Rethinking Factor Sharing in Federated LoRA: A Rank-Aware Adaptive Approach

Xinyi Xu[1,*] Bingnan Xiao[2,*] Shuang Qin[1,†] Gang Feng[1] Tony Q.S. Quek[3]

[1]University of Electronic Science and Technology of China
[2]Fudan University
[3]Singapore University of Technology and Design
[*]Equal contribution. [†]Corresponding author.
xinyixu@std.uestc.edu.cn, 22110720061@m.fudan.edu.cn, blueqs@uestc.edu.cn
fenggang@uestc.edu.cn, tonyquek@sutd.edu.sg

***Abstract*—Low-rank adaptation (LoRA) represents large language model (LLM) updates with two compact matrix factors, i.e., A and B, providing an efficient way to fine-tune large models in federated learning paradigm. Inspired by the asymmetric roles of the LoRA factors, we study whether A should be shared across clients while B remains client-specific (Share-A/Local-B), or whether B should instead be shared while A remains client-specific (Share-B/Local-A). With a least-squares surrogate, we reveal that Share-A/Local-B requires the client-specific LoRA update matrices to use a common rank-r input-side space, whereas Share-B/Local-A requires a common rank-r output-side space. The two strategies therefore incur different projection residuals, indicating that the preferred strategy is the one with the smaller aggregate residual across clients. With this insight, we propose Federated Adaptive Factor Sharing Low-Rank Adaptation (FedAS-LoRA), which selects the sharing side before training to enhance fine-tuning performance. To enable adaptive factor selection before training, we design a Rank-Aware Shared-Subspace Sufficiency (RSS) metric, which effectively assesses whether a shared rank-r input subspace is sufficient for the local data distributions using representations extracted from a frozen LLM backbone. Experiments across different tasks, data distributions, LoRA ranks, and participation settings confirm the effectiveness of RSS and the superior performance of FedAS-LoRA.**

I. Introduction

Large language models (LLMs) have demonstrated strong capabilities in language understanding and generation, supporting a wide range of natural language processing applications Brown et al. [2020], Touvron et al. [2023]. For pretrained LLMs, task-specific adaptations are needed before deployment to meet the requirements of different scenarios Houlsby et al. [2019], Li and Liang [2021], Hu et al. [2022]. In many practical settings, the data required for fine-tuning are distributed across clients or organizations and cannot be centralized due to privacy, ownership, or regulatory constraints. Federated learning (FL) enables multiple clients to collaboratively fine-tune a model without sharing their private data McMahan et al. [2017], Li et al. [2020a]. Nevertheless, full-model fine-tuning in FL incurs substantial computation, storage, and communication costs. Parameter-efficient fine-tuning provides a practical alternative by updating only a small set of additional parameters He et al. [2022]. In particular, low-rank adaptation (LoRA) keeps the pretrained LLM backbone frozen and represents each model update by two trainable low-rank factors, making it well suited to resource-constrained federated adaptation.

One critical challenge of federated LoRA lies in aggregation mismatch. For an FL system with N clients, let the LoRA update of client i be $\Delta W_i = B_i A_i$. Independently averaging the two factors B_i and A_i at the server yields

$$\Big(\frac{1}{N}\sum_{i=1}^{N} B_i\Big)\Big(\frac{1}{N}\sum_{i=1}^{N} A_i\Big) \neq \frac{1}{N}\sum_{i=1}^{N} B_i A_i, \qquad (1)$$

which differs from directly averaging the client model updates. Several studies have explored methods to address this mismatch Bian et al. [2025], Wang et al. [2026], Chen et al. [2026]. One representative line of work assigns asymmetric training and aggregation roles to the two LoRA factors. For example, FFA-LoRA fixes a common randomly initialized factor A_0, and only trains and aggregates B_i Sun et al. [2024b]. FedSA-LoRA instead keeps both factors trainable, aggregates A_i, and retains B_i locally for personalized adaptation Guo et al. [2025]. Although such methods adopt different factor-handling strategies, they hard-code the roles of A and B before training and apply the same assignment across diverse system settings. *It remains unclear whether this hard-coded assignment is consistently superior, prompting a reconsideration of the roles of the LoRA factors B and A in federated settings.*

To examine this question, we compare Share-A/Local-B and Share-B/Local-A under a label-balanced input-skew partition. As shown in Fig. 1, Share-B/Local-A outperforms Share-A/Local-B, and the performance gap varies with LoRA rank. This observation does not imply that Share-B/Local-A is always preferable. Instead, it means that fixing a sharing side is unsuitable for all federated LoRA settings. We further analyze the structural asymmetry of the two LoRA factors through a least-squares surrogate. The resultant projection characterization shows that sharing A imposes a common rank-r input-side representation across clients, whereas sharing B imposes a common rank-r output-side representation. Thus, the hard-coded sharing

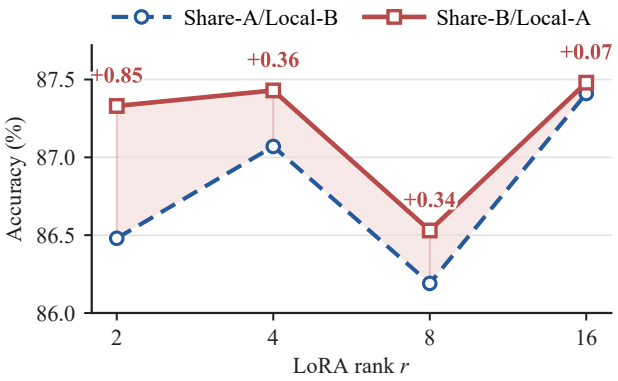


Fig. 1: Accuracy gap between Share-A/Local-B and Share-B/Local-A on the MNLI-m task under label-balanced input skew. The federated system includes 3 clients with full participation and 500 training rounds.

side assignment is not globally optimal, since the appropriate shared factor depends on whether a common rank-r input-side or output-side space yields a smaller aggregate projection residual across clients.

Based on this insight, we propose Federated Adaptive Factor Sharing Low-Rank Adaptation (FedAS-LoRA) framework, as illustrated in Fig. 2. FedAS-LoRA adaptively selects the shared factor before training instead of hard-coding the same policy. To guide this selection, we design a Rank-Aware Shared-Subspace Sufficiency (RSS) metric using sequence-level representations extracted from a frozen LLM backbone. RSS compares the second-order variation captured by a global rank-r input subspace with that captured by the corresponding client-specific subspaces. When the RSS deficit exceeds the calibrated threshold, FedAS-LoRA treats the global rank-r input subspace as insufficient and selects Share-B/Local-A; otherwise, it selects Share-A/Local-B. We further establish that FedAS-LoRA converges to a stationary neighborhood under arbitrary client participation.

Our main contributions are summarized as follows:

- We provide the first projection-based comparison between Share-A/Local-B and Share-B/Local-A in federated LoRA. We reveal that Share-A/Local-B and Share-B/Local-A yield input-side and output-side projection residuals, respectively, which explains why fixed sharing strategies are not suitable for all federated settings.
- Based on this insight, we design an RSS metric to efficiently determine the shared factor before training. With RSS, we propose FedAS-LoRA, which adapts different federated settings to enhance fine-tuning performance.
- We prove the convergence of FedAS-LoRA under arbitrary client participation, extending the analysis beyond the full-participation setting. For either Share-A/Local-B or Share-B/Local-A strategy, we prove that the corresponding iterates converge to a stationary neighborhood.
- Extensive experiments on diverse natural language tasks demonstrate the superiority of FedAS-LoRA over other methods, and validate the effectiveness of RSS-based adaptive sharing side selection.

II. Related Work

A. Federated Parameter-Efficient Fine-Tuning

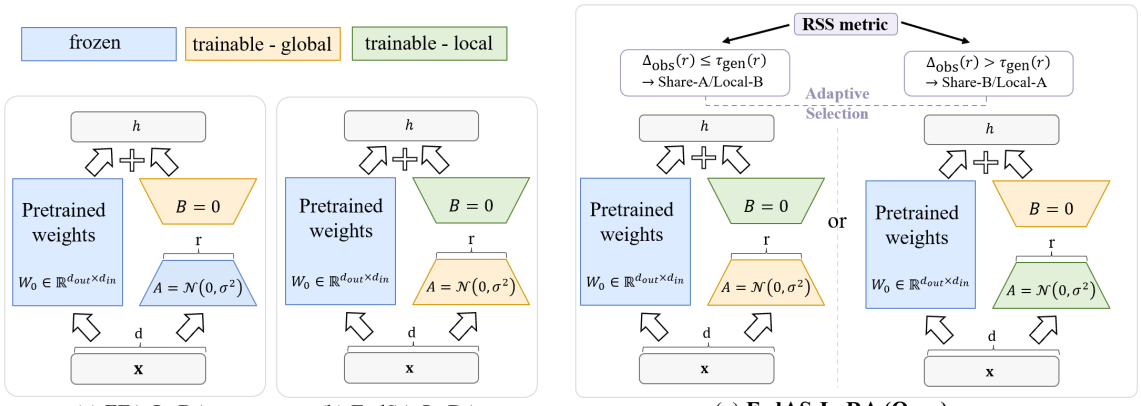
Parameter-efficient fine-tuning (PEFT) reduces the computation, memory, and communication costs of adapting pretrained models in federated learning. Existing studies apply prompt tuning and other compact trainable modules while keeping the pretrained backbone frozen Che et al. [2023], Sun et al. [2024a], Guo et al. [2024]. Low-rank adaptation (LoRA) represents each weight update with two low-rank factors, making it suitable for federated fine-tuning with limited client resources.

Existing federated LoRA methods mainly address personalization, resource heterogeneity, and aggregation error Yang et al. [2025], Koo et al. [2025], Byun and Lee [2025], Shen et al. [2025]. FedDPA maintains global and personalized adapters to capture shared and client-specific knowledge Yang et al. [2024]. HetLoRA supports resource-heterogeneous clients through rank self-pruning and sparsity-weighted aggregation Cho et al. [2024], while FlexLoRA and FLoRA aggregate LoRA modules with different ranks through reconstruction or stacking Bai et al. [2024], Wang et al. [2024]. FedEx-LoRA instead corrects the mismatch between factor-wise averaging and direct averaging of the resulting updates Singhal et al. [2025]. These methods determine shared and personalized components at the adapter level, or improve the aggregation of heterogeneous LoRA updates.

B. LoRA Factor Asymmetry and Factor-Wise Federated Aggregation

The two LoRA factors can exhibit different optimization and representation behaviors. LoRA-FA freezes A and trains only B to reduce the memory cost of fine-tuning Zhang et al. [2023], whereas LoRA+ assigns different learning rates to the two factors Hayou et al. [2024]. Zhu et al. interpret A as extracting input features and B as mapping these features to the output space Zhu et al. [2024]. HydraLoRA further adopts an asymmetric architecture with a shared A and multiple B experts Tian et al. [2024]. These studies establish factor asymmetry in centralized fine-tuning, but do not consider the division of shared and client-specific factors in federated learning.

Factor-wise handling has also been explored in federated LoRA Chen et al. [2026], Wang et al. [2026], Xu et al. [2025]. FFA-LoRA fixes a common randomly initialized A and trains and aggregates B Sun et al. [2024b]. FedSA-LoRA trains both factors, shares A, and retains B locally for personalization Guo et al. [2025]. RoLoRA alternates the optimization of the two factors during federated training Chen et al. [2025], while FedRot-LoRA aligns client factors through orthogonal transformations before aggregation Zhang et al. [2026]. Although these methods assign



(a) FFA-LoRA (b) FedSA-LoRA (c) **FedAS-LoRA (Ours)**

Fig. 2: Illustration of (a) FFA-LoRA, (b) FedSA-LoRA, and (c) FedAS-LoRA. In FFA-LoRA, A is fixed after initialization, while B is trainable and shared with the server for aggregation. In FedSA-LoRA, both A and B are trainable, while A is shared and B is retained locally. In FedAS-LoRA, the server uses RSS to determine the sharing strategy before training. For either strategy, both factors are updated locally; each client uploads only the shared factor and retains the other factor locally.

different training or aggregation roles to the two factors, the sharing side is either predetermined or not explicitly considered. Which LoRA factor should be shared across clients and which should remain client-specific remains underexplored.

III. Preliminaries and Motivation

This section formulates the federated LoRA factor-sharing problem and proves that a fixed sharing side is insufficient. We first define a general federated factor-sharing model covering Share-A, Share-B, and full-factor sharing, followed by a motivating example to illustrate that Share-B/Local-A can outperform Share-A/Local-B. Based on a least-squares surrogate, we characterize the structural asymmetry between the two LoRA factors in federated settings.

A. Factor Sharing in Federated LoRA

Consider a federated fine-tuning system with a server and N clients, indexed by $\mathcal{N} = \{1, 2, \ldots, N\}$. Client i owns a local dataset $\mathcal{D}_i$ and optimizes a local objective f_i. For a target linear layer with frozen pretrained weight $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, the LoRA update of each client i is $\Delta W_i = B_i A_i$. Here, $A_i \in \mathbb{R}^{r \times d_{\text{in}}}$, $B_i \in \mathbb{R}^{d_{\text{out}} \times r}$, and $r \ll \min\{d_{\text{in}}, d_{\text{out}}\}$ denotes the LoRA rank. Given an input representation $x \in \mathbb{R}^{d_{\text{in}}}$, A_i maps x to an r-dimensional latent representation, while B_i maps this representation to the output space.

For federated LoRA settings, we consider a fixed shared factor type $Q \in \{A, B\}$. Let $Q_i = A_i$ when $Q = A$ and $Q_i = B_i$ when $Q = B$ for brevity. The corresponding personalized federated objective is

$$\min_{\{A_i, B_i\}_{i=1}^N} \quad \frac{1}{N} \sum_{i=1}^{N} f_i(B_i A_i) \qquad (2)$$
$$\text{s.t.} \quad Q_i = Q_j, \ \forall i, j \in \mathcal{N}.$$

Here, $f_i(B_i A_i)$ denotes the local loss evaluated at $W_0 + B_i A_i$. When $Q = A$, the constraint shares A across clients while leaving B_i client-specific, yielding Share-A/Local-B. When $Q = B$, it shares B while leaving A_i client-specific, yielding Share-B/Local-A. At training round t, let Q_t be the shared factor maintained by the server and $Q_{t,i}^e$ its local copy at participating client i after e local updates.

In each training round t, the following steps are executed:

- **Client selection and downlink transmission:** The server selects an arbitrary subset of $\mathcal{S}_t \subseteq \mathcal{N}$ with S_t clients, and broadcasts Q_t, $Q \in \{A, B\}$, to clients $i \in \mathcal{S}_t$.
- **Local update:** Each selected client initializes the local copy of every shared factor as $Q_{t,i}^0 = Q_t$, and performs E local update steps with local learning rate η_ℓ on its local dataset $\mathcal{D}_i$. During local training, the client updates all trainable LoRA factors in its local model, while keeping non-shared factors client-local.
- **Uplink Aggregation:** After local training, each selected client uploads the update of the shared factor, $\Delta Q_{t,i} = Q_{t,i}^E - Q_t, Q \in \{A, B\}$ to the server for aggregation:

$$Q_{t+1} = Q_t + \eta_g \frac{1}{S_t} \sum_{i \in \mathcal{S}_t} \Delta Q_{t,i}, \qquad (3)$$

  where η_g is the server learning rate.

The sharing strategy determines which side of the LoRA adaptation is exposed to cross-client averaging and which side remains client-specific.

B. Observation: Fixed Sharing Can Be Suboptimal

One representative personalized federated LoRA design adopts Share-A/Local-B, i.e., sharing factor A while keeping $B_i, \forall i \in \mathcal{N}$ client-specific in each training round Guo et al. [2025]. However, it remains unclear whether sharing A is

always preferable to sharing B. To answer this question, we compare Share-A/Local-B and Share-B/Local-A on MNLI-m Wang et al. [2018] using a label-balanced input-skew partition. To create input skew while preserving label balance, we construct a three-client partition based on the genre annotations provided by MNLI Wang et al. [2018]. Each client is assigned premise–hypothesis pairs from one of three genres: telephone, government, and fiction. The partition is constructed such that the clients have nearly identical distributions over the three MNLI labels. Thus, the clients share the same prediction task and similar label distributions, while their inputs originate from different textual genres. This setting introduces genre-based input heterogeneity while controlling for label skew. Fig. 1 compares their test accuracies for $r \in \{2, 4, 8, 16\}$. It is noted that Share-B/Local-A achieves better performance for all rank settings, while the accuracy advantage varies with the LoRA rank r, indicating that the performance difference between the two sharing strategies depends on the LoRA rank r.

Fig. 1 demonstrates that Share-A/Local-B is not uniformly optimal across different FL settings. This observation motivates the following question: *For a given LoRA rank r and client data distributions, what determines which factor should be shared and which should remain client-specific?* We next examine this question by analyzing the structural asymmetry between the two LoRA factors.

C. Structural Asymmetry Between LoRA Factors

For Share-A/Local-B and Share-B/Local-A in federated LoRA, although both strategies parameterize the update as B_iA_i, the two factors act on different sides of this product. A_i first maps the input into an r-dimensional representation, whereas B_i maps that representation to the output. Sharing A therefore enforces a common input representation, while sharing B enforces a common set of output directions.

We clarify the difference with a single-layer least-squares surrogate. For each client $i \in \mathcal{N}$, let $\Delta_i \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ denote its desired adaptation. The error of a rank-r LoRA update B_iA_i is then formulated as

$$\mathcal{L}_i(B_i, A_i) = \|(\Delta_i - B_iA_i)\Sigma_i^{1/2}\|_F^2, \tag{4}$$

where $\Sigma_i = \mathbb{E}[x_i x_i^\top] \succ 0$ with $x_i \in \mathbb{R}^{d_{\text{in}}}$ denoting the input feature. Since each client uploads only the shared factor, the corresponding optimization problems can be formulated as $\mathcal{E}_A = \min_{A,\{B_i\}_i} \frac{1}{N}\sum_{i=1}^N \|(\Delta_i - B_iA)\Sigma_i^{1/2}\|_F^2$ and $\mathcal{E}_B = \min_{B,\{A_i\}_i} \frac{1}{N}\sum_{i=1}^N \|(\Delta_i - BA_i)\Sigma_i^{1/2}\|_F^2$. Here, $\mathcal{E}_A$ corresponds to Share-A/Local-B and $\mathcal{E}_B$ to Share-B/Local-A, respectively. Based on the formulations of $\mathcal{E}_A$ and $\mathcal{E}_B$, we derive the following theorem to reveal the differences between Share-A/Local-B and Share-B/Local-A.

Theorem 1. *Assume $\Sigma_i \succ 0, \forall i \in \mathcal{N}$. Minimizing $\mathcal{E}_A$ over $\{B_i\}_i$ and $\mathcal{E}_B$ over $\{A_i\}_i$ yields the following equivalent problems over the shared factors A and B, respectively:*

$$\mathcal{E}_A = \min_A \frac{1}{N}\sum_{i=1}^N \left\|\Delta_i\Sigma_i^{1/2}(I - \Pi_{i,A})\right\|_F^2, \tag{5}$$

$$\mathcal{E}_B = \min_B \frac{1}{N}\sum_{i=1}^N \left\|(I - \Pi_B)\Delta_i\Sigma_i^{1/2}\right\|_F^2. \tag{6}$$

where $\Pi_{i,A} = (A\Sigma_i^{1/2})^\top(A\Sigma_iA^\top)^\dagger(A\Sigma_i^{1/2})$, and $\Pi_B = B(B^\top B)^\dagger B^\top$.

From **Theorem 1**, it is noted that $\Pi_{i,A}$ keeps the part that lies in the row space of $A\Sigma_i^{1/2}$, while $I-\Pi_{i,A}$ keeps the part outside this space. Similarly, Π_B keeps the part that lies in the column space of B, while $I-\Pi_B$ keeps the part outside this space. For Share-A/Local-B, with optimal $\{B_{i,*}\}_i$, $\mathcal{E}_A$ is dominated by $\|\Delta_i\Sigma_i^{1/2}(I-\Pi_{i,A})\|_F^2, \forall i$, which measures the residual not covered by the shared input-side space induced by A. For Share-B/Local-A, with optimal $\{A_{i,*}\}_i$, $\|(I-\Pi_B)\Delta_i\Sigma_i^{1/2}\|_F^2$ in $\mathcal{E}_B$ denotes the parts not covered by the shared output-side space induced by B. Therefore, **Theorem 1** shows a structural asymmetry between the two sharing strategies: sharing A constrains the input-side space used by all clients, whereas sharing B constrains the output-side space used by all clients.

The structural asymmetry between (5) and (6) explains why the same LoRA rank r may lead to different preferred sharing strategies. For fixed $\{\Delta_i, \Sigma_i\}_{i=1}^N$, r determines the dimension of the shared input-side or output-side space. Increasing r relaxes the rank constraint in both (5) and (6), so the optimal value of neither residual can increase. Therefore, neither $\mathcal{E}_A$ nor $\mathcal{E}_B$ can increase with r. Since the two residuals measure projection errors on different sides of $\Delta_i\Sigma_i^{1/2}$, the difference $\mathcal{E}_A - \mathcal{E}_B$ is not generally invariant to r. Consequently, the preferred sharing strategy under the least-squares surrogate may depend on the target LoRA rank. The sharing strategy with the smaller residual is preferred, while the other factor remains client-specific. This leads to a design principle: *share the side on which the clients have a common rank-r structure, and keep the other side local.* Since both $\mathcal{E}_A$ and $\mathcal{E}_B$ depend on Δ_i and Σ_i, which are unavailable before fine-tuning, in the next section, we design a training-free metric to effectively choose the sharing side before training.

IV. Method

In this section, we design a rank-aware shared-subspace sufficiency metric, RSS, to determine the shared factor before training, and prove the convergence of FedAS-LoRA under arbitrary LoRA factor selection and client participation.

A. RSS-Based Sharing-Side Selection

The goal of RSS is to estimate the preferred sharing factor to enhance federated fine-tuning performance. **Theorem 1** shows that this selection is determined by comparing $\mathcal{E}_A$

and $\mathcal{E}_B$ associated with common rank-r input- and output-side spaces, respectively. Since both $\mathcal{E}_A$ and $\mathcal{E}_B$ depend on Δ_i and Σ_i, which are unavailable before training, RSS utilizes sequence-level representations extracted from a frozen LLM backbone to evaluate whether the shared rank-r input subspace is sufficient for the local data distributions. Specifically, it compares the projected global-mean-centered second moment captured by a global rank-r subspace with that retained by client-specific rank-r subspaces.

Based on local datasets $\mathcal{D}_i$ with size $|\mathcal{D}_i| = D_i$, we define the RSS score as

$$\mathrm{RSS}(r) = \sum_{i=1}^{N} p_i \frac{\mathrm{Tr}\left(U_{g,r}^\top S_i U_{g,r}\right)}{\mathrm{Tr}\left(U_{i,r}^\top S_i U_{i,r}\right) + \epsilon}, \tag{7}$$

where $p_i = \frac{D_i}{\sum_{i=1}^{N} D_i}$, and $\epsilon > 0$ is a small constant for numerical stability. S_i represents the centered covariance matrix of client i with $U_{i,r}$ containing its top r eigenvectors, and $U_{g,r}$ contains the top r eigenvectors of $S_g = \sum_{i=1}^{N} p_i S_i$. $\mathrm{RSS}(r) \to 1$ indicates that the global rank-r subspace preserves as much client-specific representation energy as the locally optimal rank-r subspaces.

We further construct a threshold $\tau(r)$ for the RSS deficit $\Delta(r) = 1 - \mathrm{RSS}(r)$, accounting for the errors in randomly reassigning samples across clients and estimating local subspaces from finite data samples. $\tau(r)$ is formulated as

$$\tau(r) = \max\left\{\tau_{\mathrm{null}}(r), \tau_{\mathrm{boot}}(r)\right\}, \tag{8}$$

where $\tau_{\mathrm{null}}(r)$ is the 95th percentile of the RSS deficits obtained by randomly reassigning samples across clients while preserving each client's sample size, and $\tau_{\mathrm{boot}}(r)$ is the empirical 95th percentile of the representation-energy losses obtained after re-estimating each client's top-r subspace from bootstrap resamples of its local data. Thus, $\Delta(r) > \tau(r)$ only when the deficit $\Delta(r)$ exceeds both thresholds, and the sharing-side strategy $\mathcal{M}(r)$ is

$$\mathcal{M}(r) = \begin{cases} \text{Share-B/Local-A}, \text{if } \Delta(r) > \tau(r), \\ \text{Share-A/Local-B}, \text{if } \Delta(r) \leq \tau(r). \end{cases} \tag{9}$$

From (9), we can see that when $\Delta(r) > \tau(r)$, the loss in representation energy from using the global rank-r subspace instead of the client-specific rank-r subspaces exceeds both calibration thresholds. Since sharing A requires all clients to use a common input-side subspace, FedAS-LoRA selects Share-B/Local-A in this case. When $\Delta(r) \leq \tau(r)$, the observed energy loss does not exceed the calibrated threshold, which leads to Share-A/Local-B.

Before training starts, RSS is computed with representations extracted by a frozen LLM backbone. It compares the representation energy retained by the global rank-r subspace with that retained by the client-specific rank-r subspaces. Together with the calibrated threshold, such comparison is used in (9) to decide whether A is shared or kept client-specific. The details of the client and global subspaces, the two calibration terms, and the complete RSS workflow are provided in Appendix.

B. Convergence Analysis

Based on the designed RSS to determine the shared LoRA factor, we establish the convergence of FedAS-LoRA under arbitrary client subset $\mathcal{S}_t$ with $|\mathcal{S}_t| = S_t, \forall t$. This starts with the following assumptions.

Assumption 1 (Smoothness)**.** *For each client $i \in \mathcal{N}$, the local objective f_i is L-smooth. For $\forall\ U_1, U_2 \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}}$,*

$$f_i(U_2) \leq f_i(U_1) + \langle \nabla f_i(U_1), U_2 - U_1 \rangle_F + \frac{L}{2}\|U_2 - U_1\|_F^2. \tag{10}$$

Assumption 2 (Unbiased and Bounded Stochastic Gradients)**.** *For any client i, let $G_{t,i}^e = \nabla f_i(U_{t,i}^e; \xi_{t,i}^e)$ and $\bar{G}_{t,i}^e = \nabla f_i(U_{t,i}^e)$, respectively. The stochastic gradient $G_{t,i}^e$ is unbiased and bounded with $G_{\max} > 0$, i.e.,*

$$\mathbb{E}\left[G_{t,i}^e \mid U_{t,i}^e\right] = \bar{G}_{t,i}^e \text{ and } \|G_{t,i}^e\|_F \leq G_{\max}. \tag{11}$$

Assumption 3 (LoRA Parameter Bounds and Alignment)**.** *There exist constants $C_A, C_B > 0$ and $c_A, c_B > 0$ such that*

$$\|A_{t,i}^e\|_F \leq C_A \text{ and } \|B_{t,i}^e\|_F \leq C_B. \tag{12}$$

$$\left\langle A_{t,i}^{e\top} A_{t,i}^e, \bar{G}_{t,i}^{e\top} \bar{G}_{t,i}^e \right\rangle_F \geq c_A \|\bar{G}_{t,i}^e\|_F^2,$$

$$\left\langle B_{t,i}^e B_{t,i}^{e\top}, \bar{G}_{t,i}^e \bar{G}_{t,i}^{e\top} \right\rangle_F \geq c_B \|\bar{G}_{t,i}^e\|_F^2. \tag{13}$$

Assumptions 1 and **2** are standard in federated optimization Li et al. [2020b], Yu et al. [2019], Reddi et al. [2021]. **Assumption 3** characterizes the bilinear LoRA parameterization, as widely used in federated LoRA analysis Guo et al. [2025], Chen et al. [2025], Park and Klabjan [2025]. Specifically, the alignment inequalities in **Assumption 3** hold when the nonzero singular values of $A_{t,i}^e$ and $B_{t,i}^e$ are uniformly bounded away from zero and the realized gradient has non-vanishing projections onto their induced row and column spaces, respectively. The norm bounds keep the factor iterates in a bounded region. By the chain rule for $U = BA$, the gradient $\bar{G}_{t,i}^e$ with respect to U induces the gradients $\bar{G}_{t,i}^e A_{t,i}^{e\top}$ and $B_{t,i}^{e\top} \bar{G}_{t,i}^e$ with respect to $B_{t,i}^e$ and $A_{t,i}^e$, respectively. Since the alignment inequalities are imposed only along the realized directions of $\bar{G}_{t,i}^e$, they do not impose a global full-rank condition on the low-rank factors. The following result applies to any realized participation sequence that satisfies the gradient-mass coverage condition, as specified in the Appendix.

Theorem 2. *Let $F_\star$ be a lower bound of the aggregate objective. For both Share-A/Local-B and Share-B/Local-A sharing strategies, the iterates of FedAS-LoRA satisfy*

$$\frac{1}{NT}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\mathbb{E}\left[\|\nabla f_i(U_{t,i})\|_F^2\right] \leq \frac{F_0 - F_\star}{\eta_\ell c T} + \frac{1}{\eta_\ell c T}\sum_{t=0}^{T-1}\left[\frac{S_t}{N}\mathcal{R}_{\mathrm{sel}} + \left(1 - \frac{S_t}{N}\right)\mathcal{R}_{\mathrm{uns}}\right], \tag{14}$$

where c is a positive constant, $C_\star := \max\{C_A, C_B\}$, $\mathcal{R}_{\mathrm{sel}} = E\eta_\ell^2 C_A C_B G_{\max}^3 + \frac{3LE}{2}\eta_\ell^2 (C_A^4 + C_B^4) G_{\max}^2 + \frac{3LE}{2} \cdot \eta_\ell^4 C_A^2 C_B^2 G_{\max}^4 + \frac{\eta_\ell}{2} G_{\max}^2 + (\frac{1}{2\eta_\ell} + \frac{L}{2})(1+\eta_g)^2 \eta_\ell^2 E^2 C_\star^4 G_{\max}^2$,

and $\mathcal{R}_{\text{uns}} = (\frac{1}{2\rho\eta_\ell} + \frac{L}{2})\eta_g^2\eta_\ell^2 E^2 C_\star^4 G_{\max}^2$. *The proof of Theorem 2 is provided in the Appendix.*

Theorem 2 shows that FedAS-LoRA reaches a stationary neighborhood under arbitrary client participation. The first term on the right-hand side (RHS) of (14) decreases as $O(1/T)$, while the second term serves as the non-vanishing error induced by stochastic local updates, multiple local steps, and client sampling drift. The proof covers both sharing strategies since their local updates are identical: each selected client updates both A and B before uploading only the shared factor. The only difference lies in the uploaded factor update, leading to a different drift bound, i.e., $\|Q_{t,k}^E - Q_t\|_F \le \eta_\ell E C_Q G_{\max}$, where $Q \in \{A, B\}$.

V. Experiments

In this section, we systematically evaluate the performance of FedAS-LoRA on two types of tasks: natural language understanding and natural language generation.

A. Experimental Setup

Datasets and implementation.: For natural language understanding tasks, we evaluate FedAS-LoRA with RoBERTa-large Liu et al. [2019] on the GLUE benchmark Wang et al. [2018], including MNLI, SST-2, QNLI, QQP, and RTE. For natural language generation tasks, we evaluate FedAS-LoRA on the GSM8K dataset Cobbe et al. [2021] with LLaMA3-8B Meta AI [2024].

For natural language understanding tasks, consistent with FedSA-LoRA Guo et al. [2025], we use the pre-trained RoBERTa-large model with 355M parameters Liu et al. [2019] from the HuggingFace Transformers library Wolf et al. [2020] as the backbone. LoRA modules are inserted into the query and value projection matrices of each attention layer. The default LoRA rank is set to $r = 8$ with the scaling factor $\alpha = 16$. For LoRA variants, rsLoRA adopts the same training configuration except for the rank-stabilized scaling rule Kalajdzievski [2023]. We use SGD for all LoRA- and rsLoRA-based methods. For VeRA Kopiczko et al. [2024], we set $r = 256$ and use AdamW optimizer Loshchilov and Hutter [2019] with separate learning rates for the classification head and the adapted layers. Across all methods, we use a batch size of 128, $E = 10$, and $T = 500$. The local learning rate η_l is tuned from $\{0.001, 0.002, 0.005, 0.01, 0.02, 0.05\}$, and the global learning rate η_g is set to 1. Experiments are conducted on NVIDIA RTX 4090 GPUs. All results are reported as mean $\pm$ standard deviation over three independent runs. See the Appendix for more experimental details.

Benchmarks.: We compare FedAS-LoRA with representative federated LoRA methods and fixed factor-sharing policies. *LoRA* denotes the standard federated implementation that trains and aggregates both factors A_i and B_i. *FFA-LoRA* Sun et al. [2024b] fixes a common randomly initialized factor A and only trains and aggregates B_i. *FedDPA-LoRA* Yang et al. [2024] maintains global and personalized LoRA modules to capture shared and client-specific knowledge. *FedSA-LoRA* Guo et al. [2025] trains both factors, aggregates A_i, and retains B_i locally, i.e., Share-A/Local-B.

B. Main Results

For natural language understanding tasks, consistent with FFA-LoRA Sun et al. [2024b], we randomly split the data across three clients for federated learning. We consider a non-IID setting using a Dirichlet distribution with $\alpha = 0.5$, i.e., Dir(0.5). From Table I, we observe that FedAS-LoRA, FedAS-rsLoRA, and FedAS-VeRA consistently outperform the compared methods in terms of average accuracy, demonstrating the effectiveness of the proposed method. FedAS-LoRA achieves an average accuracy of 90.77%, which is 0.93 percentage points higher than that of FedSA-LoRA. These results indicate that applying the same fixed factor-sharing policy across tasks and adaptation variants can lead to suboptimal performance, supporting our design of selecting the sharing side according to the specific federated setting.

For natural language generation, we evaluate LLaMA3-8B Meta AI [2024] on GSM8K using the HuggingFace Transformers library Wolf et al. [2020]. Following FederatedScope-LLM Kuang et al. [2024], we split the data across three clients under an IID distribution and adopt the same optimization settings. LoRA, FFA-LoRA, and FedAS-LoRA achieve accuracies of 55.14%, 54.51%, and 56.28%, respectively, with generated examples in the Appendix.

C. In-Depth Analyses

We proceed to utilize LoRA-based methods to conduct analyses on the natural language understanding tasks of QNLI, SST-2, and MNLI-m to assess the impact of client sampling, data heterogeneity, and LoRA rank on model performance. Detailed comparisons of RSS decisions with Share-A/Local-B and Share-B/Local-A strategies, together with the communication cost analysis, are provided in the Appendix.

1) Effect of Client Sampling: We consider 10 clients under both uniform and non-uniform sampling settings. For uniform sampling, we apply a sampling rate of 0.3 in each round. For non-uniform sampling, client i participates independently according to a Bernoulli distribution with probability of $0.1+0.4(i-1)/(N-1)$, i.e., the participation probabilities increase linearly from 0.1 to 0.5. As shown in Table II, FedAS-LoRA consistently outperforms FedSA-LoRA across all evaluated tasks under both uniform and non-uniform sampling. Additional scalability results with $N = 50$ clients are provided in the Appendix.

2) Effect of Data Heterogeneity: We consider the performance of FedAS-LoRA under different data heterogeneity levels by examining IID data distributions, Dir(1), and Dir(0.5) under full client participation. Table III shows that FedAS-LoRA outperforms the compared baselines across the evaluated IID and Dirichlet partitions. Additional results

	Method	MNLI-m	MNLI-mm	SST-2	QNLI	QQP	RTE	Avg.
LoRA	LoRA	88.36 ±0.08	88.65 ±0.05	95.19 ±0.09	90.80 ±0.89	85.42 ±1.27	85.84 ±0.36	89.04
	FFA-LoRA	86.46 ±0.04	87.47 ±0.09	95.53 ±0.03	89.24 ±0.84	86.38 ±0.53	87.05 ±0.07	88.69
	FedDPA-LoRA	88.77 ±0.06	88.02 ±0.10	96.10 ±0.05	89.92 ±0.59	86.60 ±0.81	87.36 ±0.15	89.46
	FedSA-LoRA	89.77 ±0.02	87.80 ±0.09	95.72 ±0.03	91.13 ±0.56	86.87 ±0.64	87.75 ±0.02	89.84
	FedAS-LoRA (Ours)	**89.95** ±0.06	**88.86** ±0.49	**97.17** ±0.19	**92.94** ±0.39	**87.95** ±0.56	**87.77** ±0.04	**90.77**
rsLoRA	rsLoRA	84.46 ±0.27	86.99 ±0.13	96.49 ±0.09	89.29 ±1.44	86.11 ±0.87	86.42 ±0.22	88.29
	FFA-rsLoRA	89.10 ±0.12	87.85 ±0.25	96.10 ±0.08	89.13 ±0.58	86.02 ±0.34	86.98 ±0.09	89.20
	FedDPA-rsLoRA	89.36 ±0.19	88.67 ±0.29	96.79 ±0.10	90.89 ±1.17	86.68 ±0.02	86.71 ±0.28	89.85
	FedSA-rsLoRA	90.28 ±0.09	88.13 ±0.12	96.75 ±0.06	91.34 ±0.67	86.65 ±0.28	87.52 ±0.07	90.12
	FedAS-rsLoRA (Ours)	**90.61** ±0.10	**89.79** ±0.23	**97.02** ±0.07	**91.90** ±0.45	**87.11** ±0.11	**87.58** ±0.08	**90.67**
VeRA	VeRA	89.57 ±0.02	87.79 ±0.11	94.33 ±0.17	91.94 ±0.52	88.67 ±0.48	85.63 ±0.20	89.66
	FFA-VeRA	87.06 ±0.13	86.72 ±0.41	93.12 ±0.35	90.59 ±0.11	87.22 ±0.04	85.15 ±0.29	88.31
	FedDPA-VeRA	87.98 ±0.46	86.78 ±0.36	94.39 ±0.48	91.56 ±0.31	88.71 ±0.35	86.24 ±0.06	89.28
	FedSA-VeRA	89.73 ±0.16	87.26 ±0.03	94.95 ±0.42	92.64 ±0.35	89.05 ±0.11	**87.24** ±0.02	90.15
	FedAS-VeRA (Ours)	**90.28** ±0.54	**88.38** ±0.32	**95.87** ±0.12	**93.08** ±0.11	**89.26** ±0.22	**87.22** ±0.03	**90.68**

TABLE I: Performance of different methods on the GLUE benchmark. MNLI-m denotes MNLI with matched test sets, and MNLI-mm denotes MNLI with mismatched test sets.

	QNLI	SST-2	MNLI-m
LoRA	89.11 ±0.41	96.74 ±0.67	86.87 ±0.35
FFA-LoRA	89.78 ±0.68	96.51 ±0.37	87.16 ±0.24
FedDPA-LoRA	90.26 ±0.45	96.79 ±0.21	87.90 ±0.20
FedSA-LoRA	90.34 ±0.67	96.72 ±0.44	88.16 ±0.27
FedAS-LoRA (Ours)	**91.04** ±0.34	**97.64** ±0.26	**88.67** ±0.19
LoRA	84.15 ±0.52	95.79 ±0.41	83.07 ±0.26
FFA-LoRA	84.48 ±0.63	95.83 ±0.51	84.25 ±0.42
FedDPA-LoRA	85.45 ±0.77	95.38 ±0.59	85.80 ±0.54
FedSA-LoRA	86.11 ±0.50	96.50 ±0.62	86.27 ±0.43
FedAS-LoRA (Ours)	**86.64** ±0.41	**97.58** ±0.30	**87.52** ±0.28

TABLE II: Performance on QNLI, SST-2, and MNLI-m with $N = 10$ clients under uniform and non-uniform sampling. The upper block uses a uniform per-round sampling rate of 0.3, while the lower block uses independent client-specific Bernoulli sampling probabilities linearly spaced from 0.1 to 0.5.

Method	IID	Dir(1)	Dir(0.5)
LoRA	90.85 ±0.08	90.96 ±0.46	90.80 ±0.89
FFA-LoRA	89.51 ±0.25	90.39 ±0.42	89.24 ±0.44
FedSA-LoRA	91.09 ±0.19	90.89 ±0.47	91.13 ±0.57
FedAS-LoRA (Ours)	**91.12** ±0.22	**91.58** ±0.11	**92.94** ±0.39
LoRA	95.27 ±0.02	95.41 ±0.04	95.19 ±0.09
FFA-LoRA	95.45 ±0.03	95.70 ±0.10	95.53 ±0.03
FedSA-LoRA	96.09 ±0.04	96.56 ±0.04	95.72 ±0.03
FedAS-LoRA (Ours)	**96.10** ±0.02	**96.79** ±0.04	**97.17** ±0.19
LoRA	88.30 ±0.02	87.78 ±0.05	88.36 ±0.08
FFA-LoRA	88.69 ±0.03	88.90 ±0.06	86.46 ±0.04
FedSA-LoRA	89.41 ±0.05	89.01 ±0.04	89.77 ±0.02
FedAS-LoRA (Ours)	**89.43** ±0.04	**89.02** ±0.03	**89.95** ±0.06

TABLE III: Performance comparison on the QNLI, SST-2, and MNLI-m tasks with various degrees of data heterogeneity. From top to bottom, the three blocks correspond to QNLI, SST-2, and MNLI-m, respectively.

under structured input-skew partitions are provided in Appendix.

3) Effect of LoRA Rank: We also explore the model performance over varying LoRA ranks $r \in \{2, 4, 8, 16\}$. The corresponding results are reported in Table IV. It is noted that FedAS-LoRA maintains competitive performance across the evaluated ranks. These results indicate that no sharing factor policy is consistently preferable across different ranks, demonstrating the effectiveness of the rank-aware sharing-side selection in FedAS-LoRA.

VI. Conclusion

In this paper, we studied which LoRA factor should be shared across clients and which should remain local during federated fine-tuning. Our investigations have shown that Share-A/Local-B and Share-B/Local-A induce input-side and output-side projection residuals, respectively. To address the structural asymmetry, we proposed FedAS-LoRA, which selects the LoRA sharing factor before training and keeps the other factor client-specific. We designed a training-free RSS metric that compares the energy captured by global and local rank-r subspaces to effectively determine the sharing factor. We established the convergence guarantees for both sharing strategies under arbitrary client participation. Experimental results demonstrated that the preferred sharing side can change with the data distribution, adapter rank, and participation pattern, while FedAS-LoRA achieves superior performance across the evaluated settings.

Method	QNLI	SST-2	MNLI-m
LoRA	87.12 ±0.16	95.41 ±0.19	88.48 ±0.27
FFA-LoRA	86.45 ±0.83	94.90 ±0.22	88.61 ±0.09
FedSA-LoRA	89.42 ±0.29	96.56 ±0.41	89.61 ±0.23
FedAS-LoRA (Ours)	**89.46** ±0.27	**96.79** ±0.21	**89.63** ±0.15
LoRA	90.25 ±0.32	95.67 ±0.48	87.38 ±0.35
FFA-LoRA	91.05 ±0.21	94.78 ±0.49	90.62 ±0.46
FedSA-LoRA	92.01 ±0.35	96.30 ±0.20	90.01 ±0.39
FedAS-LoRA (Ours)	**92.80** ±0.23	**96.33** ±0.14	**90.28** ±0.25
LoRA	90.80 ±0.29	95.19 ±0.09	88.36 ±0.08
FFA-LoRA	89.24 ±0.24	95.53 ±0.03	86.46 ±0.04
FedSA-LoRA	91.13 ±0.27	95.72 ±0.03	89.77 ±0.02
FedAS-LoRA (Ours)	**92.94** ±0.39	**97.17** ±0.19	**89.95** ±0.06
LoRA	88.47 ±0.52	94.95 ±0.19	88.75 ±0.27
FFA-LoRA	88.25 ±0.55	94.92 ±0.16	88.23 ±0.13
FedSA-LoRA	89.12 ±0.39	**95.73** ±0.18	89.01 ±0.17
FedAS-LoRA (Ours)	**89.56** ±0.19	**95.72** ±0.16	**89.73** ±0.07

TABLE IV: Test accuracy on QNLI, SST-2, and MNLI-m with different LoRA ranks r. From top to bottom, the four blocks correspond to $r = 2$, $r = 4$, $r = 8$, and $r = 16$, respectively.

REFERENCES

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, pages 14457–14483, 2024. doi: 10.52202/079017-0461. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/1a134b50202088aa8c595cc99b310e5a-Abstract-Conference.html.

Jieming Bian, Lei Wang, Letian Zhang, and Jie Xu. LoRA-FAIR: Federated LoRA fine-tuning with aggregation and initialization refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3737–3746, 2025.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 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)*, pages 356–362, Albuquerque, New Mexico, 4 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-short.30.

Tianshi Che, Ji Liu, Yang Zhou, Jiaxiang Ren, Jiwen Zhou, Victor Sheng, Huaiyu Dai, and Dejing Dou. Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7871–7888, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.488. URL https://aclanthology.org/2023.emnlp-main.488/.

Shuangyi Chen, Yuanxin Guo, Yue Ju, Hardik Dalal, Zhongwen Zhu, and Ashish Khisti. Robust federated finetuning of LLMs via alternating optimization of LoRA. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/adf17e7346b6be4c2f2bb40de572e5bc-Paper-Conference.pdf.

Yiming Chen, Ruanjun Li, Jiawei Shao, Lifeng Sun, Chi Zhang, Jun Zhang, and Xuelong Li. Federated LoRA fine-tuning of LLMs with only transmitting matrix A or B. *IEEE Transactions on Mobile Computing*, pages 1–17, May 2026. doi: 10.1109/TMC.2026.3696859. Early Access, Art. no. 11535031.

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*, pages 12903–12913, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.717. URL https://aclanthology.org/2024.emnlp-main.717/.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. doi: 10.48550/arXiv.2110.14168. URL https://arxiv.org/abs/2110.14168.

Pengxin Guo, Shuang Zeng, Yanran Wang, Huijie Fan, Feifei Wang, and Liangqiong Qu. Selective aggregation for low-rank adaptation in federated learning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=iX3uESGdsO.

Tao Guo, Song Guo, Junxiao Wang, Xueyang Tang, and Wenchao Xu. PromptFL: Let federated participants cooperatively learn prompts instead of models—federated learning in age of foundation model. *IEEE Transactions on Mobile Computing*, 23(5):5179–5194, 2024. doi: 10.1109/TMC.2023.3302410.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 17783–17806. PMLR, 2024. URL

https://proceedings.mlr.press/v235/hayou24a.html.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=0RDcd5Axok.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 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. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with LoRA. *arXiv preprint arXiv:2312.03732*, 2023. doi: 10.48550/arXiv.2312.03732. URL https://arxiv.org/abs/2312.03732.

Jabin Koo, Minwoo Jang, and Jungseul Ok. Towards robust and efficient federated low-rank adaptation with heterogeneous clients. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 416–429, Vienna, Austria, 7 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.19.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. VeRA: Vector-based random matrix adaptation. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NjNfLdxr3A.

Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. FederatedScope-LLM: A comprehensive package for fine-tuning large language models in federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5260–5271, New York, NY, USA, 2024. Association for Computing Machinery. doi: 10.1145/3637528.3671573. URL https://doi.org/10.1145/3637528.3671573.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37 (3):50–60, 2020a. doi: 10.1109/MSP.2020.2975749.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of FedAvg on non-IID data. In *International Conference on Learning Representations*, 2020b. URL https://openreview.net/forum?id=HJxNAnVtDS.

Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353.

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. doi: 10.48550/arXiv.1907.11692. URL https://arxiv.org/abs/1907.11692.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017.

Meta AI. Introducing Meta Llama 3: The most capable openly available LLM to date. https://ai.meta.com/blog/meta-llama-3/, April 2024. Published April 18, 2024.

Haemin Park and Diego Klabjan. Communication-efficient federated low-rank update algorithm and its connection to implicit regularization. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=DdPeCRVyCd.

Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LkFG3lB13U5.

Zhanming Shen, Tianqi Xu, Hao Wang, Jian Li, and Miao Pan. pFedGPT: Hierarchically optimizing LoRA aggregation weights for personalized federated GPT models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4766–4778, Suzhou, China, 11 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.239.

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)*, pages 1316–1336, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.67. URL https://aclanthology.org/2025.acl-long.67/.

Jingwei Sun, Ziyue Xu, Hongxu Yin, Dong Yang, Daguang Xu, Yudong Liu, Zhixu Du, Yiran Chen, and Holger R. Roth. FedBPT: Efficient federated black-box prompt tuning for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning*

Research, pages 47159–47173. PMLR, 2024a. URL https://proceedings.mlr.press/v235/sun24j.html.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving LoRA in privacy-preserving federated learning. In *International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=NLPzL6HWNl.

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Chengzhong Xu. HydraLoRA: An asymmetric LoRA architecture for efficient fine-tuning. In *Advances in Neural Information Processing Systems*, volume 37, pages 9565–9584, 2024. doi: 10.52202/079017-0304. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/123fd8a56501194823c8e0dca00733df-Abstract-Conference.html.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. doi: 10.48550/arXiv.2302.13971.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446/.

Haoran Wang, Xiong Wang, Yuqing Li, Jing Chen, Junyi Zhang, Nan Yan, Kun He, and Wei Wang. Federated LoRA fine-tuning with pipelined error-mitigated aggregation and matrix-wise freezing. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 5749–5762, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.284. URL https://aclanthology.org/2026.findings-acl.284/.

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*, volume 37, pages 22513–22533, 2024. doi: 10.52202/079017-0708. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/28312c9491d60ed0c77f7fff4ad86dd1-Abstract-Conference.html.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, 6 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

Binqian Xu, Haiyang Mei, Zechen Bai, Jinjin Gong, Rui Yan, Guosen Xie, Yazhou Yao, Basura Fernando, and Xiangbo Shu. You only communicate once: One-shot federated low-rank adaptation of MLLM. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Yiyuan Yang, Guodong Long, Tao Shen, Jing Jiang, and Michael Blumenstein. Dual-personalizing adapter for federated foundation models. In *Advances in Neural Information Processing Systems*, volume 37, pages 39409–39433, 2024. doi: 10.52202/079017-1245. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/45a30141c6719e9cfedfb51f1c665a37-Abstract-Conference.html.

Yiyuan Yang, Guodong Long, Qinghua Lu, Liming Zhu, Jing Jiang, and Chengqi Zhang. Federated low-rank adaptation for foundation models: A survey. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 10779–10787. International Joint Conferences on Artificial Intelligence Organization, 8 2025. doi: 10.24963/ijcai.2025/1196. Survey Track.

Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5693–5700, July 2019. doi: 10.1609/aaai.v33i01.33015693. URL https://doi.org/10.1609/aaai.v33i01.33015693.

Haoran Zhang, Dongjun Kim, Seohyeon Cha, and Haris Vikalo. FedRot-LoRA: Mitigating rotational misalignment in federated LoRA. In *Forty-third International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.23638.

Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. LoRA-FA: Efficient and effective low rank representation fine-tuning. *arXiv preprint arXiv:2308.03303*, 2023. doi: 10.48550/arXiv.2308.03303. URL https://arxiv.org/abs/2308.03303.

Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, Haitz Sáez De Ocáriz Borde, Rickard Brüel Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. Asymmetry in low-rank adapters of foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages

62369–62385. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhu24c.html.

APPENDIX A
TRAINING PROCESS OF FEDAS-LORA

Algorithm 1 summarizes the training flow of FedAS-LoRA. Before federated training, the server computes RSS from the sequence-level representations extracted by the frozen LLM backbone and determines the sharing policy $\mathcal{M}(r)$ according to (9). The selected policy remains fixed throughout training. Specifically, Share-A/Local-B sets $Q = A$, whereas Share-B/Local-A sets $Q = B$.

At each training round t, the server selects a nonempty client subset $\mathcal{S}_t \subseteq \mathcal{N}$ and sends the current shared factor Q_t to the selected clients. Each selected client initializes its shared-factor copy with Q_t, restores its retained factor, and updates both LoRA factors for E local steps. It then uploads only the shared-factor increment $\Delta Q_{t,i} = Q_{t,i}^E - Q_t$ and retains the updated nonshared factor locally. The server aggregates the received increments according to (3), while the client-local factors of unselected clients remain unchanged. For clarity, **Algorithm 1** is written for one LoRA-adapted layer.

APPENDIX B
DESIGN DETAILS OF THE RSS METRIC

This section provides the construction of the client and global subspaces in (7), followed by the two calibration terms in (8).

Client and global subspaces.: Let $h(x) \in \mathbb{R}^{d_h}$ denote the sequence-level representation of input x extracted from the frozen LLM backbone. Given the client datasets $\{\mathcal{D}_i\}_{i=1}^N$ with $D_i = |\mathcal{D}_i|$, the global representation mean is

$$\bar{h} = \frac{1}{\sum_{j=1}^N D_j} \sum_{j=1}^N \sum_{(x,y)\in\mathcal{D}_j} h(x). \tag{15}$$

For each client $i \in \mathcal{N}$, we then define

$$S_i = \frac{1}{D_i} \sum_{(x,y)\in\mathcal{D}_i} \big(h(x) - \bar{h}\big)\big(h(x) - \bar{h}\big)^\top. \tag{16}$$

All clients are centered using the same global mean $\bar{h}$. Hence, S_i reflects both the variation of client i's representations and the difference between its representation mean and the global mean.

Let $U_{i,r} \in \mathbb{R}^{d_h \times r}$ contain the top r orthonormal eigenvectors of S_i. The global matrix is $S_g = \sum_{i=1}^N p_i S_i = \sum_{i=1}^N \frac{D_i}{\sum_{i=1}^N D_i} S_i$, with $U_{g,r} \in \mathbb{R}^{d_h \times r}$ containing its top r orthonormal eigenvectors. By the Rayleigh–Ritz principle, $\mathrm{Tr}(U_{i,r}^\top S_i U_{i,r})$ is the maximum representation energy that a rank-r orthonormal subspace can retain for client i. In contrast, $\mathrm{Tr}(U_{g,r}^\top S_i U_{g,r})$ is the energy retained for the same client by the global rank-r subspace. Their ratio in (7) therefore measures how much of client i's locally attainable rank-r energy is preserved by the global subspace. RSS averages these ratios across clients using $\{p_i\}_{i=1}^N$.

Algorithm 1 FedAS-LoRA

Input: Client datasets $\{\mathcal{D}_i\}_{i=1}^N$, frozen LLM backbone, rank r, initial LoRA factors, rounds T, local steps E, local learning rate η_ℓ, and server learning rate η_g.
At the server, compute $\mathrm{RSS}(r)$, $\Delta(r) = 1 - \mathrm{RSS}(r)$, and $\tau(r)$, and determine $\mathcal{M}(r)$ according to (9).
if $\mathcal{M}(r) =$ Share-A/Local-B **then**
  Set $Q = A$; the server maintains $Q_t = A_t$, and client i retains $B_{t,i}$ locally.
else
  Set $Q = B$; the server maintains $Q_t = B_t$, and client i retains $A_{t,i}$ locally.
end if
for $t = 0, 1, \ldots, T-1$ **do**
  The server selects a nonempty subset $\mathcal{S}_t \subseteq \mathcal{N}$ with $S_t = |\mathcal{S}_t|$ and sends Q_t to each client $i \in \mathcal{S}_t$.
  for each client $i \in \mathcal{S}_t$ in parallel **do**
    Set $Q_{t,i}^0 = Q_t$ and initialize the nonshared factor from client i's retained local state.
    for $e = 0, 1, \ldots, E-1$ **do**
      Sample a mini-batch $\xi_{t,i}^e$ and set $U_{t,i}^e = B_{t,i}^e A_{t,i}^e$.
      Compute $G_{t,i}^e = \nabla f_i(U_{t,i}^e; \xi_{t,i}^e)$.
      Update $B_{t,i}^{e+1} = B_{t,i}^e - \eta_\ell G_{t,i}^e A_{t,i}^{e\top}$.
      Update $A_{t,i}^{e+1} = A_{t,i}^e - \eta_\ell B_{t,i}^{e\top} G_{t,i}^e$.
    end for
    Upload $\Delta Q_{t,i} = Q_{t,i}^E - Q_t$; retain $B_{t,i}^E$ if $Q = A$, or $A_{t,i}^E$ if $Q = B$.
  end for
  The server updates $Q_{t+1} = Q_t + \eta_g S_t^{-1} \sum_{i\in\mathcal{S}_t} \Delta Q_{t,i}$.
  Each client $j \notin \mathcal{S}_t$ keeps its client-local factor unchanged.
end for
return The shared factor Q_T and the retained client-local factors.

Random-split calibration.: A nonzero deficit may arise even when samples are randomly assigned to clients. To measure this variation, we randomly reassign the samples while preserving the client sizes $\{D_i\}_{i=1}^N$. For the b-th reassignment, we recompute the client and global subspaces S_i and S_g. Let $\mathrm{RSS}_{\mathrm{null}}^{(b)}(r)$ denote the resulting RSS score and define

$$\Delta_{\mathrm{null}}^{(b)}(r) = 1 - \mathrm{RSS}_{\mathrm{null}}^{(b)}(r). \tag{17}$$

The random-split calibration term is

$$\tau_{\mathrm{null}}(r) = Q_{0.95}\left(\left\{\Delta_{\mathrm{null}}^{(b)}(r)\right\}_b\right), \tag{18}$$

where $Q_{0.95}$ denotes the empirical 95th percentile. Thus, $\tau_{\mathrm{null}}(r)$ measures the RSS deficit that can be produced by random reassignment under the same client sizes.

Bootstrap calibration.: The estimated local subspaces can also vary because each client has finitely many samples. For the b-th bootstrap replicate, client i draws D_i samples

Algorithm 2 RSS Computation Workflow

Input: Client datasets $\{\mathcal{D}_i\}_{i=1}^N$, frozen LLM backbone, rank r, and $\epsilon > 0$.
Extract the sequence-level representation $h(x)$ for every sample using the frozen LLM backbone.
Compute the global representation mean $\bar{h}$ from all client samples.
for each client $i \in \mathcal{N}$ **do**
  Compute S_i according to (16).
end for
Compute $\mathrm{RSS}(r)$ according to (7), and set $\Delta(r) = 1 - \mathrm{RSS}(r)$.
for each random reassignment indexed by b **do**
  Randomly reassign the samples across clients while preserving the client sizes $\{D_i\}_{i=1}^N$.
  Recompute the client and global rank-r subspaces from the reassigned datasets.
  Compute $\mathrm{RSS}_{\mathrm{null}}^{(b)}(r)$ using (7), and set $\Delta_{\mathrm{null}}^{(b)}(r) = 1 - \mathrm{RSS}_{\mathrm{null}}^{(b)}(r)$.
end for
Compute $\tau_{\mathrm{null}}(r)$ according to (18).
for each bootstrap replicate indexed by b **do**
  for each client $i \in \mathcal{N}$ **do**
    Draw D_i samples with replacement from $\mathcal{D}_i$.
    Using the same global mean $\bar{h}$, recompute the client matrix and let $U_{i,r}^{(b)}$ contain its top r orthonormal eigenvectors.
  end for
  Compute $e_{\mathrm{boot}}^{(b)}(r)$ according to (19).
end for
Compute $\tau_{\mathrm{boot}}(r)$ according to (20).
Set $\tau(r) = \max\{\tau_{\mathrm{null}}(r), \tau_{\mathrm{boot}}(r)\}$.
if $\Delta(r) > \tau(r)$ **then**
  Set $\mathcal{M}(r) =$ Share-B/Local-A.
else
  Set $\mathcal{M}(r) =$ Share-A/Local-B.
end if
return $\mathrm{RSS}(r)$, $\Delta(r)$, $\tau(r)$, and $\mathcal{M}(r)$.

with replacement from $\mathcal{D}_i$. Using the same global mean $\bar{h}$, we recompute the client matrix and let $U_{i,r}^{(b)}$ contain its top r orthonormal eigenvectors. The corresponding loss of representation energy is

$$e_{\mathrm{boot}}^{(b)}(r) = \sum_{i=1}^{N} p_i \left[1 - \frac{\mathrm{Tr}\left(U_{i,r}^{(b)\top} S_i U_{i,r}^{(b)} \right)}{\mathrm{Tr}\left(U_{i,r}^{\top} S_i U_{i,r} \right) + \epsilon} \right]_+, \quad (19)$$

where $[z]_+ = \max\{z, 0\}$. The bootstrap calibration term is

$$\tau_{\mathrm{boot}}(r) = Q_{0.95}\left(\left\{ e_{\mathrm{boot}}^{(b)}(r) \right\}_b \right). \quad (20)$$

Here, $\tau_{\mathrm{boot}}(r)$ measures the loss caused by re-estimating the client-specific rank-r subspaces from finite samples.

The final threshold in (8) is the larger of $\tau_{\mathrm{null}}(r)$ and $\tau_{\mathrm{boot}}(r)$. Therefore, $\Delta(r) > \tau(r)$ only when the observed RSS deficit exceeds both the random-split variation and the finite-sample subspace error. The sharing policy is then determined by (9).

Algorithm 2 summarizes the computation process of RSS, whose quantities are computed before federated optimization without gradients, LoRA warm-up, or trained LoRA factors.

Relation to Theorem 1.: **Theorem 1** shows that, under the least-squares surrogate, the preferred sharing strategy is determined by comparing the aggregate input-side and output-side projection residuals, $\mathcal{E}_A$ and $\mathcal{E}_B$. Directly evaluating these residuals before training is infeasible because they depend on the client-specific desired adaptations Δ_i, which are unavailable at that stage. RSS therefore does not directly estimate $\mathcal{E}_A$, $\mathcal{E}_B$, or their difference. Instead, it evaluates whether the common rank-r input-side structure required by Share-A/Local-B is sufficient for the local data distributions, using sequence-level representations extracted from the frozen LLM backbone. When $\Delta(r) > \tau(r)$, the observed RSS deficit exceeds both the random-split variation and the finite-sample subspace error, indicating that the global rank-r input subspace is insufficient. FedAS-LoRA therefore selects Share-B/Local-A. Otherwise, the observed deficit does not exceed the calibrated threshold, and Share-A/Local-B is selected. Thus, RSS serves as a training-free sharing-side selector motivated by Theorem 1, rather than a direct estimator of the two projection residuals. Its agreement with the empirically better fixed sharing strategy is reported in Table VIII.

APPENDIX C
PROOF OF THEOREM 1

For the Share-A/Local-B case, fix $A \in \mathbb{R}^{r \times d_{\mathrm{in}}}$, and let $Y_i = \Delta_i \Sigma_i^{1/2}$ and $X_i = A\Sigma_i^{1/2}$. Optimizing the client-local factor B_i gives the least-squares problem $\min_{B_i} \|Y_i - B_i X_i\|_F^2$, and the corresponding solution is given by

$$B_i^\star(A) = Y_i X_i^\top (X_i X_i^\top)^\dagger = \Delta_i \Sigma_i A^\top (A \Sigma_i A^\top)^\dagger. \quad (21)$$

With the definition of $\Pi_{i,A}$ in **Theorem 1**, we have $B_i^\star(A) X_i = Y_i \Pi_{i,A}$. Therefore,

$$\begin{aligned} \min_{B_i} \|(\Delta_i - B_i A)\Sigma_i^{1/2}\|_F^2 &= \|Y_i (I - \Pi_{i,A})\|_F^2 \\ &= \left\| \Delta_i \Sigma_i^{1/2} (I - \Pi_{i,A}) \right\|_F^2. \end{aligned} \quad (22)$$

Averaging over the clients and minimizing over the shared factor A yields (5).

For Share-B/Local-A, fix $B \in \mathbb{R}^{d_{\mathrm{out}} \times r}$ and again let $Y_i = \Delta_i \Sigma_i^{1/2}$. Since $\Sigma_i \succ 0$, the change of variables $C_i = A_i \Sigma_i^{1/2}$ is bijective. The local problem can therefore be written as $\min_{C_i} \|Y_i - B C_i\|_F^2$. Taking $C_i^\star = B^\dagger Y_i$ gives

$$A_i^\star(B) = C_i^\star \Sigma_i^{-1/2} = B^\dagger \Delta_i. \quad (23)$$

The fitted value satisfies $BC_i^\star = BB^\dagger Y_i = \Pi_B Y_i$, where Π_B is the orthogonal projector onto the column space of B. It follows that

$$\min_{A_i} \|(\Delta_i - BA_i)\Sigma_i^{1/2}\|_F^2 = \|(I-\Pi_B)Y_i\|_F^2 = \left\|(I-\Pi_B)\Delta_i\Sigma_i^{1/2}\right\|_F^2. \quad (24)$$

Averaging over the clients and minimizing over the shared factor B yields (6), which completes the proof.

Appendix D
Proof of Theorem 2

For client i at local step e of round t, let $U_{t,i}^e = B_{t,i}^e A_{t,i}^e$ and $U_{t,i} = U_{t,i}^0$. We define the aggregate objective as $F_t = \frac{1}{N}\sum_{i=1}^N f_i(U_{t,i})$ and recall that $C_\star = \max\{C_A, C_B\}$. By **Assumption 2** and Jensen's inequality, we have

$$\|\bar{G}_{t,i}^e\|_F \le \mathbb{E}\left[\|G_{t,i}^e\|_F \mid U_{t,i}^e\right] \le G_{\max}. \quad (25)$$

During local training, both LoRA factors are updated according to $B_{t,i}^{e+1} = B_{t,i}^e - \eta_\ell G_{t,i}^e A_{t,i}^{e\top}, A_{t,i}^{e+1} = A_{t,i}^e - \eta_\ell B_{t,i}^{e\top} G_{t,i}^e$. Thus, we have

$$U_{t,i}^{e+1} - U_{t,i}^e = -\eta_\ell B_{t,i}^e B_{t,i}^{e\top} G_{t,i}^e - \eta_\ell G_{t,i}^e A_{t,i}^{e\top} A_{t,i}^e + \eta_\ell^2 G_{t,i}^e A_{t,i}^{e\top} B_{t,i}^{e\top} G_{t,i}^e. \quad (26)$$

Applying **Assumption 1** to the increment (26) gives

$$f_i(U_{t,i}^{e+1}) - f_i(U_{t,i}^e) \le \left\langle \bar{G}_{t,i}^e, U_{t,i}^{e+1} - U_{t,i}^e \right\rangle_F + \frac{L}{2}\|U_{t,i}^{e+1} - U_{t,i}^e\|_F^2. \quad (27)$$

For the first term on the RHS of (27), based on the unbiasedness of $G_{t,i}^e$ and **Assumption 3**, we have

$$-\eta_\ell \mathbb{E}\left[\left\langle \bar{G}_{t,i}^e, B_{t,i}^e B_{t,i}^{e\top} G_{t,i}^e \right\rangle_F\right] \le -\eta_\ell c_B \|\bar{G}_{t,i}^e\|_F^2,$$
$$-\eta_\ell \mathbb{E}\left[\left\langle \bar{G}_{t,i}^e, G_{t,i}^e A_{t,i}^{e\top} A_{t,i}^e \right\rangle_F\right] \le -\eta_\ell c_A \|\bar{G}_{t,i}^e\|_F^2. \quad (28)$$

For the $\eta_\ell^2 G_{t,i}^e A_{t,i}^{e\top} B_{t,i}^{e\top} G_{t,i}^e$ term in (26), we have

$$\eta_\ell^2 \left|\left\langle \bar{G}_{t,i}^e, G_{t,i}^e A_{t,i}^{e\top} B_{t,i}^{e\top} G_{t,i}^e \right\rangle_F\right| \le \eta_\ell^2 \|\bar{G}_{t,i}^e\|_F \|G_{t,i}^e\|_F^2$$
$$\|A_{t,i}^e\|_F \|B_{t,i}^e\|_F \le \eta_\ell^2 C_A C_B G_{\max}^3. \quad (29)$$

where the first inequality is due to the Cauchy–Schwarz inequality, and the second inequality comes from **Assumptions 2** and **3**.

With (28), (29) and the Cauchy–Schwarz inequality, we obtain

$$\|U_{t,i}^{e+1} - U_{t,i}^e\|_F^2 \le 3\eta_\ell^2 (C_A^4 + C_B^4) G_{\max}^2 + 3\eta_\ell^4 C_A^2 C_B^2 G_{\max}^4. \quad (30)$$

Combining the upper bounds above with (27), the second term on the RHS of (27) is bounded, as given by

$$\mathbb{E}\big[f_i(U_{t,i}^{e+1}) - f_i(U_{t,i}^e)\big] \le -\eta_\ell (c_A + c_B)\mathbb{E}\big[\|\bar{G}_{t,i}^e\|_F^2\big] + \eta_\ell^2 C_A C_B G_{\max}^3 + \frac{3L}{2}\eta_\ell^2 (C_A^4 + C_B^4) G_{\max}^2 + \frac{3L}{2}\eta_\ell^4 C_A^2 C_B^2 G_{\max}^4, \quad (31)$$

Summing (31) over $e = 0, \ldots, E-1$ gives

$$\mathbb{E}\big[f_i(U_{t,i}^E) - f_i(U_{t,i})\big] \le -\eta_\ell (c_A + c_B) \sum_{e=0}^{E-1} \mathbb{E}\big[\|\bar{G}_{t,i}^e\|_F^2\big] + E r_{\mathrm{loc}}, \quad (32)$$

where $r_{\mathrm{loc}} = \eta_\ell^2 C_A C_B G_{\max}^3 + \frac{3L}{2}\eta_\ell^2 (C_A^4 + C_B^4) G_{\max}^2 + \frac{3L}{2}\eta_\ell^4 C_A^2 C_B^2 G_{\max}^4$.

For the accumulated local gradients $\sum_{e=0}^{E-1} \mathbb{E}\big[\|\bar{G}_{t,i}^e\|_F^2\big]$ in (32), we have

$$\sum_{e=0}^{E-1} \mathbb{E}\big[\|\bar{G}_{t,i}^e\|_F^2\big] = \mathbb{E}\left[\left\|\bar{G}_{t,i}^0\right\|_F^2 + \sum_{e=1}^{E-1}\left\|\bar{G}_{t,i}^e\right\|_F^2\right] \ge \mathbb{E}\left[\left\|\bar{G}_{t,i}^0\right\|_F^2\right] \ge \kappa E \mathbb{E}\left[\|\nabla f_i(U_{t,i})\|_F^2\right], \quad (33)$$

where $\kappa \in [0, \frac{1}{E}]$. Substituting (33) into the preceding bound yields

$$\mathbb{E}\big[f_i(U_{t,i}^E) - f_i(U_{t,i})\big] \le E r_{\mathrm{loc}} - \eta_\ell (c_A + c_B)\kappa E \mathbb{E}\big[\|\nabla f_i(U_{t,i})\|_F^2\big], \ i \in \mathcal{S}_t. \quad (34)$$

We next bound the changes in the shared factor during local training and server aggregation. Telescoping the corresponding local factor updates gives

$$\|A_{t,i}^E - A_t\|_F \le \eta_\ell E C_B G_{\max}, \text{ for Share-A/Local-B},$$
$$\|B_{t,i}^E - B_t\|_F \le \eta_\ell E C_A G_{\max}, \text{ for Share-B/Local-A}. \quad (35)$$

where both inequalities come from the triangle inequality, **Assumptions 2** and **3**. Consequently, either choice of the shared factor Q satisfies

$$\|Q_{t,i}^E - Q_t\|_F \le \eta_\ell E C_\star G_{\max}, \ i \in \mathcal{S}_t. \quad (36)$$

Based on the server update in (3), we obtain

$$\|Q_{t+1} - Q_t\|_F = \left\|\frac{\eta_g}{S_t}\sum_{k\in\mathcal{S}_t}(Q_{t,k}^E - Q_t)\right\|_F \le \eta_g \eta_\ell E C_\star G_{\max}. \quad (37)$$

where the inequality follows from the triangle inequality and the upper bound of $\|Q_{t,k}^E - Q_t\|_F$ in (36).

After server aggregation, let $U_{t+1,i}$ denote the product formed from Q_{t+1} and client i's retained local factor. For $i \in \mathcal{S}_t$, $U_{t+1,i}$ and $U_{t,i}^E$ have the same local factor and differ only in Q_{t+1} and $Q_{t,i}^E$. For $j \notin \mathcal{S}_t$, $U_{t+1,j}$ and $U_{t,j}$ have the same local factor and differ only in Q_{t+1} and Q_t. Based on Frobenius-norm submultiplicativity, we obtain

$$\|U_{t+1,i} - U_{t,i}^E\|_F \le D_{\mathrm{sel}} := (1+\eta_g)\eta_\ell E C_\star^2 G_{\max}, \ i \in \mathcal{S}_t, \quad (38)$$

$$\|U_{t+1,j} - U_{t,j}\|_F \le D_{\mathrm{uns}} := \eta_g \eta_\ell E C_\star^2 G_{\max}, \ j \notin \mathcal{S}_t. \quad (39)$$

where the selected-client bound uses $\|Q_{t+1} - Q_{t,i}^E\|_F \le \|Q_{t+1} - Q_t\|_F + \|Q_{t,i}^E - Q_t\|_F$, whereas the unselected-client bound contains only $\|Q_{t+1} - Q_t\|_F$.

For a selected client $i \in \mathcal{S}^t$, applying **Assumption 1** between $U_{t,i}^E$ and $U_{t+1,i}$ gives

$$\mathbb{E}\Big[f_i(U_{t+1,i}) - f_i(U_{t,i}^E)\Big] \le \mathbb{E}\Big[\Big\langle \nabla f_i(U_{t,i}^E), U_{t+1,i} - U_{t,i}^E \Big\rangle_F\Big] + \frac{L}{2}\mathbb{E}\Big[\|U_{t+1,i} - U_{t,i}^E\|_F^2\Big] \le \frac{\eta_\ell}{2} G_{\max}^2 + \left(\frac{1}{2\eta_\ell} + \frac{L}{2}\right) D_{\text{sel}}^2, \quad (40)$$

where the second inequality follows from $\langle X, Y\rangle_F \le \frac{\eta_\ell}{2}\|X\|_F^2 + \frac{1}{2\eta_\ell}\|Y\|_F^2$, **Assumption 2**, and (38).

Combining (40) with (34) yields

$$\mathbb{E}[f_i(U_{t+1,i}) - f_i(U_{t,i})] \le -\eta_\ell(c_A + c_B)\kappa E \mathbb{E}\big[\|\nabla f_i(U_{t,i})\|_F^2\big] + \mathcal{R}_{\text{sel}}, \quad (41)$$

where $\mathcal{R}_{\text{sel}} = E\eta_\ell^2 C_A C_B G_{\max}^3 + \frac{3LE}{2}\eta_\ell^2(C_A^4 + C_B^4)G_{\max}^2 + \frac{3LE}{2}\eta_\ell^4 C_A^2 C_B^2 G_{\max}^4 + \frac{\eta_\ell}{2}G_{\max}^2 + \left(\frac{1}{2\eta_\ell} + \frac{L}{2}\right)(1+\eta_g)^2\eta_\ell^2 E^2 C_\star^4 G_{\max}^2$.

For an unselected client $j \notin \mathcal{S}^t$, although it performs no local update, its shared factor changes after aggregation:

$$\mathbb{E}[f_j(U_{t+1,j}) - f_j(U_{t,j})] \le \mathbb{E}\big[\langle \nabla f_j(U_{t,j}), U_{t+1,j} - U_{t,j}\rangle_F\big] + \frac{L}{2}\mathbb{E}\big[\|U_{t+1,j} - U_{t,j}\|_F^2\big] \le \frac{\rho\eta_\ell}{2}\mathbb{E}\big[\|\nabla f_j(U_{t,j})\|_F^2\big] + \mathcal{R}_{\text{uns}}, \quad (42)$$

where $\mathcal{R}_{\text{uns}} := \left(\frac{1}{2\rho\eta_\ell} + \frac{L}{2}\right)\eta_g^2\eta_\ell^2 E^2 C_\star^4 G_{\max}^2$. $\rho > 0$ is the parameter in $\langle X, Y\rangle_F \le \frac{\rho\eta_\ell}{2}\|X\|_F^2 + \frac{1}{2\rho\eta_\ell}\|Y\|_F^2$, and the last inequality uses (39).

Averaging (41) and (42) over all clients $i \in \mathcal{N}$ yields

$$\mathbb{E}[F_{t+1} - F_t] \le \frac{S_t}{N}\mathcal{R}_{\text{sel}} + \left(1 - \frac{S_t}{N}\right)\mathcal{R}_{\text{uns}} - \eta_\ell(c_A + c_B)\kappa E \mathbb{E}\big[\mathcal{G}_t^{\text{sel}}\big] + \frac{\rho\eta_\ell}{2}\mathbb{E}[\mathcal{G}_t^{\text{uns}}], \quad (43)$$

where $\mathcal{G}_t^{\text{sel}} := \frac{1}{N}\sum_{i\in\mathcal{S}_t}\|\nabla f_i(U_{t,i})\|_F^2$, and $\mathcal{G}_t^{\text{uns}} := \frac{1}{N}\sum_{i\notin\mathcal{S}_t}\|\nabla f_i(U_{t,i})\|_F^2$.

For $\mathcal{G}_t := \mathcal{G}_t^{\text{sel}} + \mathcal{G}_t^{\text{uns}} = \frac{1}{N}\sum_{i=1}^N\|\nabla f_i(U_{t,i})\|_F^2$, when $\mathcal{G}_t = 0$, the terms $-\eta_\ell(c_A + c_B)\kappa E\mathbb{E}\big[\mathcal{G}_t^{\text{sel}}\big] + \frac{\rho\eta_\ell}{2}\mathbb{E}[\mathcal{G}_t^{\text{uns}}]$ in (43) vanish. Otherwise, we have

$$-\eta_\ell(c_A + c_B)\kappa E \mathbb{E}\big[\mathcal{G}_t^{\text{sel}}\big] + \frac{\rho\eta_\ell}{2}\mathbb{E}[\mathcal{G}_t^{\text{uns}}] = -\eta_\ell\mathbb{E}\Big[\Big((c_A + c_B)\kappa E\lambda - \frac{\rho}{2}(1-\lambda)\Big)\mathcal{G}_t\Big] \le -\eta_\ell c\,\mathbb{E}[\mathcal{G}_t], \quad (44)$$

where $c := (c_A + c_B)\kappa E\lambda - \frac{\rho}{2}(1-\lambda) > 0$ with $\rho < \frac{2(c_A + c_B)\kappa E\lambda}{(1-\lambda)}$, and λ should satisfy the gradient-mass coverage condition, as given by

$$0 < \lambda \le \frac{\mathcal{G}_t^{\text{sel}}}{\mathcal{G}_t}, \forall t. \quad (45)$$

Here, (45) requires the selected clients to account for at least a fixed fraction λ of the total squared gradient mass in every round with $\mathcal{G}_t > 0$. Equivalently, $G_t^{\text{sel}} \ge \lambda\mathcal{G}_t$ and $G_t^{\text{uns}} \le (1-\lambda)\mathcal{G}_t$. This condition is imposed on the realized gradient mass rather than on the number or sampling distribution of the selected clients. Hence, it allows arbitrary client participation, provided that the selected subset does not capture an arbitrarily small fraction of the current gradient mass. Under full participation, one can set $\lambda = 1$. Under partial participation, any round-independent lower bound $\lambda > 0$ satisfying (45) is sufficient. Together with $c > 0$, this condition ensures that the descent contributed by the selected clients dominates the possible objective increase of the unselected clients caused by the shared-factor update.

Substituting (44) into (43) and rearranging gives

$$\eta_\ell c\,\mathbb{E}[\mathcal{G}_t] \le \mathbb{E}[F_t - F_{t+1}] + \frac{S_t}{N}\mathcal{R}_{\text{sel}} + \left(1 - \frac{S_t}{N}\right)\mathcal{R}_{\text{uns}}. \quad (46)$$

Summing (46) over $t = 0, \dots, T-1$ and dividing by $\eta_l cT$, we obtain

$$\frac{1}{NT}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\mathbb{E}\big[\|\nabla f_i(U_{t,i})\|_F^2\big] \le \frac{F_0 - F_\star}{\eta_\ell cT} + \frac{1}{\eta_\ell cT}\sum_{t=0}^{T-1}\frac{S_t}{N}\mathcal{R}_{\text{sel}} + \frac{1}{\eta_\ell cT}\sum_{t=0}^{T-1}\left(1 - \frac{S_t}{N}\right)\mathcal{R}_{\text{uns}}, \quad (47)$$

which is exactly (14) and completes the proof.

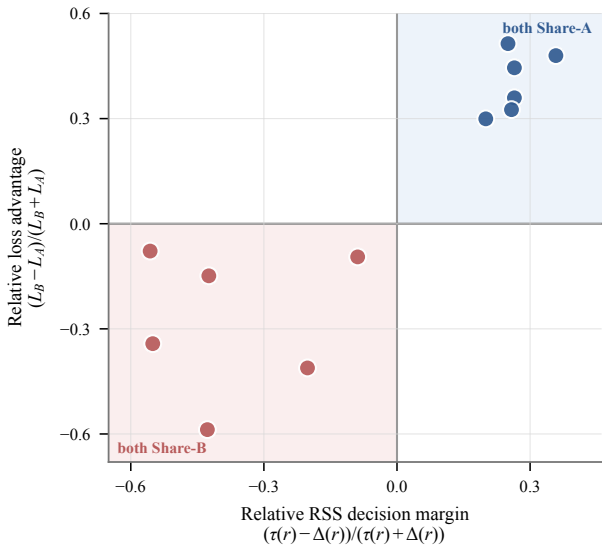


Fig. 3: Alignment Between RSS Decisions and Training-Loss Preference

APPENDIX E
FULL EXPERIMENT RESULTS

Hyperparameters

Tables V and VI show the learning rates used for LoRA-based methods and rsLoRA-based methods, respectively. For the VeRA-based methods, we chose the AdamW optimizer and used separate learning rates for the classification head and the adapted layers as used in VeRA. The learning rates used for VeRA-based methods are shown in Table VII.

Evaluation of RSS-Based Sharing-Side Selection

Alignment with Test-Accuracy Preference.: Table VIII provides the full comparison between the sharing-side decisions made by RSS and the empirical test accuracies of Share-A/Local-B and Share-B/Local-A. The experiments cover six evaluation sets, multiple client partitions, and

LoRA ranks $r \in \{2, 4, 8, 16\}$. For each setting, RSS is computed before federated training, while the empirical winner is determined by comparing the test accuracies obtained at the best validation rounds of the two sharing strategies.

Across the 38 evaluated settings, the RSS decision agrees with the empirically better sharing strategy in all 33 cases with a strict accuracy difference. The remaining five cases are marked as empirical ties in Table VIII. The preferred sharing side also changes with the data partition and LoRA rank. For example, QNLI and MNLI favor Share-A/Local-B under the evaluated IID settings, whereas Share-B/Local-A is preferred in most of their Dirichlet settings. For MNLI under Dirichlet $\alpha = 1$, the preferred strategy further changes from Share-B/Local-A at $r = 4$ to Share-A/Local-B at $r = 8$. Similarly, SST-2 favors Share-A/Local-B under IID partitions but mainly favors Share-B/Local-A under the non-IID partitions. These results support the use of a partition- and rank-aware selection rule instead of a fixed sharing policy. RSS should nevertheless be interpreted as a sharing-side selector rather than an estimator of the final accuracy gap between the two strategies.

Alignment with the training objective.: Table VIII compares the RSS decision with the final test-accuracy preference of the two fixed sharing strategies. We further examine whether the decision direction of RSS agrees with their relative behavior under the actual training objective. For visual illustration, we randomly sample several evaluated settings and compare their RSS decision margins with the corresponding training losses of Share-A/Local-B and Share-B/Local-A.

For rank r, we define the relative RSS decision margin as

$$m_{\mathrm{RSS}}(r) = \frac{\tau(r) - \Delta(r)}{\tau(r) + \Delta(r)}. \tag{48}$$

Since RSS selects Share-A/Local-B when $\Delta(r) \leq \tau(r)$, a positive $m_{\mathrm{RSS}}(r)$ favors Share-A/Local-B, whereas a negative value favors Share-B/Local-A. We also define the relative training-loss advantage as

$$m_{\mathrm{loss}} = \frac{L_B - L_A}{L_B + L_A}, \tag{49}$$

where L_A and L_B denote the training losses obtained by Share-A/Local-B and Share-B/Local-A, respectively. Accordingly, $m_{\mathrm{loss}} > 0$ means that Share-A/Local-B attains a lower training loss, while $m_{\mathrm{loss}} < 0$ means that Share-B/Local-A attains a lower training loss.

As shown in Fig. 3, all sampled cases lie in either the upper-right or lower-left quadrant. Thus, the sharing side selected by RSS also attains the lower training loss in these cases. This result provides additional evidence that the RSS decision is consistent with the optimization behavior of the two sharing strategies under the actual task objective.

This directional agreement provides empirical support for the residual-based selection principle in Theorem 1. The theorem shows that the preferred sharing side is determined by which shared-subspace constraint yields the smaller aggregate projection residual. RSS translates this principle into a training-free decision rule by assessing whether the common rank-r input-side subspace required by Share-A/Local-B is sufficient for the client representations. When this subspace is sufficient, RSS favors Share-A/Local-B; otherwise, it favors Share-B/Local-A. The consistent training-loss preference therefore indicates that RSS identifies the sharing constraint that better fits the client data under the actual task objective. This result supports RSS as a practical proxy for the residual-based sharing-side preference characterized in Theorem VIII.

Method	MNLI-m	MNLI-mm	SST-2	QNLI	QQP	RTE
LoRA	1E-2	1E-2	2E-2	1E-2	1E-2	1E-2
FFA-LoRA	5E-2	5E-2	5E-2	2E-2	5E-2	2E-2
FedDPA-LoRA	1E-2	1E-2	1E-2	5E-2	5E-2	1E-2
FedSA-LoRA	2E-2	2E-2	1E-2	5E-3	2E-2	1E-2
FedAS-LoRA	2E-2	2E-2	1E-2	5E-3	2E-2	1E-2

TABLE V: Learning rates used for LoRA-based methods on the GLUE benchmark.

Method	MNLI-m	MNLI-mm	SST-2	QNLI	QQP	RTE
rsLoRA	5E-3	5E-3	1E-2	2E-3	5E-3	2E-3
FFA-rsLoRA	2E-2	2E-2	2E-2	1E-2	2E-2	1E-2
FedDPA-rsLoRA	5E-3	5E-3	1E-2	1E-3	5E-3	1E-2
FedSA-rsLoRA	5E-3	5E-3	5E-3	1E-3	2E-3	2E-3
FedAS-rsLoRA	5E-3	5E-3	5E-3	1E-3	2E-3	2E-3

TABLE VI: Learning rates used for rsLoRA-based methods on the GLUE benchmark.

Method	Position	MNLI-m	MNLI-mm	SST-2	QNLI	QQP	RTE
VeRA	VeRA	1E-2	1E-2	2E-2	2E-3	2E-3	1E-2
	Head	6E-3	6E-3	2E-3	3E-4	3E-4	2E-4
FFA-VeRA	VeRA	2E-2	2E-2	1E-2	1E-2	1E-2	1E-2
	Head	2E-3	2E-3	6E-3	2E-4	6E-3	2E-4
FedDPA-VeRA	VeRA	1E-2	1E-2	1E-2	2E-3	2E-2	1E-2
	Head	6E-3	6E-3	6E-3	3E-4	2E-3	2E-4
FedSA-VeRA	VeRA	2E-3	2E-3	1E-2	1E-2	2E-3	1E-2
	Head	3E-5	3E-5	3E-4	3E-4	3E-4	1E-4
FedAS-VeRA	VeRA	2E-3	2E-3	1E-2	1E-2	2E-3	1E-2
	Head	3E-5	3E-5	3E-4	3E-4	3E-4	1E-4

TABLE VII: Learning rates used for VeRA-based methods on the GLUE benchmark.

Scalability Analysis under Uniform and Non-Uniform Client Sampling

To further evaluate scalability, we increase the number of clients to $N = 50$ under uniform and non-uniform sampling. As shown in Table IX, FedAS-LoRA achieves the highest accuracy on QNLI, SST-2, and MNLI-m under both sampling settings. Compared with FedSA-LoRA, its average accuracy is higher by 0.85 and 0.76 percentage points under uniform and non-uniform sampling, respectively. These results show that selecting the sharing side

Dataset	Split	r	RSS metric			Accuracy (%)			Alignment
			$\Delta(r)$	$\tau(r)$	Decision	Share-A	Share-B	Winner	
QNLI	IID	4	2.46413×10^{-5}	3.91172×10^{-5}	Share-A	**92.26**	90.18	**Share-A**	Match
		8	4.28222×10^{-5}	7.07299×10^{-5}	Share-A	**91.12**	90.66	**Share-A**	Match
	Dirichlet $\alpha = 0.5$	2	3.01960×10^{-5}	3.46345×10^{-5}	Share-A	**89.46**	88.84	**Share-A**	Match
		4	5.89316×10^{-5}	3.80830×10^{-5}	Share-B	92.01	**92.80**	**Share-B**	Match
		8	7.37366×10^{-5}	6.79052×10^{-5}	Share-B	91.13	**92.94**	**Share-B**	Match
		16	1.33945×10^{-4}	1.31365×10^{-4}	Share-B	89.12	**89.56**	**Share-B**	Match
	Dirichlet $\alpha = 1$	4	5.90747×10^{-5}	3.92434×10^{-5}	Share-B	93.28	**94.69**	**Share-B**	Match
		8	8.31570×10^{-5}	6.96169×10^{-5}	Share-B	90.89	**91.58**	**Share-B**	Match
QQP	Dirichlet $\alpha = 0.5$	4	2.11905×10^{-5}	1.28376×10^{-5}	Share-B	85.68	**86.70**	**Share-B**	Match
		8	4.14507×10^{-5}	1.67622×10^{-5}	Share-B	86.87	**87.95**	**Share-B**	Match
SST-2	IID	4	3.89196×10^{-4}	5.07871×10^{-4}	Share-A	**96.65**	96.14	**Share-A**	Match
		8	4.35904×10^{-4}	6.02930×10^{-4}	Share-A	**96.10**	95.87	**Share-A**	Match
	Dirichlet $\alpha = 0.5$	2	1.45398×10^{-3}	5.83336×10^{-4}	Share-B	96.56	**96.79**	**Share-B**	Match
		4	1.84450×10^{-3}	5.25775×10^{-4}	Share-B	**96.33**	**96.33**	**Tie**	Actual tie
		8	2.20251×10^{-3}	6.38468×10^{-4}	Share-B	95.72	**97.17**	**Share-B**	Match
		16	3.63222×10^{-3}	1.01451×10^{-3}	Share-B	**95.73**	95.72	**Tie**	Actual tie
	Dirichlet $\alpha = 1$	4	1.18446×10^{-3}	5.79445×10^{-4}	Share-B	**97.02**	**97.02**	**Tie**	Actual tie
		8	1.43308×10^{-3}	6.62178×10^{-4}	Share-B	96.56	**96.79**	**Share-B**	Match
MNLI-m	IID	4	5.98420×10^{-6}	1.02906×10^{-5}	Share-A	**87.06**	86.11	**Share-A**	Match
		8	2.91890×10^{-5}	4.13822×10^{-5}	Share-A	**89.43**	89.06	**Share-A**	Match
	Dirichlet $\alpha = 0.5$	2	1.03136×10^{-4}	1.11881×10^{-5}	Share-B	89.60	**89.63**	**Share-B**	Match
		4	4.62360×10^{-5}	9.96230×10^{-6}	Share-B	90.04	**90.28**	**Share-B**	Match
		8	9.33286×10^{-5}	3.98546×10^{-5}	Share-B	89.75	**89.95**	**Share-B**	Match
		16	8.07826×10^{-5}	3.45278×10^{-5}	Share-B	89.04	**89.73**	**Share-B**	Match
	Dirichlet $\alpha = 1$	4	1.37597×10^{-5}	1.02455×10^{-5}	Share-B	89.50	**89.62**	**Share-B**	Match
		8	3.26219×10^{-5}	3.80408×10^{-5}	Share-A	**89.02**	88.37	**Share-A**	Match
MNLI-mm	IID	4	5.98420×10^{-6}	1.02906×10^{-5}	Share-A	**86.68**	85.23	**Share-A**	Match
		8	2.91890×10^{-5}	4.13822×10^{-5}	Share-A	**88.07**	87.91	**Share-A**	Match
	Dirichlet $\alpha = 0.5$	4	4.62360×10^{-5}	9.96230×10^{-6}	Share-B	88.47	**89.97**	**Share-B**	Match
		8	9.33286×10^{-5}	3.98546×10^{-5}	Share-B	87.82	**88.86**	**Share-B**	Match
	Dirichlet $\alpha = 1$	4	1.37597×10^{-5}	1.02455×10^{-5}	Share-B	87.14	**88.79**	**Share-B**	Match
		8	3.26219×10^{-5}	3.80408×10^{-5}	Share-A	**90.44**	89.97	**Share-A**	Match
RTE	IID	4	2.07749×10^{-3}	3.52261×10^{-3}	Share-A	**88.14**	85.82	**Share-A**	Match
		8	3.99091×10^{-3}	5.98761×10^{-3}	Share-A	**87.50**	**87.50**	**Tie**	Actual tie
	Dirichlet $\alpha = 0.5$	4	1.81187×10^{-3}	3.83366×10^{-3}	Share-A	**88.71**	**88.71**	**Tie**	Actual tie
		8	3.63808×10^{-3}	5.98369×10^{-3}	Share-A	**87.77**	86.34	**Share-A**	Match
	Dirichlet $\alpha = 1$	4	1.82618×10^{-3}	3.55810×10^{-3}	Share-A	**87.19**	86.47	**Share-A**	Match
		8	3.45208×10^{-3}	5.75546×10^{-3}	Share-A	**86.26**	85.35	**Share-A**	Match

TABLE VIII: RSS metric values, threshold decisions, and Share-A/Local-B versus Share-B/Local-A accuracies under the evaluated three-client settings.

remains beneficial with a larger client population and uneven participation probabilities.

Results under Structured Input Skew

We further evaluate the compared methods under two structured input-skew partitions while controlling the client label distributions. For SST-2, we construct a label-balanced input-length-skew partition. Within each sentiment label, the samples are ordered according to their input lengths and divided into three non-overlapping groups of approximately equal size. Each client receives the corresponding length group from both sentiment labels. This construction keeps the positive and negative label proportions similar across clients, while making the client input-length distributions different.

For MNLI, we construct a genre-based partition using the genre annotations provided by MultiNLI Williams et al. [2018]. The three clients contain premise–hypothesis pairs from the telephone, government, and fiction genres, respectively. We balance the entailment, neutral, and contradiction

Method	Uniform	Non-uniform
LoRA	87.27 ±0.48	83.24 ±0.72
FFA-LoRA	86.18 ±0.45	84.90 ±0.85
FedDPA-LoRA	87.50 ±0.41	84.29 ±0.64
FedSA-LoRA	88.12 ±0.55	85.68 ±0.44
FedAS-LoRA (Ours)	**89.37** ±0.37	**86.80** ±0.56
LoRA	93.42 ±0.67	92.80 ±0.77
FFA-LoRA	93.61 ±0.71	93.51 ±0.78
FedDPA-LoRA	94.87 ±0.52	94.70 ±0.89
FedSA-LoRA	95.01 ±0.58	96.48 ±0.77
FedAS-LoRA (Ours)	**95.64** ±0.73	**96.89** ±0.79
LoRA	83.91 ±0.53	81.25 ±0.60
FFA-LoRA	84.23 ±0.56	83.93 ±0.99
FedDPA-LoRA	85.07 ±0.80	83.19 ±0.95
FedSA-LoRA	86.89 ±0.83	84.28 ±0.99
FedAS-LoRA (Ours)	**87.55** ±0.91	**85.04** ±0.84

TABLE IX: Performance on the QNLI, SST-2, and MNLI-m tasks under uniform and non-uniform client sampling with $N = 50$. From top to bottom, the three blocks correspond to QNLI, SST-2, and MNLI-m, respectively.

Rank	Method	SST-2	MNLI-m
$r = 4$	LoRA	95.87 ±0.59	86.63 ±0.21
	FFA-LoRA	95.56 ±0.39	85.44 ±0.42
	FedSA-LoRA	92.98 ±0.45	87.07 ±0.11
	FedAS-LoRA (Ours)	**96.47** ±0.19	**87.43** ±0.20
$r = 8$	LoRA	95.18 ±0.48	86.41 ±0.29
	FFA-LoRA	93.35 ±0.52	83.37 ±0.10
	FedSA-LoRA	94.04 ±0.44	86.19 ±0.11
	FedAS-LoRA (Ours)	**96.56** ±0.04	**86.53** ±0.34

TABLE X: Test accuracy under structured input-skew partitions with different LoRA ranks r. SST-2 uses a label-balanced input-length-skew partition, while MNLI-m uses a label-balanced genre partition with telephone, government, and fiction clients.

labels across the clients so that the main source of heterogeneity is the textual genre rather than the label distribution. We report performance on the MNLI matched evaluation set, denoted as MNLI-m.

As shown in Table X, FedAS-LoRA achieves the highest accuracy in all four evaluated settings. On SST-2, it obtains accuracies of 96.47% and 96.56% at $r = 4$ and $r = 8$, respectively, exceeding LoRA by 0.60 and 1.38 percentage points. On MNLI-m, FedAS-LoRA reaches 87.43% at $r = 4$ and 86.53% at $r = 8$, improving over LoRA by 0.80 and 0.12 percentage points. It also exceeds FFA-LoRA by 1.99 and 3.16 percentage points, and FedSA-LoRA by 0.36 and 0.34 percentage points at $r = 4$ and $r = 8$, respectively. These results show that selecting the sharing side remains beneficial when clients mainly differ in input characteristics rather than label proportions.

Results on GSM8K with Llama 3 8B

The results on the GSM8K dataset are shown in Table XI, demonstrating that the proposed FedAS-LoRA outperforms other methods in complex natural language generation tasks. From the given example, it can be seen that both LoRA and FFA-LoRA have reasoning errors, but FedAS-LoRA can reason accurately, demonstrating the superiority of the proposed method.

Communication Cost

Table XII compares the trainable parameter count, per-round communicated model size, per-round computation time, and the communication-round index of the best validation checkpoint. Share-A/Local-B and Share-B/Local-A train both LoRA factors and therefore retain the same number of trainable parameters as LoRA, i.e., 1.839M on MNLI-m and 1.838M on SST-2. However, each strategy communicates only the selected shared factor. Their per-round communicated model size is therefore 0.393M parameters, which is 50% lower than the 0.786M parameters communicated by LoRA and FedDPA-LoRA.

The reduction in per-round communication does not introduce substantial additional local computation. On MNLI-m, the per-round computation times of Share-A/Local-B and Share-B/Local-A are 14.14s and 14.18s, respectively, compared with 14.12s for LoRA. On SST-2, the corresponding times are 6.25s and 6.26s, compared with 6.24s for LoRA. FFA-LoRA has a slightly lower per-round computation time because it freezes one factor, whereas the two sharing strategies keep both factors trainable. FedDPA-LoRA incurs a higher parameter and computation cost because it maintains additional global and personalized LoRA modules. These results show that factor-wise sharing reduces the communicated model size per round while preserving the training capacity of both LoRA factors.

Relative Heterogeneity Amplification of Local LoRA Changes

To further examine how client data heterogeneity affects the two LoRA factors, we compare the client-specific changes of A and B after local training on MNLI. We consider three non-IID partitioning schemes: Dirichlet distributions with $\alpha = 1$ and $\alpha = 0.5$, and the label-balanced MNLI-genre partition. For each scheme, we independently generate five three-client partitions using different partition seeds. Each non-IID partition is paired with an IID partition generated using the same seed. The model initialization and all training-related random seeds are fixed across the partition instances. Therefore, the variation across the five instances is caused by the client data partitions rather than by model initialization or stochastic training.

For this analysis, the clients perform local training independently without server aggregation. We measure the change in each LoRA factor relative to its common initialization as ΔQ_i. We analyze these changes rather than the terminal factor values to exclude the initial factor values

	LoRA	FFA-LoRA	FedAS-LoRA
Accuracy	55.14 $\pm$0.25	54.51 $\pm$0.32	**56.28** $\pm$0.18
Question:	Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, mealworms and vegetables to help keep them healthy. She gives the chickens their feed in three separate meals. In the morning, she gives her flock of chickens 15 cups of feed. In the afternoon, she gives her chickens another 25 cups of feed. How many cups of feed does she need to give her chickens in the final meal of the day if the size of Wendi's flock is 20 chickens?		
Output:	Wendi gives her chickens 15 cups of feed in the morning and 25 cups of feed in the afternoon, for a total of $15 + 25 = 40$ cups of feed. She has 20 chickens, so each chicken needs $40/20 = 2$ cups of feed in the final meal of the day. The answer is 2. **[✗]**	Wendi feeds her chickens 15 cups of feed in the morning and 25 cups of feed in the afternoon. So she has already given them $15 + 25 = 40$ cups of feed. She has 20 chickens. So she needs to give them $40/20 = 2$ cups of feed each. The answer is 2. **[✗]**	Wendi has 15 cups of feed in the morning and 25 cups in the afternoon. So she has $15 + 25 = 40$ cups of feed. She has 20 chickens and she needs to give each chicken 3 cups of feed. So she needs $20 \times 3 = 60$ cups of feed. She has 40 cups of feed and she needs 60 cups of feed. So she needs $60 - 40 = 20$ more cups of feed. The answer is 20. **[✓]**

TABLE XI: Performance of different methods on the GSM8K dataset and example answers generated by each method.

Method	Trainable Parm. (MNLI-m / SST-2)	Per-round Communicated Parm.	Per-round Computation Cost		Communication Round	
			MNLI-m	SST-2	MNLI-m	SST-2
LoRA	1.839M / 1.838M	0.786M	14.12s	6.24s	186	210
FFA-LoRA	1.446M / 1.445M	0.393M	13.94s	6.04s	477	467
FedDPA-LoRA	2.626M / 2.625M	0.786M	19.46s	11.56s	454	398
FedSA-LoRA	1.839M / 1.838M	0.393M	14.14s	6.25s	495	453
FedAS-LoRA	1.839M / 1.838M	0.393M	14.18s	6.26s	472	384

TABLE XII: Time and space costs for each method on the MNLI-m and SST-2 tasks. Communication Round denotes the communication-round index at which the best validation performance is obtained.

from the interpretation. This is particularly important for factor A, which is randomly initialized with nonzero values under the standard LoRA initialization, whereas factor B is initialized to zero. Using ΔQ_i therefore places both factors on the same reference and focuses the analysis on the changes learned from each client's local data.

For a given partition and factor Q, we quantify the cross-client disagreement using the mean pairwise Frobenius distance

$$D_Q = \frac{2}{N(N-1)} \sum_{1 \le i < j \le N} \|\Delta Q_i - \Delta Q_j\|_F . \tag{50}$$

For each non-IID partition instance, the relative heterogeneity amplification is defined with respect to its seed-matched IID reference as

$$\mathrm{RHA} = \frac{D_A^{\text{non-IID}} / D_A^{\text{IID}}}{D_B^{\text{non-IID}} / D_B^{\text{IID}}}. \tag{51}$$

An RHA value greater than one means that changing from the paired IID partition to the non-IID partition produces a larger proportional increase in the cross-client disagreement of ΔA than in that of ΔB. Conversely, an RHA value below one indicates a larger relative increase in the disagreement of ΔB. Since RHA compares the non-IID-to-IID amplification of each factor, $\mathrm{RHA} > 1$ does not necessarily imply that $D_A > D_B$ in absolute value.

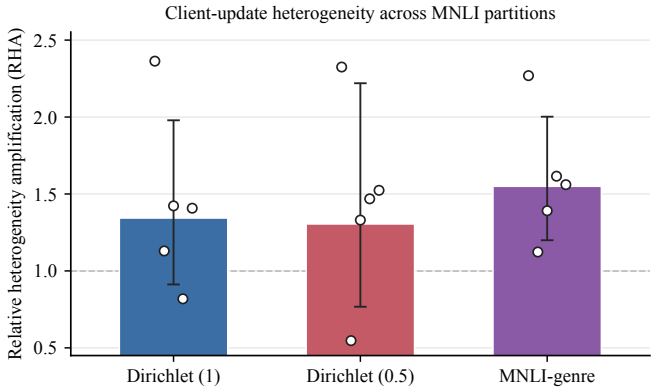
As shown in Fig. 4(a), the geometric-mean RHA is above one under all three non-IID partitioning schemes. Relative to their seed-matched IID references, the evaluated non-IID partitions therefore amplify the cross-client disagreement of ΔA more strongly than that of ΔB on average. Among the three settings, the MNLI-genre partition yields the largest overall RHA, and the five individual partition instances all remain above the $\mathrm{RHA} = 1$ reference line. This result indicates a relatively stable amplification of ΔA disagreement when the clients are separated by textual genre.

The two Dirichlet settings show greater variation across partition instances. Although their geometric-mean RHA values are above one, each setting includes an individual instance with an RHA below one. Hence, the Dirichlet concentration parameter alone does not determine whether client heterogeneity increases the relative disagreement of ΔA or ΔB more strongly. Even under the same α, the result depends on the realized client partition. A fixed sharing policy based only on the nominal heterogeneity level is therefore insufficient, which supports the partition-aware design of RSS.

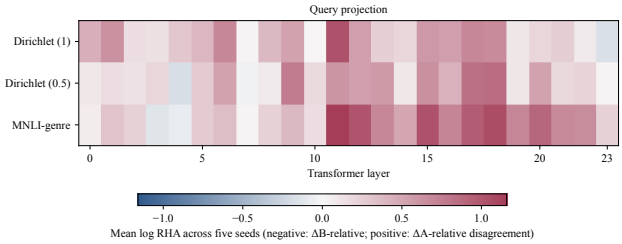
Figs. 4(b) and 4(c) further show that the relative amplification varies across Transformer layers and projection types. For the query projection, the mean log RHA is positive in most layers under the three partitioning schemes. The

positive values are particularly evident in several middle and later layers under the MNLI-genre partition, indicating stronger relative disagreement in ΔA at these layers. The value projection exhibits a less uniform pattern. While many early and middle layers have positive mean log RHA, several later layers under the Dirichlet partitions have negative values, indicating stronger relative disagreement in ΔB. These mixed layer-wise patterns show that an average factor-level trend does not fully characterize the behavior of all adapted layers and projections. This further supports evaluating shared-subspace sufficiency from the representations of the actual client partition, as done by RSS.

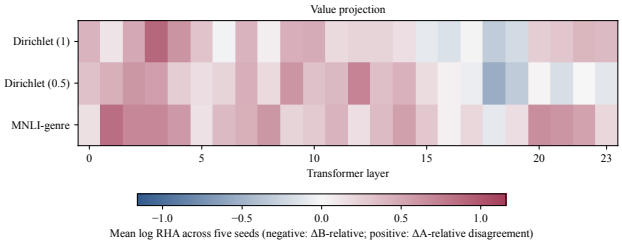
Overall, the relative disagreement between the two LoRA factors depends on the realized client partition and varies across Transformer layers and projection types. Therefore, neither the Dirichlet concentration parameter nor an average factor-level trend is sufficient to prescribe a fixed sharing side across federated settings. RSS addresses this issue by assessing, for the current client partition and target LoRA rank r, whether a shared rank-r input subspace is sufficient for the local data distributions. These results support selecting the sharing side according to the observed client partition rather than applying a fixed factor-sharing policy.



(a) Overall relative heterogeneity amplification.



(b) Layer-wise RHA for the query projection.



(c) Layer-wise RHA for the value projection.

Fig. 4: Relative heterogeneity amplification of LoRA factors across MNLI client partitions. The clients perform local training independently without server aggregation. For each of the three non-IID partitioning schemes, five three-client partitions are independently generated using different partition seeds. Each non-IID partition is paired with an IID reference partition generated using the same seed, while the model initialization and training-related random seeds are fixed across all partition instances. (a) Bars show the geometric mean across the five partition instances, white circles denote individual partition instances, and error bars represent one sample standard deviation in log-RHA space. The dashed line marks $\mathrm{RHA} = 1$. (b,c) Layer-wise mean log RHA for the query and value projections, respectively. Both heatmaps use the same color scale; positive values indicate stronger relative disagreement in ΔA, whereas negative values indicate stronger relative disagreement in ΔB.