FEDEDS: DISCOVERING AND SPECIALIZING LORA EXPERTS FOR HETEROGENEOUS FEDERATED FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In practical federated fine-tuning of large pretrained models, clients often pursue different downstream tasks, while clients working on the same task may exhibit skewed label distributions. The coexistence of task heterogeneity and intra-task label skew poses challenges to knowledge sharing among clients. In this setting, multi-expert LoRA enables cluster-specific collaboration, but it requires both reliable client grouping and experts with genuinely distinct low-rank updates. We find that intra-task non-IID data degrade the task-discriminative signal of conventional LoRA factors, making client clustering unreliable, and that even correctly grouped clients can produce experts with highly overlapping low-rank subspaces. We propose FedEDS, a federated multi-expert LoRA framework that jointly addresses these challenges. FedEDS builds task-discriminative client representations from centered query-A factors, adaptively clusters clients without task labels or a predefined number of experts, initializes group-specific experts via effective-update aggregation and truncated SVD, and encourages expert specialization through subspace overlap regularization. Extensive experiments demonstrate that FedEDS outperforms existing federated LoRA methods while achieving reliable client clustering and effective expert specialization.

1 INTRODUCTION

Large language models (LLMs) have shown strong general capabilities in natural language understanding and generation (Brown et al., 2020), code generation (Chen et al., 2021), and complex reasoning tasks (Wei et al., 2022), providing a foundation for adaptation across domains and tasks. However, deploying these capabilities across diverse domains and downstream tasks typically requires fine-tuning with high-quality domain-specific data. In practice, such data are distributed across institutions, edge devices, and user populations. Privacy regulations, institutional policies, and commercial interests restrict their centralized collection (Kairouz & McMahan, 2021). Federated learning (FL) offers a natural alternative: clients collaboratively train a model while keeping their raw data local (McMahan et al., 2017).

Although FL avoids the direct exchange of raw data, full fine-tuning of large pretrained models still incurs substantial computation, memory, and communication costs (Guo et al., 2025). Low-rank adaptation (LoRA) freezes the pretrained weights and parameterizes each weight update as $\Delta W = BA$, where the rank r is much smaller than the input and output dimensions. This reduces the number of trainable parameters (Hu et al., 2022), allowing clients to optimize and exchange compact low-rank adapters with lower local training and communication costs (Sun et al., 2024). Existing federated LoRA methods have explored factor sharing, aggregation fidelity, and communication-efficient training (Chen et al., 2025; Singhal et al., 2025; Bian et al., 2025; Bai et al., 2024). While these advances improve how client updates are shared and aggregated, they do not explicitly determine which clients should collaborate.

This limitation becomes important when task heterogeneity and intra-task statistical heterogeneity coexist. Different clients may target different downstream tasks (Zhang et al., 2024; Wang et al., 2025), while clients on the same task may still exhibit different label distributions (Zhang et al., 2022), leading to client drift (Karimireddy et al., 2020). A single global adapter risks inter-task

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

interference, whereas fully local adapters prevent knowledge sharing among clients working on the same task (Tian et al., 2024; Yang et al., 2025; Wang et al., 2025). Multi-expert LoRA offers a compromise through group-specific collaboration (Wang et al., 2025), but its effectiveness hinges on accurate client grouping and complementary expert specialization.

Two challenges remain insufficiently addressed. The first one concerns how to derive a reliable grouping from client updates. FedLEASE (Wang et al., 2025) and FedTreeLoRA (Bian et al., 2026b) organize expert assignment or hierarchical sharing using client relationships reflected in LoRA B matrices. However, cross-client similarity of B factors decreases as data heterogeneity increases, even among clients working on the same task (Guo et al., 2025). When task heterogeneity coexists with intra-task label distribution skew, B-factor similarity may therefore fail to reliably reveal latent task relationships. The second challenge lies in ensuring effective specialization among the learned experts. Even correctly grouped clients may produce LoRA experts with highly overlapping low-rank subspaces, limiting functional differentiation. Existing expert assignment methods mainly organize sharing according to clients (Wang et al., 2025), model hierarchies (Bian et al., 2026b), or local patterns (Duan et al., 2026), with less emphasis on explicitly separating the resulting expert representations. We therefore ask: **(1) How can we extract task-discriminative representations from LoRA updates that remain robust to intra-task data heterogeneity? (2) How can we reduce overlap between the low-rank update subspaces of different experts to encourage complementary specialization?**

To answer these questions, we empirically analyze federated LoRA under simultaneous task heterogeneity and intra-task non-IID data. We make two observations. **First,** centering query-A across clients reveals clearer and more stable task clusters. **Second,** the input subspaces of different LoRA experts can remain highly overlapping even when clients are grouped by their true tasks. These observations motivate FedEDS, a federated multi-expert LoRA framework for adaptive expert discovery and specialization. FedEDS constructs client task signatures from centered query-A and applies affinity propagation (AP) clustering (Frey & Dueck, 2007) to adaptively determine client clusters and the number of experts. It then aggregates LoRA updates within each group in effective weight update space and constructs compact experts through truncated SVD. During federated training, FedEDS introduces a QR-based input subspace overlap regularizer to encourage distinct adaptation directions across experts.

Our main contributions are as follows:

- We build task signatures from cross-client centered query-A and apply affinity propagation to jointly determine collaboration groups and the number of experts without task labels, achieving perfect clustering across all evaluated label-skew settings.
- We aggregate grouped clients in effective weight-update space, initialize experts via truncated SVD, and regularize expert input-subspace overlap through principal angles, reducing A-overlap from 0.8643 to 0.0061 and bridging client grouping with expert specialization.
- Extensive experiments demonstrate that FedEDS outperforms existing federated LoRA methods under task and statistical heterogeneity. It reaches 88.13% average accuracy on four GLUE tasks, outperforming FedIT by 10.02 pp and the strongest baseline by 1.01 pp.

2 Related Work

2.1 Personalized Federated Learning

Personalized federated learning (PFL) combines cross-client collaboration with adaptation to heterogeneous local data. Existing methods introduce personalization through optimization, parameter decomposition, or relationship modeling. Optimization-based methods learn initializations suitable for local adaptation, as in Per-FedAvg (Fallah et al., 2020), or regularize client-specific models toward a global solution, as in Ditto (Li et al., 2021). Parameter decomposition methods separate shared and private components. For instance, FedRep (Collins et al., 2021) learns a shared representation while retaining client-specific task heads. Relationship-based methods include MOCHA's coupled objectives (Smith et al., 2017), IFCA's alternating cluster assignment (Ghosh et al., 2020), FedEM's latent mixtures (Marfoq et al., 2021), CFL's update-cosine clustering (Sattler et al., 2021), and PACFL's data-subspace clustering (Vahidian et al., 2023). These methods assume that clients

cooperate on a shared learning objective and differ mainly in their data distributions. In federated fine-tuning of large models, clients may instead target different downstream tasks while exhibiting non-IID label distributions even within the same task, and existing PFL methods largely address these two forms of heterogeneity in isolation. In contrast, we treat task heterogeneity and intra-task label skew as a joint setting and address both the collaboration structure and the separation of the resulting experts.

2.2 Federated Fine-Tuning with LoRA

Federated LoRA research has explored parameter sharing, aggregation, and expert assignment. Parameter-sharing methods determine which LoRA parameters participate in federated collaboration. FFA-LoRA freezes the factor with nonzero initialization (Sun et al., 2024), FedSA-LoRA shares A while keeping B local (Guo et al., 2025), and FedDPA combines global and local adapters (Yang et al., 2024). Aggregation methods aim to reduce errors arising from factor-wise averaging and heterogeneous client ranks. FLoRA stacks low-rank factors (Wang et al., 2024), FlexLoRA applies SVD to accommodate heterogeneous client ranks (Bai et al., 2024), and LoRA-FAIR improves aggregation and initialization across rounds (Bian et al., 2025). While these methods improve parameter sharing and aggregation, expert-assignment methods further determine which clients should collaborate (Zhang et al., 2026). FedLEASE clusters warmup B factors and selects K by silhouette score (Wang et al., 2025); FL-TAC clusters low-rank adapters with task-count-based k-means (Ping et al., 2024); FedTreeLoRA learns hierarchical sharing (Bian et al., 2026b). However, existing methods primarily focus on client collaboration structures, with limited attention to the relationships among learned expert representations. In contrast, we explicitly regularize expert subspaces to reduce representational overlap and encourage complementary specialization.

3 Preliminaries and Motivation

3.1 Low-Rank Adaptation and Mixture of LoRA Experts

For a frozen pretrained weight matrix $W^0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, LoRA (Hu et al., 2022) parameterizes the adapted weight as $W = W^0 + sBA$, where $A \in \mathbb{R}^{r \times d_{\text{in}}}$ and $B \in \mathbb{R}^{d_{\text{out}} \times r}$ are trainable low-rank factors, $r \ll \min(d_{\text{in}}, d_{\text{out}})$ is the adapter rank, and s is the scaling factor. A mixture of LoRA experts (LoRA-MoE) (Wu et al., 2024) combines multiple adapters over a shared frozen backbone through input-dependent routing. Given K experts with factors (A_k, B_k), the combined update for input x is

$$\Delta W(x) = \sum_{k=1}^{K} \pi_k(x) \Delta W_k = \sum_{k=1}^{K} \pi_k(x) B_k A_k, \qquad (1)$$

where $\pi_k(x)$ denotes the input-dependent routing weight of expert k. Routing can combine all experts (Wu et al., 2024) or activate only a subset (Liu & Luo, 2024). By assigning relatively independent low-rank parameter spaces to different tasks, LoRA-MoE enables specialization while retaining a shared backbone and can mitigate cross-task interference (Tian et al., 2024). However, parameter-level separation does not guarantee differentiated expert representations, as examined in Section 3.3.

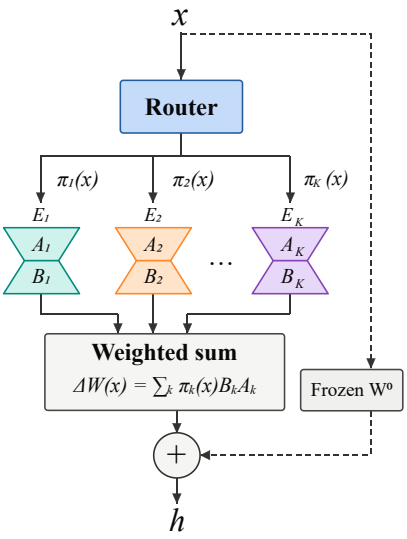


Figure 1: Illustration of LoRA-MoE.

3.2 Problem Formulation

Consider a federated LoRA fine-tuning system with N clients. Client i holds a local dataset $\mathcal{D}_i$. All clients share the same frozen pretrained model and update only their local LoRA parameters. Let z_i denote the latent downstream task of client i. When $z_i \neq z_j$, the clients have different task objectives, resulting in task heterogeneity. Even when $z_i = z_j$, their local label distributions may

satisfy $P_i(y \mid z_i) \neq P_j(y \mid z_j)$, where y denotes a class label, and $P_i(y \mid z_i)$ denotes the label distribution of client i conditioned on its task. This constitutes intra-task label distribution skew.

We learn K LoRA experts $\mathcal{E} = \{E_1, \ldots, E_K\}$. Let $\gamma : \{1, \ldots, N\} \to \{1, \ldots, K\}$ map clients to expert groups. The expert bank is combined through input-dependent routing, so the prediction for an input x is $f(x; W^0 + \Delta W(x))$ with $\Delta W(x)$ given by Eq. (1). The overall learning objective is

$$\min_{K,\, \gamma,\, \mathcal{E},\, \{\phi_g\}_{g=1}^{K}} \sum_{i=1}^{N} p_i \, \mathbb{E}_{(x,y)\sim\mathcal{D}_i} \left[\ell\big(f\big(x; W^0 + \Delta W(x)\big), y\big) \right], \tag{2}$$

where p_i is the aggregation weight of client i, (x, y) is an input–label pair from $\mathcal{D}_i$, ϕ_g denotes the router parameters shared among clients in cluster g, f denotes the prediction model, and ℓ denotes the per-example task loss. Without access to client task identities or the number of latent tasks, the server aims to adaptively determine which clients share each expert. The goal is to enable knowledge sharing among clients with similar task characteristics while reducing negative transfer across tasks.

3.3 Observations

We empirically examine whether LoRA updates retain stable, task-discriminative signals under intra-task non-IID data and whether assigning heterogeneous clients to different experts naturally yields specialization.

Observation 1: Cross-client centering reveals clearer task structure in query-A.

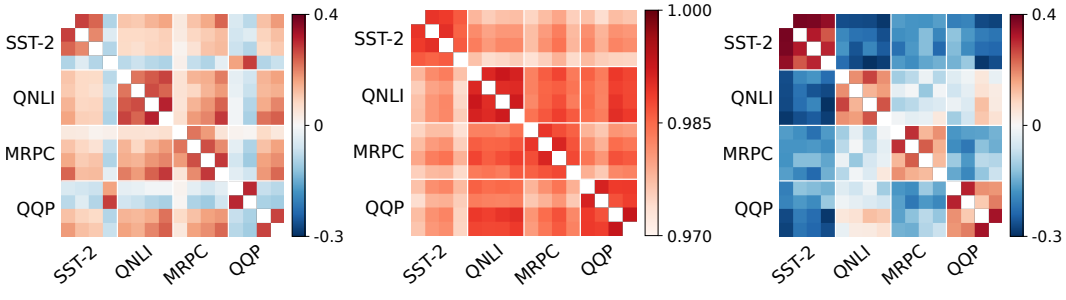
We consider a federated setting with four downstream tasks, each involving four clients, and introduce intra-task label distribution skew. After five rounds of local warmup, we compare the client similarity matrices constructed from B, raw query-A, and centered query-A.

As shown in Figure 2, the similarity matrix of B exhibits weak task-wise clustering. Raw query-A exhibits high cross-task similarity, with an average cosine similarity exceeding 0.98, making the underlying task structure difficult to distinguish. In contrast, centered query-A exhibits a clearer block-diagonal structure, with within-task similarities generally higher than cross-task similarities. Centering therefore highlights task-specific differences that the common component of the raw factors obscures. The corresponding quantitative AUC results are reported in Table A.2.

Let $A_i^{(q)}$ denote the LoRA A factor of the query projection layer for client i after warmup. We omit the layer index here and define the centered factor as

$$\widetilde{A}_i^{(q)} = A_i^{(q)} - \frac{1}{N} \sum_{j=1}^{N} A_j^{(q)}. \tag{3}$$

Under the shared initialization used during warmup, centering removes the common component and exposes each client's deviation from the mean adaptation. Appendix A.1.3 gives the derivation and its geometric interpretation. These results demonstrate that, under the examined label-skew setting, centered query-A distinguishes clients from different tasks more effectively than raw query-A, providing a basis for subsequent client clustering.



(a) All B (b) Query-A (c) Centered query-A

Figure 2: Client cosine similarity matrices.

Observation 2: Task-aligned client grouping does not guarantee subspace differentiation.

In the four-task setting, clients are grouped by their true tasks and jointly train one LoRA expert per group without subspace constraints. This isolates whether task-aligned grouping yields distinct adaptation subspaces, independently of grouping errors.

For expert k, the row space $\mathrm{row}(A_k)$ represents its input adaptation subspace. We quantify overlap by the mean squared cosine of the principal angles between two such subspaces; the formal definition is given in Eq. (10). A value of 0 indicates orthogonal subspaces, whereas 1 indicates identical subspaces. Under task-aligned grouping without subspace constraints, the mean input subspace overlap between experts reaches 0.8643. This indicates that experts for different tasks retain highly similar input adaptation directions, as illustrated in Figure 3. This result shows that client grouping determines which data jointly update an expert, but does not directly constrain the input directions read by different experts. This observation motivates a soft regularizer that penalizes input subspace overlap during expert training, balancing task fitting and differentiation of adaptation directions. We further analyze the effects of this constraint on expert subspace overlap and downstream performance in Section 5.2.2.

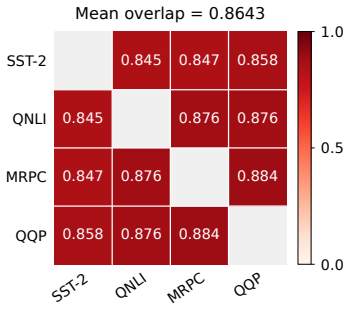


Figure 3: Input-subspace overlap between experts.

4 METHOD

In this section, we present FedEDS, which addresses simultaneous task heterogeneity and intra-task non-IID data through collaboration-aware expert construction and subspace disentanglement, as illustrated in Figure 4.

4.1 ADAPTIVE CLIENT CLUSTERING

Client representations from centered query-A. Starting from a shared LoRA initialization, all clients independently perform T_w warmup rounds, obtaining local adapters of rank r_w with scaling factor s_w. Let $A_{i,\ell}^{(q)} \in \mathbb{R}^{r_w \times d_{q,\ell}}$ denote the query-projection LoRA A factor of client i at layer ℓ. As query updates shape attention scores, we center each query-A factor across clients using Eq. (3), then vectorize and normalize it:

$$a_{i,\ell} = \frac{\mathrm{vec}(\widetilde{A}_{i,\ell}^{(q)})}{\max(\|\widetilde{A}_{i,\ell}^{(q)}\|_F, \epsilon)}. \tag{4}$$

Here, $\mathrm{vec}(\cdot)$ denotes vectorization, $\|\cdot\|_F$ denotes the Frobenius norm, and $\epsilon > 0$ is a numerical stability constant. The resulting $a_{i,\ell} \in \mathbb{R}^{r_w d_{q,\ell}}$ represents client i at layer ℓ, with normalization preventing residual magnitudes from dominating client relationships. We then define the client distance across all L query layers as

$$D_{ij} = \frac{1}{L}\sum_{\ell=1}^{L}\left(1 - a_{i,\ell}^{\top} a_{j,\ell}\right), \quad i \neq j, \qquad D_{ii} = 0. \tag{5}$$

Thus, D_{ij} measures the directional differences between the two clients' query-A deviations from the cross-client mean.

Adaptive Expert Clustering. To adaptively determine the number of experts K, we convert distances into similarities and apply affinity propagation (AP) clustering (Frey & Dueck, 2007):

$$S_{ij} = -D_{ij} \quad (i \neq j), \qquad S_{ii} = \rho, \qquad \rho = \mathrm{median}_{u \neq v}(-D_{uv}). \tag{6}$$

Here, $S \in \mathbb{R}^{N \times N}$ is the AP similarity matrix, and ρ is the common exemplar preference. The indices u and v range over distinct clients. Larger ρ values favor more exemplars. With the median preference, AP returns the client clusters $\{\mathcal{C}_g\}_{g=1}^{K}$ and their number K without a separate silhouette-based search over candidate cluster counts. This procedure neither uses the true task identities z_i nor requires a predefined number of latent tasks. The resulting $\{\mathcal{C}_g\}$, K and assignments $c_i = \gamma(i)$ serve as inputs to expert construction in the next subsection and remain fixed during subsequent training.

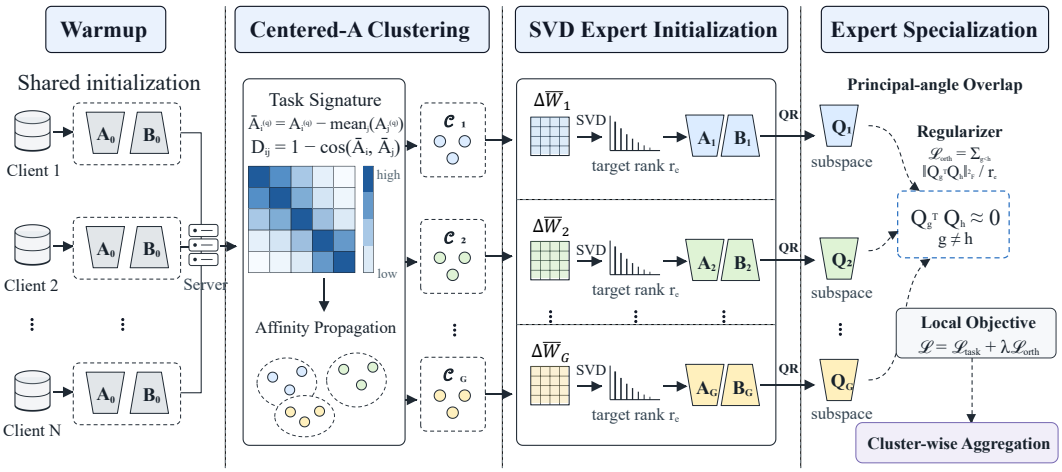


Figure 4: Overview of FedEDS. After warmup, clients are clustered by centered query-A and assigned SVD-initialized experts; experts are then trained with the overlap regularizer in every round.

4.2 Constructing Experts from Grouped Updates

Using the clusters identified by centered query-A, we construct each shared LoRA expert by aggregating its clients' effective weight updates from warmup. For a LoRA update $\Delta W = BA$, averaging the two factors separately is generally not equivalent to averaging their effective updates (Wang et al., 2024; Bian et al., 2025):

$$\left(\frac{1}{n_g}\sum_{i\in\mathcal{C}_g} B_i\right)\left(\frac{1}{n_g}\sum_{i\in\mathcal{C}_g} A_i\right) \neq \frac{1}{n_g}\sum_{i\in\mathcal{C}_g} B_i A_i, \tag{7}$$

where $n_g = |\mathcal{C}_g|$ is the size of client cluster g. For each LoRA-adapted module, we first compute the mean effective weight update within each cluster and then compress it to the expert rank r_e:

$$M_g = \frac{1}{n_g}\sum_{i\in\mathcal{C}_g} s_w B_i A_i, \qquad \widehat{M}_g \in \arg\min_{\mathrm{rank}(X)\le r_e} \|M_g - X\|_F^2, \tag{8}$$

where M_g denotes the mean effective weight update of group g and $\widehat{M}_g$ is its optimal low-rank approximation (Eckart & Young, 1936). We then retain the leading r_e singular components through truncated SVD (Bai et al., 2024) to initialize the group-shared LoRA expert:

$$\widehat{M}_g = U_{r_e}\Sigma_{r_e}V_{r_e}^\top, \qquad B_g^{(0)} = \frac{1}{\sqrt{s_e}}U_{r_e}\Sigma_{r_e}^{1/2}, \qquad A_g^{(0)} = \frac{1}{\sqrt{s_e}}\Sigma_{r_e}^{1/2}V_{r_e}^\top, \tag{9}$$

where s_e is the LoRA scaling factor during expert training. This converts the client relationships discovered in Section 4.1 into a bank of K compact LoRA experts $\mathcal{E}$, completing the warmup stage.

4.3 Encouraging Expert Specialization through Subspace Separation

Based on the observations in Section 3.3, we regularize only the input subspaces, where the overlap is concentrated, avoiding the additional QR computations required for output-side regularization.

For expert g in module m, the row space of $A_{g,m}$ characterizes the input directions that the expert can read. We obtain an orthonormal basis $Q_{g,m}$ via the reduced QR decomposition $A_{g,m}^\top = Q_{g,m}R_{g,m}$, where $Q_{g,m}^\top Q_{g,m} = I_{r_e}$, and use $Q_{g,m}$ as an orthonormal basis for this input subspace. The normalized overlap between experts g and k in module m is defined as

$$\Omega_m(g,k) = \frac{\|Q_{g,m}^\top Q_{k,m}\|_F^2}{r_e} = \frac{1}{r_e}\sum_{j=1}^{r_e}\cos^2\theta_{g,k,m,j}, \tag{10}$$

where $\theta_{g,k,m,j}$ are the principal angles between the two subspaces (Björck & Golub, 1973). Unlike constraints on raw factors, this measure captures the input directions spanned by the experts and is insensitive to factor scaling. When the reduced QR decomposition returns fewer than r_e columns because $A_{g,m}$ is numerically rank deficient, we treat the spanned subspace as the column space of the returned basis rather than padding it to r_e dimensions.

Let $\mathcal{L}_i^{\text{task}}$ be the mean task loss on $\mathcal{D}_i$ of the model that combines expert updates through routing. The local training objective is

$$\mathcal{J}_i = \mathcal{L}_i^{\text{task}} + \lambda \mathcal{R}_i, \qquad \mathcal{R}_i = \frac{1}{|\mathcal{M}|\binom{K}{2}} \sum_{m \in \mathcal{M}} \sum_{1 \le g < k \le K} \Omega_{i,m}(g,k), \tag{11}$$

where $\lambda \ge 0$ trades off task fitting and subspace separation, and $\mathcal{R}_i = 0$ when $K = 1$. Each client updates its assigned expert E_{c_i} and router parameters while keeping the other experts fixed at the server's broadcast copies, so $\mathcal{R}_i$ provides gradients to E_{c_i} only through pairs containing c_i. As a soft constraint, it encourages subspace differentiation without requiring exact orthogonality.

4.4 FedEDS Workflow

FedEDS consists of two stages: expert initialization and cluster-wise federated training.

Expert Initialization. Clients first perform T_w local warmup rounds from a shared LoRA initialization and upload their adapters. The server clusters clients using centered query-A signatures (Section 4.1) and initializes one rank-r_e expert per group through weight-update aggregation and truncated SVD (Section 4.2). The client clustering remains fixed during subsequent training.

Cluster-Wise Federated Training. At each communication round, the server sends the complete expert bank and the corresponding group router to each client. Following the adaptive routing scheme of FedLEASE (Wang et al., 2025), clients combine expert outputs according to the input, enabling access to knowledge across groups. Client i minimizes Eq. (11) by updating its assigned expert E_{c_i} and router, while keeping the backbone and other experts frozen. The subspace regularizer thus encourages the updated expert to learn input directions distinct from those of the other experts. Clients upload only their updated expert and router parameters; theoretical communication costs are discussed in Appendix A.5.3. With full client participation, the server uniformly averages the A factors, B factors, and router parameters within each group to obtain the models for the next round. Appendix Algorithm A.1 summarizes the procedure.

5 Experiments

We evaluate FedEDS on natural language understanding (NLU) tasks using RoBERTa-Large (Liu et al., 2019) as the pretrained backbone. The main experiments are repeated with three random seeds, $\{37, 42, 64\}$. Software and hardware details are provided in Appendix A.3.

5.1 Main Results on Natural Language Understanding

Task Setup. We use four GLUE (Wang et al., 2018) classification tasks, SST-2, QNLI, MRPC, and QQP, with 16 clients, four per task; each client holds data from only one task. Within each task, we partition the training data using a Dirichlet distribution $p_k \sim \text{Dir}(\alpha)$ (Yurochkin et al., 2019), with $\alpha = 0.5$ in the main experiments (smaller α induces stronger label skew). All clients participate in every round, and each client's validation set follows its local label distribution. We report per-task accuracy and the arithmetic mean across the four tasks.

Baselines. We compare with eight federated LoRA methods: **FedIT** (Zhang et al., 2024) (FedAvg (McMahan et al., 2017) on LoRA parameters), **FFA-LoRA** (Sun et al., 2024) (frozen A, aggregated B), **IFCA+LoRA** (Ghosh et al., 2020; Hu et al., 2022) (clustered FL with per-cluster adapters), **FedSA** (Guo et al., 2025) (trains both factors, aggregates only A), **FedLEASE** (Wang et al., 2025) (multiple experts from warmup LoRA relationships), **FedDPA-T** (Yang et al., 2024) (global and local adapters), **FedALT** (Bian et al., 2026a) (global sharing with local adaptation), and

Table 1: Main results under task heterogeneity and intra-task label skew ($\alpha = 0.5$), mean accuracy (%) $\pm$ standard deviation over three seeds. Δ denotes the average accuracy gain over FedIT (pp). Best and second-best results are **bold** and underlined, respectively.

Methods	SST-2	QNLI	MRPC	QQP	Avg.	Δ
FedIT	$88.53_{\pm 2.11}$	$68.09_{\pm 8.94}$	$79.57_{\pm 3.65}$	$76.26_{\pm 7.88}$	$78.11_{\pm 2.29}$	0.00
FFA-LoRA	$58.58_{\pm 4.02}$	$61.89_{\pm 14.06}$	$67.12_{\pm 5.92}$	$69.33_{\pm 9.86}$	$64.23_{\pm 3.61}$	-13.88
IFCA+LoRA	$90.02_{\pm 3.36}$	$80.61_{\pm 2.03}$	$80.10_{\pm 3.37}$	$84.50_{\pm 1.81}$	$83.81_{\pm 1.41}$	$+5.70$
FedSA	$94.40_{\pm 2.47}$	$83.50_{\pm 0.95}$	$79.35_{\pm 4.99}$	$82.59_{\pm 2.78}$	$84.96_{\pm 1.86}$	$+6.85$
FedLEASE	$92.42_{\pm 0.94}$	$83.59_{\pm 3.79}$	$80.49_{\pm 3.09}$	$83.30_{\pm 4.71}$	$84.95_{\pm 2.34}$	$+6.84$
FedDPA-T	$\mathbf{95.03_{\pm 2.31}}$	$\underline{84.73_{\pm 2.17}}$	$80.93_{\pm 1.61}$	$\underline{86.83_{\pm 2.10}}$	$86.88_{\pm 1.09}$	$+8.77$
FedALT	$94.56_{\pm 2.17}$	$83.34_{\pm 1.53}$	$83.03_{\pm 2.91}$	$\mathbf{87.53_{\pm 2.26}}$	$\underline{87.12_{\pm 0.68}}$	$\underline{+9.01}$
FedTreeLoRA	$\underline{94.98_{\pm 1.72}}$	$84.11_{\pm 3.70}$	$\underline{85.19_{\pm 3.48}}$	$83.36_{\pm 6.33}$	$86.91_{\pm 1.73}$	$+8.80$
FedEDS (Ours)	$94.43_{\pm 1.88}$	$\mathbf{86.38_{\pm 3.87}}$	$\mathbf{87.13_{\pm 0.24}}$	$84.58_{\pm 3.19}$	$\mathbf{88.13_{\pm 1.12}}$	$\mathbf{+10.02}$

FedTreeLoRA (Bian et al., 2026b) (hierarchical expert sharing). All methods share the backbone, data partitions, LoRA injection positions, participation rate, local epochs, communication rounds, and random seeds. We use official code where available and the original method-specific hyperparameters, with the learning rate set to 1×10^{-4} for all methods.

Results. As shown in Table 1, FedEDS achieves the highest average accuracy (88.13%), exceeding FedIT by 10.02 percentage points (pp) and the strongest baseline by 1.01 pp. It is best on QNLI (86.38%) and MRPC (87.13%), surpassing the strongest baselines on these tasks (FedDPA-T and FedTreeLoRA) by 1.65 and 1.94 pp, with the lowest standard deviation on MRPC (0.24 pp). FedEDS therefore combines competitive task-level accuracy with knowledge sharing across heterogeneous clients.

5.2 In-depth Analyses

We further examine the effect of intra-task statistical heterogeneity on FedEDS and analyze its client clustering and expert subspace separation. Unless otherwise specified, settings follow Section 5.1.

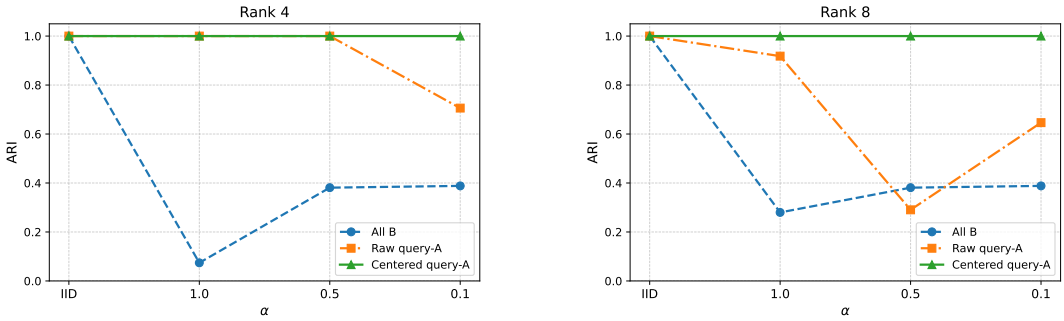
5.2.1 Analysis of Client Clustering

We test whether B, raw query-A, and centered query-A recover task groups under IID data and label skew with $\alpha \in \{1.0, 0.5, 0.1\}$, applying the same clustering procedure after five warmup rounds and measuring agreement with true tasks by the adjusted Rand index (ARI) (Hubert & Arabie, 1985).

As shown in Figure 5, centered query-A achieves perfect clustering ($\mathrm{ARI} = 1.0$) at all heterogeneity levels for both $r_w = 4$ and $r_w = 8$. In contrast, B is sensitive to statistical heterogeneity: although it recovers the correct task structure under IID data, its ARI drops sharply under label skew, falling to 0.074 for $\alpha = 1.0$ at rank 4 and remaining below 0.4 across all non-IID settings at both ranks. Raw query-A generally outperforms B but is less stable than centered query-A. At rank 4, it perfectly recovers the task groups for $\alpha = 1.0$ and 0.5 but drops to 0.706 for $\alpha = 0.1$; at rank 8, its ARI varies considerably, reaching 0.918, 0.290, and 0.646 for $\alpha = 1.0$, 0.5, and 0.1, respectively. Centering therefore recovers the latent client groups consistently across the evaluated label-skew settings, whereas the alternative signatures do not.

5.2.2 Analysis of Expert Specialization

We investigate whether LoRA experts learn distinct adaptation subspaces when clients are grouped according to their ground-truth tasks. We refer to overlap between the row spaces of the expert A factors as A-overlap and overlap between the column spaces of the expert B factors as B-overlap, both measured via Eq. (10). We compare no regularization (None), input-subspace (A-only), output-subspace (B-only), and joint (A+B) regularization. Warmup adapters and experts both use rank 4, and we report the best result for each configuration.



(a) Rank 4 (b) Rank 8

Figure 5: Clustering performance (ARI) under varying levels of statistical heterogeneity.

Table 2 shows that without regularization, the A-overlap reaches 0.8643 while the B-overlap is only 0.0401, confirming that true-task grouping does not yield sufficiently differentiated input subspaces. A-only regularization reduces A-overlap to 0.0061 and raises average accuracy from 88.28% to 88.69%, whereas B-only regularization improves accuracy by only 0.03 pp. Joint A+B regularization achieves the highest accuracy (89.13%, $+0.85$ pp) despite slightly higher A-overlap than A-only. Separation in both subspaces is therefore associated with the best accuracy, while minimizing overlap alone is not.

Table 2: Expert subspace overlap and average accuracy under intra-task IID data, using seed 42.

Reg.	A-overlap ↓	B-overlap ↓	Acc. (%) ↑
None	0.8643	0.0401	88.28
B-only	0.8532	0.0036	88.31
A-only	**0.0061**	0.0357	88.69
A+B	0.0405	**0.0035**	**89.13**

5.3 Ablation Study

We ablate two components of FedEDS. **w/o SVD** replaces effective-update aggregation and truncated SVD with separate averaging of the A and B factors within each group, and **w/o Reg.** removes the regularizer in Eq. (11). Both variants retain centered query-A clustering.

Table 3: Ablation study of FedEDS under intra-task label skew ($\alpha = 0.5$), using seed 42.

Variant	SST-2	QNLI	MRPC	QQP	Avg.
w/o Reg.	**94.20**	85.28	86.64	85.53	87.91
w/o SVD	93.95	86.25	85.38	**86.20**	87.94
FedEDS	93.43	**90.48**	**87.06**	84.78	**88.94**

As shown in Table 3, FedEDS achieves 88.94% average accuracy, outperforming w/o SVD and w/o Reg. by 1.00 and 1.03 pp, respectively. The gains concentrate on QNLI and MRPC, while both variants are slightly better on SST-2 and QQP, which indicates task-level trade-offs. Both SVD-based initialization and subspace separation therefore contribute to the average performance of FedEDS.

6 Conclusion

In this work, we reveal that when task heterogeneity coexists with intra-task label skew in federated fine-tuning, the query-A factors of LoRA, once centered across clients, carry task-discriminative information that can uncover latent client collaboration structures. Based on this observation, we introduce FedEDS, a federated multi-expert LoRA framework that handles both client clustering and expert specialization without requiring task labels or a predefined number of experts. FedEDS clusters clients by centered query-A signatures and builds group-specific experts through effective-update aggregation and truncated SVD. It further introduces an input subspace overlap regularizer that encourages experts to learn differentiated low-rank adaptation directions. Experiments show that FedEDS accurately recovers latent client groups, reduces the overlap between expert subspaces, and outperforms existing federated LoRA methods in average accuracy. Future work will extend FedEDS to more diverse downstream tasks and to settings with partial participation or changing client populations.

AI USE STATEMENT

Generative AI tools were used during the writing process, primarily to assist with literature searches and language polishing. All AI-assisted content was carefully reviewed, revised, and verified by the authors. The experimental design, code implementation, execution of experiments, and analysis and validation of all results were conducted by the authors. The authors take full responsibility for the final content of this paper.

ETHICS STATEMENT

This work studies federated fine-tuning of pretrained language models on public benchmark data and involves no human subjects. The data of each client are used only for local training. At the algorithm level, client-server communication consists of LoRA expert and router parameters; communication costs are discussed in Appendix A.5.3. We do not identify ethical concerns beyond those already associated with federated learning and parameter-efficient fine-tuning of public models.

REPRODUCIBILITY STATEMENT

We describe the experimental protocol in sufficient detail for reproduction. Section 5.1 specifies the four GLUE classification tasks, the 16-client partition with four clients per task, the Dirichlet label-skew parameter $\alpha = 0.5$, full client participation, and the learning rate 1×10^{-4}. The backbone, the number of warmup rounds, the LoRA scaling factors, the client signature computed from the query-projection layer, the affinity propagation preference, and the truncated SVD initialization are given in Section 4 and Section 5, and the warmup and expert ranks are stated in Section 5.2. Main results are averaged over the three random seeds $\{37, 42, 64\}$ and reported with standard deviations in Table 1. The clustering and subspace overlap analyses in Section 5.2 follow the same protocol unless stated otherwise, and the component ablation is reported in Table 3. The supplementary material provides the pseudocode of the complete workflow in Algorithm A.1.

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, pp. 14457–14483. Curran Associates, Inc., 2024. doi: 10.52202/079017-0461.

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*, pp. 3737–3746, 2025.

Jieming Bian, Lei Wang, Letian Zhang, and Jie Xu. Fedalt: Federated fine-tuning through adaptive local training with rest-of-world lora. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(24):19728–19736, March 2026a. ISSN 2374-3468. doi: 10.1609/aaai.v40i24.39054.

Jieming Bian, Lei Wang, Letian Zhang, and Jie Xu. Fedtreelora: Reconciling statistical and functional heterogeneity in federated lora fine-tuning. In *Forty-Third International Conference on Machine Learning*, June 2026b.

Åke Björck and Gene H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123):579–594, 1973. doi: 10.1090/S0025-5718-1973-0348991-3.

Tom Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

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

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 2089–2099. PMLR, July 2021.

Donghang Duan, Xu Zheng, Lizong Zhang, Chong Mu, and Meng Han. Fedweave: Rethinking the unit of specialization in heterogeneous federated moe-lora, 2026. URL https://arxiv.org/abs/2607.26618.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, September 1936. ISSN 0033-3123, 1860-0980. doi: 10.1007/BF02288367.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3557–3568. Curran Associates, Inc., 2020.

Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, February 2007. doi: 10.1126/science.1136800.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19586–19597. Curran Associates, Inc., 2020.

Pengxin Guo, Shuang Zeng, Yanran Wang, Huijie Fan, Feifei Wang, and Liangqiong Qu. Selective aggregation for low-rank adaptation in federated learning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 99003–99027, 2025.

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

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, December 1985. ISSN 1432-1343. doi: 10.1007/BF01908075.

Peter Kairouz and H. Brendan McMahan. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1-2):1–210, June 2021. ISSN 1935-8237. doi: 10.1561/2200000083.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 5132–5143. PMLR, November 2020.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 6357–6368. PMLR, July 2021.

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, 2019. URL https://arxiv.org/abs/1907.11692.

Zefang Liu and Jiahua Luo. Adamole: Fine-tuning large language models with adaptive mixture of low-rank adaptation experts. In *First Conference on Language Modeling*, August 2024.

Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. In *Advances in Neural Information Processing Systems*, volume 34, pp. 15434–15447. Curran Associates, Inc., 2021.

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

Siqi Ping, Yuzhu Mao, Yang Liu, Xiao-Ping Zhang, and Wenbo Ding. FL-TAC: Enhanced fine-tuning in federated learning via low-rank, task-specific adapter clustering, 2024. URL https://arxiv.org/abs/2404.15384.

Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3710–3722, August 2021. ISSN 2162-2388. doi: 10.1109/TNNLS.2020.3015958.

Raghav Singhal, Kaustubh Ponkshe, and Praneeth Vepakomma. Fedex-lora: Exact aggregation for federated and efficient fine-tuning of large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1316–1336, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.67.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving lora in privacy-preserving federated learning. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 17978–17994, 2024.

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, pp. 9565–9584. Curran Associates, Inc., 2024. doi: 10.52202/079017-0304.

Saeed Vahidian, Mahdi Morafah, Weijia Wang, Vyacheslav Kungurtsev, Chen Chen, Mubarak Shah, and Bill Lin. Efficient distribution similarity identification in clustered federated learning via principal angles between client data subspaces. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):10043–10052, June 2023. ISSN 2374-3468. doi: 10.1609/aaai.v37i8.26197.

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 Tal Linzen, Grzegorz Chrupała, and Afra Alishahi (eds.), *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446.

Lei Wang, Jieming Bian, Letian Zhang, and Jie Xu. Adaptive lora experts allocation and selection for federated fine-tuning. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 76018–76045. Curran Associates, Inc., 2025. doi: 10.52202/085713-2553.

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, pp. 22513–22533. Curran Associates, Inc., 2024. doi: 10.52202/079017-0708.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of lora experts. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 47302–47318, 2024.

Yaming Yang, Dilxat Muhtar, Yelong Shen, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Weiwei Deng, Feng Sun, Qi Zhang, Weizhu Chen, and Yunhai Tong. Mtl-lora: Low-rank adaptation for multi-task learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39 (20):22010–22018, April 2025. ISSN 2374-3468. doi: 10.1609/aaai.v39i20.35509.

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, pp. 39409–39433. Curran Associates, Inc., 2024. doi: 10.52202/079017-1245.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 7252–7261. PMLR, May 2019.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6915–6919, April 2024. doi: 10.1109/ICASSP48485.2024.10447454.

Jie Zhang, Zhiqi Li, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, and Chao Wu. Federated learning with label distribution skew via logits calibration. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 26311–26329. PMLR, June 2022.

Yicheng Zhang, Zhen Qin, Zhaomin Wu, Jian Hou, and Shuiguang Deng. Personalized federated fine-tuning for LLMs via data-driven heterogeneous model architectures. In *Proceedings of the ACM Web Conference 2026*, WWW '26, pp. 5099–5110, New York, NY, USA, April 2026. Association for Computing Machinery. ISBN 979-8-4007-2307-0. doi: 10.1145/3774904.3792147.

A Appendix

A.1 LoRA Asymmetric Initialization and Centered Query-A

This section explains why centered query-A captures relative client adaptation. We first analyze the early dynamics of a LoRA module under shared asymmetric initialization, then characterize the effect of cross-client centering. Unless stated otherwise, notation follows the main paper.

A.1.1 Asymmetric Initialization and the First Two Updates

For one LoRA module, write its effective weight as

$$W_t = W^0 + sB_tA_t, \qquad A_t \in \mathbb{R}^{r\times d_{\text{in}}}, \quad B_t \in \mathbb{R}^{d_{\text{out}}\times r}, \quad s > 0. \tag{A.1}$$

For this local dynamics analysis, consider synchronous SGD without momentum, weight decay, or additional factor regularization. Both factors use the same constant learning rate η; the initialization satisfies $B_0 = 0$, and all clients share A_0. Let $G_t = \partial\mathcal{L}_t/\partial W_t$, evaluated at the model state immediately before local step t.

Proposition A.1 (early factor displacement). Under these conditions,

$$\nabla_B\mathcal{L}_t = sG_tA_t^\top, \qquad \nabla_A\mathcal{L}_t = sB_t^\top G_t, \tag{A.2}$$

and the first two steps satisfy

$$A_1 = A_0, \qquad B_1 = -\eta sG_0A_0^\top, \qquad A_2 - A_0 = \eta^2 s^2 A_0G_0^\top G_1. \tag{A.3}$$

Proof. For arbitrary differentials $\mathrm{d}A$ and $\mathrm{d}B$, $\mathrm{d}W = s(\mathrm{d}B\,A + B\,\mathrm{d}A)$. Substituting this into $\mathrm{d}\mathcal{L} = \langle G, \mathrm{d}W\rangle_F$ and using the cyclic property of the trace gives equation A.2. The SGD updates are

$$B_{t+1} = B_t - \eta sG_tA_t^\top, \qquad A_{t+1} = A_t - \eta sB_t^\top G_t. \tag{A.4}$$

Substitution of $B_0 = 0$ yields $A_1 = A_0$ and $B_1 = -\eta sG_0A_0^\top$; inserting B_1 into the next update of A gives equation A.3. □

Thus B responds to the local gradient on the first step, whereas the task-gradient-driven displacement of A begins on the second step and contains the cross-step product $G_0^\top G_1$. Note that this identity holds without assuming $G_0 = G_1$; approximating $G_0^\top G_1$ by $G^\top G$ would further require slowly varying gradients.

A.1.2 Finite-Step Expansion and Remainder

Proposition A.2 (finite-step expansion). For any finite number H of local steps, equation A.4 gives

$$B_t = -\eta s \sum_{u<t} G_u A_u^\top, \qquad A_H - A_0 = \eta^2 s^2 \sum_{0\le u<t<H} A_u G_u^\top G_t. \tag{A.5}$$

Consequently,

$$A_H - A_0 = \eta^2 s^2 A_0 \sum_{u<t<H} G_u^\top G_t + R_H, \tag{A.6}$$

where

$$R_H = \eta^2 s^2 \sum_{u<t<H} (A_u - A_0) G_u^\top G_t. \tag{A.7}$$

If $\|G_t\|_2 \le M$ along the realized optimization trajectory and $c = \eta s M$, then

$$\|R_H\|_2 \le \|A_0\|_2 (1+c)^H c^4 \binom{H}{4}, \tag{A.8}$$

with $\binom{H}{4} = 0$ for $H < 4$. For fixed H and a uniform gradient bound as $\eta \to 0$, the remainder is $O(\eta^4)$.

Proof. Summing the updates of B in equation A.4 gives the first identity in equation A.5. Substituting it into the updates of A and summing over t gives the second identity. Decomposing $A_u = A_0 + (A_u - A_0)$ then gives equation A.6–equation A.7. Let $a_t = \|A_t\|_2$ and $b_t = \|B_t\|_2$. Equation equation A.4 implies

$$a_{t+1} \le a_t + c b_t, \qquad b_{t+1} \le b_t + c a_t, \qquad a_t + b_t \le (1+c)^t \|A_0\|_2.$$

Using equation A.5 again yields

$$\|A_u - A_0\|_2 \le c^2 \binom{u}{2} \|A_0\|_2 (1+c)^H. \tag{A.9}$$

Substitute this bound into equation A.7 and use $\sum_{t=0}^{H-1} \sum_{u=0}^{t-1} \binom{u}{2} = \binom{H}{4}$ to obtain equation A.8. □

Although our experiments use AdamW, this SGD analysis provides intuition for why query-A displacement reflects client-specific gradient structure; we do not claim it holds exactly under adaptive optimizers.

A.1.3 Cross-Client Centering and Relative Adaptation Directions

In the shared factor coordinates used by the main method, write client i's query-A factor as $A_i = A_0 + \delta A_i$ and define $\bar{A} = N^{-1} \sum_i A_i$. Its centered residual is

$$\widetilde{A}_i = A_i - \bar{A} = \delta A_i - \frac{1}{N} \sum_{j=1}^{N} \delta A_j. \tag{A.10}$$

Centering exactly removes both the shared initialization and the mean adaptation component. It preserves pairwise Euclidean differences, since $\widetilde{A}_i - \widetilde{A}_j = A_i - A_j$. FedEDS changes the comparison by normalizing the centered residuals and comparing their directions rather than those of the raw factors.

Proposition A.3 (cosine expansion under a dominant common component). Let $a_0 \neq 0$, $u = a_0/\|a_0\|_2$, and $P_\perp = I - uu^\top$. Suppose $\max_i \|\delta a_i\|_2 \le \varepsilon \|a_0\|_2$. As $\varepsilon \to 0$,

$$1 - \cos(a_0 + \delta a_i, a_0 + \delta a_j) = \frac{\|P_\perp(\delta a_i - \delta a_j)\|_2^2}{2\|a_0\|_2^2} + O(\varepsilon^3). \tag{A.11}$$

Proof. The first-order expansion of $n(a) = a/\|a\|_2$ at a_0 is $n(a_0 + \delta a) = u + P_\perp \delta a/\|a_0\|_2 + O(\varepsilon^2)$. For unit vectors v_i, v_j, $1 - v_i^\top v_j = \|v_i - v_j\|_2^2/2$. Substituting the two first-order expansions proves equation A.11. □

With shared initialization, raw-factor cosine distances are compressed by the common component. Cross-client centering in equation 3, followed by residual normalization, exposes directions of client adaptation relative to the mean. This interpretation relies on the shared initialization during warmup. Since invertible reparameterizations of LoRA factors do not preserve single-factor distances, we verify the representation empirically in Appendix A.4.1.

A.2 Mathematical Properties of Expert Construction and Subspace Separation

A.2.1 Group-Averaged Effective Weight Updates

For a client group $\mathcal{C}_g$ from Section 4.2, omit the module index and let $n_g = |\mathcal{C}_g|$, $\bar{A}_g = n_g^{-1} \sum_{i \in \mathcal{C}_g} A_i$, and $\bar{B}_g = n_g^{-1} \sum_{i \in \mathcal{C}_g} B_i$. Direct expansion gives

$$\frac{1}{n_g} \sum_{i \in \mathcal{C}_g} B_i A_i - \bar{B}_g \bar{A}_g = \frac{1}{n_g} \sum_{i \in \mathcal{C}_g} (B_i - \bar{B}_g)(A_i - \bar{A}_g). \tag{A.12}$$

Thus averaging the two LoRA factors separately generally differs from averaging their effective updates; the difference is their joint variation. FedEDS therefore constructs the group mean in effective-update space:

$$M_g = \frac{1}{n_g} \sum_{i \in \mathcal{C}_g} s B_i A_i. \tag{A.13}$$

For any invertible $T_i \in \mathbb{R}^{r_w \times r_w}$, the transformation $A_i \mapsto T_i A_i$ and $B_i \mapsto B_i T_i^{-1}$ leaves $B_i A_i$ and hence M_g unchanged. Once the client groups are fixed, expert construction depends on their effective updates rather than a particular internal factor coordinate system.

A.2.2 Rank-Constrained Approximation and LoRA Scaling

Let $M_g = U \Sigma V^\top$ have nonincreasing singular values. Truncated SVD at expert rank r_e gives

$$\widehat{M}_g = U_{r_e} \Sigma_{r_e} V_{r_e}^\top \in \arg \min_{\mathrm{rank}(X) \le r_e} \|M_g - X\|_F^2, \tag{A.14}$$

with reconstruction error

$$\|M_g - \widehat{M}_g\|_F^2 = \sum_{j > r_e} \sigma_j^2. \tag{A.15}$$

For any rank-r_e matrix X, the projection onto its column space captures at most the energy of the leading r_e singular directions of M_g. The truncated SVD attains this bound.

Let $s_e > 0$ be the LoRA scaling factor in expert training. Balanced factors can be initialized as

$$B_g^{(0)} = s_e^{-1/2} U_{r_e} \Sigma_{r_e}^{1/2}, \qquad A_g^{(0)} = s_e^{-1/2} \Sigma_{r_e}^{1/2} V_{r_e}^\top, \tag{A.16}$$

so that $s_e B_g^{(0)} A_g^{(0)} = \widehat{M}_g$. If SVD is instead applied to the unscaled mean $n_g^{-1} \sum_i B_i A_i$, both recovered factors must additionally be multiplied by $\sqrt{s/s_e}$, where s is the warmup scaling factor. In all experiments, we set $s = s_e = 2$, so this multiplier equals one.

Truncated SVD maps each group mean to the target expert rank, including when a group mean has higher rank than its constituent updates. This construction occurs once at the warmup-to-expert transition; later rounds aggregate the expert A/B factors separately.

Round-wise factor aggregation. In subsequent training rounds, each client $i \in \mathcal{C}_g$ starts from the same broadcast rank-r_e expert (A_g, B_g). Write its returned factors as $A_i = A_g + \delta A_i$ and $B_i = B_g + \delta B_i$, and let $\overline{\delta A}_g = n_g^{-1} \sum_{i \in \mathcal{C}_g} \delta A_i$ and $\overline{\delta B}_g = n_g^{-1} \sum_{i \in \mathcal{C}_g} \delta B_i$. By equation A.12,

$$\begin{aligned} &\frac{s_e}{n_g} \sum_{i \in \mathcal{C}_g} B_i A_i - s_e \bar{B}_g \bar{A}_g \\ &\quad = \frac{s_e}{n_g} \sum_{i \in \mathcal{C}_g} (\delta B_i - \overline{\delta B}_g)(\delta A_i - \overline{\delta A}_g). \end{aligned} \tag{A.17}$$

Table A.1: FedEDS hyperparameters. The warmup scaling factor s in the appendix is denoted by s_w in the main paper.

Hyperparameter	Value
Learning rate	1×10^{-4}
Warmup rounds T_w	5
Warmup rank r_w	8 (4 and 8 in the representation diagnostic)
Expert rank r_e	4
Warmup LoRA scaling $s = s_w$	2 (all experiments)
Expert LoRA scaling s_e	2 (all experiments)
Main-experiment random seeds	$\{37, 42, 64\}$
Representation-diagnostic random seed	42
Local training	2 epochs; batch 32; max length 128
Regularization / dropout	$\lambda = 0.5$; LoRA dropout 0.05
Optimizer / weight decay	AdamW (`adamw_torch`); 0.05

The discrepancy is therefore bilinear in the within-round factor deviations and is second order when these deviations are small. Averaging the factors retains rank at most r_e, whereas the average of the effective updates can have rank up to $n_g r_e$ and would require rank-r_e recompression to maintain the expert rank, resulting in additional computational overhead. We use factor-wise averaging in subsequent rounds to retain the fixed expert rank without repeating SVD.

A.2.3 Geometry of Input and Output Subspaces

FedEDS regularizes the row space of expert A, which describes the input directions read by an expert. For the expert-specialization analysis in the main paper, we also define the output-side column space of B. In module m, let $Q_{A,g,m}$ be an orthonormal basis for the columns of $A_{g,m}^\top$ and $Q_{B,g,m}$ for the columns of $B_{g,m}$. When the corresponding factors have numerical rank r, define, for $X \in \{A, B\}$,

$$\Omega_{X,m}(g,k) = \frac{1}{r}\|Q_{X,g,m}^\top Q_{X,k,m}\|_F^2 = \frac{1}{r}\sum_{j=1}^{r} \cos^2 \theta_{X,g,k,m,j}. \tag{A.18}$$

For $X = A$, this is the input-subspace overlap in equation 10; for $X = B$, it measures overlap between output subspaces. Principal-angle properties give $0 \le \Omega_{X,m}(g,k) \le 1$: zero means orthogonal subspaces, and one means identical equal-rank subspaces.

With $P_g = Q_{X,g,m}Q_{X,g,m}^\top$ and $P_k = Q_{X,k,m}Q_{X,k,m}^\top$,

$$\Omega_{X,m}(g,k) = \frac{\mathrm{tr}(P_g P_k)}{r}, \qquad \|P_g - P_k\|_F^2 = 2r\big(1 - \Omega_{X,m}(g,k)\big). \tag{A.19}$$

Minimizing overlap therefore separates the projection subspaces. The metric depends on the subspaces, not the choice of orthonormal bases, and cannot be reduced by globally rescaling a LoRA factor. For $\Delta W = BA$, the row space of A selects input directions, while the column space of B maps the low-rank response into output directions. The main FedEDS method constrains the input side; the B-side quantity describes output geometry in the expert-specialization analysis.

A.3 Experimental Setup and Implementation Details

Software and hardware. Experiments were run with Python `3.10`, PyTorch `2.3.1+cu118`, Hugging Face Transformers `4.30`, PEFT `0.4`, Datasets `4.8`, and Accelerate `1.13` on four NVIDIA GeForce RTX 3090 GPUs. Table A.1 summarizes the training and grouping hyperparameters.

A.3.1 Client-Representation Diagnostic Setup

The client-representation diagnostic uses RoBERTa-Large and 16 clients. SST-2, QNLI, MRPC, and QQP each have four clients, indexed 0–3, 4–7, 8–11, and 12–15, respectively. We evaluate seed 42 at warmup ranks $r_w \in \{4, 8\}$ under IID data and within-task label skew $\alpha \in \{1.0, 0.5, 0.1\}$. We

Algorithm A.1: FedEDS training.

Input: N clients, warmup rounds T_w, total rounds T, ranks r_w, r_e, LoRA scales s_w, s_e, regularization weight λ.

Initialize all clients with the same frozen backbone and LoRA factors.
for $t = 1, \ldots, T_w$ **do**
  for each client i **do**
    Train its local rank-r_w adapter; send its LoRA factors to the server.
  end for
end for
for each query layer ℓ **do**
  Center, vectorize, and normalize the clients' query-A factors.
end for
Compute D_{ij} by equation 4–equation 5.
Run AP on $S = -D$ with the preference in equation 6; obtain $\{\mathcal{C}_g\}_{g=1}^{K}$ and c_i.
for each group g and adapted module m **do**
  Compute $M_{g,m} = |\mathcal{C}_g|^{-1} \sum_{i \in \mathcal{C}_g} s_w B_{i,m} A_{i,m}$; use rank-$r_e$ truncated SVD at scale s_e to initialize E_g as in equation 8–equation 9.
end for
for $t = T_w + 1, \ldots, T$ **do**
  Broadcast the complete expert bank and corresponding group router to each client.
  for each client i **do**
    Minimize equation 11 by updating E_{c_i} and the router while freezing the backbone and other experts; upload the updated expert and router.
  end for
  for each group g **do**
    Uniformly average uploaded expert A/B factors and router parameters to form the next-round group state.
  end for
end for

Output: Group expert bank, group routers, and fixed client assignments.

approximate the IID partition with $\alpha = 1000$. Each setting is evaluated using the client adapters obtained after the fifth warmup round. Client signatures use query-A from all 24 Transformer layers; the All B comparison uses query/value B factors from all 24 layers. True task identities are used only for offline evaluation and display, not for FedEDS grouping.

A.3.2 Centered Query-A Distance and AP Grouping

Client signatures and distances are computed in float64. Each centered residual is normalized as in Eq. (4) with $\epsilon = 10^{-12}$, so a zero residual maps to the zero vector. Per-layer normalized inner products are clipped to $[-1, 1]$ and averaged equally over query layers, as in Eq. (5). The resulting distance matrix is symmetrized explicitly, with its diagonal set to zero.

AP receives the precomputed similarity matrix $S = -D$ and the shared preference ρ defined in Eq. (6), with random seed 0. The number of experts K is determined by the exemplars returned by AP rather than specified in advance.

A.3.3 FedEDS Training Procedure

Algorithm A.1 summarizes the complete FedEDS training procedure.

A.4 Supplementary Client-Representation Results

A.4.1 Task Discriminability of Client Representations under Label Skew

We compare All B, raw query-A, and centered query-A at the fifth-round warmup checkpoint in each of the eight settings. All B includes the LoRA B factors from every query/value branch, as

Table A.2: Mean within-task / cross-task cosine similarity and same-task pair AUC after five warmup rounds (seed 42). IID is approximated with $\alpha = 1000$.

r_w	Partition	All B		Raw query-A		Centered query-A	
		Within / cross	AUC	Within / cross	AUC	Within / cross	AUC
4	IID	$0.1509/0.0435$	0.9805	$0.9902/0.9840$	0.9401	$0.1984/-0.1252$	**1.0000**
4	$\alpha = 1.0$	$0.0797/0.0295$	0.5781	$0.9889/0.9818$	0.9414	$0.1998/-0.1284$	**0.9970**
4	$\alpha = 0.5$	$0.1417/0.0682$	0.6766	$0.9891/0.9813$	0.9436	$0.2337/-0.1349$	**0.9991**
4	$\alpha = 0.1$	$0.3447/0.0784$	0.8003	$0.9888/0.9750$	0.9601	$0.3300/-0.1588$	**0.9961**
8	IID	$0.1530/0.0355$	0.9913	$0.9906/0.9852$	0.9497	$0.1639/-0.1196$	**0.9974**
8	$\alpha = 1.0$	$0.0955/0.0251$	0.7027	$0.9898/0.9851$	0.8685	$0.1525/-0.1162$	**0.9961**
8	$\alpha = 0.5$	$0.1164/0.0500$	0.7057	$0.9897/0.9846$	0.8924	$0.1673/-0.1193$	**0.9900**
8	$\alpha = 0.1$	$0.2685/0.0620$	0.7917	$0.9914/0.9843$	0.9310	$0.2388/-0.1389$	**0.9996**

in the main-text client-similarity figure. Both query-A variants use only query-projection A factors. For each signature, client cosine similarities are computed within layers and averaged equally over the corresponding layers.

True task labels enter only the offline diagnostic. We report mean cosine similarity for same-task and cross-task client pairs, together with pairwise AUC, which measures how often a same-task pair receives a higher similarity than a cross-task pair independently of the clustering algorithm used by FedEDS.

Raw query-A has high cosine similarity for both same-task and cross-task pairs, reflecting the shared component in the uncentered factors. Centering separates these pairs more clearly and achieves an AUC of at least 0.9900 in every setting. The 16 clients form $\binom{16}{2} = 120$ unordered pairs, including 24 same-task and 96 cross-task pairs. All eight diagnostics use the fifth-round snapshot of their respective seed-42 runs.

A.5 Computation, Storage, and Communication Costs

A.5.1 One-Time Server-Side Operations

Let N be the number of clients, L the number of query layers used for grouping, r_w the warmup rank, $d_{q,\ell}$ the input dimension of query projection ℓ, and I_{AP} the number of AP iterations. Constructing centered query-A signatures, normalizing them, and computing all pairwise client distances takes

$$O\left(N^2 \sum_{\ell=1}^{L} r_w d_{q,\ell}\right) \tag{A.20}$$

time. This bound includes the lower-order linear work for client means and normalization. AP message passing takes $O(I_{\mathrm{AP}} N^2)$ time and $O(N^2)$ storage for responsibilities and availabilities. Retaining all centered client signatures takes $O\left(N \sum_{\ell=1}^{L} r_w d_{q,\ell}\right)$ additional storage. Group discovery occurs once after warmup, not at every subsequent communication round.

For expert construction, let $\mathcal{M}$ be the set of adapted LoRA modules. Explicitly forming each client's $s_w B_{i,m} A_{i,m}$ and averaging within groups takes

$$O\left(N \sum_{m \in \mathcal{M}} d_{\mathrm{out},m} d_{\mathrm{in},m} r_w\right). \tag{A.21}$$

Applying a full dense SVD to each group mean to construct K experts takes

$$O\left(K \sum_{m \in \mathcal{M}} d_{\mathrm{out},m} d_{\mathrm{in},m} \min(d_{\mathrm{out},m}, d_{\mathrm{in},m})\right). \tag{A.22}$$

Since this step is performed only once after warmup, we use a full dense SVD; truncated solvers could further reduce this cost.

Table A.3: Query/value LoRA parameter counts for RoBERTa-Large. FP32 size counts LoRA A/B parameters only; $1\,\mathrm{MiB} = 2^{20}$ bytes.

Component	Parameters	FP32 size
One rank-8 LoRA adapter	786,432	3 MiB
One rank-4 LoRA expert	393,216	1.5 MiB
Bank of four rank-4 experts	1,572,864	6 MiB

A.5.2 Local Geometric Cost of Subspace Separation

For one LoRA module, reduced QR bases for K rank-r_e input subspaces cost $O(Kd_{\mathrm{in}}r_e^2)$. In FedEDS, a client compares its assigned expert with the other $K-1$ frozen experts. Across the adapted modules, the additional geometric operations cost

$$O\left(\sum_{m\in\mathcal{M}}\left[(2K-1)d_{\mathrm{in},m}r_e^2 + Kr_e^3\right]\right). \tag{A.23}$$

Storing the orthonormal bases takes

$$O\left(Kr_e\sum_{m\in\mathcal{M}} d_{\mathrm{in},m}\right).$$

These bounds exclude the cost of forward and backward passes.

A.5.3 LoRA Expert Capacity and Communication Accounting

For the adapted-module set $\mathcal{M}$, one rank-r LoRA expert contains

$$P(r) = \sum_{m\in\mathcal{M}} r\left(d_{\mathrm{in},m} + d_{\mathrm{out},m}\right) \tag{A.24}$$

A/B parameters. A client training its assigned expert has trainable LoRA-expert capacity $P(r_e)$; a server holding the complete K-expert bank stores $KP(r_e)$ such parameters. These counts exclude the router, task head, and frozen backbone. RoBERTa-Large has 48 query/value LoRA modules with $d_{\mathrm{in}} = d_{\mathrm{out}} = 1024$ in this configuration; Table A.3 gives the resulting sizes.

Each client uploads only its assigned expert and router, while the server provides the complete expert bank and corresponding routing state. Let P_ϕ be the router parameter count and b the bytes per transmitted parameter. The per-client, per-round upload and download payloads are

$$C_\uparrow = b\left[P(r_e) + P_\phi\right], \qquad C_\downarrow = b\left[KP(r_e) + P_\phi\right]. \tag{A.25}$$

Reducing the expert rank from eight to four halves $P(r_e)$ and hence the LoRA-expert portion of the per-round upload.