FEDCSO-LORA: COORDINATED SUBSPACE OPTIMIZATION FOR HETEROGENEOUS FEDERATED LORA

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-Rank Adaptation (LoRA) enables an efficient federated fine-tuning of large language models. In practice, however, clients with heterogeneous resources may have different rank budgets, constraining the dimensions of their active latent subspaces. Some heterogeneous-rank methods randomly select the active components of a shared LoRA adapter, which may miss the informative combinations of components that could be learned even with the same rank budget. Learning subspaces from the local gradients can capture these combinations, but independently selected client subspaces may yield poorly aligned updates when data are heterogeneous. The challenge is therefore to preserve locally the informative directions under different rank budgets, while coordinating the resulting client updates towards the global objective. To this end, we propose FedCSO-LoRA, a coordinated subspace optimization framework for heterogeneous federated LoRA. Given its rank budget, each client learns the combinations of shared adapter coordinates that maximize the retained energy of its estimated local factor gradients. The coordination step aligns the selected subspace with a global update reference, while preserving a prescribed fraction of the initially retained gradient energy. The server maps the client updates into the common factor coordinates, and filters their aggregate according to the subspace coverage. We provide a nonconvex convergence analysis under the heterogeneous rank budgets that characterizes the effects of local training and server aggregation. Experiments across multiple datasets and language models demonstrate the effectiveness of FedCSO-LoRA compared with the various baselines.

1 INTRODUCTION

Large language models (LLMs) provide a foundation for a wide range of language tasks, while the fine-tuning adapts them to some specific tasks and domains (Hu et al., 2022; Wang et al., 2024b). The data needed for adaptation can be distributed across the clients, which are unavailable for a centralized training. Federated learning (FL) enables these clients to fine-tune a shared model while keeping their raw data local (McMahan et al., 2017; Wang et al., 2024b). However, updating and communicating all the model parameters can be costly for clients with limited resources. Parameter-efficient fine-tuning methods are thus proposed to reduce the number of trainable parameters (Houlsby et al., 2019). Among them, Low-Rank Adaptation (LoRA) represents each weight update with two small trainable factors, while freezing the pretrained weights (Hu et al., 2022). In federated LoRA, clients train these factors on local data and send their updates to a server for the aggregation, thus reducing the number of parameters involved in the training and communication (Fang et al., 2026; Zhang et al., 2026).

A central difficulty is that clients may have different capacities in practice. Assigning the same LoRA rank to every client can exclude the weaker devices or underuse the stronger ones, motivating the federated adaptation methods that support heterogeneous local ranks (Cho et al., 2024; Wang et al., 2024b; Bai et al., 2024). In particular, FSLoRA and Fed-PLoRA let clients train the subsets of global components within their budgets (Fang et al., 2026; Zhang et al., 2026). However, selecting existing components restricts the active subspace to coordinate directions in the shared adapter, though a rank budget limits the number of active latent directions rather than their orientation. Learn-

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.



(a) Subspace selection (b) Client rank (c) Reference alignment

Figure 1: Motivation for client subspace selection and coordination. (a) Random coordinate selection activates two global coordinates, while FedCSO-LoRA combines six coordinates into two active directions. The transformation BQ is shown, with the corresponding transformation of A given by $Q^\top A$. (b) SST-2 accuracy of FedCSO-LoRA and a random-coordinate control, with four clients fixed at rank 32 and the other 16 clients using the rank shown. (c) MRPC and QNLI accuracy changes against cosine similarity with a global update reference. "Local only" marks the subspace selection without coordination, and the rightward arrow indicates an increasing coordination.

ing combinations of these coordinates expands the available subspaces without increasing the active rank. Local factor gradients provide a criterion for choosing these combinations, allowing clients to exploit correlations across these coordinates and potentially retain more gradient information within the same budget. Figure 1(a) illustrates the distinction between selecting existing coordinates and learning their combinations. Figure 1(b) shows on SST-2 from the GLUE benchmark (Wang et al., 2018) that the performance gain between our method and a random-coordinate control enlarges as the client rank budgets decrease. This suggests that our method makes a more effective use of limited training capacity when most clients have a small rank budget.

Preserving the local gradient information does not by itself ensure a progress on the global objective. Under non-IID data, independently selected client subspaces may produce conflicting updates. Related difficulties arise from client drift and unequal local work (Karimireddy et al., 2020; Wang et al., 2020), while the subspace selection determines which gradient information enters training in the first place. The challenge is therefore to coordinate the client-specific subspaces while limiting the loss of locally informative directions. Figure 1(c) uses MRPC and QNLI, also from GLUE, to illustrate this balance as coordination increases from local-only subspace selection. Stronger alignment alone does not necessarily improve the accuracy, motivating coordination that explicitly limits the loss of local gradient information. The selected subspaces also provide uneven coverage of shared latent directions under different rank budgets. Mapping local updates into common factor coordinates enables aggregation, but does not itself address this imbalance. The server therefore needs to account for collective subspace coverage and moderate contributions along insufficiently covered directions.

Motivated by these considerations, we propose FedCSO-LoRA for a coordinated subspace optimization under heterogeneous rank budgets. Our contributions can be summarized as follows.

- We propose FedCSO-LoRA, a federated LoRA framework that learns and coordinates client training subspaces under heterogeneous rank budgets. Rather than restricting client subspaces to existing LoRA coordinates, it learns their combinations guided by both local data and shared optimization information, without increasing the active parameter count.
- We develop a coordinated strategy for client subspace selection and server aggregation. A shared global update reference guides training directions while an explicit constraint limits local gradient information loss. The server uses collective subspace coverage to adjust each direction's influence during aggregation.
- We establish a nonconvex convergence guarantee for the global factor objective under heterogeneous rank budgets. Experiments on GLUE and commonsense reasoning tasks demonstrate improved performance under resource and data heterogeneity, while ablation studies validate the contributions of the key designs.

2 Related Work

Heterogeneous Federated Adaptation. Heterogeneous federated LoRA enables clients with unequal rank budgets to contribute to a shared model. HetLoRA combines heterogeneous ranks with local pruning and sparsity-aware aggregation (Cho et al., 2024). FLoRA aligns aggregation through factor stacking (Wang et al., 2024b), while FlexLoRA decomposes aggregated weight updates into rank-specific adapters (Bai et al., 2024). RAVAN uses low-rank heads with fixed outer factors and trainable cores and scales; clients select heads within their budgets using strategies including gradient-based scoring (Raje et al., 2025). FSLoRA selects global factor coordinates through structured sketching (Fang et al., 2026), while Fed-PLoRA selects rank-one modules and folds unselected modules into the frozen weight (Zhang et al., 2026). These selection-based methods choose predefined heads or components. FedCSO-LoRA learns combinations of shared latent coordinates and coordinates the resulting client subspaces.

Low-rank Spaces and Optimization. Theoretical work characterizes how rank constrains LoRA's expressive power (Zeng & Lee, 2024). AdaLoRA allocates rank by component importance (Zhang et al., 2023), while DoRA separates weight magnitude and direction (Liu et al., 2024). For subspace initialization, PiSSA uses principal weight components (Meng et al., 2024), EVA uses activation variance (Paischer et al., 2025), and LoRA-GA and LoRA-One use gradient information (Wang et al., 2024a; Zhang et al., 2025). GaLore projects gradients to reduce optimizer memory (Zhao et al., 2024), while LoRA-Pro corrects factor updates to better approximate full fine-tuning (Wang et al., 2025). Our focus is selecting and coordinating subspaces within an existing shared adapter under heterogeneous client budgets and data.

Federated Coordination and Aggregation. SCAFFOLD corrects client drift with control variates (Karimireddy et al., 2020), FedNova normalizes updates for unequal local work (Wang et al., 2020), and adaptive federated optimization introduces server momentum and adaptive scaling (Reddi et al., 2021). In federated LoRA, FFA-LoRA freezes one factor to avoid cross-factor aggregation mismatch (Sun et al., 2024), while FedSVD refreshes that factor through server-side decomposition (Lee et al., 2025). FedSA-LoRA adjusts which factors are shared (Guo et al., 2025), and RoLoRA alternates factor optimization (Chen et al., 2025). These approaches modify update correction, factor sharing, or optimization order. FedCSO-LoRA coordinates subspace selection and incorporates collective subspace coverage into aggregation.

3 Problem Formulation

We consider federated fine-tuning of a pretrained language model with a server and N clients (Sun et al., 2024; Wang et al., 2024b). Client i holds local data drawn from distribution $\mathcal{D}_i$. For an adapted layer with input dimension n and output dimension m, LoRA parameterizes the weight matrix as

$$W = W_0 + BA, \tag{1}$$

where $W_0 \in \mathbb{R}^{m\times n}$ is the frozen pretrained weight, $B \in \mathbb{R}^{m\times R}$ and $A \in \mathbb{R}^{R\times n}$ are trainable factors, and R is the global LoRA rank. When multiple layers are adapted, B and A denote the corresponding collections of LoRA factors. We minimize the federated objective

$$J(B, A) = \sum_{i=1}^{N} p_i f_i(B, A), \tag{2}$$

where $f_i(B, A) = \mathbb{E}_{\xi\sim\mathcal{D}_i}[\ell(W_0 + BA; \xi)]$ is the local expected loss, and the aggregation weights satisfy $p_i > 0$ and $\sum_{i=1}^{N} p_i = 1$. The client distributions $\{\mathcal{D}_i\}_{i=1}^{N}$ may differ.

Resource heterogeneity imposes a client-specific rank budget $k_i \le R$, limiting the dimension of the active subspace within the shared R-dimensional LoRA latent space (Bai et al., 2024; Fang et al., 2026; Zhang et al., 2026). This budget determines how many directions a client can train, while leaving their orientation to be chosen. Our goal is to select and coordinate these local training subspaces to minimize Eq. (2), while ensuring that each client starts every round from the complete global model. We consider full client participation. In each communication round, the server broadcasts the global factors, clients perform local updates with their selected subspaces held fixed, and the server aggregates the returned updates.

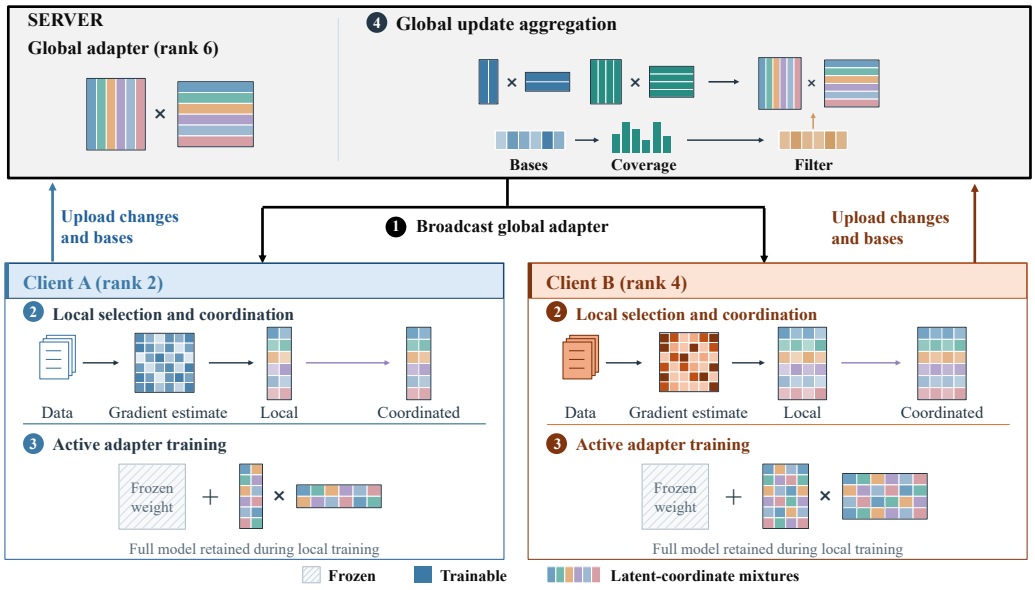


Figure 2: One training round with heterogeneous rank budgets. Each client estimates gradients locally, uses its cached previous global adapter to coordinate its selected directions, and trains only the active factors. Clients upload factor changes and their selected bases. The server lifts and aggregates these changes and applies its coverage filter.

4 Coordinated Subspace Optimization

As illustrated in Figure 2, FedCSO-LoRA learns combinations of shared LoRA coordinates to adapt each client's training subspace to its local gradients under its rank budget. The previous global update provides a common reference for coordinating these subspaces, while an energy constraint limits the loss of local gradient information. Aggregation further accounts for how the selected subspaces collectively cover the latent space, attenuating updates along weakly covered directions.

4.1 Local Subspace Selection

Under a rank budget k_i, FSLoRA and Fed-PLoRA select k_i existing rank-one LoRA components for local training (Fang et al., 2026; Zhang et al., 2026). Each component corresponds to one coordinate in the shared R-dimensional latent space. To choose training directions that better match local gradients, we extend this selection from individual coordinates to their linear combinations, as illustrated in Figure 1(a). We represent the resulting subspace by an orthonormal basis $Q_i \in \mathbb{R}^{R\times k_i}$ with $Q_i^\top Q_i = I$. The active factors BQ_i and $Q_i^\top A$ still contain $k_i(m+n)$ trainable parameters, so this flexibility does not increase the local training budget. Selecting columns of the identity matrix as Q_i recovers coordinate selection.

We choose the local subspace according to how much gradient information it retains at the received global model. Let $f_i(B, A; \mathcal{B})$ denote the average task loss on a local batch $\mathcal{B}$. When the preferred basis is initialized or refreshed, client i uses a batch $\mathcal{B}_i^{\text{init}}$ to compute

$$\widetilde{G}_{B,i} = \nabla_B f_i(B, A; \mathcal{B}_i^{\text{init}}),\ \widetilde{G}_{A,i} = \nabla_A f_i(B, A; \mathcal{B}_i^{\text{init}}). \tag{3}$$

These initial gradient estimates guide subspace selection, while local training uses gradients evaluated at the updated local model.

For any candidate orthonormal basis Q, the folded initialization in Section 4.3 preserves the complete global model, allowing all candidates to be evaluated at the same starting point. The gradients with respect to the active factors BQ and $Q^\top A$ are then $\widetilde{G}_{B,i}Q$ and $Q^\top \widetilde{G}_{A,i}$, respectively. Their total squared norm measures the gradient energy retained in the trainable subspace:

$$E_i(Q) = \left\|\widetilde{G}_{B,i}Q\right\|_F^2 + \left\|Q^\top \widetilde{G}_{A,i}\right\|_F^2 = \operatorname{tr}(Q^\top C_i Q), \tag{4}$$

where $C_i \in \mathbb{R}^{R\times R}$ is given by

$$C_i = \widetilde{G}_{B,i}^\top \widetilde{G}_{B,i} + \widetilde{G}_{A,i}\widetilde{G}_{A,i}^\top. \tag{5}$$

For coordinate selection, $E_i(Q)$ reduces to the sum of the selected diagonal entries of C_i, each measuring the gradient energy of one coordinate. Allowing combinations of coordinates also makes use of the off-diagonal entries, which capture correlations between their gradients. Maximizing the retained energy over all orthonormal bases of width k_i gives

$$Q_i^0 = \mathrm{Top}_{k_i}(C_i), \tag{6}$$

where Top_{k_i} returns the eigenvectors associated with the k_i largest eigenvalues. We denote the resulting maximum energy by $E_i^0 = E_i(Q_i^0)$.

Proposition 4.1 (Local subspace advantage)**.** *Let $\lambda_1(C_i) \geq \cdots \geq \lambda_R(C_i)$, and let I_S contain the columns of the identity matrix indexed by S. For a uniformly sampled subset S of size k_i,*

$$E_i(Q_i^0) = \sum_{j=1}^{k_i} \lambda_j(C_i) \geq \max_{|S|=k_i} E_i(I_S) \geq \mathbb{E}_S[E_i(I_S)] = \frac{k_i}{R}\,\mathrm{tr}(C_i).$$

Proof. See Appendix C.1.

Thus, learning coordinate combinations retains at least as much estimated gradient energy as the best coordinate subset under the same rank budget. This local criterion, however, does not account for the different objectives across clients. We next coordinate their subspace choices using a shared global update reference while preserving the local information captured by Q_i^0.

4.2 Subspace Coordination

The local basis Q_i^0 maximizes retained gradient energy for client i, but is selected independently of other clients. To incorporate a common optimization reference, we refine Q_i^0 into a preferred basis $Q_i^\star$ that improves agreement with the previous global update while retaining most of the local gradient energy. At round $t \geq 1$, we define the shared reference as

$$R_B^t = B^{t-1} - B^t, \qquad R_A^t = A^{t-1} - A^t. \tag{7}$$

The sign is chosen to match the gradients subtracted during descent. This reference reflects the previous global step, including the effects of local training and server optimization. Each client obtains it from the current broadcast and cached previous factors, without additional communication. When no previous update is available, we retain $Q_i^\star = Q_i^0$.

We evaluate a candidate subspace by the agreement between its projected local gradient and the shared reference. Suppressing the round index, we define

$$H_i = \mathrm{sym}\Big(\widetilde{G}_{B,i}^\top R_B + \widetilde{G}_{A,i} R_A^\top\Big), \tag{8}$$

where $\mathrm{sym}(M) = (M + M^\top)/2$. For a candidate basis Q, the agreement score is $S_i(Q) = \mathrm{tr}(Q^\top H_i Q)$. With $P = QQ^\top$, this equals the inner product between the projected gradient $(\widetilde{G}_{B,i}P, P\widetilde{G}_{A,i})$ and the reference (R_B, R_A). Thus, $E_i(Q)$ measures how much local gradient information the subspace retains, while $S_i(Q)$ measures how strongly that projected gradient agrees with the historical reference.

To balance these two criteria, we construct candidates using

$$Q_i(c) = \mathrm{Top}_{k_i}\left(\frac{C_i}{E_i^0} + c\frac{H_i}{\|H_i\|_2}\right). \tag{9}$$

The normalizations control the relative scales of the two terms. We take $Q_i(0) = Q_i^0$, and increasing $c \geq 0$ places more weight on reference agreement. We search over c and accept a candidate only if

$$E_i(Q_i(c)) \geq (1-\varepsilon_B)E_i^0,\ S_i(Q_i(c)) \geq S_i(Q_i^0), \tag{10}$$

where $0 \leq \varepsilon_B < 1$ controls the allowed loss of local gradient energy. The accepted basis retains at least a $(1-\varepsilon_B)$ fraction of the locally optimal energy without reducing reference agreement. We

denote this basis by $Q_i^\star$, with $Q_i^\star = Q_i^0$ if no candidate passes. This fallback also applies when $E_i^0 = 0$ or $\|H_i\|_2 = 0$, including a zero reference.

The preferred basis $Q_i^\star$ guides the final choice of the training subspace. To allow exploration, we complete it to an orthonormal basis $V_i = [Q_i^\star, Q_i^\perp]$. With probability $1-\epsilon_i$, the client uses $Q_i = Q_i^\star$; with probability $\epsilon_i > 0$, it selects k_i columns of V_i uniformly to form Q_i. This gives every direction in V_i a positive selection probability while preserving the rank budget. The resulting Q_i is held fixed throughout local training.

4.3 Local Training

Given the sampled subspace Q_i, client i trains the corresponding active factors while keeping the remaining global adapter frozen. With $P_i = Q_i Q_i^\top$, we initialize $\bar{B}_i^{(0)} = BQ_i$ and $\bar{A}_i^{(0)} = Q_i^\top A$, and incorporate the complementary component into the frozen weight:

$$W_{0,i}^{\text{fold}} = W_0 + B(I - P_i)A. \tag{11}$$

At step h, the layer uses $W_i^{(h)} = W_{0,i}^{\text{fold}} + \bar{B}_i^{(h)}\bar{A}_i^{(h)}$. At initialization, the frozen and active components sum to $W_i^{(0)} = W_0 + BA$. Thus, every client starts from the complete global model while training only $k_i(m+n)$ factor entries. Q_i and $W_{0,i}^{\text{fold}}$ remain fixed throughout local training.

The exploration rule in the previous section gives different directions unequal probabilities of being trained. A column v_{ij} of V_i has inclusion probability $\pi_{ij} = 1 - \epsilon_i + \epsilon_i k_i/R$ if it belongs to $Q_i^\star$, and $\pi_{ij} = \epsilon_i k_i/R$ otherwise. To compensate for these unequal frequencies, we scale each selected direction's gradient by $1/\pi_{ij}$. For $Q_i = V_i[:, \mathcal{I}_i]$, the correction is represented by $D_i = \text{diag}(1/\pi_{ij} : j \in \mathcal{I}_i)$. At the initial global model, the corrected gradients, mapped back to the global factor coordinates, recover the full local gradient in expectation over subspace sampling.

Let $\bar{f}_i^{Q_i}(\bar{B}, \bar{A}; \mathcal{B})$ denote the local batch loss under the folded model. Using batch $\mathcal{B}_i^{(h)}$, client i updates both active factors simultaneously:

$$\begin{aligned} \bar{B}_i^{(h+1)} &= \bar{B}_i^{(h)} - \gamma_i \nabla_{\bar{B}} \bar{f}_i^{Q_i}(\bar{B}_i^{(h)}, \bar{A}_i^{(h)}; \mathcal{B}_i^{(h)}) D_i, \\ \bar{A}_i^{(h+1)} &= \bar{A}_i^{(h)} - \gamma_i D_i \nabla_{\bar{A}} \bar{f}_i^{Q_i}(\bar{B}_i^{(h)}, \bar{A}_i^{(h)}; \mathcal{B}_i^{(h)}). \end{aligned} \tag{12}$$

Here γ_i is the local learning rate. The correction D_i remains fixed within the round, while gradients are evaluated at the current active factors. After K_i steps, the changes in these factors form the client's update for server aggregation.

4.4 Server Aggregation

After K_i local steps, client i returns its active-factor changes and Q_i. Since these changes are expressed in different client subspaces, the server first maps them to the shared LoRA coordinates:

$$\Delta B_i = (\bar{B}_i^{(K_i)} - \bar{B}_i^{(0)})Q_i^\top, \ \Delta A_i = Q_i(\bar{A}_i^{(K_i)} - \bar{A}_i^{(0)}). \tag{13}$$

The lifted changes have the same shapes as B and A, and the factors $B + \Delta B_i$, $A + \Delta A_i$ exactly represent client i's trained folded model. The server can therefore aggregate the changes in a common coordinate system as $\Delta B = \sum_i p_i \Delta B_i$ and $\Delta A = \sum_i p_i \Delta A_i$. This aggregation averages factor changes, not the client weight products.

Although local training corrects for unequal inclusion probabilities, the subspaces selected in a particular round can still cover global directions unevenly. To limit the influence of updates along weakly covered directions, we introduce a bounded filter based on the realized subspace coverage. We summarize this coverage by $\mathsf{K} = \sum_i p_i (R/k_i) P_i$, where $v^\top \mathsf{K} v$ measures the aggregate coverage of a unit direction v. The factor R/k_i normalizes each client's contribution for its rank budget: under uniform subspace sampling, $\mathbb{E}[(R/k_i)P_i] = I$. Thus, identity provides a reference for balanced coverage. Using K, we construct

$$F = (1-\rho)I + \rho \ \text{clip}_{[0,1]}(\mathsf{K}), \tag{14}$$

where $0 \le \rho < 1$ and clipping acts on eigenvalues. The filter leaves directions with coverage eigenvalues at least one unchanged, while attenuating those below one by factors between $1-\rho$ and 1. Its effect is therefore bounded and cannot increase either factor update's Frobenius norm.

The server applies the filter in the shared latent coordinates and updates the global factors as

$$B^+ = B + \eta_s \Delta B F,\ A^+ = A + \eta_s F \Delta A, \tag{15}$$

where η_s is the server learning rate. Setting $\rho = 0$ recovers unfiltered aggregation of the lifted changes. For $\rho > 0$, the filter attenuates weakly covered update components while preserving sufficiently covered ones. This introduces filtering bias, so any reduction in update variability must be balanced against the deviation from the unfiltered aggregate.

5 Theoretical Analysis

We establish a convergence guarantee for the global factor objective J in Eq. (2). Since J is generally nonconvex in (B, A), we measure convergence by the average expected squared gradient norm. Let $X_t = (B_t, A_t)$ and $g_t = \nabla J(X_t)$, where $\|g_t\|^2$ sums the squared Frobenius norms of the gradients over all factor blocks.

For the analysis, we consider full participation, K plain SGD steps per client with learning rate γ, and a scalar server step η_s. We assume L-smooth client and global objectives along their optimization trajectories, a lower-bounded J, and conditionally unbiased full-factor stochastic gradients with second moments bounded by G^2. Training batches are independent of the batch used for subspace construction. Each round, clients sample fresh directions conditional on the constructed bases, with inclusion probabilities uniformly bounded below by $\pi_{\min} > 0$, independently of T. We write $M = 1/\pi_{\min}$ to bound the probability correction. The formal assumptions appear in Appendix C.6.

Theorem 5.1 (Nonconvex convergence)**.** *Under Assumptions C.1–C.3 in Appendix C.6, let $J \geq J_*$ and $\Delta_J = J(X_0) - J_*$. For $0 < \eta = \eta_s \gamma K \leq 1/L$ and $0 \leq \rho < 1$,*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\,\|g_t\|^2 \leq \frac{2\Delta_J}{\eta T} + L\eta M^2 G^2 + G^2\left(\frac{L\gamma(K-1)M^2}{2} + \rho M\right)^2. \tag{16}$$

In particular, fix $K, \eta_s, a > 0$, $\lambda \geq 0$, and $0 \leq \rho_{\max} < 1$ independently of T, with $\eta_s a K \leq 1/L$. For a training horizon of T rounds, choosing $\gamma = a/\sqrt{T}$ and $\rho = \min\{\rho_{\max}, \lambda\gamma K\}$ gives

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\,\|g_t\|^2 \leq \frac{2\Delta_J/(\eta_s a K) + L\eta_s a K M^2 G^2}{\sqrt{T}} + \frac{a^2 G^2}{T}\left(\frac{L(K-1)M^2}{2} + \lambda K M\right)^2. \tag{17}$$

Hence, the average expected squared gradient norm converges to zero at rate $O(T^{-1/2})$.

Proof. See Appendix C.6.

Theorem 5.1 establishes convergence toward first-order stationarity of the original global objective under heterogeneous rank budgets. In Eq. (16), the first term reflects the initial objective gap, the second bounds the effect of stochastic updates, and the final term accounts for local drift and server filtering. Under the stated parameter choice, the first two terms decay as $O(T^{-1/2})$ and the final term as $O(T^{-1})$, so the bound contains no nonvanishing residual. The dependence on M captures the effects of local rank budgets and exploration probabilities through the probability correction.

For fixed client count and local workload, this rate matches FSLoRA's leading-order dependence on T (Fang et al., 2026). FSLoRA analyzes an expected sketched objective, whereas our guarantee applies directly to $J(B, A)$ for the complete global adapter. Further properties of subspace selection, coordination, and coverage filtering are discussed in Appendices C.1, C.2, and C.4.

6 Limitations

We discuss two limitations of our method. First, the subspace construction introduces additional computation for gradient estimation and eigendecomposition. Reusing the selected basis throughout each local round limits this overhead to subspace initialization and refresh. Second, a fixed basis may become less aligned with local gradients as the model evolves, particularly over longer local training rounds. Together, these limitations highlight a trade-off between computational overhead and maintaining alignment with evolving local gradients. Future work could develop refresh schedules based on gradient changes, reusing the basis when gradients remain stable and updating it when they change substantially.

Table 1: GLUE accuracy (%) with RoBERTa. Avg. denotes the unweighted mean, with CoLA evaluated by accuracy.

Method	GPU h	QNLI	MRPC	CoLA	MNLI	RTE	SST-2	QQP	Avg.
Base model	–	49.5	52.0	30.9	32.6	47.3	50.9	63.2	46.6
HetLoRA	**10.7**	$87.5_{\pm 0.5}$	$84.4_{\pm 0.9}$	$75.3_{\pm 1.2}$	$66.3_{\pm 0.8}$	$69.0_{\pm 1.7}$	$89.5_{\pm 0.0}$	$85.3_{\pm 0.1}$	79.6
FlexLoRA	12.6	$\underline{88.5}_{\pm 0.2}$	$81.2_{\pm 0.4}$	$77.5_{\pm 1.2}$	$63.0_{\pm 0.5}$	$62.2_{\pm 1.9}$	$92.8_{\pm 0.4}$	$\mathbf{87.4}_{\pm 0.1}$	78.9
FLoRA	12.3	$87.2_{\pm 0.3}$	$78.1_{\pm 0.7}$	$77.4_{\pm 1.7}$	$74.6_{\pm 0.5}$	$54.4_{\pm 2.1}$	$93.4_{\pm 0.1}$	$\underline{87.1}_{\pm 0.3}$	78.9
RAVAN	12.3	$83.4_{\pm 1.1}$	$76.2_{\pm 1.6}$	$71.8_{\pm 1.4}$	$66.3_{\pm 3.6}$	$63.1_{\pm 2.8}$	$92.1_{\pm 0.9}$	$83.4_{\pm 1.7}$	76.6
Fed-PLoRA	11.8	$88.1_{\pm 0.8}$	$86.8_{\pm 0.5}$	$81.6_{\pm 0.7}$	$75.6_{\pm 0.4}$	$69.3_{\pm 1.2}$	$93.2_{\pm 0.2}$	$86.2_{\pm 0.9}$	83.0
FSLoRA	$\underline{10.9}$	$88.0_{\pm 0.3}$	$\underline{87.3}_{\pm 0.2}$	$\underline{82.2}_{\pm 0.5}$	$\underline{76.4}_{\pm 0.2}$	$\underline{69.8}_{\pm 1.2}$	$\underline{93.5}_{\pm 0.1}$	$85.8_{\pm 0.1}$	$\underline{83.3}$
FedCSO-LoRA	11.3	$\mathbf{89.7}_{\pm 0.3}$	$\mathbf{89.3}_{\pm 0.5}$	$\mathbf{83.8}_{\pm 0.6}$	$\mathbf{80.9}_{\pm 0.3}$	$\mathbf{75.1}_{\pm 1.1}$	$\mathbf{94.1}_{\pm 0.2}$	$86.3_{\pm 0.2}$	**85.6**

7 EXPERIMENTS

7.1 EXPERIMENTAL SETUP

Training settings. We follow the experimental settings of FSLoRA and use its reported scores and GPU times (Fang et al., 2026). Both benchmarks use 20 clients, 200 communication rounds, Dirichlet $\alpha = 0.1$, global rank $R = 64$, and client budgets $\{8, 16, 32, 48\}$. The effective batch size is 16 and LoRA dropout is 0.1, with matched data partitions and rank assignments across methods.

Datasets and models. For natural language understanding, we fine-tune RoBERTa-base (Liu et al., 2019) on seven GLUE tasks (Wang et al., 2018), using label-based client partitions. Each client performs 50 local updates per round with learning rate 5×10^{-4}. For commonsense reasoning, we fine-tune LLaMA-3.2-3B-Instruct on eight tasks with heterogeneous client task mixtures (Hu et al., 2023). This setting uses 20 local updates per round and learning rate 3×10^{-4}. We report task accuracy and the unweighted task average, including accuracy for CoLA to match the reference protocol. Appendix B provides the full protocol.

Baselines. We compare FedCSO-LoRA with six federated LoRA methods under heterogeneous client rank budgets, including HetLoRA (Cho et al., 2024), FlexLoRA (Bai et al., 2024), FLoRA (Wang et al., 2024b), RAVAN (Raje et al., 2025), FSLoRA (Fang et al., 2026), and Fed-PLoRA (Zhang et al., 2026). We also report the pretrained model before fine-tuning as the base model.

7.2 MAIN COMPARISONS

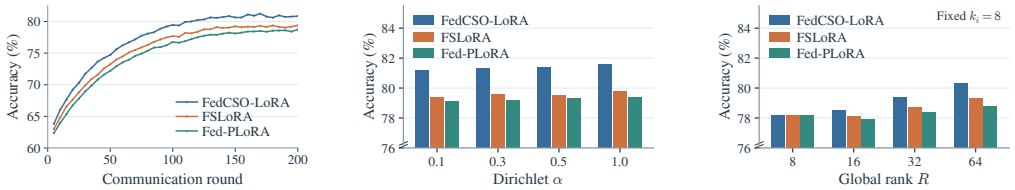
Natural Language Understanding. Table 1 compares the methods on GLUE. FedCSO-LoRA achieves an average accuracy of 85.6%, exceeding FSLoRA at 83.3% by 2.3 percentage points and Fed-PLoRA at 83.0% by 2.6 points. It outperforms both methods on all seven tasks. The gains over FSLoRA are largest on RTE and MNLI at 5.3 and 4.5 points, respectively. FedCSO-LoRA achieves the highest accuracy on six tasks, while FlexLoRA remains stronger on QQP.

Commonsense Reasoning. Table 2 shows that the gains extend to LLaMA fine-tuning. FedCSO-LoRA reaches 81.2% average accuracy, exceeding FSLoRA at 79.4% by 1.8 points and Fed-PLoRA at 79.1% by 2.1 points. It achieves the highest accuracy on all eight tasks and outperforms both methods on every task. Its gains over FSLoRA range from 0.4 points on ARC-Easy to 2.7 points on ARC-Challenge and HellaSwag. Together, the two benchmarks show gains across different model sizes and task groups under heterogeneous rank budgets.

Training and rank sensitivity. Figure 3(a) shows that FedCSO-LoRA improves throughout training and reaches 81.2% at round 200. Its accuracy remains above both FSLoRA and Fed-PLoRA across the tested Dirichlet settings in Figure 3(b). In Figure 3(c), all methods coincide at $R = k_i = 8$, when each client trains the complete adapter. With k_i fixed at 8, the gap increases as R grows, consistent with the benefit of selecting directions from a larger shared space.

Table 2: LLaMA-3.2-3B-Instruct reasoning accuracy (%). Avg. is the unweighted task average, computed before rounding. Subscripts give task-level standard deviations where reported.

Method	GPU h	ARC-c	ARC-e	BoolQ	HSwag	OBQA	PIQA	SIQA	Wino	Avg.
Base model	–	51.5	61.4	47.2	55.8	52.4	21.3	54.4	4.6	43.6
HetLoRA	**43.7**	$73.4_{\pm 0.3}$	$86.6_{\pm 0.2}$	$65.8_{\pm 0.5}$	$73.0_{\pm 0.5}$	$71.4_{\pm 0.3}$	$80.9_{\pm 0.7}$	$73.8_{\pm 0.3}$	$72.0_{\pm 0.3}$	74.6
FlexLoRA	68.3	$74.2_{\pm 0.3}$	$86.7_{\pm 0.6}$	$68.6_{\pm 0.8}$	$79.4_{\pm 0.7}$	$75.8_{\pm 0.4}$	$81.0_{\pm 0.3}$	$75.9_{\pm 0.4}$	$77.9_{\pm 0.3}$	77.4
FLoRA	49.8	$68.3_{\pm 0.6}$	$83.1_{\pm 0.5}$	$65.8_{\pm 0.9}$	$77.2_{\pm 0.5}$	$74.2_{\pm 0.3}$	$80.5_{\pm 0.6}$	$76.1_{\pm 0.5}$	$71.5_{\pm 0.5}$	74.6
RAVAN	51.2	$71.9_{\pm 0.9}$	$81.6_{\pm 1.3}$	$65.7_{\pm 3.2}$	$76.7_{\pm 1.4}$	$75.6_{\pm 1.5}$	$77.1_{\pm 1.2}$	$74.8_{\pm 0.6}$	$73.8_{\pm 0.6}$	74.7
Fed-PLoRA	46.6	$75.7_{\pm 0.5}$	$87.0_{\pm 0.6}$	$69.0_{\pm 0.8}$	$81.7_{\pm 1.1}$	$80.4_{\pm 0.8}$	$83.7_{\pm 0.4}$	$76.5_{\pm 0.4}$	$78.8_{\pm 0.5}$	79.1
FSLoRA	44.3	$76.1_{\pm 0.4}$	$87.2_{\pm 0.5}$	$69.3_{\pm 0.7}$	$82.2_{\pm 1.1}$	$80.7_{\pm 0.6}$	$84.0_{\pm 0.2}$	$76.8_{\pm 0.0}$	$79.1_{\pm 0.2}$	79.4
FedCSO-LoRA	45.8	$\mathbf{78.8}_{\pm 0.5}$	$\mathbf{87.6}_{\pm 0.5}$	$\mathbf{71.9}_{\pm 0.8}$	$\mathbf{84.9}_{\pm 1.0}$	$\mathbf{81.2}_{\pm 0.7}$	$\mathbf{84.5}_{\pm 0.3}$	$\mathbf{79.0}_{\pm 0.3}$	$\mathbf{81.5}_{\pm 0.3}$	**81.2**



(a) Training progress (b) Data heterogeneity (c) Global rank

Figure 3: Commonsense reasoning accuracy. (a) Progress over communication rounds. (b) Data heterogeneity under Dirichlet parameter α. (c) Global rank R with every client fixed at $k_i = 8$. The accuracy axes in (b) and (c) are truncated as marked.

Computational Cost. FedCSO-LoRA uses 11.3 GPU hours on GLUE and 45.8 GPU hours on commonsense reasoning. These costs are approximately 3.7% and 3.4% higher than those reported for FSLoRA, respectively, but remain below the costs of Fed-PLoRA, FlexLoRA, FLoRA, and RAVAN. Compared with FSLoRA's random coordinate selection, learning subspaces requires additional gradient estimation and basis construction. The selected basis stays fixed during local training, and reusing preferred bases between refresh rounds can reduce this overhead. The accuracy gains therefore come with a modest increase in GPU time over FSLoRA.

Ablation Study. Table 7 in Appendix B.4.1 compares three ablation variants against the full method under the same experimental setting. All variants use the same client rank budgets, local training protocol, folded forward model, exploration rule, and probability correction. The first variant removes both local subspace selection and subspace coordination, using randomly selected global coordinates as its preferred basis. The second retains local selection but removes coordination, so its preferred basis is the locally selected Q_i^0. The third retains selection and coordination but removes coverage filtering by setting $\rho = 0$ and $F = I$. Removing both local selection and coordination reduces average accuracy from 85.6% to 84.0%. When local selection is retained but coordination is removed, accuracy increases to 84.7%, showing the benefit of gradient-informed subspace selection. Adding coordination further improves accuracy to 85.6%. Removing coverage filtering reduces accuracy to 84.5%, indicating that accounting for collective subspace coverage also contributes to the final performance. Overall, the full method achieves the highest average accuracy.

8 Conclusion

We introduced FedCSO-LoRA to select and coordinate client update spaces under heterogeneous LoRA rank budgets. The proposed method could preserve the complete global adapter during rank-constrained local training and account for collective subspace coverage during server aggregation. We provided a nonconvex convergence analysis that characterized the effects of local training and aggregation. Experiments on natural language understanding and commonsense reasoning have demonstrated a consistent improvement over the comparison methods.

AI USE STATEMENT

Generative AI tools were used only to assist with language editing and improving the readability of the manuscript. Generative AI tools were not used for research tasks requiring mandatory disclosure under the ICLR 2027 AI policy. All AI-assisted edits were reviewed by the authors, who take responsibility for the final content of this work.

ETHICS STATEMENT

This work studies federated parameter-efficient fine-tuning on established public benchmarks and does not collect new human-subject data. The experiments use benchmark-based client partitions rather than deployment data from real users.

REPRODUCIBILITY STATEMENT

We provide detailed algorithmic, experimental, and theoretical information to support reproducibility. Appendix A gives the full training procedure and implementation details, including the folded factor representation, blockwise initial-gradient estimation, and optimization scope. Appendix B specifies the benchmarks, shared configuration, FedCSO-LoRA hyperparameters, evaluation and comparison protocol, component ablations, sensitivity analyses, rank-budget experiments, communication accounting, and reported training costs. Appendix C states the theoretical assumptions and provides the corresponding results and proofs.

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, 2024.

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, Main Conference, pp. 120038–120090, 2025.

Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, and Gauri Joshi. Heterogeneous lora for federated fine-tuning of on-device foundation models. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 12903–12913, 2024.

Wenzhi Fang, Dong-Jun Han, Liangqi Yuan, Seyyedali Hosseinalipour, and Christopher Brinton. Federated sketching loRA: A flexible framework for heterogeneous collaborative fine-tuning of LLMs. In *Forty-third International Conference on Machine Learning*, 2026.

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

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *Forty-first International Conference on Machine Learning*, 2024.

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*, 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.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 5254–5276, 2023.

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*, 2020.

Seanie Lee, Sangwoo Park, Dong Bok Lee, Dominik Wagner, Haebin Seong, Tobias Bocklet, Juho Lee, and Sung Ju Hwang. Fedsvd: Adaptive orthogonalization for private federated learning with lora. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 119733–119757, 2025.

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

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

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

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 121038–121072, 2024.

Fabian Paischer, Lukas Hauzenberger, Thomas Schmied, Benedikt Alkin, Marc Deisenroth, and Sepp Hochreiter. Parameter efficient fine-tuning via explained variance adaptation. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 46215–46267, 2025.

Arian Raje, Baris Askin, Divyansh Jhunjhunwala, and Gauri Joshi. Ravan: Multi-head low-rank adaptation for federated fine-tuning. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 48564–48592, 2025.

Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

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

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*, pp. 353–355, 2018.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7611–7623, 2020.

Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 54905–54931, 2024a.

Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. LoRA-pro: Are low-rank adapters properly optimized? In *The Thirteenth International Conference on Learning Representations*, 2025.

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, 2024b.

Jui-Nan Yen, Si Si, Zhao Meng, Felix Yu, Sai Surya Duvvuri, Inderjit S Dhillon, Cho-Jui Hsieh, and Sanjiv Kumar. LoRA done RITE: Robust invariant transformation equilibration for loRA optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yuchen Zeng and Kangwook Lee. The expressive power of low-rank adaptation. In *The Twelfth International Conference on Learning Representations*, 2024.

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

Yuanhe Zhang, Fanghui Liu, and Yudong Chen. LoRA-one: One-step full gradient could suffice for fine-tuning large language models, provably and efficiently. In *Forty-second International Conference on Machine Learning*, 2025.

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

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient LLM training by gradient low-rank projection. In *Forty-first International Conference on Machine Learning*, 2024.

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 *Forty-first International Conference on Machine Learning*, 2024.

APPENDIX

This appendix provides supplementary material to the main paper and is organized into three sections. Section A gives the complete training procedure, folded implementation, blockwise gradient estimation, and implementation details of FedCSO-LoRA. Section B describes the experimental protocol and presents additional ablations, mechanism controls, sensitivity studies, rank-budget experiments, and computation and communication analyses. Finally, Section C provides the theoretical results and proofs, including local subspace optimality, coordination guarantees, probability-corrected exploration, coverage-aware aggregation, and convergence.

A ALGORITHM AND IMPLEMENTATION DETAILS

A.1 FULL TRAINING PROCEDURE

Algorithm 1 gives the complete training loop. We group the initial factor gradients as $\widetilde{g}_i = (\widetilde{G}_{B,i}, \widetilde{G}_{A,i})$ and their shared update reference as $r_t = (R_B^t, R_A^t)$ from Eq. (7), adding round and layer indices when needed. The algorithm uses the local updates in Eq. (12) and the scalar server update in Eq. (15). Theorem 5.1 analyzes equal local workloads, fresh direction samples, and a learning-rate-dependent coverage filter. Unequal workloads are described in Appendix C.5 without extending that theorem.

Algorithm 1 FedCSO-LoRA

Input: global factors $X_0 = (B_0, A_0)$, weights $\{p_i\}$, rank budgets $\{k_i\}$, rounds T, tolerance ε_B, exploration probabilities $\{\epsilon_i\}$, coverage strength ρ_t.
Output: adapted factors X_T.

for $t = 0, \ldots, T-1$ **do**
  Broadcast X_t. Clients recover r_t from cached previous and current global factors, with $r_0 = 0$.
  When a preferred basis is initialized or refreshed, client i computes $\widetilde{g}_{i,t}$ locally; otherwise it reuses the cached preferred basis. The gradients are not uploaded.
  for each client i and LoRA layer ℓ **do**
    If needed, form $C_{i\ell}, H_{i\ell}$ and obtain $Q^\star_{i\ell}$ by Eqs. (5), (8), and (34); else reuse cached $Q^\star_{i\ell}$. [A]
    Sample fresh $Q_{i\ell}$ and construct $D_{i\ell}$ as in Appendix C.3. [B]
  end for
  for each client i **in parallel do**
    Initialize the folded local model as in Section 4.3, run Eq. (12) with fixed bases, and return the factor changes and selected bases.
  end for
  Lift the factor changes with Eq. (13) and average them with weights p_i.
  Form K_t from the sampled bases and F_t from Eq. (14). [C]
  Apply Eq. (15). Clients cache X_t for the next round.
end for

A.2 EXACT FACTOR REPRESENTATION AND FOLDED IMPLEMENTATION

The symbols U, V below are used only to shorten this equivalence proof. They denote differences of active factors, not additional trainable variables. The main text trains the active factors $\bar{B}, \bar{A}$ directly by Eq. (12). At any local step, define their changes from initialization by $U = \bar{B} - BQ$ and $V = \bar{A} - Q^\top A$. With the orthonormal basis $Q \in \mathbb{R}^{R\times k}$ fixed, training U, V is equivalent to training $\bar{B}, \bar{A}$ because they differ by constant matrices. In full factor coordinates, the same local model has the parameterization

$$\begin{aligned} B_i &= B + UQ^\top, \\ A_i &= A + QV. \end{aligned} \tag{18}$$

For the current batch loss, let G_B, G_A be its gradients with respect to these full-size factors. The chain rule gives $\nabla_{\bar{B}} f_i = \nabla_U f_i = G_B Q$ and $\nabla_{\bar{A}} f_i = \nabla_V f_i = Q^\top G_A$. Thus the gradient map is

$$(G_B, G_A) \mapsto (G_B Q, Q^\top G_A).$$

Let $P = QQ^\top$. Mapping these coordinate gradients back to the original factor space without a probability correction gives the projected gradient pair

$$\Pi_P(G_B, G_A) = (G_B P, P G_A). \tag{19}$$

With the corrected local updates in Eq. (12), this map becomes $(G_B \widehat{P}_i, \widehat{P}_i G_A)$ for $\widehat{P}_i = Q_i D_i Q_i^\top$, as defined in Appendix C.3. The forward identities below are unchanged by this choice of update rule. The forward pass uses the exact product

$$(B + UQ^\top)(A + QV) - BA = BQV + UQ^\top A + UV. \tag{20}$$

The right-hand side has rank at most $2k$. We therefore interpret k as the active factor-update width, not as the rank of the complete forward adapter or of the final weight increment.

With $Q^\top Q = I$, the active factors are obtained from the full factors as $B_i Q = BQ + U$ and $Q^\top A_i = Q^\top A + V$. This proves the direct relationship between the two forms in the main text. The same local model admits an exact folded implementation. With $P = QQ^\top$, decompose the global adapter as

$$BA = B(I - P)A + BPA.$$

For an explicit LoRA scaling coefficient $s > 0$, the same folded construction applies. For example, conventional LoRA uses $s = \alpha/R$. The implemented global layer is $W_0 + sBA$. Use the same coefficient for both branches, with active factors $\bar{B} = BQ + U$ and $\bar{A} = Q^\top A + V$. Define the frozen weight as

$$W_0^{\text{fold}} = W_0 + sB(I - P)A. \tag{21}$$

Then

$$W_0 + s(B + UQ^\top)(A + QV) = W_0^{\text{fold}} + s\bar{B}\bar{A}. \tag{22}$$

Thus, once Q is fixed, the inactive part of the global adapter can be absorbed into the frozen weight and local optimization can use a rank-k active LoRA branch. This equality is exact. It does not approximate the global adapter. The same coefficient is used when computing gradients through the scaled layer. In particular, replacing the global scale with a client-specific α/k only in the active branch would generally break the initialization identity. Gradient estimation and construction of Q add to the cost of local training.

A.3 Blockwise initial gradient estimation

Initial factor gradients are estimated from client training data at the current global adapter. Computing them over all client training examples with dropout disabled gives the empirical evaluation-mode factor gradient up to arithmetic error. Smaller batches introduce estimation error that affects preferred-basis quality. No held-out examples are used to select the client basis.

When device memory is limited, a client can evaluate the initial factor gradients in blocks and store completed blocks in host memory. This is an optional implementation of the same estimator. It does not replace the full set of rank coordinates with a smaller candidate space. The construction applies to both factors and does not require alternating their training.

Block gradient evaluation. Fix the received global model and the estimation batch. Partition the R coordinates into disjoint blocks of width at most $b_i \le k_i$. Let $E_j \in \mathbb{R}^{R \times b_j}$ contain the identity columns for block j, so that $\sum_j E_j E_j^\top = I$. For this estimation pass, apply the folded construction in Appendix A.2 with $Q = E_j$. Initialize the small factors as BE_j and $E_j^\top A$, and freeze the complement. With the same fixed LoRA scale s, the initial weight is

$$W_0 + sB(I - E_j E_j^\top)A + s(BE_j)(E_j^\top A) = W_0 + sBA. \tag{23}$$

Evaluate the loss and differentiate only with respect to the active factors, without taking an optimizer step. The chain rule gives

$$\begin{aligned} L_{B,i}^{(j)} &= \widetilde{G}_{B,i} E_j, \\ L_{A,i}^{(j)} &= E_j^\top \widetilde{G}_{A,i}. \end{aligned} \tag{24}$$

These identities describe derivatives obtained directly from the small factors. The implementation need not first allocate the full factor gradients on the device. Move the results to host memory,

release the computation graph, and repeat for the next block. Every pass must start from the same global model, including the frozen complement. Other adapted layers must also represent the same initial model in each pass.

The complete gradients are recovered in their original coordinate order as

$$\begin{aligned}\widetilde{G}_{B,i} &= \sum_j L_{B,i}^{(j)} E_j^\top, \\ \widetilde{G}_{A,i} &= \sum_j E_j L_{A,i}^{(j)}.\end{aligned} \tag{25}$$

In particular, the block of C_i associated with blocks j, h is

$$E_j^\top C_i E_h = (L_{B,i}^{(j)})^\top L_{B,i}^{(h)} + L_{A,i}^{(j)} (L_{A,i}^{(h)})^\top. \tag{26}$$

All pairs j, h are needed. Keeping only the diagonal blocks discards correlations and changes the selected subspace. The historical reference is recovered from cached model factors and can be read from host memory in matching blocks. All cross-block terms needed for H_i must also be retained. No current factor gradients are transmitted for coordination. If the estimation batch is processed in microbatches, accumulate the appropriately weighted gradients before forming C_i. Averaging separate microbatch Gram matrices would define a different estimator.

In exact arithmetic, Eqs. (24) and (25) reproduce the gradients of a single full evaluation when all passes use identical parameters, examples, loss normalization, and stochastic forward computation. A simple sufficient choice is to disable dropout and other stochastic layers for both the reference full evaluation and the block evaluations, without updating model buffers. If stochastic layers are retained, their realized forward functions must match across passes, including any branches affected by folding. Reusing a random seed alone does not ensure this when the graph changes. No parameters or optimizer states are updated between blocks.

Under these conditions, blocking introduces no additional mathematical approximation into the batch gradient or its Gram matrix. It does not eliminate the batch estimator's error relative to the local objective or local training drift. Finite precision and changed reduction order can cause numerical differences. Nearly tied eigenvalues can also yield different numerical bases. We therefore claim equivalence of the estimator in exact arithmetic, not bitwise equality of the implementation or exact population gradients.

For one layer, device storage for active factor gradients scales with $b_i(m+n)$ rather than $R(m+n)$, while host storage for the complete gradients remains $R(m+n)$ and the Gram matrix requires R^2 entries. The straightforward procedure uses $\lceil R/b_i \rceil$ estimation passes. For $R = 64$ and $b_i = k_i = 8$, this is eight passes before selecting an eight-dimensional subspace that can still mix all 64 coordinates. This exchanges device gradient memory for recomputation and data movement. It does not reduce the model or endpoint communication payload, and it does not by itself bound activation memory, frozen-weight storage, or folding workspace. Actual peak memory and runtime depend on the model and hardware.

A.4 Optimization and implementation scope

The objective in Eq. (2) is optimized with respect to the LoRA factors. Lifting makes each client's factor changes compatible with the global coordinates. The coverage filter then adjusts their aggregate. Neither operation is claimed to reproduce the average of client weight products. Separate factor averaging can differ from product averaging because both factors change, but equality with that alternative aggregation rule is not required by our objective. The analysis instead controls the actual factor update through its gradient residual and the server filter. We retain simultaneous local updates of both active factors.

The server may broadcast the full global rank-R adapter, as in recent heterogeneous federated LoRA protocols. Local active training after the basis is fixed scales with $k_i(m+n)$ parameters under the folded form in Eq. (21). Gradient estimation, the full-rank downlink, basis communication, cached-adapter storage, and folding add to the cost of active training.

The preferred basis and its orthonormal completion may be retained as implementation state once constructed. The client forms C_i, H_i, and V_i from its initial gradient estimate and the current

reference, then draws the active subset using the conditional sampling rule. The active factors and frozen remainder are initialized from the current global factors. The conditional identity $\mathbb{E}[Q_i D_i Q_i^\top \mid V_i] = I$ is stated with respect to the preferred basis used for that update.

FSLoRA analyzes an expected sketched objective and makes the sketch ratio explicit in its convergence rate (Fang et al., 2026). Our analysis targets the factor objective with the complete global anchor. Probability-corrected local updates give an unbiased expected direction at the round anchor, while local drift and filtering are bounded explicitly. FSLoRA's rescaled sketch appears in its forward model. Our correction acts on the gradient of an exactly folded forward model. Unequal workloads are outside our main theorem. Neither guarantee is uniformly stronger than the other.

A.5 Folding, dropout, and factor geometry

The algebraic folding identity assumes the same deterministic linear map on both sides. If dropout is applied to the entire full-rank adapter but only to the active branch after folding, the stochastic forward maps differ. A matched implementation can disable adapter dropout for the exact-equivalence check, or define dropout consistently on the same active branch and include that convention in every control. A gradient estimate computed without dropout also estimates an evaluation-mode training objective. If the local stochastic objective uses dropout, the difference belongs in the gradient estimation error relative to the local objective rather than being silently discarded.

Selection uses the Euclidean metric of the current factorization. It is invariant to orthogonal latent reparameterizations. For $B' = BO$, $A' = O^\top A$, $O^\top O = I$, the gradient Gram matrix is $C_i' = O^\top C_i O$ and the selected projector transforms accordingly. It is not invariant to arbitrary invertible rescalings $B' = BT$, $A' = T^{-1}A$. Equal learning rates for the two factor blocks are implicit in the unweighted energy criterion. Factor asymmetry and invariant preconditioning are studied by Zhu et al. (2024) and Yen et al. (2025). A different factor metric, including unequal rates as in LoRA+ (Hayou et al., 2024), changes the first-order objective for preserving gradient energy and requires an explicit weighted criterion. The current paper analyzes the stated shared-factor metric.

B Experimental Settings and Additional Results

B.1 Benchmarks and Shared Configuration

We use RoBERTa-base on seven GLUE tasks and LLaMA-3.2-3B-Instruct on eight commonsense reasoning tasks. The GLUE tasks are QNLI, MRPC, CoLA, MNLI, RTE, SST-2, and QQP; the reasoning tasks are ARC-Challenge, ARC-Easy, BoolQ, HellaSwag, OpenBookQA, PIQA, SocialIQA, and WinoGrande. Accuracy is averaged uniformly across the tasks within each benchmark. CoLA uses accuracy, following the FSLoRA comparison, rather than Matthews correlation.

Shared benchmark settings follow FSLoRA where applicable (Fang et al., 2026). Table 3 collects the principal settings. The same data partition, rank assignment, training workload, and evaluation procedure are used for matched comparisons. Unless a sweep explicitly changes them, there are 20 fully participating clients, global rank $R = 64$, heterogeneous client ranks $\{8, 16, 32, 48\}$, and Dirichlet concentration 0.1. Training updates follow Algorithm 1.

B.2 FedCSO-LoRA Hyperparameters on GLUE

Table 4 gives the method-specific GLUE configuration. Each client estimates its initial factor gradients using one local training batch when the preferred basis is initialized or refreshed. This estimate is used for subspace construction and is additional to the 50 local updates in the corresponding communication round. It uses training data only.

The energy constraint compares the coordinated subspace with the locally optimal subspace at the same estimated gradient: $E_i(Q_i^\star) \geq 0.95 E_i(Q_i^0)$. It does not require retaining 95% of the energy of the unrestricted full-rank gradient. The historical reference includes the actual server transformation; it is recovered from consecutive global factors rather than from an unfiltered aggregate. With the convention in Eq. (7), it equals the negative of the preceding parameter displacement.

Table 3: Shared training configuration. The benchmark hyperparameters follow the latest FSLoRA protocol (Fang et al., 2026).

Setting	GLUE	Commonsense reasoning
Backbone	RoBERTa-base	LLaMA-3.2-3B-Instruct
Clients / participation	20 / full	20 / full
Communication rounds	200	200
Global rank R	64	64
Client ranks k_i	$\{8, 16, 32, 48\}$	$\{8, 16, 32, 48\}$
Dirichlet concentration	0.1	0.1
Local updates	50 per round	20 per round
Effective batch size	16	16
Local learning rate	5×10^{-4}	3×10^{-4}
LoRA dropout	0.1	0.1
Target modules	Query, value, classification head	`q_proj`, `k_proj`, `v_proj`, `up_proj`, `down_proj`

Table 4: FedCSO-LoRA settings for GLUE. Shared training hyperparameters follow Table 3 and FSLoRA.

Parameter	Setting
Initial-gradient estimation	One local training batch per client when the preferred basis is initialized or refreshed
Coordination tolerance ε_B	0.05; retain at least 95% of the estimated energy of the locally optimal subspace
Historical reference	Previous actual server update, expressed in the gradient-direction sign convention of Eq. (7)
First communication round	Zero reference; skip coordination
Coverage strength ρ	Target 0.25; ramp up during the first three rounds
Exploration probability ϵ_i	0.2 for all clients
Spectral search	Relative bisection tolerance 10^{-3}; at most 20 expansions and 28 bisection steps

The coverage sweep varies the target strength while retaining the three-round ramp-up. The coefficients ε_B and ρ control the coordination acceptance constraint and coverage attenuation, respectively. The numerical settings in Table 4 are used for GLUE; reasoning experiments follow the benchmark-specific settings described in Section B.1.

B.3 Evaluation and Comparison Protocol

We retain the task scores and module ablations already reported in the main manuscript. FSLoRA benchmark scores are taken from its published comparison (Fang et al., 2026). The Fed-PLoRA scores in the main comparison are obtained under the same common benchmark protocol for a matched reproduction, rather than copied from the original Fed-PLoRA tables, whose model and federated settings differ (Zhang et al., 2026). The additional rank studies use the same fixed-round convention across methods: 200 communication rounds and the benchmark-specific local workload. This differs from a fixed-communication-budget comparison, in which methods may complete different numbers of rounds.

For sensitivity experiments, only the selected method-specific parameter changes. For rank experiments, both data partition and local workload remain fixed while the specified global rank, local rank, or allocation changes. All sweeps use three independent seeds. Matched methods and ablations reuse the same three seed identifiers so that each comparison uses paired data partitions and stochastic initialization conditions. All sweep results report the mean over the same three independent seeds.

Paired-seed analysis. Table 5 reports the measured gains in unweighted benchmark-average accuracy over FSLoRA for three matched seeds. Each gain is computed from the two methods under the same seed. We use a two-sided paired t-test with two degrees of freedom for each benchmark and apply Holm correction across the two benchmark tests.

Table 5: Three-seed paired comparison with FSLoRA. Gains are percentage points.

Benchmark	Paired gains (pp)	Avg. gain (pp)	Paired p	Holm-adjusted p
GLUE	2.0, 2.1, 2.8	+2.3	0.0118	0.0235
Commonsense reasoning	1.3, 2.0, 2.1	+1.8	0.0190	0.0235

B.4 Ablations, Mechanism Controls, and Hyperparameter Sensitivity

B.4.1 Module ablations

Table 6 defines the ablations used in Table 7. All variants use the same client rank budgets, full global forward anchor, local training workload, and probability-corrected local update rule. The coordinate-only variant uses uniform selection of identity columns. Its removal of both local selection and coordination isolates the gain from learning client-specific directions while preserving the rest of the training procedure. The local-only variant isolates coordination, and the unfiltered variant isolates server coverage filtering.

Table 6: Definition of the module ablations. The preferred subspace is specified before exploration.

Variant	Preferred directions	Server filtering
Full method	Local energy selection and coordination	Enabled
w/o selection + coordination	Uniform global coordinates	Enabled
w/o coordination	$Q_i^0 = \mathrm{Top}_{k_i}(C_i)$	Enabled
w/o coverage filtering	Local energy selection and coordination	$\rho = 0$, $F = I$

Removing both local selection and coordination reduces the average from 85.6% to 84.0%. Retaining local selection without coordination recovers the average to 84.7%, showing the benefit of gradient-informed subspace selection. Adding coordination further raises the average to 85.6%. Disabling coverage filtering reduces the average to 84.5%, indicating that coverage-aware aggregation also contributes to the final performance.

Table 7: GLUE ablation accuracy (%), reported as mean ± standard deviation over three independent runs. Values are shown to one decimal; bold and underline denote the best and second-best unrounded means in each column, respectively. Avg. is the unweighted task mean.

Method	QNLI	MRPC	CoLA	MNLI	RTE	SST-2	QQP	Avg.
Full method	$89.7_{\pm 0.3}$	$89.3_{\pm 0.5}$	$83.8_{\pm 0.6}$	$80.9_{\pm 0.3}$	$75.1_{\pm 1.1}$	$\underline{94.1_{\pm 0.2}}$	$86.3_{\pm 0.2}$	$85.6_{\pm 0.3}$
w/o selection + coordination	$88.3_{\pm 0.4}$	$87.7_{\pm 0.6}$	$82.5_{\pm 0.7}$	$78.0_{\pm 0.4}$	$72.3_{\pm 1.3}$	$93.5_{\pm 0.2}$	$85.6_{\pm 0.2}$	$84.0_{\pm 0.4}$
w/o coordination	$\underline{89.0_{\pm 0.3}}$	$\underline{88.4_{\pm 0.5}}$	$\underline{83.2_{\pm 0.6}}$	$\underline{79.3_{\pm 0.3}}$	$\underline{73.0_{\pm 1.2}}$	$94.1_{\pm 0.2}$	$85.5_{\pm 0.2}$	$\underline{84.7_{\pm 0.3}}$
w/o coverage filtering	$88.8_{\pm 0.3}$	$88.3_{\pm 0.5}$	$83.0_{\pm 0.6}$	$79.1_{\pm 0.3}$	$72.9_{\pm 1.2}$	$93.7_{\pm 0.2}$	$\underline{85.7_{\pm 0.2}}$	$84.5_{\pm 0.3}$

The task-level differences in Figure 4 show that the largest degradations from removing coverage filtering occur on RTE and MNLI, while SST-2 and QQP are less affected. Together with the other ablations, this pattern suggests that local selection, coordination, and coverage-aware aggregation provide complementary gains whose effects vary across tasks rather than a fixed additive offset.

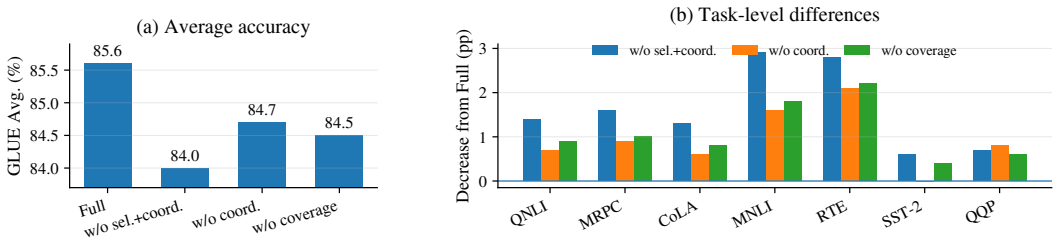


Figure 4: Module ablations from Table 7. Left: the reported GLUE average for each variant. Right: task-level accuracy decreases relative to the full method, computed from the displayed task values. Larger values indicate a larger decrease after removing the corresponding component.

B.4.2 Coordinate importance versus learned combinations

Uniform coordinate selection, importance-based coordinate selection, and a dense learned basis separate two effects. Uniform selection does not use local gradient information. The importance-coordinate control selects the k_i largest diagonal entries of C_i, using gradient magnitudes while remaining coordinate-aligned. The dense local basis instead uses the leading eigenspace of C_i and can exploit off-diagonal correlations between latent coordinates.

For the importance-coordinate control, let I_i^{imp} contain the selected indices and set $Q_i^{\mathrm{imp}} = I[:, I_i^{\mathrm{imp}}]$. Its retained initial energy is $\sum_{j \in I_i^{\mathrm{imp}}} (C_i)_{jj}$, whereas the dense local optimum retains $\sum_{j=1}^{k_i} \lambda_j(C_i)$. Proposition 4.1 therefore gives a nonnegative energy gap in favor of the dense basis at the same estimated gradient. This is an initial projection statement and does not by itself imply a fixed end-of-training accuracy gain.

Table 8: Orientation controls under the GLUE configuration. Accuracy values are means over three independent runs and are shown to one decimal. Bold and underline denote the best and second-best values in each numeric column. Energy retention refers to the preferred subspace before exploration.

Preferred subspace	Energy retention (%)	GLUE Avg. (%)
Uniform coordinates	40.8	84.0
Importance coordinates	57.6	84.4
Dense local basis	**68.1**	<u>84.7</u>
Coordinated dense basis	<u>65.4</u>	**85.6**

Importance-based coordinate selection increases retained energy from 40.8% to 57.6%, while the dense local basis further raises it to 68.1% by exploiting off-diagonal correlations in C_i. After coordination, the preferred basis retains 65.4% energy, giving an aggregate retention ratio

of $65.4/68.1 \approx 96.0\%$, consistent with the client-wise 95% acceptance constraint induced by $\varepsilon_B = 0.05$. The corresponding GLUE averages increase from 84.0% for uniform coordinates to 84.4% for importance coordinates, 84.7% for the dense local basis, and 85.6% after coordination.

B.4.3 Sensitivity to method-specific hyperparameters

Table 9 reports one-at-a-time GLUE sensitivity studies centered on the default configuration. The tolerance sweep uses $\varepsilon_B \in \{0, 0.01, 0.025, 0.05, 0.10, 0.20\}$ and the coverage sweep uses $\rho \in \{0, 0.10, 0.25, 0.40, 0.60, 0.80\}$. Each study keeps the other parameter at its default. Reported values are means over three independent seeds.

Table 9: Sensitivity of GLUE average accuracy (%) to method-specific hyperparameters. Values are shown to one decimal; bold parameter values denote the default settings.

Parameter	Tested value					
ε_B	0	0.01	0.025	0.05	0.10	0.20
GLUE Avg.	84.7	85.3	85.5	85.6	85.6	84.9
ρ	0	0.10	0.25	0.40	0.60	0.80
GLUE Avg.	84.5	85.3	85.6	85.6	85.2	84.7

Coordination tolerance. A small relaxation of locally optimal energy retention permits useful coordination, whereas larger relaxations can discard too much local information. Performance increases from 84.7% at $\varepsilon_B = 0$ to 85.6% at the default 0.05, remains essentially unchanged at 85.6% for 0.10, and declines to 84.9% at 0.20. The broad favorable region indicates that moderate coordination improves the shared update without requiring a sharply tuned tolerance.

Coverage strength. At $\rho = 0$, the coverage filter is the identity, matching the existing unfiltered ablation. Increasing the target strength attenuates under-covered directions more strongly. The no-filter setting gives 84.5%, while the default $\rho = 0.25$ reaches 85.6%. Performance remains close at 85.6% for $\rho = 0.40$ and decreases under stronger filtering, indicating that moderate attenuation is beneficial whereas aggressive filtering can suppress useful updates.

B.5 Additional Experiments under Varying Rank Budgets

The additional experiments separate three questions: how much active capacity each client needs, how unequal budgets affect collaboration, and whether a larger global adapter helps when local capacity stays fixed. This design follows the local/global-rank studies in FSLoRA and the resource-allocation studies in Fed-PLoRA (Fang et al., 2026; Zhang et al., 2026).

B.5.1 Reference performance under the default heterogeneous setting

Figure 5 presents task-level differences against FSLoRA under the default configuration. It grounds the scale of the additional studies in the existing comparison while keeping that configuration separate from homogeneous-rank sweeps. In particular, the published heterogeneous-rank average should not be treated as a measured point for an all-rank-8 or all-rank-32 population.

B.5.2 Different client rank budgets

We fix $R = 64$ and assign a common active rank $k \in \{8, 16, 32, 48, 64\}$ to all 20 clients. This separates active client capacity from the heterogeneity of its allocation. The FSLoRA local-rank experiment motivates the design; the intermediate ranks provide finer resolution around the budgets used in our main comparison. Each point is averaged over three independent seeds.

Figure 6 shows the effect of homogeneous client rank under fixed $R = 64$. On GLUE, both methods improve with increasing client rank and the gap decreases from 2.2 points at $k = 8$ to 0.5 points at $k = 48$. For reasoning, the $k = 8$ endpoint is the same measured $R = 64, k_i = 8$ configuration reported in the main-text global-rank sweep, giving 79.3% for FSLoRA and 80.3% for FedCSO-LoRA. At $k = 16$, FSLoRA reaches 79.1% while FedCSO-LoRA reaches 80.5%, a 1.4-point gain. Across the other restricted ranks, FedCSO-LoRA is higher by 1.0, 0.9, and 0.5 points at

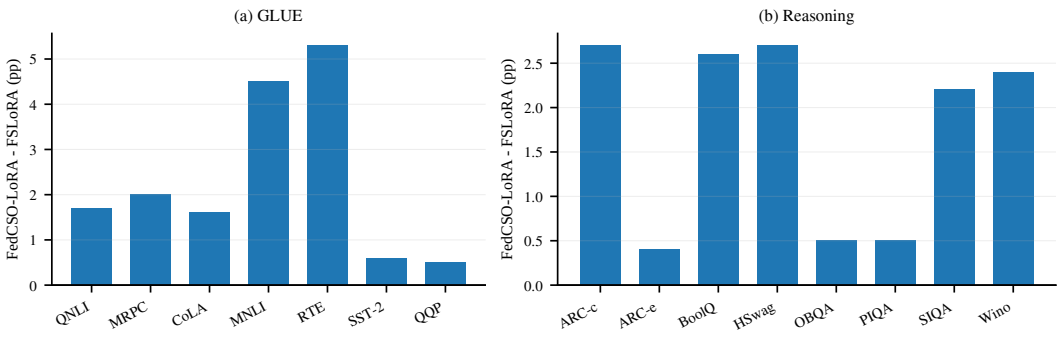


Figure 5: Reported task-level accuracy differences relative to FSLoRA. FedCSO-LoRA values are taken from the main manuscript; FSLoRA values follow its published benchmark. These are the default heterogeneous-rank comparisons, not additional sweep results.

$k = 8, 32, 48$, respectively. At $k = R = 64$, there is no subspace restriction and the two methods coincide at the matched full-rank boundary.

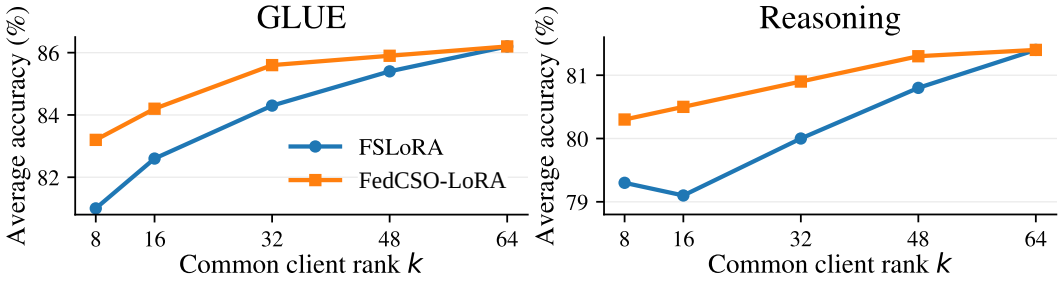


Figure 6: Performance under homogeneous client ranks with $R = 64$. All clients share the rank on the horizontal axis. Each point is the mean over three independent seeds under the matched GLUE and reasoning settings.

B.5.3 Different resource heterogeneity settings

The first study varies the client rank set while assigning five of the 20 clients to each of its four entries. Table 10 specifies every assignment. These configurations vary both average capacity and its distribution; they are stress tests of resource settings rather than an isolated test of rank variance.

Table 10: Rank allocations for the GLUE resource-heterogeneity study. Each of the four listed ranks is assigned to five clients; $R = 64$ throughout.

Setting	Client rank set	Mean active rank
A: constrained	$\{4, 8, 16, 32\}$	15
B: default	$\{8, 16, 32, 48\}$	26
C: wide spread	$\{4, 8, 32, 64\}$	27
D: higher capacity	$\{16, 32, 48, 64\}$	40

The second study fixes the two ranks to $k_{\text{low}} = 8$ and $k_{\text{high}} = 32$, and changes the low-rank fraction to $20\%, 40\%, 60\%, 80\%, 100\%$. With 20 clients, these fractions correspond to $4, 8, 12, 16, 20$ low-rank clients. This complements Figure 1(b), which keeps the low-rank group size fixed and changes its rank. Here the group ranks stay fixed and only their population sizes change. Every setting is evaluated with three independent seeds.

Setting B reproduces the default allocation and yields 83.3% for FSLoRA and 85.6% for FedCSO-LoRA. Setting C has a slightly larger mean active rank than B but a wider allocation and gives 82.9% and 85.1%, respectively, showing that average capacity alone does not determine performance. Setting D raises both methods under higher client capacity, while the FedCSO-LoRA advantage narrows to 1.2 points. In the low-rank-fraction study, increasing the fraction of rank-8 clients from 20% to 100% decreases both methods, but the gain of FedCSO-LoRA increases from 1.1 to 2.2 points. The 100% endpoint matches the homogeneous $k = 8$ setting.

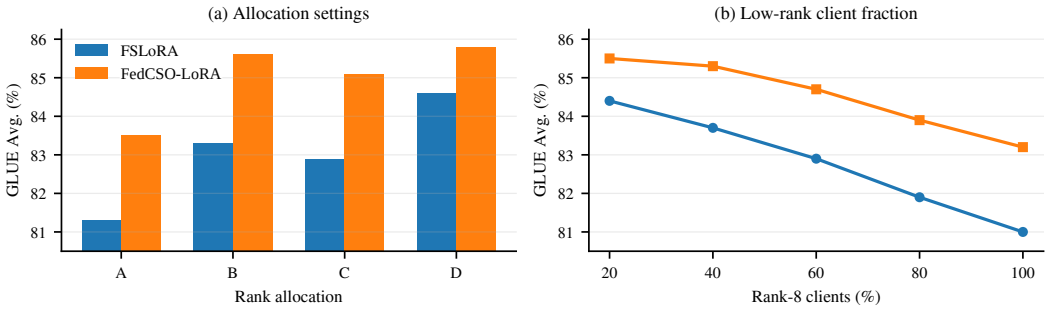


Figure 7: Resource heterogeneity on GLUE. Left: the four allocations in Table 10. Right: changing the fraction of rank-8 clients among rank-8 and rank-32 clients. Each point is the mean over three independent seeds; setting B is the default allocation.

B.5.4 Aggregate results under different global ranks

We fix every client's active rank to $k_i = 8$ and vary $R \in \{8, 16, 32, 64\}$, following the global-rank design in FSLoRA. Figure 8 reports the eight-task averages from the measured sweep summarized in the main text. This experiment holds the local trainable width fixed while increasing the latent space from which directions can be selected. Each point is averaged over three independent seeds.

At $R = k_i = 8$, all three methods reach 78.2%, corresponding to the unrestricted rank-8 boundary. With k_i fixed at 8, FedCSO-LoRA reaches 78.5%, 79.4%, and 80.3% at $R = 16, 32, 64$, compared with 78.1%, 78.7%, and 79.3% for FSLoRA and 77.9%, 78.4%, and 78.8% for Fed-PLoRA. The FedCSO-LoRA margin over FSLoRA therefore grows from 0.4 to 0.7 and 1.0 points as the available global rank increases. This trend is consistent with the role of learned subspaces: under a fixed local rank budget, selecting effective directions becomes more consequential as the shared latent space expands. A larger global rank also increases the full-adapter downlink, as detailed in Appendix B.6.

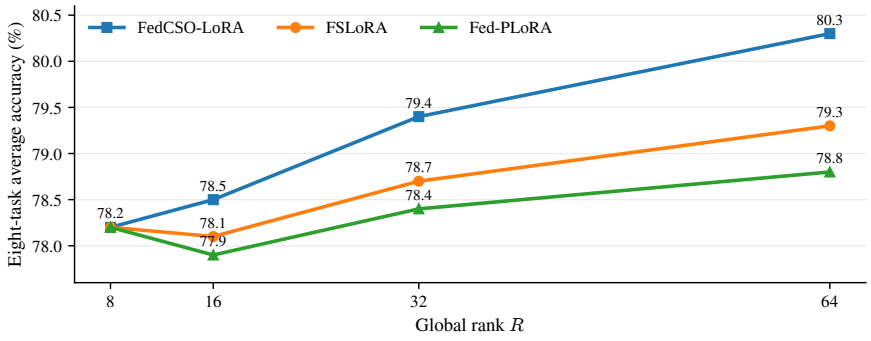


Figure 8: Eight-task reasoning accuracy under different global ranks, with every client fixed at $k_i = 8$. The values are the same measured global-rank sweep summarized in the main text, shown here with aggregate values. Each point is the mean over three independent seeds.

B.6 Communication and Reported Training Cost

B.6.1 Communication Accounting

The full-participation protocol broadcasts $R(m+n)$ factor values per layer and client. Each client sends its compact active-factor endpoint, containing $k_i(m+n)$ values, together with its basis Q_i when the server does not already hold that basis. The latter contains Rk_i values. The reference uses the cached previous broadcast and adds no current-gradient upload. Hence the per-client payload counts are

$$C_i^{\text{down}} = R(m+n), \tag{27}$$

$$C_i^{\text{up}} = k_i(m+n) + Rk_i. \tag{28}$$

These counts exclude headers and compression. Different endpoint and basis precisions require separate byte counts. With equal precision, the basis overhead is $R/(m+n)$ relative to the active endpoint. The total upload relative to full rank-R factors is

$$\frac{C_i^{\text{up}}}{R(m+n)} = \frac{k_i}{R}\left(1 + \frac{R}{m+n}\right). \tag{29}$$

The full-adapter downlink remains proportional to R. Thus a compact upload does not imply a corresponding reduction in total bidirectional communication.

Table 11: Exact payload counts for a representative square layer with $m = n = 4096$ and $R = 64$. Each numeric value occupies two bytes in this calculation. The table is an analytical payload calculation, not a runtime measurement; it excludes protocol overhead.

Active rank	Endpoint (KiB)	Basis (KiB)	Total upload (KiB)	Fraction of full upload
8	128	1	129	12.60%
16	256	2	258	25.20%
32	512	4	516	50.39%
48	768	6	774	75.59%

In Table 11, the full downlink is 1,024 KiB, and the basis adds 0.78% to the active endpoint at each rank. Narrower layers have a larger relative basis overhead. The counts assume explicit basis transmission.

The cached previous adapter adds $R(m+n)$ stored values per layer, potentially on the host. Missing caches or partial participation require a fallback or an explicit reference payload. The stated accounting assumes full participation and available caches, with no current-gradient exchange.

B.6.2 Reported Training Cost

Table 12 collects the total GPU hours from the main comparison. FedCSO-LoRA uses the same four-NVIDIA-A100 hardware configuration as the FSLoRA benchmark implementation, with the matched benchmark workload. The numerical differences relative to FSLoRA are 0.4 GPU hours for GLUE and 1.5 GPU hours for reasoning, or approximately 3.7% and 3.4%. These totals support a hardware-matched runtime comparison, while the aggregate times do not identify stage-level overhead.

Table 12: GPU hours reported in the main comparison under the matched four-NVIDIA-A100 configuration. FSLoRA values follow its published benchmark; the totals do not isolate stage-level runtime.

Method	GLUE	Commonsense reasoning
FSLoRA	**10.9**	**44.3**
Fed-PLoRA	11.8	46.6
FedCSO-LoRA	<u>11.3</u>	<u>45.8</u>

C THEORETICAL RESULTS AND PROOFS

This section follows the method components before deriving the corrected-update convergence bound.

C.1 OPTIMAL LOCAL SUBSPACE SELECTION

For a rank-k_i orthogonal projector P, define

$$E_i(P) = \left\|\widetilde{G}_{B,i}P\right\|_F^2 + \left\|P\widetilde{G}_{A,i}\right\|_F^2 = \operatorname{tr}(PC_i),$$

where C_i is given in Eq. (5).

We first prove Proposition 4.1 and then record its corresponding first-step descent certificate. Among all rank-k_i orthogonal projectors, $P_i^0 = Q_i^0(Q_i^0)^\top$ with $Q_i^0 = \operatorname{Top}_{k_i}(C_i)$ maximizes $E_i(P)$. Equivalently, Q_i^0 maximizes Eq. (4) among orthonormal $R \times k_i$ bases. If the eigenvalues of C_i are sorted in decreasing order, then

$$E_i(P_i^0) = \sum_{j \le k_i} \lambda_j(C_i),$$

$$\left\|(I - \Pi_{P_i^0})\widetilde{g}_i\right\|^2 = \sum_{j > k_i} \lambda_j(C_i).$$

A uniformly sampled k_i-coordinate projector preserves $(k_i/R)\operatorname{tr}(C_i)$ squared projected-gradient norm in expectation. The same identity holds for a rotation-invariant random rank-k_i projector.

For any coordinate projector P_I of rank k_i,

$$E_i(P_i^0) = \sum_{j=1}^{k_i} \lambda_j(C_i) \ge \max_{|I|=k_i} \sum_{j \in I} (C_i)_{jj} \ge \frac{k_i}{R}\operatorname{tr}(C_i). \tag{30}$$

Thus local spectral selection has no larger discarded estimated squared gradient norm than any coordinate choice at the same model and active width. The first inequality can be strict when useful directions mix coordinates. As an auxiliary comparison for a hypothetical uncorrected projected step, if the estimate equals the true local gradient and f_i is L_i-smooth along the step, then

$$f_i(X - \gamma\Pi_P g_i) \le f_i(X) - \gamma(1 - L_i\gamma/2)E_i(P). \tag{31}$$

Here $0 < \gamma < 2/L_i$. The selected space maximizes this auxiliary first-step decrease certificate. The implemented local SGD instead uses probability-corrected gradients and can sample a different active space. This certificate is not a guarantee for its realized step or multiple-step endpoint.

Proof. Cyclicity of the trace and $P^2 = P$ give

$$\left\|\widetilde{G}_B P\right\|_F^2 + \left\|P\widetilde{G}_A\right\|_F^2 = \operatorname{tr}(P\widetilde{G}_B^\top\widetilde{G}_B) + \operatorname{tr}(P\widetilde{G}_A\widetilde{G}_A^\top) = \operatorname{tr}(PC).$$

Diagonalize $C = V\Lambda V^\top$ with decreasing eigenvalues. Let $a_j = v_j^\top P v_j$. Then $0 \le a_j \le 1$ and $\sum_j a_j = k$, so $\operatorname{tr}(PC) = \sum_j \lambda_j a_j \le \sum_{j \le k} \lambda_j$, with equality for the leading eigenspace. The residual follows by Pythagoras. For either random projector, $\mathbb{E}P = (k/R)I$, giving the expected energy. Coordinate projectors form a subset of feasible projectors, and the largest k diagonal entries have sum at least $(k/R)\operatorname{tr}(C)$. Finally, smoothness and $\langle g_i, \Pi_P g_i\rangle = \|\Pi_P g_i\|^2$ give Eq. (31). □

C.2 COORDINATION OBJECTIVE AND SPECTRAL SEARCH

At the beginning of round t, the reference $r_t = (B^{t-1} - B^t, A^{t-1} - A^t)$ is available from consecutive global models. In the full-participation protocol, each client caches its previous broadcast. There is no exchange of current initial gradients. The first round uses $r_0 = 0$ and local selection. A missing or inconsistent cache also uses local selection unless the server explicitly supplies the missing reference

and accounts for its communication. Cache storage, potentially in host memory, costs $R(m+n)$ values per layer.

The reference includes previous subspace restrictions, local training, and server filtering. It is a common direction, not an unbiased current global gradient. Multiplying it by any positive scalar leaves the normalized spectral path and the acceptance decisions unchanged. The claims below therefore hold for any fixed positive rescaling of r_t, without introducing a new algorithm parameter.

Write $P_i^0 = Q_i^0 (Q_i^0)^\top$ and let

$$H_i = \mathrm{sym}\Big(\widetilde{G}_{B,i}^\top R_B + \widetilde{G}_{A,i} R_A^\top\Big). \tag{32}$$

The coordination score is $S_i(P) = \mathrm{tr}(PH_i)$. Here $\widetilde{g}_i = (\widetilde{G}_{B,i}, \widetilde{G}_{A,i})$, $r = (R_B, R_A)$, and $\mathrm{sym}(M) = (M + M^\top)/2$. The trace expression equals $\langle r, \Pi_P \widetilde{g}_i \rangle$. The underlying constrained problem is

$$\max_{P \in \mathcal{P}_{k_i}} S_i(P) \quad \text{subject to} \quad E_i(P) \geq (1-\varepsilon_B) E_i^0, \tag{33}$$

where $\mathcal{P}_{k_i} = \{P = P^\top = P^2 : \mathrm{tr}(P) = k_i\}$ and $0 \leq \varepsilon_B < 1$. We search the following family of candidates rather than claiming a global solution of this constrained problem

$$Q_i(c) = \mathrm{Top}_{k_i}\left(\frac{C_i}{E_i(P_i^0)} + c\frac{H_i}{\|H_i\|_2}\right). \tag{34}$$

For $c \geq 0$, put $P_i(c) = Q_i(c) Q_i(c)^\top$. The proposed search first checks the leading eigenspace of H_i, then performs bounded expansion and bisection on c. A candidate is accepted only if

$$\begin{aligned} E_i(P_i) &\geq (1-\varepsilon_B) E_i(P_i^0), \\ S_i(P_i) &\geq S_i(P_i^0). \end{aligned} \tag{35}$$

If no candidate with $c > 0$ passes the checks, the method returns P_i^0. Denote the accepted preferred projector by $P_i^\star = Q_i^\star (Q_i^\star)^\top$. In this subsection the candidate P_i and Π_i in the guarantees below refer to $P_i^\star$ before exploration. If $E_i(P_i^0) = 0$ or $\|H_i\|_2 = 0$, the normalization is skipped and the local projector is retained. For a gradient estimate with numerically negligible norm, a deterministic orthonormal completion supplies any remaining budget directions. No division by the small energy is performed. Repeated eigenvalues can yield multiple optimal bases. All acceptance decisions depend on projectors and trace values, not on individual eigenvector signs.

Proposition C.1 (Initial coordination guarantee)**.** *Let $d_A = \sum_i a_i \Pi_{P_i^0} \widetilde{g}_i$ and $d_B = \sum_i a_i \Pi_{P_i} \widetilde{g}_i$ for any fixed $a_i \geq 0$. If Eq. (35) hold for all clients, then*

$$\langle r, d_B - d_A \rangle = \sum_i a_i \big[S_i(P_i) - S_i(P_i^0)\big] \geq 0. \tag{36}$$

Moreover,

$$\|(I - \Pi_{P_i})\widetilde{g}_i\|^2 \leq \mathcal{E}_i^{\mathrm{rank}} + \varepsilon_B E_i(P_i^0),$$

where $\mathcal{E}_i^{\mathrm{rank}} = \mathrm{tr}(C_i) - E_i(P_i^0)$.

Let $g = \nabla J(X)$ on the same factor blocks, $v = d_B - d_A$, $\Delta = \langle r, v \rangle \geq 0$, and $\delta = \|g - r\|$. Then

$$\langle g, d_B - d_A \rangle \geq \Delta - \delta \|d_B - d_A\|. \tag{37}$$

In particular, true initial alignment improves whenever $\Delta > \delta \|d_B - d_A\|$. For the smoothness decrease certificate $\mathcal{D}_\eta(d) = \langle g, d \rangle - (L\eta/2)\|d\|^2$,

$$\mathcal{D}_\eta(d_B) - \mathcal{D}_\eta(d_A) \geq \Delta - \delta \|v\| - \frac{L\eta}{2}\big(\|d_B\|^2 - \|d_A\|^2\big). \tag{38}$$

A positive right-hand side gives a larger certified initial decrease. These statements concern the same anchor and initial projected gradients, before the local endpoint residual.

Proof. The identity in Eq. (32) follows from

$$\left\langle R_B, \widetilde{G}_{B,i}P \right\rangle + \left\langle R_A, P\widetilde{G}_{A,i} \right\rangle = \operatorname{tr}\Big(P \operatorname{sym}(\widetilde{G}_{B,i}^\top R_B + \widetilde{G}_{A,i} R_A^\top)\Big).$$

Linearity then gives Eq. (36). The residual bound follows from $\|(I - \Pi_P)\widetilde{g}_i\|^2 = \operatorname{tr}(C_i) - E_i(P)$. Writing $g = r + (g - r)$ and applying Cauchy–Schwarz proves Eq. (37). Subtracting the two squared-norm penalties proves Eq. (38). □

For the population weights p_i, Proposition C.1 gives

$$\left\langle r, \sum_i p_i(\Pi_i - \Pi_i^0)\widetilde{g}_i \right\rangle \geq 0, \tag{39}$$

$$\|(I - \Pi_i)\widetilde{g}_i\|^2 \leq \mathcal{E}_i^{\mathrm{rank}} + \varepsilon_B E_i^0. \tag{40}$$

Let $e(P) = E_i(P)/E_i(P_i^0)$ and $a(P) = S_i(P)/\|H_i\|_2$. If P_1 and P_2 globally maximize $e + c_1 a$ and $e + c_2 a$ with $c_2 > c_1 \geq 0$, then optimality implies $a(P_2) \geq a(P_1)$ and $e(P_2) \leq e(P_1)$. For completeness, write $e_j = e(P_j)$ and $a_j = a(P_j)$. Optimality gives $e_1 + c_1 a_1 \geq e_2 + c_1 a_2$ and $e_2 + c_2 a_2 \geq e_1 + c_2 a_1$. Adding these inequalities yields $(c_2 - c_1)(a_2 - a_1) \geq 0$. Substituting back gives $e_2 - e_1 \leq c_1(a_1 - a_2) \leq 0$. The exact maximizing path can jump at eigenvalue ties, so the practical procedure should be viewed as a feasible spectral search rather than a solver for the unrestricted constrained problem. The specified procedure uses bounded expansion and bisection, followed by explicit checks on preserved energy and alignment in training precision.

If the checks admit numerical slacks ϵ_i^E and ϵ_i^S, the corresponding bounds become

$$\|(I - \Pi_i)\widetilde{g}_i\|^2 \leq \mathcal{E}_i^{\mathrm{rank}} + \varepsilon_B E_i(P_i^0) + \epsilon_i^E,$$
$$\langle r, d_B - d_A \rangle \geq -\sum_i a_i \epsilon_i^S.$$

The coordination result concerns the initial projected gradient. With a historical reference, inexact local gradient estimates, and multi-step local training, the true endpoint additionally contains errors from gradient estimation and local training. Therefore Proposition C.1 is not a claim that every coordinated local epoch has lower loss than its uncoordinated counterpart.

For any positive normalization a of the previous update reference, smoothness gives

$$\|g_t - a r_t\| \leq \|g_{t-1} - a r_t\| + L\,\|X_t - X_{t-1}\|. \tag{41}$$

The first term includes previous projection, local-training, and server-update effects. The second measures model change. Neither is assumed to be zero. Proposition C.1 does not guarantee that an arbitrary historical reference helps. The energy constraint and fallback remain valid even when the reference is inaccurate.

Analytical coordination example. Figure 1(a) compares coordinate selection and dense mixing at the same width in a six-dimensional latent space. Matrix shading shows how components contribute to each active direction.

For an analytical coordination example, take two equally weighted clients with gradient estimates $g_1 = (1, 2)^\top$ and $g_2 = (1, -1)^\top$, and a historical reference that happens to equal $g = (1, 1/2)^\top$. This is a favorable reference configuration for illustration, not an assumption on the training protocol. Embed these vectors in a single nonzero row of G_B, with $G_A = 0$. Then $C_i = g_i g_i^\top$, so its leading rank-one projector retains all local energy. Their aggregate equals g and has alignment $\langle g, g \rangle = 1.25$.

Rotate the first basis clockwise by 10 degrees and the second counterclockwise by 10 degrees. For the resulting unit vectors q_i, let $d_i = q_i q_i^\top g_i$. Each retains the energy fraction $\cos^2(10^\circ) = 0.969846\ldots$, while

$$\left\langle g, \frac{d_1 + d_2}{2} \right\rangle = 1.468822\ldots > 1.25.$$

Individual alignments also increase, so both adjustments pass the acceptance checks with an energy loss tolerance above 0.030154. This shows that maximizing local gradient energy need not maximize shared alignment before local drift arises. It compares preferred bases, not every sampled active basis.

C.3 Exploration and probability correction

For each client and layer, complete the preferred basis to an orthogonal matrix $V_i = [Q_i^\star, Q_i^\perp] \in \mathbb{R}^{R\times R}$. The completion is fixed conditional on the current gradient-estimation batch and reference. Let $\mathcal{I}_i^\star = \{1, \dots, k_i\}$ and fix $0 < \epsilon_i \le 1$. In each round, choose $\mathcal{I}_i = \mathcal{I}_i^\star$ with probability $1 - \epsilon_i$. Otherwise draw a uniformly random size-k_i subset of $\{1, \dots, R\}$ without replacement. The uniform branch may also return $\mathcal{I}_i^\star$. Set $Q_i = V_i[:, \mathcal{I}_i]$, ordering its columns by the selected indices. The marginal probabilities are

$$\pi_{ij} = \Pr(j \in \mathcal{I}_i \mid V_i) = \begin{cases} 1 - \epsilon_i + \epsilon_i k_i/R, & j \le k_i, \\ \epsilon_i k_i/R, & j > k_i. \end{cases} \tag{42}$$

When $k_i = R$, all probabilities are one. Each round samples a fresh active set, even if its preferred basis is cached. Reusing one sampled active set for multiple rounds falls outside Theorem 5.1. The active width remains k_i. Its directions can combine the original LoRA coordinates.

Write $D_i = \operatorname{diag}(1/\pi_{ij} : j \in \mathcal{I}_i)$ and distinguish the active projector $P_i = Q_i Q_i^\top$ from the corrected update matrix $\widehat{P}_i = Q_i D_i Q_i^\top$. For a pair of factor gradients define $\mathcal{S}_i(G_B, G_A) = (G_B \widehat{P}_i, \widehat{P}_i G_A)$. If Z_{ij} indicates inclusion, conditioning on the completed basis gives

$$\mathbb{E}[\widehat{P}_i \mid V_i] = \sum_{j=1}^{R} \frac{\mathbb{E}[Z_{ij} \mid V_i]}{\pi_{ij}} v_{ij} v_{ij}^\top = I. \tag{43}$$

The identity permits dependence among the directions drawn without replacement. It also permits data-dependent V_i by conditioning on its construction. Applied to the gradients of the exactly folded model at the initial global model, it gives an unbiased expected factor update. It does not state that the endpoint after multiple local steps is unbiased. The server receives active endpoints and Q_i. It does not need V_i, D_i, or current initial gradients. The correction is not repeated at aggregation.

The correction has a variance cost. If $\pi_{\min} = \inf_{i,j,t} \pi_{ij} > 0$ uniformly over the analyzed horizons, then $\left\|\widehat{P}_i\right\|_2 \le M = 1/\pi_{\min}$. At a fixed true factor gradient $g = (G_B, G_A)$, let $C = G_B^\top G_B + G_A G_A^\top$. Orthogonality gives

$$\mathbb{E}\left\|(G_B \widehat{P}_i, \widehat{P}_i G_A)\right\|^2 = \sum_{j=1}^{R} \frac{v_{ij}^\top C v_{ij}}{\pi_{ij}}. \tag{44}$$

For $k_i < R$ and $\epsilon_i < 1$, write $\pi_{\mathrm{hi}} = 1 - \epsilon_i + \epsilon_i k_i/R$ and $\pi_{\mathrm{lo}} = \epsilon_i k_i/R$. Equation (44) becomes $\operatorname{tr}(C)/\pi_{\mathrm{lo}} + (1/\pi_{\mathrm{hi}} - 1/\pi_{\mathrm{lo}}) \operatorname{tr}((Q_i^\star)^\top C Q_i^\star)$. For a fixed C and these probabilities, the top-k_i eigenspace minimizes the second moment. Our C_i uses an initial batch, so this conclusion applies exactly to its estimated gradient and need not hold for the unknown population gradient. Coordination can increase the corresponding estimated second moment by at most $\varepsilon_B E_i^0 (1/\pi_{\mathrm{lo}} - 1/\pi_{\mathrm{hi}})$. As ϵ_i decreases, this bound and M can grow. Probability correction gives unbiasedness at the anchor without guaranteeing a uniform empirical gain from coordination.

C.4 Server aggregation with subspace coverage

The coverage matrix K used in Eq. (14) is positive semidefinite because it is a positive weighted sum of projectors. If $\mathsf{K} = V \operatorname{diag}(k_j) V^\top$, then

$$F = V \operatorname{diag}\big(1 - \rho + \rho \min(k_j, 1)\big) V^\top.$$

Hence

$$(1 - \rho) I \preceq F \preceq I. \tag{45}$$

In particular, $\|F\|_2 \le 1$. The lifted operator $\mathcal{F}(X_B, X_A) = (X_B F, F X_A)$ is self-adjoint under the Frobenius inner product and has the same spectral bounds. Orthogonal changes of basis inside a selected subspace do not change $Q_i Q_i^\top$ and therefore do not change the coverage matrix.

The normalization R/k_i is calibrated to uniformly random rank-k_i subspaces. For a uniformly random projector P_i, $\mathbb{E}[(R/k_i) P_i] = I$. Our preferred-basis mixture is nonuniform, so the same

expectation need not equal I for the actual sampling rule. The filter remains a bounded measure of realized collective support, not an inverse correction for missing directions or a certificate that low-coverage directions are noisy. The filter strength in the main theorem decreases with the local learning rate.

Proposition C.2 (Conditional advantage of coverage filtering)**.** *Condition on the global model, initial estimates, selected spaces, and fixed workload scales before local training. Let $g = \nabla J(X)$, let z be the normalized aggregate endpoint from Appendix C.5, and write $\mu = \mathbb{E}[z \mid \mathcal{I}]$, $\Sigma = \mathrm{Cov}(z \mid \mathcal{I})$. The lifted filter $\mathcal{F}$ is fixed under this conditioning. Define*

$$\Gamma_C = \mathrm{tr}\big((I - \mathcal{F}^2)\Sigma\big) - \big(\,\|\mathcal{F}\mu - g\|^2 - \|\mu - g\|^2\,\big). \tag{46}$$

Then

$$\mathbb{E}[\|\mathcal{F}z - g\|^2 \mid \mathcal{I}] = \mathbb{E}[\|z - g\|^2 \mid \mathcal{I}] - \Gamma_C. \tag{47}$$

Thus filtering reduces gradient approximation error precisely when variance reduction exceeds the increase in squared bias, namely $\Gamma_C > 0$.

Proof. Writing $z = \mu + (z - \mu)$ eliminates the cross term after conditional expectation. The two errors are $\|\mathcal{F}\mu - g\|^2 + \mathrm{tr}(\mathcal{F}\Sigma\mathcal{F})$ and $\|\mu - g\|^2 + \mathrm{tr}(\Sigma)$. Subtract and use symmetry of $\mathcal{F}$. No unbiasedness or independence between the initial gradient estimate and selected spaces is required. □

For interpretation only, consider the special case $\mu = g$. In an eigenbasis of $\mathcal{F}$, let f_j be its eigenvalues and σ_j^2 the conditional directional variances. Then

$$\Gamma_C = \sum_j \big[(1 - f_j^2)\sigma_j^2 - (1 - f_j)^2 g_j^2\big]. \tag{48}$$

For $f_j < 1$, the contribution is positive when $\sigma_j^2 > (1 - f_j)g_j^2/(1 + f_j)$. This explains a possible benefit in directions with high variability and weak signal. Low coverage alone does not establish this condition. If $z = g$ deterministically, filtering instead adds $\|(I - \mathcal{F})g\|^2$ error. The proposition therefore supplies a conditional advantage and a measurable diagnostic, not a universal dominance claim.

C.5 Endpoint error and local workloads

The main theorem analyzes a common K and plain SGD. For completeness, the following identity describes how unequal local work or reset momentum changes an endpoint. It is not an extension of the theorem. For a layer ℓ, let client i perform H_i local steps with learning rate $\eta_{i\ell}$ and reset non-Nesterov momentum $\mu \in [0, 1)$. With the sampled basis fixed during the local round, its lifted momentum and factors satisfy $b_{i,s,\ell} = \mu b_{i,s-1,\ell} + \mathcal{S}_{i\ell}\widehat{g}_{i,s,\ell}$ and $X_{i,s+1,\ell} = X_{i,s,\ell} - \eta_{i\ell} b_{i,s,\ell}$. Here $\mathcal{S}_{i\ell}$ is the probability-corrected operator from Appendix C.3, and the initial momentum is $b_{i,-1,\ell} = 0$.

Set $c_{i,u} = \sum_{s=u}^{H_i - 1} \mu^{s-u}$, $\tau_{i\ell} = \eta_{i\ell} \sum_u c_{i,u}$, and $\omega_{i,u} = c_{i,u} / \sum_v c_{i,v}$. Let $g_{i\ell} = \nabla_\ell f_i(X_t)$ be the full local factor gradient at the initial global model. The negative lifted endpoint displacement equals

$$q_{i\ell} = \tau_{i\ell}\left(\mathcal{S}_{i\ell} g_{i\ell} + e_{i\ell}^{\mathrm{loc}}\right), \tag{49}$$

where $e_{i\ell}^{\mathrm{loc}} = \sum_u \omega_{i,u} \mathcal{S}_{i\ell}(\widehat{g}_{i,u,\ell} - g_{i\ell})$. This is an identity, including local stochastic noise and drift. No second sampling correction is used at the server. With $\kappa_\ell = \sum_i p_i \tau_{i\ell}$ and $\alpha_{i\ell} = p_i \tau_{i\ell}/\kappa_\ell$, the normalized layer endpoint satisfies

$$z_\ell - g_\ell = \sum_i \alpha_{i\ell}(\mathcal{S}_{i\ell} - I) g_{i\ell} + \sum_i (\alpha_{i\ell} - p_i) g_{i\ell} + \sum_i \alpha_{i\ell} e_{i\ell}^{\mathrm{loc}}. \tag{50}$$

Here $g_\ell = \sum_i p_i g_{i\ell}$. The terms correspond to random direction sampling, unequal effective workloads, and local training. Conditional on the completed bases, the first term has zero mean before local training but generally has nonzero variance. The workload term does not cancel when the $\alpha_{i\ell}$ differ from p_i, even if the sampling is unbiased. Thus unequal local work or reset momentum does not automatically inherit the vanishing bound of Theorem 5.1.

C.6 NONCONVEX CONVERGENCE

This section proves Theorem 5.1 for the factor objective J in Eq. (2). It treats all adapted factor blocks with the squared sum of their Frobenius norms. The exact forward identity in Appendix A.2 ensures that gradients of the folded client layer agree with those of its lifted full-size factors at every local iterate.

Let $\mathcal{H}_t$ include X_t, the previous update reference, and all randomness preceding this round's preferred-basis construction. Write $\mathbb{E}_t = \mathbb{E}[\cdot \mid \mathcal{H}_t]$, including the initial gradient estimates used for preferred-basis construction, direction samples, and training batches. The following assumptions hold uniformly over the horizons and step sizes considered.

Assumption C.1 (Smoothness and lower bound)**.** *Each client objective f_i has an L-Lipschitz factor gradient on the local trajectories. The global objective J is L-smooth along the server update segments and satisfies $J \geq J_*$.*

Assumption C.2 (Stochastic gradients)**.** *Let $\widehat{g}_{i,t}^h$ be the full factor gradient at the lifted local model on a fresh training batch. Conditional on the sampled basis and local history, its expectation is $\nabla f_i(X_{i,t}^h)$ and its second moment is at most G^2. Training batches are independent of the batch used for initial gradient estimation.*

Assumption C.3 (Direction sampling)**.** *Conditional on the complete orthonormal basis $V_{i,t}$, each client draws a fresh active subset each round according to Appendix C.3. Every direction has inclusion probability $\pi_{ij} \geq \pi_{\min} > 0$, where $\pi_{\min}$ is independent of t and T. The correction uses these exact marginal probabilities.*

All clients participate with fixed weights p_i, a common number $K \geq 1$ of plain SGD steps, and step size $\gamma > 0$. The server uses a scalar step size. We write $M = 1/\pi_{\min}$. Training batches drawn independently with replacement satisfy the conditional sampling requirement, whereas arbitrary random reshuffling need not. The complete basis may depend on local data or cached information because the correction identity is conditional on that basis. No bound on client gradient dissimilarity is needed, but bounded second moments and the probability floor remain explicit assumptions. Local momentum and unequal workloads are treated separately in Appendix C.5.

Local updates. For a sampled basis define $\mathcal{S}_{i,t}(G_B, G_A) = (G_B \widehat{P}_{i,t}, \widehat{P}_{i,t} G_A)$ with $\widehat{P}_{i,t} = Q_{i,t} D_{i,t} Q_{i,t}^\top$. Then $\|\mathcal{S}_{i,t}\| \leq M$. By the chain rule and the exact folded representation, the lifted local factors obey

$$X_{i,t}^{h+1} = X_{i,t}^h - \gamma \mathcal{S}_{i,t} \widehat{g}_{i,t}^h. \tag{51}$$

All local trajectories start at $X_{i,t}^0 = X_t$. The normalized average endpoint is therefore

$$z_t = \sum_i \frac{p_i}{K} \sum_{h=0}^{K-1} \mathcal{S}_{i,t} \widehat{g}_{i,t}^h. \tag{52}$$

Condition first on the constructed complete orthonormal bases. Equation (43) and the tower property give

$$\mathbb{E}_t \sum_i p_i \mathcal{S}_{i,t} \nabla f_i(X_t) = \sum_i p_i \nabla f_i(X_t) = g_t. \tag{53}$$

This anchor identity remains valid for bases constructed from the initial gradient estimates because subsequent active-set sampling is conditional on those bases. It is an expectation statement, not a claim that every sampled direction agrees with g_t.

Minkowski's inequality and the uniform second-moment bound give $(\mathbb{E}_t \left\| X_{i,t}^h - X_t \right\|^2)^{1/2} \leq \gamma h M G$. Conditional unbiasedness of training batches removes their noise in the mean of Eq. (52). Applying client smoothness, $\|\mathcal{S}_{i,t}\| \leq M$, and $K^{-1} \sum_{h=0}^{K-1} h = (K-1)/2$ gives

$$\|\mathbb{E}_t z_t - g_t\| \leq d_\gamma := \frac{L\gamma(K-1)M^2 G}{2}. \tag{54}$$

Weighted Jensen also gives $\mathbb{E}_t \|z_t\|^2 \leq M^2 G^2$. For $K = 1$ the mean local-drift term vanishes exactly.

Server filtering. The server constructs F_t from the actual sampled projectors. Define the factor-pair operator $\mathcal{F}_t(G_B, G_A) = (G_B F_t, F_t G_A)$. It satisfies $\|\mathcal{F}_t\| \le 1$ and $\|\mathcal{F}_t - I\| \le \rho$. The random F_t can depend on z_t through the sampled spaces; no independence is needed. Equation (15) gives $X_{t+1} = X_t - \eta u_t$, with $u_t = \mathcal{F}_t z_t$ and $\eta = \eta_s \gamma K$. Letting $b_t = \mathbb{E}_t u_t - g_t$ and adding and subtracting $\mathbb{E}_t z_t$ yields

$$\|b_t\| \le d_\gamma + \rho M G. \tag{55}$$

Indeed $b_t = (\mathbb{E}_t z_t - g_t) + \mathbb{E}_t[(\mathcal{F}_t - I) z_t]$, so Eq. (55) follows from the pointwise filter norm bound and $\mathbb{E}_t \|z_t\|^2 \le M^2 G^2$. Similarly $\mathbb{E}_t \|u_t\|^2 \le M^2 G^2$. A fixed $\rho > 0$ leaves a nonvanishing term in this sufficient bound. Scheduling $\rho \le \lambda \gamma K$ makes its contribution vanish along with the local learning rate.

Descent and rate. Let $\mu_t = \mathbb{E}_t u_t = g_t + b_t$, $v_t = \mathbb{E}_t \|u_t - \mu_t\|^2 \le M^2 G^2$, and $\eta = \eta_s \gamma K$. Smoothness and conditional variance decomposition give $\mathbb{E}_t J(X_{t+1}) \le J(X_t) - \eta \langle g_t, \mu_t \rangle + L\eta^2(\|\mu_t\|^2 + v_t)/2$. Using $2\langle g_t, \mu_t \rangle = \|g_t\|^2 + \|\mu_t\|^2 - \|b_t\|^2$ and $\eta \le 1/L$ yields

$$\mathbb{E}_t J(X_{t+1}) \le J(X_t) - \frac{\eta}{2}\|g_t\|^2 + \frac{\eta}{2}(d_\gamma + \rho M G)^2 + \frac{L\eta^2 M^2 G^2}{2}. \tag{56}$$

The additional term $-\eta(1 - L\eta)\|\mu_t\|^2/2$ is nonpositive under the step-size condition. Take expectations, sum over $t < T$, and use $\mathbb{E}J(X_T) \ge J_*$ to obtain Eq. (16). Finally fix $K, \eta_s, a, \lambda, \rho_{\max}$ independently of T, choose $\gamma = a/\sqrt{T}$ with $\eta_s a K \le 1/L$, and set $\rho = \min\{\rho_{\max}, \lambda\gamma K\}$. Substitution proves Eq. (17). The guarantee concerns the average expected squared factor-gradient norm. It does not establish a last-iterate or global-optimality guarantee. □

Proposition 4.1 optimizes the preferred subspace's estimated local energy. Equation (44) shows how preferred orientation changes the variance of the corrected initial update. The theorem's coarse constant $M^2 G^2$ does not certify a strict improvement from coordination. Proposition C.1 concerns an uncorrected projected initial gradient and the historical reference, so it does not imply that the corrected expected initial update surpasses another unbiased method. Proposition C.2 provides a conditional test of the tradeoff between bias and variance for the filter, not uniform dominance. The exploration rate, smallest inclusion probability, corrected-gradient norms, coverage strength, local steps, and optimizer settings all affect the size of the error terms. A small exploration probability may amplify rare-direction gradients, while a fixed positive filter strength is outside the vanishing-rate schedule.