FEDRDAC: REPRESENTATION–DECISION ADAPTATION AND GLOBAL–LOCAL COORDINATION FOR PERSONALIZED FEDERATED LORA

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated low-rank adaptation (LoRA) enables parameter-efficient adaptation of pretrained language models without sharing raw client data. However, statistically heterogeneous client data make it difficult to preserve transferable knowledge across clients while supporting client-specific adaptation. Representative personalized federated LoRA methods mainly organize personalization along the representation pathway, leaving decision-level adaptation and the coordination of shared and local components insufficiently unified. We refer to this joint design issue as the personalization coverage–coordination gap. To address it, we propose FedRDAC, a representation–decision adaptation and global–local coordination framework for personalized federated LoRA. Its representation–decision dual adaptation mechanism extends client-specific adaptation across representation and task-decision levels, while its adaptive global–local coordination mechanism controls local contributions and supports module-specific optimization during client training. We evaluate FedRDAC on sentiment classification and semantic equivalence tasks under heterogeneous client partitions using average client accuracy, worst-client accuracy, and cross-client variability. FedRDAC achieves the best average and worst-client results among the evaluated baselines on both tasks. Heterogeneity, hyperparameter sensitivity, and ablation analyses further support the robustness and component-level contributions of the proposed design.

1 INTRODUCTION

Pretrained language models (PLMs) are widely used across natural language understanding and generation tasks, but adapting them with full-parameter fine-tuning remains expensive in computation, memory, storage, and communication. Parameter-efficient fine-tuning (PEFT) reduces this cost by updating a small set of parameters, and Low-Rank Adaptation (LoRA) (Hu et al., 2022) is widely used because it represents task-specific updates with compact low-rank factors. In privacy- or ownership-constrained settings, federated learning enables collaborative adaptation without sharing raw client data, and federated LoRA further reduces the trainable and communicated parameter footprint. However, client data are typically non-independent and identically distributed (non-IID), so efficient federated LoRA must also support client-level personalization: different clients may require different representation and decision behaviors while still benefiting from shared knowledge.

As summarized in Fig. 1(a), recent personalized federated LoRA methods explore several representative strategies for balancing shared knowledge and client-specific adaptation. Selective sharing retains part of the adaptation state locally while aggregating shared components, as in FedSA-LoRA (Guo et al., 2025). Global–local adapter designs maintain both shared and client-local adaptation branches, as in FedDPA (Yang et al., 2024). Adaptive weighting further adjusts how shared and personalized information are integrated for each client, a pattern reflected in the dynamic integration of FedDPA (Yang et al., 2024) and the adaptive mixer of FedALT (Bian et al., 2026).

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.

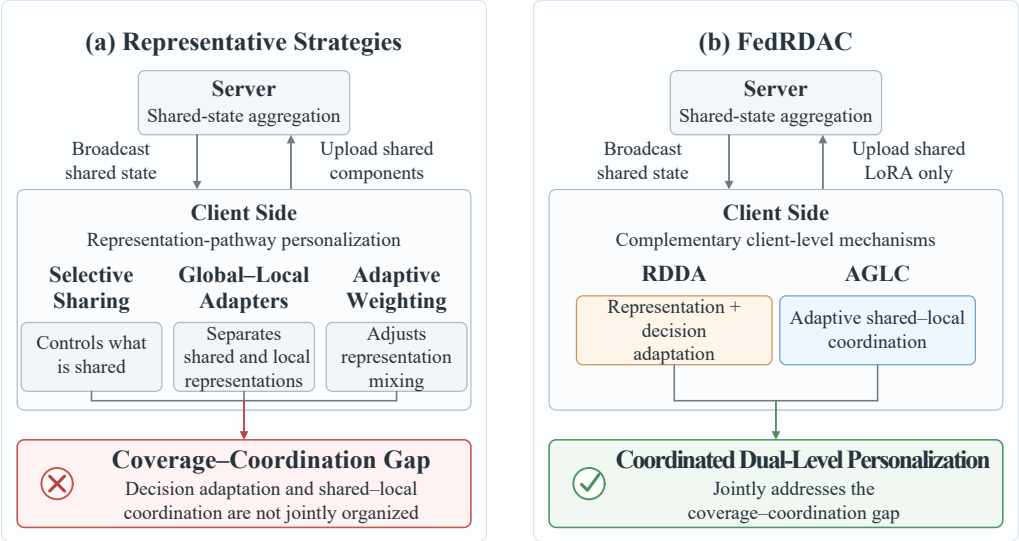

Figure 1: Motivation of FedRDAC. Representative federated LoRA strategies mainly organize personalization along the representation pathway and do not jointly organize decision-level adaptation and shared–local coordination (left). FedRDAC addresses this coverage–coordination gap through Representation–Decision Dual Adaptation (RDDA) and Adaptive Global–Local Coordination (AGLC) (right).

Together, these methods provide useful forms of representation-level personalization by determining what to share, how to combine shared and local adapters, or how to weight their contributions. However, they do not jointly organize where client-specific personalization should occur and how shared and local components should be coordinated during local training. We refer to this unresolved joint design issue as the personalization coverage–coordination gap. Specifically, client-specific adaptation should span both representation and task-decision levels, while its interaction with the shared adaptation pathway should be coordinated during local optimization. Accordingly, this work studies two coupled design issues: the placement of client-specific personalization and the coordination of shared and personalized components.

Figure 1(b) summarizes our answer to these two questions through FedRDAC, a unified personalized federated LoRA framework. Representation–Decision Dual Adaptation (RDDA) answers where personalization should occur: a client-local LoRA branch provides representation-level adaptation, while a client-local task head provides decision-level adaptation. Adaptive Global–Local Coordination (AGLC) answers how shared and personalized components should be coordinated: a learnable client-specific gate serves as the primary adaptive mechanism for controlling the local LoRA contribution, while module-specific learning rates, implemented through optimizer parameter groups, provide a supporting design for jointly optimizing the shared LoRA, local LoRA, local head, and gate parameters. The server side uses the residual-corrected FedEx-LoRA shared-state update as an adopted aggregation mechanism rather than as a new contribution of this work.

The main contributions of this paper are as follows:

- We identify a personalization coverage–coordination gap in representative personalized federated LoRA designs: personalization remains centered on representation adaptation, while decision-level adaptation and the coordination of shared and local components are not jointly organized.
- We propose FedRDAC, integrating RDDA to extend client-specific adaptation across representation and decision levels and AGLC to adaptively coordinate shared and personalized components during local training.

- We provide a controlled evaluation on SST-2 and MRPC using Client Avg, Client Worst, and Client Std, complemented by analyses of data heterogeneity, hyperparameter sensitivity, and component ablations under the reported experimental protocol.

2 Related Work

2.1 Federated LoRA Fine-Tuning

Federated learning trains models through server–client parameter exchange without centralizing raw data (McMahan et al., 2017; Kairouz et al., 2021), but non-IID client data can induce update drift; FedProx (Li et al., 2020), SCAFFOLD (Karimireddy et al., 2020), and FedOpt (Reddi et al., 2021) address this issue from regularization, variance correction, and server optimization perspectives. For pretrained language models (Devlin et al., 2019), parameter-efficient fine-tuning reduces the cost of federated adaptation. Adapter (Houlsby et al., 2019), prefix (Li & Liang, 2021), and prompt tuning (Lester et al., 2021) are common PEFT routes, while LoRA (Hu et al., 2022) is especially relevant here because it represents trainable updates with compact low-rank factors.

Federated LoRA methods reduce trainable and communicated parameters by exchanging lightweight adapter states, but the product form of LoRA makes aggregation nontrivial. FedIT/FedAvg-LoRA-style methods (Zhang et al., 2024) average LoRA factors separately, which does not generally recover the average of client-side updates. SLoRA (Babakniya et al., 2023), FFA-LoRA (Sun et al., 2024), FLoRA (Wang et al., 2024), and FedSA-LoRA (Guo et al., 2025) study initialization, factor sharing, heterogeneous ranks, and selective aggregation. FedEx-LoRA (Singhal et al., 2025) introduces a residual-corrected server update for exact shared-state aggregation; this paper adopts that update as the server-side aggregation rule and focuses on client-level personalization.

2.2 Personalized Federated Learning and Foundation-Model Adaptation

Personalized federated learning improves client-level performance under non-IID data by separating shared and client-specific components or by mixing global and local models. FedPer (Arivazhagan et al., 2019), FedRep (Collins et al., 2021), LG-FedAvg (Liang et al., 2020), and FedBABU (Oh et al., 2022) motivate representation/head separation or local task-head adaptation, while Ditto (Li et al., 2021), pFedMe (Dinh et al., 2020), APFL (Deng et al., 2020), Per-FedAvg (Fallah et al., 2020), and MOCHA (Smith et al., 2017) provide representative regularization, interpolation, meta-learning, and multi-task formulations.

Personalized federated foundation-model adaptation is closer to the setting studied here. FedDPA (Yang et al., 2024) maintains global and local adapters and dynamically integrates them for personalization, and FedALT (Bian et al., 2026) retains individual client LoRA parameters while incorporating shared knowledge through a Rest-of-World LoRA and adaptive mixer. In contrast, FedRDAC treats personalization coverage and global–local coordination as a joint design problem: it places personalization at both representation and decision levels, keeps these personalized modules client-local, and coordinates them with the shared LoRA branch through a client-specific gate and supporting module-specific learning rates implemented through optimizer parameter groups.

3 Preliminaries

3.1 Federated LoRA Setup

Consider a federated system with K clients, where client i holds local data D_i and does not transmit raw examples. In round t, the server broadcasts the shared state, clients train locally, and the server aggregates the uploaded shared LoRA state. The implementation uses full client participation and uniform averaging, so the method formulation below uses $1/K$ client weights. For a pretrained weight $W_0 \in \mathbb{R}^{m\times n}$ that is frozen during client-side gradient updates, LoRA represents the effective update as

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

where $A \in \mathbb{R}^{r\times n}$, $B \in \mathbb{R}^{m\times r}$, and $r \ll \min(m, n)$. The method below uses W_0^t for the shared backbone state and $\theta^g = \{A^g, B^g\}$ for the shared LoRA branch.

3.2 SHARED-STATE AGGREGATION BACKGROUND

For uploaded client LoRA states $\{A_i, B_i\}_{i=1}^K$, a common server update averages factors separately:

$$\bar{A} = \frac{1}{K}\sum_{i=1}^{K} A_i, \qquad \bar{B} = \frac{1}{K}\sum_{i=1}^{K} B_i. \tag{2}$$

Because the effective update is B_iA_i, factor-wise averaging generally does not recover the average client-side update:

$$\bar{B}\bar{A} \neq \frac{1}{K}\sum_{i=1}^{K} B_iA_i. \tag{3}$$

FedEx-LoRA (Singhal et al., 2025) provides a residual-corrected update for this shared state. With

$$\Delta W_{\text{res}} = \frac{1}{K}\sum_{i=1}^{K} B_iA_i - \bar{B}\bar{A}. \tag{4}$$

the server applies

$$W_0^{t+1} = W_0^t + \Delta W_{\text{res}}, \qquad \theta^{g,t+1} = \{\bar{A}, \bar{B}\}. \tag{5}$$

This paper uses the residual-corrected FedEx-LoRA update as the adopted server-side shared-state update. The contribution is the client-local personalization pathway and its coordination with the shared LoRA branch, not the aggregation correction itself.

4 METHOD

4.1 OVERVIEW

FedRDAC separates the federated shared state from persistent client-local personalization state. At communication round t, the server shared state is

$$\mathcal{G}^t = \{W_0^t, \theta^{g,t}\}, \qquad \theta^{g,t} = \{A^{g,t}, B^{g,t}\}, \tag{6}$$

where W_0^t denotes the shared backbone state and $\theta^{g,t}$ denotes the shared LoRA branch. Client i keeps the personalized state

$$\phi_i^t = \{\theta_i^{l,t}, \theta_i^{h,t}, s_i^t\}, \qquad \theta_i^{l,t} = \{A_i^{l,t}, B_i^{l,t}\}. \tag{7}$$

The shared LoRA branch is initialized from the server state in each round and participates in upload and aggregation, whereas ϕ_i^t is retained on client i across communication rounds.

Figure 2 illustrates the overall server–client workflow of FedRDAC and the roles of RDDA and AGLC during local training.

The method contains two coupled mechanisms. Representation–Decision Dual Adaptation (RDDA) determines where personalization occurs: a client-local LoRA branch adapts internal representations, and a client-local task head adapts the task decision boundary. Adaptive Global–Local Coordination (AGLC) determines how shared and personalized components are coordinated: a client-specific gate scales the local LoRA contribution, and module-specific learning rates coordinate the optimization of the shared LoRA branch, client-local LoRA branch, client-local task head, and gate under one local objective.

4.2 PERSONALIZED FEDERATED OBJECTIVE

For a trainable layer on client i, let $\theta_i^g = \{A_i^g, B_i^g\}$ denote the local copy of the shared LoRA branch and $\theta_i^l = \{A_i^l, B_i^l\}$ denote the client-local LoRA branch. The effective layer weight used by client i is

$$W_i^{\text{eff}} = W_0^t + \Delta W(\theta_i^g) + \alpha_i \Delta W(\theta_i^l), \qquad \Delta W(\theta) = BA, \tag{8}$$

where

$$\alpha_i = \alpha_{\max}\sigma(s_i). \tag{9}$$

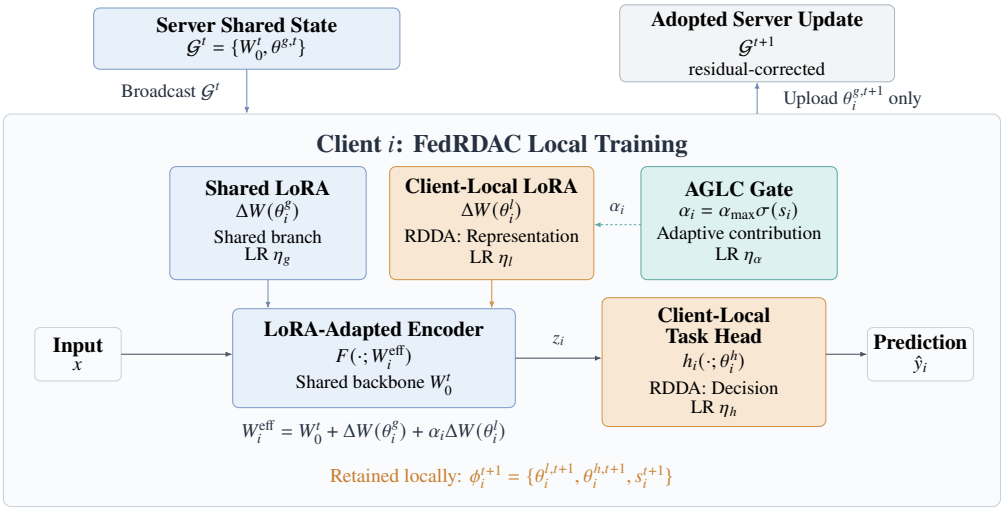

Figure 2: Overview of FedRDAC. The server broadcasts the shared state, while each client performs local training with shared and persistent client-local components. RDDA provides representation- and decision-level adaptation through the client-local LoRA and task head, respectively, while AGLC coordinates shared and local adaptation through a client-specific gate and module-specific learning rates. Only the updated shared LoRA is uploaded; personalized components remain local, and the server uses the adopted residual-corrected shared-state update.

The shared branch $\Delta W(\theta_i^g)$ is always present. The gate α_i controls the additional contribution of the local LoRA branch relative to this shared branch; it is not a convex mixing coefficient between two mutually exclusive branches.

Let $F(\cdot; W_i^{\text{eff}})$ denote the encoder with the effective LoRA-adapted weights. For input x, client i computes

$$z_i(x) = F(x; W_i^{\text{eff}}), \qquad \hat{y}_i(x) = h_i(z_i(x); \theta_i^h). \tag{10}$$

The empirical task loss on client i is

$$\mathcal{L}_i = \frac{1}{|D_i|} \sum_{(x,y)\in D_i} \ell(\hat{y}_i(x), y)\,, \qquad \mathcal{L}_i = \mathcal{L}_i(\theta_i^g, \theta_i^l, \theta_i^h, s_i; W_0^t). \tag{11}$$

The reported experiments use the standard task loss without additional regularization. In the main formulation, W_0^t is treated as part of the broadcast shared state rather than a local gradient-updated variable.

With uniform client averaging, the personalized federated objective can be summarized as

$$\min_{\theta^g, \{\phi_i\}_{i=1}^K} \mathcal{J} = \frac{1}{K} \sum_{i=1}^{K} \mathcal{L}_i(\theta^g, \phi_i; W_0^t), \tag{12}$$

where θ^g is the shared LoRA state coordinated by the server and $\phi_i = \{\theta_i^l, \theta_i^h, s_i\}$ is the persistent personalized state of client i.

4.3 Representation–Decision Dual Adaptation

RDDA is motivated by the fact that client heterogeneity can affect both internal representations and task decision boundaries. A single shared LoRA branch can encode cross-client transferable adaptation, but client-specific input and label distributions may still require local flexibility inside the representation pathway. For this reason, client i maintains a local LoRA branch $\theta_i^l = \{A_i^l, B_i^l\}$, whose update follows the LoRA form in Eq. (1) and enters the effective weight in Eq. (8). This branch is trained on client i, retained locally across communication rounds, and not aggregated

into the server shared state. It provides client-specific representation adaptation while leaving the federated shared LoRA branch available for cross-client knowledge transfer.

Representation-level adaptation alone does not fully address heterogeneity in label distributions or client-specific task decision boundaries. RDDA therefore also keeps the task head $h_i(\cdot;\theta_i^h)$ in Eq. (10) local to each client. The head parameters θ_i^h are trained and stored locally and are not uploaded to the server. Thus, the client-local LoRA branch adapts client-specific representations inside the PLM, while the client-local task head adapts the task decision boundary.

4.4 Adaptive Global–Local Coordination

AGLC coordinates the shared and personalized components during local training. Its main adaptive mechanism is the client-specific gate in Eq. (9), where s_i is retained on client i and $\alpha_{\max}$ upper-bounds the local LoRA contribution. In implementation, the gate scales the forward contribution of the local LoRA adapter, so Eq. (8) should be read as an effective-update view of the implemented adapter composition. The initialization of s_i can be chosen so that α_i starts from a preset value α_0, but this is an implementation detail rather than a separate methodological component.

As supporting coordination, AGLC uses module-specific learning rates, implemented through optimizer parameter groups, because the shared LoRA branch, client-local LoRA branch, client-local task head, and gate have different roles and optimization sensitivities. Let

$$\Theta_i = \{\theta_i^g, \theta_i^l, \theta_i^h, s_i\}, \qquad \eta = \{\eta_g, \eta_l, \eta_h, \eta_\alpha\}. \tag{13}$$

The local update is written abstractly as

$$\Theta_i \leftarrow \mathrm{Opt}_\eta\left(\Theta_i, \nabla_{\Theta_i}\mathcal{L}_i\right), \tag{14}$$

where Opt_η denotes an optimizer with module-specific learning rates. The implementation uses AdamW optimizer parameter groups for module-specific coordinated joint optimization of the shared LoRA branch, client-local LoRA branch, client-local task head, and gate.

4.5 Federated Training Procedure

Algorithm 1 summarizes FedRDAC training with RDDA and AGLC. The local personalized state ϕ_i^t persists across communication rounds. In each round, clients receive the shared state, train both shared and personalized components locally, retain personalized parameters, and upload only the shared LoRA branch for server aggregation. The aggregation rule $\mathcal{A}$ updates the shared state; in the implementation, $\mathcal{A}$ is the residual-corrected FedEx-LoRA server update described in Section 3.2.

Algorithm 1 FedRDAC Training

Input: clients $\{1,\dots,K\}$, rounds T, local epochs E, initial shared state $\mathcal{G}^0 = \{W_0^0, \theta^{g,0}\}$
Output: final shared state $\mathcal{G}^T$ and personalized states $\{\phi_i^T\}_{i=1}^K$

```
Server initializes G^0
Each client i initializes φ_i^0 = {θ_i^{l,0}, θ_i^{h,0}, s_i^0}
for t = 0, ..., T − 1 do
    Server broadcasts G^t = {W_0^t, θ^{g,t}} to participating clients S_t
    for each client i ∈ S_t in parallel do
        Load θ_i^{g,t} ← θ^{g,t} and restore persistent φ_i^t
        Compute α_i^t = α_max σ(s_i^t)
        Train Θ_i = {θ_i^g, θ_i^l, θ_i^h, s_i} on D_i for E epochs using Opt_η
        Retain φ_i^{t+1} = {θ_i^{l,t+1}, θ_i^{h,t+1}, s_i^{t+1}} locally
        Upload only θ_i^{g,t+1} to the server
    end for
    G^{t+1} ← A(G^t, {θ_i^{g,t+1}}_{i∈S_t})
end for
return G^T and {φ_i^T}_{i=1}^K
```

5 EXPERIMENTAL SETUP

5.1 DATASETS AND NON-IID PARTITIONING

This study evaluates FedRDAC on the Stanford Sentiment Treebank (SST-2) and Microsoft Research Paraphrase Corpus (MRPC) from the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019), covering sentence-level sentiment classification and sentence-pair semantic equivalence. Accuracy is used as the client-level metric m_i for both tasks to maintain a consistent evaluation protocol.

Non-IID client data are constructed by Dirichlet label partitioning, where smaller α indicates stronger label-distribution heterogeneity. The main experiments use 8 clients and $\alpha = 1.0$; the heterogeneity study further evaluates $\alpha \in \{2.0, 1.0, 0.8, 0.5\}$ to examine robustness across different levels of client data heterogeneity.

5.2 BASELINES

FedRDAC is compared with representative federated LoRA methods under the same backbone, client partitioning, and training protocol. FedAvg-LoRA (FedIT-style) (Zhang et al., 2024) applies FedAvg separately to LoRA factors. FedEx-LoRA (Singhal et al., 2025) is the exact-aggregation baseline with residual correction. FedSA-inspired (Guo et al., 2025) trains both LoRA factors locally while uploading and aggregating A only. FedDPA-inspired (Yang et al., 2024) uses global and local adapters fused by a client-level scalar weight β. FedALT-inspired (Bian et al., 2026) maintains a client-specific individual LoRA together with a Rest-of-World LoRA constructed from other clients' individual LoRA states, and adaptively combines the two branches through a learned router. We adapt this mechanism to the unified RoBERTa classification protocol used in our experiments.

5.3 IMPLEMENTATION DETAILS

The setup uses RoBERTa-base (Liu et al., 2019) with LoRA modules inserted into attention layers. All compared methods use the same backbone, realized client partition, communication schedule, and evaluation rule. Unless otherwise specified, experiments use eight clients and Dirichlet label partitioning with default $\alpha = 1.0$. Models are implemented with PyTorch and Hugging Face Transformers and trained on a single NVIDIA GeForce RTX 4090 GPU.

Each run uses 50 communication rounds, 1 local epoch, batch size 16, and maximum sequence length 64. The LoRA rank is $r = 4$ with scaling factor 8 and a warmup ratio of 0.06; for FedRDAC, the shared and client-local LoRA branches use dropout rates of 0.1 and 0.0, respectively. The global/local LoRA learning rates are 1×10^{-3}, and the local head/gate learning rates are 3×10^{-4}. The fixed-fusion baseline uses $\lambda = 0.25$; the adaptive gate uses $\alpha_0 = 0.25$ and $\alpha_{\max} = 1.0$. SST-2 and MRPC use the same default hyperparameters, and no large-scale dataset-specific grid search is performed.

5.4 EVALUATION METRICS

The evaluation focuses on client-level personalization performance. Let m_i denote the Accuracy on client i. The aggregate metrics are

$$\text{Client Avg} = \frac{1}{K}\sum_{i=1}^{K} m_i, \tag{15}$$

$$\text{Client Worst} = \min_{i\in\{1,\dots,K\}} m_i, \tag{16}$$

$$\text{Client Std} = \sqrt{\frac{1}{K}\sum_{i=1}^{K}(m_i - \bar{m})^2}, \qquad \bar{m} = \text{Client Avg}. \tag{17}$$

Client Avg measures overall personalization performance, Client Worst measures worst-client performance, and Client Std measures cross-client variability. After each communication round, client-

Table 1: Main results on SST-2 and MRPC

Method	SST-2			MRPC		
	Client Avg ↑	Client Worst ↑	Client Std ↓	Client Avg ↑	Client Worst ↑	Client Std ↓
FedAvg-LoRA / FedIT-style	94.80	90.72	2.54	84.32	71.05	8.55
FedEx-LoRA	94.50	91.05	2.22	85.18	71.05	8.47
FedDPA-inspired	95.90	89.69	3.35	83.79	72.00	7.63
FedSA-inspired	95.83	91.01	2.57	84.06	68.75	11.11
FedALT-inspired	95.47	90.59	3.08	80.30	63.41	11.80
FedRDAC	**96.70**	**93.81**	**1.97**	**87.30**	**75.00**	**6.82**

Note: All metrics are percentages; ↑ and ↓ indicate that higher and lower values are better, respectively. FedSA-inspired, FedDPA-inspired, and FedALT-inspired follow the unified implementations described in Section 5.2.

level performance is evaluated on the held-out GLUE validation split; the round with the highest Client Avg is selected, and all three metrics are reported from that same round.

6 Client Accuracy Comparison

Table 1 compares FedRDAC with representative federated LoRA baselines under the unified non-IID protocol using Client Avg, Client Worst, and Client Std.

Table 1 shows that FedRDAC achieves the highest Client Avg and Client Worst and the lowest Client Std on both tasks. Overall, FedRDAC improves average and worst-client personalization performance while reducing cross-client variability across the evaluated tasks.

7 Robustness to Data Heterogeneity

To assess robustness to client heterogeneity, we evaluate SST-2 under Dirichlet $\alpha \in \{2.0, 1.0, 0.8, 0.5\}$ and compare FedRDAC with FedSA-inspired and FedDPA-inspired using the same client-level metrics. Smaller α indicates stronger label-distribution heterogeneity.

Table 2: Robustness results on SST-2 under different Dirichlet α values

Dirichlet α	Method	Client Avg ↑	Client Worst ↑	Client Std ↓
2.0	FedSA-inspired	95.04	89.15	3.32
	FedDPA-inspired	95.29	89.15	3.49
	FedRDAC	**95.80**	**89.92**	**2.97**
1.0	FedSA-inspired	95.83	91.01	2.57
	FedDPA-inspired	95.90	89.69	3.35
	FedRDAC	**96.70**	**93.81**	**1.97**
0.8	FedSA-inspired	92.93	82.69	4.73
	FedDPA-inspired	93.39	86.54	3.83
	FedRDAC	**94.27**	**88.46**	**2.80**
0.5	FedSA-inspired	94.51	89.52	3.05
	FedDPA-inspired	94.97	91.13	2.56
	FedRDAC	**95.42**	**91.95**	**1.65**

Note: Smaller α indicates stronger client data heterogeneity; all metrics are percentages.

Because absolute performance also depends on the specific client partition and local sample composition, the metrics need not vary monotonically with α. Across all evaluated Dirichlet settings, FedRDAC achieves the highest Client Avg and Client Worst and the lowest Client Std among the compared methods. This consistent advantage across both milder and stronger heterogeneity suggests that the observed benefit is not tied to a single client-partition setting.

8 HYPERPARAMETER SENSITIVITY

To assess the sensitivity of AGLC to the maximum local-contribution scale $\alpha_{\max}$ in Eq. (9), we vary $\alpha_{\max} \in \{0.5, 1.0, 1.5\}$ on SST-2 under the default Dirichlet setting $\alpha = 1.0$, while keeping $\alpha_0 = 0.25$ and all other training and evaluation settings unchanged.

Table 3: Sensitivity of FedRDAC to the maximum local-contribution scale $\alpha_{\max}$ on SST-2

$\alpha_{\max}$	Client Avg ↑	Client Worst ↑	Client Std ↓
0.5	96.17	92.78	**1.93**
1.0	**96.70**	**93.81**	1.97
1.5	96.59	92.78	1.97

Across the evaluated settings, Client Avg ranges from 96.17 to 96.70, Client Worst from 92.78 to 93.81, and Client Std from 1.93 to 1.97. The default $\alpha_{\max} = 1.0$ yields the highest Client Avg and Client Worst, whereas $\alpha_{\max} = 0.5$ gives the lowest Client Std. The small variations across these settings indicate limited sensitivity to the local-contribution upper bound within the evaluated range.

9 ABLATION STUDY

On SST-2, the ablation evaluates RDDA's client-local task head and LoRA for decision and representation adaptation, respectively, and AGLC's client-specific gate and module-specific learning rates. AGLC components are evaluated on RDDA because they coordinate shared and personalized components; fixed fusion uses $\lambda = 0.25$, matching the gate's initial local contribution $\alpha_0 = 0.25$.

Table 4: Ablation of RDDA and AGLC components on SST-2

Method	LH	LL	Gate	LR	Avg ↑	Worst ↑	Std ↓
FedEx-LoRA	×	×	×	×	94.50	91.05	2.22
Decision Adapt. (LH)	✓	×	×	×	95.79	92.10	2.06
Representation Adapt. (LL)	×	✓	×	×	95.15	91.10	2.33
RDDA + Fixed Fusion	✓	✓	×	×	95.44	89.61	2.77
RDDA + Adaptive Gate	✓	✓	✓	×	95.69	91.76	2.14
RDDA + Module-Specific LR	✓	✓	×	✓	95.67	91.74	2.32
FedRDAC	✓	✓	✓	✓	**96.70**	**93.81**	**1.97**

Note: Metrics are percentages. LH and LL denote the client-local task head and client-local LoRA; Gate denotes the client-specific adaptive gate; and LR denotes module-specific learning rates. When LR is disabled, all active trainable components use the common learning rate of 1×10^{-3}. ✓ indicates that the component is used, and × indicates that it is not used.

Table 4 shows that decision and representation adaptation each improve Client Avg over FedEx-LoRA, but fixed-fusion RDDA does not reliably improve worst-client performance or cross-client variability. Either AGLC component improves this baseline, and complete FedRDAC achieves the best overall result. These results suggest that extending personalization coverage alone is insufficient to consistently improve all client-level metrics; coordination of shared and local components is also important.

10 CONCLUSION

FedRDAC addresses the personalization coverage–coordination gap by combining representation–decision adaptation with adaptive global–local coordination. Across SST-2 and MRPC, it achieves the best average and worst-client performance and the lowest cross-client variability among the compared baselines. It maintains favorable performance across evaluated heterogeneity levels, shows limited sensitivity to $\alpha_{\max}$, and ablations support both mechanisms. Future work will extend the evaluation to broader tasks and larger backbone models.

Reproducibility Statement

Sections 4 and 5 specify the model formulation, federated training procedure, datasets, client partitioning, optimization settings, and evaluation rule used in the experiments. Algorithm 1 summarizes the complete training process, while Appendix A documents the parameter and communication accounting. All experiments use publicly available datasets and pretrained models.

AI use statement

In this work, generative AI tools were used to assist with language and presentation refinement, literature retrieval and discovery, software implementation, and research workflow support, including feedback on experimental design and interpretation of experimental results. All AI-assisted outputs were reviewed and verified by the authors, and all reported experimental results were obtained from executed experiments rather than generated or altered by AI. The authors take responsibility for the final content of this work.

References

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019. URL https://arxiv.org/abs/1912.00818.

Sara Babakniya, Ahmed Roushdy Elkordy, Yahya H. Ezzeldin, Qingfeng Liu, Kee-Bong Song, Mostafa El-Khamy, and Salman Avestimehr. SLoRA: Federated parameter efficient fine-tuning of language models. *arXiv preprint arXiv:2308.06522*, 2023. URL https://arxiv.org/abs/2308.06522.

Jieming Bian, Lei Wang, Letian Zhang, and Jie Xu. FedALT: Federated fine-tuning through adaptive local training with Rest-of-World LoRA. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19728–19736, 2026. doi: 10.1609/aaai.v40i24.39054.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 2089–2099, 2021. URL https://proceedings.mlr.press/v139/collins21a.html.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020. URL https://arxiv.org/abs/2003.13461.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1423.

Canh T. Dinh, Nguyen H. Tran, and Tuan Dung Nguyen. Personalized federated learning with Moreau envelopes. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f4f1f13c8289ac1b1ee0ff176b56fc60-Abstract.html.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/24389bfe4fe2eba8bf9aa9203a44cdad-Abstract.html.

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

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2790–2799, 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

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.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021. doi: 10.1561/2200000083.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 5132–5143, 2020. URL https://proceedings.mlr.press/v119/karimireddy20a.html.

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*, pp. 3045–3059. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.243.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, pp. 429–450, 2020. URL https://proceedings.mlsys.org/paper_files/paper/2020/hash/1f5fe83998a09396ebe6477d9475ba0c-Abstract.html.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 6357–6368, 2021. URL https://proceedings.mlr.press/v139/li21h.html.

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)*, pp. 4582–4597. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.353.

Paul Pu Liang, Terrance Liu, Liu Ziyin, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020. URL https://arxiv.org/abs/2001.01523.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. URL https://arxiv.org/abs/1907.11692.

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

Jaehoon Oh, Sangmook Kim, and Se-Young Yun. FedBABU: Toward enhanced representation for federated image classification. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=HuaYQfggn5u.

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

Raghav Singhal, Kaustubh Ponkshe, and Praneeth Vepakomma. FedEx-LoRA: Exact aggregation for federated and efficient fine-tuning of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1316–1336, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.67.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/6211080fa89981f66b1a0c9d55c61d0f-Abstract.html.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving LoRA in privacy-preserving federated learning. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2403.12313.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. FLoRA: Federated fine-tuning large language models with heterogeneous low-rank adaptations. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0708.

Yiyuan Yang, Guodong Long, Tao Shen, Jing Jiang, and Michael Blumenstein. Dual-personalizing adapter for federated foundation models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2403.19211.

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

A Parameter and Communication Analysis

Appendix A reports trainable parameters, uplink payloads, local state, and server aggregation scope.

Trainable / Client is locally optimized parameters. Total Uploaded / Round is eight-client uplink excluding broadcast. Local-only / Client is retained, non-aggregated parameters; Uploaded / Aggregated Components lists components processed by the server.

Table 5: Parameter count and upload communication analysis

Method	Trainable/Client (M)	Total Uploaded/Round (M)	Local-only/Client (M)	Uploaded/Aggregated Components
FedEx-LoRA	0.740	5.917	0	Global LoRA + classifier head
FedDPA-inspired	1.479	15.391	0.147	Global LoRA + classifier payload
FedRDAC	0.887	1.180	0.740	Global LoRA only

Note: Counts (millions) use RoBERTa-base with LoRA rank $r = 4$ and eight clients. Total Uploaded/Round is eight-client uplink excluding broadcast; counts are parameters, not bytes. FedDPA-inspired follows Section 5.2; FedEx residual correction is server-side and adds no client upload.

FedRDAC retains client-local LoRA, gate, and classifier head and uploads only Global LoRA; it has 0.887M vs 0.740M trainable parameters and 1.180M vs 5.917M eight-client uplink per round.

B CONVERGENCE ANALYSIS

Figure 3 plots unsmoothed SST-2 Client Avg records from the complete 50-round logs; rounds 1–10 are shown because most changes occur early.

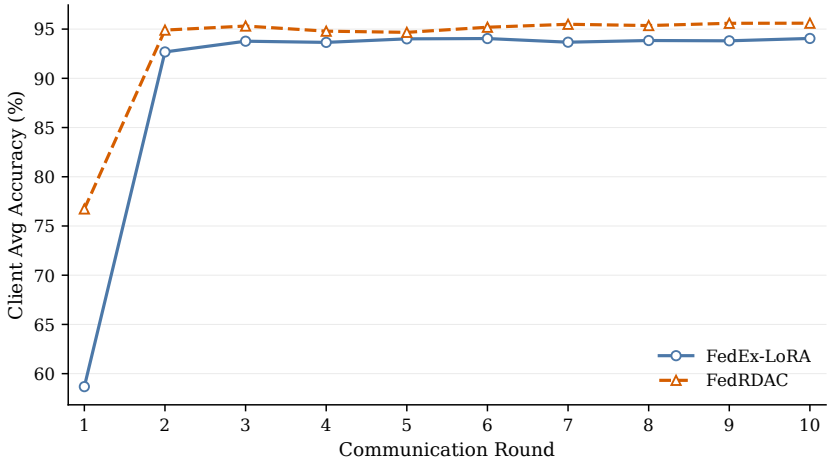

Figure 3: SST-2 Client Avg over rounds 1–10 (unsmoothed records).

FedRDAC remains above FedEx-LoRA throughout the displayed early rounds, suggesting no obvious early-round convergence penalty from the added personalization mechanisms.