frac{1}{\sqrt{n-f}}+\sqrt{\kappa}\right)\alpha^{\frac{h-1}{2}}G^{\frac{h}{2}}\lambda^{1-\frac{h}{2}}\sqrt{\log\frac{3T}{\omega}}}_{\mathscr{K}}\Bigg].
\end{aligned}$$

Next, we balance the two trade-offs in λ: one between terms $\mathscr{I}$ and $\mathscr{J}$, the other between $\mathscr{J}$ and $\mathscr{K}$, which leads to $\lambda_1 = \left(\frac{1+\sqrt{\kappa}}{\frac{1}{n-f}+\sqrt{\kappa}}\right)^{\frac{1}{h}}\frac{\alpha^{\frac{h-1}{h}}G}{\left(\log\frac{3T}{\omega}\right)^{\frac{1}{h}}}$, and $\lambda_2 = \left(\frac{1+\sqrt{\kappa}}{\frac{1}{\sqrt{n-f}}+\sqrt{\kappa}}\right)^{\frac{2}{h}}\frac{\alpha^{\frac{h-1}{h}}G}{\left(\log\frac{3T}{\omega}\right)^{\frac{1}{h}}}$, respectively. Note that,

$$\frac{\lambda_1}{\lambda_2}=\left(\frac{1}{1+\sqrt{\kappa}}\cdot\frac{\left(\frac{1}{\sqrt{n-f}}+\sqrt{\kappa}\right)^2}{\frac{1}{n-f}+\sqrt{\kappa}}\right)^{\frac{1}{h}}\leq 1.$$

Hence, setting $\lambda=\lambda_1\wedge\lambda_2=\lambda_1$ gives us:

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\|\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\|\leq O\Bigg[\sqrt{\frac{L(\mathcal{L}_{\mathcal{H}}(x_0)-\mathcal{L}_{\mathcal{H}}(x_T))}{\alpha T}} \\
&+G\left(1+\sqrt{\kappa}\right)^{\frac{1}{h}}\left(\frac{1}{n-f}+\sqrt{\kappa}\right)^{\frac{h-1}{h}}\alpha^{\frac{h-1}{h}}\left(\log\frac{3T}{\omega}\right)^{\frac{h-1}{h}} \\
&+\frac{G}{\alpha T}+\sqrt{\kappa}H\Bigg].
\end{aligned}$$

We note that, this upper bound is the same as what we obtained in (8) for the gradient clipping case. Then following the same argument regarding the set of α there concludes the proof. □

Remark 5 (Equivalence of Convergence Guarantees)**.** For the first time, we establish through Theorem 2 that momentum clipping yields a convergence rate equivalent to that of gradient clipping. This finding substantiates the efficacy of both methods in ensuring convergence under the stated conditions, though each operates via a distinct mechanism. Specifically, momentum clipping operates directly on the accumulated gradients encapsulated within the momentum terms, simplifying the clipping process, while gradient clipping directly mitigates the effect of extreme gradient values at the source, preventing their accumulation. The demonstrated equivalence in convergence rates highlights the robustness of clipping-based strategies while offering flexibility in algorithm design based on practical considerations such as computational efficiency and implementation simplicity.

Algorithm 4: ANNM−RP: Approximate Nearest Neighbor Mixing via Random Projection

Input: number of inputs n, vectors $x_1,\ldots,x_n\in\mathbb{R}^d$, fraction of Byzantine inputs $\delta<1/2$.
Generate the random projection matrix $\Phi\in\mathbb{R}^{p\times d}$ (e.g., Gaussian or sparse JL matrix) ;
Project every $x_i\in\mathbb{R}^d$ to $\widehat{x}_i\in\mathbb{R}^p$ via $\widehat{x}_i\leftarrow\Phi x_i$
for $i=1\ldots n$ **do**
  Sort $(\widehat{x}_1,\cdots,\widehat{x}_n)$ to get $(\widehat{x}_{i_1},\ldots,\widehat{x}_{i_n})$ such that
  $$\|\widehat{x}_{i_1}-\widehat{x}_i\|\leq\ldots\leq\|\widehat{x}_{i_n}-\widehat{x}_i\|\ ;$$
  Let $\mathcal{N}_i$ be the set of indices of the $(1-\delta)n$ nearest neighbors of $\widehat{x}_i$, i.e.,
  $$\mathcal{N}_i\leftarrow\{i_1,\cdots,i_{(1-\delta)n}\}\ ;$$
  Average the approximate $(1-\delta)n$ nearest neighbors of x_i via $\mathcal{N}_i$, i.e.,
  $$y_i=\frac{1}{(1-\delta)n}\sum_{j\in\mathcal{N}_i}x_j\ ;$$
return $y_1,\ldots,y_n$;

B. Towards Better Efficiency for NNM in High-Dimensions

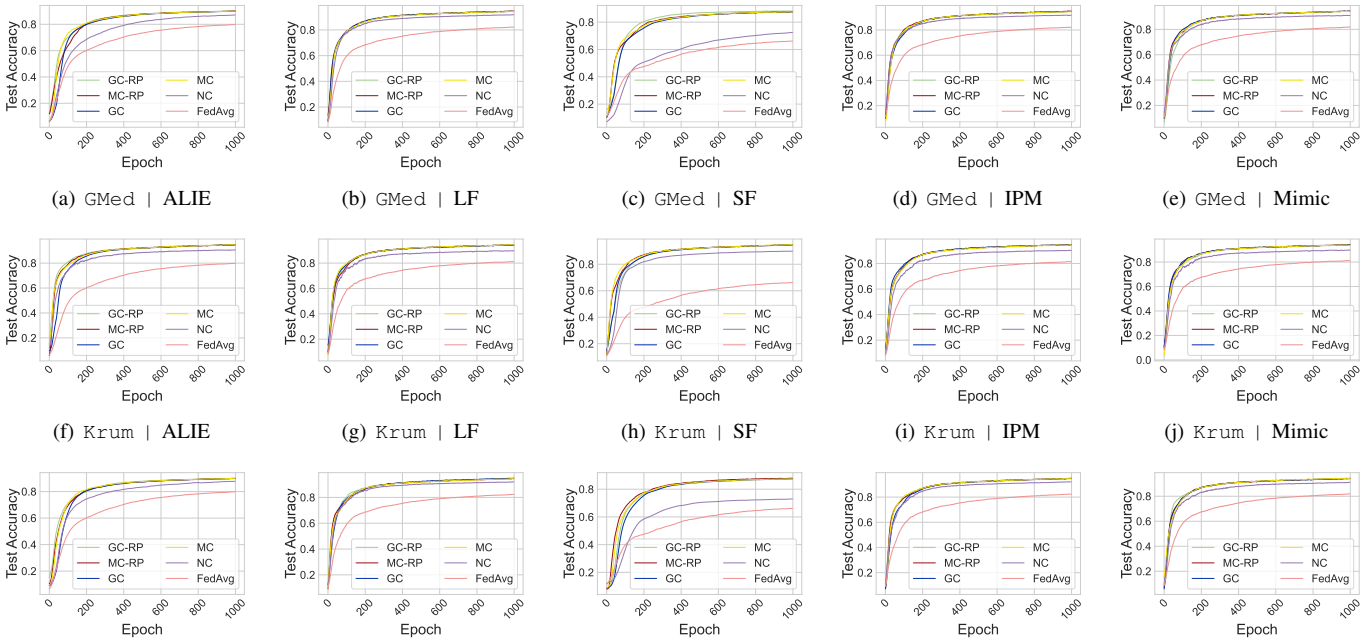
The original NNM algorithm (Algorithm 1) has a worst-case complexity of $O(dn^2)$ at the server, primarily due to the need to identify the $n-f$ nearest neighbors for each input vector. However, in many modern federated learning (FL) applications, the dimension $d\gg n$ (e.g., deep learning models with millions of parameters but relatively few clients in cross-silo FL). This makes a direct nearest neighbor search expensive and leads to significant latency.

To address this high-dimensional bottleneck, we introduce an optimized variant: ANNM−RP (Approximate Nearest Neighbor Mixing via Random Projection), detailed in Algorithm 4. Our approach leverages the Johnson–Lindenstrauss (JL) Lemma [39] to reduce the dimensionality of input vectors from $\mathbb{R}^d$ to $\mathbb{R}^p$ with $p=O(n\log n)$. This random projection can be done locally at each client, with system-wide common randomness. Clients then transmit both original and projected vectors to the server. This additional step does not change the asymptotic computational complexity at the client side. Then at the server side, by performing nearest neighbor selection in this lower-dimensional space, we efficiently approximate the $(1-\delta)n$ nearest neighbors $\mathcal{N}_i$ for each vector x_i. This dimensionality reduction technique significantly enhances computational efficiency, reducing the overall complexity to $O(n^3\log n)$. Importantly, this complexity is independent of the original dimension d, making it highly scalable for high-dimensional FL settings.

VII. Experiments

A. Experimental Setup

1) Computing Infrastructure: All experiments were conducted on Ubuntu 22.04.4 LTS with an AMD EPYC 7513 32-Core Processor, 503GB CPU memory, and an NVIDIA RTX



(a) GMed | ALIE (b) GMed | LF (c) GMed | SF (d) GMed | IPM (e) GMed | Mimic

(f) Krum | ALIE (g) Krum | LF (h) Krum | SF (i) Krum | IPM (j) Krum | Mimic

(k) CWTM | ALIE (l) CWTM | LF (m) CWTM | SF (n) CWTM | IPM (o) CWTM | Mimic

Fig. 2. Test accuracy comparison of different methods on the **MNIST** dataset. Each row represents a robust aggregation technique applied after pre-aggregation processing (**Row 1:** GMed, **Row 2:** Krum, and **Row 3:** CWTM). Each column corresponds to a specific Byzantine attack (**Column 1:** ALIE, **Column 2:** LF, **Column 3:** SF, **Column 4:** IPM, **Column 5:** Mimic).

A6000 GPU (48GB memory). The implementation was done in Python 3.11.8 using PyTorch 2.2.2 with CUDA 12.1.

2) Tasks, Datasets & Models: We conduct image classification tasks on two benchmark datasets: MNIST [40] and CIFAR-10 [41]. To introduce data heterogeneity, the datasets are partitioned among honest clients by sampling class proportion vectors from a Dirichlet distribution $\mathrm{Dir}(\iota)$ with $\iota = 0.1$, which corresponds to an extreme non-i.i.d. scenario. Unless otherwise specified, we use a simple fully connected neural network with a single hidden layer of 256 units and ReLU activation. The loss function is the standard cross-entropy loss. The model is initialized using Kaiming initialization [42], with biases set to zero by default.

3) Learning System & Byzantine Attacks: The experiments simulate a FL system $n = 10$, of which 20% are Byzantine clients. We evaluate Byzantine resilience of FL algorithms under five state-of-the-art Byzantine attacks: 1) A Little Is Enough (ALIE) [16]; 2) Label Flipping (LF) [17]; 3) Sign Flipping (SF) [17]; 4) Inner Product Manipulation (IPM) [15]; 5) Mimic attack [10].

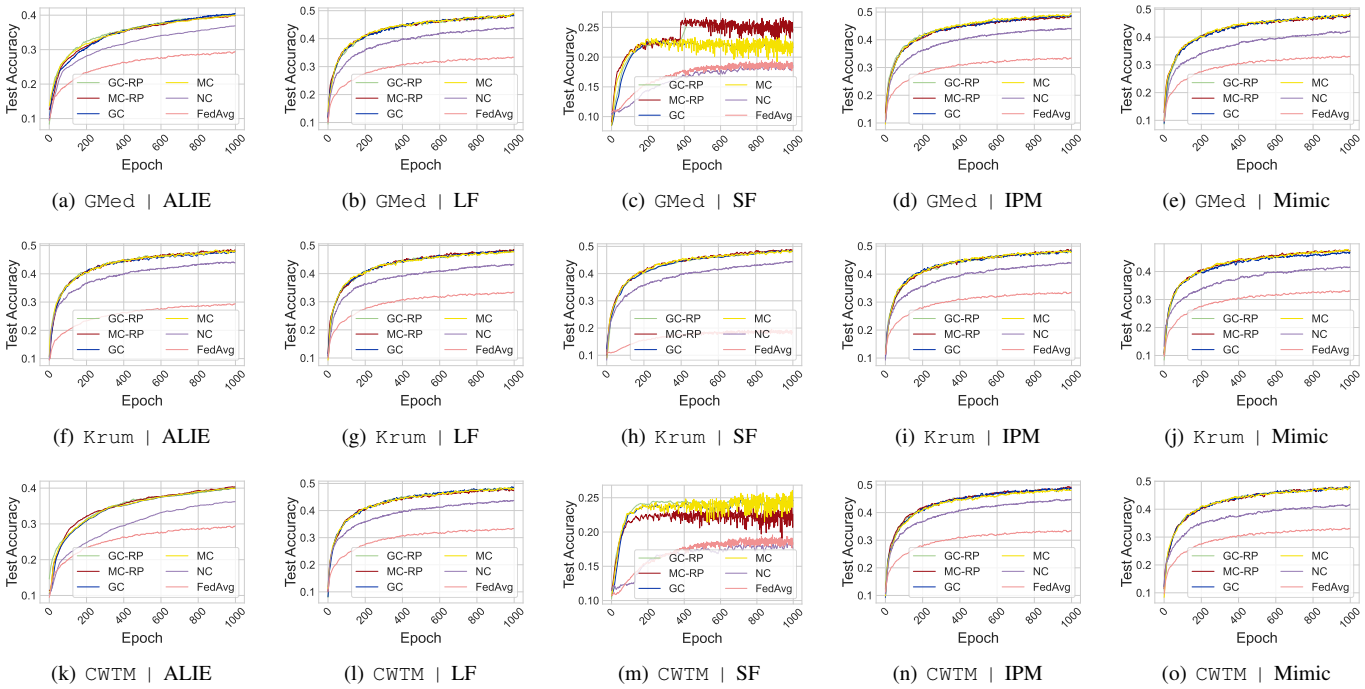
4) Algorithms & Baseline: We test four algorithm variants of our proposed framework, differing in clipping strategy (gradient vs. momentum) and pre-aggregation method (original NNM vs. optimized ANNM−RP). These variants include:

- GC (Gradient Clipping + Original NNM)
- MC (Momentum Clipping + Original NNM)
- GC−RP (Gradient Clipping + ANNM−RP)
- MC−RP (Momentum Clipping + ANNM−RP)

For comparison, we include the Byzantine-resilient baseline from [11], denoted as NC (No Clipping + original NNM). In addition, we incorporate FedAvg as a standard non-robust federated learning baseline, which performs simple averaging without any defense against Byzantine behavior or heavy-tailed gradient noise.

B. Results and Discussion

1) Effectiveness of clipping-based methods: The results in Fig. 2, Fig. 3, and Table III demonstrate that both gradient clipping (GC) and momentum clipping (MC) significantly improve Byzantine resilience compared to the Byzantine robust baseline with no clipping (NC) and the standard non-robust FedAvg. The improvements are evident across different datasets, robust aggregation strategies, and Byzantine attack scenarios. As shown in Fig. 2 and Fig. 3, FedAvg achieves the lowest test accuracy in nearly all Byzantine settings, reflecting its vulnerability in adversarial environments. Introducing Byzantine-resilient aggregation without clipping (NC) improves performance over FedAvg, but remains substantially inferior to our clipping-based methods. This comparison confirms that clipping does not artificially suppress performance; rather, it enables more stable and accurate training under heavy-tailed gradient noise and Byzantine attacks. On MNIST (Fig. 2), the performance gains from clipping are relatively modest. This is expected, as MNIST is a low-dimensional and well-conditioned dataset, where gradients are less prone to extreme outliers even under Byzantine behavior. In contrast, on CIFAR-10 (Fig. 3), which exhibits more pronounced heavy-tailed gradient noise due to the more complex data distributions and higher-dimension, the advantage of clipping is more pronounced. GC(−RP) and MC(−RP) show a clear advantage over NC and FedAvg, particularly under the SF attack. This suggests that clipping is crucial for mitigating the

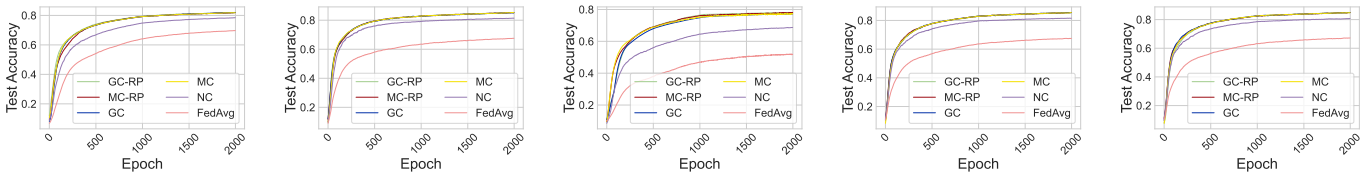


(a) GMed | ALIE (b) GMed | LF (c) GMed | SF (d) GMed | IPM (e) GMed | Mimic

(f) Krum | ALIE (g) Krum | LF (h) Krum | SF (i) Krum | IPM (j) Krum | Mimic

(k) CWTM | ALIE (l) CWTM | LF (m) CWTM | SF (n) CWTM | IPM (o) CWTM | Mimic

Fig. 3. Test accuracy comparison of different methods on the **CIFAR-10** dataset. Each row represents a robust aggregation technique applied after pre-aggregation processing (**Row 1:** GMed, **Row 2:** Krum, and **Row 3:** CWTM). Each column corresponds to a specific Byzantine attack (**Column 1:** ALIE, **Column 2:** LF, **Column 3:** SF, **Column 4:** IPM, **Column 5:** Mimic).



(a) GMed | ALIE (b) GMed | LF (c) GMed | SF (d) GMed | IPM (e) GMed | Mimic

Fig. 4. Test accuracy on CIFAR-10 using ResNet-18 with geometric median aggregation applied after pre-aggregation processing. Each column corresponds to a specific Byzantine attack (**Column 1:** ALIE, **Column 2:** LF, **Column 3:** SF, **Column 4:** IPM, **Column 5:** Mimic).

impact of outlier gradients in heavy-tailed settings. Another key observation is that momentum clipping exhibits larger fluctuations than gradient clipping in some cases, particularly under the SF attack on CIFAR-10 (Fig. 3(c) & 3(m)). This indicates that while MC smooths updates over time, it may also amplify certain adversarial effects when heavy-tailed noise is present. Finally, to assess whether the relatively low absolute CIFAR-10 accuracy is an artifact of conservative updates or architectural limitations, we further evaluate all methods using a more expressive convolutional model, ResNet-18. The results are shown in Fig. 4. As expected, the absolute accuracy of all methods improves substantially in this setting. Importantly, our clipping-based methods continue to consistently outperform baseline approaches across different Byzantine attacks, demonstrating that the proposed techniques remain effective in higher-accuracy regimes and do not intrinsically limit achievable performance.

2) Impact of different robust aggregation methods: Since all tested methods incorporate NNM, we compare the performance of different robust aggregation rules (GMed, Krum, and CWTM) under the same pre-aggregation setting. The results in Table III show that all three methods effectively mitigate Byzantine attacks when combined with clipping. However, the best-performing aggregation strategy depends on the attack type: CWTM-based approaches achieving the highest accuracy under Mimic and IPM attacks, while Krum-based methods perform best under SF and LF attacks. Within each robust aggregation category, GC(−RP) and MC(−RP) consistently outperform NC, underscoring the benefit of clipping for mitigating heavy-tailed noise. However, the gap between GC(−RP) and MC(−RP) varies across different aggregation strategies and attack types, indicating that the optimal choice of aggregation-clipping combination is attack-dependent.

3) Performance under different Byzantine attacks: The results in Fig. 3 and Table III confirm that different Byzantine attacks affect model performance in distinct ways. The SF attack is particularly detrimental on CIFAR-10, causing major fluctuations in test accuracy (Fig. 3(c) & 3(m)). Meanwhile, Mimic and IPM attacks lead to steady but significant accuracy degradation (Fig. 3, last two columns), demonstrating the importance of robust aggregation. Notably, the best-performing combinations of aggregation and clipping achieve nearly up to

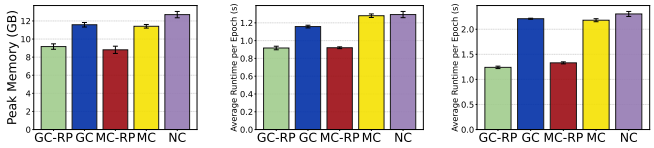
TABLE III
TEST ACCURACIES AFTER 1000 LEARNING EPOCHS ON **MNIST** AND **CIFAR-10** DATASETS UNDER FIVE BYZANTINE ATTACKS. IN EACH OF THE THREE HORIZONTAL BLOCKS AND UNDER EACH ATTACK, WE HIGHLIGHT IN **BOLD** THE BEST ACCURACY.

Robust Aggregation Rule	Algorithm	MNIST					CIFAR-10				
		ALIE	LF	SF	IPM	Mimic	ALIE	LF	SF	IPM	Mimic
NNM(−RP) ∘ GMed	GC−RP	90.00	94.49	**88.14**	94.73	94.21	40.29	48.10	**24.81**	48.29	47.67
	MC−RP	**90.01**	94.54	87.57	**94.92**	**94.41**	39.89	48.40	24.60	48.48	47.83
	GC	89.94	**94.60**	87.53	94.62	94.36	**40.45**	48.33	21.65	48.73	47.82
	MC	89.98	94.45	87.54	94.53	94.21	39.86	**48.91**	22.40	**49.48**	**48.46**
	NC	87.06	91.82	72.46	91.68	91.04	36.87	43.95	18.62	44.04	42.17
NNM(−RP) ∘ Krum	GC−RP	94.68	94.36	94.63	94.67	94.01	47.88	48.13	48.16	48.04	47.48
	MC−RP	**94.73**	94.13	**94.80**	**94.69**	94.13	**48.38**	**48.62**	**48.62**	**48.62**	47.43
	GC	94.62	94.30	94.70	94.65	**94.14**	47.87	48.06	48.31	48.05	46.87
	MC	94.70	**94.47**	94.78	94.66	93.57	47.82	47.69	48.51	47.78	**47.49**
	NC	90.64	89.73	89.89	90.12	89.73	43.89	43.34	44.26	44.04	41.48
NNM(−RP) ∘ CWTM	GC−RP	89.96	94.80	87.82	94.74	94.24	40.13	**48.34**	24.16	48.71	47.75
	MC−RP	89.88	94.59	**87.84**	94.70	94.12	**40.33**	47.37	22.60	**48.91**	**47.85**
	GC	89.90	**94.83**	87.42	**94.93**	94.30	39.94	48.16	**24.57**	48.71	47.79
	MC	**89.99**	94.48	87.39	94.61	**94.37**	39.95	48.21	23.70	47.99	47.77
	NC	87.75	92.01	72.97	92.03	91.13	36.20	43.76	17.64	44.78	41.62

$1.5\times$ higher accuracy than NC under certain attacks (Table III, SF column under CIFAR-10), highlighting the necessity of effective Byzantine resilience mechanisms in handling heavy-tailed noise.

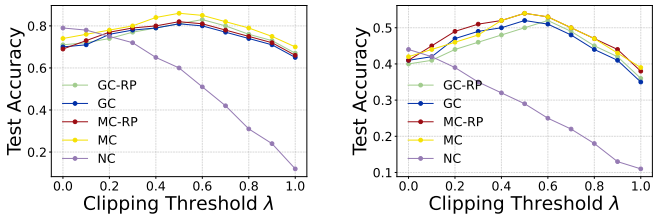
4) Comparison of NNM and ANNM-RP: The results indicate that the proposed approximate nearest neighbor mixing via random projection (ANNM−RP) achieves performance comparable to the original NNM but does not consistently outperform it. This suggests that while ANNM−RP is designed to improve computational efficiency in high-dimensional settings, its impact on accuracy remains similar to NNM. The comparable results suggest that the dimensionality reduction introduced by random projection effectively preserves key information. While ANNM−RP does not provide an accuracy advantage over NNM, its computational efficiency makes it a practical choice for large-scale FL scenarios with high-dimensional models, where the complexity of NNM can be a bottleneck.

5) Runtime and memory efficiency: To further evaluate the efficiency of our proposed framework, we measure both runtime and memory consumption across different methods. Specifically, we report the average per-epoch wall-clock time (seconds) and the average memory consumption (GB), where memory is measured as the sum of peak GPU memory (allocated/peak) and host resident memory (RSS). For runtime measurement, we adopt a warm-up protocol (the warm-up epochs are excluded from timing), and then compute the mean per-epoch cost over the full training run. For fairness, all methods share the same data-loading pipeline and augmentation strategy, and the reported runtime includes data loading, logging, and evaluation overheads. The results are shown in Fig. 5. They demonstrate that random projection-based methods achieve substantial computational savings in both runtime and memory compared to the other approaches, thereby confirming the practical advantages of using random projection. We further observe that clipping-based methods (whether gradient clipping or momentum clipping) are slightly more efficient than those without clipping, in terms of both memory and runtime. Moreover, the efficiency difference between gradient clipping and momentum clipping is negligible, which is consistent with our theoretical result on the equivalence of these two strategies.



(a) Peak memory usage across methods (b) Average per-epoch runtime on MNIST (c) Average per-epoch runtime on CIFAR-10

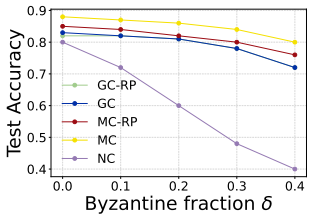
Fig. 5. Comparison of runtime and memory consumption of different methods. (a) Peak memory usage; (b) Average per-epoch runtime on MNIST; (c) Average per-epoch runtime on CIFAR-10.



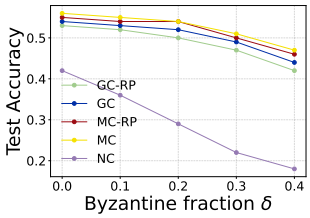
(a) Accuracy versus clipping threshold λ on MNIST (b) Accuracy versus clipping threshold λ on CIFAR-10

Fig. 6. Sensitivity of test accuracy to the clipping threshold λ. (a) MNIST; (b) CIFAR-10.

6) Sensitivity to the clipping threshold λ: We evaluate the impact of the clipping threshold by fixing all other hyperparameters and varying $\lambda \in [0, 1]$ with step size 0.1. The results, shown in Fig. 5, demonstrate that accuracy is relatively stable across a wide range of values. Performance peaks when λ is set to a moderate level $(0.4 - 0.6)$, while very small or very large thresholds lead to performance degradation. This confirms that the method is robust to the choice of λ, provided it is not set at the extremes.

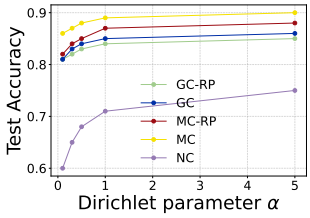


(a) Accuracy versus clipping threshold δ on MNIST

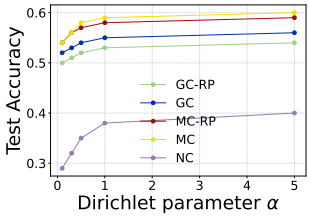


(b) Accuracy versus clipping threshold δ on CIFAR-10

Fig. 7. Sensitivity of test accuracy to the Byzantine fraction δ. (a) MNIST; (b) CIFAR-10.



(a) Accuracy versus clipping threshold α on MNIST



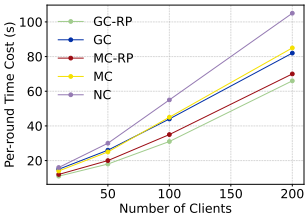
(b) Accuracy versus clipping threshold α on CIFAR-10

Fig. 8. Sensitivity of test accuracy to data heterogeneity controlled by the Dirichlet parameter ι. Smaller ι corresponds to more heterogeneous data, which induces heavier-tailed gradient noise. (a) MNIST; (b) CIFAR-10.

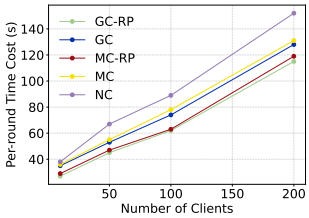
7) Sensitivity to the Byzantine fraction δ: We evaluate robustness with respect to the Byzantine fraction δ. Fig. 7 reports test accuracy on MNIST and CIFAR-10 as δ increases from 0 to 0.4. As expected, larger δ degrades performance for all methods. However, the non-clipping baseline (NC) deteriorates much more rapidly, while both gradient clipping and momentum clipping exhibit significantly more graceful degradation. Across the entire range of δ, clipping-based methods consistently achieve higher accuracy, indicating improved robustness to increasing adversarial participation. These results align with our theoretical analysis, which predicts unavoidable δ-dependent error, and empirically demonstrate that clipping mitigates the interaction between Byzantine updates and heavy-tailed gradient noise.

8) Sensitivity to data heterogeneity: We study sensitivity to data heterogeneity controlled by the Dirichlet parameter ι. Smaller ι corresponds to more heterogeneous data. Fig. 8 shows test accuracy on MNIST and CIFAR-10 as ι varies. As ι decreases, all methods degrade due to increased heterogeneity. However, the non-clipping baseline (NC) experiences a sharp performance drop, whereas clipping-based methods degrade much more gradually, especially in highly non-i.i.d. regimes. Since prior work [13] shows that increased heterogeneity empirically induces heavier-tailed gradient noise, varying ι serves as a practical proxy for tail behavior. The results indicate that clipping-based methods are substantially more robust to heterogeneity-induced heavy-tailed noise than methods without clipping.

9) Scalability with the number of clients: To further evaluate scalability, we vary the number of clients from 10 to 200 and measure the per-round time cost (seconds) under comparable settings (same total dataset size and batch size). The results, presented in Fig. 9, show consistent trends across both MNIST and CIFAR-10. As the number of clients increases, all methods exhibit higher per-round cost, but the increase is much slower for random projection-based methods (GC−RP and MC−RP). This demonstrates the superior scalability of ANNM−RP in handling large-scale FL systems, where both the model dimension and number of clients are large.



(a) Scalability analysis on MNIST



(b) Scalability analysis on CIFAR-10

Fig. 9. Scalability analysis: per-round time cost versus number of clients. Random projection-based methods (GC-RP, MC-RP) achieve significantly better scalability

VIII. Conclusions

This work studies Byzantine-resilient federated learning (FL) under heterogeneous data distributions and heavy-tailed gradient noise. We identified the key challenges posed by Byzantine failures in the presence of both data heterogeneity and infinite-variance noise, which render existing Byzantine-robust methods ineffective. We proposed a principled Byzantine-resilient FL framework that incorporates momentum normalization and clipping techniques to mitigate heavy-tail effects. We established the first high-probability convergence guarantees for Byzantine-resilient FL under heavy-tailed conditions. To address scalability in high-dimensional settings, we further introduced an efficient variant of the nearest neighbor mixing (NNM) technique based on random projections. Our theoretical analysis demonstrates the fundamental interaction between Byzantine robustness, heavy tails, and data heterogeneity, and our empirical results confirm that the proposed methods significantly outperform existing approaches in adversarial and heavy-tailed environments.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *International Conference on Artificial Intelligence and Statistics (AISTATS 2017)*. Fort Lauderdale, FL: PMLR, Apr. 2017, pp. 1273–1282.

[2] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[3] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated Learning for Internet of Things: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, Jul. 2021.

[4] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li, "A survey of federated learning for edge computing: Research problems and solutions," *Elsevier High-Confidence Computing*, vol. 1, no. 1, 2021.

[5] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, 1982.

[6] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *35th International Conference on Machine Learning (ICML 2018)*, J. Dy and A. Krause, Eds. Stockholm, Sweden: PMLR, Jul. 2018, pp. 5650–5659.

[7] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous federated learning: State-of-the-art and research challenges," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–44, Oct. 2023.

[8] M. Charikar, J. Steinhardt, and G. Valiant, "Learning from untrusted data," in *49th ACM Symposium on Theory of Computing (STOC 2017)*. Montréal, Canada: ACM, Jun. 2017, pp. 47–60.

[9] S. Liu, N. Gupta, and N. H. Vaidya, "Approximate byzantine fault-tolerance in distributed optimization," in *40th ACM Symposium on Principles of Distributed Computing (PODC 2021)*. Virtual Conference: ACM, Jul. 2021, pp. 379–389.

[10] S. P. Karimireddy, L. He, and M. Jaggi, "Byzantine-Robust Learning on Heterogeneous Datasets via Bucketing," in *10th International Conference on Learning Representations (ICLR 2022)*. Virtual Conference: OpenReview.net, Apr. 2022.

[11] Y. Allouah, S. Farhadkhani, R. Guerraoui, N. Gupta, R. Pinot, and J. Stephan, "Fixing by Mixing: A Recipe for Optimal Byzantine ML under Heterogeneity," in *26th International Conference on Artificial Intelligence and Statistics (AISTATS 2023)*. Valencia, Spain: PMLR, Apr. 2023, pp. 1232–1300.

[12] Z. Charles, Z. Garrett, Z. Huo, S. Shmulyian, and V. Smith, "On large-cohort training for federated learning," in *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Virtual Conference: Curran Associates Inc., Dec. 2021.

[13] H. Yang, P. Qiu, and J. Liu, "Taming Fat-Tailed ("Heavier-Tailed" with Potentially Infinite Variance) Noise in Federated Learning," in *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. New Orleans, LA: Curran Associates Inc., Nov. 2022.

[14] A. Cutkosky and H. Mehta, "High-probability bounds for non-convex stochastic optimization with heavy tails," in *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Virtual Conference: Curran Associates Inc., Dec. 2021.

[15] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation," in *35th Uncertainty in Artificial Intelligence Conference (UAI 2020)*. Virtual Conference: PMLR, Oct. 2020, pp. 261–270.

[16] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," in *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*. Vancouver, Canada: Curran Associates Inc., Dec. 2019.

[17] Z. Allen-Zhu, F. Ebrahimianghazani, J. Li, and D. Alistarh, "Byzantine-Resilient Non-Convex Stochastic Gradient Descent," in *9th International Conference on Learning Representations (ICLR 2021)*. Virtual Conference: OpenReview.net, May 2021.

[18] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *28th Annual Network and Distributed System Security Symposium (NDSS 2021)*. Virtual Conference: The Internet Society, Feb. 2021.

[19] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust Aggregation for Federated Learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.

[20] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, pp. 1–25, 2017.

[21] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed Optimisation for Non-Convex Problems," in *35th International Conference on Machine Learning (ICML 2018)*, J. Dy and A. Krause, Eds. Stockholm, Sweden: PMLR, Jul. 2018, pp. 560–569.

[22] R. Jin, Y. Huang, X. He, H. Dai, and T. Wu, "Stochastic-Sign SGD for Federated Learning with Theoretical Guarantees," arXiv, cs.LG, Sep. 2021.

[23] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *31st International Conference on Neural Information Processing Systems (NIPS 2017)*. Long Beach, CA: Curran Associates Inc., Dec. 2017.

[24] E. M. El Mhamdi, R. Guerraoui, and S. Rouault, "The Hidden Vulnerability of Distributed Learning in Byzantium," in *35th International Conference on Machine Learning (ICML 2018)*, J. Dy and A. Krause, Eds. Stockholm, Sweden: PMLR, Jul. 2018.

[25] U. Simsekli, L. Sagun, and M. Gurbuzbalaban, "A tail-index analysis of stochastic gradient noise in deep neural networks," in *36th International Conference on Machine Learning (ICML 2019)*, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, CA: PMLR, Jun. 2019, pp. 5827–5837.

[26] J. Zhang, S. P. Karimireddy, A. Veit, S. Kim, S. Reddi, S. Kumar, and S. Sra, "Why are adaptive methods good for attention models?" in *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds. Virtual Conference: Curran Associates Inc., Dec. 2020.

[27] E. Gorbunov, M. Danilova, and A. Gasnikov, "Stochastic optimization with heavy-tailed noise via accelerated gradient clipping," in *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds. Virtual Conference: Curran Associates Inc., Dec. 2020.

[28] M. Gurbuzbalaban, U. Simsekli, and L. Zhu, "The heavy-tail phenomenon in SGD," in *38th International Conference on Machine Learning (ICML 2021)*, M. Meila and T. Zhang, Eds., vol. 139. Virtual Conference: PMLR, Jul. 2021, pp. 3964–3975.

[29] T. H. Nguyen, U. Simsekli, M. Gurbuzbalaban, and G. Richard, "First exit time analysis of stochastic gradient descent under heavy-tailed gradient noise," in *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*. Vancouver, Canada: Curran Associates Inc., Dec. 2019.

[30] U. Simsekli, L. Zhu, Y. W. Teh, and M. Gurbuzbalaban, "Fractional underdamped langevin dynamics: Retargeting sgd with momentum under heavy-tailed gradient noise," in *37th International Conference on Machine Learning (ICML 2020)*. Virtual Conference: PMLR, Jul. 2020, pp. 8970–8980.

[31] L. Hodgkinson and M. Mahoney, "Multiplicative noise and heavy tails in stochastic optimization," in *38th International Conference on Machine Learning (ICML 2021)*, M. Meila and T. Zhang, Eds., vol. 139. Virtual Conference: PMLR, Jul. 2021, pp. 4262–4274.

[32] H. Wang, M. Gurbuzbalaban, L. Zhu, U. Simsekli, and M. A. Erdogdu, "Convergence rates of stochastic gradient descent under infinite noise variance," in *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, M. a. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Virtual Conference: Curran Associates Inc., Dec. 2021.

[33] Y. Tao, S. Cui, W. Xu, H. Yin, D. Yu, W. Liang, and X. Cheng, "Byzantine-Resilient Federated Learning at Edge," *IEEE Transactions on Computers*, vol. 72, no. 9, pp. 2600–2614, Sep. 2023.

[34] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[35] E. Gorbunov, G. Horváth, P. Richtárik, and G. Gidel, "Variance Reduction is an Antidote to Byzantines: Better Rates, Weaker Assumptions and Communication Compression as a Cherry on the Top," in *11th International Conference on Learning Representations (ICLR 2023)*. Kigali, Rwanda: OpenReview.net, May 2023.

[36] S. Farhadkhani, R. Guerraoui, N. Gupta, R. Pinot, and J. Stephan, "Byzantine machine learning made easy by resilient averaging of momentums," in *39th International Conference on Machine Learning (ICML 2022)*, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. Baltimore, MD: PMLR, Jul. 2022, pp. 6246–6283.

[37] E. M. El Mhamdi, R. Guerraoui, and S. Rouault, "Distributed momentum for byzantine-resilient stochastic gradient descent," in *9th International Conference on Learning Representations (ICLR 2021)*. Virtual Conference: OpenReview.net, May 2021.

[38] S. P. Karimireddy, L. He, and M. Jaggi, "Learning from history for byzantine robust optimization," in *38th International Conference on Machine Learning (ICML 2021)*, M. Meila and T. Zhang, Eds., vol. 139. Virtual Conference: PMLR, Jul. 2021, pp. 5311–5319.

[39] K. G. Larsen and J. Nelson, "Optimality of the Johnson-Lindenstrauss lemma," in *58th IEEE Symposium on Foundations of Computer Science (FOCS)*. Berkeley, CA: IEEE, Oct. 2017, pp. 633–638.

[40] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[41] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Technical Report TR-2009, Apr. 2009.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE International Conference on Computer Vision (ICCV 2015)*. Washington, D.C., United States: IEEE, Dec. 2015.

APPENDIX A
IN-EXPECTATION CONVERGENCE ANALYSIS FOR ALGORITHM 2

We begin our convergence analysis by bounding the bias and variance of the (aggregated) clipped gradients.

Lemma 10. Given a collection of K data distributions $\{\mathcal{D}_i\}_{i=1}^K$. For each $\mathcal{D}_i$, assume the stochastic loss gradients $\nabla\ell(x;z)$ computed over $\mathcal{D}$ satisfy the bounded h-th moment, i.e., $\mathbb{E}_{z\sim\mathcal{D}_i}[\|\nabla\ell(x;z)\|^h] \leq G^h$. The clipped gradient is defined as $\tilde{g}^{(i)}(x;z) \coloneqq \nabla\ell(x;z)\cdot\left(\frac{\lambda}{\|\nabla\ell(x;z)\|}\wedge 1\right)$, where $z\sim\mathcal{D}_i$. Let $\nabla\mathcal{L}_i(x) \coloneqq \mathbb{E}_{z\sim\mathcal{D}_i}[\nabla\ell(x;z)] =$ and $\nabla\mathcal{L}(x) \coloneqq \frac{1}{K}\sum_{i=1}^K \nabla\mathcal{L}_i(x)$. Then the aggregated clipped gradients $\tilde{g}(x,z) \coloneqq \frac{1}{K}\sum_{i=1}^K \tilde{g}^{(i)}(x,z)$ have bounded bias and variance:

- Bias: $\|\mathbb{E}[\tilde{g}(x;z)] - \nabla\mathcal{L}(x)\| \leq \frac{G^h}{\lambda^{h-1}}$
- Variance: $\mathbb{E}[\|\tilde{g}(x;z) - \mathbb{E}[\tilde{g}(x;z)]\|^2] \leq \frac{G^h\lambda^{2-h}}{K}$

Proof. First, we bound the bias for each individual gradient, i.e., $\|\mathbb{E}[\tilde{g}^{(i)}(x;z)] - \nabla\mathcal{L}_i(x)\|$, for $\forall i\in[K]$ and $z\sim\mathcal{D}_i$:

$$\begin{aligned}\|\mathbb{E}[\tilde{g}^{(i)}(x;z)] - \nabla\mathcal{L}_i(x)\| &= \|\mathbb{E}[\tilde{g}^{(i)}(x;z) - \nabla\ell(x;z)]\| \leq \mathbb{E}[\|\tilde{g}^{(i)}(x;z) - \nabla\ell(x;z)\|]\\ &= \mathbb{E}[\|\tilde{g}^{(i)}(x;z) - \nabla\ell(x;z)\|\cdot\mathbb{1}_{\|\nabla\ell(x;z)\|\geq\lambda}]\\ &\leq \mathbb{E}[\|\nabla\ell(x;z)\|\cdot\mathbb{1}_{\|\nabla\ell(x;z)\|\geq\lambda}] \leq \mathbb{E}\left[\frac{\|\nabla\ell(x;z)\|^h}{\lambda^{h-1}}\right] \leq \frac{G^h}{\lambda^{h-1}},\end{aligned}$$

where the first inequality follows by the Jensen's inequality, the second inequality follows by definition of $\tilde{g}^{(i)}$, and the third inequality follows by $\|\nabla\ell(x;z)\|^h\mathbb{1}_{\|\nabla\ell(x;z)\|\geq\lambda} \geq \|\nabla\ell(x;z)\|\lambda^{h-1}\mathbb{1}_{\|\nabla\ell(x;z)\|\geq\lambda}$.

Then, the bias of the aggregated clipped gradients is bounded as follows:

$$\|\mathbb{E}[\tilde{g}(x;z)] - \nabla\mathcal{L}(x)\| = \left\|\frac{1}{K}\sum_{i=1}^K\Big(\mathbb{E}[\tilde{g}^{(i)}(x;z)] - \nabla\mathcal{L}_i(x)\Big)\right\| \leq \frac{1}{K}\sum_{i=1}^K\left\|\mathbb{E}[\tilde{g}^{(i)}(x;z)] - \nabla\mathcal{L}_i(x)\right\| \leq \frac{G^h}{\lambda^{h-1}}.$$

Next, we bound the variance for each individual gradient, i.e, $\mathbb{E}[\|\tilde{g}^{(i)}(x;z) - \mathbb{E}[\tilde{g}^{(i)}(x;z)]\|^2]$:

$$\mathbb{E}[\|\tilde{g}^{(i)}(x;z) - \mathbb{E}[\tilde{g}^{(i)}(x;z)]\|^2] \leq \mathbb{E}[\|\tilde{g}^{(i)}(x;z)\|^2] \leq \mathbb{E}[\|\tilde{g}^{(i)}(x;z)\|^h\lambda^{2-h}] \leq G^h\lambda^{2-h}.$$

Finally, we bound the variance of the aggregated clipped gradients as follows:

$$\mathbb{E}[\|\tilde{g}(x;z) - \mathbb{E}[\tilde{g}(x;z)]\|^2] = \mathbb{E}\left[\left\|\frac{1}{K}\sum_{i=1}^K\tilde{g}^{(i)}(x;z) - \mathbb{E}[\tilde{g}^{(i)}(x;z)]\right\|^2\right] \leq \frac{1}{K^2}\sum_{i=1}^K\mathbb{E}\left[\|\tilde{g}^{(i)}(x;z) - \mathbb{E}[\tilde{g}^{(i)}(x;z)\|^2\right] = \frac{G^h\lambda^{2-h}}{K},$$

where the inequality follows from the fact that $\left\{\tilde{g}^{(i)}(x;z) - \mathbb{E}[\tilde{g}^{(i)}(x;z)]\right\}$ form a martingale difference sequence. □

To establish the in-expectation convergence guarantee of Algorithm 2, we analyze the expected error in estimating the true gradient $\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})$ using the aggregated momentum $\widehat{m}_t$, i.e., $\mathbb{E}[\|\widehat{m}_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\|]$. Define the aggregated honest gradients and momentum terms as:

$$\widetilde{g}_t = \frac{1}{n-f}\sum_{i\in\mathcal{H}}\widetilde{g}_t^{(i)}, \quad m_t \coloneqq \frac{1}{n-f}\sum_{i\in\mathcal{H}}m_t^{(i)}. \tag{9}$$

Then the expected estimation error can be decomposed using the triangle inequality as follows:

$$\mathbb{E}[\|\widehat{m}_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\|] \leq \underbrace{\mathbb{E}[\|\widehat{m}_t - m_t\|]}_{\text{Expected Aggregation Error}} + \underbrace{\mathbb{E}[\|m_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\|]}_{\text{Expected Momentum Deviation}}.$$

Next, we separately analyze the *expected aggregation error* and *expected momentum deviation*. Notably, due to the (δ,κ)-robustness, the expected aggregation error reduces to analyzing the *expected momentum drift* which is defined as $\mathbb{E}\left[\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|m_t^{(i)} - m_t\|^2\right]$. The following Lemma 11, Lemma 12 and Lemma 13, provide upper bounds for *expected momentum drift*, *expected aggregation error*, and *expected momentum deviation*, respectively. Throughout the analysis, we use $\alpha \coloneqq 1-\beta$.

Lemma 11 (Expected Momentum Drift)**.** With probability at least $1-\omega$, for each $t\in[T]$, we have

$$\mathbb{E}\left[\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|m_t^{(i)} - m_t\|^2\right] \leq 6\frac{\alpha G^h}{\lambda^{h-1}} + 3\left(1+\frac{1}{n-f}\right)\alpha G^h\lambda^{2-h} + 3H^2.$$

Proof of Lemma 11.

$$
\begin{aligned}
&\mathbb{E}\left[\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|m_t^{(i)}-m_t\|^2\right]=(1-\beta)^2\mathbb{E}\left[\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\left\|\sum_{k=1}^{t}\beta^{t-k}(\widetilde{g}_k^{(i)}-\widetilde{g}_k)\right\|^2\right]\\
\leq&3(1-\beta)^2\left(\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\mathbb{E}\left[\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\widetilde{g}_k^{(i)}-\nabla\mathcal{L}_i(x_{k-1})\right)\right\|^2\right]+\mathbb{E}\left[\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\widetilde{g}_k-\nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\right)\right\|^2\right]\right.\\
&\left.+\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\nabla\mathcal{L}_i(x_{k-1})-\nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\right)\right\|^2\right)\\
=&3(1-\beta)^2\left(\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\mathbb{E}\left[\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\widetilde{g}_k^{(i)}-\mathbb{E}[\widetilde{g}_k^{(i)}]+\mathbb{E}[\widetilde{g}_k^{(i)}]-\nabla\mathcal{L}_i(x_{k-1})\right)\right\|^2\right]\right.\\
&+\mathbb{E}\left[\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\widetilde{g}_k-\mathbb{E}[\widetilde{g}_k]+\mathbb{E}[\widetilde{g}_k]-\nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\right)\right\|^2\right]\\
&\left.+\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\nabla\mathcal{L}_i(x_{k-1})-\nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\right)\right\|^2\right)\\
\leq&3(1-\beta)^2\left(\frac{2}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\mathbb{E}\left[\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\widetilde{g}_k^{(i)}-\mathbb{E}\left[\widetilde{g}_k^{(i)}\right]\right)\right\|^2\right]+\frac{2}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\mathbb{E}[\widetilde{g}_k^{(i)}]-\nabla\mathcal{L}_i(x_{k-1})\right)\right\|^2\right.\\
&+2\mathbb{E}\left[\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\widetilde{g}_k-\mathbb{E}\left[\widetilde{g}_k\right]\right)\right\|^2\right]+2\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\mathbb{E}[\widetilde{g}_k]-\nabla\mathcal{L}_i(x_{k-1})\right)\right\|^2\\
&\left.+\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\nabla\mathcal{L}_i(x_{k-1})-\nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\right)\right\|^2\right)\\
\leq&3(1-\beta)^2\left(\frac{2}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\sum_{k=1}^{t}\beta^{2(t-k)}\mathbb{E}\left[\left\|\widetilde{g}_k^{(i)}-\mathbb{E}\left[\widetilde{g}_k^{(i)}\right]\right\|^2\right]+\frac{2}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\frac{1}{1-\beta}\sum_{k=1}^{t}\beta^{t-k}\left\|\mathbb{E}[\widetilde{g}_k^{(i)}]-\nabla\mathcal{L}_i(x_{k-1})\right\|^2\right.\\
&+2\sum_{k=1}^{t}\beta^{2(t-k)}\mathbb{E}\left[\left\|\widetilde{g}_k-\mathbb{E}\left[\widetilde{g}_k\right]\right\|^2\right]+\frac{2}{1-\beta}\sum_{k=1}^{t}\beta^{t-k}\left\|\mathbb{E}[\widetilde{g}_k]-\nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\right\|^2\\
&\left.+\frac{1}{1-\beta}\sum_{k=1}^{t}\beta^{t-k}\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\left\|\nabla\mathcal{L}_i(x_{k-1})-\nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\right\|^2\right)\\
\leq&3(1-\beta)^2\left(\frac{2}{1-\beta^2}G^h\lambda^{2-h}+\frac{2}{(1-\beta)^2}\frac{G^{2h}}{\lambda^{2(h-1)}}+\frac{2}{1-\beta^2}\frac{G^h\lambda^{2-h}}{n-f}+\frac{2}{(1-\beta)^2}\frac{G^{2h}}{\lambda^{2(h-1)}}+\frac{H^2}{(1-\beta)^2}\right)\\
=&12\frac{G^{2h}}{\lambda^{2(h-1)}}+6\frac{1-\beta}{1+\beta}\left(1+\frac{1}{n-f}\right)G^h\lambda^{2-h}+3H^2\\
\leq&12\frac{G^{2h}}{\lambda^{2(h-1)}}+6\left(1+\frac{1}{n-f}\right)\alpha G^h\lambda^{2-h}+3H^2.
\end{aligned}
$$

In the third inequality, the variance-related terms follow by the fact that $\{\widetilde{g}_k^{(i)}-\mathbb{E}[\widetilde{g}_k^{(i)}]\}$ and $\{\widetilde{g}_k-\mathbb{E}[\widetilde{g}_k]\}$ are martingale difference sequences, and the bias-related terms follow by the weighted Cauchy-Schwarz inequality. The third inequality follows by Lemma 10 and Assumption 2 (bounded heterogeneity). □

Lemma 12 (Expected Aggregation Error)**.** Let $\texttt{Agg}$ be (δ,κ)-robust. Define $\epsilon_t \coloneqq m_t-\widehat{m}_t$. For all $t\in[T]$, we have

$$\mathbb{E}[\|\epsilon_t\|]\leq 2\sqrt{3\kappa}\frac{G^h}{\lambda^{h-1}}+2\sqrt{3\kappa\alpha}G^{\frac{h}{2}}\lambda^{1-\frac{h}{2}}+\sqrt{3\kappa}H.$$

Proof of Lemma 12. The proof follows directly from Lemma 2 and the (δ,κ)-robustness of $\texttt{Agg}$. □

Lemma 13 (Expected Momentum Deviation)**.** Define $\Delta_t \coloneqq m_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})$. For all $t \in [T]$, we have

$$\mathbb{E}[\|\Delta_t\|] \leq \beta^t G + \frac{\sqrt{\alpha} G^{\frac{h}{2}} \lambda^{1-\frac{h}{2}}}{\sqrt{n-f}} + \frac{G^h}{\lambda^{h-1}} + \frac{\beta}{1-\beta}\eta L.$$

Proof of Lemma 13. First note that

$$\Delta_t = m_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1}) = \beta m_{t-1} + (1-\beta)\widetilde{g}_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1}).$$

Then by adding and subtracting $\beta\nabla\mathcal{L}_{\mathcal{H}}(x_{t-2})$ and $\beta\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})$, we obtain that, for $t \geq 2$,

$$\begin{aligned}\Delta_t &= \beta m_{t-1} - \beta\nabla\mathcal{L}_{\mathcal{H}}(x_{t-2}) + (1-\beta)\widetilde{g}_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1}) \\ &\quad + \beta\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1}) + \beta\nabla\mathcal{L}_{\mathcal{H}}(x_{t-2}) - \beta\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1}) \\ &= \beta\Delta_{t-1} + (1-\beta)(\widetilde{g}_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})) \\ &\quad + \beta(\nabla\mathcal{L}_{\mathcal{H}}(x_{t-2}) - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})).\end{aligned}$$

Defining $m_0 = 0$, $w_{-1} = w_0$ and $\Delta_0 = -\nabla\mathcal{L}_{\mathcal{H}}(x_0)$, then $\Delta_t = \beta\Delta_{t-1} + (1-\beta)(\widetilde{g}_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})) + \beta(\nabla\mathcal{L}_{\mathcal{H}}(x_{t-2}) - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1}))$ holds also for $t = 1$. Hence, for all $t \geq 1$, we have

$$\Delta_t = \beta\Delta_{t-1} + (1-\beta)\cdot(\tilde{g}_t - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})) + \beta\left(\nabla\mathcal{L}_{\mathcal{H}}(x_{t-2}) - \nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\right).$$

By unraveling the recursion for all iterations, we have

$$\Delta_t = \beta^t\Delta_0 + (1-\beta)\sum_{k=1}^{t}\beta^{t-k}\left(\tilde{g}_k - \mathbb{E}[\tilde{g}_k]\right) + (1-\beta)\sum_{k=1}^{t}\beta^{t-k}\left(\mathbb{E}[\tilde{g}_k] - \nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\right) + \sum_{k=1}^{t}\beta^{t-k+1}\left(\nabla\mathcal{L}_{\mathcal{H}}(x_{k-2}) - \nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\right)$$

Using the triangle inequality, we have

$$\begin{aligned}\|\Delta_t\| \leq& \beta^t\|\Delta_0\| + (1-\beta)\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\tilde{g}_k - \mathbb{E}[\tilde{g}_k]\right)\right\| + (1-\beta)\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\mathbb{E}[\tilde{g}_k] - \nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\right)\right\| \\ &+ \sum_{k=1}^{t}\beta^{t-k+1}\|\nabla\mathcal{L}_{\mathcal{H}}(x_{k-2}) - \nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\| \\ \leq& \beta^t G + (1-\beta)\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\tilde{g}_k - \mathbb{E}[\tilde{g}_k]\right)\right\| + (1-\beta)\sum_{k=1}^{t}\beta^{t-k}\|\mathbb{E}[\tilde{g}_k] - \nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\| + \frac{\beta}{1-\beta}\eta L\end{aligned}$$

where in the second inequality we used $\|\Delta_0\| = \|\nabla\mathcal{L}_{\mathcal{H}}(x_0)\| \leq \mathbb{E}_z[\|\nabla\ell(x_0,z)\|^h]^{\frac{1}{h}} \leq G$ for the first term and the L-smooth of $\nabla\mathcal{L}_{\mathcal{H}}$ for the last term. Taking the expectation on both sides, we obtain

$$\begin{aligned}\mathbb{E}[\|\Delta_t\|] &\leq \beta^t G + (1-\beta)\mathbb{E}\left[\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\tilde{g}_k - \mathbb{E}[\tilde{g}_k]\right)\right\|\right] + (1-\beta)\sum_{k=1}^{t}\beta^{t-k}\|\mathbb{E}[\tilde{g}_k] - \nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\| + \frac{\beta}{1-\beta}\eta L \\ &\leq \beta^t G + (1-\beta)\sqrt{\mathbb{E}\left[\left\|\sum_{k=1}^{t}\beta^{t-k}\left(\tilde{g}_k - \mathbb{E}[\tilde{g}_k]\right)\right\|^2\right]} + (1-\beta)\sum_{k=1}^{t}\beta^{t-k}\|\mathbb{E}[\tilde{g}_k] - \nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\| + \frac{\beta}{1-\beta}\eta L \\ &\leq \beta^t G + (1-\beta)\sqrt{\sum_{k=1}^{t}\beta^{2(t-k)}\mathbb{E}\left[\|\tilde{g}_k - \mathbb{E}[\tilde{g}_k]\|^2\right]} + (1-\beta)\sum_{k=1}^{t}\beta^{t-k}\|\mathbb{E}[\tilde{g}_k] - \nabla\mathcal{L}_{\mathcal{H}}(x_{k-1})\| + \frac{\beta}{1-\beta}\eta L \\ &\leq \beta^t G + \sqrt{\frac{1-\beta}{1+\beta}}\frac{G^{\frac{h}{2}}\lambda^{1-\frac{h}{2}}}{\sqrt{n-f}} + \frac{G^h}{\lambda^{h-1}} + \frac{\beta}{1-\beta}\eta L \\ &\leq \beta^t G + \sqrt{1-\beta}\frac{G^{\frac{h}{2}}\lambda^{1-\frac{h}{2}}}{\sqrt{n-f}} + \frac{G^h}{\lambda^{h-1}} + \frac{\beta}{1-\beta}\eta L\end{aligned}$$

□

A critical component of the expected convergence analysis is the expected one-step descent bound.

Lemma 14 (Expected Descent Bound)**.** Let Assumption 1 hold. For every model update from x_{t-1} to x_t, the following one-step descent result holds in expectation:

$$\mathbb{E}[\mathcal{L}_{\mathcal{H}}(x_t) - \mathcal{L}_{\mathcal{H}}(x_{t-1})] \leq -\eta\mathbb{E}[\|\nabla\mathcal{L}_{\mathcal{H}}(x_{t-1})\|] + 2\eta\mathbb{E}[\|\epsilon_t + \Delta_t\|] + \frac{L}{2}\eta^2.$$

Proof of Lemma 14. Consider an arbitrary step t. Since $\mathcal{L}_{\mathcal{H}}$ is L-smooth, we have

$$\begin{aligned}
&\mathcal{L}_{\mathcal{H}}(x_t) - \mathcal{L}_{\mathcal{H}}(x_{t-1}) \\
\leq &\langle \nabla \mathcal{L}_{\mathcal{H}}(x_{t-1}), x_t - x_{t-1} \rangle + \frac{L}{2}\|x_t - x_{t-1}\|^2 = -\eta \left\langle \nabla \mathcal{L}_{\mathcal{H}}(x_{t-1}), \frac{\widehat{m}_t}{\|\widehat{m}_t\|} \right\rangle + \frac{L}{2}\eta^2 \\
= &-\eta\|\widehat{m}_t\| + \eta \left\langle \epsilon_t + \Delta_t, \frac{\widehat{m}_t}{\|\widehat{m}_t\|} \right\rangle + \frac{L}{2}\eta^2 \leq -\eta\|\nabla \mathcal{L}_{\mathcal{H}}(x_{t-1}) + \Delta_t + \epsilon_t\| + \eta\|\epsilon_t + \Delta_t\| + \frac{L}{2}\eta^2 \\
\leq &-\eta\|\nabla \mathcal{L}_{\mathcal{H}}(x_{t-1})\| + 2\eta\|\epsilon_t + \Delta_t\| + \frac{L}{2}\eta^2
\end{aligned}$$

Taking expectation on both sides above completes the proof. □

Theorem 3 (Expected Convergence Guarantee for Algorithm 2)**.** Suppose Assumptions 1, 2, and 3 hold. Define $C := \left(\frac{L(\mathcal{L}_{\mathcal{H}}(x_0) - \mathcal{L}_{\mathcal{H}}(x_T))}{G^2}\right)^{\frac{h}{3h-2}} \vee 1$ and parameter $\alpha := C \cdot \left(\frac{1}{T^{\frac{h}{2}}\left(1+\sqrt{\kappa}\right)^{2-h}\left(\frac{1}{n-f}+\kappa\right)^{h-1}}\right)^{\frac{2}{3h-2}}$. Set $\beta = 1-\alpha$, $\eta = \left(\frac{\mathcal{L}_{\mathcal{H}}(x_0) - \mathcal{L}_{\mathcal{H}}(x_T)}{L} \cdot \frac{\alpha}{T}\right)^{\frac{1}{2}}$, and $\lambda = \frac{G}{\alpha^{\frac{1}{h}}}\left(\frac{1+\sqrt{\kappa}}{\frac{1}{\sqrt{n-f}}+\sqrt{\kappa}}\right)^{\frac{2}{h}}$ in Algorithm 2. Then,

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla \mathcal{L}_{\mathcal{H}}(x_{t-1})\|] \leq \widetilde{O}\left(\left[\left(1+\sqrt{\kappa}\right)^{\frac{2-h}{h-1}}\left(\frac{1}{n}+\kappa\right)\frac{1}{T}\right]^{\frac{h-1}{3h-2}} + \sqrt{\kappa}H\right).$$

Proof of Theorem 3. From Lemma 14, for each iteration $t \in [T]$, we have that,

$$\mathbb{E}[\|\nabla \mathcal{L}_{\mathcal{H}}(x_{t-1})\|] \leq \frac{\mathbb{E}[\mathcal{L}_{\mathcal{H}}(x_{t-1}) - \mathcal{L}_{\mathcal{H}}(x_t)]}{\eta} + 2\mathbb{E}[\|\epsilon_t\|] + 2\mathbb{E}[\|\Delta_t\|] + \frac{L}{2}\eta.$$

Let $\mathcal{L}^*$ be the minimum value of $\mathcal{L}_{\mathcal{H}}$ and $\mathcal{L}_0 \coloneqq \mathcal{L}_{\mathcal{H}}(x_0)$. Take average from $t=1$ to T, and we have

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla \mathcal{L}_{\mathcal{H}}(x_{t-1})\|] \leq &\frac{\mathcal{L}_{\mathcal{H}}(x_0) - \mathbb{E}[\mathcal{L}_{\mathcal{H}}(x_T)]}{\eta T} + \frac{2}{T}\sum_{t=1}^{T}\mathbb{E}[\|\epsilon_t\|] + \frac{2}{T}\sum_{t=1}^{T}\mathbb{E}[\|\Delta_t\|] + \frac{L\eta}{2} \\
\leq &O\left(\underbrace{\frac{\mathcal{L}_0 - \mathcal{L}^*}{\eta T}}_{\mathscr{A}} + \underbrace{(1+\sqrt{\kappa})\frac{G^h}{\lambda^{h-1}}}_{\mathscr{B}} + \underbrace{\left(\frac{1}{\sqrt{n-f}} + \sqrt{\kappa}\right)\sqrt{\alpha}G^{\frac{h}{2}}\lambda^{1-\frac{h}{2}}}_{\mathscr{C}} + \underbrace{\frac{\eta L}{\alpha}}_{\mathscr{D}} + \underbrace{\frac{G}{\alpha T}}_{\mathscr{E}} + \sqrt{\kappa}H\right).
\end{aligned}$$

First, we minimize the upper bound by appropriately setting the parameters η, λ and α. Firstly, balancing the trade-off between terms $\mathscr{A}$ and $\mathscr{D}$ in η gives us $\eta = \left(\frac{\mathcal{L}_{\mathcal{H}}(x_0) - \mathcal{L}_{\mathcal{H}}(x_T)}{L} \cdot \frac{\alpha}{T}\right)^{\frac{1}{2}}$. The upper bound becomes

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla \mathcal{L}_{\mathcal{H}}(x_{t-1})\|] \leq O\left(\underbrace{\sqrt{\frac{L(\mathcal{L}_0 - \mathcal{L}^*)}{\alpha T}}}_{\mathscr{F}} + \underbrace{(1+\sqrt{\kappa})\frac{G^h}{\lambda^{h-1}}}_{\mathscr{B}} + \underbrace{\left(\frac{1}{\sqrt{n-f}} + \sqrt{\kappa}\right)\sqrt{\alpha}G^{\frac{h}{2}}\lambda^{1-\frac{h}{2}}}_{\mathscr{C}} + \underbrace{\frac{G}{\alpha T}}_{\mathscr{E}} + \sqrt{\kappa}H\right)$$

Next we balance the trade-off in λ between terms $\mathscr{B}$ and $\mathscr{C}$, which gives us $\lambda = \frac{G}{\alpha^{\frac{1}{h}}}\left(\frac{1+\sqrt{\kappa}}{\frac{1}{\sqrt{n-f}}+\sqrt{\kappa}}\right)^{\frac{2}{h}}$. Then we get the convergence upper bound of

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla \mathcal{L}_{\mathcal{H}}(x_{t-1})\|] \leq O\left(\underbrace{\sqrt{\frac{L(\mathcal{L}_0 - \mathcal{L}^*)}{\alpha T}}}_{\mathscr{F}} + \underbrace{G(1+\sqrt{\kappa})^{\frac{2-h}{h}}\left(\frac{1}{n-f}+\kappa\right)^{\frac{h-1}{h}}\alpha^{\frac{h-1}{h}}}_{\mathscr{G}} + \underbrace{\frac{G}{\alpha T}}_{\mathscr{E}} + \sqrt{\kappa}H\right)$$

Finally, we balance the two trade-offs in α: one between terms $\mathscr{G}$ and $\mathscr{F}$, the other between $\mathscr{G}$ and $\mathscr{E}$, which gives us

$$\alpha_1 = \left(\frac{L^h(\mathcal{L}_0 - \mathcal{L}^*)^h}{G^{2h}T^h(1+\sqrt{\kappa})^{2(2-h)}\left(\frac{1}{n-f}+\kappa\right)^{2(h-1)}}\right)^{\frac{1}{3h-2}}$$

and

$$\alpha_2 = \left(\frac{1}{T^h\left(1+\sqrt{\kappa}\right)^{2-h}\left(\frac{1}{n-f}+\kappa\right)^{h-1}}\right)^{\frac{1}{2h-1}},$$

respectively. Denote $C = \left(\frac{L(\mathcal{L}_0 - \mathcal{L}^*)}{G^2}\right)^{\frac{h}{3h-2}} \vee 1$. Setting $\alpha = C \cdot \left(\frac{1}{T^{\frac{h}{2}}\left(1+\sqrt{\kappa}\right)^{2-h}\left(\frac{1}{n-f}+\kappa\right)^{h-1}}\right)^{\frac{2}{3h-2}}$ gives us that:

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla \mathcal{L}_{\mathcal{H}}(x_{t-1})\|] &\leq \widetilde{O}\left(\left(1+\sqrt{\kappa}\right)^{\frac{2-h}{3h-2}}\left(\frac{1}{n-f}+\kappa\right)^{\frac{h-1}{3h-2}}\frac{1}{T^{\frac{h-1}{3h-2}}}+\sqrt{\kappa}H\right)\\ &\leq \widetilde{O}\left(\left[\left(1+\sqrt{\kappa}\right)^{\frac{2-h}{h-1}}\left(\frac{1}{n}+\kappa\right)\frac{1}{T}\right]^{\frac{h-1}{3h-2}}+\sqrt{\kappa}H\right),\end{aligned}$$

where $\kappa = \Theta(\delta)$ when applying NNM with standard robust aggregation rules. □