BEYOND ZERO-PADDING: NESTED-SVD AGGREGATION FOR HETEROGENEOUS-RANK FEDERATED LORA

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated fine-tuning with LoRA typically assumes all clients share one adapter rank, even though heterogeneous devices may need different ranks to meet local memory and compute constraints. The common remedy, zero-padding smaller LoRA factors to a shared maximum rank, preserves a uniform parameter shape but communicates the update in a larger shape than it needs. We propose **NSVD-Agg**, a nested-SVD aggregation method that instead projects each client's native-rank update onto a shared reference subspace of dimension $r_{\max}$. The subspace is estimated from the raw client updates and refreshed across rounds, while the current basis stays fixed during aggregation – a temporal ordering required for the non-convex convergence guarantee we prove. On Llama-3.2-3B with simulated heterogeneous ranks $\{4, 8, 16, 32\}$, NSVD-Agg achieves lower final loss than zero-padding in all seven random seeds while reducing client-to-server LoRA communication by an analytical 53.1%; the comparison replicates on Qwen2.5-3B, where the improvement is larger and more consistent. We further introduce an untracked exact-product baseline that shares NSVD-Agg's communication savings but, by construction, lacks its guarantee; across seven seeds on both models it matches or slightly exceeds NSVD-Agg's loss, indicating the gain over zero-padding comes primarily from exact-product aggregation rather than from tracking itself. Tracking's role is therefore to supply a proven convergence guarantee at a small, measured empirical cost. NSVD-Agg is thus a communication-efficient, theoretically grounded aggregation scheme whose empirical advantage and theoretical guarantee arise from distinct, separately-verified mechanisms.

1 INTRODUCTION

Federated fine-tuning with low-rank adaptation (LoRA) enables parameter-efficient personalization while reducing the amount of model state that must be optimized and communicated. However, federated clients may have substantially different memory and compute budgets, making a single LoRA rank impractical: resource-constrained clients may require smaller adapters, whereas more capable clients can benefit from larger ranks. This creates a heterogeneous-rank aggregation problem in which clients produce updates with different intrinsic dimensions.

A common solution is to zero-pad smaller LoRA factors to a shared maximum rank before aggregation. Although this provides a uniform parameter shape, it forces every client update to be represented at the largest rank and therefore communicates unused dimensions for lower-rank clients. More importantly, zero-padding does not attempt to exploit the possibility that heterogeneous client updates may benefit from being aggregated in a shared low-dimensional subspace — a hypothesis this paper tests directly, rather than assumes. We therefore ask whether heterogeneous-rank updates can instead be aggregated in a common subspace while preserving their native rank.

We propose **NSVD-Agg**, a nested singular value decomposition (SVD) aggregation method for heterogeneous-rank federated LoRA. Rather than padding LoRA factors to the maximum rank, NSVD-Agg constructs a shared reference subspace from the raw client updates and projects each native-rank update onto this subspace before aggregation. Figure 1 illustrates the idea for a single

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.

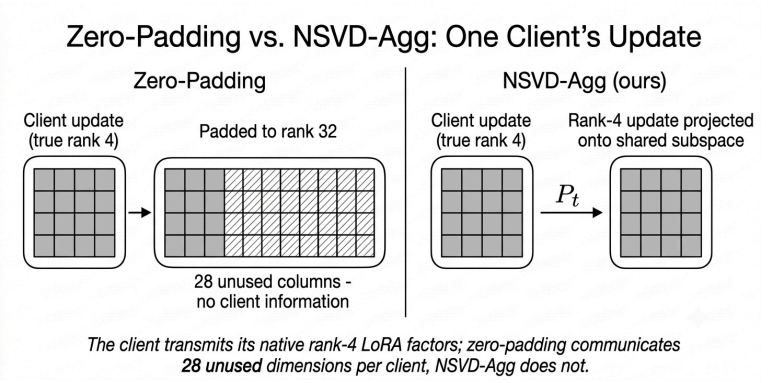

Figure 1: Zero-padding (left) transmits $r_{\max}$ columns per client regardless of true rank, of which $r_{\max} - r_k$ are unused, zero-valued dimensions; NSVD-Agg (right) projects each client's native-rank update onto a tracked subspace instead, transmitting only its true rank.

rank-4 client: zero-padding sends 28 unused columns to reach $r_{\max} = 32$, while NSVD-Agg sends exactly 4.

The reference basis for a round is estimated from the previous round's unprojected updates and held fixed while that round's updates are aggregated, avoiding the subspace-locking effect that arises if the basis is estimated from already-projected updates. Ordering by eigenvalue makes the leading columns a nested chain of subspaces — a property the current operator does not yet exploit, since every client is projected through the same full-rank basis; rank-matched per-client sub-projections are left to future work.

The projection introduces an optimization error absent from ordinary homogeneous-rank aggregation, so we analyze the update through the exact decomposition $d_t = \nabla f(w_t) + \delta_t + e_t^{\text{rank}} + \xi_t$, where δ_t captures client drift, ξ_t stochastic gradient noise, and e_t^{rank} the error induced by projection onto the shared subspace. Under smoothness and bounded-moment assumptions we derive a stationarity bound isolating this rank-induced error, and show it admits an alignment-dependent characterization: updates whose dominant directions align well with the tracked subspace incur smaller projection error. Critically, this guarantee depends on the basis being fixed *before* a round's updates are observed; Section 5.8 shows that dropping this requirement – while empirically competitive – forfeits the guarantee entirely.

We evaluate NSVD-Agg against zero-padding for heterogeneous-rank federated LoRA on Llama-3.2-3B and Qwen2.5-3B. Using simulated heterogeneous ranks $\{4, 8, 16, 32\}$, NSVD-Agg achieves lower final loss than zero-padding in all seven random seeds on both models while reducing client-to-server LoRA communication by an analytical 53.1%. We further isolate the source of this improvement with an untracked exact-product baseline, and examine subspace tracking, maximum-rank scaling, rank distributions, and a mid-training rank change (Appendix).

Our contributions are:

- We introduce **NSVD-Agg**, a native-rank aggregation method that uses a tracked, nested reference subspace instead of zero-padding heterogeneous LoRA updates.
- We provide an optimization analysis based on an exact four-term update decomposition and derive a stationarity bound that explicitly isolates client drift, stochastic noise, and rank-induced projection error, including an alignment-dependent characterization of the latter.
- We evaluate under simulated heterogeneous ranks on Llama-3.2-3B and Qwen2.5-3B, improving final loss across seven seeds on both models with an analytical 53.1% communication reduction; we then isolate the source of this gain with an untracked exact-product baseline that shares the savings but not the guarantee, and which matches or slightly exceeds NSVD-Agg in all seven seeds on both models.

2 Related Work

Federated LoRA and parameter-efficient fine-tuning. LoRA (Hu et al., 2022) adapts models through low-rank updates, substantially reducing trainable parameters relative to full fine-tuning; QLoRA (Dettmers et al., 2023) adds quantization, which we use directly (4-bit NormalFloat (NF4) quantization of the frozen base model). Federated variants aggregate client-side LoRA updates while keeping the pretrained model fixed, typically building on Federated Averaging (FedAvg) (McMahan et al., 2017). Most assume a common adapter rank across clients, which is restrictive under heterogeneous device budgets — and even at matched ranks, our aggregation does not reduce to FedAvg, since the tracked-subspace projection $P_t \neq I$ in general.

Heterogeneous-rank federated adaptation. Several approaches map heterogeneous client adapters into a common representation before aggregation. The direct strategy, introduced by HetLoRA (Cho et al., 2024) alongside rank self-pruning and sparsity-weighted aggregation, pads smaller factors to a shared maximum rank; this preserves compatibility but communicates dimensions unused by lower-rank clients. Our baseline throughout is this zero-padding step with weighted averaging, not HetLoRA's full method: we do not implement its self-pruning or sparsity-weighted aggregation. Others avoid zero-padding differently: RBLA (Chen et al., 2024) proposes a rank-based weighting scheme, and FLoRA (Wang et al., 2024) stacks client updates in the full-rank update space rather than averaging factored ones. FlexLoRA (Bai et al., 2024) synthesizes a full-size update and redistributes it by SVD; our exact-product baseline (Section 5.8) is close in spirit, and our finding that it matches NSVD-Agg suggests FlexLoRA-style aggregation deserves direct comparison in future work. FedHL (Peng et al., 2025) aggregates against a full-rank global basis to remove truncation bias; concurrent work on prefix-nested adapters (Waseem et al., 2026) organizes adapter dimensions into a rank hierarchy. In contrast, NSVD-Agg retains each client's native-rank update and resolves heterogeneity by projection onto a shared reference subspace whose temporal construction supports a non-convex convergence guarantee; direct comparisons against RBLA, FLoRA, FlexLoRA, and FedHL are future work (Section 6).

Low-rank subspace methods and SVD-based aggregation. Singular-value and eigenspace methods are widely used to compress parameter updates and exploit shared structure, including low-rank gradient compression for distributed training (Vogels et al., 2019) and low-rank projection of optimizer state (Zhao et al., 2024). NSVD-Agg differs in how the subspace is constructed and used: the server estimates it from the raw client updates, orders directions by eigenvalue to obtain nested prefix subspaces, and projects with the current basis while reserving the newly estimated one for the next round. This separation lets the subspace evolve with the observed update geometry without making the current aggregation depend on its own projected outputs, and is what supports our convergence guarantee (Section 4). Related but distinct work does not target rank heterogeneity: SLoRA (Babakniya et al., 2023) uses data-driven initialization to narrow the LoRA/full-fine-tuning gap under client heterogeneity, while FedEx-LoRA (Singhal et al., 2025) corrects an aggregation bias present even at homogeneous rank, since averaging low-rank factor products does not equal the product of their averages — orthogonal to, and possibly compounding with, the problem addressed here.

Positioning. Our work thus targets the gap between heterogeneous-rank federated adaptation and subspace-based aggregation: not eliminating rank heterogeneity, but aggregating native-rank updates through a shared, dynamically tracked subspace with a proven convergence guarantee. Where standard non-convex federated analyses (Kairouz et al., 2021) assume bounded-variance client gradient noise, we use a descent-lemma argument under a bounded second moment of the effective update, avoiding an explicit variance-recursion step.

3 Method: Nested-SVD Aggregation

3.1 Problem Setup

Consider a federated learning system with n clients. At communication round t, client k produces a LoRA update

$$\Delta w_{k,t} = A_{k,t} B_{k,t}^{\top}, \qquad A_{k,t}, B_{k,t} \in \mathbb{R}^{d \times r_k},$$

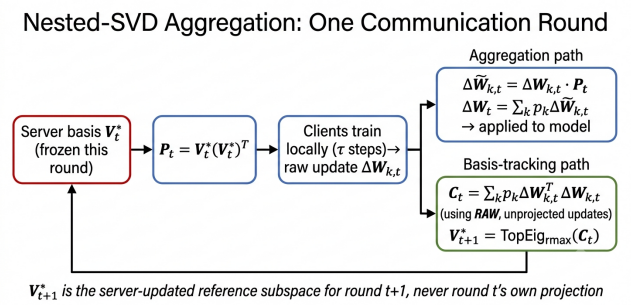

Figure 2: One round of NSVD-Agg. The reference basis $V_t^\star$ is frozen for the round (required for $\mathcal{F}_t$-measurability, Appendix A.9); clients' raw updates feed *two parallel* paths — projection/aggregation (applied to the model this round) and the basis update for round $t+1$ (from raw updates only, never the projected output).

where r_k is the client's native adapter rank and $r_k \in [r_{\min}, r_{\max}]$. Let $p_k \geq 0$ denote the aggregation weight of client k, with $\sum_{k=1}^n p_k = 1$. The goal is to aggregate these updates without requiring all clients to represent their adapters at the common maximum rank $r_{\max}$.

3.2 From Zero-Padding to a Shared Subspace

A standard way to make heterogeneous-rank updates compatible, used by the baseline (Cho et al., 2024), is to zero-pad the LoRA factors. For a client with rank $r_k < r_{\max}$, define

$$\bar{A}_{k,t} = [A_{k,t} \quad 0], \qquad \bar{B}_{k,t} = [B_{k,t} \quad 0],$$

where the zero blocks have $r_{\max} - r_k$ columns. This preserves the represented update because $\bar{A}_{k,t}\bar{B}_{k,t}^\top = A_{k,t}B_{k,t}^\top$. However, the resulting representation is communicated at the maximum rank for every client, even when the client's native update has much smaller rank.

NSVD-Agg instead keeps each update in its native-rank form and introduces a shared reference subspace. Let $V_t^\star \in \mathbb{R}^{d \times r_{\max}}$ with $(V_t^\star)^\top V_t^\star = I$, and define the associated projection matrix $P_t = V_t^\star (V_t^\star)^\top$. Each client update is projected before aggregation:

$$\widetilde{\Delta w}_{k,t} = A_{k,t}(P_t B_{k,t})^\top = \Delta w_{k,t} P_t.$$

The server then aggregates $\Delta w_t = \sum_{k=1}^n p_k \widetilde{\Delta w}_{k,t}$. Dimensional compatibility is thus obtained through a shared subspace rather than by expanding every client update to the maximum rank.

3.3 Nested-SVD Subspace Construction

The reference subspace is estimated from the *raw*, unprojected client updates. Specifically, the server forms

$$C_t = \sum_{k=1}^n p_k \Delta w_{k,t}^\top \Delta w_{k,t}, \qquad V_{t+1}^\star = \text{TopEig}_{r_{\max}}(C_t),$$

where $\text{TopEig}_r(\cdot)$ denotes the operator returning the leading r eigenvectors (by eigenvalue) of its positive semidefinite argument. Because C_t is positive semidefinite, its eigenvectors can be ordered by decreasing eigenvalue, so the first r columns span the top-r eigenspace for every $r \leq r_{\max}$ — a nested chain $S_1 \subset S_2 \subset \cdots \subset S_{r_{\max}}$.

The basis used for projection in round t is kept fixed during that round; the basis estimated from the current raw updates, $V_{t+1}^\star$, is used only in round $t+1$. This ordering is essential: if the basis were instead estimated from the already-projected updates, then $\Delta w_t = (\sum_k p_k \Delta w_{k,t})P_t$, whose row space already lies in the current subspace — re-estimating from such updates provides no new geometric information and locks the reference basis onto its previous value permanently. Figure 2 illustrates one communication round, and Algorithm 1 states it step by step.

Appendix A.9 explains why this ordering is required by the convergence proof and how the first round's basis is bootstrapped.

Algorithm 1 NSVD-Agg: one communication round

Require: Current basis $V_t^\star$ (fixed at round start; not updated until Step 9)
1: $P_t \leftarrow V_t^\star (V_t^\star)^\top$ {frozen for this round — $\mathcal{F}_t$-measurable}
2: Broadcast global model to all clients
3: **for** each client k **do**
4: Run τ local stochastic gradient descent (SGD) steps; obtain raw update $\Delta w_{k,t} = A_{k,t} B_{k,t}^\top$
5: $\widetilde{\Delta w}_{k,t} \leftarrow A_{k,t}(P_t B_{k,t})^\top$ {project using frozen P_t}
6: **end for**
7: $\Delta w_t \leftarrow \sum_k p_k \widetilde{\Delta w}_{k,t}$; apply to global model
8: $C_t \leftarrow \sum_k p_k \Delta w_{k,t}^\top \Delta w_{k,t}$ {RAW updates, not $\widetilde{\Delta w}_{k,t}$ — see subtlety above}
9: $V_{t+1}^\star \leftarrow \mathrm{TopEig}_{r_{\max}}(C_t)$ {used starting next round only}

3.4 Projection Scope and Per-Module Application

NSVD-Agg applies the same $r_{\max}$-dimensional reference projection to every native-rank update, regardless of each client's own r_k, resolving heterogeneous parameter shapes without client-specific prefix projections or a rank-dependent server representation. It runs independently per adapted module, each tracking its own $V_t^\star$. Adaptation is restricted to query and output projections, since key/value projections are non-square under grouped-query attention (Ainslie et al., 2023); generalizing to non-square modules is left as future work (Section 6).

4 Theoretical Analysis

Full statements and proofs appear in Appendix A; this section summarizes. We decompose the effective update direction as $d_t = \nabla f(w_t) + \delta_t + e_t^{\mathrm{rank}} + \xi_t$ — client drift, NSVD projection error, and stochastic noise, respectively — rather than assuming any of these vanish. Under standard smoothness and a bounded second moment of d_t, we establish a non-convex stationarity bound in the style of Ghadimi & Lan (2013):

$$\min_{0 \le t < T} \mathbb{E}\|\nabla f(w_t)\|^2 = O\left(\frac{1}{\sqrt{T}} + D_T + R_T\right),$$

where $D_T = O(L^2 \eta_\ell^2 \tau^2 G^2)$ (client drift, bounded explicitly in Appendix A) and R_T (NSVD projection error) admits a deterministic bound and a tighter alignment-conditional bound (proven for our specific operator):

$$R_T \le \frac{1}{T\eta^2} \sum_{t,k} p_k \, \sigma_1^2(\Delta w_{k,t}) \, r_k \, \alpha_{k,t}, \qquad \alpha_{k,t} = \frac{\|(I - P_t) V_{k,t}\|_F^2}{r_k}.$$

This guarantee relies on P_t being $\mathcal{F}_t$-measurable: fixed before round t's client updates are observed (Appendix A.9). An operator that instead recomputes its projection from the *current* round's aggregate – for instance, by taking the SVD of the exact-product aggregate each round, as in Section 5.8 – is not $\mathcal{F}_t$-measurable, so this bound does not apply to it, regardless of its empirical performance.

On the stronger asymptotic claim. If $\alpha_{k,t} \to 0$, this bound would give the full $O(1/\sqrt{T})$ rate unconditionally; measured $\alpha_{k,t}$ instead plateaus at a bounded, nonzero level (Section 5.5), so we report the alignment-conditional bound as our final result on R_T.

5 Experiments

5.1 Experimental Setup

Unless otherwise stated, experiments use Llama-3.2-3B (Grattafiori et al., 2024) with 4-bit NF4 quantization and LoRA on the query and output projections only. The corpus is tiny-Shakespeare (Karpathy, 2015), partitioned across 20 simulated clients by a Dirichlet(α) distribution (Hsu et al., 2019) with $\alpha = 0.1$ (verified non-independent-and-identically-distributed (non-IID): mean pairwise

Table 1: Llama-3.2-3B: final eval loss, zero-padding vs. NSVD-Agg, across 7 random seeds. Held-out evaluation window.

Seed	Zero-pad	NSVD-Agg	NSVD-Agg wins?
0	3.8647	3.8200	Yes
1	3.5730	3.5518	Yes
2	3.6894	3.6672	Yes
3	3.8317	3.8273	Yes
4	3.9500	3.9355	Yes
5	3.8563	3.8326	Yes
6	3.7737	3.7571	Yes
Mean $\pm$ SD	3.7913 ± 0.1257	3.7702 ± 0.1261	**7/7**

speaker-overlap Jaccard 0.160 at $\alpha = 0.1$ vs. 0.564 at $\alpha = 10$). Each round performs five local steps of AdamW (Loshchilov & Hutter, 2019) at learning rate 10^{-4}, whereas the analysis of Section 4 assumes local SGD for tractability. All reported losses are evaluation losses on a held-out window of each client's own text (tokens 1280–1536 of the first five clients) that local training never reads.

We compare three aggregation settings: (i) equal-rank FedAvg-style aggregation, in which all clients use the same LoRA rank; (ii) heterogeneous-rank zero-padding, in which lower-rank LoRA factors are padded to a common maximum rank before aggregation; and (iii) NSVD-Agg, which retains native-rank updates and projects them onto the tracked reference subspace.

For heterogeneous-rank experiments, clients are assigned ranks from $\{4, 8, 16, 32\}$. To isolate the aggregation effect, heterogeneous rank is simulated by first obtaining updates at the uniform maximum capacity and then truncating each update to its assigned rank before aggregation. Thus, these experiments model heterogeneous communication and update representation, but do not reproduce the additional training-time compute and memory differences that would arise if clients were trained from the outset with different native ranks. This distinction is discussed further in Section 6.

5.2 Homogeneous-Rank Sanity Check

We first verify the pipeline with all clients at rank 16. Over 10 rounds the eval loss decreases from 3.9824 to 3.8044, showing expected behavior before introducing heterogeneity. At this seed equal-rank FedAvg reaches a lower final loss (3.8044) than either heterogeneous-rank method (zero-padding 3.8647, NSVD-Agg 3.8200), indicating heterogeneity itself carries a cost beyond the aggregation scheme; NSVD-Agg narrows this gap without eliminating it.

5.3 Heterogeneous-Rank Aggregation

We next compare zero-padding and NSVD-Agg under the rank assignment $\{4, 8, 16, 32\}$, five clients per rank, $r_{\max} = 32$, 10 rounds, across seven random seeds (Table 1).

NSVD-Agg obtains lower final loss in all seven seeds, with a mean improvement of 0.0210 (standard deviation (SD) 0.0123). At $n = 7$ the non-parametric Wilcoxon signed-rank test (Wilcoxon, 1945) is the more appropriate primary test and gives $p = 0.0156$ (the maximum attainable at this n, since all differences share a sign); the paired t-test agrees ($t = 4.52$, $p = 0.0040$) with a large effect size (Cohen's $d = 1.71$; Cohen, 1988). The improvement is consistent across the seven paired runs, though substantially smaller than on the in-training-window text these clients also trained on; the experiment remains limited to one dataset, client population, and training horizon. A per-seed bar chart appears in Appendix B.2.

5.4 Convergence and Communication

Round-by-round loss trajectories for the heterogeneous-rank setting (seed 0) show the performance difference emerging early in training, with NSVD-Agg reaching a lower final loss than zero-padding throughout; see Figure 4 in Appendix B.1.

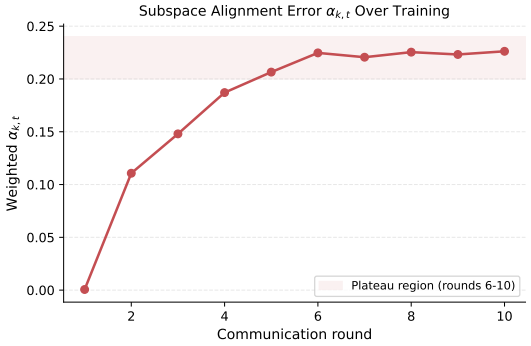

Figure 3: Weighted misalignment α_t (layer-0 query projection) rises sharply over rounds 1–5 then plateaus, rather than shrinking toward zero. This is why the alignment-conditional bound, not the stronger asymptotic $\alpha_{k,t} \to 0$ claim, is reported as this paper's final result on R_T.

The communication advantage follows from retaining native-rank factors. With ranks $\{4, 8, 16, 32\}$ assigned uniformly, the average rank is 15 against a communicated rank of 32 for zero-padding. For a module of input dimension d with bfloat16 factors, client-to-server communication is proportional to $4dr$ bytes per update per adapted module, giving a reduction of $1 - 15/32 = 53.1\%$ — an analytical ratio implied by the rank assignment, not a measured quantity, reported per adapted module rather than as a whole-model total. It counts only client-to-server LoRA-factor traffic, excluding the shared reference basis, optimizer states, and server-to-client broadcast. Since the saving recurs every round, it accumulates approximately linearly (Appendix B.3).

5.5 Reference-Subspace Misalignment

To examine the geometric quantity in the theoretical analysis, we track the average subspace misalignment $\alpha_t = \sum_k p_k \alpha_{k,t}$ — larger values mean client updates lie further outside the tracked subspace — on a representative module (the layer-0 query projection). Across 10 rounds α_t rises sharply from the round-1 bootstrap value of 0.0006, then plateaus around 0.22–0.23 rather than shrinking toward zero. Because the first-round basis is bootstrapped from same-round updates (Appendix A.9), the round-1 value is not evidence about steady-state alignment. Notably, this plateau is nonzero yet NSVD-Agg still consistently outperforms zero-padding (Table 1): it does not require perfect subspace recovery to aggregate usefully.

Figure 3 makes this pattern directly visible: a rapid initial rise followed by a stable plateau, not a decay curve.

5.6 Additional Experiments

Three further experiments are reported in full in the appendix. A mid-training rank change, reducing five clients from rank 32 to 8 after round 10 of 20, produces no discontinuity at the switch for either method (held-out final loss 3.8003 for NSVD-Agg vs. 3.8137 for zero-padding on Llama), though NSVD-Agg's Llama advantage narrows to within measurement noise by round 20 while remaining stable on Qwen (Appendix B.4). Single-seed sweeps over $r_{\max}$ show a larger improvement at $r_{\max} = 32$ than at 16 on both models, with the trend to 64 differing by model (Appendix B.5). A single-seed sweep over the shape of the rank distribution shows the improvement largest for near-homogeneous and balanced patterns on Llama, with a weaker and partly different pattern on Qwen (Appendix B.6).

5.7 Ablation of Reference-Basis Tracking

To separate the effect of projection from the effect of dynamically tracking the update geometry, we compare four variants using the same seed: zero-padding, projection onto a fixed reference basis (bootstrapped once at round 1, then frozen), projection onto a random basis (unrelated to any real

Table 2: Exact-product aggregation (no tracking, no round lag) vs. NSVD-Agg, 7 seeds, held-out evaluation. Positive difference means exact-product has the lower (better) loss. NSVD-Agg here is a separate 7-seed run of the Table 1 configuration (3.7698 vs. 3.7702 Llama; 3.9324 vs. 3.9336 Qwen), within rerun variation.

	Llama-3.2-3B			Qwen2.5-3B		
	Mean ± SD	Diff. (±SD)	Wins	Mean ± SD	Diff. (±SD)	Wins
Exact-product	3.7664 ± 0.1260	0.0033 ± 0.0021	7/7	3.9280 ± 0.0795	0.0043 ± 0.0018	7/7
NSVD-Agg	3.7698 ± 0.1259			3.9324 ± 0.0787		

update), and NSVD-Agg with round-wise basis tracking. Same seed-0, 20-client, $r_{\max} = 32$ setup as the main result.

Final eval losses are 3.8208 for NSVD-Agg (tracked), 3.8635 for zero-padding, 3.9450 for the fixed basis, and 3.9810 for the random basis. These four variants come from one common rerun of the seed-0 configuration, so their absolute values differ slightly from the seed-0 row of Table 1 (3.8200 / 3.8647); the same applies to the seed-0 rows of Tables 3 and 5. The spread across these reruns (≤ 0.008) is small relative to the between-method gaps being compared, and all comparisons are made within a run, never across runs. Both Fixed and Random are worse than zero-padding, not merely worse than NSVD-Agg. Random (no relationship to real updates) underperforms Fixed (a stale but once-real snapshot), which in turn underperforms NSVD-Agg (continuously tracked) — a monotonic ordering consistent with projection quality depending on how well the basis reflects the current client-update geometry. This shows that a poorly chosen subspace actively hurts, and that projection alone does not explain NSVD-Agg's advantage. It does not, by itself, show that *tracking* is what produces the advantage over zero-padding: Section 5.8 adds a fifth variant that isolates this question directly. Since this ablation uses one seed, it should be interpreted as mechanistic evidence rather than a statistically established effect.

5.8 Isolating the Source of the Improvement: Exact-Product Aggregation

The four variants above cannot separate two effects NSVD-Agg combines: aggregating via *exact* client-update products, $\sum_k p_k A_k B_k^\top$, rather than zero-padding's separate-factor averaging $(\sum_k p_k \bar{A}_k)(\sum_k p_k \bar{B}_k)^\top$; and projecting onto a subspace *tracked* across rounds. We add a fifth variant that keeps the first property and drops the second: each round, the server forms the exact weighted aggregate from clients' truncated, native-rank updates, then keeps its best rank-$r_{\max}$ approximation (via SVD) to continue training. No state is carried between rounds, and the current round's own aggregate determines its own projection – so, unlike NSVD-Agg, this operator is not $\mathcal{F}_t$-measurable and the guarantee of Section 4 does not apply to it (Appendix A.9). This construction is close in spirit to FlexLoRA (Bai et al., 2024), though it truncates only at upload rather than redistributing a client-specific rank on download.

On the same single-seed setup as Section 5.7, this exact-product variant reaches 3.8144 on Llama and 3.9585 on Qwen, compared with NSVD-Agg's 3.8208 and 3.9638 – slightly ahead of NSVD-Agg on one seed for both models. Table 2 confirms this holds across seeds: on both models, across all seven seeds, the exact-product variant matches or slightly exceeds NSVD-Agg's loss.

The margin is small but directionally consistent: paired t-tests give $t = 4.27$ ($p = 0.0053$, $d = 1.61$) on Llama and $t = 6.40$ ($p = 0.00069$, $d = 2.42$) on Qwen, and the same ordering appears on the in-training-window metric with a clearer margin (Appendix A.9). We read this as evidence that exact-product aggregation, not subspace tracking, accounts for essentially all of NSVD-Agg's advantage over zero-padding. A simple explanation fits: with all clients initialized from the same random LoRA basis, their updates' dominant directions are already closely aligned, so a per-round exact-aggregate SVD loses little in truncating to rank $r_{\max}$, leaving little room for a tracked subspace to improve on it. What tracking buys is the $\mathcal{F}_t$-measurability the guarantee of Section 4 requires, at a measured cost of about 0.003–0.004 loss. The Fixed/Random comparison in Section 5.7 still shows that subspace *choice* matters, so this narrows what tracking explains without erasing the role of projection quality.

5.9 CROSS-MODEL VALIDATION: QWEN2.5-3B

To test whether these findings are specific to Llama-3.2-3B, we repeat the full experimental protocol on Qwen2.5-3B (Yang et al., 2024) — a different model family, also using grouped-query attention, with a different hidden size ($d = 2048$ vs. 3072 for Llama) — keeping every other experimental choice identical: the same 7 seeds, round counts, rank sets, client population, and Dirichlet partition.

The homogeneous-rank sanity check behaves as expected (eval loss $4.1087 \rightarrow 3.9194$ over 10 rounds), confirming the pipeline transfers correctly before introducing heterogeneity. On the main 7-seed comparison, NSVD-Agg again beats zero-padding in every seed (Table 7 in Appendix B.7); the non-parametric Wilcoxon signed-rank test gives $p = 0.0156$ (again the maximum attainable at $n = 7$), the paired t-test gives $t = 12.42$ ($p = 1.7\times10^{-5}$), and the paired effect size is large (Cohen's $d = 4.69$). The mean improvement is larger than on Llama (0.0267 vs. 0.0210) and the per-seed spread is tighter (SD 0.0057 vs. 0.0123), so Qwen shows both the larger and the more consistent effect of the two models — a real, model-specific difference rather than a uniform replication.

The mechanism findings replicate qualitatively. The communication reduction is identical at 53.1%, following purely from the rank ratio and not from d. The basis-tracking ablation shows the same ordering as Llama (NSVD-Agg 3.9638 < Fixed 4.0515 < Random 4.1068, single-seed), and the exact-product variant again matches NSVD-Agg closely (Table 2), so the finding that tracking is not the source of the gain holds on both models. Model-specific differences: the $r_{\max}$ ablation increases monotonically on Qwen but not Llama, and the rank-shape ablation is weaker and partly different on Qwen (Appendices B.5, B.6). The dynamic rank switch again shows no spike, with NSVD-Agg's advantage stable on Qwen through round 20 but narrowing to measurement noise on Llama (Appendix B.4).

Most notably for the theoretical claims in Section 4, the misalignment measure $\alpha_{k,t}$ shows the same qualitative behavior on Qwen as on Llama: it rises over early rounds and plateaus at a persistent, nonzero level rather than shrinking toward zero (first-half mean 0.1147, second-half mean 0.1719). This is independent evidence, on a second model, for reporting the alignment-conditional bound as the paper's final theoretical result rather than the stronger, unconditional $\alpha_{k,t} \rightarrow 0$ claim.

6 LIMITATIONS

Method scope. LoRA adaptation is restricted to query and output projections, since key/value under grouped-query attention are non-square and unsupported (Section 3); and although the tracked basis is nested, the operator applies the same full-rank projection to every client rather than exploiting rank-matched per-client sub-projections. *Theoretical claim.* Declining to pursue the asymptotic $\alpha_{k,t} \rightarrow 0$ result (Section 5.5) rests on a single module and single-seed runs per model, each only 10 rounds. *Mechanism.* Section 5.8 shows an untracked exact-product baseline matching or slightly exceeding NSVD-Agg on one setup, seven seeds, two models, so our contribution is best read as a convergence guarantee obtained at a small empirical cost — a trade-off measured only at this scale. *Empirical scope.* Experiments use 20 simulated clients and a held-out slice of clients' own text rather than an independent validation set; beyond the two 3B families of Section 5.9, larger models, more clients, other architectures, and comparisons beyond zero-padding (RBLA/FLoRA/FlexLoRA/FedHL) remain future work. *Simulation methodology.* Heterogeneous ranks are simulated by truncating a uniformly-trained update to each client's assigned rank, not by training under a genuinely smaller local adapter from the start; this models communication-time rank constraints more directly than training-time compute or memory constraints.

7 CONCLUSION

We presented NSVD-Agg for aggregating heterogeneous-rank federated LoRA updates without zero-padding, with a convergence analysis that bounds client drift and projection error explicitly rather than assuming either away. Across 7 seeds on Llama-3.2-3B and Qwen2.5-3B it improves final loss and communication efficiency over zero-padding, with no discontinuity at a mid-training rank change. An exact-product baseline sharing its savings but not its guarantee matches or slightly exceeds it empirically, so NSVD-Agg's value lies in a proven guarantee at a small, measured cost rather than a larger gain from tracking.

AI USE STATEMENT

The author(s) used a generative artificial intelligence (AI) tool to assist with preparing the manuscript, including improving the clarity and grammar of the writing and preparing, formatting, and debugging the LaTeX source (for example, layout, figure and table placement, and bibliography formatting). Generative AI was not used to fabricate data, results, or citations. The author(s) take full responsibility for all content in this work, including any text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Full experimental configuration (model, dataset partitioning, hyperparameters) is given in Section 5. Complete proofs of all theoretical claims, including precise assumption statements, appear in Appendix A. Source code will be made available as an anonymous repository for the review period.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, 2023.

Sara Babakniya, Ahmed Roushdy Elkordy, Yahya H Ezzeldin, Qingfeng Liu, Kee-Bong Song, Mostafa El-Khamy, and Salman Avestimehr. SLoRA: Federated parameter efficient fine-tuning of language models. In *International Workshop on Federated Learning in the Age of Foundation Models, NeurIPS*, 2023.

Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. Federated fine-tuning of large language models under heterogeneous tasks and client resources. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Shuaijun Chen, Omid Tavallaie, Niousha Nazemi, and Albert Y Zomaya. RBLA: Rank-based-LoRA-aggregation for fine-tuning heterogeneous models in FLaaS. *arXiv preprint arXiv:2408.08699*, 2024.

Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, and Gauri Joshi. Heterogeneous LoRA for federated fine-tuning of on-device foundation models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 12903–12913, 2024.

Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 2nd edition, 1988.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, 2023.

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Aaron Grattafiori et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Peter Kairouz et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 2021.

Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. `http://karpathy.github.io/2015/05/21/rnn-effectiveness/`, 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1273–1282, 2017.

Yongle Peng et al. FedHL: Federated learning for heterogeneous low-rank adaptation via unbiased aggregation. *arXiv preprint arXiv:2505.18494*, 2025.

Raghav Singhal, Kaustubh Ponkshe, and Praneeth Vepakomma. FedEx-LoRA: Exact aggregation for federated and efficient fine-tuning of foundation models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1316–1336, 2025.

Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. PowerSGD: Practical low-rank gradient compression for distributed optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. FLoRA: Federated fine-tuning large language models with heterogeneous low-rank adaptations. In *Advances in Neural Information Processing Systems*, 2024.

Muhammad Waseem et al. PreLort: Prefix-nested LoRA for federated fine-tuning under rank heterogeneity. *arXiv preprint arXiv:2606.15963*, 2026.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

An Yang et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61121–61143, 2024.

A Full Theoretical Analysis

A.1 Problem Setup

Let n clients have weights $p_k = N_k/N$, $\sum_k p_k = 1$, with global objective $f(w) = \sum_{k=1}^{n} p_k f_k(w)$, where $\{f_k\}$ may be arbitrarily non-IID. At round t, the server broadcasts w_t; client k runs τ local steps and returns a LoRA update $\Delta w_{k,t} = A_{k,t}B_{k,t}^{\top}$ with $\operatorname{rank}(\Delta w_{k,t}) \le r_k \in [r_{\min}, r_{\max}]$. The server applies the Nested-SVD operator $\mathcal{P}_t$ to obtain $\widetilde{\Delta w}_{k,t} = \mathcal{P}_t(\Delta w_{k,t})$ and aggregates $\Delta w_t = \sum_k p_k \widetilde{\Delta w}_{k,t}$. The effective descent direction is $d_t = -\frac{1}{\eta}\Delta w_t$, so $w_{t+1} = w_t - \eta d_t$.

A.2 Error Decomposition

$$d_t = \nabla f(w_t) + \delta_t + e_t^{\mathrm{rank}} + \xi_t,$$

where $\delta_t = \sum_k p_k\big(\bar{\nabla}_{k,t} - \nabla f_k(w_t)\big)$ is client drift, with $\bar{\nabla}_{k,t} := \frac{1}{\tau}\sum_{e=0}^{\tau-1} \nabla f_k(w_{t,e}^k)$ the trajectory-averaged true local gradient; $e_t^{\mathrm{rank}} = \frac{1}{\eta}\sum_k p_k\big(\Delta w_{k,t} - \widetilde{\Delta w}_{k,t}\big)$ is the rank-projection error; and $\xi_t := \sum_k p_k(\bar{g}_{k,t} - \bar{\nabla}_{k,t})$, with $\bar{g}_{k,t} := \frac{1}{\tau}\sum_{e=0}^{\tau-1} g_{k,t,e}$ the trajectory-averaged stochastic gradient, is stochastic noise. **This decomposition identifies the server-level step size with the effective per-round local step size,** $\eta := \eta_\ell\tau$: since $\Delta w_{k,t} = -\eta_\ell \sum_{e=0}^{\tau-1} g_{k,t,e} = -\eta\bar{g}_{k,t}$ under local SGD, $d_t = -\Delta w_t/\eta$ decomposes exactly (not approximately) as $\sum_k p_k \bar{g}_{k,t}$ (before projection), matching $\nabla f(w_t) + \delta_t + \xi_t$ by construction. δ_t **is defined via the trajectory-averaged gradient** $\bar{\nabla}_{k,t}$**, not the final local iterate,** because this is what makes $\mathbb{E}[\xi_t \mid \mathcal{F}_t] = 0$ hold rigorously: by

the tower property, $\mathbb{E}[\bar{g}_{k,t} \mid \mathcal{F}_t] = \mathbb{E}[\bar{\nabla}_{k,t} \mid \mathcal{F}_t]$ follows directly from each local step's conditional unbiasedness $\mathbb{E}[g_{k,t,e} \mid w^k_{t,e}] = \nabla f_k(w^k_{t,e})$, giving $\mathbb{E}[\xi_t \mid \mathcal{F}_t] = 0$ with no further assumption. Defining δ_t via the final iterate $w^k_{t,\tau}$ instead is a common simplification in this literature, but does not by itself guarantee $\mathbb{E}[\xi_t \mid \mathcal{F}_t] = 0$, since the trajectory-average and the final-point gradient need not have the same conditional mean; we therefore use the averaged definition throughout.

A.3 Assumptions

A1 (Smoothness). Each f_k is L_k-smooth; set $L = \max_k L_k$. Then $f = \sum_k p_k f_k$ is L-smooth. **A2 (Bounded effective-update second moment).** $\mathbb{E}[\|d_t\|^2 \mid \mathcal{F}_t] \le G_d^2$. **A3 (Bounded local stochastic gradients).** $\|g_{k,t,e}\| \le G$ almost surely, for every local step $e < \tau$ and every round — a bound on the stochastic gradient samples actually used by local SGD, not on $\|\nabla f_k(w)\|$ directly. **A4 (Bounded NSVD reconstruction error).** $R_T = \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|e_t^{\mathrm{rank}}\|^2$ is left explicit, no spectral decay law assumed.

A bounded second moment (A2) implies a bounded variance; we use it as a different proof route that avoids an explicit variance-recursion step. For Lemma A.6's martingale property to hold, $\mathcal{P}_t$ must be $\mathcal{F}_t$-measurable: determined from information strictly prior to round t's client randomness.

A.4 Lemma 1: Client Drift

Under A3 and τ local steps at rate η_ℓ: $\|\delta_t\| \le L\eta_\ell \tau G$, hence $D_T := \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\delta_t\|^2 \le L^2\eta_\ell^2\tau^2 G^2$.

Proof. Telescoping the actual local SGD update $w^k_{t,e+1} = w^k_{t,e} - \eta_\ell g_{k,t,e}$ gives $\|w^k_{t,e} - w_t\| \le e\eta_\ell G$ for each $e \le \tau$, directly from A3. By A1 (smoothness), $\|\nabla f_k(w^k_{t,e}) - \nabla f_k(w_t)\| \le L\|w^k_{t,e} - w_t\| \le Le\eta_\ell G$. Averaging over $e = 0, \dots, \tau-1$ (matching δ_t's definition via $\bar{\nabla}_{k,t}$) gives $\|\bar{\nabla}_{k,t} - \nabla f_k(w_t)\| \le \frac{1}{\tau}\sum_{e=0}^{\tau-1} Le\eta_\ell G = L\eta_\ell G \cdot \frac{\tau-1}{2} \le L\eta_\ell \tau G$; average over clients, square, average over t. □

A.5 Lemma 2: Deterministic NSVD Projection Error

Let $\Delta w_{k,t} = U_{k,t}\Sigma_{k,t}V_{k,t}^\top$ and P_t the orthogonal projector onto the server's reference subspace. Then

$$\|\Delta w_{k,t} - \widetilde{\Delta w}_{k,t}\|_F \le \|\Delta w_{k,t}\|_F, \qquad \|e_t^{\mathrm{rank}}\|_F^2 \le \frac{1}{\eta^2}\sum_k p_k \|\Delta w_{k,t}\|_F^2.$$

Proof. $\|I - P_t\|_2 \le 1$ for an orthogonal projector; apply Jensen's inequality. □

This bound is always true but does not establish a rank-only penalty: without an alignment assumption, orthogonal subspaces make it tight. If $\|(I - P_t)V_{k,t}\|_F^2 \le \alpha_{k,t} r_k$ for some $\alpha_{k,t} \in [0,1]$, then

$$R_T \le \frac{1}{T\eta^2}\sum_{t,k} p_k\, \sigma_1^2(\Delta w_{k,t})\, r_k\, \alpha_{k,t},$$

which is the NSVD-specific alignment-conditional bound, and holds only under the stated alignment condition. The $1/\eta^2$ prefactor does not diverge as $\eta \to 0$: under local SGD $\Delta w_{k,t} = -\eta \bar{g}_{k,t}$, so $\sigma_1^2(\Delta w_{k,t})$ itself scales as η^2 and the two factors cancel.

A.6 Lemma 3: Descent Under Drift, Rank Error, and Stochastic Noise

$$\mathbb{E}[f(w_{t+1})] \le \mathbb{E}[f(w_t)] - \frac{\eta}{2}\mathbb{E}\|\nabla f(w_t)\|^2 + \eta\mathbb{E}\|\delta_t\|^2 + \eta\mathbb{E}\|e_t^{\mathrm{rank}}\|^2 + \frac{L\eta^2}{2}G_d^2.$$

Proof. L-smoothness gives $f(w_{t+1}) \le f(w_t) - \eta\langle \nabla f(w_t), d_t\rangle + \frac{L\eta^2}{2}\|d_t\|^2$. Substitute the decomposition; the ξ_t cross-term vanishes in expectation. Apply Young's inequality to the $\delta_t, e_t^{\mathrm{rank}}$ cross-terms; bound $\mathbb{E}\|d_t\|^2 \le G_d^2$. □

A.7 Main Theorem

Under A1–A4, with $\Delta_f = f(w_0) - f^\star$ and $\eta \le 1/L$:

$$\min_{0\le t<T} \mathbb{E}\|\nabla f(w_t)\|^2 \le \frac{2\Delta_f}{\eta T} + 2D_T + 2R_T + L\eta G_d^2.$$

Choosing $\eta = \Theta(T^{-1/2})$:

$$\min_{0\le t<T} \mathbb{E}\|\nabla f(w_t)\|^2 = O\left(\frac{\Delta_f + LG_d^2}{\sqrt{T}} + D_T + R_T\right).$$

Proof. Telescope Lemma A.6 from $t = 0$ to $T-1$, use $f(w_T) \ge f^\star$, divide by $\eta T/2$; the minimum is bounded by the average. □

Corollary ($O(1/\sqrt{T})$ stationarity). If additionally $D_T = O(T^{-1/2})$ and $R_T = O(T^{-1/2})$, then $\min_{0\le t<T} \mathbb{E}\|\nabla f(w_t)\|^2 = O(T^{-1/2})$.

Corollary (Explicit client drift). Combining Lemma 1: $D_T = O(L^2\eta_\ell^2\tau^2G^2)$, scaling as τ^2. With $\eta_\ell = O(T^{-1/4})$, this term is $O(T^{-1/2})$ for fixed τ. **Note on the experiments:** achieving the full $O(1/\sqrt{T})$ rate requires $\eta = \eta_\ell\tau = \Theta(T^{-1/2})$, i.e., η_ℓ shrinking across rounds; the experiments in Section 5 instead use a fixed $\eta_\ell = 10^{-4}$ throughout. The reported empirical results should be read as validating the method and the qualitative error decomposition, not as an implementation of the theorem's asymptotic step-size schedule.

A.8 What the Theorem Does and Does Not Claim

Established: an upper bound on the *minimum* stationarity gap over the trajectory, conditional on $D_T, R_T = O(1/\sqrt{T})$; D_T resolved explicitly; a fully deterministic bound on R_T. Not established: last-iterate convergence; any $\Theta(\cdot)$ lower bound; a rank-gap or logarithmic penalty on R_T; a rank-only bound independent of alignment; elimination of non-IID drift (bounded, not zero); any guarantee for operators, such as the exact-product baseline of Section 5.8, whose projection is not $\mathcal{F}_t$-measurable.

A.9 The Nested-SVD Operator: Full Specification

Given server basis $V_t^\star \in \mathbb{R}^{d\times r_{\max}}$ (orthonormal columns) and client update $A_{k,t}, B_{k,t} \in \mathbb{R}^{d\times r_k}$:

$$P_t = V_t^\star (V_t^\star)^\top, \quad \widetilde{B}_{k,t} = P_t B_{k,t}, \quad \widetilde{A}_{k,t} = A_{k,t}, \quad \widetilde{\Delta w}_{k,t} = \widetilde{A}_{k,t}\widetilde{B}_{k,t}^\top.$$

The basis is updated from raw, unprojected client updates rather than from the projected aggregate, whose row space is already contained in $\mathrm{span}(V_t^\star)$, so that re-estimating from it would lock the basis at its current value:

$$C_t = \sum_k p_k \Delta w_{k,t}^\top \Delta w_{k,t}, \qquad V_{t+1}^\star = \mathrm{TopEig}_{r_{\max}}(C_t).$$

Chronological order (required for P_t's $\mathcal{F}_t$-measurability): $V_t^\star \to P_t \to$ clients compute updates $\to$ project $\to$ aggregate $\to V_{t+1}^\star$ (from round t's raw updates, used at round $t+1$).

P_t must be determined before round t's client randomness is realized (formally: $\mathcal{F}_t$-measurable), or the martingale argument in our convergence proof (Section 4) is not justified. Algorithm 1 enforces this ordering structurally: $V_{t+1}^\star$ is not computed until Step 9, strictly after P_t (Step 1) has already been used for every client's projection.

The first round is a one-time exception to this: no prior round exists to draw $V_1^\star$ from, so it is bootstrapped directly from round 1's own raw updates and used to project those same updates, meaning it is not $\mathcal{F}_1$-measurable in the strict sense required for $t \ge 2$. This is a boundary case affecting round 1 only — every subsequent $V_t^\star$ is correctly estimated from the prior round's raw updates alone, satisfying the measurability the convergence proof requires.

The exact-product baseline (Section 5.8) is not $\mathcal{F}_t$-measurable. It forms $P_t' = V'(V')^\top$, where V' holds the top $r_{\max}$ singular vectors of round t's own exact-product aggregate $\sum_k p_k A_{k,t}B_{k,t}^\top$ –

the same client updates P'_t is then applied to. This construction is implemented via a factored route (stacking $\bar{A}_k = \sqrt{p_k}A_{k,t}$, $\bar{B}_k = \sqrt{p_k}B_{k,t}$ and truncating $\bar{A}\bar{B}^\top$ to rank $r_{\max}$), verified against a dense SVD on real updates to within 8.2×10^{-5} relative error. On the in-training-window metric, the exact-product variant scores 3.8358 (Llama) and 4.0157 (Qwen), against NSVD-Agg's 3.8665 and 4.0308 – the same ordering as the held-out result in Table 2, with a clearer margin.

The asymptotic $\alpha_{k,t} \to 0$ claim. A stronger claim — that $\alpha_{k,t} \to 0$ as training progresses — was tested empirically before attempting a proof. Measured $\alpha_{k,t}$ rises over rounds 1–5 then plateaus around 0.22–0.23 for rounds 6–10, stabilizing at a bounded, persistent level rather than shrinking toward zero. The data do not support this asymptotic claim, so the alignment-conditional bound stands as this paper's final theoretical result on R_T.

B Additional Empirical Results

B.1 Convergence Curves

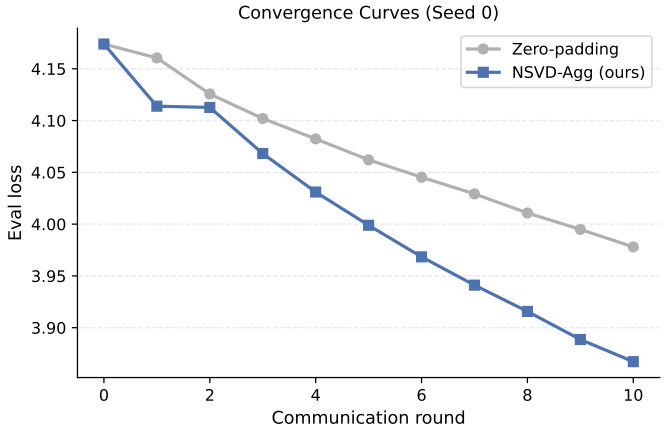

Figure 4: Round-by-round eval loss for seed 0 (in-training-window metric shown; see Table 1 for the held-out values reported in the main text). NSVD-Agg's advantage is visible from round 1.

Figure 4 shows the loss trajectories for the heterogeneous-rank setting (seed 0), on the in-training-window metric used to generate the original figure asset. The held-out per-seed values reported in the main text are given in Table 1.

B.2 Per-Seed Comparison (Llama-3.2-3B)

Figure 5 shows the same seven-seed comparison visually, on the in-training-window metric used to generate the original figure asset; the held-out margin per seed is smaller but follows the same ordering (Table 1).

B.3 Communication Cost Over Training

Figure 6 makes this cumulative effect visible: the savings are not a one-time reduction but accumulate linearly, reaching more than a $2\times$ total communication difference by round 10. This communication result is independent of the choice of evaluation metric, since it is computed analytically from the rank assignment rather than measured.

B.4 Dynamic Rank Change

We further test whether the aggregation procedure remains stable when client ranks change during training. We train for 20 rounds and, after round 10, reduce the rank of five clients from 32 to 8. The loss curves show no visible discontinuity at the switch for either strategy or model. Held-out

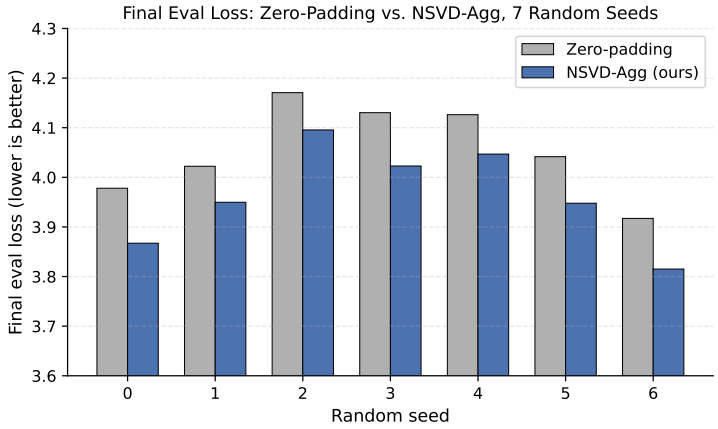

Figure 5: Final eval loss, zero-padding vs. NSVD-Agg, across all 7 random seeds (in-training-window metric shown; see Table 1 for the held-out values reported in the main text). NSVD-Agg (blue) is lower (better) in every seed.

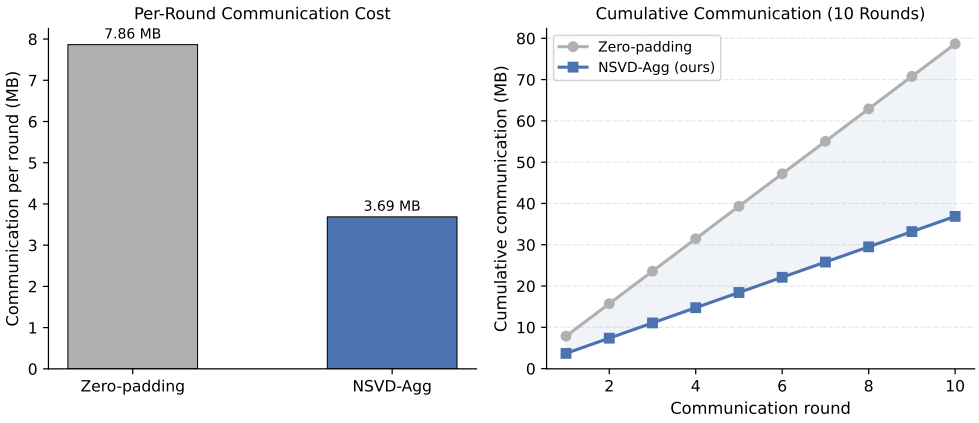

Figure 6: Per-round (left) and cumulative over 10 rounds (right) communication cost per adapted module (20 clients, one $r_{\max} = 32$ module), zero-padding vs. NSVD-Agg. The gap widens every round because the 53.1% reduction applies to every round, accumulating linearly rather than as a one-time saving.

final losses at round 20 are 3.8137 (zero-padding) and 3.8003 (NSVD-Agg) on Llama, and 3.9448 (zero-padding) and 3.9179 (NSVD-Agg) on Qwen. On Llama, NSVD-Agg's advantage narrows steadily after round 14 and is within measurement noise by round 20; on Qwen the advantage remains approximately stable (about 0.027) throughout. This pattern – a shrinking held-out advantage alongside a widening in-training-window one – also appears in the main sweep (Section 5) and is consistent with later rounds increasingly fitting text the model has already trained on.

Figure 7 shows both trajectories across the full 20 rounds on the in-training-window metric used to generate the original figure asset: neither curve shows a visible discontinuity at the switch point. This experiment illustrates the smooth transition observed under this particular rank-change configuration; it does not establish robustness to arbitrary rank schedules or more frequent rank changes.

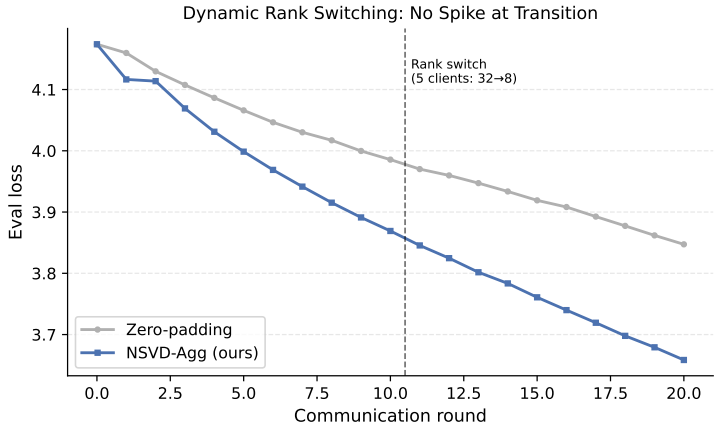

Figure 7: Both strategies' loss curves continue smoothly through the round-10/11 rank switch — no visible discontinuity for either strategy (in-training-window metric shown; see text for held-out final losses).

B.5 Additional Ablation: Maximum-Rank Scaling

We vary the maximum rank while preserving the relative heterogeneous-rank pattern and report results from a single seed, held-out evaluation ($r_{\max} = 16$: ranks $\{2, 4, 8, 16\}$; $r_{\max} = 64$: ranks $\{8, 16, 32, 64\}$). Results appear in Table 3 (Llama) and Table 4 (Qwen).

Table 3: NSVD-Agg's improvement over zero-padding across three $r_{\max}$ settings (seed 0, held-out evaluation).

$r_{\max}$	Zero-pad	NSVD-Agg	Improvement
16	3.8789	3.8470	0.0319
32	3.8663	3.8205	0.0458
64	3.8396	3.8055	0.0341

On Llama, the improvement rises from $r_{\max} = 16$ to 32 but does not continue to 64. The change from 32 to 64 (-0.0117) is larger than the ≈ 0.008 zero-padding rerun spread observed elsewhere in this study (Section 5.7), so whether it reflects a genuine decline or run-to-run noise cannot be determined from this single seed alone. This is a single-seed, single-model observation and should not be read as evidence that communication waste alone drives the optimization improvement.

Table 4: Same ablation on Qwen2.5-3B (seed 0, held-out evaluation).

$r_{\max}$	Zero-pad	NSVD-Agg	Improvement
16	4.0025	3.9837	0.0188
32	3.9880	3.9626	0.0254
64	3.9775	3.9465	0.0310

On Qwen, the improvement increases monotonically with $r_{\max}$ ($0.0188 \rightarrow 0.0254 \rightarrow 0.0310$), unlike on Llama. Both trends are single-seed observations; the model-dependence itself should be treated as a description of these two runs rather than a general law.

B.6 Additional Ablation: Rank-Distribution Shape

We vary the shape of the heterogeneous rank distribution while keeping the maximum rank at 32 (single seed, held-out evaluation); see Table 5 (Llama) and Table 6 (Qwen).

Table 5: Final loss for different rank distributions after 10 rounds, Llama-3.2-3B (single seed, held-out evaluation).

Rank distribution	Zero-padding	NSVD-Agg	Difference	Wins?
Near-homogeneous $\{16, 16, 16, 32\}$	3.8604	3.8180	0.0424	Yes
Balanced $\{4, 8, 16, 32\}$	3.8586	3.8220	0.0366	Yes
Highly skewed $\{2, 4, 8, 32\}$	3.8595	3.8267	0.0328	Yes
Extreme $\{2, 2, 4, 32\}$	3.8619	3.8313	0.0306	Yes

NSVD-Agg obtains lower final loss than zero-padding for all four rank distributions tested, and the ordering (near-homogeneous $>$ balanced $>$ skewed $>$ extreme) is preserved. The spread across patterns is small (0.012), close to the zero-padding rerun variation observed elsewhere; the ordering should be read as descriptive of this single seed rather than a statistically established effect.

Table 6: Same ablation on Qwen2.5-3B (single seed, held-out evaluation).

Rank distribution	Zero-padding	NSVD-Agg	Difference	Wins?
Near-homogeneous $\{16, 16, 16, 32\}$	3.9929	3.9613	0.0316	Yes
Balanced $\{4, 8, 16, 32\}$	3.9928	3.9626	0.0302	Yes
Highly skewed $\{2, 4, 8, 32\}$	3.9972	3.9663	0.0309	Yes
Extreme $\{2, 2, 4, 32\}$	3.9912	3.9704	0.0208	Yes

On Qwen, near-homogeneous, balanced, and highly-skewed distributions are close to each other (0.030–0.032), with only the extreme pattern showing a clearly smaller gain (0.0208); the smooth decline seen on Llama does not appear. NSVD-Agg still wins on all four patterns for both models. These results are descriptive and should not be interpreted as establishing a universal relationship between rank distribution and aggregation benefit.

B.7 QWEN2.5-3B PER-SEED RESULTS

Table 7: Qwen2.5-3B: final eval loss, zero-padding vs. NSVD-Agg, across the same 7 seeds used for Llama. Held-out evaluation window.

Seed	Zero-pad	NSVD-Agg	NSVD-Agg wins?
0	3.9945	3.9632	Yes
1	3.8571	3.8309	Yes
2	3.8628	3.8440	Yes
3	4.0243	3.9964	Yes
4	4.0701	4.0468	Yes
5	3.9430	3.9070	Yes
6	3.9705	3.9469	Yes
Mean $\pm$ SD	3.9603 ± 0.0794	3.9336 ± 0.0786	**7/7**