FLoSER: Spectral Experts and Response-Aware Routing for Federated Sparse LoRA Adaptation

Anonymous authors
Paper under double-blind review

Abstract

Federated fine-tuning adapts large language models to distributed private data, but training and communicating task-specific parameters remain costly for resource-constrained clients. A recent resource-efficient paradigm reuses pretrained LoRA adapters as sparse rank-wise experts by freezing their rank-one components and training and aggregating only a lightweight token router. Although this enables token-level composition with few trainable and communicated parameters, it assumes that LoRA factor coordinates define stable expert identities. We show that equivalent factorizations preserve the same dense update but can induce different rank-one expert banks and sparse-routing families. Moreover, a stable expert bank alone does not ensure effective token-wise selection. We therefore propose Federated LoRA with Spectral Experts and Response-Aware Routing (FLoSER). FLoSER constructs an ordered spectral expert bank from the complete LoRA update—rank-one canonical for distinct singular values and block-canonical for repeated values—and augments learned routing with a signed, token-dependent correction derived from each expert's instantaneous response magnitude. This correction provides a learnable salience signal without changing the active expert budget; during federation, the backbone and expert bank remain frozen, and only router parameters are optimized and communicated. Mechanism analyses show that spectral construction removes factor-coordinate dependence and admits better sparse solutions, although learned routing does not automatically recover them. Experimental results show that FLoSER-S improves matched Raw rank-wise routing by up to 16.40 percentage points, while full FLoSER yields additional setting-dependent gains. Additional transfer and cross-task studies delineate backbone- and scale-sensitivity boundaries, and grouped controls contextualize the benefit of fine-grained learned routing under the primary heterogeneous protocol.

1 Introduction

Federated learning adapts a shared model without centralizing private data (McMahan et al., 2017), but large-model fine-tuning remains expensive. Low-Rank Adaptation (LoRA) reduces training, storage, and communication by freezing the backbone and learning compact factors (Hu et al., 2022), enabling heterogeneity-aware federated adaptation (Cho et al., 2024; Liu et al., 2025). Pretrained LoRA collections offer a complementary opportunity: reuse existing adaptations instead of learning each target from scratch.

Recent work composes pretrained LoRAs at the adapter or instance level (Huang et al., 2023; Wang et al., 2024; Feng et al., 2024), while finer-grained approaches expose individual rank components as experts (Zhao et al., 2025; Li et al., 2025). SmartFed brings this idea to federated fine-tuning: retrieved adapters are decomposed into rank-one components, these components are frozen as a reusable expert bank, and a lightweight token-level router is trained and communicated (Wu et al., 2026). This design is resource-efficient, but it implicitly assumes that each factor-coordinate component defines a stable expert identity.

We show that this assumption need not hold. For $\Delta\mathbf{W} = \Gamma\mathbf{B}\mathbf{A}$ and any invertible matrix $\mathbf{Q}$, $(\mathbf{B}\mathbf{Q}, \mathbf{Q}^{-1}\mathbf{A})$ represents the same dense update, yet its rank-one terms generally differ from

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.

$\{\boldsymbol{b}_i\boldsymbol{a}_i^\top\}_{i=1}^R$. Once the terms receive persistent identities and only a Top-K subset is active, equivalent dense factorizations can induce different routing families. The ambiguity lies in sparse execution, not in the dense LoRA function. Unlike representation dependence during cross-client factor aggregation (Meng et al., 2026; Chen et al., 2026), this problem persists even when the complete pretrained update is fixed and never aggregated.

To address these two challenges, we propose Federated LoRA with Spectral Experts and Response-Aware Routing (FLoSER). FLoSER first derives an ordered spectral bank from the complete LoRA update rather than its arbitrary factor coordinates. Its rank-one experts are gauge-stable for distinct nonzero singular values; repeated values define invariant blocks rather than unique rank-one experts. FLoSER then augments learned routing with a signed correction based on each expert's token-dependent response magnitude. The two components separate expert construction from expert selection while preserving the resource-efficient federated protocol: the backbone and expert bank remain frozen, and only the lightweight router is optimized and communicated.

Our contributions are threefold:

- We formalize the coordinate dependence created when LoRA factor components are materialized as independently routable experts, and show that dense equivalence does not imply sparse-routing equivalence.
- We introduce FLoSER, combining a function-defined spectral expert bank with Response-Aware routing as a complementary, setting-dependent selection refinement that does not increase the active expert budget.
- Matched GSM8K evaluations show a positive centralized FLoSER-S gain and consistent gains under IID and strong Non-IID federation. Comparisons with frozen reuse, whole-adapter routing, and FedAvg-LoRA support fine-grained routing, while fixed-quota, second-backbone, and MBPP studies delineate allocation, transfer, and scale-sensitivity boundaries.

2 Related Work

Federated parameter-efficient adaptation. Federated averaging enables decentralized optimization while limiting the communication of local training data (McMahan et al., 2017), and LoRA makes large-model adaptation practical by training compact low-rank updates (Hu et al., 2022). Their combination has motivated methods for heterogeneous client ranks and communication-efficient federated fine-tuning (Cho et al., 2024; Liu et al., 2025). These approaches primarily optimize client-side adapters and study how their trainable states should be communicated or aggregated. In contrast, our expert bank consists of frozen pretrained LoRAs; only the sparse router is trained and communicated.

LoRA composition and routing. Pretrained LoRAs can be combined at the task level (Huang et al., 2023), selected per instance (Wang et al., 2024), or treated as routed module-level experts (Feng et al., 2024). Finer-grained methods expose the rank dimension: SMoRA treats individual ranks as experts (Zhao et al., 2025), while LoRACoE dynamically constructs experts through rank-level parameter composition (Li et al., 2025). Most directly, SmartFed decomposes retrieved adapters into rank-one experts, trains a token-level router, and uses EEQA to allocate activation budgets in federated fine-tuning (Wu et al., 2026). We retain this resource-efficient setting but revisit the representation that defines an expert: granularity alone does not ensure that an expert identity is preserved under equivalent LoRA factorizations. Recent PMDRouter independently uses response magnitude for zero-shot selection among complete LoRA modules (Ji et al., 2026). Our response signal instead operates online over rank-one spectral experts and enters a learned router through a signed coefficient, reflecting that response magnitude measures salience rather than whether an expert is beneficial.

Spectral representations and low-rank alignment. Singular structure has previously been used to improve low-rank adaptation; for example, PiSSA initializes trainable adapters from principal components of the pretrained weight matrix (Meng et al., 2024). A separate line of federated LoRA work addresses representation dependence during client-update aggregation. FLoRG uses low-rank Gram matrices and Procrustes alignment, while GLoRA constructs a gauge-aware server

representation for combining equivalent client factorizations (Meng et al., 2026; Chen et al., 2026). Our construction has a different object and purpose: we decompose the complete, already-trained LoRA update rather than the pretrained backbone weight, and use its spectral components to define factorization-independent candidates for sparse routing. The issue is therefore neither adapter initialization nor cross-client factor aggregation, but the expert identities and routing family induced by sparse execution.

3 METHOD

We study SmartFed-style federated rank-wise expert assembly, which freezes rank-one components of pretrained LoRAs as candidate experts and communicates only a token router (Wu et al., 2026). We separate two questions conflated by raw routing: *what defines an expert* and *which experts should a token activate*. Spectral construction addresses the first; response-aware routing addresses the second. We denote the spectral-only variant by *FLoSER-S*; *FLoSER* additionally incorporates Response-Aware routing. The construction is applied independently to each retrieved adapter and routed weight matrix, after which the resulting rank-one candidates are concatenated into a shared M-expert bank for that matrix.

Framework overview. Figure 1 follows the full FLoSER pipeline. Equivalent LoRA factors can preserve the dense update while changing the raw sparse choices (a). FLoSER therefore constructs routing candidates from the spectral structure of the complete update (b), then combines learned router logits with a signed token-dependent response correction before Top-K selection (c). During federated optimization, clients train only these router parameters and the server aggregates them with FedAvg; the backbone and expert banks remain frozen (d). Thus, FLoSER changes expert identity and the selection score, but not the active expert budget or the communicated parameter class.

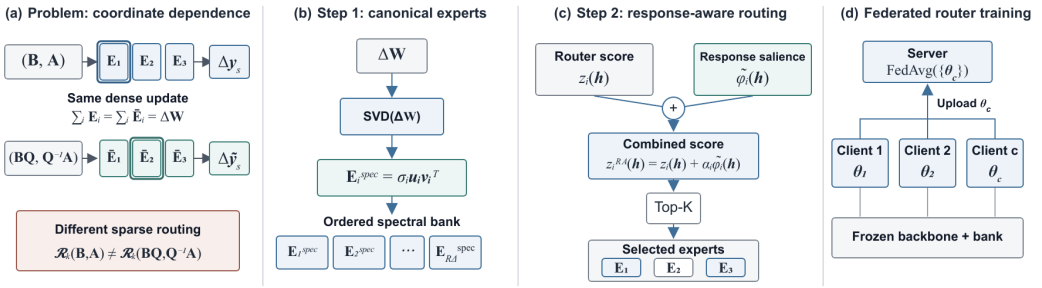


Figure 1: FLoSER overview. (a) Equivalent LoRA factorizations preserve the dense update but can induce different sparse routing families. (b) Spectral decomposition defines an ordered bank from the complete update; rank-one canonicality requires a distinct spectrum, while repeated blocks are canonical only as subspaces. (c) Response-Aware routing combines learned scores with token-dependent response salience before Top-K selection. (d) The backbone and expert bank remain frozen in federation; only router parameters are trained and communicated.

Throughout, bold upright uppercase symbols denote matrices, bold lowercase symbols denote vectors, calligraphic symbols denote sets, uppercase unbolded symbols denote fixed scalar or integer constants, and lowercase italic symbols denote scalar variables and indices.

3.1 COORDINATE DEPENDENCE OF RANK-WISE ROUTING

For a frozen weight matrix, LoRA represents its adaptation as $\Delta\mathbf{W} = \Gamma\mathbf{B}\mathbf{A}$, where $\mathbf{B} \in \mathbb{R}^{D_{\text{out}} \times R}$, $\mathbf{A} \in \mathbb{R}^{R \times D_{\text{in}}}$, R is the fixed LoRA rank, and Γ is the fixed LoRA scale. Denote column i of $\mathbf{B}$ by $\boldsymbol{b}_i$ and row i of $\mathbf{A}$ by $\boldsymbol{a}_i^\top$. Raw routing treats

$$\mathbf{E}_i^{\text{raw}} = \Gamma \boldsymbol{b}_i \boldsymbol{a}_i^\top, \qquad \Delta\mathbf{W} = \sum_{i=1}^{R} \mathbf{E}_i^{\text{raw}}, \tag{1}$$

as candidate experts. Given hidden state $\boldsymbol{h}$, a router selects $\mathcal{S}(\boldsymbol{h})$ with $|\mathcal{S}(\boldsymbol{h})| \le K$ and returns $\Delta\boldsymbol{y}(\boldsymbol{h}) = \sum_{i\in\mathcal{S}(\boldsymbol{h})} \mathbf{E}_i^{\mathrm{raw}}\boldsymbol{h}$.

This construction turns arbitrary factor coordinates into expert identities. Indeed, for any invertible $\mathbf{Q} \in \mathbb{R}^{R\times R}$,

$$\mathbf{BA} = (\mathbf{BQ})(\mathbf{Q}^{-1}\mathbf{A}), \tag{2}$$

so the dense LoRA update is unchanged, whereas the induced experts $\widetilde{\mathbf{E}}_i = \Gamma\widetilde{\boldsymbol{b}}_i\widetilde{\boldsymbol{a}}_i^\top$ generally differ, where the transformed vectors are columns and rows of $\mathbf{BQ}$ and $\mathbf{Q}^{-1}\mathbf{A}$. Let $\mathcal{I}_R = \{1,\dots,R\}$ and define the updates realizable with at most K active experts as

$$\mathcal{R}_K(\mathbf{B},\mathbf{A}) = \left\{ \Gamma\sum_{i\in\mathcal{S}} \boldsymbol{b}_i\boldsymbol{a}_i^\top : \mathcal{S}\subseteq\mathcal{I}_R,\ |\mathcal{S}|\le K \right\}. \tag{3}$$

We use $\mathcal{R}_K$ as a minimal candidate-family characterization; the $K=1$ counterexample below also applies directly to the selected-softmax execution used in our experiments. Although the dense update is invariant, the induced sparse routing family need not be invariant: for $K<R$, one can have $\mathcal{R}_K(\mathbf{B},\mathbf{A}) \ne \mathcal{R}_K(\mathbf{BQ},\mathbf{Q}^{-1}\mathbf{A})$. For a minimal example, let $\mathbf{B}=\mathbf{A}=\mathbf{I}_2$, $K=1$, and $\mathbf{Q}=\left[\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right]$. Writing $\boldsymbol{e}_i$ for the i-th standard basis vector, the original bank is $\{\boldsymbol{e}_1\boldsymbol{e}_1^\top, \boldsymbol{e}_2\boldsymbol{e}_2^\top\}$, whereas the equivalent factors induce $\{\boldsymbol{e}_1(\boldsymbol{e}_1-\boldsymbol{e}_2)^\top, (\boldsymbol{e}_1+\boldsymbol{e}_2)\boldsymbol{e}_2^\top\}$. Both banks sum to $\mathbf{I}_2$, but their one-sparse routing families differ. Thus, factor freedom that is harmless under dense execution becomes functionally consequential once coordinate components are instantiated as independently routable experts.

3.2 Spectral expert bank

We instead construct experts from the functionally defined update $\Delta\mathbf{W}$. For each adapted matrix, let

$$\Delta\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \sum_{i=1}^{R_\Delta} \sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^\top, \qquad \sigma_1 \ge \cdots \ge \sigma_{R_\Delta} > 0, \tag{4}$$

where $R_\Delta = \mathrm{rank}(\Delta\mathbf{W})$. We define

$$\mathbf{E}_i^{\mathrm{spec}} = \sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^\top. \tag{5}$$

When the nonzero singular values are distinct, each rank-one expert is invariant to LoRA reparameterization: the paired sign ambiguity of the singular vectors cancels in their outer product. For a repeated singular value, however, only the associated singular subspaces and their block update are invariant; the individual rank-one terms, and hence Top-K choices that split the block, remain basis-dependent. We use a deterministic SVD basis only as an implementation convention, not as a resolution of this mathematical degeneracy. Subject to this scope, the experts are ordered by spectral strength, mutually Frobenius-orthogonal, and their first k terms optimally approximate $\Delta\mathbf{W}$ at rank k.

The bank can be constructed without materializing the dense $D_{\mathrm{out}}\times D_{\mathrm{in}}$ update. Using thin factorizations $\mathbf{B}=\mathbf{Q}_B\mathbf{R}_B$ and $\mathbf{A}^\top=\mathbf{Q}_A\mathbf{R}_A$, we decompose only the $R\times R$ core $\Gamma\mathbf{R}_B\mathbf{R}_A^\top$ and lift its singular vectors through $\mathbf{Q}_B$ and $\mathbf{Q}_A$. This one-time offline step costs $\mathcal{O}((D_{\mathrm{out}}+D_{\mathrm{in}})R^2+R^3)$ time and does not require storing $\Delta\mathbf{W}$. Appendix A gives the derivation and compact construction.

3.3 Response-aware spectral routing

Canonicalization determines the candidates but not their token-wise relevance. A spectral expert responds to $\boldsymbol{h}$ as

$$\mathbf{E}_i^{\mathrm{spec}}\boldsymbol{h} = \sigma_i\boldsymbol{u}_i(\boldsymbol{v}_i^\top\boldsymbol{h}), \qquad \rho_i(\boldsymbol{h}) = \|\mathbf{E}_i^{\mathrm{spec}}\boldsymbol{h}\|_2 = \sigma_i|\boldsymbol{v}_i^\top\boldsymbol{h}|. \tag{6}$$

Here $\rho_i(\boldsymbol{h})$ measures token-dependent influence magnitude, or *salience*; it does not determine whether the influence benefits the task. We therefore use it to complement, rather than replace, the learned router.

Let $\phi_i(\boldsymbol{h}) = \log(1+\rho_i(\boldsymbol{h}))$ and standardize it within the M-expert candidate bank:

$$\widetilde{\phi}_i(\boldsymbol{h}) = \frac{\phi_i(\boldsymbol{h})-\mu_\phi(\boldsymbol{h})}{\sqrt{M^{-1}\sum_{j=1}^M(\phi_j(\boldsymbol{h})-\mu_\phi(\boldsymbol{h}))^2 + C_0}}, \quad \mu_\phi(\boldsymbol{h}) = M^{-1}\sum_{j=1}^M \phi_j(\boldsymbol{h}). \tag{7}$$

For baseline score $z_i(\boldsymbol{h}) = \boldsymbol{w}_i^\top \widehat{\boldsymbol{h}} + b_i$, our response-aware score is

$$z_i^{\mathrm{RA}}(\boldsymbol{h}) = z_i(\boldsymbol{h}) + \alpha_i \widetilde{\phi}_i(\boldsymbol{h}), \tag{8}$$

where $C_0 = 10^{-6}$ and the router input uses RMS normalization, $\widehat{\boldsymbol{h}} = \boldsymbol{h}/\sqrt{D_{\mathrm{in}}^{-1}\|\boldsymbol{h}\|_2^2 + C_0}$. The signed, expert-specific coefficient α_i learns whether the token-dependent response feature $\widetilde{\phi}_i(\boldsymbol{h})$ should encourage or suppress expert i. We initialize $\alpha_i = 0$, so the model initially recovers the spectral baseline exactly, and select $\mathcal{S}(\boldsymbol{h}) = \mathrm{TopK}_i\, z_i^{\mathrm{RA}}(\boldsymbol{h})$. The method leaves the candidate count and active budget unchanged. Operationally, the router computes all M corrected scores but executes only the selected experts. The correction adds one learned scalar per expert and $\mathcal{O}(MD_{\mathrm{in}})$ response-score work per token; it introduces no additional expert execution. Thus FLoSER preserves the active expert count, but we do not claim identical FLOPs, latency, or memory use relative to FLoSER-S.

3.4 Federated router optimization

In the federated setting, the backbone, pretrained LoRA updates, and resulting expert banks remain shared and frozen; clients optimize only router parameters $\boldsymbol{\theta} = (\{\boldsymbol{w}_i, b_i\}_i, \{\alpha_i\}_i)$, with the spectral baseline omitting α_i. At round t, each selected client $c \in \mathcal{C}_t$ initializes from $\boldsymbol{\theta}^t$ and locally minimizes its task loss to obtain $\boldsymbol{\theta}_c^{t+1}$. The server applies sample-weighted aggregation,

$$\boldsymbol{\theta}^{t+1} = \sum_{c \in \mathcal{C}_t} \frac{n_c}{\sum_{j \in \mathcal{C}_t} n_j} \boldsymbol{\theta}_c^{t+1}, \tag{9}$$

where n_c is the local sample count. This keeps the aggregation rule and communicated parameter class fixed; our experiments additionally match the total sparse activation budget. Matrix-specific quotas may nevertheless differ under adaptive allocation, as detailed in Appendix E.

4 Experiments

Our evaluation follows four axes: representation and routing mechanisms; downstream behavior across backbone–adapter banks; dependence on SmartFed's federated quota allocation; and cross-task inference stability. We additionally compare representative reuse and adaptation alternatives to test whether simpler strategies explain the primary gain.

4.1 Experimental setup

Models, expert banks, and tasks. Primary GSM8K experiments freeze Qwen2.5-7B and form each bank from three rank-8 math, code, and chat LoRAs. We route every layer's query and value projections ($\mathbf{W}_q, \mathbf{W}_v$), giving 24 candidates per matrix; only routing parameters are trained, and the main metric is greedy exact-answer accuracy. A second-backbone GSM8K audit uses Mistral-7B-Instruct-v0.2 and three rank-16 adapters. Because its bank and activation ratio also differ, this is a backbone–adapter transfer boundary rather than a backbone-only intervention (Appendix F). For cross-task evaluation, MBPP reuses the Qwen backbone and 24-candidate bank, trains routing parameters on code generation, and reports Pass@1 together with termination and syntax diagnostics; inference-gain calibration is analyzed in Section 4.6. Validation loss is diagnostic only.

Controls and evaluation protocol. Paired comparisons match model, bank, split, router initialization, training and activation budgets, questions, and runtime. We use *Raw* for the factor-coordinate baseline, *FLoSER-S* for the spectral-only variant, and *FLoSER* for the full Response-Aware method. FLoSER-S and FLoSER share the same bank and differ only by Eq. 8. Development and fresh evaluation subsets are disjoint. We report three router seeds and paired uncertainty where available; seed labels align only within each stated comparison, not across independently trained studies.

Federated protocol. We use 20 clients, two per round, for 20 rounds and 10 local steps. FedAvg communicates only routers. Primary comparisons use Elastic Expert Quota Allocation (EEQA) following SmartFed (Wu et al., 2026); it preserves the total quota while recomputing per-matrix quotas. Under strong Non-IID we also disable EEQA and fix each matrix's quota to 12. We test IID and difficulty-skewed Dirichlet-0.1 partitions; full details are in Appendix B.

4.2 MECHANISM DIAGNOSTICS

Before task accuracy, Figure 2 tests whether spectral experts are stable and compact, and whether a learned router exploits the improved bank.

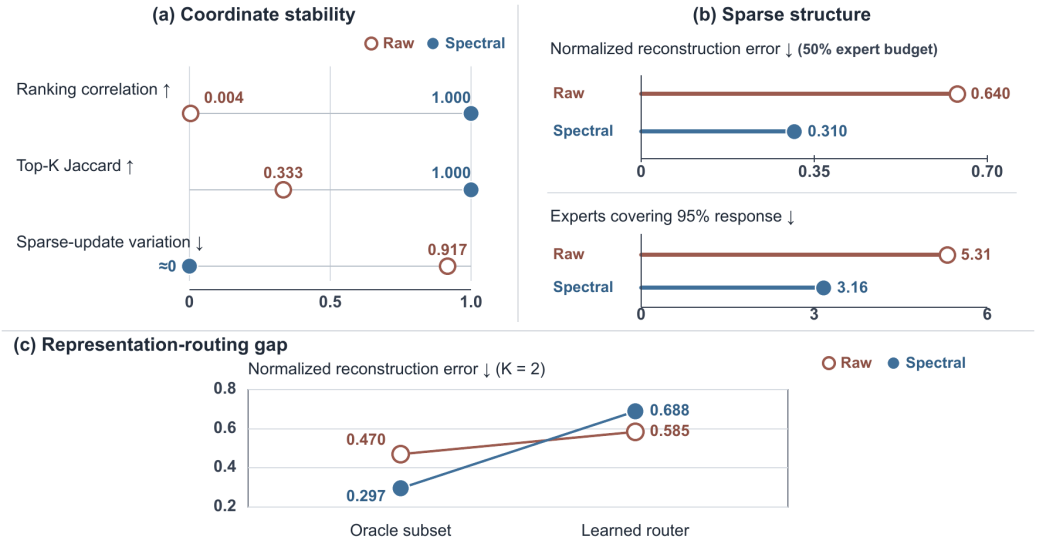


Figure 2: Mechanism diagnostics. (a) Under equivalent LoRA factor coordinates, Raw rankings, sparse selections, and routed updates vary, whereas spectral experts remain stable. (b) Spectral experts reduce sparse reconstruction error and concentrate 95% of the response in fewer experts. (c) At $K = 2$, the spectral bank admits a better oracle subset, but a learned hard Top-K router does not recover this advantage, exposing the representation–routing gap. Preference directions are marked in each panel.

Spectral construction stabilizes the expert bank. Holding $\Delta\mathbf{W}$ fixed, we randomly rotate its factors. Raw rankings, selections, and routed updates change substantially, whereas spectral experts remain stable (Figure 2(a–b)). Spectral coordinates also improve 50%-budget reconstruction and concentrate 95% of the response in fewer directions. A cross-source audit gives Raw Top-K variation 0.8918 (95% CI $[0.8380, 0.9379]$). Thus, for non-degenerate directions, spectral construction removes factor-coordinate dependence and yields a more compact bank; repeated or near-degenerate blocks are interpreted only at the subspace level in Appendix C.

A better representation is not sufficient. For frozen states, we exhaustively find the size-K subset best reconstructing the full response. At $K = 2$, the spectral oracle lowers error from 0.4695 to 0.2967, yet learned-router error rises from 0.5847 to 0.6876 (Figure 2(c)); other budgets agree (Appendix D). Spectral construction improves available sparse solutions but not the router's ability to identify them, motivating a token-dependent signal.

Response magnitude measures salience, not utility. We compare Eq. 6 with first-order expert utility. For all (answer) tokens, response correlates 0.0204 (0.0241) with signed utility but 0.3732 (0.4587) with its magnitude; Top-K overlap is 47.31% (47.84%). Response is therefore salience, not a standalone routing criterion, motivating a signed correction. Appendix D gives the full audit.

4.3 CENTRALIZED EVALUATION ACROSS BACKBONES

We first ask whether the mechanism gains produce a centralized downstream signal across backbone–adapter banks (Table 1).

On Qwen, FLoSER-S fixes 31 of 128 paired errors and breaks 10 (95% CI $[+7.0, +25.8]$; McNemar $p = 0.0015$). FLoSER is positive across three seeds on a disjoint study, but its hierarchical CI crosses

Table 1: Centralized GSM8K exact-answer accuracy (%). Each cell is baseline → proposed (Δ). The two Qwen cells are independent matched studies; Mistral reports means from three-way matched runs over three router seeds, with seed-wise results and SD in Appendix F. Bold marks the higher mean endpoint, not a statistically established cross-backbone ranking.

Backbone	Protocol	Raw → FLoSER-S	FLoSER-S → FLoSER
Qwen2.5-7B	Matched studies	$14.10 \rightarrow \mathbf{30.50}\ (+16.40)$	$26.04 \rightarrow \mathbf{29.43}\ (+3.39)$
Mistral-7B	3-seed matched	$31.77 \rightarrow \mathbf{32.55}\ (+0.78)$	$\mathbf{32.55} \rightarrow 31.77\ (-0.78)$

zero. On Mistral, FLoSER-S has a $+0.78$-point aggregate mean shift, but all three method means differ by less than one point and the seed-wise ordering changes. The second bank therefore provides a transfer boundary rather than a robust centralized gain. The Qwen Raw–FLoSER-S analysis is reported in Table 13, its fresh FLoSER-S–FLoSER comparison in Table 19, and the Mistral seed-wise results in Table 29.

4.4 Federated evaluation across backbones

We next ask whether the downstream signal persists under federation and a second backbone–adapter bank. Each row follows its prespecified protocol and is interpreted within-bank, not as a backbone-only intervention.

Table 2: Three-seed federated GSM8K accuracy (%) under each backbone–adapter bank's prespecified protocol. Each cell reports baseline → proposed (Δ). Qwen uses EEQA and separate matched comparisons; Mistral uses fixed quota 12 and three-way matched runs. Bold marks the higher endpoint within each comparison.

Backbone–bank	Partition	Raw → FLoSER-S	FLoSER-S → FLoSER
Qwen2.5-7B	IID	$23.18 \rightarrow \mathbf{38.80}\ (+15.62)$	$37.89 \rightarrow \mathbf{40.36}\ (+2.47)$
	Strong Non-IID	$27.21 \rightarrow \mathbf{33.72}\ (+6.51)$	$38.28 \rightarrow 38.28\ (0.00)$
Mistral-7B	IID	$28.78 \rightarrow \mathbf{29.82}\ (+1.04)$	$\mathbf{29.82} \rightarrow 29.04\ (-0.78)$
	Strong Non-IID	$27.99 \rightarrow \mathbf{28.26}\ (+0.26)$	$28.26 \rightarrow \mathbf{28.39}\ (+0.13)$

FLoSER-S improves Raw in every Qwen seed under both partitions, whereas the Mistral effects are small and directionally inconsistent. Thus the strong Qwen signal transfers only weakly to the second bank, and the FLoSER refinement remains setting-dependent.

Controlling for SmartFed's quota allocation. SmartFed's full pipeline includes EEQA, which derives per-matrix quotas from expert-importance statistics. Replacing Raw components with spectral experts can change these statistics and therefore confound expert construction with an induced allocation change. To test whether the spectral gain persists without adaptive allocation, we disable EEQA and fix every routed matrix's quota at 12 throughout training and inference.

Table 3: Strong Non-IID Qwen accuracy (%), averaged over three router seeds. Each cell reports baseline → proposed (Δ). The EEQA cells are independent matched comparisons and do not form a three-way ranking; the fixed-quota row is a matched three-way study. Bold marks the higher endpoint.

Backbone–bank	Allocation rule	Raw → FLoSER-S	FLoSER-S → FLoSER
Qwen2.5-7B	EEQA	$27.21 \rightarrow \mathbf{33.72}\ (+6.51)$	$38.28 \rightarrow 38.28\ (0.00)$
	Fixed quota 12	$29.04 \rightarrow \mathbf{34.38}\ (+5.34)$	$34.38 \rightarrow \mathbf{36.85}\ (+2.47)$

Fixed quotas preserve a three-seed FLoSER-S improvement, showing that its gain does not require EEQA in this setting. The FLoSER increment changes with the allocation rule, but both intervals cross zero; we therefore treat this only as suggestive evidence of an allocation interaction. Complete CIs, FedAvg geometry, and seed-wise audits are in Appendices E and F.

4.5 COMPARISON WITH REPRESENTATIVE ALTERNATIVES

Baseline design. We compare against the following representative alternatives:

- **Base model:** the frozen backbone without a retrieved adapter.
- **Best single (Code):** the strongest individual frozen adapter on the shared evaluation set.
- **Uniform merge:** an equal-weight merge of all three frozen adapters, without learned routing.
- **Static Top-σ:** the 12 highest-singular-value spectral directions, testing spectral truncation without a learned router.
- **Whole-Adapter Top-2:** our learned adapter-level granularity control, motivated by prior composition and routing work (Huang et al., 2023; Wang et al., 2024; Feng et al., 2024); it is not an official reproduction of those methods.
- **FedAvg-LoRA:** a rank-12 LoRA trained from scratch by combining FedAvg (McMahan et al., 2017) with LoRA (Hu et al., 2022), without a pretrained expert bank.
- **Raw (SmartFed-style):** our primary learned rank-wise baseline following SmartFed's routing paradigm (Wu et al., 2026); it directly routes LoRA factor-coordinate components under the same federated protocol.

All rows use the same evaluation set but differ in pretrained information, trained objects, and active computation; Table 4 is therefore a grouped diagnostic rather than an equal-budget ranking.

Table 4: Representative GSM8K alternatives under strong heterogeneity. Trained results are three-seed means $\pm$ sample SD where shown; frozen references use one fixed predictor. Active rank is an upper bound per adapted matrix and does not imply matched FLOPs, pretrained information, or trainable parameters. Full variants are in Appendix E.6.

Group	Method	Fed. update	Active rank	Acc. (%)
Frozen ref.	Base model	None	0	5.47
	Best single (Code)	None	8	12.89
	Uniform merge	None	≤ 24	5.86
	Static Top-σ	None	12	11.72
Trained	Whole-Adapter Top-2	Router	≤ 16	12.76 ± 0.60
	FedAvg-LoRA, rank 12	LoRA factors	12	9.24 ± 4.73
Rank-wise	Raw	Router	12	29.04
	FLoSER-S	Router	12	34.38
	FLoSER	Router	12	**36.85**

All frozen-reuse, whole-adapter, and from-scratch controls trail Raw, while FLoSER-S and FLoSER further improve the matched rank-wise baseline. Subject to the unequal-budget caveat above, these controls contextualize the benefit of fine-grained learned routing; the matched Raw–FLoSER-S comparison isolates the gain associated with spectral expert construction.

4.6 FEDERATED CROSS-TASK STABILITY ON MBPP

Finally, we ask whether the method remains stable after moving to code generation. Without retraining, each MBPP-trained IID federated checkpoint is evaluated at $\Lambda \in \{1, 0.5\}$ using identical data, quotas, and decoding; all routers were trained with $\Lambda = 1$. The full three-seed gain matrix and generation diagnostics are in Appendix G.

Table 5: MBPP scale-sensitivity summary. Pass@1 is the three-seed mean on 90 shared tasks; Δ is the change from $\Lambda = 1$ to 0.5. The same checkpoint is reused at both gains. Bold marks the better gain per method.

Method	$\Lambda = 1$	$\Lambda = 0.5$	Δ	512-cap ($1 \to 0.5$)
Raw	**49.63**	41.48	-8.15	$0.0 \to 0.0$
FLoSER-S	21.85	**47.41**	$+25.56$	$41.1 \to 0.0$
FLoSER	20.37	**49.26**	$+28.89$	$40.4 \to 0.0$

Gain reduction harms Raw but recovers both spectral variants and removes their 512-token truncations. The interaction contrasts relative to Raw are $+33.70$ and $+37.04$ points, indicating representation-dependent rather than generic scaling. Yet at each method's better validation operating point, FLoSER matches rather than surpasses Raw (49.26% versus 49.63%); this is also not a pure representation effect because the routers and generation trajectories differ. We therefore treat MBPP as a scale-sensitivity boundary with calibration recovery, not as a main performance claim. Earlier uncalibrated failures and frozen centralized, IID, and code-length-skewed test checks remain reported in Appendix G.

5 Conclusion

We introduced FLoSER, a federated sparse LoRA framework that separates expert construction from token-wise selection. Sparse rank-wise routing assigns expert identities to arbitrary LoRA factor coordinates: equivalent dense updates can induce different sparse routing families. Its spectral-only variant, FLoSER-S, constructs a spectral expert bank from the complete update, with rank-one canonicality for distinct singular values and only block-level canonicality for repeated values; full FLoSER additionally refines selection through signed, token-dependent response corrections. Mechanism diagnostics and matched GSM8K experiments identify spectral construction as the larger, more consistent gain. It remains positive under IID and strong Non-IID federation, including fixed-quota runs; the adaptive–fixed contrast suggests an allocation interaction for Response-Aware. On the shared fresh GSM8K evaluation, frozen reuse, whole-adapter learned routing, and from-scratch FedAvg-LoRA all remain below rank-wise learned routing, although these groups are not strictly matched in information or computation. On Mistral, both centralized and federated three-seed comparisons show only small, seed-inconsistent differences, indicating that gains do not uniformly transfer across backbone–adapter banks. MBPP exposes a scale-sensitivity/stability boundary: uncalibrated spectral routing destabilizes generation, while calibration restores performance near Raw. The matched-gain results support a representation-dependent scaling interaction rather than a generic benefit from reducing gain.

Canonicality is subspace-level under repeated singular values; neither the quota contrast nor the MBPP gain–representation interaction contrast is a pure representation effect. Broader backbones, adapters, tasks, and client partitions remain to be tested. Subject to these limits, sparse-routing candidates should be defined by the represented function, not arbitrary factor coordinates.

AI Use Statement

In this work, we used generative AI tools to assist with conceptual framing, feedback on research methodology and experimental design, and the interpretation and presentation of experimental results. We additionally used these tools for literature discovery and organization, manuscript drafting and language editing, LaTeX formatting, and the preparation and refinement of figures and tables. We did not use generative AI tools to generate synthetic datasets or experimental results. All AI-assisted content was reviewed by the authors, who manually verified the cited literature, mathematical statements, experimental protocols, code, and reported results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Reproducibility Statement

We provide the information needed to reproduce the reported experiments. Section 3 specifies the proposed method, and the appendices document the derivations, implementation details, hyperparameters, data partitions, random seeds, checkpoint-selection rules, and complete results. An anonymized implementation, fixed experiment configurations, deterministic split audits, and reproduction commands are provided in the supplementary material.

References

Jinqian Chen, Chang Liu, and Jihua Zhu. Beyond factor aggregation: Gauge-aware low-rank server representations for federated LoRA. *arXiv preprint arXiv:2605.06733*, 2026. URL https://arxiv.org/abs/2605.06733.

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. doi: 10.18653/v1/2024.emnlp-main.717. URL https://aclanthology.org/2024.emnlp-main.717/.

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. Mixture-of-LoRAs: An efficient multitask tuning method for large language models. In *Proceedings of LREC-COLING 2024*, pp. 11371–11380, 2024. URL https://aclanthology.org/2024.lrec-main.994/.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. LoRAHub: Efficient cross-task generalization via dynamic LoRA composition. *arXiv preprint arXiv:2307.13269*, 2023. URL https://arxiv.org/abs/2307.13269.

Fengxian Ji, Zhuohan Xie, Jingpu Yang, Fan Zhang, Zirui Song, and Xiuying Chen. Parametric memory decoding for zero-shot routing in LoRA-based external parametric memory. *arXiv preprint arXiv:2607.04118*, 2026. URL https://arxiv.org/abs/2607.04118.

Guanyu Li, Zhiheng Xi, Zhihao Zhang, Boyang Hong, Tao Gui, Qi Zhang, and Xuanjing Huang. LoRACoE: Improving large language model via composition-based LoRA expert. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 31290–31304, 2025. doi: 10.18653/v1/2025.emnlp-main.1594. URL https://aclanthology.org/2025.emnlp-main.1594/.

Han Liu, Ruoyao Wen, Srijith Nair, Jia Liu, Wenjing Lou, Chongjie Zhang, William Yeoh, Yevgeniy Vorobeychik, and Ning Zhang. EcoLoRA: Communication-efficient federated fine-tuning of large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20732–20746, 2025. doi: 10.18653/v1/2025.emnlp-main.1046. URL https://aclanthology.org/2025.emnlp-main.1046/.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

Chuiyang Meng, Ming Tang, and Vincent W. S. Wong. FLoRG: Federated fine-tuning with low-rank gram matrices and procrustes alignment. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kntrZOm2AQ.

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, 2024. URL https://papers.nips.cc/paper/2024/hash/db36f4d603cc9e3a2a5e10b93e6428f2-Abstract-Conference.html.

Zhiqi Wang, Shizhu He, Kang Liu, and Jun Zhao. Instance-level dynamic LoRAs composition for cross-task generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 5699–5708, 2024. doi: 10.18653/v1/2024.findings-emnlp.326. URL https://aclanthology.org/2024.findings-emnlp.326/.

Yebo Wu, Jingguang Li, Zhijiang Guo, and Li Li. Don't reinvent the wheel, just realign the spokes: Resource-efficient federated fine-tuning via rank-wise expert assembly. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=HdsEUNo4C4.

Ziyu Zhao, Yixiao Zhou, Zhi Zhang, Didi Zhu, Tao Shen, Zexi Li, Jinluan Yang, Xuwu Wang, Jing Su, Kun Kuang, Zhongyu Wei, Fei Wu, and Yu Cheng. Each rank could be an expert: Single-ranked mixture of experts LoRA for multi-task learning. *arXiv preprint arXiv:2501.15103*, 2025. URL https://arxiv.org/abs/2501.15103.

A Additional method details

A.1 A minimal counterexample for sparse coordinate dependence

The example in Section 3.1 can be verified directly. Let $\mathbf{B} = \mathbf{A} = \mathbf{I}_2$, $K = 1$, and

$$\mathbf{Q} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{Q}^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}. \tag{10}$$

The original rank-one bank is $\{\boldsymbol{e}_1\boldsymbol{e}_1^\top, \boldsymbol{e}_2\boldsymbol{e}_2^\top\}$. The equivalent factors are

$$\mathbf{BQ} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{Q}^{-1}\mathbf{A} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \tag{11}$$

and therefore induce

$$\widetilde{\mathbf{E}}_1 = \boldsymbol{e}_1(\boldsymbol{e}_1 - \boldsymbol{e}_2)^\top, \qquad \widetilde{\mathbf{E}}_2 = (\boldsymbol{e}_1 + \boldsymbol{e}_2)\boldsymbol{e}_2^\top. \tag{12}$$

Both banks sum to $\mathbf{I}_2$, but neither transformed component equals $\boldsymbol{e}_1\boldsymbol{e}_1^\top$. Hence $\boldsymbol{e}_1\boldsymbol{e}_1^\top \in \mathcal{R}_1(\mathbf{B}, \mathbf{A})$ but $\boldsymbol{e}_1\boldsymbol{e}_1^\top \notin \mathcal{R}_1(\mathbf{BQ}, \mathbf{Q}^{-1}\mathbf{A})$. This establishes that the sparse families need not be equal even though the dense update is identical.

The counterexample is intentionally two-dimensional and concerns only the candidate function class. It does not assert that every invertible $\mathbf{Q}$ changes every sparse selection. Algebraically, invariance holds for arbitrary invertible $\mathbf{Q}$ only when the complete update is executed; our numerical stress tests in Appendix C sample orthogonal gauges.

A.2 Canonicality and repeated singular values

For a distinct nonzero singular value σ_i, the left and right singular vectors are unique up to a simultaneous sign change. Because $(-\boldsymbol{u}_i)(-\boldsymbol{v}_i)^\top = \boldsymbol{u}_i\boldsymbol{v}_i^\top$, the rank-one update $\mathbf{E}_i^{\mathrm{spec}} = \sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^\top$ is determined by $\Delta\mathbf{W}$ and is therefore identical for all of its factorizations. The Eckart–Young–Mirsky theorem further gives

$$\sum_{i=1}^{k} \sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^\top \in \arg\min_{\mathrm{rank}(\mathbf{X})\le k} \|\Delta\mathbf{W} - \mathbf{X}\|_{\mathrm{F}}. \tag{13}$$

If a set $\mathcal{I}$ indexes a repeated singular value, an orthogonal rotation within that singular subspace changes its individual rank-one terms. The invariant object is instead the block update

$$\mathbf{E}_{\mathcal{I}}^{\mathrm{spec}} = \mathbf{U}_{\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}}\mathbf{V}_{\mathcal{I}}^\top = \mathbf{P}_{U,\mathcal{I}}\Delta\mathbf{W} = \Delta\mathbf{W}\mathbf{P}_{V,\mathcal{I}}, \tag{14}$$

where $\mathbf{P}_{U,\mathcal{I}} = \mathbf{U}_{\mathcal{I}}\mathbf{U}_{\mathcal{I}}^\top$ and $\mathbf{P}_{V,\mathcal{I}} = \mathbf{V}_{\mathcal{I}}\mathbf{V}_{\mathcal{I}}^\top$. Consequently, a sparse Top-K rule that selects only part of a repeated block need not be invariant to rotations within that block. We use a deterministic numerical SVD basis to make implementation outputs reproducible, but this convention does not make the individual rank-one terms mathematically canonical. Exact and near-degenerate blocks are therefore interpreted only at the subspace level. Routing whole degenerate blocks would preserve this invariant object, but blockwise routing is a natural extension rather than part of the present rank-one Top-K algorithm.

A.3 Compact construction from LoRA factors

The complete $D_{\mathrm{out}} \times D_{\mathrm{in}}$ update need not be materialized. With thin QR factorizations

$$\mathbf{B} = \mathbf{Q}_B\mathbf{R}_B, \qquad \mathbf{A}^\top = \mathbf{Q}_A\mathbf{R}_A, \tag{15}$$

we decompose only the $R \times R$ core

$$\Gamma\mathbf{R}_B\mathbf{R}_A^\top = \widetilde{\mathbf{U}}\mathbf{\Sigma}\widetilde{\mathbf{V}}^\top. \tag{16}$$

The singular factors of the LoRA update are then $\mathbf{U} = \mathbf{Q}_B\widetilde{\mathbf{U}}$ and $\mathbf{V} = \mathbf{Q}_A\widetilde{\mathbf{V}}$. This construction has the same nonzero spectrum and reconstructs the same complete update as a dense SVD, while its expensive factorization is confined to the LoRA rank. The resulting time complexity is

$$\mathcal{O}\big((D_{\mathrm{out}} + D_{\mathrm{in}})R^2 + R^3\big). \tag{17}$$

The construction operates directly on the thin LoRA factors and therefore does not require materializing or storing the dense matrix $\Delta\mathbf{W} \in \mathbb{R}^{D_{\mathrm{out}} \times D_{\mathrm{in}}}$.

A.4 SPARSE GATES AND RESPONSE NORMALIZATION

For router logits $z_i(\boldsymbol{h})$ and temperature $T > 0$, let $\mathcal{S}(\boldsymbol{h})$ denote the K largest entries. The routed gate is

$$g_i(\boldsymbol{h}) = \begin{cases} K \dfrac{\exp(z_i(\boldsymbol{h})/T)}{\sum_{j \in \mathcal{S}(\boldsymbol{h})} \exp(z_j(\boldsymbol{h})/T)}, & i \in \mathcal{S}(\boldsymbol{h}), \\ 0, & \text{otherwise}, \end{cases} \tag{18}$$

so $g_i \geq 0$, at most K gates are nonzero, and $\sum_i g_i = K$. The gate-mass convention makes uniform full activation recover the complete expert sum when K equals the bank size. FLoSER routing replaces z_i by Eq. 8; the bank, K, temperature, and gate mass are unchanged. All reported experiments use $T = 1$.

For numerical stability, response magnitudes are transformed with $\phi_i = \log(1 + \rho_i)$ and standardized within each matrix-specific candidate bank. The normalization centers and standardizes the token-wise response features, while the learned signed α_i retains expert-specific directionality. All α_i are initialized to zero, so the initial FLoSER logits equal the matched FLoSER-S logits exactly.

B ADDITIONAL EXPERIMENTAL DETAILS

B.1 BACKBONE, ADAPTERS, AND ROUTED MODULES

The backbone is Qwen2.5-7B. Frozen-hidden-state analyses use revision `d149729398750b98c0af14eb82c78cfe92750796`; all paired downstream comparisons use the same locally verified backbone revision. The main bank contains three rank-8 PEFT LoRA adapters specialized for math, code, and chat/instruction data. Each adapted matrix contributes eight rank-one experts per adapter, giving 24 candidates per routed matrix. Unless explicitly identified as an all-module diagnostic, we route `q_proj` and `v_proj` in all 28 Transformer layers (56 routed matrices).

Table 6: Primary LoRA bank used in the Qwen2.5-7B experiments. The adapters are frozen throughout router training.

Role	Public adapter identifier	Rank
Math	`ASethi04/Qwen-Qwen2.5-7B-gsm8k-1000-lora-4-0.0001`	8
Code	`ASethi04/Qwen-Qwen2.5-7B-tulu-code_alpaca-first-lora-4-0.0001`	8
Chat	`ASethi04/Qwen-Qwen2.5-7B-opc-sft-20-lora-1-0.0002`	8

The structural generalization audit is broader: it contains 12 public PEFT LoRAs from eight authors, five normalized backbone/version groups, ranks $8/16/32/64$, and every available attention or MLP projection among `q_proj`, `k_proj`, `v_proj`, `o_proj`, `up_proj`, `down_proj`, and `gate_proj`. These weights are used only for frozen structural analysis, not for the main downstream ranking.

B.2 TRAINING AND EVALUATION PROTOCOLS

Table 7 separates the independently trained studies. This distinction is important: a shared seed label denotes a matched initialization only within one stated comparison, not across rows.

Table 7: Summary of the main downstream protocols. "Quota" is the average number of active experts per routed matrix; adaptive EEQA allocations preserve the stated total budget.

Study	Data	Training	Candidates	Quota	Seeds	Generation set
Central Raw–FLoSER-S	GSM8K	512 train, 128 dev, 100 steps	24	8	1–3	128 paired (seed 1)
Central FLoSER-S–FLoSER	GSM8K	512 train, 128 dev, 100 steps	24	8	1–3	fresh 256 per seed
Federated Raw–FLoSER-S	GSM8K	20 clients, 20 rounds, 10 local steps	24	12	1–3	256 per seed
Federated FLoSER-S–FLoSER	GSM8K	20 clients, 20 rounds, 10 local steps	24	12	1–3	fresh 256 per seed
Strong Non-IID fixed quota	GSM8K	20 clients, 20 rounds, 10 local steps	24	12 fixed	1–3	fresh 256 per seed
Federated MBPP	MBPP	374 train, 90 dev; same FL budget	24	8	1–3	500 per seed

Centralized router training uses AdamW, learning rate 5×10^{-4}, 100 optimizer steps, and effective batch size four. The backbone and all LoRA experts remain frozen. The federated GSM8K FLoSER

comparison uses local batch size one with gradient accumulation 16 and the same learning rate. Each round selects two of 20 clients and performs ten local optimizer steps; only router parameters are aggregated. Partition seed 20260727 and client sampling seed 20260728 are fixed within the paired studies. The MBPP runs use local batch size one with gradient accumulation four.

Within each paired comparison, the methods use the same serialized router initialization; equality is checked from parameter-state hashes rather than inferred from a shared seed label. Sparse gates preserve a total gate mass of K (`gate_mass=topk`), and matched methods use the same router temperature. We report only settings verified from the retained protocols and run artifacts rather than reconstructing unspecified framework defaults.

For the paper-guided EEQA allocation, a selected expert i in matrix m is assigned the scalar importance

$$\eta_{m,i}(\boldsymbol{h}) = g_{m,i}(\boldsymbol{h}) \left[1 - \cos\big(\boldsymbol{y}_m^{\text{base}}(\boldsymbol{h}), \boldsymbol{y}_m^{\text{base}}(\boldsymbol{h}) + \mathbf{E}_{m,i}\boldsymbol{h}\big)\right]. \tag{19}$$

The matrix-level importance sums s_m are aggregated without token-count normalization. We compute $p_m = \exp(s_m)/\sum_j \exp(s_j)$, initialize each integer quota as $q_m = \min\{\lfloor p_m K_{\text{tot}} \rfloor, M\}$, and assign the residual budget one unit at a time in descending p_m order among matrices below the M-expert capacity (ties follow matrix index), cycling until $\sum_m q_m = K_{\text{tot}}$. The audited GSM8K runs use one fixed-order importance example per selected client and round. Because these quotas are representation dependent, the adaptive-quota Raw–FLoSER-S comparisons measure end-to-end pipelines rather than a pure representation-only intervention. The fixed-quota strong Non-IID ablation disables EEQA and assigns quota 12 to every routed matrix throughout training and inference.

GSM8K generation uses greedy decoding and numerical exact-answer matching. MBPP uses greedy decoding with at most 512 new tokens and treats both `<|endoftext|>` and `<|im_end|>` as EOS; Pass@1 is evaluated with the same isolated scorer for every method. No sampling temperature is used in either benchmark.

B.3 Partitions and checkpoint selection

The IID partition shuffles the training set deterministically and distributes examples approximately evenly across clients. For strong GSM8K heterogeneity, we assign difficulty-proxy labels from the number of explicit solution steps: easy (≤ 2, 186 examples), medium ($= 3$, 150), and hard (≥ 4, 176). A balanced-capacity Dirichlet partition with concentration 0.1 gives each client 25 or 26 examples. Mean client-to-global label total variation is 0.4343 (range $[0.2716, 0.6563]$). This is a controlled difficulty-proxy skew rather than a claim about natural user heterogeneity.

Checkpoints are chosen without consulting the reported generation questions. The Raw–FLoSER-S IID runs select the minimum-development-loss checkpoint; all six selected checkpoints occur at round 18. Under strong GSM8K heterogeneity, the same predeclared rule selects Raw at round 18 for all seeds and FLoSER-S at rounds 9, 9, and 11. The matched FLoSER-S–FLoSER IID comparison uses the frozen round-18 checkpoints. MBPP uses the minimum validation-loss round, breaking exact ties toward the earlier round and carrying forward that round's quota allocation.

B.4 Metrics and uncertainty

For an update $\mathbf{X}$ and its sparse approximation $\widehat{\mathbf{X}}$, weight-space error is $\|\mathbf{X} - \widehat{\mathbf{X}}\|_{\text{F}}/\|\mathbf{X}\|_{\text{F}}$. Sparse-update variation compares two equivalent-factor Top-K updates using the original sparse update norm in the denominator. For hidden states $\{\boldsymbol{h}_t : t \in \mathcal{T}\}$, response reconstruction error is

$$\frac{\left(\sum_{t\in\mathcal{T}} \|\Delta\mathbf{W}\boldsymbol{h}_t - \widehat{\Delta\mathbf{W}}\boldsymbol{h}_t\|_2^2\right)^{1/2}}{\left(\sum_{t\in\mathcal{T}} \|\Delta\mathbf{W}\boldsymbol{h}_t\|_2^2\right)^{1/2}}. \tag{20}$$

We define k_{95} as the smallest number of ordered components whose squared norms contain at least 95% of the relevant energy and report $f_{95} = k_{95}/R$ when comparing different LoRA ranks. The learned-to-oracle gap is the learned-router response error minus the exact sparse-oracle error.

Question-level paired bootstrap intervals resample aligned questions. When three seeds evaluate the same questions, the aligned-question bootstrap first averages the paired difference within each question. Hierarchical bootstrap resamples seeds and then questions. Neither procedure turns repeated

predictions of the same question into independent test problems, and intervals over three fixed seeds should not be interpreted as certainty over the full training-seed population.

C CANONICALIZATION ANALYSES

C.1 CROSS-SOURCE GAUGE STABILITY

We analyze 1,832 LoRA matrices from the 12-adapter structural collection and apply 20 fixed random orthogonal gauges per matrix (36,640 comparisons). All invariance calculations use float64. The maximum relative reconstruction error of the full spectral reconstruction is 1.05×10^{-14}; the maximum relative singular-value deviation is 7.66×10^{-15}. For non-degenerate components, the minimum matched spectral-expert cosine is 0.999999999999984.

Table 8: Raw coordinate sensitivity by LoRA rank. Values are medians over matrix–gauge comparisons; lower Jaccard and larger update variation indicate greater sensitivity.

Rank	Matrices	Spearman	Top-K Jaccard	Variation
8	956	0.0000	0.3333	0.9099
16	484	0.0029	0.3333	0.8840
32	196	-0.0015	0.3333	0.8568
64	196	-0.0000	0.3333	0.9420

For Spectral, energy-rank Spearman and Top-K Jaccard equal one at every rank, and the largest Top-K relative change is 2.86×10^{-13}. Thus the empirical audit confirms orthogonal-gauge stability under the deterministic SVD convention. Algebraically, rank-one invariance to arbitrary invertible factor reparameterization holds for distinct spectral directions because they are defined by $\Delta\mathbf{W}$; repeated blocks support only subspace-level invariance, not invariant Top-K choices within the block.

At relative spectral-gap thresholds 10^{-3} and 10^{-2}, respectively 23/1,832 and 424/1,832 matrices contain a near-degenerate block. The latter count is dominated by the single rank-32 and rank-64 adapters. These cases are reported as subspace-level rather than rank-one canonicality.

C.2 SPECTRAL CONCENTRATION ACROSS ADAPTER SOURCES

Real LoRA updates are more spectrally concentrated than norm-matched random controls throughout the structural collection.

Table 9: Spectral concentration by source LoRA rank. Real quantities are adapter-equal-weighted means.

Rank	Adapters	Real f_{95}	Random f_{95}	Eff. rank (real/random)
8	6	0.7207	1.0000	4.30 / 7.90
16	4	0.6350	0.9418	6.46 / 15.65
32	1	0.6570	0.9334	11.42 / 30.95
64	1	0.5330	0.9280	12.55 / 61.92

The real-minus-random f_{95} difference has an adapter-bootstrap 95% interval below zero for every analyzed module: down $[-0.5153, -0.3345]$, gate $[-0.4990, -0.2785]$, value $[-0.3820, -0.2786]$, up $[-0.3217, -0.2186]$, output $[-0.3146, -0.1850]$, query $[-0.2874, -0.1957]$, and key $[-0.2544, -0.1785]$. A held-out adapter group also reproduces the effect (mean difference -0.3273, 95% CI $[-0.3928, -0.2447]$). Because ranks 32 and 64 each contain only one natural adapter, we do not infer a monotone rank–compressibility law.

C.3 MATCHED-BUDGET WEIGHT RECONSTRUCTION

We compare ordered Spectral Top-K, Raw components selected by Frobenius norm, random Raw subsets, and norm selection after random equivalent gauges. Every comparison uses the same retained component count.

Table 10: Relative Frobenius reconstruction error under matched retained-rank budgets. Lower is better.

Retained	Spectral	Gauge-Raw	Raw-Norm	Raw-Random
25%	0.4754	0.7063	0.8230	0.8457
50%	0.3100	0.5086	0.6397	0.6731
75%	0.1838	0.3198	0.4270	0.4628

Across all 13,317 matrix–budget pairs with $k < R$, Spectral is strictly better than Raw-Norm; the mean and median error reductions are 0.3148 and 0.3008. This empirical sweep quantifies the consequence of the classical optimal rank-k property on real LoRA banks; it is not presented as a new matrix approximation theorem.

At 95% spectral energy, the median expert-count reduction is 29.69%, with mean relative Frobenius error 0.1983. Mean reductions by module are 44.59% for down, 40.81% for gate, 36.90% for value, 29.94% for output, 29.23% for up, 27.29% for query, and 25.57% for key projections.

C.4 Frozen-hidden-state response geometry

We extract the last non-padding-token hidden state from 512 math, 512 code, and 512 chat prompts at layers 0, 14, and 27. Combining three rank-8 adapters with q/v projections gives 54 task–matrix response units. These probes use the frozen backbone and do not train a router.

Table 11: Expert-response structure on real frozen hidden states. Up arrows indicate that larger values mean greater response concentration or separation; down arrows indicate that smaller values are preferred.

Metric	Raw	Spectral-Full	Spectral-k_{95}
Hoyer sparsity ↑	0.2597	0.5533	0.5260
Gini ↑	0.3578	0.5549	0.5108
Top1/Top2 margin ↑	0.2329	0.5478	0.5497
Response k_{95} ↓	5.3118	3.1638	2.9480
Response effective rank ↓	4.9221	2.7648	2.6279
Off-diagonal response cosine ↓	0.1959	5.49×10^{-17}	6.57×10^{-17}

Hoyer, Gini, margin, and response k_{95} improve in 51, 51, 50, and 51 of 54 paired units, respectively. The probe also exposes limitations. Response Gram condition number increases from 37.09 to 1030.96, dead-expert rate from 1.39% to 2.78%, and always-on rate from 1.62% to 4.86%. Greater response concentration therefore does not imply balanced utilization or improved conditioning.

The 18 rank-8 matrices retain 6.33 components on average at 95% weight energy, but the resulting relative response error has mean 0.0861, 95th percentile 0.1808, and maximum 0.3111. Weight-energy preservation is consequently not a guarantee of small activation-space error.

C.5 Exact sparse-response oracle

For each correct task adapter's eight-expert bank, we enumerate every nonempty support of size at most $\min(K, 8)$ and solve the convex gate-weight problem on that support subject to nonnegative gates and total mass K. This is an exact small-dictionary oracle for reconstructing the complete LoRA response, not an oracle for final language-model accuracy.

Spectral has lower mean error in 45 of 72 budget–task–layer–module units, Raw in nine, with 18 numerical ties (principally full activation). The 24-expert cross-adapter analysis uses approximate support search and is not treated as an exact lower bound.

C.6 Spectral energy truncation is not generation-lossless

In the centralized q/v pilot, Spectral-Full and Spectral-k_{95} reduce the candidate count and router parameters differently but have similar teacher-forced validation loss. Generation separates them.

Table 12: Normalized response error for the exact sparse oracle.

K	Raw mean	Spectral mean	Raw P95	Spectral P95
2	0.4695	0.2967	0.5935	0.4761
4	0.2081	0.1442	0.3132	0.2622
6	0.0694	0.0542	0.1411	0.1127
8	0.0000	0.0000	0.0000	0.0000

Table 13: Centralized Spectral-k_{95} diagnostic. Generation uses the same paired 128 GSM8K questions as the Raw–FLoSER-S pilot.

Method	Dev loss (mean $\pm$ SD)	Experts	Accuracy
Raw	0.4943 ± 0.0044	1,344	14.1
Spectral-Full	0.3905 ± 0.0050	1,344	30.5
Spectral-k_{95}	0.3890 ± 0.0066	1,083	23.4

Spectral-k_{95} removes 19.4% of candidates and router parameters, but is 7.0 points below Spectral-Full in generation (paired bootstrap 95% CI $[-14.8, +0.8]$, McNemar $p = 0.1221$). We therefore use the full spectral bank for the main comparisons and make no lossless-compression claim.

C.7 Routed-module scope diagnostic

The main experiments focus on $\mathbf{W}_q/\mathbf{W}_v$ projections. As an exploratory scope diagnostic, we additionally routed every module shared by the three source adapters ($q/k/v/gate/up/down$ across 28 layers), giving 168 routed matrices, 4,032 candidates, and a fixed Top-8 budget. Spectral reduced three-seed mean development loss from 0.2863 ± 0.0016 to 0.2755 ± 0.0023, but the paired seed-1 accuracy changed only from 67/256 (26.2%) to 73/256 (28.5%). The $+2.3$-point difference has paired bootstrap interval $[-4.3, +9.4]$ and McNemar $p = 0.5811$. This is a scope ablation, not evidence that additional modules are generally harmful; adapter coverage and routing budget change substantially when the module set expands.

D Response-aware analyses

D.1 Response salience and first-order utility

For a selected expert contribution $\boldsymbol{c}_i(\boldsymbol{h}) = g_i(\boldsymbol{h})\mathbf{E}_i\boldsymbol{h}$, removing that contribution changes the loss to first order by

$$\kappa_i(\boldsymbol{h}) = -\nabla_{\boldsymbol{y}}\ell(\boldsymbol{y})^\top \boldsymbol{c}_i(\boldsymbol{h}). \tag{21}$$

A positive κ_i means that removing the expert is predicted to increase loss; a negative value means its current contribution is locally harmful. The audit uses 16 examples, five layers, and both $\mathbf{W}_q/\mathbf{W}_v$ projections. We correlate $\rho_i(\boldsymbol{h})$ with $\kappa_i(\boldsymbol{h})$ and $|\kappa_i(\boldsymbol{h})|$, and compare their Top-$K$ sets.

Table 14: Spearman relationship between response magnitude and first-order expert utility.

Tokens	Signed utility	Absolute utility	Top-K overlap
All	0.0204	0.3732	47.31%
Answer	0.0241	0.4587	47.84%

The response signal is informative about influence magnitude but nearly uninformative about its sign. This is why Eq. 8 uses a learned signed coefficient rather than treating response magnitude as a fixed positive routing preference.

D.2 Oracle-to-learned routing gap

The learned-router diagnostic uses 512 frozen hidden states per unit, split 320/96/96 for training/development/test; three adapters, layers 0/14/27, q/v, $K \in \{2, 4, 6\}$, and three router seeds

produce 162 paired runs. Raw and Spectral share the linear router architecture, numerical initialization, minibatch order, optimizer, 120-step budget, and hard Top-K validation.

Table 15: Learned hard Top-K response error and exact-oracle gap. Lower is better.

K	Raw router	Spectral router	Raw gap	Spectral gap
2	0.5847	0.6876	0.1144	0.3908
4	0.3652	0.4393	0.1567	0.2949
6	0.2431	0.2552	0.1734	0.2013

Despite the better Spectral oracle, the hard router favors Spectral in only 73/162 paired runs; Spectral has the smaller oracle gap in 44/162. Its best checkpoint remains at initialization in 38.89% of runs, versus 3.09% for Raw. This does not imply that Spectral is generally harder to route: the states are off-policy, only the final input token is used, each task receives its known eight-expert adapter, and the objective reconstructs a layer response rather than optimizing language-model loss.

D.3 Support-exploration diagnostic

We test whether hard support selection contributes to the gap by linearly annealing the training gate from dense soft to hard Top-K over the first 60 of 120 steps; development and test always use hard Top-K. This intervention restores nonzero gradients to every unselected logit during the soft phase.

Table 16: Frozen-state reconstruction error under hard and Soft-to-Hard (S2H) training. Final evaluation uses the same hard sparse budget.

Training	Raw	Spectral	Spectral − Raw
Hard	0.3976	0.4607	+0.0631
S2H-60	0.3416	0.2293	−0.1123

S2H improves Spectral in 155/162 runs and reduces its initialization-selected checkpoint rate from 38.89% to 0.62%. At $K = 4$, exact-oracle support recall increases from 0.5182 to 0.7639. Annealing lengths 30/60/90 yield Spectral errors 0.2170/0.2082/0.2072 in the $K = 4$ analysis, showing that the effect is not an isolated 60-step setting.

In a separate downstream 2×2 check, however, Spectral improves teacher-forced GSM8K loss under both Hard and S2H, while the incremental S2H effect on Spectral is small and seed-inconsistent. Thus S2H is a mechanism diagnostic rather than part of the proposed method.

Table 17: GSM8K development loss in the downstream representation–schedule 2×2 diagnostic (mean $\pm$ SD over three seeds).

Expert bank	Hard	S2H-60
Raw	0.41989 ± 0.00135	0.41303 ± 0.00128
FLoSER-S	0.35832 ± 0.00936	0.35611 ± 0.00334

D.4 Centralized development and fresh evaluation

The first FLoSER comparison is a development study on 128 questions. After freezing the method form, we evaluate the corresponding matched checkpoints on a disjoint fresh set of 256 questions. The fresh comparison is the confirmatory result used in the main text.

Across the fresh predictions, FLoSER fixes 83 FLoSER-S errors and breaks 57 FLoSER-S successes. The aligned-question bootstrap 95% CI is $[+0.39, +6.38]$ points, while the hierarchical seed–question interval is $[-0.65, +8.07]$. The 3/3 positive seed differences support a positive but seed-sensitive trend, not a statistically established population advantage.

Table 18: Centralized FLoSER development results (accuracy, %).

Method	Seed 1	Seed 2	Seed 3	Mean
FLoSER-S	32.03	21.09	34.38	29.17
FLoSER	42.97	28.13	33.59	34.90
Δ	+10.94	+7.03	-0.78	+5.73

Table 19: Centralized FLoSER fresh results on 256 paired questions per seed (accuracy, %).

Method	Seed 1	Seed 2	Seed 3	Mean
FLoSER-S	29.30	19.92	28.91	26.04
FLoSER	30.47	27.34	30.47	29.43
Δ	+1.17	+7.42	+1.56	+3.39

D.5 Alpha-only mechanism pilot

To test whether the response feature can change useful decisions without retraining the original router, we freeze one existing FLoSER-S checkpoint and optimize only the 1,344 response coefficients. Accuracy changes from 45/128 (35.16%) to 49/128 (38.28%): ten questions are fixed and six are broken. This +3.13-point single-seed pilot is descriptive and is not part of the main method comparison.

Response-coefficient scale audit. Across the retained formal FLoSER runs, mean $|\alpha|$ lies between roughly 0.005 and 0.020. Qwen maxima lie between 0.019 and 0.068, and the Mistral federated maxima lie between 0.041 and 0.046. The largest observed value is $\max_i |\alpha_i| = 0.11017$ in centralized Mistral seed 2, whose mean $|\alpha|$ is 0.01183. All inspected training and validation losses are finite, and the formal logs contain no recorded NaN, Inf, overflow, or non-finite event. These observations provide no evidence of coefficient explosion in the reported runs, but they are a descriptive audit rather than a general optimization-stability guarantee.

D.6 Exploratory activation-conditioned banks

Before adopting online response features, we explored rebuilding the expert bank from fixed hidden-state second moments. On prompt-disjoint frozen states, the balanced activation-conditioned bank reduced fixed-$k = 4$ response MSE from 0.07501 to 0.06235 across all 28 layers (16.88% ratio-of-means reduction; 483/504 paired units improved). This verifies local response reconstruction, but not autoregressive task quality.

The distinction matters empirically. A static prompt/answer mixture with $\beta = 0.5$ improved an initial 128-question set by 5.99 points but was 2.86 points below FLoSER-S on a disjoint 256-question set. A separately frozen $\beta = 0.75$ choice produced seed-wise fresh differences of $+5.08$, -1.56, and -4.30 points (mean -0.26). We therefore do not include an activation-conditioned bank in the proposed method. These exploratory results motivated the separation used in this paper: expert identity remains a function of $\Delta\mathbf{W}$, while current-token response information enters only the learned routing score.

E Federated analyses

E.1 Seed-wise FLoSER-S results

The per-seed paired bootstrap intervals are $[+12.11, +25.39]$, $[+12.50, +24.61]$, and $[+3.12, +16.02]$ points. The aligned-question three-seed bootstrap interval is $[+10.94, +20.31]$. FLoSER-S has lower validation loss in 57/60 same-seed, same-round comparisons; all six minimum-loss checkpoints occur at round 18.

Table 20: GSM8K IID federated Raw–FLoSER-S results on 256 questions per seed (accuracy, %).

Method	Seed 1	Seed 2	Seed 3	Mean ± SD
Raw	24.61	21.09	23.83	23.18 ± 1.85
FLoSER-S	43.36	39.45	33.59	38.80 ± 4.92
Δ	+18.75	+18.36	+9.77	$+15.62 \pm 5.08$

Table 21: GSM8K federated Raw–FLoSER-S results under adaptive EEQA and the fixed difficulty-proxy Dirichlet-0.1 partition (accuracy, %).

Method	Seed 1	Seed 2	Seed 3	Mean
Raw	28.52	27.34	25.78	27.21
FLoSER-S	35.94	33.98	31.25	33.72
Δ	+7.42	+6.64	+5.47	+6.51

E.2 Strong GSM8K heterogeneity with adaptive quotas

FLoSER-S fixes 134 seed–question errors and breaks 84 successes. The aligned-question and hierarchical bootstrap 95% intervals are $[+1.95, +11.07]$ and $[+2.73, +10.29]$ points. A sign test over only the three training seeds is necessarily weak (two-sided $p = 0.25$), so the intervals must not be read as establishing general superiority over all training randomness.

Table 22: Best and final GSM8K development losses under strong heterogeneity.

Method	Seed 1 best	Seed 2 best	Seed 3 best	Final mean
Raw	0.4049 (R18)	0.3999 (R18)	0.4009 (R18)	0.4099
FLoSER-S	0.3504 (R9)	0.3552 (R9)	0.3491 (R11)	0.4202

FLoSER-S's best loss is lower for all seeds and in 56/60 same-round comparisons, but its final loss is higher for all seeds. This late-round rebound is why checkpoint selection follows the predeclared development rule rather than the final communication round.

E.3 FedAvg update geometry

For selected-client updates $\boldsymbol{\delta}_c$, we report their mean pairwise cosine and the aggregation-retention ratio

$$\text{retention} = \frac{\|\sum_c p_c \boldsymbol{\delta}_c\|_2}{\sum_c p_c \|\boldsymbol{\delta}_c\|_2}. \tag{22}$$

Table 23: Client-update geometry under strong GSM8K heterogeneity, averaged over three seeds.

Method	Pairwise cosine	Retention
Raw	0.0756	0.7323
FLoSER-S	0.0704	0.7306

Neither diagnostic improves for FLoSER-S. We therefore find no evidence that its accuracy gain is mediated by greater client-update alignment or less FedAvg cancellation.

E.4 Federated FLoSER results

FLoSER fixes 107 FLoSER-S errors and breaks 88 successes. The aligned-question and hierarchical bootstrap intervals are approximately $[-1.04, +5.86]$ and $[-1.43, +6.64]$ points. The mean absolute learned response coefficient is about 0.020 and approximately 98% of coefficients are positive; the correction therefore does not remain at its zero initialization.

This comparison establishes a boundary specific to adaptive allocation: the IID mean improvement disappears and seed-wise directions disagree. Earlier balanced-sampling, FedProx, and alpha-only

Table 24: GSM8K IID federated FLoSER-S–FLoSER results on the fresh 256 questions (accuracy, %).

Method	Seed 1	Seed 2	Seed 3	Mean
FLoSER-S	37.89	41.02	34.77	37.89
FLoSER	39.84	41.80	39.45	40.36
Δ	+1.95	+0.78	+4.69	+2.47

Table 25: GSM8K FLoSER-S–FLoSER results under adaptive EEQA and the strong difficulty-proxy skew (accuracy, %).

Method	Seed 1	Seed 2	Seed 3	Mean
FLoSER-S	36.33	40.23	38.28	38.28
FLoSER	41.80	38.28	34.77	38.28
Δ	+5.47	-1.95	-3.52	0.00

diagnostics used nonuniform, post-outcome protocols and are not used for ranking. The controlled fixed-quota follow-up below tests whether the positive signals require EEQA.

E.5 Fixed-quota ablation and allocation interaction

We disable EEQA from the start of training and assign quota 12 to each of the 56 routed matrices, preserving a total budget of 672. All methods use $\Lambda = 1$, the same clients, Dirichlet-0.1 partition, client schedule, and 20-round budget; generation uses the predeclared round-18 checkpoint and the same fresh 256 questions.

Table 26: Strong Non-IID fixed-quota accuracy (%).

Method	Seed 1	Seed 2	Seed 3	Mean
Raw	30.47	28.91	27.73	29.04
FLoSER-S	41.02	30.08	32.03	34.38
FLoSER	41.02	36.33	33.20	36.85
Spec. $-$ Raw	+10.55	+1.17	+4.30	+5.34
RA $-$ Spec.	0.00	+6.25	+1.17	+2.47

FLoSER-S and FLoSER exceed Raw for all three seeds, by 5.34 and 7.81 points on average. FLoSER versus FLoSER-S yields two wins and one tie, but its aligned-question and hierarchical bootstrap intervals are $[-1.43, +6.25]$ and $[-2.21, +7.42]$; we do not claim a statistically established incremental advantage. For FLoSER-S versus Raw, the corresponding intervals are $[+0.65, +10.03]$ and $[-0.26, +11.46]$.

The fixed-quota result shows that the positive FLoSER-S signal does not require adaptive EEQA allocation in this setting. FLoSER has zero mean incremental gain with adaptive EEQA but a positive mean with fixed quotas. This contrast is not a causal estimate of EEQA's effect, because allocation changes the complete training trajectory and the experiment groups are not hardware-identical (seed 1 of the fixed-quota run used an RTX 4090; seeds 2–3 used A100-PCIE-40GB GPUs). Within each seed, all three methods use the same hardware, so the paired comparisons remain valid.

E.6 External baseline protocols and complete results

Comparison scope. All methods below are evaluated on the same fresh 256-question GSM8K set. Frozen references use a single fixed predictor and no federated training data. The trained external baselines and our rank-wise routers use the same strong Non-IID client partition and the predeclared round-18 generation protocol. They nevertheless differ in pretrained information, trained parameter class, and active rank; accordingly, we group them rather than claim a strictly matched parameter- or compute-budget ranking.

Table 27: Complete frozen-reference results on the fresh GSM8K evaluation set. No row uses federated training data.

Method	Accuracy (%)
Base model	5.47
Single Math adapter	4.69
Single Code adapter	**12.89**
Single Chat adapter	4.69
Uniform merge	5.86
Static Top-σ	11.72
Dense/full-bank activation	11.33

Frozen reuse references. The single Code adapter is the strongest frozen reference. Uniform merging is near the base model, indicating negative transfer from naive composition. Static Top-σ confirms that spectral strength is a useful inductive bias, but its 11.72% accuracy cannot explain the 34.38% obtained by learned FLoSER-S routing. Dense activation likewise shows that activating more experts does not automatically improve generation.

Whole-adapter learned routing. We train a token-level gate over complete frozen adapters using the same strong Non-IID partition. This is a project-specific whole-adapter, LoRA-Flow-style baseline, not an official reproduction of LoRA-Flow. Top-1 obtains 12.50 ± 2.03 and Top-2 obtains 12.76 ± 0.60 accuracy over three router seeds. The 0.26-point difference is small even though Top-2 activates two full rank-8 adapters. Under this protocol, adapter-level learned routing therefore does not reproduce the gain of rank-wise learned routing.

FedAvg-LoRA from scratch. The FedAvg-LoRA baseline trains rank-12 LoRA factors rather than reusing the frozen pretrained bank. It uses the same small-data, strong Non-IID partition and fixed training budget. We score free generation with both prescribed EOS markers; extracting the first intermediate answer marker from an unterminated trajectory is not used.

Table 28: FedAvg-LoRA generation accuracy under the dual-EOS protocol.

Seed	Correct	Accuracy (%)
1	10/256	3.91
2	33/256	12.89
3	28/256	10.94
Mean $\pm$ SD	–	9.24 ± 4.73

Best validation losses lie between approximately 0.262 and 0.269, but do not translate into strong free-generation accuracy, illustrating a gap between teacher-forced loss and autoregressive generation. This result supports only the protocol-specific conclusion that factor-wise FedAvg-LoRA from scratch is ineffective and seed-variable under the present small-data, strongly heterogeneous, fixed-budget setting; it is not a general claim about FedAvg-LoRA.

Relation to the matched rank-wise comparison. Under fixed quota, Raw, FLoSER-S, and FLoSER reach 29.04%, 34.38%, and 36.85%, respectively (Table 26). FLoSER-S exceeds Raw in all three seeds. FLoSER exceeds Raw in all three seeds and is positive in two seeds and tied in one relative to FLoSER-S, but its hierarchical interval crosses zero. Subject to their different information and computation budgets, the external controls contextualize the benefit of learned rank-wise routing. The matched internal comparison more directly supports spectral construction as the core contribution and Response-Aware routing as a smaller, setting-dependent refinement.

F SECOND-BACKBONE EVALUATION

F.1 PROTOCOL AND COMPARABILITY BOUNDARY

We evaluate Mistral-7B-Instruct-v0.2 at revision `63a8b081895390a26e140280378bc85ec8bce07a` with frozen rank-16 adapters `Lots-of-LoRAs/Mistral-7B-Instruct-v0.2-4b-r16-task753`, `Lots-of-LoRAs/Mistral-7B-Instruct-v0.2-4b-r16-task200`, and `Lots-of-LoRAs/Mistral-7B-Instruct-v0.2-4b-r16-task362` for math, natural-language inference, and chat, respectively. The complete 12-adapter structural-audit manifest is provided in `experiments/configs/stage2_adapters.json` in the supplementary code. Query and value projections in all 32 layers give 64 routed matrices and 48 candidates per matrix. Every method uses fixed quota 12, Top-K gate mass, $\Lambda = 1$, eager attention, BF16, the official Mistral chat template, greedy decoding, and at most 192 new tokens. The backbone and bank are frozen; all methods train routers, and FLoSER additionally trains response coefficients. The maximum training length is 384 tokens.

Federated runs use 20 clients, two clients per round, 20 rounds, 10 local steps, batch size one with 16-step accumulation, and learning rate 5×10^{-4}. Router seeds are 1–3; partition and sampling seeds are fixed at 20260727 and 20260728. We evaluate the pre-specified round-18 checkpoint. Centralized runs use router seeds 1–3 and 360 optimizer steps per seed. Both protocols process 5,760 examples nominally, but their optimization trajectories, optimizer states, token counts, FLOPs, and communication differ and therefore do not support a causal centralized–federated comparison.

Non-IID runs use two RTX 4090 GPUs; most IID and all centralized runs use an A100-PCIE-40GB. IID Raw seed 1 is trained on an RTX 4090 and evaluated on the A100. We therefore do not describe the study as an identical-hardware comparison. One interrupted IID FLoSER job is resumed after reordering serialized state keys to the local key order; parameter tensors and the experimental protocol are unchanged.

Training uses 512 GSM8K examples and 128 validation examples; evaluation uses the same 256-question held-out set for all variants. This set had been used earlier in the research process and is not historically untouched. Relative to the Qwen study, Mistral changes the backbone, adapters, rank, layer count, candidate count, and active fraction ($12/48$ rather than $12/24$). The audit therefore tests transfer to a second backbone–adapter bank, not a controlled backbone-only effect.

F.2 ACCURACY AND PAIRED OUTCOMES

Table 29: Mistral GSM8K accuracy (%) by router seed. All means report sample SD over three router seeds. Bold marks the largest point estimate within each training setting and column, not significance.

Training	Method	Seed 1	Seed 2	Seed 3	Mean ± SD
Fed. Non-IID	Raw	25.78	28.52	**29.69**	27.99 ± 2.00
	FLoSER-S	**28.91**	29.30	26.56	28.26 ± 1.48
	FLoSER	26.17	**30.86**	28.12	$\mathbf{28.39 \pm 2.35}$
Fed. IID	Raw	27.34	26.17	**32.81**	28.78 ± 3.54
	FLoSER-S	**28.52**	**32.03**	28.91	$\mathbf{29.82 \pm 1.93}$
	FLoSER	25.39	29.30	32.42	29.04 ± 3.52
Centralized	Raw	30.08	35.16	**30.08**	31.77 ± 2.93
	FLoSER-S	32.42	**36.33**	28.91	$\mathbf{32.55 \pm 3.71}$
	FLoSER	**36.33**	28.91	**30.08**	31.77 ± 3.99

Under Non-IID federation, mean changes relative to Raw are $+0.26$ points for FLoSER-S and $+0.39$ for FLoSER; both comparisons contain two positive and one negative seed. Under IID federation, the corresponding changes are $+1.04$ and $+0.26$ points, with inconsistent seed-wise directions. These close means do not establish equivalence. Centralized mean changes relative to Raw are $+0.78$ points for FLoSER-S and 0.00 for FLoSER; FLoSER is -0.78 points relative to FLoSER-S. The changing seed-wise order again provides no evidence of seed-robust or centralized-over-federated superiority.

The aggregated fixed/broken counts are 108/106 (Raw → FLoSER-S), 104/101 (Raw → FLoSER), and 96/95 (FLoSER-S → FLoSER) under Non-IID federation. The IID counts are 109/101, 106/104, and 89/95, respectively. Paired transition counts are available only for the originally audited centralized seed 1: 35/29, 41/25, and 39/29, respectively. These counts are not used to characterize the three-seed mean ordering. The large bidirectional changes with small federated net differences show that the variants do not merely reproduce identical outputs.

F.3 Optimization and response-coefficient audit

All nine IID federated runs attain their minimum validation loss at round 9, whereas evaluation remains fixed at round 18 to avoid selection on the held-out accuracy. In the centralized three-seed results, FLoSER-S shifts the mean from 31.77% to 32.55% ($+0.78$ points), whereas full FLoSER returns to 31.77%; the ordering also changes across seeds. The available seed-1 step-360 losses are 0.7327, 0.7363, and 0.7606 for Raw, FLoSER-S, and FLoSER, respectively, so this single-seed loss audit is only an optimization diagnostic and cannot establish the three-seed accuracy ordering.

Table 30: Mistral response-coefficient audit. Federated entries show all three router seeds; the centralized entry is the available seed-1 diagnostic. Coefficients move away from zero initialization in every reported run.

Training	Seed	Count	Mean $\lvert\alpha\rvert$	Positive (%)
Fed. Non-IID	1	3,072	0.00823	82.26
	2	3,072	0.00797	81.45
	3	3,072	0.00792	82.45
Fed. IID	1	3,072	0.00828	83.82
	2	3,072	0.00835	83.14
	3	3,072	0.00795	82.55
Centralized	1	3,072	0.01205	82.85

The coefficient audit rules out the narrower failure mode in which the response coefficients remain at their zero initialization. FLoSER-S has positive aggregate mean shifts of $+0.78$, $+1.04$, and $+0.26$ points centrally, under IID federation, and under Non-IID federation, respectively, but the seed-wise directions are inconsistent. The Qwen fixed-quota Non-IID gains of 5.34 points for FLoSER-S and 7.81 points for FLoSER do not reproduce at comparable magnitude here. We therefore treat Mistral as a transfer boundary: downstream conversion depends on the backbone, adapter bank, and training regime.

G MBPP evaluation protocol and diagnostics

G.1 Why inference-gain calibration is reported separately

All MBPP routers are trained with routed-residual gain $\Lambda = 1$. For token t, inference applies

$$\boldsymbol{y}_t = f_{\text{base}}(\boldsymbol{h}_t) + \Lambda \Delta \boldsymbol{y}_{\text{routed}}(\boldsymbol{h}_t). \tag{23}$$

Changing Λ does not modify trained parameters. Given the current $\boldsymbol{h}_t$, it also leaves the pre-scaling router logits, Top-K support, and relative gate weights unchanged. It does change $\boldsymbol{y}_t$ and can therefore alter later hidden states and subsequent routing decisions. A post-hoc diagnostic found that the more anisotropic Spectral bank can over-amplify autoregressive updates under the original fixed gate mass: expert-norm coefficient of variation is 0.4008 for Raw and 0.7550 for Spectral, while the maximum-expert-to-median norm ratio is 2.03 versus 4.89.

The original uncalibrated MBPP evaluation is retained as a failure boundary: despite lower teacher-forced loss, spectral routing at $\Lambda = 1$ produced long generations, frequent 512-token truncation, invalid syntax, and lower Pass@1 than Raw. The subsequent calibration analysis explains rather than replaces this negative result.

On 24 deliberately selected runaway cases, changing Λ from one to 0.5 restores normal EOS and syntactic validity in all cases and reduces mean generation length from 128.00 to 35.75 tokens. Because this diagnosis followed inspection of earlier MBPP behavior, it is not itself treated as a

confirmatory test result. It motivates the later validation-only shared selection rule and the explicit representation–scaling caveat in the main text.

G.2 THREE-SEED MATCHED-GAIN VALIDATION

We reuse the same three IID federated checkpoints within every method–seed pair and change only the inference gain; no router is retrained. All checkpoints were trained with $\Lambda = 1$, and the 90 validation tasks, client partition, sampling history, quotas, prompts, and decoding rules are fixed.

Table 31: MBPP validation Pass@1 (%) under matched inference gains. Each row uses the same 90 tasks; mean and sample SD are over three router seeds.

Method	Λ	Seed 1	Seed 2	Seed 3	Mean $\pm$ SD
Raw	1.0	47.78	50.00	51.11	49.63 ± 1.70
Raw	0.5	43.33	38.89	42.22	41.48 ± 2.31
FLoSER-S	1.0	24.44	22.22	18.89	21.85 ± 2.80
FLoSER-S	0.5	44.44	48.89	48.89	47.41 ± 2.57
FLoSER	1.0	24.44	24.44	12.22	20.37 ± 7.06
FLoSER	0.5	48.89	48.89	50.00	49.26 ± 0.64

Table 32: Matched-gain generation diagnostics aggregated over 270 validation outputs (three seeds and 90 shared tasks).

Method	Λ	Normal EOS	Trunc.	Invalid	Mean tok.	Median tok.
Raw	1.0	270	0	46	57.17	42.0
Raw	0.5	270	0	67	63.63	44.5
FLoSER-S	1.0	159	111	156	260.96	152.0
FLoSER-S	0.5	270	0	54	54.99	42.0
FLoSER	1.0	161	109	153	255.11	131.0
FLoSER	0.5	270	0	48	53.85	40.0

Let $a_m(\Lambda)$ denote mean validation Pass@1 for method m. We report the gain–representation interaction contrast relative to Raw,

$$\delta^m_{\mathrm{GR}} = [a_m(0.5) - a_m(1)] - [a_{\mathrm{Raw}}(0.5) - a_{\mathrm{Raw}}(1)]\,. \tag{24}$$

It equals $+33.70$ points for FLoSER-S and $+37.04$ for FLoSER. Halving the gain decreases Raw by 8.15 points but increases the two spectral methods by 25.56 and 28.89 points, respectively, with the same direction in all seeds. At $\Lambda = 0.5$, FLoSER-S and FLoSER exceed Raw by 5.93 and 7.78 points, again in all seeds. This rules out a generic benefit from gain reduction and shows that the intervention is strongly representation-dependent in this setting. It is not a pure representation effect: the methods use separately trained routers, and autoregressive trajectories remain part of the observed outcome.

Using each method's better of the two validation operating points gives 49.63% for Raw, 47.41% for FLoSER-S, and 49.26% for FLoSER. Accordingly, the MBPP conclusion is calibration recovery to the Raw vicinity, not superiority over Raw. Because these validation tasks informed gain development, the table is mechanism evidence rather than an independent generalization result. We also did not conduct a fully symmetric, method-specific gain search for Raw, so these results must not be read as a gain-optimized ranking across methods.

G.3 CENTRALIZED FIXED-GAIN CHECK

We reuse the centralized seed-1 checkpoints previously trained on MBPP, with no additional training or new gain search. Raw is evaluated with $\Lambda = 1$, whereas FLoSER-S and FLoSER use the shared $\Lambda = 0.5$ inherited from the federated validation protocol. All methods use the same 500 MBPP problems, greedy decoding, a 512-token generation limit, two EOS markers, Top-$K = 8$, and eager attention.

Table 33: Centralized single-seed MBPP results under the fixed-gain protocol.

Method	Pass@1	Δ Raw	Normal EOS	Trunc.	Invalid	Mean tokens
Raw	44.00	–	492	8	70	68.86
FLoSER-S	**50.40**	+6.40	495	5	47	51.29
FLoSER	49.20	+5.20	494	6	46	53.92

Relative to Raw, FLoSER-S fixes 54 failures and breaks 22 successes, for a net gain of 32 problems. FLoSER fixes 48 and breaks 22, for a net gain of 26. Relative to FLoSER-S, FLoSER fixes 14 and breaks 20, for a net loss of six. These are single-seed descriptive results and cannot establish seed-consistent superiority or a difference between centralized and federated training.

G.4 Federated shared-gain calibration

We reuse the three $\Lambda = 1$ IID federated checkpoints previously trained on MBPP, with no additional training during this frozen evaluation. The shared inference-time gain is selected from $\{0.25, 0.5, 0.75, 1.0\}$ on 90 validation problems using only mean FLoSER-S Pass@1 across three seeds, with ties resolved toward the larger value. This procedure selects $\Lambda^* = 0.5$, which is applied without further tuning to both FLoSER-S and FLoSER; Raw remains at $\Lambda = 1$. The selection is frozen before the reported 500-problem evaluation. Checkpoints, quotas, prompts, and decoding rules are fixed across methods. The official test split had been used during an earlier stage of the project, so this result is a frozen-configuration evaluation, not a historically untouched-test claim.

Table 34: Seed-wise MBPP Pass@1 (%) for the frozen IID federated checkpoints.

Method	Seed 1	Seed 2	Seed 3	Mean ± SD
Raw	48.40	49.00	47.80	48.40 ± 0.60
FLoSER-S	48.20	48.60	49.80	48.87 ± 0.83
FLoSER	50.40	48.40	50.00	$\mathbf{49.60 \pm 1.06}$

Table 35: Federated MBPP generation diagnostics. Counts aggregate 1,500 outputs per method (three seeds and 500 shared problems); these are not 1,500 independent test problems.

Method	Normal EOS	Trunc.	Invalid syntax	Mean tokens
Raw	1478	22	169	60.49
FLoSER-S	1483	17	151	53.54
FLoSER	1492	8	138	49.57

Table 36: Paired MBPP outcome transitions aggregated over the three IID federated seeds. "Fixed" means incorrect-to-correct and "broken" means correct-to-incorrect relative to the first method.

Comparison	Fixed	Broken	Net
Raw → FLoSER-S	110	103	+7
Raw → FLoSER	119	101	+18
FLoSER-S → FLoSER	47	36	+11

FLoSER is 1.20 points above Raw on average, with seed-wise differences $+2.00, -0.60, +2.20$; it is 0.73 points above FLoSER-S. FLoSER-S is 0.47 points above Raw. These are modest positive means without unanimous seed-level superiority or an established significance claim.

G.5 Federated Non-IID with code-length skew

We group the 374 MBPP training problems by the number of nonempty lines in the reference solution: at most four lines (154 problems), 5–7 lines (119), and at least eight lines (101). A fixed balanced Dirichlet partition with concentration 0.1 assigns these labels across 20 clients, each containing 18 or

19 problems; mean client label-distribution total variation is 0.4160. This is controlled code-length skew, not intrinsic difficulty. Partition and client-sampling seeds are fixed across methods and router seeds.

All methods train for 20 federated rounds with $\Lambda = 1$. We select the minimum-validation-loss checkpoint, breaking ties toward the earlier round, and evaluate using that round's quota allocation. Inference uses $\Lambda = 1$ for Raw and the previously selected shared $\Lambda = 0.5$ for FLoSER-S and FLoSER, with no new scale search.

Table 37: Seed-wise MBPP Pass@1 (%) under federated code-length skew.

Method	Seed 1	Seed 2	Seed 3	Mean $\pm$ SD
Raw	48.40	47.60	47.20	47.73 ± 0.61
FLoSER-S	50.40	48.80	50.60	49.93 ± 0.99
FLoSER	50.60	49.00	50.60	50.07 ± 0.92

Table 38: Generation diagnostics under federated code-length skew. Counts aggregate 1,500 outputs per method and are descriptive rather than independent test samples.

Method	Normal EOS	Trunc.	Invalid syntax	Mean tokens
Raw	1477	23	177	64.15
FLoSER-S	1491	9	147	52.47
FLoSER	1488	12	140	52.55

Table 39: Paired MBPP outcome transitions under federated code-length skew, aggregated over three router seeds.

Comparison	Fixed	Broken	Net
Raw $\rightarrow$ FLoSER-S	108	75	+33
Raw $\rightarrow$ FLoSER	111	76	+35
FLoSER-S $\rightarrow$ FLoSER	45	43	+2

Both calibrated spectral pipelines outperform Raw for all three router seeds. FLoSER-S and FLoSER improve the mean by 2.20 and 2.33 points. The increment from FLoSER-S to FLoSER is only 0.13 points and two net seed–problem outcomes, so this setting provides no evidence of a separate response-routing gain. The comparison is limited to one fixed partition and sampling sequence, reuses the same 500 questions across seeds, and confounds spectral representation with the different fixed inference gain.