PREVENTING RANK COLLAPSE IN FEDERATED LOW-RANK ADAPTATION WITH CLIENT HETEROGENEITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated low-rank adaptation (FedLoRA) has facilitated communication-efficient and privacy-preserving fine-tuning of foundation models for downstream tasks. In practical federated learning scenarios, client heterogeneity in system resources and data distributions motivates the use of heterogeneous LoRA ranks across clients. However, we identify a previously overlooked phenomenon in heterogeneous FedLoRA with SVD-based allocation, termed *rank collapse*, where the energy of the global update becomes concentrated in the minimum shared rank, resulting in suboptimal performance and high sensitivity to rank configurations. Through theoretical analysis, we reveal the underlying mechanism of rank collapse: a mismatch between rank-agnostic aggregation weights and rank-dependent client contributions, which systematically suppresses higher-rank updates at a geometric rate over rounds. Motivated by this insight, we propose raFLoRA, a rank-partitioned aggregation method that decomposes local updates into rank partitions and then aggregates each partition weighted by its effective client contributions. Extensive experiments across vision, language, and reasoning tasks show that raFLoRA prevents rank collapse, improves model performance, and enhances robustness across diverse heterogeneous configurations compared with strong FedLoRA baselines.

1 INTRODUCTION

Pre-trained foundation models (FMs) have become the cornerstone of generative AI tasks across natural language processing (NLP) and computer vision (CV) domains Vaswani et al. (2017); Grattafiori et al. (2024); Dosovitskiy et al. (2021). Fine-tuning these models has emerged as a standard paradigm for efficiently adapting them to diverse downstream tasks. However, high-quality public data are increasingly scarce, and privacy concerns over training on private data continue to grow. Federated learning (FL) McMahan et al. (2017) enables privacy-preserving collaboration to effectively leverage distributed private data for training high-quality models. Therefore, FL has been combined with fine-tuning of FMs in recent works Hu et al. (2022); Zhang et al. (2024); Chen et al. (2025); Guo et al. (2025); Sun et al. (2024); Singhal et al. (2025b).

However, the prohibitive communication overhead due to the scale of FMs has motivated the development of the federated low-rank adaptation (FedLoRA) framework Hu et al. (2022); Zhang et al. (2024). Existing studies primarily focus on improving performance Chen et al. (2025); Guo et al. (2025); Bian et al. (2026) or mitigating aggregation bias Sun et al. (2024); Singhal et al. (2025b). Despite their effectiveness, they typically assume homogeneous LoRA ranks across clients.

In practice, clients in FL exhibit inherent heterogeneity in both system resources and data distributions. Client resources vary widely in computational capability, memory size, and communication bandwidth, while client data are typically non-independent and non-identically distributed (non-IID) across domains. In this context, the LoRA rank inherently controls the trade-off between resource usage and adaptation capacity. For example, in centralized fine-tuning of LLaMA-3.1-8B Grattafiori et al. (2024) on MetaMathQA40K Yu et al. (2024) and evaluation on GSM8K Cobbe et al. (2021), increasing the LoRA rank from 8 to 256 enlarges the update size from 13 MB to 416 MB, while improving accuracy from 70.3% to 74.1%. Consequently, heterogeneous LoRA ranks offer a principled way to accommodate client heterogeneity in FL Tran et al. (2026).

Several works have explored FedLoRA with heterogeneous ranks through aggregation and rank allocation across clients Cho et al. (2024); Byun & Lee (2025); Wang et al. (2024); Bai et al. (2024);

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.

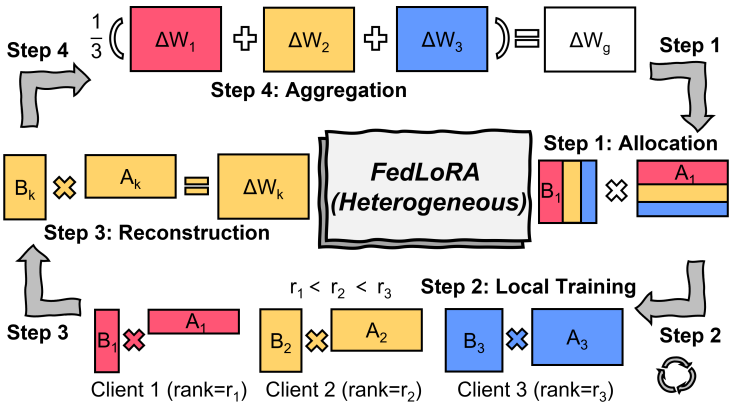
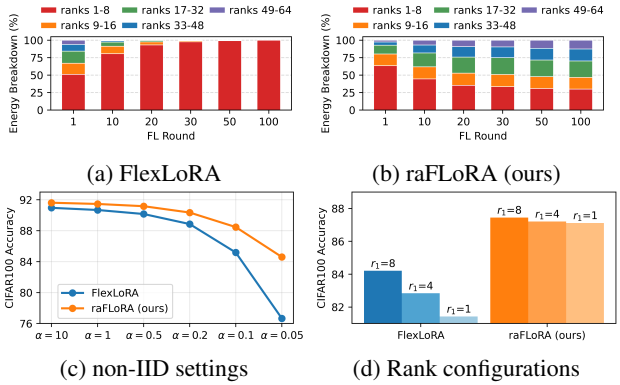


Figure 1: The overview of heterogeneous-rank FedLoRA with SVD-based allocation.

Xia et al. (2026). Among them, FlexLoRA Bai et al. (2024) leverages SVD-based allocation to avoid aggregation bias in HetLoRA Cho et al. (2024) caused by aggregating the B and A matrices separately, while maintaining communication efficiency. Figure 1 provides an overview of its framework.

Nevertheless, our observation identifies a previously overlooked phenomenon in the state-of-the-art FlexLoRA: although higher-rank clients contribute more parameters during training, almost all of the energy[1] of the global update is captured by the shared rank, as illustrated in Figure 2a. We term this behavior *rank collapse*, where the collapse reflects the concentration of the energy spectrum, instead of a reduction in algebraic rank. Consequently, the additional adaptation capacity introduced by higher-rank clients is not effectively translated into global model gains, leading to suboptimal performance across non-IID settings and high sensitivity to the shared rank, as shown in Figures 2c and 2d. Therefore, this study aims to address the following question:



(a) FlexLoRA (b) raFLoRA (ours)

(c) non-IID settings (d) Rank configurations

Figure 2: Energy breakdown of global update and accuracy under various settings. In (a) and (b), the global update has an algebraic rank of 64. Client ranks are selected from $\{8, 16, 32, 48, 64\}$. In (c), α controls the degree of data heterogeneity across clients. In (d), only the shared rank r_1 varies across different settings, with details in Appendix G.

How can we prevent rank collapse and improve the performance of heterogeneous-rank FedLoRA?

Addressing this question poses two non-trivial challenges: (i) Despite the empirical evidence, there remains a lack of principled understanding of why rank collapse arises under joint data and rank heterogeneity, where the aggregation and allocation processes across heterogeneous clients are tightly coupled. (ii) Translating this understanding into a heterogeneous FedLoRA framework is challenging because preventing rank collapse may require the aggregation rule to account for how heterogeneous ranks shape client updates. Together, these challenges call for a deeper understanding of rank collapse and a new rank-aware aggregation principle for heterogeneous-rank FedLoRA.

To address these challenges, we conduct a theoretical analysis and reveal the underlying mechanism of rank collapse as a rank-wise averaging mismatch in heterogeneous-rank aggregation: the same aggregation weight is applied to all rank directions, while higher-rank directions (non-shared) are supported by only a subset of clients. As a result, updates along higher-rank directions are systematically diluted under uniform averaging. More importantly, this dilution accumulates across successive rounds, geometrically suppressing higher-rank contributions until they become negligible.

[1]In this paper, the term "energy" refers to the squared singular values of a matrix, not physical energy consumption at devices.

Table 1: Comparison of federated low-rank adaptation methods. d and n denote the input and output dimensions, r denotes the LoRA rank, and M denotes the number of selected clients per round.

Methods	Heterogeneous Rank Support	Bias-Free Aggregation	Communication Overhead	Rank-aware Aggregation
FedIT Zhang et al. (2024)	✗	✗	$\mathcal{O}((d+n)r)$	N/A
HetLoRA Cho et al. (2024)	✓	✗	$\mathcal{O}((d+n)r)$	✗
FLoRA Wang et al. (2024)	✓	✓	$\mathcal{O}(\min(M(d+n)r, dn))$	✗
FlexLoRA Bai et al. (2024)	✓	✓	$\mathcal{O}((d+n)r)$	✗
raFLoRA (ours)	✓	✓	$\mathcal{O}((d+n)r)$	✓

Motivated by this insight, we propose raFLoRA, a rank-partitioned aggregation method. Specifically, it decomposes local updates into non-overlapping rank partitions and aggregates each part using weights determined by its rank-wise effective contributors. By correcting the mismatch, raFLoRA eliminates the per-round dilution of higher-rank updates and thereby prevents rank collapse. Table 1 summarizes the key differences between raFLoRA and baselines. Figure 2b shows raFLoRA reshapes the energy structure, and Figures 2c and 2d demonstrate its improved performance and enhanced robustness across diverse non-IID data distributions and various rank configurations.

We summarize the main contributions of this work below:

- We identify a previously overlooked phenomenon in SVD-based heterogeneous FedLoRA, termed *rank collapse*, reveal its underlying mechanism as a mismatch between rank-agnostic aggregation weights and rank-dependent client contributions, and prove that the collapse proceeds geometrically.
- We propose raFLoRA, a novel rank-partitioned aggregation method that resolves the mismatch by aligning aggregation weights with rank-wise effective client contributions.
- We empirically demonstrate that raFLoRA effectively prevents rank collapse, consistently improves global model performance, and enhances robustness across diverse tasks and non-IID data distributions, while maintaining communication efficiency.

2 Related Work

Homogeneous FedLoRA. To reduce communication in federated fine-tuning for FMs, FedIT Zhang et al. (2024) integrates LoRA into local training. Building on this, RoLoRA Chen et al. (2025) improves convergence through alternating optimization, while FedSA-LoRA Guo et al. (2025) decouples global and personalized LoRA into two components. To address data heterogeneity, SLoRA Babakniya et al. (2023) adopts a two-stage initialization strategy, and FRLoRA Yan et al. (2025) increases effective update rank through residual-based updates. FedALT Bian et al. (2026) reduces cross-client interference via adaptive LoRA mixing. To mitigate aggregation bias, FFA-LoRA Sun et al. (2024) updates only the LoRA B matrix, and FedEx-LoRA Singhal et al. (2025b) applies local bias correction. However, these methods are limited to homogeneous FedLoRA settings.

Heterogeneous FedLoRA. In practical FL scenarios, client heterogeneity motivates heterogeneous FedLoRA ranks across clients. Zero-padding-based HetLoRA Cho et al. (2024) and replication-based padding Byun & Lee (2025) enable aggregation across heterogeneous ranks, but introduce aggregation bias. To address this issue, FLoRA Wang et al. (2024) adopts stacking-based aggregation to avoid factor-wise aggregation bias, at the cost of additional communication overhead and cold-start LoRA initialization. SeFoRA Xia et al. (2026) instead aggregates heterogeneous-rank updates via linear sketches. FlexLoRA Bai et al. (2024) reconstructs full-size updates for aggregation and uses SVD to reassign rank-specific LoRA updates without additional communication overhead. Fed-PLoRA Zhang et al. (2026), ECLoRA Ning et al. (2025), and FedHL Peng et al. (2025) further focus on mitigating the bias introduced by SVD. However, our analysis reveals that SVD-based heterogeneous FedLoRA suffers from rank collapse, which raFLoRA prevents through rank-partitioned aggregation to improve performance and robustness.

3 Preliminaries

In this section, we formalize heterogeneous FedLoRA with SVD-based rank allocation, using FlexLoRA Bai et al. (2024) as a representative framework that enables aggregation in the full parameter space while preserving communication efficiency.

Consider K clients with heterogeneous LoRA ranks $r_k \in \{r_1, r_2, \ldots, r_H\}$, where $r_1 < r_2 < \cdots < r_H = r_{\max}$. At round t, the server uniformly samples M clients $\mathcal{M}_t$. We define the *rank coverage* of direction i as $p_i = \mathbb{P}(r_k \geq i) = |\mathcal{K}_i|/K$, where $\mathcal{K}_i = \{k : r_k \geq i\}$, yielding

$$p_1 = \cdots = p_{r_1} = 1 > p_{r_1+1} \geq \cdots \geq p_{r_{\max}} > 0. \tag{1}$$

Thus, higher-rank directions are supported by progressively fewer clients.

For a pre-trained layer $W_{\text{pre}} \in \mathbb{R}^{d\times n}$, the server maintains a rank-$r_{\max}$ global update $\Delta W_g^{(t)} \approx B_g^{(t)} A_g^{(t)}$, where $B_g^{(t)} \in \mathbb{R}^{d\times r_{\max}}$ and $A_g^{(t)} \in \mathbb{R}^{r_{\max}\times n}$. Each selected client k receives the first r_k components of the global LoRA factors, trains them on its local data, and returns $\Delta W_k^{(t)} = B_k^{(t)} A_k^{(t)}$. The local updates are aggregated in the full $d \times n$ space using FedAvg McMahan et al. (2017)

$$\Delta W_g^{(t+1)} = \frac{1}{M} \sum_{k\in\mathcal{M}_t} \Delta W_k^{(t)}, \tag{2}$$

where we assume equal client sample sizes for analytical simplicity. The aggregated update is then decomposed as $\Delta W_g^{(t+1)} \approx \sum_{i=1}^{r_{\max}} \sigma_i^{(t+1)} u_i^{(t+1)} v_i^{(t+1)\top}$ via truncated SVD Bai et al. (2024), from which the global LoRA factors are reconstructed for the next round.

To characterize how heterogeneous ranks affect the global update, we define the expected energy of singular direction i as $e_i^{(t)} = \mathbb{E}[(\sigma_i^{(t)})^2]$, the cumulative energy as $E_r^{(t)} = \sum_{i=1}^{r} e_i^{(t)}$, and the normalized energy ratio as $\rho_r^{(t)} = E_r^{(t)}/E_{r_{\max}}^{(t)}$. We refer to the first r_1 directions as the *shared rank* and the remaining $r_{\max} - r_1$ directions as the *higher ranks*. Accordingly, $\rho_{r_1}^{(t)}$ measures the energy fraction retained in the shared rank, while $1 - \rho_{r_1}^{(t)}$ measures that in the higher ranks.

4 Problem Analysis

Based on the above formulation, we define rank collapse and analyze its emergence in SVD-based heterogeneous FedLoRA, providing theoretical insights into the underlying mechanism.

4.1 Definition of Rank Collapse

According to our observations in Figure 2a, although the global update is decomposed into rank-$r_{\max}$ components, its singular-value energy increasingly concentrates on the shared rank r_1 over rounds. This observation motivates the following definition:

Definition 1 (Rank Collapse)**.** *In SVD-based heterogeneous FedLoRA, we define rank collapse to occur when $1 - \rho_{r_1}^{(t)}$ progressively diminishes and becomes negligible over training rounds.*

Under rank collapse, although the global update retains algebraic rank $r_{\max}$ after SVD, its effective rank no longer reflects the available higher-rank capacity. Instead, the learning dynamics are governed by the shared rank r_1, rendering higher-rank components progressively ineffective.

4.2 Theoretical Analysis: From Contributor Dilution to Rank Collapse

We next identify the mechanism that drives rank collapse. Our analysis separates an exact one-round effect from its multi-round accumulation, and then extends the analysis to more general settings.

One-round contributor dilution. Let $r_0 = 0$ and define the h-th rank partition as $\mathcal{I}_h = \{r_{h-1} + 1, \ldots, r_h\}$. For partition h, let $\mathcal{C}_h^{(t)} = \{k \in \mathcal{M}_t : r_k \geq r_h\}$ and $C_h^{(t)} = |\mathcal{C}_h^{(t)}|$ denote its contributor set and contributor count. When $C_h^{(t)} > 0$, define its contributor average as $\overline{\Delta W}_h^{(t)} = (C_h^{(t)})^{-1} \sum_{k\in\mathcal{C}_h^{(t)}} B_k^{(t)}[:, \mathcal{I}_h] A_k^{(t)}[\mathcal{I}_h, :]$. Rearranging Eq. 2 by rank partition gives (see Appendix A for the full derivation)

$$\Delta W_g^{(t+1)} = \sum_{h=1}^{H} \frac{C_h^{(t)}}{M} \overline{\Delta W}_h^{(t)}, \tag{3}$$

where the global update is decomposed into contributions from individual rank partitions

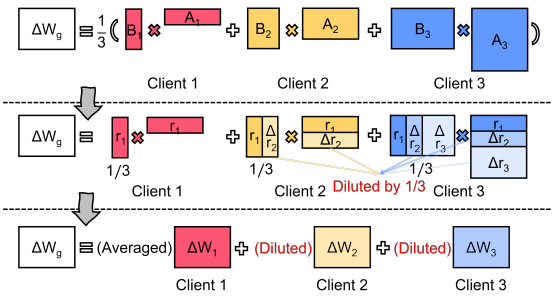


Figure 3: Illustration of contributor dilution.

As illustrated in Figure 3, different rank partitions have different numbers of effective contributors, whereas conventional FedAvg McMahan et al. (2017) aggregation normalizes all partitions by the same number of participating clients in each round. We refer to this as a mismatch between rank-agnostic aggregation weights and rank-dependent client contributions. Eq. 3 reveals its direct consequence, namely one-round *contributor dilution*. The shared partition has $N_1^{(t)} = M$, whereas a higher-rank partition is scaled by $C_h^{(t)}/M < 1$ whenever only a subset of sampled clients supports that partition. Under uniform client sampling without replacement, $\mathbb{E}[C_h^{(t)}/M] = p_{r_h}$, where p_i denotes the corresponding rank coverage defined in Section 3.

Multi-round accumulation. To clearly characterize how the one-round dilution accumulates across communication rounds, we introduce a simplifying assumption that isolates the evolution of rank-wise energy and yields a tractable closed-form recursion.

Assumption 1 (Stable spectral routing)**.** *For the analytical model, we assume that the retained singular directions remain fixed across rounds without singular-value crossings, and that local training preserves these directions, with each eligible client rescaling direction i by a gain $\beta_{k,i}^{(t)}$.*

Let $c_{k,i} = \mathbf{1}\{r_k \geq i\}$ indicate whether client k supports direction i. Under Assumption 1, the aggregated gain is $a_i^{(t)} = M^{-1} \sum_{k \in \mathcal{M}_t} c_{k,i} \beta_{k,i}^{(t)}$, giving $\sigma_i^{(t+1)} = |a_i^{(t)}| \sigma_i^{(t)}$. Defining $q_i^{(t)} = \mathbb{E}[(a_i^{(t)})^2]$ and taking expectation over the round-t client sampling yields $e_i^{(t+1)} = q_i^{(t)} e_i^{(t)}$. For uniform sampling without replacement, $q_i^{(t)} = (1-\tau) p_i^2 (\mu_i^{(t)})^2 + \tau p_i \nu_i^{(t)}$, where $\tau = (K-M)/(M(K-1))$, and $\mu_i^{(t)}$ and $\nu_i^{(t)}$ are the first and second moments of the local gains among clients supporting direction i. Thus, lower contributor coverage reduces the effective energy multiplier, allowing one-round dilution to accumulate across rounds when the retention gap persists, as formalized in Theorem 1. A complete derivation of the multi-round recursion and the proof of the theorem are provided in Appendix B.

Theorem 1 (Geometric spectral concentration)**.** *Under Assumption 1, let $q_{\text{high}}^{(t)} = \max_{i > r_1} q_i^{(t)}$ and $q_{\text{shared}}^{(t)} = \min_{i \leq r_1} q_i^{(t)}$. If $\gamma := \sup_t q_{\text{high}}^{(t)} / q_{\text{shared}}^{(t)} < 1$, then*

$$1 - \rho_{r_1}^{(T)} \leq \frac{1 - \rho_{r_1}^{(0)}}{\rho_{r_1}^{(0)}} \gamma^T. \tag{4}$$

Hence, the higher-rank energy fraction vanishes geometrically as $T \to \infty$, yielding rank collapse.

Intuitively, higher-rank directions are supported by fewer clients and thus tend to have smaller effective energy multipliers. Unless stronger local gains offset this lower coverage, the gap compounds across rounds, geometrically concentrating spectral energy in the shared rank r_1.

Beyond stable spectral routing. Assumption 1 is required only for the tractable multi-round recursion across communication rounds. The one-round dilution in Eq. 3 remains exact without it. In practice, SVD reordering can move spectral components between rank positions with different contributor coverage, while basis drift and cross-direction mixing further redistribute energy across directions. As derived in Appendix C, these effects yield

$$e_i^{(t+1)} = \mathbb{E}\left[(x_{(i)}^{(t)})^2\right] + \bar{\xi}_i^{(t)}, \tag{5}$$

where $x_{(i)}^{(t)}$ is the i-th singular magnitude of the reordered aligned component and $\bar{\xi}_i^{(t)}$ captures the resulting perturbation induced by basis drift and cross-direction mixing. When no crossing occurs and $R_t = 0$, we have $x_{(i)}^{(t)} = |a_i^{(t)}| \sigma_i^{(t)}$ and $\bar{\xi}_i^{(t)} = 0$, exactly recovering the stable-routing recursion. Thus, spectral reordering and basis drift perturb the exact geometric recursion in Theorem 1, while the coverage-dependent one-round dilution mechanism remains unchanged.

5 raFLoRA Method

The analysis identifies the rank-wise averaging mismatch as the underlying mechanism of rank collapse. Motivated by this insight, we propose a new aggregation strategy to correct this mismatch. We formally introduce raFLoRA, which performs **r**ank-partitioned **a**ggregation for **F**ederated **Lo**w-**R**ank **A**daptation with client heterogeneity. Unlike conventional FedAvg McMahan et al. (2017), which uniformly averages all uploaded updates, raFLoRA partitions the local updates into independent rank-wise components, with aggregation weights aligned to effective rank-wise contributors.

Following the rank partitions defined in Section 4.2, let $\mathcal{I}_h = \{r_{h-1}+1, \ldots, r_h\}$ for $h = 1, \ldots, H$, where $r_0 = 0$ and $r_H = r_{\max}$. The full rank $r_{\max}$ is thus partitioned into non-overlapping rank partitions. As illustrated in Figure 4, when three clients have local ranks r_1, r_2, and r_3, these boundaries divide the full rank into three partitions $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{I}_3$.

For each rank partition $\mathcal{I}_h$, aggregation uses only the clients whose local rank satisfies $r_k \geq r_h$, so that global aggregation weights reflect the clients that effectively contribute information at this rank. Following Section 4.2, we denote this set of effective contributors by $\mathcal{C}_h^{(t)} = \{k \in \mathcal{M}_t \mid r_k \geq r_h\}$, with the corresponding total sample size $N_h^{(t)} = \sum_{k \in \mathcal{C}_h^{(t)}} n_k$, where n_k denotes the local sample size of client k. As illustrated in Figure 4, all clients contribute to the shared-rank partition $\mathcal{I}_1 = \{1, \ldots, r_1\}$, fewer clients contribute to the intermediate partition $\mathcal{I}_2 = \{r_1+1, \ldots, r_2\}$, and only the highest-rank client contributes to the last partition $\mathcal{I}_3 = \{r_2+1, \ldots, r_3\}$.

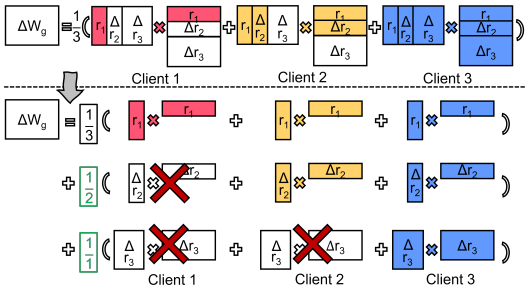


Figure 4: Overview of the rank-partitioned aggregation.

Given a specific rank partition $\mathcal{I}_h$, its contribution at round t is defined as

$$\Delta W_{g,h}^{(t+1)} = \begin{cases} \sum_{k \in \mathcal{C}_h^{(t)}} \frac{n_k}{N_h^{(t)}} \left(B_k^{(t)}[:, \mathcal{I}_h] \, A_k^{(t)}[\mathcal{I}_h, :] \right), & \mathcal{C}_h^{(t)} \neq \emptyset, \\ B_g^{(t)}[:, \mathcal{I}_h] \, A_g^{(t)}[\mathcal{I}_h, :], & \mathcal{C}_h^{(t)} = \emptyset. \end{cases} \tag{6}$$

where $[:, \mathcal{I}_h]$ selects all rows and the rank indices in $\mathcal{I}_h$ of $B_k^{(t)}$, and $[\mathcal{I}_h, :]$ selects the corresponding rank indices and all columns of $A_k^{(t)}$. If no sampled client covers this rank partition in a given round, i.e., $\mathcal{C}_h^{(t)} = \emptyset$, we fall back to the current global LoRA update for this partition rather than skipping it. This fallback prevents previously retained higher-rank information from being discarded. Appendix D further derives how this design removes the coverage-dependent attenuation identified in Section 4.2 while preserving previously retained information when a partition has no contributors.

The global update is then obtained by summing the contributions from all rank partitions, $\Delta W_g^{(t+1)} = \sum_{h=1}^{H} \Delta W_{g,h}^{(t+1)}$. Accordingly, in the example illustrated in Figure 4, the aggregation weights are $1/3$ for the shared-rank partition where three clients contribute, $1/2$ for the second partition where two clients contribute, and 1 for the last partition where only a single client provides updates. After aggregation, $\Delta W_g^{(t+1)}$ is projected via SVD to obtain $B_g^{(t+1)}$ and $A_g^{(t+1)}$.

Algorithm 1 summarizes the raFLoRA workflow, which incorporates rank-partitioned aggregation with heterogeneous client ranks. In each round, the server uniformly samples clients, broadcasts the global LoRA updates, and collects local updates in parallel (Lines 3-5). The server then performs rank-partitioned aggregation, aggregating each rank partition using only its effective contributors (Lines 6-11). An SVD then yields the global low-rank updates for the next round (Line 12).

For L LoRA layers, M participating clients, and H rank partitions, reconstructing local updates has the same dominant cost as FlexLoRA, i.e., $O(L \sum_{k=1}^{M} dnr_k) = O(LMdn\bar{r})$, where $\bar{r}$ is the average client rank. The partition-wise aggregation introduces an additional cost of $O(LHMdn)$, leading to a total complexity of $O(LMdn\bar{r}(1 + H/\bar{r}))$. Since $H \ll \bar{r}$ in practice, this extra overhead remains limited. In the next section, we empirically demonstrate the effectiveness of raFLoRA.

Algorithm 1 Rank-partitioned Aggregation for Federated LoRA with Client Heterogeneity

Input: Rounds T; Clients K; participation ratio per round ρ; ranks $r_k \in \{r_1, r_2, \ldots, r_H\}$ with $r_H = r_{\max}$; $W_{\text{pre}} \in \mathbb{R}^{d\times n}$; initial global LoRA updates $(B_g^{(0)}, A_g^{(0)})$ with $B_g^{(0)} \in \mathbb{R}^{d\times r_{\max}}$, $A_g^{(0)} \in \mathbb{R}^{r_{\max}\times n}$
for $t = 0$ **to** $T-1$ **do**
  Sample participating clients $\mathcal{M}_t \subseteq \{1, \ldots, K\}$ uniformly with $|\mathcal{M}_t| = M$
  Only broadcast LoRA updates for each client $k \in \mathcal{M}_t$: $\widetilde{B}_k^{(t)} = B_g^{(t)}[:, : r_k], \widetilde{A}_k^{(t)} = A_g^{(t)}[: r_k, :]$
  In parallel, each client $k \in \mathcal{M}_t$ trains from $(\widetilde{B}_k^{(t)}, \widetilde{A}_k^{(t)})$ and uploads $(B_k^{(t)}, A_k^{(t)})$
  Rank-partitioned Aggregation at Server: Initialization $\Delta W_g^{(t+1)} = 0$
  for $h = 1$ **to** H **do**
    $\mathcal{C}_h^{(t)} \leftarrow \{k \in \mathcal{M}_t : r_k \geq r_h\}, \quad N_h^{(t)} = \sum_{j\in\mathcal{C}_h^{(t)}} n_j$
    $\Delta W_{g,h}^{(t+1)} \leftarrow \begin{cases} \sum_{k\in\mathcal{C}_h^{(t)}} \frac{n_k}{N_h^{(t)}} \big(B_k^{(t)}[:, \mathcal{I}_h] A_k^{(t)}[\mathcal{I}_h, :]\big), & \mathcal{C}_h^{(t)} \neq \emptyset, \\ B_g^{(t)}[:, \mathcal{I}_h] A_g^{(t)}[\mathcal{I}_h, :], & \mathcal{C}_h^{(t)} = \emptyset. \end{cases}$
    $\Delta W_g^{(t+1)} \leftarrow \Delta W_g^{(t+1)} + \Delta W_{g,h}^{(t+1)}$
  end for
  Decompose $\Delta W_g^{(t+1)}$ to $(B_g^{(t+1)}, A_g^{(t+1)})$ via SVD
end for
return $W_{\text{pre}} + B_g^{(T)} A_g^{(T)}$

6 Experiments

In this section, we present the experimental setup, then we evaluate raFLoRA in terms of accuracy, rank collapse prevention, efficiency, sensitivity & robustness, and extended settings.

6.1 Experimental Setup

The experimental setup specifies the models and datasets for each task, metrics for evaluation, non-IID data partitioning, baselines, and hyperparameter settings. All experiments are performed on a single GPU, using an NVIDIA RTX 5090 (32 GB) or an NVIDIA H100-SXM (80 GB). Additional implementation details and more hyperparameter settings are provided in Appendix G.

Models and Datasets. We evaluate encoder-only and decoder-only models across image classification, text classification, mathematical reasoning, and commonsense reasoning. We use ViT-base Dosovitskiy et al. (2021) on CIFAR100 Krizhevsky & Hinton (2009), RoBERTa-base Liu et al. (2019) on 20 Newsgroups Lang (1995), and LLaMA-3.2-3B Meta AI (2024) and LLaMA-3.1-8B Grattafiori et al. (2024) on GSM8K Cobbe et al. (2021). For commonsense reasoning, the two LLaMA models are fine-tuned on Commonsense15K Hu et al. (2023) and evaluated on eight benchmarks.

Metrics. We report test accuracy as mean $\pm$ standard deviation over three random seeds and the energy ratio. Communication cost is measured by the total upload and download volume per client per round, while computational cost is measured by total training runtime.

Data Partitioning. We consider IID and multiple non-IID partitions with the following default settings. For vision and NLU tasks, we use Dirichlet partitioning, where smaller α indicates stronger heterogeneity, and pathological partitioning Zhang et al. (2025), where each client accesses a subset of labels. For example, c20($\alpha = 1$) denotes 20 labels per client with $\alpha = 1$. GSM8K is uniformly partitioned, while Commonsense15K is partitioned by answer type with $\alpha = 0.5$.

Baselines. We compare against state-of-the-art heterogeneous-rank FedLoRA methods. HetLoRA Cho et al. (2024) aligns heterogeneous ranks through zero padding. FLoRA Wang et al. (2024) aggregates heterogeneous updates by stacking and broadcasts the stacked updates to clients. FlexLoRA Bai et al. (2024) uses bias-free aggregation and allocates heterogeneous ranks via SVD. For a fair comparison, the same fallback mechanism is applied to all baselines.

Hyperparameters. Unless otherwise specified, we use 100 clients with 10% participation per round, AdamW, and one local epoch. The learning rate starts at 5×10^{-4} and decays linearly. LoRA ranks are uniformly sampled from $\{8, 16, 32, 48, 64\}$, with LoRA alpha set to r_k for unit scaling. Vision and NLU tasks run for 100 rounds with LoRA applied to all linear layers, while reasoning tasks run for 20 rounds with LoRA applied only to the Q and V modules.

Table 2: Accuracy comparison across diverse tasks. Accuracy is reported in (%), with the best results highlighted in bold. Results are obtained using ViT-base for vision, RoBERTa-base for text, LLaMA-3.2-3B/LLaMA-3.1-8B for mathematical and commonsense reasoning.

Methods	Vision **CIFAR100**	Language **20NG**	Mathematical Reasoning **GSM8K**	Commonsense Reasoning **Commonsense15K**
HetLoRA	$84.04_{\pm 1.37}$	$62.06_{\pm 0.67}$	$36.75_{\pm 1.23}$/$56.43_{\pm 0.29}$	$71.89_{\pm 0.41}$/$79.45_{\pm 1.00}$
FLoRA	$86.30_{\pm 0.79}$	$61.91_{\pm 0.52}$	$36.09_{\pm 0.57}$/$56.25_{\pm 0.85}$	$71.02_{\pm 0.18}$/$79.01_{\pm 0.38}$
FlexLoRA	$84.02_{\pm 1.17}$	$63.05_{\pm 0.83}$	$40.21_{\pm 0.42}$/$58.43_{\pm 0.74}$	$74.33_{\pm 0.50}$/$80.88_{\pm 0.67}$
raFLoRA	**86.59**$_{\pm 0.75}$	**64.80**$_{\pm 0.34}$	**41.72**$_{\pm 0.70}$/**59.06**$_{\pm 0.27}$	**74.86**$_{\pm 0.39}$/**81.15**$_{\pm 0.08}$

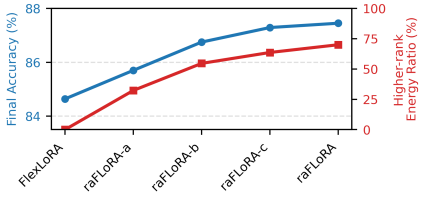
6.2 Accuracy across Diverse Tasks

We evaluate raFLoRA across vision, language, and reasoning tasks, with data partitioning and hyperparameter details provided in Appendix G. As shown in Table 2, **raFLoRA delivers consistent improvements over baselines across a wide range of tasks.** Specifically, raFLoRA outperforms FlexLoRA and HetLoRA by about 2.6% on CIFAR100, and improves over FLoRA by nearly 3.0% on 20NG. On more challenging reasoning tasks, raFLoRA also achieves higher performance on GSM8K with both LLaMA-3.2-3B and LLaMA-3.1-8B models. For commonsense reasoning, raFLoRA achieves the highest average accuracy across eight sub-datasets, with detailed results in Appendix H. FLoRA's weaker performance on some tasks may stem from its cold-start behavior, where LoRA parameters are reinitialized each round. Additional training dynamics and generality evaluations across other heterogeneous FedLoRA methods are provided in Appendices F and I, respectively.

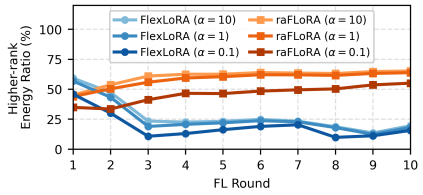
6.3 Rank Collapse Prevention

We construct partial variants (raFLoRA-a/b/c) on CIFAR100, applying effective-contributor weighting up to rank partitions $\{8, 16\}$, $\{8, 16, 32\}$, and $\{8, 16, 32, 48\}$, respectively, while retaining baseline aggregation for the remaining partitions. **By preserving higher-rank energy, rank-partitioned aggregation prevents rank collapse and improves final performance.** As shown in Figure 5a, applying rank-partitioned aggregation to more partitions improves higher-rank energy preservation and is accompanied by better accuracy. A post-SVD alternative is evaluated in Appendix J.

Additionally, we analyze the higher-rank energy ratio dynamics of FlexLoRA and raFLoRA under varying data heterogeneity on CIFAR100. As shown in Figure 5b, stronger data heterogeneity reduces the preservation of higher-rank energy. This observation supports our analysis in Section 4, indicating that data heterogeneity weakens the alignment and effective accumulation of higher-rank updates during aggregation.



(a) Rank collapse prevention



(b) Energy under data heterogeneity

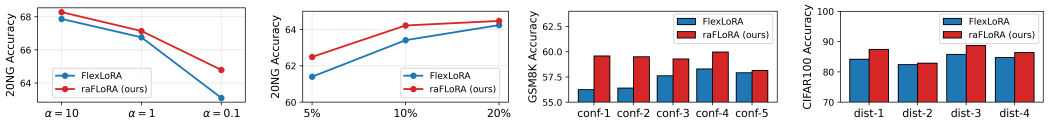
Figure 5: Rank collapse prevention and higher-rank energy ratio dynamics.

6.4 Communication and Computation Costs

We evaluate the communication and computational costs of different methods during fine-tuning. As shown in Table 3, **raFLoRA maintains competitive communication efficiency while introducing only modest additional computational overhead**. Compared with FLoRA, raFLoRA reduces the communication cost to about 18% on both ViT-base and LLaMA-3.1-8B, as it avoids synchronizing the full global update. Although rank-partitioned aggregation slightly increases runtime, the overhead remains controlled and is consistent with the complexity analysis in Section 5.

Table 3: Comparison of total training runtime and average communication cost per client per round.

Methods	ViT-base		LLaMA-3.1-8B	
	Comp.	**Comm.**	**Comp.**	**Comm.**
HetLoRA	40m50s	41MB	1h33m55s	109MB
FLoRA	47m05s	226MB	1h40m11s	580MB
FlexLoRA	40m40s	41MB	1h38m09s	109MB
raFLoRA	42m55s	41MB	1h50m56s	109MB



(a) non-IID (b) Client Participation (c) Rank-configs (d) Distributions

Figure 6: Sensitivity and robustness analyses of FlexLoRA and raFLoRA under different settings. The detailed configurations for (c) and (d) are provided in Section 6.5.

6.5 Sensitivity and Robustness Analyses

Since rank collapse increases sensitivity to rank configurations and data heterogeneity, we compare raFLoRA and the strongest baseline FlexLoRA across diverse heterogeneous settings.

Effect of non-IID settings. We evaluate CIFAR100 and 20NG under varying data heterogeneity. Figures 2c and 6a show that raFLoRA is more robust than FlexLoRA. As heterogeneity increases, raFLoRA mitigates this degradation and consistently achieves higher accuracy.

Effect of client participation rates. We evaluate 20NG under varying client participation rates. Figure 6b shows that raFLoRA consistently outperforms FlexLoRA, with accuracy first increasing and then stabilizing as participation increases. With only one participating client, raFLoRA reduces to FlexLoRA because no rank-wise dilution occurs. The reliability of sparsely supported higher-rank partitions is further discussed in Appendix K.

Effect of rank configurations. We evaluate CIFAR100 and GSM8K under five rank configurations: conf-1 to conf-5 correspond to $\{1, 16, 32, 48, 64\}$, $\{4, 16, 32, 48, 64\}$, $\{8, 16, 32, 48, 64\}$, $\{8, 16, 32, 48, 96\}$, and $\{8, 16, 32, 48, 128\}$, respectively. Figures 2d and 6c show that raFLoRA consistently outperforms FlexLoRA and avoids its pronounced sensitivity to the shared rank. Additional rank configurations and LoRA module insertion settings are evaluated in Appendix E.

Effect of rank distributions. We evaluate CIFAR100 with ranks $\{8, 16, 32, 48, 64\}$ under four distributions: uniform (dist-1: $\{0.2, 0.2, 0.2, 0.2, 0.2\}$), low-rank skewed (dist-2: $\{0.7, 0.1, 0.1, 0.05, 0.05\}$), high-rank skewed (dist-3: $\{0.05, 0.05, 0.1, 0.1, 0.7\}$), and bell-shaped (dist-4: $\{0.05, 0.1, 0.7, 0.1, 0.05\}$). Figure 6d shows that raFLoRA outperforms FlexLoRA under most distributions and remains comparable under dist-2. This suggests that its benefit in preserving the hierarchical rank-energy structure diminishes when most clients use low ranks.

Table 4: Performance under Gaussian noise.

Methods	$\nu = 0.0$	$\nu = 0.1$	$\nu = 0.3$	$\nu = 0.5$
FlexLoRA	83.94%	85.15%	82.67%	81.45%
raFLoRA	**87.43%**	**87.47%**	**86.91%**	**86.61%**

Table 5: Extension to PEFT variants.

Methods	QLoRA	AdaLoRA	DoRA
FlexLoRA	85.66%	87.34%	73.88%
raFLoRA	**86.53%**	**88.73%**	**86.63%**

6.6 Extended Experiments

Extension to noisy low-rank clients. To evaluate the case where low-rank clients (rank=8) also have lower-quality data sources, we inject zero-mean Gaussian noise $\epsilon \sim \mathcal{N}(0, \nu^2)$ into their data on CIFAR100, where ν denotes the noise standard deviation. As shown in Table 4, raFLoRA consistently outperforms FlexLoRA as the noise level increases, suggesting robustness to noisy low-rank clients.

Extension to LoRA variants. We extend raFLoRA to QLoRA Dettmers et al. (2023), AdaLoRA Zhang et al. (2023), and DoRA Liu et al. (2024). Table 5 shows accuracy gains across variants on CIFAR100. The degradation of FlexLoRA-DoRA suggests that DoRA is sensitive to rank collapse, since magnitude reweighting cannot recover attenuated directional information, whereas raFLoRA avoids this issue by rank-partitioned aggregation.

7 Conclusion

We identify rank collapse in heterogeneous FedLoRA with SVD-based allocation and provide a theoretical analysis showing that it results from a rank-wise averaging mismatch. To address this issue, we propose raFLoRA, a rank-partitioned aggregation strategy that aligns aggregation weights with rank-wise effective contributors. Extensive experiments across diverse tasks demonstrate its effectiveness and robustness, enabling efficient and scalable adaptation of large models.

AI USE STATEMENT

In this work, we used generative AI tools to help develop theoretical models and conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, implement methods, and assist with translation. We did not use generative AI tools to generate synthetic datasets, propose or refine research hypotheses, design or provide feedback on research methodology or experiments, clean or reformat datasets, support qualitative or thematic data analysis, or interpret results. Additionally, we used generative AI tools for code inspection and to generate scripts for running multiple experiments.

All AI-assisted content was reviewed and independently verified by the authors. In particular, all AI-assisted mathematical arguments and proofs were checked for correctness, all LLM-polished text was manually reviewed for factual accuracy and to ensure that no fabricated information was introduced, and all AI-generated code and scripts were tested and verified by the authors. We take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the implementation of raFLoRA and the experimental evaluation framework in the supplementary materials. The derivation of the one-round contributor dilution is provided in Appendix A. The assumptions and complete proof of Theorem 1 are given in Appendix B, with an analysis beyond stable spectral routing in Appendix C. Appendix D further analyzes how raFLoRA removes the coverage-dependent attenuation identified by the theory. Detailed experimental settings and data processing procedures are provided in Appendix G.

REFERENCES

Sara Babakniya, Ahmed Elkordy, Yahya 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 in Conjunction with NeurIPS 2023*, 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. *Advances in Neural Information Processing Systems*, 37:14457–14483, 2024.

Jieming Bian, Lei Wang, Letian Zhang, and Jie Xu. Fedalt: Federated fine-tuning through adaptive local training with rest-of-world lora. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19728–19736, 2026.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Yuji Byun and Jaeho Lee. Towards federated low-rank adaptation of language models with rank heterogeneity. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 356–362, 2025.

Shuangyi Chen, Yuanxin Guo, Yue Ju, Hardik Dalal, Zhongwen Zhu, and Ashish J Khisti. Robust federated finetuning of LLMs via alternating optimization of LoRA. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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*, 2024.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 2924–2936, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *The Ninth International Conference on Learning Representations*, 2021.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Pengxin Guo, Shuang Zeng, Yanran Wang, Huijie Fan, Feifei Wang, and Liangqiong Qu. Selective aggregation for low-rank adaptation in federated learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

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 *The Tenth International Conference on Learning Representations*, 2022.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5254–5276, 2023.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, Toronto, ON, Canada, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Ken Lang. Newsweeder: learning to filter netnews. In *Proceedings of the Twelfth International Conference on International Conference on Machine Learning*, pp. 331–339, 1995.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Meta AI. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models. `https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/`, September 2024.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, 2018.

Wanyi Ning, Jingyu Wang, Qi Qi, Haifeng Sun, Daixuan Cheng, Cong Liu, Lei Zhang, Zirui Zhuang, and Jianxin Liao. Federated fine-tuning on heterogeneous loras with error-compensated aggregation. *IEEE Transactions on Neural Networks and Learning Systems*, 36(10):17826–17840, 2025.

Zihao Peng, Jiandian Zeng, Boyuan Li, Guo Li, Shengbo Chen, and Tian Wang. Fedhl: Federated learning for heterogeneous low-rank adaptation via unbiased aggregation. *arXiv preprint arXiv:2505.18494*, 2025.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473, 2019.

Raghav Singhal, Kaustubh Ponkshe, Rohit Vartak, Lav R. Varshney, and Praneeth Vepakomma. FedSB: A silver bullet for extreme communication efficiency and performance in (private) federated LoRA fine-tuning. In *ES-FoMo III: 3rd Workshop on Efficient Systems for Foundation Models*, 2025a.

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

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving LoRA in privacy-preserving federated learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Van-Tuan Tran, Hong-Hanh Nguyen-Le, Marco Ruffini, and Merim Dzaferagic. UB-SMoe: Universally balanced sparse mixture-of-experts for resource-adaptive federated fine-tuning of foundation models. In *Forty-third International Conference on Machine Learning*, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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. *Advances in Neural Information Processing Systems*, 37:22513–22533, 2024.

Yue Xia, Tayyebeh Jahani-Nezhad, Mayank Bakshi, and Rawad Bitar. Sefora: Sketch-aggregated federated low-rank adaptation with heterogeneous client ranks. *arXiv preprint arXiv:2608.10144*, 2026.

Yunlu Yan, Chun-Mei Feng, Wangmeng Zuo, Lei Zhu, Rick Siow Mong Goh, and Yong Liu. Federated residual low-rank adaptation of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023.

Zikai Zhang, Ping Liu, Jiahao Xu, and Rui Hu. Fed-hello: Efficient federated foundation model fine-tuning with heterogeneous LoRA allocation. *IEEE Transactions on Neural Networks and Learning Systems*, 36(10):17556–17569, 2025.

Zikai Zhang, Rui Hu, and Jiahao Xu. Heterogeneous federated fine-tuning with parallel one-rank adaptation. In *The Fourteenth International Conference on Learning Representations*, 2026.

A Exact One-Round Contributor Dilution

We derive Eq. 3 one-round contributor dilution. For client k, define its update in partition h as

$$\Delta W_{k,h}^{(t)} = B_k^{(t)}[:, \mathcal{I}_h] A_k^{(t)}[\mathcal{I}_h, :].$$

Since client k contains exactly the partitions satisfying $r_k \geq r_h$,

$$\Delta W_k^{(t)} = \sum_{h: r_k \geq r_h} \Delta W_{k,h}^{(t)}.$$

Substituting this decomposition into FedAvg gives

$$\Delta W_g^{(t+1)} = \frac{1}{M} \sum_{k \in \mathcal{M}_t} \sum_{h: r_k \geq r_h} \Delta W_{k,h}^{(t)} = \sum_{h=1}^{H} \frac{1}{M} \sum_{k \in \mathcal{C}_h^{(t)}} \Delta W_{k,h}^{(t)} = \sum_{h=1}^{H} \frac{C_h^{(t)}}{M} \overline{\Delta W}_h^{(t)},$$

which proves Eq. 3.

Under uniform sampling without replacement, $C_h^{(t)} \sim \mathrm{Hypergeo}(K, Kp_{r_h}, M)$, and hence

$$\mathbb{E}\left[\frac{C_h^{(t)}}{M}\right] = p_{r_h}.$$

Therefore, higher-rank partitions receive systematically smaller normalization factors whenever their contributor coverage is lower.

B Multi-Round Dynamics under Stable Spectral Routing

Under Assumption 1, direction i preserves its spectral identity across rounds. We treat the round-wise client gains as fixed with respect to the sampling history, with randomness arising from uniform client sampling. Recall

$$a_i^{(t)} = \frac{1}{M} \sum_{k \in \mathcal{M}_t} c_{k,i} \beta_{k,i}^{(t)}, \qquad \sigma_i^{(t+1)} = |a_i^{(t)}| \sigma_i^{(t)}.$$

Let

$$\mu_i^{(t)} = \mathbb{E}[\beta_{k,i}^{(t)} \mid r_k \geq i], \qquad \nu_i^{(t)} = \mathbb{E}[(\beta_{k,i}^{(t)})^2 \mid r_k \geq i].$$

For $X_{k,i}^{(t)} = c_{k,i} \beta_{k,i}^{(t)}$, rank coverage gives

$$\mathbb{E}[X_{k,i}^{(t)}] = p_i \mu_i^{(t)}, \qquad \mathbb{E}[(X_{k,i}^{(t)})^2] = p_i \nu_i^{(t)}.$$

Since $a_i^{(t)}$ is the mean of M clients sampled uniformly without replacement, letting

$$\tau = \frac{K - M}{M(K-1)}$$

gives

$$q_i^{(t)} := \mathbb{E}[(a_i^{(t)})^2] = (1-\tau) p_i^2 (\mu_i^{(t)})^2 + \tau p_i \nu_i^{(t)}. \tag{7}$$

Thus,

$$e_i^{(t+1)} = q_i^{(t)} e_i^{(t)}. \tag{8}$$

We now prove Theorem 1. Define

$$S_t = \sum_{i \leq r_1} e_i^{(t)}, \qquad H_t = \sum_{i > r_1} e_i^{(t)}.$$

Using Eq. 8,

$$H_{t+1} \leq q_{\mathrm{high}}^{(t)} H_t, \qquad S_{t+1} \geq q_{\mathrm{shared}}^{(t)} S_t.$$

Therefore,

$$\frac{H_T}{S_T} \le \frac{H_0}{S_0} \prod_{t=0}^{T-1} \frac{q_{\text{high}}^{(t)}}{q_{\text{shared}}^{(t)}} \le \frac{H_0}{S_0} \gamma^T.$$

Since

$$1 - \rho_{r_1}^{(T)} = \frac{H_T}{S_T + H_T} \le \frac{H_T}{S_T}$$

and

$$\frac{H_0}{S_0} = \frac{1 - \rho_{r_1}^{(0)}}{\rho_{r_1}^{(0)}},$$

we obtain

$$1 - \rho_{r_1}^{(T)} \le \frac{1 - \rho_{r_1}^{(0)}}{\rho_{r_1}^{(0)}} \gamma^T,$$

which proves Theorem 1.

Equation 7 separates contributor coverage from local training: lower p_i suppresses a direction, while stronger local gains can compensate. Thus, rank collapse does not require weaker higher-rank updates, but occurs when their lower coverage is insufficiently offset.

C Analysis Beyond Stable Spectral Routing

We next examine the dynamics when Assumption 1 does not hold. The one-round contributor dilution in Eq. 3 remains exact, while spectral reordering and basis drift complicate the multi-round evolution.

Let $\Delta W_g^{(t)} = U^{(t)} \Sigma^{(t)} V^{(t)\top}$. We decompose each selected client's update as

$$\Delta W_k^{(t)} = U^{(t)} D_k^{(t)} \Sigma^{(t)} V^{(t)\top} + P_k^{(t)},$$

where $D_k^{(t)} = \text{diag}(c_{k,1}\beta_{k,1}^{(t)}, \ldots, c_{k,r_{\max}}\beta_{k,r_{\max}}^{(t)})$ captures evolution aligned with the current singular basis, and $P_k^{(t)}$ captures basis drift and cross-direction mixing. Averaging over the selected clients gives

$$\Delta W_g^{(t+1)} = U^{(t)} D^{(t)} \Sigma^{(t)} V^{(t)\top} + R_t,$$

where $D^{(t)} = M^{-1} \sum_{k \in \mathcal{M}_t} D_k^{(t)}$ and $R_t = M^{-1} \sum_{k \in \mathcal{M}_t} P_k^{(t)}$.

The i-th diagonal entry of $D^{(t)}$ is $a_i^{(t)} = M^{-1} \sum_{k \in \mathcal{M}_t} c_{k,i}\beta_{k,i}^{(t)}$. Hence, the aligned component has singular magnitudes $x_i^{(t)} = |a_i^{(t)}| \sigma_i^{(t)}$. Let $x_{(1)}^{(t)} \ge \cdots \ge x_{(r_{\max})}^{(t)}$ denote these values after sorting. Thus, singular-value crossings can move a spectral component to a different rank position and expose it to a different contributor coverage in the next round.

Let $\epsilon_t = \|R_t\|_2$. By the singular-value perturbation bound,

$$\left| \sigma_i^{(t+1)} - x_{(i)}^{(t)} \right| \le \epsilon_t.$$

It follows that

$$(\sigma_i^{(t+1)})^2 = (x_{(i)}^{(t)})^2 + \xi_i^{(t)},$$

where

$$|\xi_i^{(t)}| \le 2\epsilon_t x_{(i)}^{(t)} + \epsilon_t^2.$$

Taking expectation yields

$$e_i^{(t+1)} = \mathbb{E}\left[(x_{(i)}^{(t)})^2 \right] + \bar{\xi}_i^{(t)}, \tag{9}$$

with $|\bar{\xi}_i^{(t)}| \le \mathbb{E}[2\epsilon_t x_{(i)}^{(t)} + \epsilon_t^2]$.

Unlike under stable spectral routing, $x_{(i)}^{(t)}$ need not correspond to direction i from the previous round. Therefore, once singular-value crossings or basis drift occur, the multi-round dynamics generally cannot be reduced to a direction-wise multiplicative recursion of the form $e_i^{(t+1)} = q_i^{(t)} e_i^{(t)}$.

When there is no singular-value crossing and $R_t = 0$, however, $x_{(i)}^{(t)} = |a_i^{(t)}|\sigma_i^{(t)}$. Under the sampling conditions used in Appendix B, Eq. 9 then reduces to $e_i^{(t+1)} = q_i^{(t)} e_i^{(t)}$.

This decomposition clarifies the role of SVD beyond the simplified model. Rank heterogeneity creates the coverage imbalance, while SVD determines which spectral components occupy rank positions with different coverage. Reordering can therefore move a component between coverage levels, and basis drift can transfer energy across directions. These effects may alter the exact multi-round trajectory and invalidate the geometric rate in Theorem 1, but they do not change the one-round contributor dilution in Eq. 3.

D Why raFLoRA Mitigates Rank Collapse

We next connect the proposed aggregation rule to the contributor-dilution mechanism identified above. For a nonempty partition h, conventional aggregation gives

$$\Delta W_{g,h,\text{base}}^{(t+1)} = \frac{C_h^{(t)}}{M} \overline{\Delta W}_h^{(t)},$$

whereas raFLoRA gives

$$\Delta W_{g,h,\text{ra}}^{(t+1)} = \overline{\Delta W}_h^{(t)}.$$

Thus, raFLoRA removes the factor $C_h^{(t)}/M$ responsible for one-round contributor dilution. Rather than amplifying higher-rank updates, it normalizes each partition over its effective contributors.

The same effect appears in the stable-routing model. Fix a direction $i \in \mathcal{I}_h$. Since client ranks take values only from $\{r_1, \ldots, r_H\}$, a client supports direction i if and only if it supports partition h, i.e.,

$$c_{k,i} = \mathbf{1}\{r_k \geq i\} = \mathbf{1}\{r_k \geq r_h\}, \qquad i \in \mathcal{I}_h.$$

Hence,

$$C_h^{(t)} = \sum_{k \in \mathcal{M}_t} c_{k,i}, \qquad i \in \mathcal{I}_h.$$

Conventional aggregation uses

$$a_{i,\text{base}}^{(t)} = \frac{1}{M} \sum_{k \in \mathcal{M}_t} c_{k,i} \beta_{k,i}^{(t)},$$

with

$$\mathbb{E}[a_{i,\text{base}}^{(t)}] = p_i \mu_i^{(t)}.$$

For $C_h^{(t)} > 0$, raFLoRA instead uses

$$a_{i,\text{ra}}^{(t)} = \frac{1}{C_h^{(t)}} \sum_{k \in \mathcal{M}_t} c_{k,i} \beta_{k,i}^{(t)},$$

so that

$$\mathbb{E}\left[a_{i,\text{ra}}^{(t)} \mid C_h^{(t)} > 0\right] = \mu_i^{(t)}.$$

Hence, conditional on receiving an update, the coverage-dependent attenuation $p_i \mu_i^{(t)}$ is removed.

For illustration, suppose all eligible clients have the same directional gain $\beta_i^{(t)}$. Conventional aggregation gives

$$q_{i,\text{base}}^{(t)} = (\beta_i^{(t)})^2 \left[(1-\tau)p_i^2 + \tau p_i\right],$$

whereas, conditional on $C_h^{(t)} > 0$, raFLoRA gives

$$q_{i,\text{ra}}^{(t)} = (\beta_i^{(t)})^2.$$

Thus, the systematic coverage penalty that creates the retention gap in Theorem 1 is removed in nonempty rounds.

When $C_h^{(t)} = 0$, raFLoRA retains the previous global partition instead of replacing it with zero. Under the same simplified model, the corresponding multiplier is therefore one. Thus, for equal directional gains, the unconditional multiplier becomes

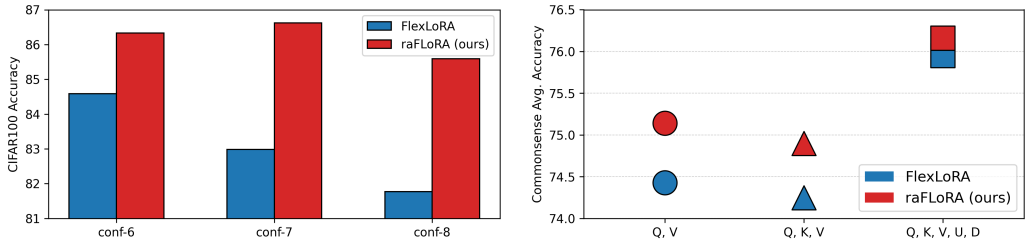
$$q_{i,\mathrm{ra}}^{(t)} = \Pr(C_h^{(t)} > 0)(\beta_i^{(t)})^2 + \Pr(C_h^{(t)} = 0).$$

The fallback introduces no new information, but prevents an empty-contributor round from discarding previously retained higher-rank information.

Overall, raFLoRA removes the systematic contributor-dependent attenuation identified by our analysis and preserves existing information in empty partitions. Other effects, including SVD reordering and spectral mixing, may still affect the multi-round spectrum.

E Additional Experiments

To further examine the impact of rank heterogeneity and LoRA module insertion, we conduct extended experiments that vary both rank configurations and LoRA insertions.



(a) Additional rank configurations

(b) Different LoRA insertion modules

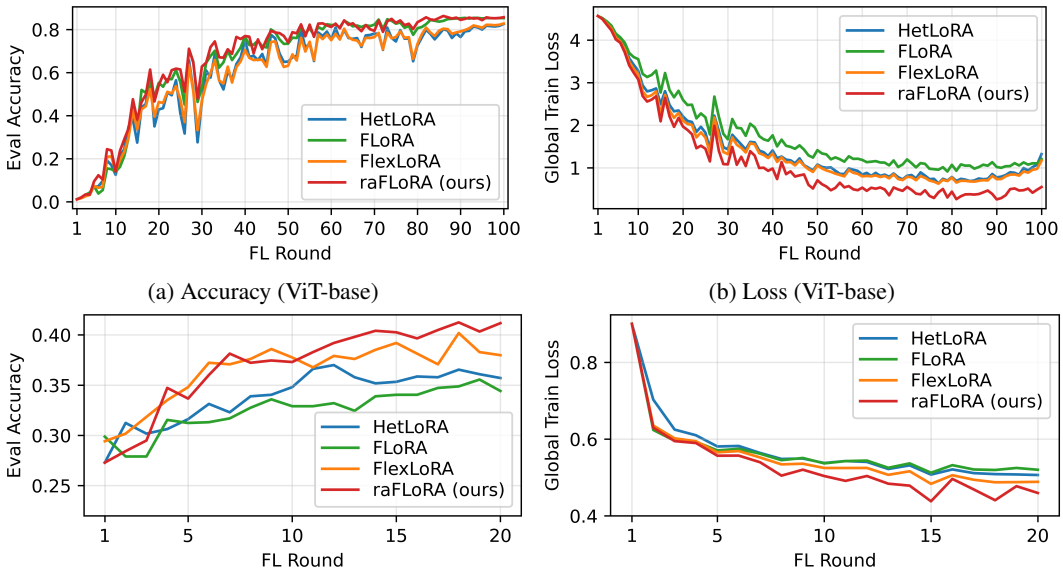
Figure 7: Extended experiments comparing raFLoRA and FlexLoRA under varying rank configurations and LoRA module insertion settings.

Effect of additional rank configurations. We evaluate a broader range of rank configurations, including conf-6 $\{8, 12, 16, 20, 24\}$, conf-7 $\{4, 8, 16, 32, 64\}$, and conf-8 $\{1, 4, 16, 64, 256\}$, which exhibit progressively larger rank gaps. As shown in Figure 7a, raFLoRA consistently achieves larger accuracy improvements over FlexLoRA as rank heterogeneity increases, with gains of up to 4%. This trend reflects the strong dependence of FlexLoRA on the minimum client rank, which increasingly constrains the effective rank of the global update under larger rank gaps. In contrast, raFLoRA mitigates rank collapse through rank-partitioned aggregation, enabling higher-rank clients to contribute more effectively and thereby better exploiting larger rank configurations.

Effect of different LoRA module insertion settings. We further evaluate the robustness of raFLoRA under various LoRA module insertion settings using LLaMA-3.2-3B, including $\{Q, V\}$, $\{Q, K, V\}$, and $\{Q, K, V, U, D\}$, where Q, K, and V denote the query, key, and value projections in attention layers, and U and D correspond to the MLP up- and down-projection layers, respectively. These configurations progressively increase the number of adapted modules. As shown in Figure 7b, raFLoRA consistently achieves higher average accuracy across all commonsense reasoning benchmarks under these settings. These results indicate that raFLoRA scales well with the number of inserted LoRA modules and remains robust across diverse module configurations, highlighting its suitability for flexible and scalable adaptation of large models.

F Training Dynamics of Accuracy and Loss

We track the global evaluation accuracy and global training loss during fine-tuning. As illustrated in Figure 8, raFLoRA consistently achieves higher global accuracy and lower global loss across training rounds on both ViT-base and LLaMA-3.2-3B. Notably, FLoRA re-initializes its LoRA parameters at each round of local training. Although global updates are merged into the base model, the low-rank adaptation does not persist across rounds, causing the optimization trajectory in the low-rank subspace to be repeatedly reset. This may slow convergence and degrade performance on more complex tasks.



(a) Accuracy (ViT-base)

(b) Loss (ViT-base)

(c) Accuracy (LLaMA-3.2-3B)

(d) Loss (LLaMA-3.2-3B)

Figure 8: Global evaluation accuracy and global training loss over communication rounds. Results on CIFAR100 with ViT-base are shown in (a) and (b), while results on GSM8K with LLaMA-3.2-3B are shown in (c) and (d).

G Hyperparameter Settings for Main Experiments

G.1 Image Classification for Vision

Datasets For image classification, we conduct experiments on CIFAR100 Krizhevsky & Hinton (2009), which contains 60K color images of size 32×32 from 100 classes. Each class includes 500 training samples and 100 test samples, resulting in 50K training images and 10K test images in total.

Table 6: Hyperparameter settings for image classification experiments on CIFAR100.

Hyperparameters	**Values**
Number of Clients	100
Number of Rounds	100
Client Participation Ratio per Round	10%
Data Partitioning	c20($\alpha = 1$)
Local Training Epoch	1
Batch Size	32
Optimizer	AdamW
Learning Rate	5e-4
Learning Rate Scheduler	Linear decay per round
Inserted Modules of LoRA	All linear layers
LoRA Rank Configurations	$\{8, 16, 32, 48, 64\}$
Rank Probability Distributions	$\{0.2, 0.2, 0.2, 0.2, 0.2\}$

Hyperparameters For image classification, we follow standard federated learning configurations. Unless otherwise specified, all experiments adopt identical optimization and communication settings, with the complete hyperparameter configuration summarized in Table 6. Given its relatively large label space, CIFAR100 supports flexible non-IID data partitioning, and we therefore fix its data partition to the pathological non-IID setting c20($\alpha = 1$) in the main experiments.

For the experiments in Section 1, Figures 2a and 2b use the same configuration as Table 6. Figure 2c changes only the data partitioning scheme to the regular non-IID setting, with the LoRA rank selected

from $\{1, 16, 32, 48, 64\}$. Figure 2d changes only the shared rank r_1, considering three rank sets: $\{1, 16, 32, 48, 64\}$ for $r_1 = 1$, $\{4, 16, 32, 48, 64\}$ for $r_1 = 4$, and $\{8, 16, 32, 48, 64\}$ for $r_1 = 8$.

G.2 Text Classification for Language

Datasets For text classification, we evaluate our method on the 20 Newsgroups Lang (1995) dataset, a topic classification benchmark consisting of newsgroup posts spanning 20 distinct categories. The dataset contains approximately 11.3K training samples and 7.5K test samples.

Table 7: Hyperparameter settings for text classification experiments on 20NG.

Hyperparameters	**Values**
Number of Clients	100
Number of Rounds	100
Client Participation Ratio per Round	10%
Data Partitioning	c5($\alpha = 1$)
Local Training Epoch	1
Batch Size	32
Optimizer	AdamW
Learning Rate	5e-4
Learning Rate Scheduler	Linear decay per round
Inserted Modules of LoRA	All linear layers
LoRA Rank Configurations	$\{8, 16, 32, 48, 64\}$
Rank Probability Distributions	$\{0.2, 0.2, 0.2, 0.2, 0.2\}$

Hyperparameters For text classification, we similarly follow standard federated learning configurations. Unless otherwise specified, all experiments adopt identical optimization and communication settings, with the complete hyperparameter configuration summarized in Table 7. Owing to its larger number of classes, the 20 Newsgroups dataset allows flexible non-IID partitioning, and we fix its data partition to the pathological non-IID setting c5($\alpha = 1$) in the main experiments.

G.3 Mathematical Reasoning

Datasets For mathematical reasoning, we use the GSM8K Cobbe et al. (2021) dataset, which contains approximately 8.5K high-quality, linguistically diverse grade-school math word problems, including 7.5K training samples and 1.3K test samples. The task focuses on question answering for basic mathematical problems that require multi-step reasoning, typically involving 2 to 8 solution steps. The problems require no concepts beyond early algebra and are primarily solved through sequences of elementary arithmetic operations (e.g., $+$, $-$, $\times$, $\div$). Each solution is provided in natural language rather than symbolic expressions, making the dataset particularly suitable for evaluating step-by-step reasoning capabilities of language models.

Table 8: Hyperparameter settings for mathematical reasoning experiments on GSM8K.

Hyperparameters	**Values**
Number of Clients	100
Number of Rounds	20
Client Participation Ratio per Round	10%
Data Partitioning	iid
Local Training Epoch	1
Batch Size	4
Optimizer	AdamW
Learning Rate	5e-4
Learning Rate Scheduler	Linear decay per round
Inserted Modules of LoRA	Query and Value
LoRA Rank Configurations	$\{8, 16, 32, 48, 64\}$
Rank Probability Distributions	$\{0.2, 0.2, 0.2, 0.2, 0.2\}$

Hyperparameters Following prior work on federated low-rank adaptation of large-scale models, such as Fed-SB Singhal et al. (2025a) and FedEx-LoRA Singhal et al. (2025b), we adopt similar training protocols with necessary modifications to accommodate heterogeneous LoRA rank settings. The detailed hyperparameter configurations used in our main experiments are summarized in Table 8. For the GSM8K dataset, we evenly partition the training data across all clients.

G.4 COMMONSENSE REASONING

Datasets For commonsense reasoning, we use the Commonsense15K Hu et al. (2023) benchmark, which comprises eight sub-tasks: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), Winogrande (Sakaguchi et al., 2021), ARC-Easy and ARC-Challenge (Clark et al., 2018), and OpenBookQA (Mihaylov et al., 2018). Models are fine-tuned on Commonsense15K and evaluated separately on the test sets of each of the eight tasks. Since these sub-tasks involve different discrete answer formats, such as True/False, Answer, Solution, Option, and Ending, we use the answer-format identifiers as categorical labels to construct non-IID client partitions.

Table 9: Hyperparameter settings for commonsense reasoning experiments on Commonsense15K.

Hyperparameters	**Values**
Number of Clients	100
Number of Rounds	20
Client Participation Ratio per Round	10%
Data Partitioning	$\alpha = 0.5$
Local Training Epoch	1
Batch Size	16
Optimizer	AdamW
Learning Rate	5e-4
Learning Rate Scheduler	Linear decay per round
Inserted Modules of LoRA	Query and Value
LoRA Rank Configurations	$\{8, 16, 32, 48, 64\}$
Rank Probability Distributions	$\{0.2, 0.2, 0.2, 0.2, 0.2\}$

Hyperparameters Building on established protocols for federated low-rank adaptation of large-scale models Singhal et al. (2025b;a), we adopt a consistent training setup with minimal adjustments to support heterogeneous LoRA rank settings. The complete set of hyperparameters is reported in Table 9. For Commonsense15K, we apply Dirichlet partitioning over the answer-format labels and fix the partition to the $\alpha = 0.5$ non-IID setting in the main experiments. This yields a controlled and reproducible label-skew non-IID split, where different clients are biased toward different answer-format categories and thus receive different mixtures of the underlying sub-tasks.

H DETAILED RESULTS ON COMMONSENSE REASONING

For commonsense reasoning, we use Commonsense15K Hu et al. (2023), a benchmark including eight datasets: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), Winogrande (Sakaguchi et al., 2021), ARC-Easy and ARC-Challenge (Clark et al., 2018), and OpenBookQA (Mihaylov et al., 2018). We fine-tune the global model on Commonsense15K in a federated setting using LLaMA-3.2-3B Meta AI (2024) and LLaMA-3.1-8B Grattafiori et al. (2024), and evaluate the fine-tuned global model on the test sets of all eight sub-tasks.

As shown in Tables 10 and 11, raFLoRA achieves the highest average accuracy across the eight commonsense reasoning tasks. The Avg. column is computed by first averaging the results over the eight sub-tasks for each random seed, and then reporting the mean and standard deviation across three random seeds. This metric reflects the overall capability of the federated fine-tuned global model, as it summarizes generalization across diverse commonsense reasoning tasks rather than performance on a single sub-task. raFLoRA consistently outperforms HetLoRA and FLoRA on both model scales. Compared with the strong baseline FlexLoRA, raFLoRA achieves higher average accuracy while

Table 10: Performance comparison on commonsense reasoning tasks using LLaMA-3.2-3B.

Methods	BoolQ	PIQA	SIQA	HS	WG	ARC-e	ARC-c	OBQA	Avg.
HetLoRA	$63.78_{\pm 0.23}$	$79.60_{\pm 1.04}$	$67.59_{\pm 0.58}$	$77.54_{\pm 0.71}$	$60.09_{\pm 1.03}$	$84.92_{\pm 0.24}$	$70.36_{\pm 0.47}$	$71.20_{\pm 1.39}$	$71.89_{\pm 0.41}$
FLoRA	$62.94_{\pm 0.56}$	$79.61_{\pm 0.68}$	$66.79_{\pm 0.58}$	$74.85_{\pm 1.16}$	$58.17_{\pm 0.21}$	$84.33_{\pm 0.07}$	$70.39_{\pm 0.44}$	$71.07_{\pm 0.42}$	$71.02_{\pm 0.18}$
FlexLoRA	$62.44_{\pm 2.82}$	$81.11_{\pm 0.49}$	$70.13_{\pm 0.50}$	$81.76_{\pm 0.84}$	$64.30_{\pm 0.56}$	$87.00_{\pm 0.34}$	$72.55_{\pm 0.73}$	$75.33_{\pm 1.10}$	$74.33_{\pm 0.50}$
raFLoRA (ours)	$64.98_{\pm 1.94}$	$81.10_{\pm 0.35}$	$70.79_{\pm 1.47}$	$82.58_{\pm 1.16}$	$65.40_{\pm 0.81}$	$86.95_{\pm 0.59}$	$72.27_{\pm 0.31}$	$74.80_{\pm 1.25}$	$\mathbf{74.86}_{\pm 0.39}$

Table 11: Performance comparison on commonsense reasoning tasks using LLaMA-3.1-8B.

Methods	BoolQ	PIQA	SIQA	HS	WG	ARC-e	ARC-c	OBQA	Avg.
HetLoRA	$70.61_{\pm 0.17}$	$84.80_{\pm 0.93}$	$72.84_{\pm 1.54}$	$86.58_{\pm 1.26}$	$69.56_{\pm 1.60}$	$91.63_{\pm 0.51}$	$78.78_{\pm 0.44}$	$80.80_{\pm 1.73}$	$79.45_{\pm 1.00}$
FLoRA	$69.67_{\pm 1.78}$	$85.29_{\pm 0.33}$	$72.96_{\pm 0.31}$	$86.36_{\pm 0.57}$	$66.72_{\pm 0.77}$	$91.65_{\pm 0.16}$	$79.15_{\pm 0.30}$	$80.27_{\pm 0.90}$	$79.01_{\pm 0.38}$
FlexLoRA	$70.76_{\pm 0.89}$	$85.76_{\pm 1.28}$	$73.70_{\pm 1.47}$	$88.99_{\pm 0.57}$	$73.56_{\pm 0.55}$	$92.06_{\pm 0.50}$	$79.49_{\pm 0.70}$	$82.73_{\pm 1.53}$	$80.88_{\pm 0.67}$
raFLoRA (ours)	$70.88_{\pm 1.15}$	$85.38_{\pm 1.39}$	$74.65_{\pm 0.73}$	$89.06_{\pm 0.31}$	$74.74_{\pm 0.48}$	$91.96_{\pm 0.34}$	$80.66_{\pm 0.13}$	$81.87_{\pm 1.10}$	$\mathbf{81.15}_{\pm 0.08}$

maintaining on-par or superior performance on all sub-tasks except OBQA. These results demonstrate the benefit of rank-partitioned aggregation over existing heterogeneous-rank FedLoRA methods.

I Generality of the Rank-Wise Aggregation Mismatch

Our theoretical analysis targets heterogeneous FedLoRA methods that reconstruct client updates, aggregate them using rank-agnostic weights, and subsequently apply SVD-based heterogeneous rank allocation. FlexLoRA serves as the representative method in the main experiments. We next examine whether the identified rank-wise aggregation mismatch is specific to FlexLoRA.

I.1 Another SVD-Based Heterogeneous FedLoRA Method

We further examine whether rank collapse also arises in ECLoRA, which is another heterogeneous FedLoRA method with SVD-based rank allocation. During reproduction, we identified an implementation discrepancy in the released ECLoRA code, where RSVD appears to be applied to the uncompensated update rather than the compensated update specified in the paper. We corrected this issue and report all ECLoRA results below using the corrected implementation.

Table 12: Energy distribution and accuracy after 100 FL rounds for ECLoRA with and without rank-partitioned aggregation on CIFAR100 using ViT-base.

Method	Rank 1-8 (%)	Rank 9-16 (%)	Rank 17-32 (%)	Rank 33-48 (%)	Rank 49-64 (%)	Accuracy (%)
ECLoRA	99.9986	0.0012	0.0001	0.0000	0.0000	84.07
ECLoRA + raFLoRA	38.2581	22.6282	26.9840	12.1232	0.0165	**86.42**

As shown in Table 12, ECLoRA also exhibits energy concentration, with almost all singular-value energy remaining within ranks 1-8. Applying rank-partitioned weighting broadens the energy distribution across higher-rank components and improves accuracy from 84.07% to 86.42%.

These results show that the observed rank collapse is not specific to FlexLoRA and can also arise in other heterogeneous FedLoRA methods with SVD-based rank allocation. They further suggest that rank-partitioned aggregation can benefit such methods.

I.2 Extension Beyond the Analyzed SVD-Based Framework

We further examine whether accounting for rank-dependent client support can benefit heterogeneous FedLoRA methods beyond the specific SVD-based framework considered in our theory. We apply the same rank-partitioned weighting principle to HetLoRA and FLoRA under the same experimental setting.

As shown in Table 13, rank-partitioned weighting improves HetLoRA from 84.30% to 84.72% and FLoRA from 86.92% to 87.26%. Although these methods fall outside the scope of Theorem 1, the

Table 13: Applying rank-partitioned weighting to heterogeneous FedLoRA methods outside the SVD-based framework.

Method	Accuracy (%)
HetLoRA	84.30
HetLoRA + raFLoRA	**84.72**
FLoRA	86.92
FLoRA + raFLoRA	**87.26**

consistent improvements suggest that explicitly accounting for rank-dependent client support can also benefit heterogeneous FedLoRA methods beyond the SVD-based setting analyzed in our theory.

Overall, these results indicate that the aggregation principle identified by our analysis is not specific to FlexLoRA and may generalize to a broader range of heterogeneous FedLoRA methods.

J Post-SVD Coverage Correction

A natural alternative to rank-partitioned aggregation is to compensate for rank-dependent attenuation after server-side SVD. Since higher-rank directions have lower coverage p_i, one may directly rescale the i-th singular component by $1/p_i$. Specifically, given the aggregated update

$$\Delta W_g = U\Sigma V^\top, \qquad \Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_{r_{\max}}),$$

the post-SVD correction replaces each singular value by

$$\widehat{\sigma}_i = \frac{\sigma_i}{p_i}.$$

Although this correction appears consistent with the coverage-dependent attenuation identified in our analysis, it differs fundamentally from raFLoRA. The post-SVD component at position i does not generally correspond to the pre-SVD rank partition with coverage p_i. Dense aggregation can mix client updates, while SVD can rotate and reorder the resulting spectral directions. Therefore, multiplying the i-th post-SVD component by $1/p_i$ corrects an SVD position rather than the original rank partition supported by a particular set of clients. In contrast, raFLoRA performs contributor-aware normalization before SVD, when the correspondence between each rank partition and its effective contributors is still explicit.

We empirically evaluate this alternative correction on CIFAR100 under the same setting used in our main analysis. Table 14 reports the final accuracy, higher-rank energy ratio, and relative Frobenius aggregation gap.

Table 14: Comparison between conventional aggregation and post-SVD $1/p_i$ scaling.

Method	Accuracy (%)	Higher-rank energy (%)	Frobenius gap (%)
FlexLoRA	84.21%	$\sim$0.001%	0.00014%
Post-SVD $1/p_i$ scaling	86.73%	10.4%	8.99%

Post-SVD scaling substantially increases the higher-rank energy ratio from approximately 0.001% to 10.4% and improves accuracy from 84.21% to 86.73%. This result provides additional evidence that restoring suppressed higher-rank capacity can translate into meaningful performance gains. However, the correction also increases the relative Frobenius aggregation gap from 0.00014% to 8.99%, indicating that rescaling the post-SVD spectrum can distort the aggregated update.

Moreover, its accuracy remains below the 87.45% achieved by raFLoRA under the same setting. These results suggest that correcting rank-dependent attenuation is more effective when performed at the contributor level before SVD rather than by rescaling spectral positions after SVD. raFLoRA therefore addresses the source of the mismatch directly, while post-SVD scaling provides only an approximate correction after spectral mixing and reordering have already occurred.

K LIMITATIONS

Although raFLoRA mitigates rank collapse by correcting the rank-wise aggregation mismatch, preserving higher-rank capacity does not imply that every higher-rank component is useful. To examine the functional contribution of different rank ranges, we perform a rank-prefix ablation by retaining only the first r components of the final global update.

Table 15: Rank-prefix ablation on CIFAR100 using ViT-base. Each row retains only the indicated prefix of the final global LoRA update.

Retained ranks	FlexLoRA (%)	raFLoRA (%)
Classifier only, rank 0	50.84	49.03
Rank 1-8	83.82	83.07
Rank 1-16	83.79	86.90
Rank 1-32	83.82	87.38
Rank 1-48	83.83	87.25
Rank 1-64	83.82	87.14

Table 15 shows that FlexLoRA gains little beyond ranks 1-8, whereas ranks 9-32 substantially improve raFLoRA from 83.07% to 87.38%. Extending further to ranks 48 or 64 slightly reduces accuracy, indicating that preserving higher-rank capacity is beneficial overall, although not every additional high-rank component contributes positively.

A remaining limitation is the reliability of sparsely supported partitions. Contributor-aware normalization corrects rank-dependent attenuation but provides limited variance reduction when only a few clients contribute, making single-contributor partitions more sensitive to noisy updates. Future extensions could improve their stability through shrinkage toward the previous global partition or reliability-aware weighting. Thus, raFLoRA addresses rank-dependent attenuation, while variance control for sparsely supported partitions remains a complementary problem.