MAKING DISTANCE-BASED ROBUST AGGREGATION FAST: A SERVER-SIDE EMBEDDING WRAPPER

Anonymous authors
Paper under double-blind review

ABSTRACT

Distance-based robust aggregators such as Krum, Bulyan, Geometric Median, and MCA defend Federated Learning against Byzantine clients, but they are slow on the server: Krum costs $\mathcal{O}(M^2p)$, and related pairwise geometry is likewise super-linear in the model dimension p. Existing JL and sketch protocols mainly reduce communication or cryptographic overhead; they do not accelerate the standard centralized interface in which clients still upload full-dimensional gradients. We propose Projected Dimensionality Reduction (PDR), a plug-and-play server-side wrapper: after full-dimensional uploads, the server applies a sparse embedding, computes reliability weights in $\mathbb{R}^k$, and reconstructs the update in $\mathbb{R}^p$, reducing aggregation to $\mathcal{O}(Mps + C_{\mathrm{RAgg}}(M,k))$. For this wrapper we prove that the usual robust-FL rates $\mathcal{O}(1/\sqrt{T})$ (non-convex) and $\tilde{\mathcal{O}}(1/T)$ (strongly convex) are retained, with the Byzantine error floor inflated only by $\frac{1+\epsilon}{1-\epsilon}$. Experiments show large reductions in server wall-clock time while tracking the accuracy of the wrapped aggregators. To our knowledge, PDR is the first server-side sparse-embedding wrapper for existing distance-based RAggs under full-dimensional uploads, together with a convergence analysis of that wrapper.

1 INTRODUCTION

Federated Learning (FL) lets clients train a shared model without releasing raw data (Zuo et al., 2025a; Konečnỳ et al., 2016; Li et al., 2020; Dorfman et al., 2023; Zhao et al., 2024). In the standard centralized protocol, clients upload gradient vectors in $\mathbb{R}^p$ and the server aggregates them (Wang et al., 2019; Guo et al., 2023). When some clients are Byzantine (Zuo et al., 2025b; Yang et al., 2020; Cao & Lai, 2019; Chen et al., 2017), that aggregation cannot be a plain mean: distance-based robust rules such as Krum (Blanchard et al., 2017), Bulyan (El-Mhamdi et al., 2018), Geometric Median (Chen et al., 2017), and MCA (Luan et al., 2025) identify outliers from pairwise Euclidean geometry (Blanchard et al., 2017; Yin et al., 2018; Chen et al., 2017). These rules work, but they are slow. Even at the model dimensions already common in robust FL (Cao et al., 2021; Shejwalkar et al., 2022), Krum costs $\mathcal{O}(M^2p)$ on the server after concurrent uploads, and related geometry in Geometric Median and Bulyan is likewise super-linear in p, so pairwise geometry occupies a large fraction of each round even though the uplink is already full-dimensional. This paper targets that bottleneck: make existing distance-based aggregators fast on the server, without changing what clients upload.

Recent work reduces dimension in robust FL, but not for this interface. DnC (Shejwalkar & Houmansadr, 2021) scores random coordinate subsets, with limited guarantees and known bypasses (Baruch et al., 2019; Shejwalkar et al., 2022). RobAJoL / Fed-DPRoC (Xia et al., 2025) applies JL (e.g., Count Sketch) *on the client* so that communication and robust averaging both live in a compressed domain, often with DP. SketchGuard (Rangwala et al., 2025) uses Count Sketch for neighbor screening in *decentralized* FL. ABBR (Zhang et al., 2025) uses random projection to speed pairwise distances inside *MPC* aggregation. These results show that distance geometry survives sketching, yet they optimize communication or cryptography. We instead keep the standard centralized, full-upload interface and accelerate the *server arithmetic* of existing vector-level RAggs: reliability weights are computed in $\mathbb{R}^k$, the update is reconstructed in $\mathbb{R}^p$, and the uplink payload is unchanged. The construction is complementary to the protocols above. To our knowledge, PDR is the first server-side sparse-embedding wrapper for existing distance-based RAggs (Krum, Geometric

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Median, Bulyan, and MCA) under full-dimensional uploads, together with a convergence analysis of that wrapper.

Distance-based rules depend on *relative distances* among updates, not on exact coordinates in $\mathbb{R}^p$. Inspired by the JL lemma and subspace embeddings (Woodruff, 2014; Nelson & Nguyên, 2013; Kane & Nelson, 2014), we propose Projected Dimensionality Reduction (PDR): the server projects the uploaded gradients into $\mathbb{R}^k$ ($k \ll p$) with a sparse random map, runs the base robust aggregator to obtain reliability weights, and reconstructs the global update as a weighted combination of the *original* high-dimensional gradients.

Our main contributions are:

- **A drop-in server accelerator** for existing vector-level distance-based RAggs under full-dimensional uploads. PDR does not compress communication. It reduces server arithmetic from a p-dimensional robust rule to a sparse projection of cost $\mathcal{O}(Mps)$ plus aggregation in dimension k (Remark 2.4), e.g., Krum from $\mathcal{O}(M^2p)$ to $\mathcal{O}(Mps + M^2k)$.
- **Convergence of the wrapper.** Point-set JL is not enough, because the reconstruction weights depend on P. Using sparse subspace embeddings, we give the first convergence analysis of this interface: standard robust-FL rates $\mathcal{O}(1/\sqrt{T})$ and $\tilde{\mathcal{O}}(1/T)$ are retained, with the Byzantine error floor inflated only by $\frac{1+\epsilon}{1-\epsilon}$.
- **Inclusive wall-clock speedups** on TinyImageNet and CIFAR-100. PDR cuts server aggregation time while tracking the accuracy of the wrapped aggregators under Byzantine attacks and non-IID data.

2 METHODOLOGY

In this section we specify the FL and Byzantine setting, then present PDR as a server-side accelerator for distance-based robust aggregators. Throughout, $\|x\|$ is the Euclidean ℓ_2-norm and $\langle x, y\rangle$ the standard inner product. We use standard asymptotic notation $\mathcal{O}$, Ω, and Θ; $\tilde{\mathcal{O}}$ hides polylogarithmic factors in the relevant parameters (e.g., M, $1/\delta$, $1/\epsilon$).

2.1 PROBLEM SETUP

FL optimization problem: A server maintains $w \in \mathbb{R}^p$ for M clients $\mathcal{M} \triangleq \{1, 2, \ldots, M\}$. Client m holds a local dataset $\mathcal{S}_m$ of size S_m. Writing $\mathcal{H} \triangleq \mathcal{M} \setminus \mathcal{B}$ for the honest clients, where $\mathcal{B}$ is the Byzantine set introduced below, the goal is

$$\min_{w\in\mathbb{R}^p} F(w) = \frac{1}{\sum_{i\in\mathcal{H}} S_i} \sum_{m\in\mathcal{H}} S_m F_m(w), \quad F_m(w) = \frac{1}{S_m} \sum_{i\in\mathcal{S}_m} f(w, s_{m,i}), \tag{1}$$

where $f(\cdot, s)$ is the per-sample loss. Write $\nabla F(w)$ and $\nabla F_m(w)$ for the global and local full gradients.

Byzantine attacks: Let $\mathcal{B}$ be a set of B Byzantine clients, fixed over time and unknown to the server. Following Karimireddy et al. (2022) and Gorbunov et al. (2023), the sums in (1) range over the honest clients $\mathcal{H}$, and every assumption we make is imposed on honest clients only. At round t, client m uploads $g_m^t \in \mathbb{R}^p$: an honest client sends a stochastic gradient $\nabla F_m(w^t, \xi_m^t)$, while any $m \in \mathcal{B}$ may send an arbitrary vector

$$g_m^t = \star, \quad \text{for } m \in \mathcal{B}. \tag{2}$$

The data-weighted malicious fraction is

$$b \triangleq \frac{1}{\sum_{i\in\mathcal{M}} S_i} \sum_{m\in\mathcal{B}} S_m. \tag{3}$$

Without a clean server dataset or other side information, robust aggregation requires $b < 0.5$.

Robust aggregator: We follow the (b, c)-robustness framework of Gorbunov et al. (2023), which generalizes Karimireddy et al. (2022). Their reference mean is the uniform average of honest vectors; we use the S_i-weighted honest mean to match the global objective (1). The two coincide when all honest S_i are equal.

Definition 2.1 ((b, c)-Robustness Aggregator)**.** Let $\mathcal{G} = \{g_1, g_2, \ldots, g_M\}$ be vectors in $\mathbb{R}^p$, and let $\mathcal{G}_{\text{true}} \subseteq \mathcal{G}$ be an honest subset carrying at least a $(1-b)$ fraction of the data, i.e. $\sum_{i\in\mathcal{G}_{\text{true}}} S_i \geq (1-b)\sum_{i\in\mathcal{M}} S_i$, for some $b \leq b_{\max} < 0.5$ measured in the same data-weighted sense as (1). Write $\nu^2 = \max_{i,j\in\mathcal{G}_{\text{true}}} \mathbb{E}\left\|g_i - g_j\right\|^2$. A (b, c)-robust aggregator (RAgg) returns $\hat{g} = \text{RAgg}(g_1, g_2, \ldots, g_M)$ such that, for a constant $c > 0$ that does not depend on ν^2,

$$\mathbb{E}\left\|\hat{g} - \bar{g}\right\|^2 \leq cb\nu^2, \tag{4}$$

where $\bar{g} = \frac{1}{\sum_{i\in\mathcal{G}_{\text{true}}} S_i} \sum_{i\in\mathcal{G}_{\text{true}}} S_i g_i$ is the S_i-weighted mean of $\mathcal{G}_{\text{true}}$, matching $\bar{g}^t$ in Lemma 3.6.

2.2 Algorithm

PDR does not change the client uplink: each client still uploads a vector in $\mathbb{R}^p$. What it accelerates is the server arithmetic of a vector-level distance-based RAgg on $\{g_m\} \subset \mathbb{R}^p$. The wrapper has three steps: (i) a sparse projection into $\mathbb{R}^k$, (ii) the original robust rule run in $\mathbb{R}^k$ to obtain reliability weights, and (iii) reconstruction of the server update as a weighted combination of the original high-dimensional gradients.

Distance-based robust aggregators (Krum, Geometric Median, Bulyan, MCA) depend on relative Euclidean geometry among the uploaded vectors $\{g_m\}$, not on their ambient coordinates in $\mathbb{R}^p$. That is why a random projection $P \in \mathbb{R}^{k\times p}$ with $k \ll p$ can replace the p-dimensional robust comparisons, provided the subsequent reconstruction still uses the original vectors.

A natural but insufficient requirement would be to preserve only the pairwise distances among the M uploaded vectors $\{g_m\}$ themselves. The classical Johnson-Lindenstrauss (JL) lemma provides exactly this guarantee for a fixed point set, with sketch size $k = \mathcal{O}(\epsilon^{-2}\log(M/\delta))$ (Kane & Nelson, 2014). It is not enough for PDR: the reliability weights $\tilde{\alpha}_m$ are computed *after* P is drawn, so the reconstructed update $g = \sum_m \tilde{\alpha}_m g_m$ is a data-dependent linear combination of $\{g_m\}$. Hence P must act as an approximate isometry on the entire subspace $\text{span}\{g_m\}_{m\in\mathcal{M}}$ (dimension at most M), not merely on those M points. We therefore invoke subspace embeddings, recalled next in a dense form and then in the sparse form used by PDR.

Lemma 2.2 (Subspace Embedding Theorem; dense case)**.** *Let $\mathcal{V} = \text{span}\{g_1, g_2, \ldots, g_M\}$ be a linear subspace of dimension at most M in $\mathbb{R}^p$. Let $P \in \mathbb{R}^{k\times p}$ be generated by sampling i.i.d. zero-mean sub-Gaussian entries of variance $1/k$. Given $\epsilon \in (0,1)$ and $\delta \in (0,1)$, if*

$$k \geq \frac{18}{\epsilon^2}\left(M + 2\ln\left(\frac{2}{\delta}\right)\right), \tag{5}$$

then with probability at least $1-\delta$, for every $v \in \mathcal{V}$,

$$(1-\epsilon)\left\|v\right\|_2^2 \leq \left\|Pv\right\|_2^2 \leq (1+\epsilon)\left\|v\right\|_2^2. \tag{6}$$

Dense P already yields a valid embedding, but materializing or applying it costs $\mathcal{O}(kp)$ per vector. PDR instead uses the sparse JL matrix of Kane-Nelson (Kane & Nelson, 2014), analyzed as an OSNAP oblivious subspace embedding (Nelson & Nguyên, 2013; Woodruff, 2014).

How P is generated (sparse case). For a sketch size k and column sparsity s, the server builds P *column-wise*: for each coordinate $j \in [p]$, sample s distinct rows in $\{1, \ldots, k\}$ together with independent signs $\sigma_1, \ldots, \sigma_s \in \{\pm 1\}$, and place the values $\sigma_\ell\, s^{-1/2}$ in those rows of column j (all other entries of the column are zero). Equivalently, one may store only a random seed and expand these supports on the fly. The resulting matrix has exactly s nonzeros per column, so sparse storage is $\mathcal{O}(ps)$ (or $\mathcal{O}(1)$ with a seed), never a dense $k \times p$ array of size $\mathcal{O}(kp)$. Sampling the supports costs $\mathcal{O}(ps)$; applying P to one vector costs $\mathcal{O}(ps)$.

Lemma 2.3 (Sparse subspace embedding)**.** *Let P be generated as above (Kane-Nelson sparse JL / OSNAP). Under the OSNAP analysis (Nelson & Nguyên, 2013; Woodruff, 2014), P is a $(1\pm\epsilon)$ embedding of any subspace of dimension at most M with probability at least $1-\delta$ whenever*

$$k = \tilde{\mathcal{O}}\left(\frac{M}{\epsilon^2}\right), \qquad s = \mathcal{O}\left(\frac{\text{polylog}(M, 1/\delta, 1/\epsilon)}{\epsilon}\right) \;=\; \tilde{\mathcal{O}}\left(\frac{1}{\epsilon}\right), \tag{7}$$

where $\tilde{\mathcal{O}}$ hides the stated polylog factors (alternatively, $s = \mathcal{O}_\gamma(1/\epsilon)$ with $k = \mathcal{O}(M^{1+\gamma}/\epsilon^2)$ for any constant $\gamma > 0$ (Nelson & Nguyên, 2013)). Consequently, projecting M client vectors costs

$\mathcal{O}(Mps)$ *arithmetic in total (support sampling* $\mathcal{O}(ps)$ *is absorbed). Relative to the classical fixed-set JL guarantee above, the price of the subspace guarantee is the larger sketch size* $k = \tilde{\mathcal{O}}(M/\epsilon^2)$ *rather than* $k = \mathcal{O}(\epsilon^{-2}\log(M/\delta))$.

Round-t workflow. An honest client uploads $g_m^t = \nabla F_m(w^t, \xi_m^t)$; a Byzantine client may upload an arbitrary $g_m^t = \star$. The server first collects $\{g_m^t\}_{m\in\mathcal{M}}$. Only afterward does it draw a fresh sparse P^t (new seed, independent of past rounds given the current uploads) as above, form $\tilde{g}_m^t = P^t g_m^t$, run RAgg on $\{\tilde{g}_m^t\} \subset \mathbb{R}^k$ to obtain reliability weights, and reconstruct

$$g^t = \sum_{m\in\mathcal{M}} \tilde{\alpha}_m^t g_m^t, \tag{8}$$

where $\tilde{\alpha}_m^t \geq 0$ and $\sum_m \tilde{\alpha}_m^t = 1$. Finally, $w^{t+1} = w^t - \eta^t g^t$ is broadcast. Drawing P^t after the uploads fixes $\mathcal{V}^t = \mathrm{span}\{g_m^t\}$ before sampling, so Lemmas 2.2 and 2.3 apply conditionally on the received gradients. We do not reuse one fixed P across all rounds, which would couple successive subspaces to the same sketch. Within a round, the sparse supports may be cached once ($\mathcal{O}(ps)$ memory) and shared by all M clients. Algorithm 1 summarizes one communication round.

Algorithm 1: PDR Algorithm

Input: Initial global model parameter w^0, clients set $\mathcal{M}$, number of rounds T, sketch size k, column sparsity s.
Output: Updated global model parameter w^T.
Every honest client m holds local data $\mathcal{S}_m$ for $m \in \mathcal{H}$.
for $t = 0, 1, 2, \cdots, T-1$ **do**
  for every client $m \in \mathcal{H}$ in parallel **do**
    Receive w^t. Sample mini-batch ξ_m^t from $\mathcal{S}_m$, compute $g_m^t = \nabla F_m(w^t, \xi_m^t)$, and upload g_m^t to the server.
  end for
  for every client $m \in \mathcal{B}$ in parallel **do**
    Upload an arbitrary (possibly malicious) vector $g_m^t \in \mathbb{R}^p$.
  end for
  Receive all uploaded vectors $\{g_m^t\}_{m\in\mathcal{M}}$.
  Draw a fresh sparse projection P^t (Lemma 2.3; seeded, not densely stored) *after* receiving $\{g_m^t\}$, form $\tilde{g}_m^t = P^t g_m^t$, and compute reliability weights $\{\tilde{\alpha}_m^t\}$ by running RAgg in $\mathbb{R}^k$.
  Set $g^t = \sum_{m\in\mathcal{M}} \tilde{\alpha}_m^t g_m^t$ and $w^{t+1} = w^t - \eta^t g^t$.
  Broadcast w^{t+1} to all clients.
end for
Output w^T.

Remark 2.4 (Scope and (b, c)-robustness)**.** PDR targets **vector-level distance-based** aggregators (Krum, Geometric Median, Bulyan, MCA). It does not apply to coordinate-wise rules (e.g., median or trimmed mean), which projection destroys. Definition 2.1 holds in any Euclidean space: running such a rule on $\{\tilde{g}_m^t\} \subset \mathbb{R}^k$ yields a (b, c)-RAgg w.r.t. distances in $\mathbb{R}^k$ with the same structural constant c; Lemmas 2.2 and 2.3 only produce the lift factor $\frac{1+\epsilon}{1-\epsilon}$ in Section 3.

3 Theoretical Results

In this section we first state the server complexity of the PDR wrapper, then prove convergence for this interface under non-IID data. We begin with standard assumptions.

3.1 Assumptions

The following assumptions are standard.

Assumption 3.1 (L-Smoothness)**.** The loss function $f(w, s_{m,i})$ is L-smooth, i.e., for $\forall w_1, w_2 \in \mathbb{R}^p$, it follows that

$$f(w_1, s_{m,i}) - f(w_2, s_{m,i}) \leqslant \langle \nabla f(w_2, s_{m,i}), w_1 - w_2 \rangle + \frac{L}{2} \|w_1 - w_2\|^2 . \tag{9}$$

Finite averages preserve the same L, so each F_m and the global F are L-smooth.

Assumption 3.2 (μ-Strong Convexity)**.** The loss function $f(w, s_{m,i})$ has μ-strong convexity, i.e., for $\forall w_1, w_2 \in \mathbb{R}^p$, it follows that

$$f(w_1, s_{m,i}) - f(w_2, s_{m,i}) \geq \langle \nabla f(w_2, s_{m,i}), w_1 - w_2 \rangle + \frac{\mu}{2} \|w_1 - w_2\|^2 . \tag{10}$$

Finite averages likewise preserve μ for each F_m and for F.

Assumption 3.3 (Local Unbiased Gradient)**.** For $\xi_m \subseteq \mathcal{S}_m$, the gradient of local training loss function $F_m(w, \xi_m)$ is unbiased, which implies that

$$\mathbb{E}\{\nabla F_m(w, \xi_m)\} = \nabla F_m(w), m \in \mathcal{H}. \tag{11}$$

This assumption is widely adopted in the literature (Huang et al., 2023; Wu et al., 2023; Xiao & Ji, 2023), and $\nabla F_m(w, \xi_m)$ reduces to the exact gradient $\nabla F_m(w)$ when $\xi_m = \mathcal{S}_m$.

Assumption 3.4 (Bounded Inner Error)**.** For $\forall w \in \mathbb{R}^p$, the inner error of gradients is uniformly bounded, i.e.,

$$\mathbb{E}\left\{ \|\nabla F_m(w, \xi_m) - \nabla F_m(w)\|^2 \right\} \leqslant \sigma^2, m \in \mathcal{H}. \tag{12}$$

This assumption is also assumed in Huang et al. (2023); Wu et al. (2023); Xiao & Ji (2023).

Assumption 3.5 (Bounded Data Heterogeneity)**.** We define a representation of data heterogeneity by inspecting gradient direction, i.e.,

$$\kappa_m^2 = \|\nabla F_m(w) - \nabla F(w)\|^2 , m \in \mathcal{H}. \tag{13}$$

We also assume the data heterogeneity is bounded, which implies

$$\kappa_m^2 \leqslant \kappa^2, m \in \mathcal{H}, \tag{14}$$

where κ^2 is the heterogeneity upper bound.

This assumption is also assumed in Huang et al. (2023); Wu et al. (2023); Xiao & Ji (2023). Note that it is imposed only on honest clients: nothing whatsoever is assumed about $m \in \mathcal{B}$, whose data and messages are both arbitrary. This is consistent with the honest-restricted objective (1), for which κ_m is well defined precisely for $m \in \mathcal{H}$.

3.2 Time Complexity Analysis

Server-side aggregation under PDR consists of three steps: (1) **Sparse projection**, which itself splits into sampling the supports of P^t and applying it: drawing s row indices and signs for each of the p columns costs $\mathcal{O}(ps)$ (Kane & Nelson, 2014; Nelson & Nguyên, 2013), and multiplying the resulting sparse map by M uploaded vectors costs $\mathcal{O}(Mps)$; the sampling term is absorbed by the multiply, so step (1) is $\mathcal{O}(Mps)$ overall (Lemma 2.3); (2) **Low-dimensional aggregation**, costing $\mathcal{O}(C_{\text{RAgg}}(M, k))$; (3) **High-dimensional reconstruction** $\sum_m \tilde{\alpha}_m^t g_m^t$, costing $\mathcal{O}(Mp)$.

Thus the total arithmetic complexity is

$$\mathcal{O}(\text{Total}) = \mathcal{O}\Big(Mps + C_{\text{RAgg}}(M, k)\Big). \tag{15}$$

With the OSNAP choice $s = \mathcal{O}(\text{polylog}(M, 1/\delta, 1/\epsilon)/\epsilon) = \tilde{\mathcal{O}}(1/\epsilon)$ (Nelson & Nguyên, 2013), this is $\tilde{\mathcal{O}}\big(Mp + C_{\text{RAgg}}(M, k)\big)$ (polylog factors in M absorbed in $\tilde{\mathcal{O}}$). Crucially, neither generating nor applying P costs $\mathcal{O}(Mkp)$: column sparsity s need not grow with k, which is what makes the projection compatible with the embedding size $k = \tilde{\mathcal{O}}(M/\epsilon^2)$ required by Lemma 2.3 (in particular $k = \Omega(M)$ whenever ϵ is treated as a constant).

Concrete examples (same s):

- **Krum:** $\mathcal{O}(M^2 p)$ becomes $\mathcal{O}(Mps + M^2 k)$. Whenever $k \ll p$ (typical once $M/\epsilon^2 \ll p$), the pairwise stage is substantially cheaper even though $k = \Omega(M)$.
- **Geometric Median:** With a *fixed* Weiszfeld iteration budget I (an algorithmic early-stopping parameter, identical in $\mathbb{R}^p$ and $\mathbb{R}^k$ for fair wall-clock comparison), $\mathcal{O}(I \cdot Mp)$ becomes $\mathcal{O}(Mps + I \cdot Mk)$. Sparse projection is also highly parallelizable, whereas Weiszfeld iterations are inherently sequential.

When $C_{\text{RAgg}}(M,k)$ is dominated by terms such as M^2k or IMk with $k \ll p$, the leading dependence on the model dimension reduces to reading the gradients and applying a sparse sketch, i.e., the robust stage is no longer super-linear in p. We do *not* claim a communication saving: clients still upload full p-dimensional vectors; PDR targets server arithmetic only.

3.3 Convergence Analysis

In this subsection we give the convergence analysis of the PDR wrapper for non-convex losses that satisfy Assumption 3.1. All detailed proofs are deferred to Appendix B.

Lemma 3.6. *Based on Definition 2.1, Remark 2.4, and Assumptions 3.3, 3.4, and 3.5, if P^t is a $(1 \pm \epsilon)$ subspace embedding of $\operatorname{span}\{g_m^t\}_{m\in\mathcal{M}}$ (Lemma 2.2 or 2.3), an event of probability at least $1-\delta$ over the draw of P^t, then*

$$\mathbb{E}\left\|g^t - \bar{g}^t\right\|^2 \le 2cb\frac{1+\epsilon}{1-\epsilon}(\sigma^2 + 2\kappa^2), \tag{16}$$

where c is the Euclidean-space robustness constant of the base aggregator run in $\mathbb{R}^k$, and

$$\bar{g}^t = \frac{1}{\sum_{i\in\mathcal{H}} S_i}\sum_{m\in\mathcal{H}} S_m \cdot g_m^t. \tag{17}$$

Proof. Please refer to Appendix B. □

Remark 3.7 (Theoretical Implications of Lemma 3.6)**.** Lemma 3.6 is the safety net for the acceleration in the previous subsection:

- **Controlled distortion:** Moving geometry to $\mathbb{R}^k$ scales the inherent robustness bound (cb) by $\frac{1+\epsilon}{1-\epsilon}$, tunable via k (and s for sparse embeddings). PDR does not invent new successful regimes and does not enlarge the breakdown of the wrapped rule.
- **Plug-and-play:** Abstracting the aggregator's error into cb (Remark 2.4) yields the same guarantee for any aggregator in that remark.

Theorem 3.8. *Under Assumptions 3.1, 3.3, 3.4, and 3.5, suppose $\eta^t \le \frac{1}{4L}$ for all $t \in \{0,1,\dots,T-1\}$. For $\epsilon \in (0,1)$ and global failure probability $\delta \in (0,1)$, assume each P^t is a $(1\pm\epsilon)$ subspace embedding with failure probability at most δ/T, e.g., the dense choice*

$$k \ge \frac{18}{\epsilon^2}\left(M + 2\ln\left(\frac{2T}{\delta}\right)\right) \tag{18}$$

or a sparse embedding as in Lemma 2.3. Write $F^\star = \inf_w F(w)$,

$$\Delta_\epsilon \triangleq 2cb\frac{1+\epsilon}{1-\epsilon}(\sigma^2+2\kappa^2), \qquad \omega \triangleq \sum_{m\in\mathcal{H}}\left(\frac{S_m}{\sum_{i\in\mathcal{H}} S_i}\right)^2, \tag{19}$$

where Δ_ϵ is exactly the bound of Lemma 3.6 and $\omega = \frac{1}{M-B}$ when the honest S_m are equal. Then, with probability at least $1-\delta$,

$$\frac{\sum_{t=0}^{T-1}\eta^t\mathbb{E}\left\|\nabla F(w^t)\right\|^2}{\sum_{t=0}^{T-1}\eta^t} \le \frac{4(F(w^0)-F^\star)}{\sum_{t=0}^{T-1}\eta^t} + \underbrace{2\Delta_\epsilon}_{\textit{Byzantine floor}} + \underbrace{\frac{4L\sum_{t=0}^{T-1}(\eta^t)^2}{\sum_{t=0}^{T-1}\eta^t}\left(\Delta_\epsilon + \omega\sigma^2\right)}_{\textit{vanishing sampling term}}. \tag{20}$$

Proof. Please refer to Appendix C. □

Remark 3.9. Theorem 3.8 characterizes training of the wrapper:

- **Convergence Rate:** With the constant learning rate $\eta^t = \frac{1}{4L\sqrt{T}}$, which satisfies $\eta^t \le \frac{1}{4L}$, the bound becomes

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(w^t)\|^2 \le \underbrace{\frac{16L(F(w^0)-F^\star)+\Delta_\epsilon+\omega\sigma^2}{\sqrt{T}}}_{\mathcal{O}(1/\sqrt{T})} + \underbrace{2\Delta_\epsilon}_{\text{Byzantine floor}}.$$

- **Optimality Gap:** The floor is $2\Delta_\epsilon = 4cb\frac{1+\epsilon}{1-\epsilon}(\sigma^2 + 2\kappa^2)$, which is proportional to b. Setting $b = 0$ therefore drives it to zero and recovers exactly the standard non-convex SGD rate $\mathcal{O}(1/\sqrt{T})$ with no residual error, as any correct bound must. The sole effect of the projection is the factor $\frac{1+\epsilon}{1-\epsilon}$ inside Δ_ϵ, so the wrapper's entire price is visible in one place. Sampling noise never survives in the floor: σ^2 enters only as $\omega\sigma^2$ inside the $\mathcal{O}(1/\sqrt{T})$ term, damped by the averaging factor ω, which equals $\frac{1}{M-B}$ for equal honest sample sizes.

Theorem 3.10. *Under Assumptions 3.1, 3.2, 3.3, 3.4, and 3.5, for a given error tolerance $\epsilon \in (0, 1)$ and a global failure probability $\delta \in (0, 1)$, suppose each P^t is a $(1 \pm \epsilon)$ subspace embedding with failure probability at most δ/T as in Theorem 3.8. With Δ_ϵ and ω as in (19), the following hold with probability at least $1 - \delta$ over the randomness of the projection matrices. For any learning rate $\eta^t \le \frac{1}{2L}$, the expected distance to the unique global minimum w^* satisfies the one-step recurrence*

$$\mathbb{E}\left\|w^{t+1} - w^*\right\|^2 \le \left(1 - \frac{\eta^t\mu}{2}\right)\mathbb{E}\left\|w^t - w^*\right\|^2 + \frac{2\eta^t}{\mu}\Delta_\epsilon + 2(\eta^t)^2\left(\Delta_\epsilon + \omega\sigma^2\right). \tag{21}$$

Furthermore, with the decaying learning rate $\eta^t = \frac{2}{\mu(t+\gamma)}$ and $\gamma = \frac{4L}{\mu} > 1$, the iterates generated over T communication rounds satisfy

$$\mathbb{E}\left\|w^T - w^*\right\|^2 \le \frac{\gamma - 1}{T + \gamma - 1}\left\|w^0 - w^*\right\|^2 + \frac{4\Delta_\epsilon}{\mu^2} + \frac{8\left(\Delta_\epsilon + \omega\sigma^2\right)}{\mu^2(T + \gamma - 1)}\ln\left(1 + \frac{T}{\gamma - 1}\right). \tag{22}$$

Proof. Please refer to Appendix D. □

Remark 3.11. Theorem 3.10 gives the corresponding guarantee under strong convexity:

- **Convergence Rate:** With $\eta^t = \mathcal{O}(1/t)$, the initial-condition term decays at $\mathcal{O}(1/T)$ and the sampling term at $\tilde{\mathcal{O}}(1/T)$, so the transient rate is $\tilde{\mathcal{O}}(1/T)$.
- **Optimality Gap:** The only non-vanishing term is $\frac{4\Delta_\epsilon}{\mu^2}$, again proportional to b. Setting $b = 0$ removes it entirely and leaves $\mathcal{O}(1/T) + \tilde{\mathcal{O}}\big(\omega\sigma^2/(\mu^2 T)\big)$, the classical strongly convex SGD guarantee. As in the non-convex case, the projection enters only through $\frac{1+\epsilon}{1-\epsilon}$, and σ^2 never survives in the floor.

4 Experiments

4.1 Setups

Experimental Setups: We evaluate on TinyImageNet (MobileNetV3 (Howard et al., 2019)) and CIFAR-100 (VGG16 (Simonyan & Zisserman, 2015)). Data are partitioned across $M = 50$ clients by a Dirichlet label distribution with $\beta \in \{0.2, 0.6\}$. Each run lasts 100 communication rounds with local batch size 32. For PDR we use the sparse column embedding of Lemma 2.3 with default sketch size $k = 8192$ and per-column sparsity $s = 8$ (each coordinate contributes to s randomly chosen sketch rows with signs $\pm s^{-1/2}$), generated from a random seed without materializing a dense $k \times p$ matrix. Uplink payloads remain full-dimensional: PDR accelerates server aggregation and does not compress communication.

Byzantine Scenarios: We consider four threat models: Gaussian, Sign-flip, LIE (Baruch et al., 2019), and FoE (Xie et al., 2020), with malicious ratios $b \in \{0.1, 0.3\}$. Attack definitions and remaining implementation details are in Appendix A.

Baselines: We wrap Krum (Blanchard et al., 2017), Bulyan (El-Mhamdi et al., 2018), Geometric Median (Chen et al., 2017), and MCA (Luan et al., 2025), yielding PDR+Krum, PDR+Bulyan, PDR+Geometric Median, and PDR+MCA.

Metrics: We report (i) maximum test accuracy (%) over the 100 rounds and (ii) server-side wall-clock time of one aggregation step. For baselines the time is the cost of the robust aggregator in $\mathbb{R}^p$; for PDR it includes generating P, projecting all uploads, and aggregating in $\mathbb{R}^k$. Non-convergent runs (accuracy $< 10\%$) are marked "-" for both accuracy and wall time.

Table 1: Maximum test accuracy (%) and server wall time (s) under $\beta = 0.6$ on TinyImageNet and CIFAR100. For each method, the gray row is the baseline and the white row is PDR. PDR wall time includes generating P, projection, and robust aggregation. "-" denotes non-convergence (both accuracy and wall time).

Methods	Attack	TinyImageNet				CIFAR100			
		$b = 0.1$		$b = 0.3$		$b = 0.1$		$b = 0.3$	
		Acc.	Time	Acc.	Time	Acc.	Time	Acc.	Time
Krum	Gaussian	32.32	8.7583	32.15	8.7092	33.95	8.3156	31.95	8.2352
PDR+Krum		32.22	**0.3319**	32.03	**0.3298**	33.09	**0.1421**	30.13	**0.1392**
	Sign-flip	31.92	8.7113	32.49	9.0235	34.57	8.1791	29.91	8.2151
		31.92	**0.339**	32.05	**0.34**	33.87	**0.1392**	29.78	**0.139**
	LIE	32.28	8.7	31.58	8.6914	32.43	8.1938	31.01	8.1935
		31.93	**0.3234**	32.10	**0.3206**	32.79	**0.1396**	29.84	**0.1358**
	FoE	-	-	-	-	32.40	8.2506	29.34	8.2948
		-	-	-	-	30.39	**0.1408**	29.19	**0.1389**
Bulyan	Gaussian	54.47	9.3945	50.76	9.0512	58.31	8.5972	54.34	8.4082
PDR+Bulyan		54.17	**0.39**	45.97	**0.3784**	58.20	**0.177**	50.92	**0.1676**
	Sign-flip	54.38	9.5733	48.99	9.0253	58.18	8.5215	54.74	8.3996
		54.17	**0.4042**	45.63	**0.3908**	58.15	**0.1802**	50.78	**0.1663**
	LIE	54.57	9.382	49.66	9.1335	58.24	8.4861	54.13	8.3766
		54.11	**0.3817**	45.77	**0.3723**	58.10	**0.1737**	51.58	**0.1644**
	FoE	52.48	9.4193	-	-	57.26	8.586	27.25	8.404
		51.95	**0.4008**	-	-	57.09	**0.1768**	24.96	**0.165**
Geometric Median	Gaussian	54.33	2.8878	54.32	2.9474	58.79	0.9003	58.45	0.918
PDR+Geometric Median		54.82	**0.4448**	54.07	**0.4487**	58.78	**0.2374**	58.47	**0.2279**
	Sign-flip	51.25	2.856	-	-	53.39	0.9113	21.60	1.1859
		50.78	**0.4492**	-	-	53.40	**0.2492**	22.07	**0.3109**
	LIE	54.83	2.8669	54.16	2.8961	58.73	0.8749	58.31	0.9026
		54.75	**0.4442**	54.01	**0.4476**	58.64	**0.2257**	58.54	**0.224**
	FoE	50.35	5.0288	-	-	52.50	3.4242	-	-
		50.83	**0.5605**	-	-	52.52	**0.8336**	-	-
MCA	Gaussian	54.49	1.1212	54.26	1.1583	58.76	0.4219	58.30	0.3709
PDR+MCA		54.86	**0.4242**	54.10	**0.421**	58.63	**0.2014**	58.69	**0.2025**
	Sign-flip	54.82	1.3847	-	-	58.35	0.477	-	-
		54.88	**0.4359**	-	-	58.66	**0.2427**	-	-
	LIE	55.00	1.1584	54.16	1.1538	58.67	0.3671	58.36	0.3669
		54.87	**0.4221**	54.08	**0.4205**	58.73	**0.202**	58.48	**0.2054**
	FoE	53.27	1.1742	48.81	1.3625	58.41	0.4394	56.69	0.439
		53.79	**0.4211**	48.45	**0.4344**	58.15	**0.2138**	56.91	**0.2147**

4.2 RESULTS

Table 1 reports the main comparison under moderate heterogeneity ($\beta = 0.6$) and Table 2 the full (k, s) ablation on TinyImageNet (Gaussian, $b = 0.1$, $\beta = 0.6$). The matching $\beta = 0.2$ tables and extended discussion are in Appendix E.

Large server-side speedups under an inclusive timing protocol. Across all four base aggregators and both datasets, PDR substantially reduces server wall-clock time, and the comparison is conservative because PDR times already include generating P, applying it to all M uploads, and running the robust rule in $\mathbb{R}^k$. The gain is largest for distance-heavy rules: on TinyImageNet, Krum and Bulyan drop from 8.7 to 9.6 s down to 0.32 to 0.40 s (20 to 25 times), and on CIFAR-100 Krum improves

Table 2: Ablation of sketch size k (rows) and sparsity s (columns) for PDR on TinyImageNet under Gaussian attack with $b = 0.1$ and $\beta = 0.6$. Default $(k, s) = (8192, 8)$. Each s reports Acc. and Time. Wall time includes generating P, projection, and robust aggregation.

Methods	k	$s = 8$		$s = 4$		$s = 2$		$s = 1$	
		Acc.	Time	Acc.	Time	Acc.	Time	Acc.	Time
PDR+Krum	8192	32.22	0.3319	32.25	0.2300	32.19	0.1811	32.79	0.1745
	4096	33.02	0.3237	32.92	0.2200	32.58	0.1725	32.38	0.1717
	2048	34.10	0.3207	32.87	0.2383	32.76	0.1842	32.04	0.1726
PDR+Bulyan	8192	54.17	0.3900	54.08	0.3178	54.40	0.2645	54.08	0.2615
	4096	54.11	0.4121	54.60	0.2831	54.40	0.2480	54.26	0.2596
	2048	54.09	0.4085	54.63	0.3013	54.56	0.2488	54.43	0.2598
PDR+Geometric Median	8192	54.82	0.4448	54.59	0.3721	54.65	0.3270	54.76	0.3188
	4096	54.63	0.4674	54.75	0.3415	54.68	0.2925	54.82	0.3190
	2048	54.73	0.4645	54.75	0.3503	54.74	0.2971	54.78	0.3240
PDR+MCA	8192	54.86	0.4242	54.73	0.3552	54.74	0.3071	54.82	0.3001
	4096	54.81	0.4590	54.85	0.3183	54.75	0.2825	54.78	0.2993
	2048	54.73	0.4441	54.81	0.3236	54.78	0.2715	54.86	0.2991

from about 8.2 s to 0.14 s (about 50 times). Geometric Median and MCA also become faster, with a smaller relative gain as expected from their milder baseline cost.

Accuracy tracks the wrapped aggregator. In most settings of Table 1, PDR stays close to the baseline accuracy and sometimes slightly improves it. A sparse subspace embedding therefore preserves the geometry used by vector-level distance-based rules well enough for practical training, while moving the dominant arithmetic to dimension k.

Robustness behavior is inherited, including failures. Under FoE with Krum on TinyImageNet both fail to converge, and under Sign-flip with $b = 0.3$ Geometric Median and MCA collapse with or without PDR; conversely, whenever the baseline reaches a usable accuracy PDR remains in the same regime. This matches the theory: projection scales the Byzantine floor by $\frac{1+\epsilon}{1-\epsilon}$ but does not change the structural scope of the wrapped (b, c)-RAgg.

Sketch ablations: sparsity s drives wall time; accuracy is largely flat. Over the twelve (k, s) settings of Table 2, Bulyan, Geometric Median, and MCA stay near 54% and Krum in the low-30% range, with no systematic collapse as k or s decreases. Reducing s gives the clearest timing gains (PDR+Krum from 0.33 s at $(8192, 8)$ to 0.17 to 0.23 s for $s \leq 4$), whereas shrinking k at fixed s changes inclusive time only mildly. Thus $(8192, 8)$ is a conservative default rather than a necessary one.

5 Conclusion

This paper asked how to make existing distance-based robust aggregators fast on the server when clients still upload full-dimensional gradients. PDR is a drop-in wrapper: a sparse JL / OSNAP embedding computes reliability weights in $\mathbb{R}^k$, the update is reconstructed in $\mathbb{R}^p$, and aggregation arithmetic drops to $\mathcal{O}(Mps + C_{\mathrm{RAgg}}(M, k))$. We give a convergence analysis of this exact interface, showing that standard robust-FL rates carry over with a Byzantine floor inflated only by $\frac{1+\epsilon}{1-\epsilon}$, and we observe large reductions in server wall-clock time with competitive accuracy under heterogeneous data and diverse attacks.

Reproducibility Statement

Algorithm 1 in Section 2 specifies the complete PDR procedure, including how the sparse projection P^t is generated from a random seed and when it is drawn. All assumptions are collected in Section 3.1, and full proofs of Lemma 3.6, Theorem 3.8, and Theorem 3.10 are given in Appendices B, C, and D. Appendix A documents the datasets and their non-IID partitioning, the models, the training hyperparameters, the attack definitions, the (k, s) settings, and the timing protocol; Appendix E reports the complete result and ablation tables.

References

Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems*, 32, 2019.

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

Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. FLTrust: Byzantine-robust federated learning via trust bootstrapping. In *Network and Distributed System Security Symposium*, 2021.

Xinyang Cao and Lifeng Lai. Distributed gradient descent algorithm robust to an arbitrary number of Byzantine attackers. *IEEE Transactions on Signal Processing*, 67(22):5850–5864, 2019.

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

Ron Dorfman, Shay Vargaftik, Yaniv Ben-Itzhak, and Kfir Yehuda Levy. DoCoFL: Downlink compression for cross-device federated learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 8356–8388. PMLR, 2023.

El-Mahdi El-Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in Byzantium. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 3521–3530. PMLR, 2018.

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

Yongxin Guo, Xiaoying Tang, and Tao Lin. FedBR: Improving federated learning on heterogeneous data via local learning bias reduction. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 12034–12054. PMLR, 2023.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1314–1324, 2019.

Minhui Huang, Dewei Zhang, and Kaiyi Ji. Achieving linear speedup in non-IID federated bilevel learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 14039–14059. PMLR, 2023.

Daniel M. Kane and Jelani Nelson. Sparser Johnson-Lindenstrauss transforms. *Journal of the ACM*, 61(1):1–23, 2014.

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

Jakub Konečnỳ, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

Zhirong Luan, Wenrui Li, Meiqin Liu, and Badong Chen. Robust federated learning: Maximum correntropy aggregation against Byzantine attacks. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1):62–75, 2025.

Jelani Nelson and Huy L. Nguyên. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 117–126, 2013.

Murtaza Rangwala, Farag Azzedin, Richard O. Sinnott, and Rajkumar Buyya. SketchGuard: Scaling Byzantine-robust decentralized federated learning via sketch-based screening. *arXiv preprint arXiv:2510.07922*, 2025.

Virat Shejwalkar and Amir Houmansadr. Manipulating the Byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *Network and Distributed System Security Symposium*, 2021.

Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1354–1371. IEEE, 2022.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019.

David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1–2):1–157, 2014.

Feijie Wu, Song Guo, Zhihao Qu, Shiqi He, Ziming Liu, and Jing Gao. Anchor sampling for federated learning with partial client participation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 37379–37416. PMLR, 2023.

Yue Xia, Tayyebeh Jahani-Nezhad, and Rawad Bitar. Beyond trade-offs: A unified framework for privacy, robustness, and communication efficiency in federated learning. *arXiv preprint arXiv:2508.12978*, 2025.

Peiyao Xiao and Kaiyi Ji. Communication-efficient federated hypergradient computation via aggregated iterative differentiation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 38059–38086. PMLR, 2023.

Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation. In *Proceedings of the 35th Uncertainty in Artificial Intelligence Conference*, volume 115, pp. 261–270. PMLR, 2020.

Zhixiong Yang, Arpita Gang, and Waheed U. Bajwa. Adversary-resilient distributed and decentralized statistical inference and machine learning: An overview of recent advances under the Byzantine threat model. *IEEE Signal Processing Magazine*, 37(3):146–159, 2020.

Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 5650–5659. PMLR, 2018.

Baolei Zhang, Minghong Fang, Zhuqing Liu, Biao Yi, Peizhao Zhou, Yuan Wang, Tong Li, and Zheli Liu. Practical framework for privacy-preserving and Byzantine-robust federated learning. *arXiv preprint arXiv:2512.17254*, 2025.

Puning Zhao, Fei Yu, and Zhiguo Wan. A Huber loss minimization approach to Byzantine robust federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 21806–21814, 2024.

Shiyuan Zuo, Xingrun Yan, Rongfei Fan, Han Hu, Hangguan Shan, Tony Q. S. Quek, and Puning Zhao. Federated learning resilient to Byzantine attacks and data heterogeneity. *IEEE Transactions on Mobile Computing*, 24(10):10729–10742, 2025a.

Shiyuan Zuo, Xingrun Yan, Rongfei Fan, Li Shen, Puning Zhao, Jie Xu, and Han Hu. Efficient federated learning against Byzantine attacks and data heterogeneity via aggregating normalized gradients. In *Advances in Neural Information Processing Systems*, 2025b.

AI USE STATEMENT

We used generative AI tools to assist with LaTeX formatting and copy-editing. We reviewed all AI-assisted text and take responsibility for the final content.

A EXPERIMENTAL SETUPS IN DETAIL

Datasets:

- **TinyImageNet:** 200 classes with 500 training images each (100,000 training samples) and 10,000 validation / test images; each image is a 64×64 RGB sample.
- **CIFAR-100:** 50,000 training and 10,000 test 32×32 RGB images over 100 fine-grained classes.

To emulate federated non-IID settings, we partition each dataset across $M = 50$ clients by sampling client label distributions from a Dirichlet distribution with concentration $\beta \in \{0.2, 0.6\}$.

Models: We use MobileNetV3 (Howard et al., 2019) on TinyImageNet and a CIFAR-adapted VGG16 (Simonyan & Zisserman, 2015) on CIFAR-100 (final linear layer sized for 100 classes).

Hyperparameter Settings: Unless otherwise stated, all baselines and PDR wrappers share the same training protocol: local batch size 32, 100 communication rounds, and an automatically tuned learning rate (no hand-crafted decay schedule). For PDR, the default sparse embedding uses sketch size $k = 8192$ and per-column sparsity $s = 8$, in the regime of Lemma 2.3; P is drawn from a random seed and never stored as a dense $k \times p$ matrix. We additionally ablate (k, s) with $k \in \{2048, 4096, 8192\}$ and $s \in \{1, 2, 4, 8\}$ (full grids in Tables 2 and 4) on TinyImageNet under Gaussian attack with $b = 0.1$.

Byzantine Attacks: We evaluate four representative threat models with malicious fractions $b \in \{0.1, 0.3\}$:

- **Gaussian Attack:** Byzantine clients ignore local data and upload i.i.d. noise from $\mathcal{N}(0, 90)$.
- **Sign-flip Attack:** At round t, compromised clients upload $-3 \cdot \sum_{m \in \mathcal{H}} g_m^t$, reversing the honest descent direction.
- **LIE Attack (Baruch et al., 2019):** Attackers estimate the coordinate-wise mean a and standard deviation ν of honest updates and upload $a + c\nu$. We set $c = 0.7$.
- **FoE Attack (Xie et al., 2020):** Compromised clients upload a scaled average of honest gradients, $\frac{q}{M-B} \sum_{m \in \mathcal{H}} g_m^t$ (or an analogous honest subset average), with $q = -0.1$.

Metrics:

- **Test Accuracy:** Top-1 accuracy of the global model on the held-out test set. Tables report the maximum accuracy over the 100 rounds.
- **Wall Time:** Server-side wall-clock time of one aggregation step. For baselines this is the robust aggregator in $\mathbb{R}^p$; for PDR it is the sum of (i) generating the sparse projection P, (ii) applying P to all uploaded gradients, and (iii) running the aggregator in $\mathbb{R}^k$. Client training and communication are excluded. If a run fails to converge (maximum accuracy $< 10\%$), both accuracy and wall time are reported as "-".

B PROOF OF LEMMA 3.6

Fix round t. Let $\mathcal{F}^t$ denote the sigma-algebra generated by the history up to (and including) the uploaded gradients $\{g_m^t\}_{m \in \mathcal{M}}$; in particular w^t and all past projections $\{P^\tau\}_{\tau < t}$ are $\mathcal{F}^t$-measurable. The server then draws P^t independently of $\mathcal{F}^t$. Conditionally on $\mathcal{F}^t$, the subspace $\mathcal{V}^t = \mathrm{span}\{g_m^t\}$ is fixed, so Lemmas 2.2 and 2.3 apply and yield a $(1 \pm \epsilon)$ embedding with probability at least $1 - \delta$. Let E^t denote this event.

We fix the following convention, used consistently in this appendix and in Appendices C and D. The high-probability statement is over the server's internal randomness $\{P^t\}$, which is drawn independently of the data, whereas $\mathbb{E}$ always denotes expectation over the mini-batch sampling ξ^t. This separation is what makes the two ingredients usable at once: $\mathbb{P}(E^t \mid \mathcal{F}^t) \geq 1-\delta$ holds *uniformly* over realizations of $\{g_m^t\}$, so the embedding may be assumed to succeed without biasing the distribution of ξ^t, while Assumptions 3.3-3.5 remain available for the ξ^t-expectations. All inequalities below are stated on E^t; the complementary event has probability at most δ and is handled by the union bound in Appendices C and D.

Since $g^t = \sum_{m\in\mathcal{M}} \tilde{\alpha}_m^t g_m^t$ and $\bar{g}^t$ are both linear combinations of $\{g_m^t\}$, the difference $g^t - \bar{g}^t$ lies in $\mathcal{V}^t$. On E^t the embedding therefore applies to it, giving the pathwise inequality

$$(1-\epsilon)\left\|g^t - \bar{g}^t\right\|^2 \leq \left\|P^t(g^t - \bar{g}^t)\right\|^2 = \left\|P^t g^t - P^t \bar{g}^t\right\|^2. \tag{23}$$

This is the only step in which E^t is used.

Let $\tilde{g}_m^t = P^t g_m^t$. By linearity of P^t and the definition of the reliability weights in (8) and (17),

$$P^t g^t = \sum_{m\in\mathcal{M}} \tilde{\alpha}_m^t \tilde{g}_m^t = \mathrm{RAgg}(\tilde{g}_1^t, \ldots, \tilde{g}_M^t), \qquad P^t \bar{g}^t = \frac{1}{\sum_{i\in\mathcal{H}} S_i} \sum_{m\in\mathcal{H}} S_m \tilde{g}_m^t, \tag{24}$$

both of which are exact identities rather than inequalities: the projection commutes with the reconstruction precisely because the weights are scalars.

Taking expectations over ξ^t in (23) and substituting (24), the right-hand side is exactly the aggregation error of the base rule acting on the ensemble $\{\tilde{g}_m^t\} \subset \mathbb{R}^k$, so Definition 2.1 applied in $\mathbb{R}^k$ (Remark 2.4) gives

$$\begin{aligned}
&(1-\epsilon)\mathbb{E}\left\|g^t - \bar{g}^t\right\|^2 \\
&\leq \mathbb{E}\left\| \sum_{m\in\mathcal{M}} \tilde{\alpha}_m^t \tilde{g}_m^t - \frac{1}{\sum_{i\in\mathcal{H}} S_i} \sum_{m\in\mathcal{H}} S_m \cdot \tilde{g}_m^t \right\|^2 \\
&\leq cb \max_{i,j\in\mathcal{H}} \mathbb{E}\left\|\tilde{g}_i^t - \tilde{g}_j^t\right\|^2.
\end{aligned} \tag{25}$$

The dispersion appearing here is a maximum of expectations, matching Definition 2.1 literally, so no interchange of $\max$ with $\mathbb{E}$ is involved. Applying Lemma 2.2 to the honest differences $g_i^t - g_j^t \in \mathcal{V}^t$, we have:

$$\begin{aligned}
&\max_{i,j\in\mathcal{H}} \mathbb{E}\left\|\tilde{g}_i^t - \tilde{g}_j^t\right\|^2 \\
&\leq \max_{i,j\in\mathcal{H}} (1+\epsilon) \cdot \mathbb{E}\left\|g_i^t - g_j^t\right\|^2 \\
&= \max_{i,j\in\mathcal{H}} (1+\epsilon) \cdot \mathbb{E}\left\|\nabla F_i(w^t, \xi_i^t) - \nabla F_j(w^t, \xi_j^t)\right\|^2 \\
&= \max_{i,j\in\mathcal{H}} (1+\epsilon) \cdot \mathbb{E}\left\|\left(\nabla F_i(w^t, \xi_i^t) - \nabla F_i(w^t)\right) - \left(\nabla F_j(w^t, \xi_j^t) - \nabla F_j(w^t)\right) + \left(\nabla F_i(w^t) - \nabla F_j(w^t)\right)\right\|^2.
\end{aligned} \tag{26}$$

Due to the independence of the data sampling across different clients and Assumption 3.3, the cross-terms in the expectation evaluate to zero. Thus, we have:

$$\begin{aligned}
&\max_{i,j\in\mathcal{H}} \mathbb{E}\left\|\tilde{g}_i^t - \tilde{g}_j^t\right\|^2 \\
&\leq \max_{i,j\in\mathcal{H}} (1+\epsilon) \cdot \left\{\mathbb{E}\left\|\nabla F_i(w^t, \xi_i^t) - \nabla F_i(w^t)\right\|^2 + \mathbb{E}\left\|\nabla F_j(w^t, \xi_j^t) - \nabla F_j(w^t)\right\|^2 + \left\|\nabla F_i(w^t) - \nabla F_j(w^t)\right\|^2\right\}.
\end{aligned} \tag{27}$$

Applying Assumption 3.4, we obtain:

$$\begin{aligned}
&\max_{i,j\in\mathcal{H}} \mathbb{E}\left\|\tilde{g}_i^t - \tilde{g}_j^t\right\|^2 \\
&\leq \max_{i,j\in\mathcal{H}} (1+\epsilon) \cdot \left(2\sigma^2 + \left\|\nabla F_i(w^t) - \nabla F_j(w^t)\right\|^2\right) \\
&= \max_{i,j\in\mathcal{H}} (1+\epsilon) \cdot \left(2\sigma^2 + \left\|\left(\nabla F_i(w^t) - \nabla F(w^t)\right) - \left(\nabla F_j(w^t) - \nabla F(w^t)\right)\right\|^2\right).
\end{aligned} \tag{28}$$

With Assumption 3.5 and the inequality $\|x-y\|^2 \le 2\|x\|^2 + 2\|y\|^2$, we can bound the terms as follows:

$$\max_{i,j\in\mathcal{H}} \mathbb{E}\left\|\tilde{g}_i^t - \tilde{g}_j^t\right\|^2 \le (1+\epsilon)(2\sigma^2 + 4\kappa^2). \tag{29}$$

Finally, substituting (29) back into (25), we have:

$$\begin{aligned}
&(1-\epsilon)\mathbb{E}\left\|g^t - \bar{g}^t\right\|^2 \\
&\le cb \max_{i,j\in\mathcal{H}} \mathbb{E}\left\|\tilde{g}_i^t - \tilde{g}_j^t\right\|^2 \\
&\le cb(1+\epsilon)(2\sigma^2 + 4\kappa^2).
\end{aligned} \tag{30}$$

Dividing both sides by $(1-\epsilon)$, we arrive at the final bound:

$$\mathbb{E}\left\|g^t - \bar{g}^t\right\|^2 \le 2cb\frac{1+\epsilon}{1-\epsilon}(\sigma^2 + 2\kappa^2). \tag{31}$$

Apart from the single use of E^t in (23), every step above is either an exact identity or an elementary inequality, and the aggregator enters only through Definition 2.1. This completes the proof of Lemma 3.6.

C Proof of Theorem 3.8

To establish a global convergence guarantee over T rounds, apply a union bound over rounds. Require that, conditionally on $\mathcal{F}^t$, the freshly drawn P^t is a $(1\pm\epsilon)$ embedding of $\mathcal{V}^t$ with probability at least $1-\delta/T$ (Lemma 2.2 or 2.3). A union bound then yields probability at least $1-\delta$ that every round succeeds simultaneously. No sample splitting is required: dependence of g_m^t on past projections is absorbed into $\mathcal{F}^t$, after which P^t is drawn independently.

Conditioned on this high-probability event, the following inequalities hold for all $t \in \{0, 1, \ldots, T-1\}$.

With Assumption 3.1, we have

$$\begin{aligned}
&F(w^{t+1}) - F(w^t) \\
&\le \langle \nabla F(w^t), w^{t+1} - w^t\rangle + \frac{L}{2}\left\|w^{t+1} - w^t\right\|^2 \\
&= -\langle \nabla F(w^t), \eta^t g^t\rangle + \frac{L}{2}\left\|\eta^t g^t\right\|^2
\end{aligned} \tag{32}$$

Taking the expectation of both sides conditioned on w^t, we have:

$$\mathbb{E}\left\{F(w^{t+1}) - F(w^t)\right\} \le -\eta^t \mathbb{E}\langle \nabla F(w^t), g^t\rangle + \frac{L(\eta^t)^2}{2}\mathbb{E}\left\|g^t\right\|^2. \tag{33}$$

The decisive step is *how* the aggregated update is decomposed. Write

$$g^t = \bar{g}^t + e^t, \qquad e^t \triangleq g^t - \bar{g}^t, \tag{34}$$

which separates the *biased* part e^t, produced by the robust aggregator acting in $\mathbb{R}^k$, from the *unbiased* part $\bar{g}^t$. These two parts must be treated differently: e^t is a genuine bias and can only be absorbed into the first-order term, whereas $\bar{g}^t$ is unbiased and should be routed through the second-order term $\frac{L(\eta^t)^2}{2}\mathbb{E}\|g^t\|^2$, where it retains a factor η^t. Bounding the whole of $g^t - \nabla F(w^t)$ as if it were bias would strip that factor from the sampling noise and leave a spurious residual term that neither decays in T nor shrinks with b.

Because (1) is the honest-restricted objective and the weights $\frac{S_m}{\sum_{i\in\mathcal{H}} S_i}$ sum to one, Assumption 3.3 gives exactly

$$\mathbb{E}\left\{\bar{g}^t\right\} = \sum_{m\in\mathcal{H}} \frac{S_m}{\sum_{i\in\mathcal{H}} S_i}\nabla F_m(w^t) = \nabla F(w^t), \tag{35}$$

with *no* residual bias term, since no expectation is taken over $m \in \mathcal{B}$. Moreover, as the honest mini-batches ξ_m^t are drawn independently across clients, the cross-terms vanish and Assumption 3.4 yields

$$\mathbb{E}\left\|\bar{g}^t - \nabla F(w^t)\right\|^2 = \sum_{m\in\mathcal{H}} \left(\frac{S_m}{\sum_{i\in\mathcal{H}} S_i}\right)^2 \mathbb{E}\left\|\nabla F_m(w^t,\xi_m^t) - \nabla F_m(w^t)\right\|^2 \le \omega\sigma^2, \tag{36}$$

with ω as in (19). Note that (36) retains the averaging gain ω, which equals $\frac{1}{M-B}$ for equal honest sample sizes; replacing this identity by Jensen's inequality would discard that gain and inflate the bound by a factor $M-B$.

For the first-order term, substituting (34) and using (35),

$$-\eta^t \mathbb{E}\langle \nabla F(w^t), g^t\rangle = -\eta^t \left\|\nabla F(w^t)\right\|^2 - \eta^t \langle \nabla F(w^t), \mathbb{E}e^t\rangle. \tag{37}$$

Applying Young's inequality $-\langle x,y\rangle \le \frac{1}{2}\|x\|^2 + \frac{1}{2}\|y\|^2$ to the residual, then Jensen's inequality $\|\mathbb{E}e^t\|^2 \le \mathbb{E}\|e^t\|^2$ and Lemma 3.6,

$$-\eta^t \mathbb{E}\langle \nabla F(w^t), g^t\rangle \le -\frac{\eta^t}{2}\left\|\nabla F(w^t)\right\|^2 + \frac{\eta^t}{2}\Delta_\epsilon. \tag{38}$$

For the second-order term, applying $\|x+y\|^2 \le 2\|x\|^2 + 2\|y\|^2$ to (34), then (35), (36) and Lemma 3.6,

$$\begin{aligned}\frac{L(\eta^t)^2}{2}\mathbb{E}\left\|g^t\right\|^2 &\le L(\eta^t)^2\left(\mathbb{E}\left\|\bar{g}^t\right\|^2 + \mathbb{E}\left\|e^t\right\|^2\right)\\ &= L(\eta^t)^2\left(\left\|\nabla F(w^t)\right\|^2 + \mathbb{E}\left\|\bar{g}^t - \nabla F(w^t)\right\|^2 + \mathbb{E}\left\|e^t\right\|^2\right)\\ &\le L(\eta^t)^2\left\|\nabla F(w^t)\right\|^2 + L(\eta^t)^2\left(\omega\sigma^2 + \Delta_\epsilon\right).\end{aligned} \tag{39}$$

This is where the sampling noise lands, and it lands carrying the factor $(\eta^t)^2$.

Substituting (38) and (39) into (33), and using $\eta^t \le \frac{1}{4L}$ so that $L(\eta^t)^2\left\|\nabla F(w^t)\right\|^2 \le \frac{\eta^t}{4}\left\|\nabla F(w^t)\right\|^2$,

$$\mathbb{E}\left\{F(w^{t+1}) - F(w^t)\right\} \le -\frac{\eta^t}{4}\left\|\nabla F(w^t)\right\|^2 + \frac{\eta^t}{2}\Delta_\epsilon + L(\eta^t)^2\left(\omega\sigma^2 + \Delta_\epsilon\right). \tag{40}$$

Rearranging (40) to isolate the gradient norm and multiplying by 4,

$$\eta^t \mathbb{E}\left\|\nabla F(w^t)\right\|^2 \le 4\,\mathbb{E}\left\{F(w^t) - F(w^{t+1})\right\} + 2\eta^t\Delta_\epsilon + 4L(\eta^t)^2\left(\omega\sigma^2 + \Delta_\epsilon\right). \tag{41}$$

Summing over $t \in \{0,1,\dots,T-1\}$, telescoping the function values, using $F(w^T) \ge F^\star$, and dividing by $\sum_{t=0}^{T-1}\eta^t$,

$$\frac{\sum_{t=0}^{T-1}\eta^t \mathbb{E}\left\|\nabla F(w^t)\right\|^2}{\sum_{t=0}^{T-1}\eta^t} \le \frac{4\left(F(w^0) - F^\star\right)}{\sum_{t=0}^{T-1}\eta^t} + 2\Delta_\epsilon + \frac{4L\sum_{t=0}^{T-1}(\eta^t)^2}{\sum_{t=0}^{T-1}\eta^t}\left(\omega\sigma^2 + \Delta_\epsilon\right), \tag{42}$$

which is the claimed bound. Specializing to $\eta^t = \frac{1}{4L\sqrt{T}}$ gives $\sum_t \eta^t = \frac{\sqrt{T}}{4L}$ and $\frac{\sum_t(\eta^t)^2}{\sum_t \eta^t} = \frac{1}{4L\sqrt{T}}$, hence

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla F(w^t)\right\|^2 \le \frac{16L\left(F(w^0) - F^\star\right) + \omega\sigma^2 + \Delta_\epsilon}{\sqrt{T}} + 2\Delta_\epsilon. \tag{43}$$

Two structural checks confirm the bound. First, every appearance of σ^2 that is not already multiplied by cb carries the factor $\frac{1}{\sqrt{T}}$ and the averaging gain ω, so it vanishes as T or $M-B$ grows. Second, setting $b=0$ makes $\Delta_\epsilon = 0$ and collapses the bound to $\mathcal{O}(1/\sqrt{T})$ with no residual floor, recovering the standard guarantee for smooth non-convex SGD; setting in addition $\epsilon = 0$ removes the wrapper entirely. The projection therefore contributes exactly one factor, $\frac{1+\epsilon}{1-\epsilon}$, carried inside Δ_ϵ. This completes the convergence proof of Theorem 3.8.

D Proof of Theorem 3.10

To establish a global convergence guarantee over T communication rounds, we apply a union bound over rounds. Require that, conditionally on $\mathcal{F}^t$, the freshly drawn P^t is a $(1 \pm \epsilon)$ embedding of $\mathcal{V}^t$ with probability at least $1 - \delta/T$ (Lemma 2.2 or 2.3). For the dense construction this means

$$k \geq \frac{18}{\epsilon^2}\left(M + 2\ln\left(\frac{2T}{\delta}\right)\right); \tag{44}$$

for the sparse OSNAP / Kane-Nelson construction it means choosing (k, s) as in Lemma 2.3 with failure probability δ/T per round. A union bound then yields probability at least $1 - \delta$ that every round succeeds simultaneously. No sample splitting is required: dependence of g_m^t on past projections is absorbed into $\mathcal{F}^t$, after which P^t is drawn independently.

Conditioned on this global high-probability event, the following holds for all $t \in \{0, 1, \ldots, T-1\}$. As in Appendix C we keep the biased and unbiased parts of the update separate via the decomposition (34), $g^t = \bar{g}^t + e^t$, together with the two identities established there: $\mathbb{E}\{\bar{g}^t\} = \nabla F(w^t)$ from (35) and $\mathbb{E}\|\bar{g}^t - \nabla F(w^t)\|^2 \leq \omega\sigma^2$ from (36), and Lemma 3.6 in the form $\mathbb{E}\|e^t\|^2 \leq \Delta_\epsilon$. Only e^t is a bias and may be charged to the first-order term; $\bar{g}^t$ is unbiased and is charged to the $(\eta^t)^2$ term.

We evaluate the expected distance to the global optimum w^*. Using $w^{t+1} = w^t - \eta^t g^t$:

$$\left\|w^{t+1} - w^*\right\|^2 = \left\|w^t - w^*\right\|^2 - 2\eta^t\langle g^t, w^t - w^*\rangle + (\eta^t)^2\left\|g^t\right\|^2. \tag{45}$$

Taking the expectation over the stochastic data sampling conditioned on w^t and using $\mathbb{E}\{g^t\} = \nabla F(w^t) + \mathbb{E}\{e^t\}$, which follows from (34) and (35), the inner product splits into a true-gradient part and a pure-bias part:

$$\begin{aligned}\mathbb{E}\left\|w^{t+1} - w^*\right\|^2 = \left\|w^t - w^*\right\|^2 &- 2\eta^t\langle\nabla F(w^t), w^t - w^*\rangle \\ &- 2\eta^t\langle\mathbb{E}\{e^t\}, w^t - w^*\rangle + (\eta^t)^2\mathbb{E}\left\|g^t\right\|^2.\end{aligned} \tag{46}$$

Note that the sampling noise of $\bar{g}^t$ has already disappeared from the first-order terms, because it is zero-mean; this is precisely the step that keeps σ^2 out of the eventual error floor.

We bound the three terms on the right-hand side separately. First, from the μ-strong convexity (Assumption 3.2), the true gradient inner product is bounded by:

$$-2\eta^t\langle\nabla F(w^t), w^t - w^*\rangle \leq -2\eta^t\left(F(w^t) - F(w^*)\right) - \eta^t\mu\left\|w^t - w^*\right\|^2. \tag{47}$$

Second, applying the Peter-Paul inequality ($-2\langle a, b\rangle \leq \theta\|b\|^2 + \frac{1}{\theta}\|a\|^2$) with $\theta = \frac{\mu}{2}$ to the bias term, followed by Jensen's inequality $\|\mathbb{E}\{e^t\}\|^2 \leq \mathbb{E}\|e^t\|^2$ and Lemma 3.6:

$$-2\eta^t\langle\mathbb{E}\{e^t\}, w^t - w^*\rangle \leq \frac{\eta^t\mu}{2}\left\|w^t - w^*\right\|^2 + \frac{2\eta^t}{\mu}\Delta_\epsilon. \tag{48}$$

Only the aggregation bias Δ_ϵ, and not σ^2, is amplified by the factor $\frac{1}{\mu}$ here.

Third, for the gradient norm squared, we use the decomposition (34) with (35), (36), Lemma 3.6, and the L-smoothness bound $\|\nabla F(w^t)\|^2 \leq 2L(F(w^t) - F(w^*))$:

$$\begin{aligned}(\eta^t)^2\mathbb{E}\left\|g^t\right\|^2 &\leq 2(\eta^t)^2\mathbb{E}\left\|\bar{g}^t\right\|^2 + 2(\eta^t)^2\mathbb{E}\left\|e^t\right\|^2 \\ &= 2(\eta^t)^2\left(\left\|\nabla F(w^t)\right\|^2 + \mathbb{E}\left\|\bar{g}^t - \nabla F(w^t)\right\|^2\right) + 2(\eta^t)^2\mathbb{E}\left\|e^t\right\|^2 \\ &\leq 4(\eta^t)^2 L\left(F(w^t) - F(w^*)\right) + 2(\eta^t)^2\left(\omega\sigma^2 + \Delta_\epsilon\right).\end{aligned} \tag{49}$$

Substituting (47), (48), and (49) back into (46), we obtain:

$$\begin{aligned}\mathbb{E}\left\|w^{t+1} - w^*\right\|^2 \leq \left(1 - \eta^t\mu + \frac{\eta^t\mu}{2}\right)\left\|w^t - w^*\right\|^2 &- 2\eta^t(1 - 2\eta^t L)\left(F(w^t) - F(w^*)\right) \\ &+ \frac{2\eta^t}{\mu}\Delta_\epsilon + 2(\eta^t)^2\left(\omega\sigma^2 + \Delta_\epsilon\right).\end{aligned} \tag{50}$$

We set the decaying learning rate as $\eta^t = \frac{2}{\mu(t+\gamma)}$ with $\gamma = \frac{4L}{\mu}$. Since $t \geq 0$, it strictly holds that $\eta^t \leq \eta^0 = \frac{2}{\mu(4L/\mu)} = \frac{1}{2L}$, which guarantees $1 - 2\eta^t L \geq 0$ and allows us to drop the non-positive function value term. Note also that $\gamma = \frac{4L}{\mu} \geq 4 > 1$ because $\mu \leq L$. Writing $a_t \triangleq \mathbb{E}\|w^t - w^*\|^2$ and using $1 - \frac{\eta^t \mu}{2} = \frac{t+\gamma-1}{t+\gamma}$, the recurrence becomes

$$a_{t+1} \leq \frac{t+\gamma-1}{t+\gamma}\, a_t + \frac{4\Delta_\epsilon}{\mu^2(t+\gamma)} + \frac{8\left(\omega\sigma^2 + \Delta_\epsilon\right)}{\mu^2(t+\gamma)^2}. \tag{51}$$

The two residual terms now decay at different rates in t, so the exact-telescoping trick used for a single $\mathcal{O}(\eta^t)$ residual no longer applies. Instead we rescale. Multiplying (51) by $(t+\gamma)$ and setting $u_t \triangleq (t+\gamma-1)a_t$, the recurrence linearizes:

$$u_{t+1} = (t+\gamma)a_{t+1} \leq u_t + \frac{4\Delta_\epsilon}{\mu^2} + \frac{8\left(\omega\sigma^2 + \Delta_\epsilon\right)}{\mu^2(t+\gamma)}. \tag{52}$$

Summing over $t = 0, 1, \ldots, T-1$ telescopes the left-hand side:

$$(T+\gamma-1)a_T \leq (\gamma-1)a_0 + \frac{4\Delta_\epsilon T}{\mu^2} + \frac{8\left(\omega\sigma^2 + \Delta_\epsilon\right)}{\mu^2} \sum_{t=0}^{T-1} \frac{1}{t+\gamma}. \tag{53}$$

Bounding the harmonic sum by an integral, $\sum_{t=0}^{T-1} \frac{1}{t+\gamma} \leq \int_{-1}^{T-1} \frac{\mathrm{d}\tau}{\tau+\gamma} = \ln\left(1 + \frac{T}{\gamma-1}\right)$, and dividing by $T+\gamma-1$,

$$a_T \leq \frac{\gamma-1}{T+\gamma-1} a_0 + \frac{4\Delta_\epsilon}{\mu^2} \cdot \frac{T}{T+\gamma-1} + \frac{8\left(\omega\sigma^2 + \Delta_\epsilon\right)}{\mu^2(T+\gamma-1)} \ln\left(1 + \frac{T}{\gamma-1}\right). \tag{54}$$

Finally, using $\frac{T}{T+\gamma-1} \leq 1$ and $a_0 = \|w^0 - w^*\|^2$, we obtain the claimed bound

$$\mathbb{E}\left\|w^T - w^*\right\|^2 \leq \frac{\gamma-1}{T+\gamma-1}\left\|w^0 - w^*\right\|^2 + \frac{4\Delta_\epsilon}{\mu^2} + \frac{8\left(\omega\sigma^2 + \Delta_\epsilon\right)}{\mu^2(T+\gamma-1)} \ln\left(1 + \frac{T}{\gamma-1}\right). \tag{55}$$

The same two checks apply as in Appendix C. The only term that does not vanish as $T \to \infty$ is $\frac{4\Delta_\epsilon}{\mu^2}$, which is proportional to cb and therefore disappears when $b = 0$; the sampling variance $\omega\sigma^2$ appears only in the third term, which decays at $\tilde{\mathcal{O}}(1/T)$. Setting $b = 0$ thus leaves $\mathcal{O}(1/T) + \tilde{\mathcal{O}}\big(\omega\sigma^2/(\mu^2 T)\big)$, the classical strongly convex SGD guarantee, so the bound degenerates correctly. This completes the proof.

E Results in Detail

This appendix reports the results for the stronger non-IID level that the main text does not have room for. Tables 1 and 2 in Section 4 cover $\beta = 0.6$; Tables 3 and 4 below are their $\beta = 0.2$ counterparts, in the same layout. For PDR, wall time is the server-side wall-clock cost of one aggregation step and includes generating the sparse projection P, applying P to all uploaded gradients, and running the base robust aggregator in $\mathbb{R}^k$. As in the main text, "-" denotes non-convergence and omits both accuracy and wall time.

Table 3 expands the main-text comparison to the stronger non-IID level. We emphasize four points that complement Section 4.

Server-side speedups under the inclusive timing protocol. Whenever a run converges, PDR substantially reduces wall-clock time relative to the wrapped aggregator. Reported PDR times already include generating P, projecting all M uploads, and running the base rule in $\mathbb{R}^k$, so the comparison is conservative. The gain is largest for distance-heavy rules (Krum / Bulyan), whose pairwise stage moves from ambient dimension p to sketch size k, while Geometric Median and MCA also become faster. We omit wall time for non-convergent runs ("-"), since timing is not meaningful when training collapses.

Table 3: Maximum test accuracy (%) and server wall time (s) under $\beta = 0.2$ on TinyImageNet and CIFAR-100. For each method, the gray row is the baseline and the white row is PDR. PDR wall time includes generating P, projection, and robust aggregation. "-" denotes non-convergence (both accuracy and wall time).

Methods	Attack	TinyImageNet				CIFAR-100			
		$b = 0.1$		$b = 0.3$		$b = 0.1$		$b = 0.3$	
		Acc.	Time	Acc.	Time	Acc.	Time	Acc.	Time
Krum	Gaussian	26.77	9.3925	26.29	8.4992	24.31	8.2309	23.31	8.0115
PDR+Krum		27.38	**0.3281**	26.60	**0.3271**	25.56	**0.1384**	29.78	**0.1381**
	Sign-flip	27.05	8.7762	26.23	8.62	30.04	7.9797	29.18	7.9817
		25.94	**0.3368**	26.93	**0.3373**	30.75	**0.1379**	29.49	**0.1367**
	LIE	26.51	8.5075	25.53	8.5014	25.64	7.9686	29.58	8.6707
		27.33	**0.3197**	26.64	**0.3237**	31.01	**0.1396**	28.47	**0.1351**
	FoE	-	-	-	-	23.29	8.0353	23.34	8.0222
		-	-	-	-	23.17	**0.1424**	23.08	**0.1373**
Bulyan	Gaussian	53.46	9.2169	45.06	8.871	57.88	8.3849	51.09	8.1728
PDR+Bulyan		52.59	**0.3867**	36.62	**0.3768**	57.81	**0.1761**	43.33	**0.1641**
	Sign-flip	53.69	9.5493	45.41	8.8384	58.22	8.2949	50.87	8.1393
		52.62	**0.4012**	36.62	**0.3899**	57.75	**0.1741**	43.62	**0.1641**
	LIE	54.09	9.5552	43.11	8.8412	57.87	9.0898	50.04	8.1654
		52.42	**0.3861**	36.74	**0.3678**	57.76	**0.1797**	43.29	**0.161**
	FoE	51.09	9.2176	-	-	57.18	8.3615	24.41	8.1008
		48.45	**0.3981**	-	-	55.56	**0.1882**	20.24	**0.1611**
Geometric Median	Gaussian	54.26	3.1229	53.20	3.2103	58.30	0.9768	57.73	0.9659
PDR+Geometric Median		54.30	**0.4454**	52.28	**0.4453**	58.40	**0.223**	57.63	**0.2363**
	Sign-flip	47.24	3.074	-	-	50.74	0.9866	21.86	1.2535
		46.56	**0.4459**	-	-	50.06	**0.2219**	20.08	**0.3009**
	LIE	54.86	3.0845	52.51	3.2097	58.50	0.9742	57.53	0.9583
		54.23	**0.4454**	52.40	**0.4461**	58.38	**0.2415**	57.57	**0.2329**
	FoE	47.21	6.1518	-	-	48.05	4.8993	-	-
		46.91	**0.6057**	-	-	47.37	**1.192**	-	-
MCA	Gaussian	54.35	1.1732	53.47	1.1617	58.52	0.4234	57.69	0.3733
PDR+MCA		54.34	**0.4287**	52.40	**0.4199**	58.20	**0.1994**	57.55	**0.2008**
	Sign-flip	54.23	1.4947	-	-	57.47	0.5364	-	-
		53.98	**0.4388**	-	-	57.46	**0.2311**	-	-
	LIE	54.85	1.1558	52.52	1.1646	58.81	0.3745	57.45	0.3693
		54.44	**0.4236**	52.35	**0.4201**	58.35	**0.2302**	57.54	**0.2005**
	FoE	53.12	1.2468	46.77	1.2838	57.96	0.443	56.31	0.4352
		53.06	**0.4257**	44.77	**0.4232**	57.88	**0.2322**	55.82	**0.2094**

Table 4: Ablation of sketch size k (rows) and sparsity s (columns) for PDR on TinyImageNet under Gaussian attack with $b = 0.1$ and $\beta = 0.2$. Default $(k, s) = (8192, 8)$. Each s reports Acc. and Time. Wall time includes generating P, projection, and robust aggregation.

Methods	k	$s = 8$		$s = 4$		$s = 2$		$s = 1$	
		Acc.	Time	Acc.	Time	Acc.	Time	Acc.	Time
PDR+Krum	8192	27.38	0.3281	26.71	0.2316	25.63	0.1841	26.91	0.1761
	4096	27.00	0.3267	25.98	0.2252	26.50	0.1761	26.29	0.1736
	2048	26.92	0.3275	26.40	0.2374	26.81	0.1843	22.25	0.1744
PDR+Bulyan	8192	52.59	0.3867	53.22	0.3324	52.95	0.2650	52.34	0.2793
	4096	52.66	0.4191	52.92	0.2921	52.91	0.2537	52.48	0.2777
	2048	52.94	0.4131	53.17	0.3040	52.83	0.2367	52.47	0.2573
PDR+Geometric Median	8192	54.30	0.4454	54.57	0.3954	54.57	0.3478	54.21	0.3403
	4096	54.05	0.4815	54.20	0.3519	54.07	0.3024	54.19	0.3383
	2048	54.01	0.4788	54.00	0.3548	54.07	0.3041	54.29	0.3336
PDR+MCA	8192	54.34	0.4287	54.61	0.3711	54.62	0.3159	54.19	0.3181
	4096	54.14	0.4570	54.13	0.3244	54.13	0.2796	54.25	0.3142
	2048	54.21	0.4543	54.07	0.3274	54.14	0.2730	54.17	0.3067

Accuracy tracks the wrapped aggregator. Across TinyImageNet and CIFAR-100, PDR typically stays close to the baseline accuracy and occasionally slightly improves it. The sparse subspace embedding preserves the geometry used by vector-level distance-based rules well enough for practical training.

Stronger non-IID ($\beta = 0.2$) does not change the qualitative message. Comparing Table 1 ($\beta = 0.6$) with Table 3 ($\beta = 0.2$), absolute accuracies generally decrease as heterogeneity increases, as expected. PDR continues to track the baseline accuracy trends and retains large speedups in convergent regimes.

Robustness is inherited, including failure regimes. At $b = 0.3$, aggressive attacks such as Sign-flip and FoE expose the breakdown points of some aggregators (e.g., Geometric Median / MCA under Sign-flip, and several FoE settings). PDR neither invents new successful regimes where the base rule fails nor systematically breaks regimes where the base rule works. This matches the theory of the wrapper: subspace embedding scales the Byzantine floor by $\frac{1+\epsilon}{1-\epsilon}$ but does not enlarge the structural scope of the wrapped (b, c)-RAgg.

Ablation tables

Sensitivity to (k, s): s dominates time; accuracy is largely flat. Table 4 presents the same full $k \times s$ grid as Table 2, now at $\beta = 0.2$. Inclusive wall time is primarily controlled by per-column sparsity s: moving from $s = 8$ to $s \leq 4$ cuts PDR+Krum from about 0.33 s to 0.17 to 0.23 s, with analogous reductions for Bulyan, Geometric Median, and MCA, whereas reducing k at fixed $s = 8$ changes time only mildly. Under moderate heterogeneity ($\beta = 0.6$), accuracy is largely flat over the twelve settings: Bulyan / Geometric Median / MCA stay near 54%, and Krum remains in the low-30% range, so the default $(8192, 8)$ is conservative rather than uniquely necessary. At $\beta = 0.2$, Bulyan, Geometric Median, and MCA remain close to their default accuracies, and moderate reductions of s (and optionally k) still offer a favorable accuracy-time trade-off.