FedReCall: Recalling Client-Specific Directions in Federated LoRA Fine-tuning

Chang Liu[1,2†] **Jinqian Chen**[1,2†] **Xiangyang Yang**[1] **Jihua Zhu**[1,2*]
[1]School of Software Engineering, Xi'an Jiaotong University, Xi'an, China
[2]State Key Laboratory of Human-Machine Hybrid Augmented Intelligence
Xi'an Jiaotong University, Xi'an, China
chang.liu@stu.xjtu.edu.cn, zhujh@xjtu.edu.cn

Abstract

Low-rank adaptation (LoRA) enables communication-efficient federated fine-tuning of large language models, but standard FedLoRA averages the factors A and B separately even though the model update is determined by their product $\Delta W = sBA$. Under task heterogeneity, directions that are locally useful but weakly shared across clients may be insufficiently retained in the aggregated update; we call this selective attenuation *directional dilution*. We analyze this effect through the singular directions of client LoRA updates and find that low-strength directions are most vulnerable to dilution. Their repeated owner-specific recovery suggests client-specific adaptation signals rather than random noise. We propose FEDRECALL, a client-private plug-in that captures observable dilution gaps, selects reliable diluted directions into a private frozen cache, and recalls them through a low-rank bypass guarded by a single-batch loss probe when the client next participates. As a plug-in, FEDRECALL can be seamlessly integrated into existing FedLoRA methods without changing their aggregation rules, communication payloads, or trainable-parameter sets, consistently improving personalized adaptation and improving global evaluation metrics on most aggregators.

1 Introduction

Large language models (LLMs) are increasingly adapted to specialized tasks, but full fine-tuning remains expensive at scale [33, 11]. Low-rank adaptation (LoRA) [14] offers a practical alternative by training a compact update $\Delta W = \frac{\alpha}{r} BA$ while keeping the backbone frozen. When task data is distributed across privacy-sensitive clients, federated LoRA (FedLoRA) further reduces communication because clients exchange only lightweight adapter factors rather than full model weights [25, 9].

Existing FedLoRA methods mainly improve how LoRA factors are aggregated or decide which components remain private [39, 31, 4, 35, 2, 17, 29]. Personalized FL offers another route, but typical PFL designs introduce extra local modules, optimization stages, or personalization-specific structures [1, 32, 19, 28]. These approaches improve different parts of the overhead–capability trade-off, as summarized in Table 1, but they do not directly ask which directions aggregation actually erases.

Federated learning compresses multiple client objectives into a single shared update, so task heterogeneity naturally creates *update conflict*: directions with broad cross-client support tend to dominate, while directions that are useful to one client but weakly shared are more likely to be suppressed. Even exact averaging of effective updates need not fully retain each client's locally useful directions. Under

*Corresponding author.
†Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

factor-wise aggregation, the server averages A and B separately, whereas each client's effective update is $\Delta W = sBA$. We examine how well the aggregated update retains the directions in these local updates. We call the resulting selective attenuation of locally useful directions *directional dilution*. Our key insight is that the directions most vulnerable to aggregation are not arbitrary noise, but recurring client-specific adaptation signals. As shown in Figure 2, these directions suffer larger relative dilution, are repeatedly recovered by local training and suppressed again by aggregation, and correspond to the fragile client-specific components that standard sharing tends to smooth, distort, or cancel (Figure 2c). Existing methods do not explicitly identify and locally restore this diluted directional signal.

We propose *FedReCall* (*Federated Re-injection of Client-Adaptive Lost directions*) to answer this question with a client-private recall mechanism. After each aggregation, a participating client decomposes its effective LoRA update, measures how much the new global update retains along each local direction, and converts the missing positive strength into an observable dilution gap. Reliable diluted directions are stored as a private frozen cache and recalled as a low-rank bypass when the client next participates; a single-batch quality gate enables the bypass only when it improves the client's local loss. Each client keeps its cache locally and uploads the LoRA factors required by the base method, whose server aggregation rule remains unchanged.

In our diagnostic, the relative dilution gap for tail and floor directions falls toward zero with FedReCall. In the main experiment, all four base methods achieve higher personalized and global ROUGE-L with FedReCall (Figure 1).

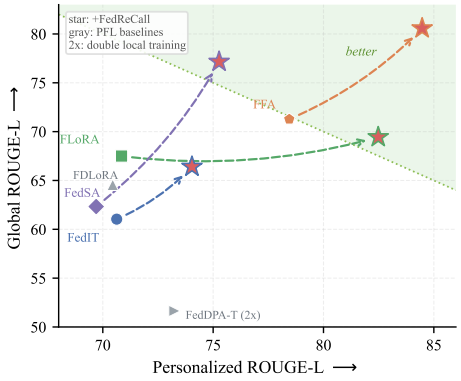


Figure 1: Performance balance between client-side and global performance.

Table 1: Comparison of FedLoRA methods.

Method	Extra-param free	Extra-optim free	Extra-comm free	Personal-ization	Plug-in
FedIT	✓	✓	✓	✗	–
FFA-LoRA	✓	✓	✓	✗	✗
FedSA-LoRA	✓	✓	✓	✓	✗
FLoRA	✓	✓	✗	✗	✗
FDLoRA	✗	✗	✓	✓	✗
FedDPA-T	✗	✗	✓	✓	✗
FedReCall	✓	✓	✓	✓	✓

Contributions.

- We uncover directional dilution, a structured form of personalization loss in which client-specific directions in local LoRA updates are insufficiently retained after aggregation.
- We propose FedReCall, a client-private recall plug-in that restores diluted directions with an observable dilution gap, a frozen low-rank bypass, and a lightweight quality gate.
- We show that FedReCall improves personalized and global performance across representative FedLoRA aggregators while preserving the original server rule, communication payload, and trainable-parameter set.

2 Related Work

Parameter-Efficient Fine-Tuning Parameter-efficient fine-tuning (PEFT) aims to adapt a large pre-trained model to downstream tasks while updating only a small subset of task-specific parameters and keeping the backbone largely frozen[11]. A variety of PEFT techniques have been explored, including prompt-based tuning[20, 18, 16], adapter-based methods[13], low-rank adaptation methods such as LoRA[14, 23, 26], and so on. In this paper, we focus on LoRA, as it has become one of the most widely used PEFT methods for large models due to its strong adaptation performance, parameter efficiency, and broad applicability across downstream tasks[14, 9, 8]. While we adopt LoRA's parameterization, our focus is on client-specific directions that are insufficiently retained after federated aggregation.

Parameter-Efficient Fine-Tuning in Federated Learning Recent work has introduced PEFT into federated learning to make the fine-tuning of pre-trained or foundation models feasible under limited client-side computation, memory, and communication budgets, since only a small subset of parameters needs to be optimized and exchanged[30, 4]. Existing studies have explored several PEFT mechanisms in federated settings, including prompt-based tuning[22], adapter-style modules[38, 36], and LoRA-based fine-tuning[4, 40, 35, 2]. Among them, LoRA has become a particularly active direction in federated fine-tuning, with prior work addressing model heterogeneity and personalization[40], aggregation accuracy for heterogeneous LoRA adapters[5], privacy-preserving and more stable LoRA optimization[31, 15], and robustness under increasing data heterogeneity[17, 4].

Personalized Federated Learning Personalized federated learning (PFL) studies how to tailor federated models to individual clients under statistical heterogeneity. Because client data distributions often differ substantially, a single global model may not perform equally well for every client[1, 7, 32, 19]. Existing PFL methods address this issue in several ways. Some split the model into globally shared and client-specific components, such as keeping a shared backbone with local personalization layers[1, 27] or learning a shared representation with client-specific heads[6, 37]. Others learn personalized models together with shared knowledge, for example by adaptively combining local and global models[7, 12], or by learning personalized weighted combinations of available models[42], decoupling personalized optimization from global model learning in a bi-level formulation[32], or improving fairness and robustness through personalization[19, 21].

3 Method

3.1 Preliminaries and notations

Consider a federated learning system with N clients, each holding a private dataset $\mathcal{D}_c$. All clients share a pre-trained backbone W_{base} and fine-tune it via LoRA [14]: each LoRA-adapted layer l adds a low-rank update $\Delta W_l = s_l B_l A_l$, where $B_l \in \mathbb{R}^{d_{\text{out}} \times r}$, $A_l \in \mathbb{R}^{r \times d_{\text{in}}}$, and $s_l = \alpha_l / r$ is the LoRA scaling factor. In each communication round t, a subset $S_t \subseteq [N]$ of clients is selected. Each selected client c downloads the current global LoRA factors $(A_l^{t,\text{glob}}, B_l^{t,\text{glob}})$, performs local training, and uploads the trained factors $(A_{c,l}^t, B_{c,l}^t)$.

The server aggregates LoRA factors using the standard factor-wise FedIT protocol:

$$A_l^{t+1,\text{glob}} = \textstyle\sum_{j \in S_t} p_j A_{j,l}^t, \qquad B_l^{t+1,\text{glob}} = \textstyle\sum_{j \in S_t} p_j B_{j,l}^t, \qquad \textstyle\sum_{j \in S_t} p_j = 1. \tag{1}$$

This produces the actual global LoRA update used in the next round,

$$\Delta W_l^{t+1,\text{glob}} = s_l B_l^{t+1,\text{glob}} A_l^{t+1,\text{glob}}. \tag{2}$$

Because the LoRA product is bilinear, factor-wise averaging does not preserve the effective update:

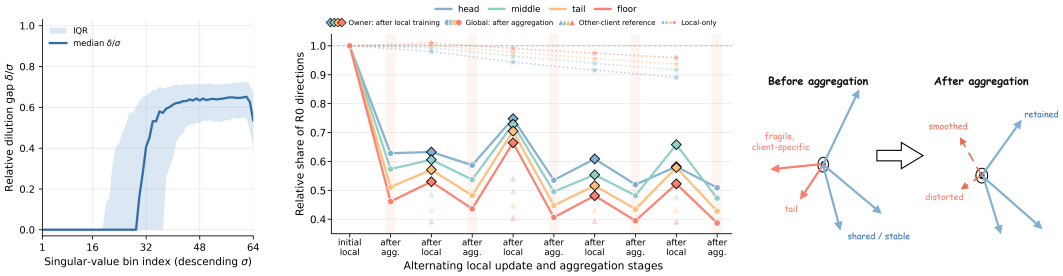
$$s_l B_l^{t+1,\text{glob}} A_l^{t+1,\text{glob}} \neq \textstyle\sum_{j \in S_t} p_j \, s_l B_{j,l}^t A_{j,l}^t \tag{3}$$

in general, meaning that averaging the factors separately need not reproduce the average of the clients' effective updates. Appendix K derives this algebraic discrepancy and clarifies why exact effective-update averaging need not preserve client-specific directions.

Direction decomposition. Each client's effective LoRA update admits a singular value decomposition

$$\Delta W_{c,l}^t = \textstyle\sum_{i=0}^{r-1} \sigma_i \, u_i v_i^\top, \tag{4}$$

where $\sigma_0 \geq \cdots \geq \sigma_{r-1} \geq 0$ are the singular values (amplitudes) and $u_i \in \mathbb{R}^{d_{\text{out}}}$, $v_i \in \mathbb{R}^{d_{\text{in}}}$ are the corresponding singular vectors. We call each rank-one atom $\phi_i = u_i v_i^\top$ a *direction* in the update space; because u_i and v_i are columns of orthonormal bases, $\|\phi_i\|_F = 1$ and distinct atoms are Frobenius-orthogonal. The Frobenius inner product $\langle M, \phi_i \rangle_F = u_i^\top M v_i$ measures how much of direction ϕ_i is present in any matrix M. We group directions by index position $q_i = i/r$ into four bands: *head* ($q < 0.2$), *middle* ($0.2 \leq q < 0.6$), *tail* ($0.6 \leq q < 0.9$), and *floor* ($q \geq 0.9$). The head band carries the largest-amplitude directions; the floor band carries the weakest non-trivial ones.



(a) (b) (c)

Figure 2: **Directional dilution in federated LoRA fine-tuning.** (a) Relative dilution gap across direction bands. (b) Recovery–disruption cycles across communication rounds. (c) Schematic illustration of directional dilution.

3.2 Motivation: directional dilution under factor-wise aggregation

Each direction ϕ_i in a client's update carries a portion of its adaptation signal, weighted by its amplitude σ_i. Under factor-wise averaging, the resulting global update does not preserve client directions equally. Our diagnostics show that low-strength (tail/floor) directions exhibit larger relative local-to-global retention gaps than high-strength (head) directions. These weak directions can contain client-specific adaptation signals, as examined below.

To quantify this directional attenuation, we define two per-direction quantities. The *global retention* of direction ϕ_i is

$$\alpha_i = \langle \Delta W_l^{t+1,\text{glob}},\, \phi_i \rangle_F = s_l\, u_i^\top B_l^{t+1,\text{glob}} (A_l^{t+1,\text{glob}} v_i), \tag{5}$$

and the *dilution gap* is the local strength missing from this projection,

$$\delta_i = \max(0,\, \sigma_i - \alpha_i^+), \qquad \alpha_i^+ = \max(\alpha_i, 0). \tag{6}$$

The relative gap $\delta_i/\sigma_i \in [0, 1]$ (for $\sigma_i > 0$) serves as a scale-invariant diagnostic; the absolute gap δ_i will be reused as the cache amplitude in Section 3.3. We make two observations: *which* directions are most affected (Obs. 1), and *what role* they play (Obs. 2).

Observation 1: Dilution is structured by within-client direction rank. Grouping directions by their rank position within each client's update, we observe that under factor-wise averaging, the median relative local-to-global retention gap δ_i/σ_i increases monotonically from head to floor (Figure 2).

Observation 2: The diluted directions encode client-specific adaptation knowledge. Three independent lines of evidence show that the directions most affected by aggregation carry functionally meaningful, client-owned signal. *(i) Persistence under local training.* When aggregation is turned off (dashed lines in Figure 2), the band ordering *reverses*: tail and floor directions are preserved best, while head and middle decay fastest. The persistence of tail/floor directions under local training is consistent with their carrying useful local adaptation signal. *(ii) Owner specificity.* Triangles in Figure 2 show that a direction defined by client c is recovered better by c in subsequent rounds than by any other client, with the specificity gap largest in tail/floor. These directions therefore encode adaptation signal that is client-specific rather than population-shared. *(iii) Direction-blind validation.* Table 6 shows that the gate accepts 78% of gap-weighted tail/floor cache injections on average, while rejecting all tested head/middle injections. This direction-blind asymmetry supports the usefulness of tail/floor cache corrections. Held-out frozen-model interventions further support useful, owner-specific information in the tested weak directions (Appendix J).

When weak directions contain client-specific signal. For a fixed layer and round, write the effective update as

$$X_c = S + P_c + E_c, \qquad \varepsilon_c = \|E_c\|_2, \tag{7}$$

where S, P_c, and E_c denote shared signal, structured client-specific signal, and perturbations, respectively.

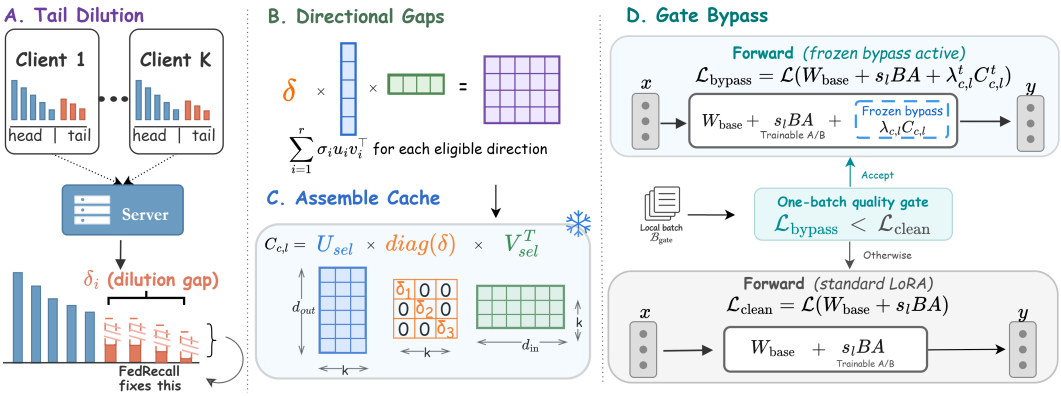


Figure 3: **Overview of FedReCall.** FedReCall targets observable dilution gaps in weak client-specific directions. After aggregation, each client estimates the missing directional strength, assembles the reliable gaps into a private frozen cache, and reuses the cache as a low-rank bypass in later local training. A one-batch gate enables the bypass only when it lowers the client's local loss; otherwise the client falls back to standard LoRA.

Proposition 1. Assume that the singular vectors associated with nonzero singular values of S and P_c are orthogonal across the two components on both the left and the right. If

$$\sigma^+_{\min}(S) > \sigma_{\max}(P_c) + 2\varepsilon_c, \qquad \sigma^+_{\min}(P_c) > 2\varepsilon_c, \tag{8}$$

then Weyl's bound places the private-associated singular values of X_c between the shared and perturbation-only groups. Here $\sigma^+_{\min}$ is the smallest nonzero singular value.

For normalized weights with $\sum_j p_j P_j = 0$, private signal cancels even under exact averaging:

$$\bar{X}_{\text{exact}} = \sum_j p_j X_j = S + \sum_j p_j E_j. \tag{9}$$

Together, these observations show that client-specific adaptation signals carried by tail/floor directions are not fully retained in the aggregated update. The persistent zigzag in Figure 2 further shows that this knowledge loss is not a one-time event: these directions are repeatedly recovered by local training and attenuated by subsequent aggregation, creating an avoidable recovery cost across participations. We retain these diluted directions as private memory on their originating clients while preserving the server aggregation rule.

3.3 FedReCall: client-private directional recall

Guided by the dilution analysis above, FedReCall protects diluted tail/floor directions as client-private memory rather than as shared parameters. It is a personalization plugin on top of standard factor-wise FedIT: clients still upload only clean LoRA factors, and the server aggregation rule is unchanged. After each aggregation a participating client builds a correction cache (**cache construction**); when next selected, it loads the cache as a frozen bypass during local adaptation (**bypass injection**), gated against stale caches by a one-batch loss probe (**quality gate**).

Cache construction. After client c completes local training in round t and receives the new post-aggregation global factors $(A_l^{t+1,\text{glob}}, B_l^{t+1,\text{glob}})$, it builds the correction cache in three steps. First, the client decomposes its *effective update* via SVD to obtain the singular triplets $\{(\sigma_i, u_i, v_i)\}_{i=0}^{n_{\text{svd}}-1}$, where n_{svd} is the number of triplets returned before numerical filtering. The same spectral bands defined in Section 3.1 are applied using $q_i = i/n_{\text{svd}}$. The effective update captures both the trained LoRA factors and any active bypass from the current round,

$$\widetilde{\Delta W}^t_{c,l} = s_l B^t_{c,l} A^t_{c,l} + m^t_c \lambda^t_{c,l} C^{\text{prev}}_{c,l}, \tag{10}$$

where $m^t_c \in \{0,1\}$ indicates whether the bypass was enabled in round t and $C^{\text{prev}}_{c,l}$ is the cache from the client's most recent prior participation; for a first-time client, $m^t_c = 0$ and the effective update

reduces to the trained LoRA factors alone. Second, it computes the global retention α_i and dilution gap δ_i (Eqs. 5–6) for each direction, using the Frobenius projection defined in Section 3.2; the value δ_i estimates how much client signal is missing from the current global model along direction ϕ_i and becomes the amplitude stored in the cache. Third, motivated by the finding that tail/floor directions are both more strongly diluted and more client-specific (Section 3.2), it selects the eligible tail/floor set:

$$I_{c,l}^{t,\mathrm{elig}} = \big\{\, i \,:\, \delta_i > 0,\ \sigma_i \geq \tau_l,\ i \in \mathrm{tail} \cup \mathrm{floor} \,\big\}, \tag{11}$$

where $\tau_l = \epsilon_{\mathrm{rank}} \sigma_{l,\max}$ with $\epsilon_{\mathrm{rank}} = 10^{-4}$ excludes near-zero directions, and the index i is restricted to the tail and floor bands defined in Section 3.1. The private correction cache for layer l is

$$C_{c,l}^{t} = \textstyle\sum_{i \in I_{c,l}^{t,\mathrm{elig}}} \delta_i \, u_i v_i^{\top}. \tag{12}$$

The cache stores the selected singular triplets, and SVD is computed via QR-compressed low-rank algebra (Appendix D).

Bypass injection and calibration. When client c is next selected, it loads the cached correction and injects it as a frozen additive bypass in each layer's forward pass:

$$y = W_{\mathrm{base}} x + s_l B_l A_l x + \lambda_{c,l} C_{c,l} x. \tag{13}$$

During local training, only A_l and B_l are optimized; $C_{c,l}$ is frozen and excluded from the optimizer, and the gradient pathways to A_l and B_l are unchanged (Appendix E). The bypass exploits the low-rank structure of $C_{c,l}$:

$$C_{c,l} x = U_{\mathrm{sel}} \big[\mathrm{diag}(\boldsymbol{\delta}) (V_{\mathrm{sel}}^{\top} x) \big], \tag{14}$$

where U_{sel} and V_{sel} stack the selected left and right singular vectors. The fixed hyperparameter η sets the target bypass-to-LoRA norm ratio at the start of each participation. Calibration uses the downloaded global LoRA factors and the stored cache, and the resulting $\lambda_{c,l}$ remains fixed during local training:

$$\lambda_{c,l} = \eta \cdot \frac{\| s_l B_l A_l \|_F}{\| C_{c,l} \|_F + \epsilon}. \tag{15}$$

Because the selected direction atoms are Frobenius-orthonormal, $\|C_{c,l}\|_F = \|\boldsymbol{\delta}\|_2$, which can be computed without materializing the full correction matrix. For nonzero LoRA updates and cache norms well above ϵ, this calibration sets the initial bypass-to-LoRA norm ratio approximately to η (derivation in Appendix F). A single global η thus suffices without per-layer tuning ($\eta = 0$ recovers standard federated LoRA).

Algorithm 1 FedReCall: client c, round t

Require: $\eta > 0$
Input: $\{A_l^t, B_l^t\}_l,\ \mathcal{D}_c$; optional C_c^{prev}
// Phase 1: Bypass injection
if C_c^{prev} exists **then**
  $\lambda_{c,l}^t \leftarrow \eta \| s_l B_l^t A_l^t \|_F / (\| C_{c,l}^{\mathrm{prev}} \|_F + \epsilon)$
  $m_c^t \leftarrow \mathbf{1}[\mathcal{L}(\theta, \lambda^t C^{\mathrm{prev}}; \mathcal{B}) < \mathcal{L}(\theta, \mathbf{0}; \mathcal{B})]$
  if $m_c^t = 1$ **then**
    $y \mathrel{+}= \lambda_{c,l}^t C_{c,l}^{\mathrm{prev}} x$ (frozen)
  end if
else
  $m_c^t \leftarrow 0$ (bootstrap)
end if
// Phase 2: Local training
$\{(A_{c,l}^t, B_{c,l}^t)\}_l \leftarrow \mathrm{SGD}(\{A_l^t, B_l^t\}_l; \mathcal{D}_c)$
Upload $\{(A_{c,l}^t, B_{c,l}^t)\}_l$
// Phase 3: Cache construction
Receive $\{A_l^{t+1}, B_l^{t+1}\}_l$; compute $\widetilde{\Delta W}_{c,l}^{t}$
$\{(\sigma_i, u_i, v_i)\}_{i=0}^{n_{\mathrm{svd}}-1} \leftarrow \mathrm{QRSVD}(\widetilde{\Delta W}_{c,l}^{t})$
$\alpha_i \leftarrow \langle \Delta W_l^{t+1,\mathrm{glob}}, \phi_i \rangle_F$; $\delta_i \leftarrow \max(0, \sigma_i - \alpha_i^{+})$
Select $I_{c,l}^t$ using Eq. 11.
$C_{c,l}^t \leftarrow \sum_{i \in I_{c,l}^t} \delta_i u_i v_i^{\top}$; store locally

One-batch quality gate. A client's cached correction may become stale if the global model has evolved substantially since the cache was built. Before enabling the bypass, FedReCall evaluates a single local mini-batch $\mathcal{B}_{\mathrm{gate}}$ under both the model without bypass and the bypassed model:

$$m_c^t = \mathbf{1}\big[\mathcal{L}\big(\theta^{\mathrm{LoRA}},\, \Lambda_c \, \mathbf{C}_c;\, \mathcal{B}_{\mathrm{gate}}\big) < \mathcal{L}\big(\theta^{\mathrm{LoRA}},\, \mathbf{0};\, \mathcal{B}_{\mathrm{gate}}\big) \big], \tag{16}$$

where $\theta^{\mathrm{LoRA}} = \{(A_l, B_l)\}_l$ denotes the shared LoRA factors, $\Lambda_c \, \mathbf{C}_c = \{\lambda_{c,l} \, C_{c,l}\}_l$ denotes the per-layer frozen bypass, and $\mathbf{0}$ indicates the bypass is disabled. If the gate rejects the cache, the client trains with standard LoRA for that round. The gate's overhead is a single forward pass per model variant.

The full per-round procedure is summarized in Algorithm 1; beyond standard federated LoRA, FedReCall adds only the client-side cache construction (Phase 3) and bypass injection (Phase 1), with the server aggregation unchanged and a single hyperparameter η.

Table 2: Main performance on the task-heterogeneous federation. Sample standard deviations are included where available.

	Personalized client model			Global model	
Method	Avg Pers. Loss ↓	Worst Loss ↓	Pers. ROUGE-L ↑	Avg Loss ↓	ROUGE-L ↑
FedIT	1.225 ±0.041	2.201 ±0.107	70.63 ±2.31	2.252 ±0.087	61.04 ±3.59
+ FedReCall	**1.035 ± 0.070**	**2.130 ± 0.094**	**74.04 ± 2.57**	**2.029 ± 0.027**	**66.41 ± 2.31**
FLoRA	1.273 ±0.063	1.460 ±0.069	70.85 ±2.84	2.397 ±0.083	67.50 ±3.49
+ FedReCall	**1.146 ± 0.050**	**1.406 ± 0.039**	**82.48 ± 1.66**	**1.807 ± 0.072**	**69.44 ± 3.14**
FFA-LoRA	1.345 ±0.063	1.960 ±0.099	78.45 ±2.66	1.985 ±0.049	71.28 ±3.27
+ FedReCall	**1.089 ± 0.068**	**1.554 ± 0.119**	**84.47 ± 2.10**	**1.244 ± 0.110**	**80.58 ± 1.53**
FedSA-LoRA	1.291 ±0.037	2.168 ±0.071	69.70 ±1.34	2.045 ±0.065	62.33 ±2.49
+ FedReCall	**1.236 ± 0.017**	**2.164 ± 0.054**	**75.27 ± 1.12**	**1.414 ± 0.060**	**77.16 ± 1.88**

4 Experiments

4.1 Experimental setup

Following Bai et al. [2], we construct a task-heterogeneous federated setting from the Super-Natural Instructions (SuperNI) dataset [34]. The full pool contains 511 NLP tasks from 76 task types; each client holds a single task. We randomly draw 50 clients (seed 42) as the training federation; details of the client selection are provided in Appendix G. This federation is intentionally heterogeneous: each client corresponds to a distinct SuperNI task, the selected 50-client cohort spans 23 task types, and client data volumes are highly uneven. The base model is DataJuicer-LLaMA-1.3B [33, 3], fine-tuned with LoRA [14] (r=32); all methods share the same rank for fair comparison. Unless otherwise stated, all reported numbers are averaged over three training seeds. We compare against two families of baselines. *(i) Aggregation-side methods*—FedIT [41], FLoRA [35], FFA-LoRA [31], and FedSA-LoRA [10]—differ in how they aggregate LoRA factors; FedReCall is layered on top of each as a plug-in (Table 2). *(ii) Dual-LoRA personalization methods*—FedDPA-T [38] and FDLoRA [24]—each maintain a global and a local adapter; they provide comparisons of personalized performance and zero-shot generalization (Table 3).

Table 3: Personalization and zero-shot generalization. Entries report loss ↓ / ROUGE-L ↑. Recall denotes FedReCall.

Method	Personalized	Zero-shot
FedIT + Recall	**1.035** / 74.04	2.19 / 51.14
FFA-LoRA + Recall	1.089 / **84.47**	**1.95 / 55.62**
FedDPA-T (1×)	1.809 / 57.51	2.09 / 52.69
FedDPA-T (2×)	1.177 / 73.25	2.25 / 51.86
FDLoRA (1×)	1.326 / 70.45	2.42 / 50.41

Personalized metrics (*Avg Pers. Loss*, *Avg Pers. ROUGE-L*) measure each client's post-local-adaptation performance. *Global* metrics (*Avg Loss*, *Avg ROUGE-L*) assess the global model before any local adaptation, tracking how the plug-in influences the shared model.

4.2 Main results: FedReCall as a plug-in

Table 2 compares four aggregation-side baselines with and without FedReCall on the 50-client SuperNI-derived federation. We report both client-side metrics after local adaptation and global-model metrics before any local adaptation. FedReCall uses the same candidate grid for its bypass-strength parameter η across all plug-in settings. The full sweep is reported in Appendix H; no per-client or per-layer η tuning is used.

FedReCall consistently improves personalized adaptation as a plug-in. Table 2 shows that adding FedReCall improves average personalized loss and personalized ROUGE-L for every aggregation-side baseline. The improvement is not limited to the standard FedIT setting: it also appears when the base method changes which LoRA factors are shared, fixed, or aggregated. FedSA-LoRA, for example, keeps B local, which can reduce the exposure of B-side client-specific information to direct averaging. However, this is a factor-level design choice, whereas FedReCall

identifies and recalls diluted directions in the effective update $\Delta W = BA$. The strongest gains occur for FFA-LoRA and FLoRA, while FedSA-LoRA shows a smaller but still positive personalized improvement. This pattern is consistent with the view that FedReCall is most useful when the base aggregator leaves an observable directional gap to repair; Section 4.3 examines this interpretation through the η sensitivity and fragility diagnostics. In the same-round paired analysis, FedReCall lowers average loss in multiple client groups (Appendix C).

FedReCall improves global model quality alongside personalization. FedReCall improves global model quality across all four aggregators while preserving the base aggregation rule. We interpret these gains as a benefit of retaining recurring client-specific corrections locally, allowing the uploaded clean LoRA factors to devote more capacity to shared adaptation. FFA-LoRA shows the largest relative reduction in global loss after adding FedReCall (37%), alongside a 9.3-point increase in global ROUGE-L. These paired gains show that client-private recall can improve both personalized adaptation and global-model quality. FedReCall introduces additional client-side computation for cache construction and gate evaluation; detailed wall-clock measurements are reported in Appendix I.

Personalization and zero-shot generalization. Table 3 compares two FedReCall configurations with dual-adapter personalization methods. Both configurations outperform the dual-adapter baselines on personalized loss and ROUGE-L, including FedDPA-T with twice the local-training budget. FFA-LoRA + FedReCall also achieves the best zero-shot loss and ROUGE-L in this comparison. Zero-shot metrics evaluate the final shared model on unseen task types without local adaptation or private recall. These results support a favorable balance between personalization and shared-model generalization without a second trainable adapter.

Table 4: FedReCall component ablation.

Bypass	Gate	Avg. Pers. loss ↓	Global loss ↓	Δ vs. FedIT
–	–	1.225	2.252	–
✓	–	1.115	2.329	−9.0%
✓	✓	**1.035**	**2.029**	−15.5%

4.3 Ablation and analysis

We examine how FedReCall's design choices contribute to directional repair and personalized performance. The ablations follow the mechanism in Section 3.2: a useful correction should restore diluted directions locally, reject stale private signal, and operate on the effective update ΔW where dilution is defined.

Direction-specific functional value. Figure 4 examines whether the selected weak directions contain useful, client-specific information. We test their effects through restoration, removal, and sign reversal, without gate selection or retraining. Restoration generally yields greater gains for selected than random directions (a), and for owners than other clients on average (b). Removing selected directions is more harmful than removing random directions (c), while sign reversal generally increases loss relative to the original correction (d). Appendix J reports practical evaluations of the complete bypass.

Table 5: Recall-operand ablation on FFA-LoRA.

Recall operand	Avg. Pers. Loss ↓	ROUGE-L ↑
None	1.345	78.45
B	1.344	77.83
BA	**1.089**	**84.47**

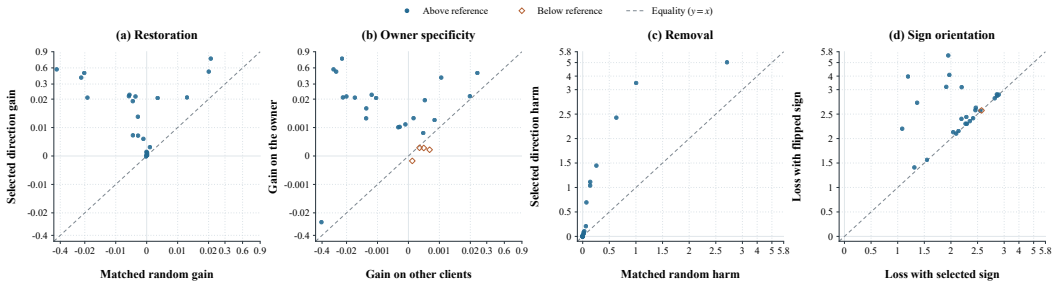


Figure 4: **Restoring, removing, and reversing selected directions.** Each point represents one client.

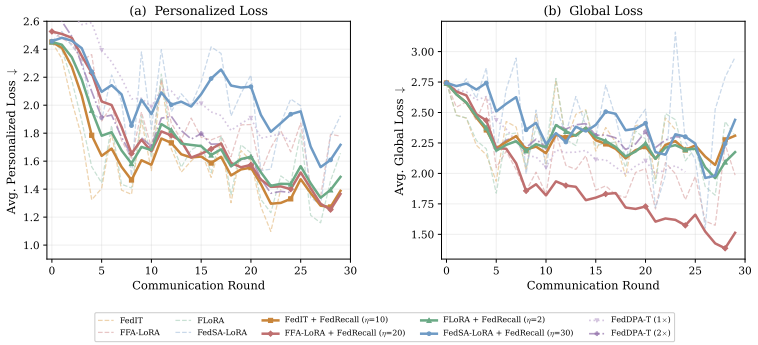


Figure 5: **Training dynamics with and without FedReCall.** Dashed curves are base aggregators; solid curves add FedReCall, with EMA smoothing used only for visualization.

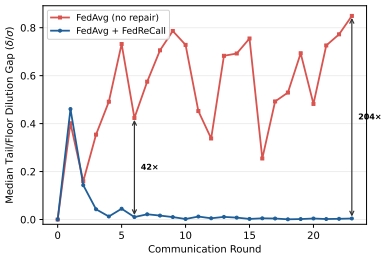


Figure 6: **Fragility repair.** Our method reduces tail/floor dilution.

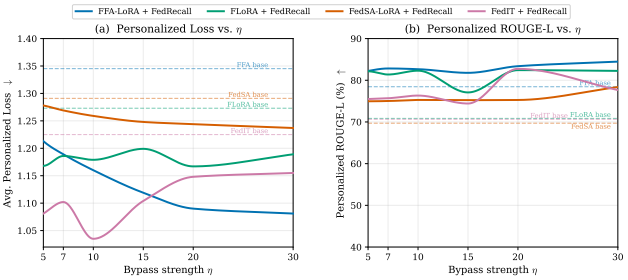


Figure 7: **Bypass strength sensitivity.** The useful η range depends on the base aggregator.

Recall operand. Table 5 compares recall constructed from an individual factor with recall constructed from the effective update. The naive B-only variant provides no benefit, whereas BA-based recall improves both loss and ROUGE-L. This supports measuring directional gaps in the effective update rather than directly in a single factor.

Component ablation and gate validation. The frozen bypass and the quality gate play different roles. The bypass supplies the missing directions during local adaptation, which accounts for most of the personalized improvement. The gate validates the cached correction as a whole, enabling it only when it lowers loss on the client's probe batch. This matters because early caches can be stale or over-amplified; without the gate, recall remains helpful for personalization but can degrade the shared model. Table 6 in Appendix A shows that enabling the bypass with gap-weighted tail/floor caches lowered loss on average, and the gate often accepted these caches. For the tested head/middle caches, enabling the bypass increased loss on average, and the gate rejected all of them.

Training dynamics and fragility repair. Figure 5 traces the per-round personalized and global loss across all compared methods. Baselines without FedReCall show round-to-round oscillations consistent with the recovery–disruption cycle in Section 3.2; FedReCall dampens these oscillations and converges to lower loss. Figure 6 tracks the same relative dilution gap as Figure 2(a) across rounds with FedReCall enabled. The gap decreases toward zero, showing reduced directional dilution during training.

Sensitivity and robustness. Figure 7 sweeps the bypass strength η. FedReCall remains effective over a non-trivial range of η, with the preferred value tracking how much directional dilution the base aggregator leaves: aggregators with stronger dilution benefit from stronger recall, while aggregators that already preserve more private signal peak at moderate η. This indicates η is a coarse global compensation knob rather than a narrow optimum. Appendix B reports additional diagnostics over LoRA rank and client participation rate. The rank sweep checks whether recall becomes more useful as the adapter exposes more weak local directions, while the participation-rate sweep checks whether the effect persists when more clients are selected per communication round.

5 Conclusion

Taken together, our analysis and experiments identify directional dilution as a structured form of personalization loss in federated LoRA: under task heterogeneity, locally useful directions that are weakly shared across clients are insufficiently retained after aggregation. FedReCall addresses this gap through selective, client-private recall across rounds. It estimates observable dilution gaps after aggregation, stores reliable missing directions in a private frozen cache, and recalls them through a gated low-rank bypass during later local adaptation. Because the cache is neither uploaded nor optimized, FedReCall leaves the server aggregation rule, communication payload, and trainable-parameter set unchanged.

Across representative FedLoRA aggregators, the results show that selectively restoring diluted client-specific directions improves personalized adaptation and also strengthens the global model in the main experiments, while remaining competitive on unseen zero-shot evaluation. Our evaluation focuses on federated LoRA instruction tuning; broader applicability and formal privacy guarantees for client-specific memory remain open.

Acknowledgments and Disclosure of Funding

This work was supported by The Natural Science Basis Research Plan in Shaanxi Province of China under Grant No. 2026JC-QYCX-061.

References

[1] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[2] Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. Federated fine-tuning of large language models under heterogeneous tasks and client resources. *Advances in Neural Information Processing Systems*, 37:14457–14483, 2024.

[3] Daoyuan Chen, Yilun Huang, Zhijian Ma, Hesen Chen, Xuchen Pan, Ce Ge, Dawei Gao, Yuexiang Xie, Zhaoyang Liu, Jinyang Gao, et al. Data-juicer: A one-stop data processing system for large language models. In *Companion of the 2024 International Conference on Management of Data*, pages 120–134, 2024.

[4] Shuangyi Chen, Yue Ju, Hardik Dalal, Zhongwen Zhu, and Ashish Khisti. Robust federated finetuning of foundation models via alternating minimization of lora. *arXiv preprint arXiv:2409.02346*, 2024.

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

[6] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pages 2089–2099. PMLR, 2021.

[7] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

[8] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

[9] Haiyang Guo, Fanhu Zeng, Fei Zhu, Wenzhuo Liu, Da-Han Wang, Jian Xu, Xu-Yao Zhang, and Cheng-Lin Liu. Federated continual instruction tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1325–1335, 2025.

[10] Pengxin Guo, Shuang Zeng, Yanran Wang, Huijie Fan, Feifei Wang, and Liangqiong Qu. Selective aggregation for low-rank adaptation in federated learning. In *International Conference on Learning Representations*, volume 2025, pages 99003–99027, 2025.

[11] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

[12] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020.

[13] 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 *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

[14] 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. *arXiv preprint arXiv:2106.09685*, 2021.

[15] Zhu JianHao, Changze Lv, Xiaohua Wang, Muling Wu, Wenhao Liu, Tianlong Li, Zixuan Ling, Cenyuan Zhang, Xiaoqing Zheng, and Xuan-Jing Huang. Promoting data and model privacy in federated learning through quantized lora. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10501–10512, 2024.

[16] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19113–19122, 2023.

[17] Jabin Koo, Minwoo Jang, and Jungseul Ok. Towards robust and efficient federated low-rank adaptation with heterogeneous clients. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 416–429, 2025.

[18] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 3045–3059, 2021.

[19] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, pages 6357–6368. PMLR, 2021.

[20] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021.

[21] Shiyun Lin, Yuze Han, Xiang Li, and Zhihua Zhang. Personalized federated learning towards communication efficiency, robustness and fairness. *Advances in Neural Information Processing Systems*, 35:30471–30485, 2022.

[22] Zihao Lin, Yan Sun, Yifan Shi, Xueqian Wang, Lifu Huang, Li Shen, and Dacheng Tao. Efficient federated prompt tuning for black-box large pre-trained models. *arXiv preprint arXiv:2310.03123*, 2023.

[23] 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.

[24] Yao Lu, Jiaxing Qi, Zhongzhi Luan, Shaohan Huang, Carol Fung, Hailong Yang, and Depei Qian. Fdlora: Personalized federated learning of large language model via dual lora tuning. *arXiv preprint arXiv:2406.07925*, 2024.

[25] 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*, pages 1273–1282. Pmlr, 2017.

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

[27] Krishna Pillutla, Kshitiz Malik, Abdel-Rahman Mohamed, Mike Rabbat, Maziar Sanjabi, and Lin Xiao. Federated learning with partial model personalization. In *International Conference on Machine Learning*, pages 17716–17758. PMLR, 2022.

[28] Zhanming Shen, Tianqi Xu, Hao Wang, Jian Li, and Miao Pan. pFedGPT: Hierarchically optimizing LoRA aggregation weights for personalized federated GPT models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4766–4778, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.239. URL `https://aclanthology.org/2025.emnlp-main.239/`.

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

[30] Guangyu Sun, Umar Khalid, Matias Mendieta, Pu Wang, and Chen Chen. Exploring parameter-efficient fine-tuning to enable foundation models in federated learning. In *2024 IEEE International Conference on Big Data (BigData)*, pages 8015–8024. IEEE, 2024.

[31] Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving lora in privacy-preserving federated learning. *arXiv preprint arXiv:2403.12313*, 2024.

[32] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in neural information processing systems*, 33:21394–21405, 2020.

[33] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[34] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 5085–5109, 2022.

[35] Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. *Advances in Neural Information Processing Systems*, 37:22513–22533, 2024.

[36] Baochen Xiong, Xiaoshan Yang, Yaguang Song, Yaowei Wang, and Changsheng Xu. Pilot: Building the federated multimodal instruction tuning framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 21716–21724, 2025.

[37] Jian Xu, Xinyi Tong, and Shao-Lun Huang. Personalized federated learning with feature alignment and classifier collaboration. *arXiv preprint arXiv:2306.11867*, 2023.

[38] Yiyuan Yang, Guodong Long, Tao Shen, Jing Jiang, and Michael Blumenstein. Dual-personalizing adapter for federated foundation models. *Advances in Neural Information Processing Systems*, 37:39409–39433, 2024.

[39] Yiyuan Yang, Guodong Long, Qinghua Lu, Liming Zhu, Jing Jiang, and Chengqi Zhang. Federated low-rank adaptation for foundation models: A survey. *arXiv preprint arXiv:2505.13502*, 2025.

[40] Liping Yi, Han Yu, Gang Wang, Xiaoguang Liu, and Xiaoxiao Li. pfedlora: Model-heterogeneous personalized federated learning with lora tuning. *arXiv preprint arXiv:2310.13283*, 2023.

[41] Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP 2024-2024 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 6915–6919. IEEE, 2024.

[42] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. Personalized federated learning with first order model optimization. *arXiv preprint arXiv:2012.08565*, 2020.

A Gate behaviour details

Table 6 examines gate behavior for corrections constructed from different spectral ranges. Low-strength corrections reduce loss on average with either gap-based or uniform amplitudes, whereas the tested high/mid-strength corrections increase loss and are rejected. Here, *Bypassed loss* evaluates the candidate with the bypass enabled, including rejected candidates; it is not the deployed loss after gating. These results illustrate that spectral selection defines the candidate pool, while the loss-based gate determines whether recall is enabled.

Table 6: Gate behaviour under direction-strength candidate pools.

Candidate range	Gate pass rate	No-bypass loss	Bypassed loss
Low-strength range ($q_i \geq 0.6$)	0.78 ± 0.19	2.00 ± 0.04	1.95 ± 0.04
Low-strength range, uniform amplitude	1.00 ± 0.00	2.07 ± 0.10	1.98 ± 0.07
High/mid-strength range ($q_i < 0.6$)	0.00 ± 0.00	1.96 ± 0.11	11.51 ± 0.33

B Robustness diagnostics

Table 7 reports a LoRA-rank diagnostic using the same federation and client-selection seed. Results for the default rank ($r = 32$) are reported in Table 2 and are omitted here. The comparison asks whether directional recall is tied to the default rank or becomes more useful as the adapter exposes more weak local directions. Table 8 complements the default 10% participation setting by increasing the number of clients selected in each round. Results for the default 10% participation rate are reported in Table 2 and are omitted here. This diagnostic keeps the same 50-client federation and client-selection seed, runs for 15 communication rounds, and compares all runs at round 10.

Table 7: LoRA-rank robustness diagnostic.

Rank	Method	Avg Pers. Loss ↓	Pers. ROUGE-L ↑
16	FedIT	1.146	83.96
	+ FedReCall	1.144	83.55
64	FedIT	1.039	76.18
	+ FedReCall	0.960	77.88

Table 8: Participation-rate robustness diagnostic.

Part.	Method	Avg Pers. Loss ↓	Pers. ROUGE-L ↑
20%	FedIT	1.39	68.61
	+ FedReCall	1.39	70.28
30%	FedIT	1.401	66.32
	+ FedReCall	1.448	67.60

C Same-round subgroup diagnostic

We compare each base method and its FedReCall variant on the same five clients in a selected round. For each paired client, we compute ΔLoss as base loss minus FedReCall loss and ΔROUGE-L as FedReCall ROUGE-L minus base ROUGE-L. We then group the paired clients by data volume and by baseline difficulty.

D QR-compressed SVD for cache construction

The cache construction requires the SVD of the effective update $\widetilde{\Delta W}_{c,l}^{t} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ (Eq. 10), which can be prohibitively expensive if materialized as a dense matrix. We exploit the low-rank factored form to avoid this.

Table 9: Same-round paired subgroup diagnostic. Positive ΔLoss means FedReCall lowers client loss. The last two columns count how many data-volume and baseline-difficulty groups have positive mean ΔLoss.

Base method	Paired clients	ΔLoss ↑	ΔROUGE-L ↑	Data-volume groups	Difficulty groups
FedIT	5	0.2881	5.70	3/3	3/3
FLoRA	5	0.0127	0.45	2/3	2/3
FFA-LoRA	5	0.2554	4.93	3/3	3/3
FedSA-LoRA	5	0.1016	3.09	2/3	2/3

When no bypass is active ($m_c^t = 0$), the effective update is $s_l B_{c,l}^t A_{c,l}^t$, which already has rank at most r. When the bypass is active, we have

$$\widetilde{\Delta W}_{c,l}^t = \underbrace{[\, s_l B_{c,l}^t \;\mid\; \lambda_{c,l}^t U_{\mathrm{sel}}^{\mathrm{prev}} \mathrm{diag}(\boldsymbol{\delta}^{\mathrm{prev}}) \,]}_{L \in \mathbb{R}^{d_{\mathrm{out}} \times (r+k)}} \underbrace{\begin{bmatrix} A_{c,l}^t \\ V_{\mathrm{sel}}^{\mathrm{prev}\top} \end{bmatrix}}_{R \in \mathbb{R}^{(r+k) \times d_{\mathrm{in}}}},$$

where $k = |I_{c,l}^{\mathrm{prev}}|$ is the number of cached tail/floor directions (at most $|\mathrm{tail}| + |\mathrm{floor}|$). The product LR has rank at most $r + k$, and its SVD can be obtained without forming the $d_{\mathrm{out}} \times d_{\mathrm{in}}$ matrix:

1. Compute a thin QR factorization $L = Q_L R_L$, where $Q_L \in \mathbb{R}^{d_{\mathrm{out}} \times (r+k)}$ has orthonormal columns and $R_L \in \mathbb{R}^{(r+k)\times(r+k)}$.
2. Form the small matrix $M = R_L R \in \mathbb{R}^{(r+k)\times d_{\mathrm{in}}}$.
3. Compute the SVD $M = U_M \Sigma V^\top$.
4. The left singular vectors of $\widetilde{\Delta W}_{c,l}^t$ are $Q_L U_M$; the singular values and right singular vectors are Σ and V.

All operations involve matrices of size at most $(r + k) \times \max(d_{\mathrm{out}}, d_{\mathrm{in}})$, giving a total cost of $O\big((r + k)^2(d_{\mathrm{out}} + d_{\mathrm{in}})\big)$ per layer, compared to $O(d_{\mathrm{out}} d_{\mathrm{in}} \min(d_{\mathrm{out}}, d_{\mathrm{in}}))$ for a dense SVD.

E Gradient flow through the frozen bypass

The frozen bypass contributes a cached correction to the layer output, while the LoRA factors are updated through backpropagation.

The bypassed forward pass for layer l is (Eq. 13)

$$y = W_{\mathrm{base}} x + s_l B_l A_l x + \lambda_{c,l} C_{c,l} x.$$

For a gate-accepted bypass, $C_{c,l}$ and $\lambda_{c,l}$ remain fixed during local training. The layer differential is

$$dy = (W_{\mathrm{base}} + s_l B_l A_l + \lambda_{c,l} C_{c,l})\, dx + s_l (dB_l) A_l x + s_l B_l (dA_l) x.$$

Let $g = \nabla_y \mathcal{L}$ denote the gradient arriving at this layer. The parameter gradients are

$$\nabla_{A_l} \mathcal{L} = s_l B_l^\top g x^\top, \qquad \nabla_{B_l} \mathcal{L} = s_l g (A_l x)^\top.$$

These expressions retain the standard LoRA form, with g evaluated at the bypassed output. The bypass term $\lambda_{c,l} C_{c,l} x$ affects the loss $\mathcal{L}$ by changing the output y, and therefore changes the upstream gradient $\partial \mathcal{L} / \partial y$.

In effect, the frozen bypass shifts the output baseline from which LoRA operates: during local adaptation, the LoRA factors need not spend capacity re-learning the directions already restored by the cache, and can focus on further task-specific adaptation. No stop-gradient is applied to the shared input x, so gradient information flows freely to all layers below.

F Bypass strength calibration derivation

We show that the per-layer scaling $\lambda_{c,l}$ in Eq. 15 achieves the target bypass-to-LoRA norm ratio η up to negligible ϵ-stabilizer terms.

Setup. The cache admits the factorization $C_{c,l} = U_{\text{sel}} \operatorname{diag}(\boldsymbol{\delta}) V_{\text{sel}}^{\top}$, where the columns of U_{sel} and V_{sel} are subsets of orthonormal singular vectors. Because distinct direction atoms $\phi_i = u_i v_i^{\top}$ are Frobenius-orthogonal,

$$\|C_{c,l}\|_F^2 = \sum_{i \in I_{c,l}^{t,\text{elig}}} \delta_i^2 = \|\boldsymbol{\delta}\|_2^2.$$

Derivation. Substituting Eq. 15 into the bypass norm:

$$\|\lambda_{c,l} C_{c,l}\|_F = \lambda_{c,l} \|C_{c,l}\|_F = \eta \cdot \frac{\|s_l B_l A_l\|_F}{\|C_{c,l}\|_F + \epsilon} \cdot \|C_{c,l}\|_F.$$

Dividing by $\|s_l B_l A_l\|_F$:

$$\frac{\|\lambda_{c,l} C_{c,l}\|_F}{\|s_l B_l A_l\|_F} = \eta \cdot \frac{\|C_{c,l}\|_F}{\|C_{c,l}\|_F + \epsilon}.$$

When $\|C_{c,l}\|_F \gg \epsilon$, the ratio approaches η. The ϵ term safeguards division by zero.

G Data preparation and client selection

Client pool. Following Bai et al. [2], we use Super-Natural Instructions [34] to construct a task-heterogeneous federation of 511 clients covering 76 NLP task types. Each client holds a single task, and within-task training data is subsampled to 10% following the original FlexLoRA protocol.

50-client subset. From the 511-client pool we draw 50 clients uniformly at random (`random.sample(range(511), 50)`, seed=42) and re-index them as clients $0, \ldots, 49$. A deterministic mapping from new to original client IDs is stored for reproducibility. The selected 50 clients span 23 NLP task types; per-client training-set sizes range from 16 to 520 samples (median 127, mean 232).

Per-round participation. In each communication round, 5 clients (10%) are sampled without replacement. The client-selection seed is fixed to 42 and shared across all training seeds $\{42, 43, 44\}$, so that every run uses an identical participation schedule. This design isolates the reported variance to training randomness alone, excluding participation-schedule noise.

H Bypass strength selection

Table 10 reports a single-seed diagnostic sweep over the bypass-strength parameter η (seed 42, q=0.6, 30 rounds). This sweep examines how bypass strength affects personalized performance across different base aggregators. Since η is applied through the norm calibration in Eq. 15, it acts as one coarse global compensation factor across layers and clients.

Table 10: Average personalized loss of each baseline with FedReCall across η.

η	2	3.5	5	7	10	15	20	30
FFA-LoRA	1.282	1.248	1.213	1.189	1.160	1.119	1.090	1.081
FLoRA	1.147	1.176	1.168	1.186	1.179	1.199	1.167	1.189
FedSA-LoRA	1.285	1.282	1.278	1.269	1.259	1.248	1.244	1.237
FedIT	1.283	1.248	1.081	1.102	1.035	1.104	1.148	1.155

I Computational infrastructure

All experiments are conducted on a single NVIDIA GeForce RTX 5090 GPU. The codebase is built on PyTorch 2.7.0 (CUDA 12.8).

Client-side compute and storage. FedReCall does not change the communication payload or the trainable-parameter set, but it does add client-side computation and private state. The extra computation comes from QR-compressed SVD cache construction and a one-batch quality gate. The gate reuses a local mini-batch already available on the client; it does not require uploading data or

storing a separate gate dataset. The private cache stores selected triplets (δ_i, u_i, v_i) and remains local to the client. Table 11 reports the average per-round wall-clock time measured over the stable training phase (excluding initial warm-up rounds).

Table 11: Client-side wall-clock overhead of FedReCall. The increment is measured relative to each corresponding base method, not relative to FedIT.

Base method	Base (min)	+ FedReCall (min)	Increment
FedIT	3.4	4.7	+1.3 min / +38%
FFA-LoRA	4.1	4.9	+0.8 min / +20%
FLoRA	5.5	6.5	+1.0 min / +18%
FedSA-LoRA	4.4	4.7	+0.3 min / +7%
Dual-LoRA personalized baselines, no FedReCall			
FedDPA-T (1×)		5.1	–
FedDPA-T (2×)		6.3	–
FDLoRA		4.8	–

J Does the benefit depend on direction identity?

Does direction identity matter?

Figure 4 reports paired interventions on frozen models evaluated on held-out data, without gate selection or retraining. Each point represents one client. Panels (a) and (c) compare restoration gains and removal harms for selected and matched random directions, respectively. Panel (b) compares restoration gains for owners and other clients, while panel (d) compares loss under the original and sign-reversed corrections.

Does this functional value translate into practical gains?

These comparisons test direction choice, client specificity, and sign orientation. We next examine whether the complete directional correction also provides practical benefits when used through the deployed bypass. Table 12 reports paired personalized-loss results on Dolly-15k-derived data with the 1.3B model.

Table 12: Paired training gains on Dolly-15k-derived data (1.3B). Values are averaged over the last three rounds, with matched clients within each baseline/FedReCall pair.

Base method	Baseline loss	+ FedReCall loss	Loss reduction
FedIT	1.878	**1.652**	**12.04%**
FFA-LoRA	1.993	**1.622**	**18.62%**

The earlier eight-round evaluation in Table 13 complements these training results with bypass controls. At the same model endpoint, enabling the bypass reduces held-out loss in all 30 evaluated events across nine clients, with a mean reduction of 0.0618 (3.08%). The client-level and final-three-round comparisons also show improvements in loss and ROUGE-L. The same-endpoint comparison isolates the immediate contribution of the bypass, while the remaining comparisons summarize performance under the matched training runs.

Table 13: Practical benefits of the complete bypass on Dolly-15k-derived data. Positive values indicate improvements; a dash denotes an unreported metric.

Comparison	Loss reduction ↑	ROUGE-L gain ↑	Consistency
Same-endpoint bypass off/on	**0.0618** (3.08%)	–	30/30 events; 9/9 clients
Post-cache client mean	**0.0406**	**+0.994**	9/10 loss; 10/10 ROUGE-L
Last-three-round mean	**0.0667** (3.66%)	**+2.23**	Schedule matched

Together, the controlled interventions support owner-specific functional information in the tested directions, while the deployment results show that the complete correction can yield practical benefits. These complementary findings motivate selectively retaining diluted client information as a private bypass.

K Factor-wise discrepancy and directional attenuation

For a fixed layer and round, suppress the layer and round indices and assume a common LoRA scaling s. Let $p_j \geq 0$, $\sum_j p_j = 1$, $\bar{A} = \sum_j p_j A_j$, and $\bar{B} = \sum_j p_j B_j$. The exact and factor-wise aggregate updates are

$$X_{\text{exact}} = s \sum_j p_j B_j A_j, \qquad X_{\text{fw}} = s \bar{B} \bar{A}. \tag{17}$$

Their difference satisfies

$$R := X_{\text{exact}} - X_{\text{fw}} = s \sum_j p_j (B_j - \bar{B})(A_j - \bar{A}). \tag{18}$$

Expanding the centered product gives $\sum_j p_j B_j A_j - \bar{B}\bar{A}$, establishing the identity.

This residual describes the difference between two aggregation rules, whereas δ_i in Eq. 6 measures a local direction's retention deficit relative to the actual aggregate. The projection $\langle R, \phi_i \rangle_F$ can have either sign, so a nonzero residual does not imply attenuation of every direction.

If either factor is identical across clients, $R = 0$. Nevertheless, exact averaging can still attenuate or cancel client-specific directions, as illustrated by Eq. 9. Algebraic exactness and client-specific directional retention are therefore distinct properties. FedReCall constructs its memory from the observed directional deficit rather than from R itself.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions of FedReCall, including the identification of directional dilution in federated LoRA, the client-private dilution-gap compensation mechanism, and the empirical improvements across FedLoRA aggregators. The claims are limited to the federated LoRA fine-tuning setting and are supported by the method analysis and experiments reported in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the main limitations in the conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Proposition 1 states explicit assumptions and is proved using Weyl's inequality. The appendices derive the aggregation residual and bypass norm calibration.

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides the key information needed to reproduce the main experiments, including the SuperNI-derived task-heterogeneous federation, the 50-client subset, the base model, LoRA rank, client participation rate, training seeds, client-selection seed, evaluation metrics, and FedReCall hyperparameters. Algorithm 1 specifies the full client-side procedure, while the appendix provides details on client selection, bypass-strength sensitivity, and computational infrastructure.

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The submission includes an anonymized code package with the implementation and reproduction scripts. The experiments use publicly available datasets and pretrained models, and the paper provides the necessary experimental details, including the dataset construction, model configuration, federated protocol, hyperparameters, seeds, and evaluation metrics. The code package and the paper together provide sufficient information to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification:The paper specifies the main training and evaluation settings, including the task-heterogeneous SuperNI client construction, the number of clients, the client participation rate, the base model, LoRA rank, personalized and global evaluation metrics, training seeds, and client-selection seed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: We report sample standard deviations across three training seeds for the main results (Table 2). Figure 4 presents paired client-level outcomes for the functional controls.

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the computational infrastructure used for the experiments, including the GPU type and software stack. It also provides per-round wall-clock measurements for each base method and its FedReCall variant, and explicitly separates FedReCall's added client-side overhead from the base methods.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses existing public datasets and pretrained models, does not collect new personal data, and does not involve human-subject studies. The public release provides the method implementation rather than client-specific cache contents or newly collected sensitive data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper discusses its technical scope and privacy limitations, but does not explicitly discuss both positive and negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new pretrained language model, image generator, scraped dataset, or other high-risk asset requiring special release safeguards. The work proposes and evaluates a federated LoRA fine-tuning method using existing models and datasets, and the client-specific cache is kept local rather than released as a public model artifact.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper properly cites the existing datasets, pretrained models, and prior methods used in the experiments, including Super-Natural Instructions, LLaMA/DataJuicer-LLaMA, LoRA, and the FedLoRA/PFL baselines. The implementation will respect the licenses and terms of use of these assets, and the released code package will include the corresponding license and attribution information for all external assets used.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce or release a new dataset, benchmark, pretrained model, or other standalone research asset. The main contribution is an algorithmic method and its experimental analysis.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, user studies, human-subject experiments, or newly collected human annotations. All experiments are conducted using existing public datasets and automatic evaluation metrics.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, crowdsourcing, user studies, or the collection of new personal data. Therefore, IRB approval or an equivalent human-subjects review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Although the experiments fine-tune a large language model using LoRA, the core method development does not use an LLM as an important, original, or non-standard component of the research methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.