Averaging is Not Enough: Preserving Client-Specific Knowledge in Federated PEFT with One-Round Aggregation

Anonymous ACL submission

Abstract

Federated Learning (FL) provides a privacy-preserving framework for fine-tuning Pre-trained Language Models (PLMs) on decentralized data. To reduce the computational and communication costs arising from the massive parameters of PLMs, parameter-efficient fine-tuning (PEFT) techniques have been widely adopted. However, integrating PEFT into FL remains challenging, especially under non-IID settings, where significant performance degradation is commonly observed. In this work, we identify the root cause of this degradation as a fundamental incompatibility between PEFT methods and the aggregation mechanism in FL. Specifically, conventional averaging fails to effectively preserve the personalized knowledge encoded in each client's PEFT updates, resulting in suboptimal performance and slower convergence. To address this issue, we propose an expert-guided aggregation strategy designed to better retain client-specific information. We instantiate this strategy with FedELoRA, a novel LoRA-based framework for FL that requires only a single round of communication. FedELoRA treats each client's locally trained LoRA adapter as an expert and employs a trainable gating network to dynamically combine them after local training. This enables effective integration of heterogeneous client knowledge while significantly reducing communication overhead. Extensive experiments across diverse domains demonstrate that FedELoRA consistently outperforms state-of-the-art baselines under both IID and non-IID settings, while using only 15.4% of the communication cost of the most efficient prior method. Our code is available at https://anonymous.4open.science/r/FedELoRA-30C0.

1 Introduction

Pre-trained Large Language Models (PLMs), such as GPT-4 (Achiam et al., 2023), have become the cornerstone of natural language processing. Adapting these PLMs to specialized downstream tasks typically requires fine-tuning on domain-specific datasets (Hadi et al., 2023; Zhao et al., 2023), which are often siloed across organizations and cannot be shared directly (Voigt and Von dem Bussche, 2017). Federated Learning (FL) offers a promising solution by enabling collaborative model training without disclosing raw data (McMahan et al., 2017; Li et al., 2020). Despite its potential, directly fine-tuning PLMs in FL leads to prohibitive communication and computation overhead. To alleviate this issue, parameter-efficient fine-tuning (PEFT) methods have been widely adopted, which introduce lightweight adapters to reduce training costs in computation and communication (Ye et al., 2024; Kuang et al., 2024).

However, the application of PEFT in FL still suffers from notable challenges. Even with advanced techniques such as LoRA (Hu et al., 2022) and 4-bit quantization, fine-tuning a 731B-parameter DeepSeek-R1 model across only four clients incurs a communication cost of up to 1160 GB (DeepSeek-AI, 2025). More critically, the performance of PEFT methods degrades substantially in non-IID data distributions (Che et al., 2023; Babakniya et al., 2023; Cho et al., 2024; Zhang et al., 2023), which are prevalent in FL. Although increasing adapter capacity can partially address this issue, it inevitably undermines the core efficiency benefits of PEFT, revealing a fundamental trade-off between model performance and communication efficiency in federated scenarios.

Recent studies have proposed various enhancements to PEFT methods in FL, with a particular focus on LoRA-based approaches. These efforts generally fall into two categories. The first line of work addresses the suboptimal aggregation of LoRA adapters, which can lead to degraded performance. For instance, FFA-LoRA (Sun et al., 2024) updates only the matrix $\mathbf{B}$ during training

to ensure the correctness, while RoLoRA (Chen et al., 2024) employs an alternating aggregation strategy to improve robustness. The second line of work mitigates the effects of data heterogeneity by designing client-specific adapters that better align with local data distributions (Kim et al., 2023; Babakniya et al., 2023; Guo et al., 2025; Cho et al., 2024). While recent efforts have improved model performance, they often assume that multi-round averaging is an effective aggregation strategy, neglecting its incompatibility with the goals of PEFT.

In this work, we point out a fundamental mismatch between PEFT and the aggregation mechanism in FL. While PEFT methods encode personalized knowledge in a small number of trainable parameters, standard FL aggregation averages them across clients, thereby erasing individual adaptations. To empirically validate this mismatch, we conduct experiments on LoRA (Hu et al., 2022), a widely-adopted PEFT method. Our analysis focuses on the performance variation of its low-rank matrices under averaging. We find that averaging the $\mathbf{A}$ matrix across clients has minimal impact on local performance, suggesting that shared knowledge is preserved. In contrast, averaging the $\mathbf{B}$ matrix significantly degrades local performance, indicating a loss of client-specific information. These findings empirically support our hypothesis and also align with prior observations that $\mathbf{A}$ captures generalizable features, while $\mathbf{B}$ encodes personalized ones (Tian et al., 2024; Guo et al., 2025). Furthermore, we observe that the degradation of personalized knowledge necessitates more communication rounds to converge, ultimately resulting in higher communication overhead.

To this end, we propose an expert-guided aggregation strategy that models each client as a domain expert and leverages a Mixture of Experts (MoE) architecture (Shazeer et al., 2017) to effectively preserve client-specific knowledge during aggregation. We instantiate this strategy with Federated Expert-Gated LoRA (FedELoRA), a novel FL framework that enables collaborative PLM fine-tuning in a single communication round. FedELoRA redefines FL aggregation by treating each client's locally trained $\mathbf{B}$ matrix as an independent expert and integrating them via a lightweight gating network that dynamically assigns expert weights based on input relevance. This forms the Expert-Gated LoRA (EGL) network, which enables input-adaptive expert selection and enhances generalization across diverse domains. Meanwhile, based on empirical evidence that $\mathbf{A}$ matrices capture generalizable features, FedELoRA averages them across clients to reduce communication overhead. This design enhances model performance while reducing communication overhead, offering a principled alternative to conventional averaging in FL.

Our contributions are summarized as follows:

- We identify the root cause of the performance degradation when combining FL with PEFT methods: a fundamental mismatch between PEFT's limited parameter updates and FL's averaging aggregation mechanism.
- We propose an expert-guided aggregation mechanism that enables more efficient aggregation of client-specific knowledge with only one round of communication. We instantiate this strategy with FedELoRA, a novel LoRA-based framework for FL.
- We conduct comprehensive experiments under both IID and Non-IID settings, where FedELoRA consistently achieves the best average rank across diverse tasks and heterogeneity levels, demonstrating superior generalization. In addition, FedELoRA reduces communication cost by up to 84.6% compared to the most efficient baseline.

2 Background and Motivation

2.1 Federated Fine-tuning of PLMs

Federated Learning (FL) enables multiple clients to collaboratively fine-tune a PLM without exposing their private data (Hadi et al., 2023; Zhao et al., 2023; Zhang et al., 2023). This paradigm is particularly valuable in sensitive domains such as healthcare and finance, where data are often siloed across institutions (Ye et al., 2024; Kuang et al., 2024). However, directly fine-tuning PLMs in the FL setting imposes prohibitive computational and communication costs owing to the models' massive parameter sizes. To mitigate this, parameter efficient fine-tuning (PEFT) methods have been applied to FL, such as adapter-tuning (Ghiasvand et al., 2024), prompt-tuning (Cui et al., 2024), and LoRA (Hu et al., 2022), which significantly reduce the number of trainable parameters.

Among these methods, LoRA stands out for its simplicity and effectiveness (Zhang et al., 2023; Guo et al., 2025; Sun et al., 2024). Specifically, LoRA introduces two low-rank matrices $\mathbf{A}$ and $\mathbf{B}$

to approximate the weight update:

$$\mathbf{W}' = \mathbf{W} + \Delta\mathbf{W} = \mathbf{W} + \mathbf{BA} \quad (1)$$

where $\mathbf{W}$ is the frozen pre-trained weight. Accordingly, each client in FL trains local LoRA updates $\Delta\mathbf{W}_i = \{\mathbf{A}_i, \mathbf{B}_i\}$, which are aggregated at the server via:

$$\mathbf{A}_g = \frac{1}{N}\sum_{i=1}^{N}\mathbf{A}_i, \quad \mathbf{B}_g = \frac{1}{N}\sum_{i=1}^{N}\mathbf{B}_i \quad (2)$$

The aggregated parameters are then sent back to all clients for the next training round until convergence. This paradigm, referred to as FedLoRA, serves as a representative framework in our study.

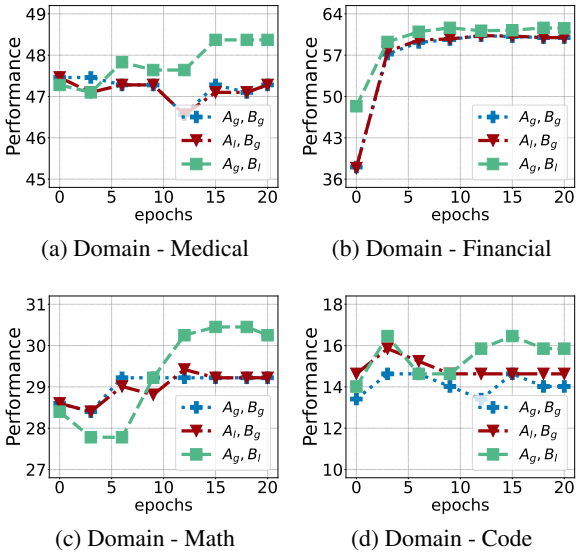
2.2 PEFT Methods' Practical Dilemma

While PEFT methods reduce the computational and communication costs of federated fine-tuning, they often underperform in heterogeneous settings, a common scenario in real-world FL. Recent studies consistently report a significant performance gap between PEFT and full-model fine-tuning under such conditions (Tian et al., 2024; Guo et al., 2025). A straightforward approach is to training more parameters locally, which can enhance the capacity of adapters and mitigate the performance drop. However, this improvement comes at the cost of increased communication, undermining the core efficiency advantages that motivate the use of PEFT. This inherent trade-off between communication efficiency and adaptation performance presents a practical dilemma in the integration of PEFT and FL, and motivates a deeper examination of its root causes and potential architectural alternatives.

2.3 Aggregation Pitfalls in FL

This dilemma raises a key question: *Why do PEFT methods underperform in federated settings, particularly under non-IID data distributions?* We hypothesize that this stems from a fundamental mismatch between FL's averaging mechanism and the design principles of PEFT.

To investigate this hypothesis, we take FedLoRA, a widely adopted PEFT methods in FL, as a case study. Prior works have suggested that the $\mathbf{A}$ matrix tends to encode generalizable information shared across clients, while $\mathbf{B}$ captures more client-specific knowledge (Tian et al., 2024; Zhu et al., 2024). Building on this insight, we design an experiment where each client independently trains its local LoRA adapters $(\mathbf{A}_i, \mathbf{B}_i)$, while the server maintains global LoRA adapters $(\mathbf{A}_g, \mathbf{B}_g)$ through periodic aggregation, as defined in Eq. 2. We evaluate the impact of applying local versus global LoRA matrices on each client's performance. Results in Figure 1 yield the following observation:



(a) Domain - Medical (b) Domain - Financial

(c) Domain - Math (d) Domain - Code

Figure 1: Performance comparison of different LoRA matrix applications across clients. A_g and B_g denote the global A and B matrices, respectively, while A_l and B_l represent the local A and B matrices.

Observation 1: *Across all clients, using local $\mathbf{B}_i$ matrices consistently outperforms using the global average $\mathbf{B}_g$, whereas substituting the global $\mathbf{A}_g$ with local $\mathbf{A}_i$ results in negligible performance differences.*

These results suggest that averaging $\mathbf{B}_i$ fails to retain critical client-specific knowledge, whereas $\mathbf{A}_g$ effectively captures shared patterns across clients. In addition, we monitor the performance of the aggregated matrices $\mathbf{A}_g$ and $\mathbf{B}_g$ across training rounds, evaluating them on each client's local dataset. The results are provided in Figure 5, from which we derive the following observation:

Observation 2: *The global averaged models exhibit significant performance fluctuations within local client data, leading to slower convergence and requiring more communication rounds.*

In summary, our observations highlight a fundamental mismatch: PEFT methods are designed to retain client-specific knowledge within a small set of parameters, whereas the standard averaging aggregation in FL tends to dilute this personalized information. This incompatibility not only degrades performance but also increases the number of communication rounds required for conver-

gence. To this end, we argue that a new aggregation paradigm is needed to integrate the efficiency benefits of PEFT within the FL framework.

3 Methodology

Building on the empirical findings in Section 2.3, we identify two core objectives for designing a PEFT-compatible aggregation strategy: (1) enhancing the global model's ability to retain domain-specific knowledge after aggregation, and (2) reducing the number of communication rounds to minimize overall cost.

We instantiate these principles in a novel framework named **FedELoRA**, which replaces the conventional averaging mechanism with an expert-guided aggregation strategy that preserves client-specific knowledge through a single round of communication. At the core of FedELoRA lies the Expert-Gated LoRA (EGL) network, which integrates all local adapters as expert components and dynamically selects relevant knowledge at inference via a lightweight gating function. We first introduce the EGL network architecture, followed by the three-stage workflow of FedELoRA.

3.1 Expert-Gated LoRA Network

Motivated by our observations in Section 2.3, which show that local $\mathbf{B}_i$ matrices are critical to domain-specific performance, EGL freezes all $\mathbf{B}_i$ after local training to prevent knowledge degradation. In parallel, EGL averages the $\mathbf{A}_i$ matrices to obtain a shared projection $\mathbf{A}_{egl}$, thereby reducing communication cost.

Furthermore, inspired by the Mixture-of-Experts (MoE) framework (Shazeer et al., 2017), a lightweight gating function parameterized by a two-layer MLP W_g assigns weights to each expert during inference. This allows the EGL network to dynamically compose expert knowledge based on the input, enhancing generalization across domains. The overall adaptation can be formulated as:

$$\Delta \mathbf{W}_{egl} = \sum_{i=1}^{N} \omega_i \cdot \mathbf{B}_i \mathbf{A}_{egl}, \tag{3}$$

where $\omega_i = \text{softmax}(W_g^\top x)$ reflect the relevance of each expert to the input sample x, $\mathbf{A}_{egl} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{A}_i$ and N is the number of clients.

3.2 Workflow of FedELoRA

As illustrated in Figure 2, FedELoRA operates in three stages: local fine-tuning, server adaptation, and local inference. This design decouples client-specific specialization from server-side generalization, enabling a communication-efficient and expert-adaptive federated fine-tuning framework.

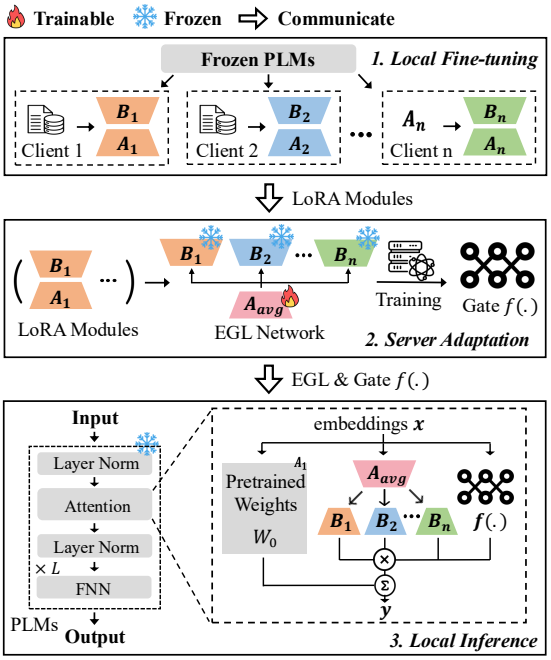


Figure 2: Overview of the FedELoRA architecture and workflow. (1) **local fine-tuning phase**: each client trains its own LoRA adapters and uploads them to the server. (2) **server adaptation phase**: the server uses a trainable gating mechanism to integrate uploaded adapters, constructing an EGL network via fine-tuning on auxiliary data. (3) **local inference phase**: each client uses the trained gate function to dynamically combine expert adapters $\mathbf{B}_i$, enabling flexible adaptation to inputs.

Local Fine-tuning. Each client i initializes from a pre-trained language model and fine-tunes it on its local dataset using the LoRA framework (Hu et al., 2022) until convergence. The resulting LoRA adapters, denoted as $\Delta \mathbf{W}_i = \{\mathbf{A}_i, \mathbf{B}_i\}$, are then uploaded to the server.

Server Adaptation. Upon receiving LoRA adapters from all clients, the server initializes the EGL network as described in Section 3.1. To preserve client-specific knowledge, all $\mathbf{B}_i$ matrices are kept frozen. Instead, the server fine-tunes the aggregated matrix $\mathbf{A}_{egl}$ and the gating function f using a small auxiliary dataset $\mathcal{D}_{aux}$, which can be collected from abundant public resources (Li et al., 2021; Wang et al., 2022). This allows the EGL network to learn adaptive expert selection and improve generalization without multiple communication rounds. Once trained, the EGL network is distributed to all clients.

Local Inference. During inference, each client leverages the received EGL network to perform predictions. Specifically, FedELoRA dynamically integrates the outputs of all expert matrices $\mathbf{B}_i$ as:

$$y = W_0 x + \sum_{i=1}^{N} \omega_i \cdot \mathbf{B}_i \mathbf{A}_{egl} x, \tag{4}$$

where W_0 is the frozen weight of the pre-trained model, x is the input, and ω_i denotes the gating weight for expert i. This formulation enables dynamic inference conditioned on input x, ensuring that the most relevant expert knowledge is utilized for each prediction.

4 Evaluation

In this section, we detail the principal experiments. We begin with an overview of the experimental setup and implementation details. We then share our findings and offer a succinct interpretation.

4.1 Experimental Setup

Dataset and Benchmarks. To comprehensively evaluate the effectiveness of FedELoRA under varying data distribution scenarios, we adopt a diverse set of datasets and benchmarks. For the **IID setting**, we employ the Databricks-Dolly-15k dataset (Conover et al., 2023) where all clients are assumed to share an identical data distribution. Evaluation is conducted on multiple benchmarks, each targeting distinct capabilities: BBH (Suzgun et al., 2022) for general knowledge and reasoning, DROP (Dua et al., 2019) for reading comprehension and numerical reasoning, and HumanEval (Chen et al., 2021) for code generation and functional correctness.

For the **Non-IID setting**, we construct four domain-specific tasks to reflect practical heterogeneous data distributions, where each client is specialized in a distinct domain. The details are as follows: *(1) Medical Domain:* Models are fine-tuned on the Medical Meadow Flashcards dataset (Han et al., 2023), and evaluated on relevant MMLU subtasks (Hendrycks et al., 2020), including anatomy, college biology, college medicine, and medical genetics. *(2) Mathematical Domain:* Fine-tuning is conducted using the MathInstruct dataset (Yue et al., 2023), with evaluation on MMLU sub-tasks related to high school mathematics and statistics. *(3) Financial Domain:* The Financial Sentiment Analysis dataset (FinGPT, 2024) is used for training, and performance is evaluated on the FPB benchmark (Malo et al., 2014). *(4) Coding Domain:* Models are trained on the CodeAlpaca dataset (Chaudhary, 2023), and evaluated using the HumanEval benchmark (Chen et al., 2021) for functional correctness in code generation.

FL Configuration. We conduct main experiments in a 4-client cross-silo FL setting. A more comprehensive analysis of the impact of the number of clients is presented in Section 4.4. For the IID scenario, 2,000 samples are randomly drawn from the Databricks-Dolly-15k dataset (Conover et al., 2023) and evenly split among the four clients. For the Non-IID scenario, we consider two heterogeneity levels: (i) mid heterogeneity, where each client's 5,000-sample dataset is a mixture of the four domains (medical, financial, mathematical, coding) in proportions (0.7, 0.1, 0.1, 0.1); (ii) high heterogeneity, where each client holds 5,000 samples exclusively from one domain.

Comparison Baselines. We compare FedELoRA with five baselines, described as follows: (i) **LoRA**(Hu et al., 2022): Each client independently fine-tunes the LoRA adapter on its local data without communication. (ii) **FedLoRA**: Each client transmits both the $\mathbf{A}$ and $\mathbf{B}$ matrices to the server for global aggregation. (iii) **FFA-LoRA**(Sun et al., 2024): Each client freezes $\mathbf{A}$ matrix, trains only the $\mathbf{B}$ matrix, and sends it to the server for global aggregation. (iv) **FedSA**(Guo et al., 2025): Clients transmit only $\mathbf{A}$ matrices to the server for global aggregation, while locally train $\mathbf{B}$ matrices. (v) **RoLoRA**(Chen et al., 2024): The clients alternate between updating the $\mathbf{A}$ matrices with frozen $\mathbf{B}$ matrices and vice versa across rounds.

Implementation Details. We adopt the pre-trained LLaMa-2-7B model[1] with 8-bit quantization as the backbone. Local training on each client is performed for 3 epochs using the AdamW optimizer (Loshchilov, 2017), with a batch size of 16 and 20 total communication rounds. A cosine learning rate schedule is applied, following Ye et al. (2024), starting at 5e-5 and decaying to 1e-6. For domain-specific datasets, we set the maximum sequence length to 512 and use LoRA with rank 32 and scaling factor $\alpha = 64$. For general datasets, the sequence length is increased to 1024, with LoRA rank 4 and $\alpha = 32$. The proportion λ of the auxiliary dataset $\mathcal{D}_{\text{aux}}$ is set to 0.1. All experiments are

[1] https://huggingface.co/NousResearch/Llama-2-7b-hf

#Hetero	Method	Evaluation Benchmarks					Cost	
		Medical	Financial	Math	Code	Rank	%Params	Commu.
Mid	LoRA - Med	46.56 (5)	58.25 (5)	28.81 (3)	13.41 (6)	4.75	0.248	–
	LoRA - Fin	45.83 (8)	59.24 (3)	27.57 (5)	14.63 (4)	5.00	0.248	–
	LoRA - Math	45.83 (8)	55.20 (7)	29.63 (2)	10.98 (9)	6.50	0.248	–
	LoRA - Code	46.20 (7)	54.21 (8)	26.95 (7)	15.24 (3)	6.25	0.248	–
	FedLoRA	47.46 (2)	**59.82** (1)	27.16 (6)	13.41 (6)	3.75	0.248	10.00
	FFA-LoRA	**48.01** (1)	53.63 (9)	26.34 (9)	**15.85** (1)	5.00	0.124	5.00
	FedSA	46.56 (5)	59.32 (2)	26.75 (8)	**15.85** (1)	4.00	0.248	5.00
	RoLoRA	47.28 (4)	58.00 (6)	27.98 (4)	13.41 (6)	5.00	0.248	5.00
	FedELoRA	47.46 (2)	59.24 (3)	**31.07** (1)	14.63 (4)	**2.50**	0.283	0.77
High	LoRA - Med	47.83 (2)	55.25 (6)	28.19 (7)	12.20 (9)	6.00	0.248	–
	LoRA - Fin	47.10 (4)	58.82 (3)	29.84 (3)	12.80 (7)	4.25	0.248	–
	LoRA - Math	47.46 (3)	50.99 (9)	30.45 (2)	12.80 (7)	5.25	0.248	–
	LoRA - Code	46.74 (8)	53.14 (8)	27.57 (8)	**15.85** (1)	6.25	0.248	–
	FedLoRA	46.92 (7)	**60.97** (1)	27.57 (8)	14.02 (5)	5.25	0.248	10.00
	FFA-LoRA	47.10 (4)	54.21 (7)	28.81 (6)	15.24 (2)	4.75	0.124	5.00
	FedSA	47.10 (4)	60.48 (2)	29.22 (4)	15.24 (2)	3.00	0.248	5.00
	RoLoRA	46.74 (8)	58.17 (4)	29.01 (5)	13.41 (6)	5.75	0.248	5.00
	FedELoRA	**49.28** (1)	58.00 (5)	**31.28** (1)	14.63 (4)	**2.75**	0.283	0.77

Table 1: Comparison with baseline methods in different levels of data heterogeneity. Results show performance metrics and communication costs (Comm. in GB).

conducted on NVIDIA A100 GPUs.

4.2 Overall Performance

As evaluation metrics across different tasks vary in scale, we adopt the average rank metric to measure the models' overall performance, which was adopted in previous work (Ye et al., 2024). The detailed results for the Non-IID and IID scenarios are presented in Table 1 and Table 4, respectively.

The results demonstrate that FedELoRA consistently achieves the highest average rank across various data heterogeneity levels, indicating superior generalization performance compared to the baseline methods. Furthermore, FedELoRA exhibits significant performance improvements across multiple subtasks. For instance, in the Math task under the medium heterogeneity scenario, FedELoRA achieved a score of 31.07, while the best performance from the baseline methods was 27.98. Notably, FedELoRA also performs well in the IID scenario. We attribute this to the larger adapter parameter space in FedELoRA, which enhances its ability to capture the knowledge of local models and ultimately leads to better performance than the comparison methods.

Additionally, FedELoRA's communication overhead is substantially lower than that of the baseline methods across all scenarios. Specifically, it requires only 15.4% of the communication cost of the baseline method with the lowest overhead. This is attributed to FedELoRA's communication-efficient mechanism, which eliminates the need for multiple communication rounds. This advantage enables FedELoRA to improve model performance by increasing the number of fine-tuning parameters in practical applications.

Method	Evaluation Benchmarks			
	Med.	Fin.	Math	Code
FedELoRA	**49.28**	**58.00**	**31.28**	**14.63**
w/o EGL	48.19	55.61	29.42	12.80
w/o gate	47.83	52.97	29.42	12.80

Table 2: Ablation study for FedELoRA.

4.3 Ablation Study

FedELoRA integrates the EGL network to preserve client-specific expert knowledge and employs a dynamic gating mechanism to select the most relevant expert during inference. We conduct ablation experiments in highly heterogeneous scenarios to explore the effects of these two strategies.

N	Method	Evaluation Benchmarks					Cost	
		Medical	Financial	Math	Code	Rank	%Params	Commu.
2	FedLoRA	47.64 (2)	**60.97** (1)	28.60 (3)	15.24 (2)	2.00	0.248	5.00
	FFA-LoRA	46.74 (4)	58.25 (4)	26.13 (4)	14.02 (4)	4.00	0.124	2.50
	FedSA	46.92 (3)	60.81 (2)	29.42 (2)	**15.85** (1)	2.00	0.248	2.50
	RoLoRA	45.65 (5)	57.43 (5)	25.51 (5)	13.41 (5)	5.00	0.248	2.50
	FedELoRA	**48.73** (1)	58.50 (3)	**30.45** (1)	15.24 (2)	**1.75**	0.314	0.25
4	FedLoRA	46.92 (4)	**60.97** (1)	27.57 (5)	14.02 (4)	3.50	0.248	10.00
	FFA-LoRA	47.10 (2)	54.21 (5)	28.81 (4)	**15.24** (1)	3.00	0.124	5.00
	FedSA	47.10 (2)	60.48 (2)	29.22 (2)	**15.24** (1)	**1.75**	0.248	5.00
	RoLoRA	46.74 (5)	58.17 (3)	29.01 (3)	13.41 (5)	4.00	0.248	5.00
	FedELoRA	**49.28** (1)	58.00 (4)	**31.28** (1)	14.63 (3)	2.25	0.283	0.77
8	FedLoRA	47.10 (2)	**61.14** (1)	27.78 (3)	13.41 (3)	2.25	0.248	20.00
	FFA-LoRA	46.74 (4)	54.79 (3)	27.16 (4)	12.20 (4)	3.75	0.124	10.00
	FedSA	OOM	OOM	OOM	OOM	OOM	0.248	10.00
	RoLoRA	47.10 (2)	58.09 (2)	29.22 (2)	14.02 (2)	2.00	0.248	10.00
	FedELoRA	**48.73** (1)	54.04 (4)	**30.66** (1)	**15.85** (1)	**1.75**	0.268	2.56

Table 3: Comparison with baseline methods in different number of clients N. Results show performance metrics and communication costs (Comm. in GB).

Specifically, we consider two variants of FedELoRA: (i) without Expert-Gated LoRA (w/o EGL), which removes the EGL architecture, reverting FedELoRA to FedLoRA with a single communication round; (ii) without the gating mechanism (w/o Gate), which assigns uniform weights to experts instead of leveraging the gate function. As shown in Table 2, both ablated variants perform significantly worse than FedELoRA across all tasks. These results highlight the essential roles of both the EGL architecture and the adaptive gating mechanism in ensuring the effectiveness of FedELoRA.

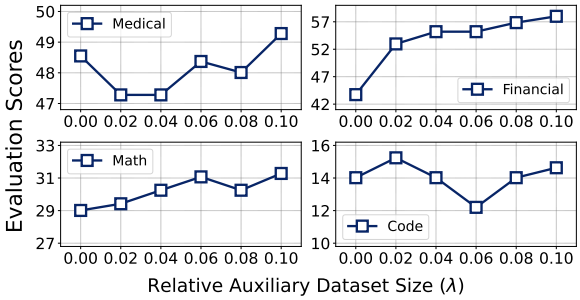


Figure 3: The performance of FedELoRA across different task w.r.t the proportion λ of $\mathcal{D}_{\text{aux}}$.

4.4 Hyper-parameter Analysis

This section investigates several factors that may affect the performance of FedELoRA, including one method-dependent factor: the proportion λ of the auxiliary dataset $\mathcal{D}_{\text{aux}}$, and two method-independent factors: the number of clients N and the rank r of the LoRA adapters.

The proportion λ of $\mathcal{D}_{\text{aux}}$. To assess the impact of the auxiliary dataset $\mathcal{D}_{aux}$ on FedELoRA, we configure varying proportions λ of $\mathcal{D}_{aux}$, where when $\lambda = 0$, the gating function is the mean function. As shown in Figure 3, On different tasks, the overall performance of FedELoRA increases with the increase of λ, which is in line with our expectations, because more auxiliary datasets can help FedELoRA learn how to schedule experts. In addition, with only 0.1 ratio of auxiliary datasets, FedELoRA can achieve a significant performance improvement, proving its practicality.

The number of clients N. As discussed in Section 2.1, unlike traditional FL, the fine-tuning of PLMs typically involves a limited number of clients. Therefore, we configure N to vary between 2 and 8 in order to evaluate the performance of FedELoRA across different client settings. As shown in Table 3, despite not achieving optimal performance on some tasks, FedELoRA consistently outperforms the comparison method in terms of its overall generalization performance (Rank), demonstrating its robust scalability.

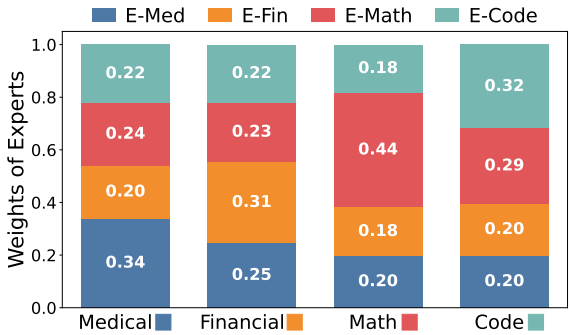


Figure 4: Expert weight distributions across tasks. Each bar shows the normalized contributions of four experts to a specific domain.

The rank r of LoRA Adapters. We evaluate the performance of FedELoRA alongside baseline methods for varying ranks r of LoRA adapters, with the results presented in Table 5. FedELoRA consistently outperforms the baseline methods in terms of generalization performance across different values of r, while also maintaining significantly lower communication overhead. Notably, although the number of trainable parameters increases as r grows, the communication cost of FedELoRA is much lower than that of the baseline methods. Therefore, in practical applications, FedELoRA can utilize a larger r to improve performance without incurring substantial communication overhead.

4.5 Case Study

To further understand how FedELoRA utilizes expert knowledge during inference, we visualize the expert weights across four domains, as shown in Figure 4. Each bar in the visualization is normalized to have a total weight of 1, where the length of each colored segment indicates the relative contribution of a specific expert. The results reveal that the contribution of different expert matrices varies significantly across tasks, with domain-specific experts contributing more prominently within their respective domains. This trend indicates that FedELoRA is able to dynamically and effectively leverage relevant expert knowledge during inference, thereby enhancing its adaptability and performance across heterogeneous tasks.

5 Related Work

Recent studies have shown that LoRA can achieve performance comparable to full-parameter fine-tuning, making it increasingly attractive in federated settings (Zhang et al., 2023; Han et al., 2024; Bian et al., 2025). Existing approaches in this area generally fall into two categories. The first line of work focuses on addressing performance degradation caused by directly aggregating LoRA updates across clients. For instance, FFA-LoRA (Sun et al., 2024) fixes the randomly initialized matrix A and only fine-tunes matrix B. FLora (Wang et al., 2024) introduces a stacked aggregation mechanism that reduces noise during module merging, albeit with increased communication overhead. RoLoRA (Chen et al., 2024) further proposes an alternating optimization scheme to improve the robustness of LoRA adaptation in federated environments.

The second category targets challenges arising from data heterogeneity. C2A (Kim et al., 2023) generates client-specific LoRA modules based on local data distributions. Other works tailor LoRA configurations to device capabilities using techniques such as zero-padding (Cho et al., 2024) or module duplication (Byun and Lee, 2024). FedSA (Guo et al., 2025) selectively aggregates only matrix A to capture global knowledge, while keeping matrix B local to support personalization and reduce communication costs.

Our approach diverges from prior work in both motivation and methodology. Rather than relying on the multi-round averaging paradigm, which is misaligned with the principles of PEFT methods, we propose a novel expert-guided one-shot aggregation strategy. This design better captures the goals of both personalization and communication efficiency in federated PEFT settings.

6 Conclusion

In this work, we identify a key limitation of applying PEFT methods to FL: the multi-round averaging aggregation mechanism fails to preserve client-specific knowledge, leading to degraded performance and increased communication cost. To this end, we propose a novel aggregation paradigm that replaces the averaging mechanism with expert-guided aggregation, enabling the retention of client-specific knowledge in a single communication round. We instantiate this strategy with FedELoRA, a novel LoRA-based framework for FL. Empirical results demonstrate that FedE-LoRA consistently achieves the best average rank in both IID and non-IID settings while reducing communication overhead. We believe that our findings offers a broader insight into the integration of PEFT and FL.

Limitations

Despite the promising results of FedELoRA in reducing communication overhead and enhancing model performance, several limitations persist in our study. First, our research focuses on integrating federated learning with parameter-efficient fine-tuning techniques, with FedELoRA serving as the primary instantiation of this approach. However, other potential methods, such as Prompt Tuning, remain unexplored within this framework, limiting the scope of our investigation into alternative paradigms. Second, our method requires the server to collect publicly available auxiliary datasets and perform fine-tuning. While our experiments, detailed in Experiment 4 and Appendix, demonstrate that this approach does not place stringent demands on the quality or size of the auxiliary datasets, it does necessitate greater computational resources on the server side compared to traditional federated learning setups. This increased resource demand may hinder the practical applicability of FedELoRA in resource-constrained real-world scenarios. Finally, due to computational constraints associated with fine-tuning large-scale models, we were unable to conduct experiments involving hundreds of clients in a cross-device setting. Instead, our evaluation relied on a smaller number of clients within a cross-silo configuration, which may not fully reflect the challenges of broader, device-diverse deployments.

Ethical Considerations

In developing FedELoRA to advance efficient fine-tuning for federated large models, we have exercised significant caution in our data practices. All datasets utilized in this study are sourced from widely recognized and previously published works, ensuring they are free of personally identifiable information. Moreover, the evaluation benchmarks we adopted align with those established in prior research, effectively eliminating risks of privacy violations or data breaches.

Acknowledgments

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sara Babakniya, Ahmed Elkordy, Yahya Ezzeldin, Qingfeng Liu, Kee-Bong Song, MOSTAFA ELKhamy, and Salman Avestimehr. 2023. SLoRA: Federated parameter efficient fine-tuning of language models. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*.

Jieming Bian, Yuanzhe Peng, Lei Wang, Yin Huang, and Jie Xu. 2025. A survey on parameter-efficient fine-tuning for foundation models in federated learning. *arXiv preprint arXiv:2504.21099*.

Yuji Byun and Jaeho Lee. 2024. Towards federated low-rank adaptation of language models with rank heterogeneity. *arXiv preprint arXiv:2406.17477*.

Sahil Chaudhary. 2023. Code alpaca: An instruction-following llama model for code generation. https://github.com/sahil280114/codealpaca.

Tianshi Che, Ji Liu, Yang Zhou, Jiaxiang Ren, Jiwen Zhou, Victor S. Sheng, Huaiyu Dai, and Dejing Dou. 2023. Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization. In *EMNLP*, pages 7871–7888. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

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

Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, and Gauri Joshi. 2024. Heterogeneous lora for federated fine-tuning of on-device foundation models. In *EMNLP*, pages 12903–12913. Association for Computational Linguistics.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. Accessed: 2023-06-30.

Tianyu Cui, Hongxia Li, Jingya Wang, and Ye Shi. 2024. Harmonizing generalization and personalization in federated prompt learning. In *ICML*. OpenReview.net.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*.

FinGPT. 2023. fingpt-sentiment-cls.

FinGPT. 2024. fingpt-sentiment-train.

Sajjad Ghiasvand, Yifan Yang, Zhiyu Xue, Mahnoosh Alizadeh, Zheng Zhang, and Ramtin Pedarsani. 2024. Communication-efficient and tensorized federated fine-tuning of large language models. *arXiv preprint arXiv:2410.13097*.

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

Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*, 3.

Tianyu Han, Lisa C Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K Bressem. 2023. Medalpaca–an open-source collection of medical conversational ai models and training data. *arXiv preprint arXiv:2304.08247*.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*. OpenReview.net.

Iamtarun. 2023. code_instructions_120k_alpaca.

Yeachan Kim, Junho Kim, Wing-Lam Mok, Jun-Hyung Park, and SangKeun Lee. 2023. Client-customized adaptation for parameter-efficient federated learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1159–1172.

Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. 2024. Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5260–5271.

Anran Li, Lan Zhang, Juntao Tan, Yaxuan Qin, Junhao Wang, and Xiang-Yang Li. 2021. Sample-level data selection for federated learning. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60.

I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796.

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

MedAlpaca. 2023. medical_meadow_wikidoc.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

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

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Chengzhong Xu. 2024. Hydralora: An asymmetric lora architecture for efficient fine-tuning. *arXiv preprint arXiv:2404.19245*.

TIGER-Lab. 2024. Math-plus.

Paul Voigt and Axel Von dem Bussche. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555.

Junhao Wang, Lan Zhang, Anran Li, Xuanke You, and Haoran Cheng. 2022. Efficient participant contribution evaluation for horizontal and vertical federated learning. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 911–923. IEEE.

Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. 2024. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. *arXiv preprint arXiv:2409.05976*.

Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. 2024. Openfedllm: Training large language models on decentralized private data via federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6137–6147.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.

Zhuo Zhang, Yuanhang Yang, Yong Dai, Qifan Wang, Yue Yu, Lizhen Qu, and Zenglin Xu. 2023. Fedpetuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In *Annual Meeting of the Association of Computational Linguistics 2023*, pages 9963–9977. Association for Computational Linguistics (ACL).

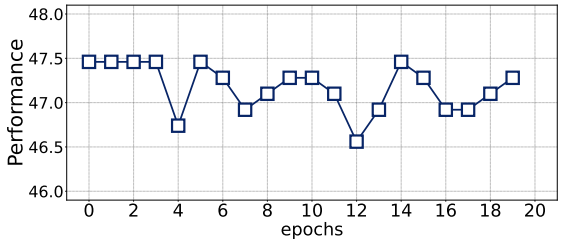
Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Jiacheng Zhu, Kristjan H. Greenewald, Kimia Nadjahi, Haitz Sáez de Ocáriz Borde, Rickard Brüel Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. 2024. Asymmetry in low-rank adapters of foundation models. In *ICML*. OpenReview.net.

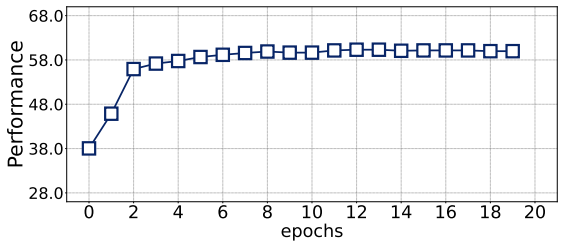
A Appendix

A.1 Motivation Experiments

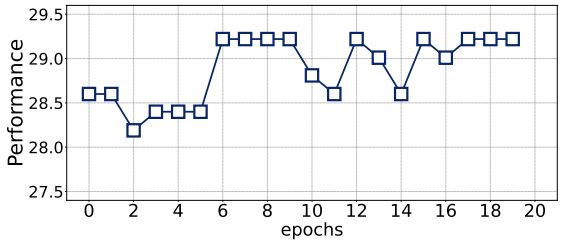
Performance variation of global models. We evaluated the performance of the globally aggregated model on each client's local dataset, with the results presented in Figure 5. The figure reveals that the model's performance exhibits significant variability across different domains, necessitating additional communication rounds to achieve convergence.



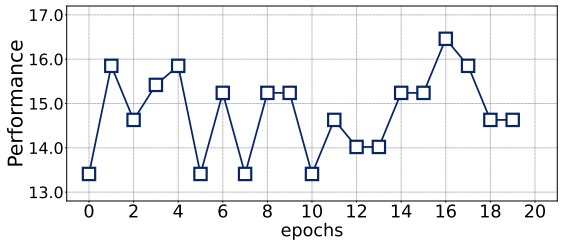
(a) Domain - Medical



(b) Domain - Financial



(c) Domain - Math



(d) Domain - Code

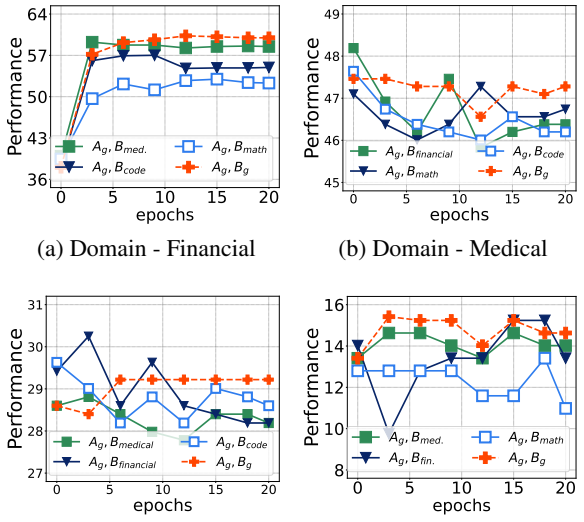
Figure 5: Performance variation of global models across datasets from diverse domains.

Performance with local B matrices. Although average aggregation is not an effective mechanism, our experiments also reveal that relying solely on

Method	Evaluation Benchmarks					Cost	
	BBH	DROP	HumanEval	CRASS	Rank	%Params	Commu
LoRA	31.67 (3)	**34.08** (1)	14.33 (6)	43.38 (4)	3.50	0.031	–
FedLoRA	31.45 (5)	33.94 (4)	14.63 (3)	44.10 (2)	3.50	0.031	1.25
FFA-LoRA	31.68 (2)	33.89 (5)	**15.24** (1)	42.24 (5)	3.25	0.016	0.63
FedSA	31.64 (4)	33.97 (2)	14.63 (3)	44.10 (2)	2.75	0.031	0.63
RoLoRA	31.04 (6)	32.92 (6)	14.63 (3)	39.75 (6)	5.25	0.031	0.63
FedELoRA	**32.03** (1)	33.95 (3)	**15.24** (1)	**45.96** (1)	**1.50**	0.039	0.11

Table 4: Comparison with baseline methods in IID setting (Comm. in GB).

local $\mathbf{B}$ matrices will compromises the generalization of fine-tuned models. Specifically, we apply each client's local $\mathbf{B}$ matrix individually and evaluate the model performance on the datasets of the other three clients. As shown in Figure 6, the generalization achieved with local $\mathbf{B}$ matrices is notably weaker than that of the global $\mathbf{B}_g$ matrix. Therefore, there is a need for a novel aggregation mechanism that effectively integrates diverse expert knowledge while minimizing training iterations.



(a) Domain - Financial (b) Domain - Medical

(c) Domain - Math (d) Domain - Code

Figure 6: Performance comparison of global and non-local B matrix applications across clients. $B_{med.}$ and $B_{fin.}$ denote B matrices trained on medical and financial dataset clients, respectively.

A.2 Overall Performance Results

Due to space limitations, we present the performance of FedELoRA and the comparison methods under the IID scenario in Table 4.

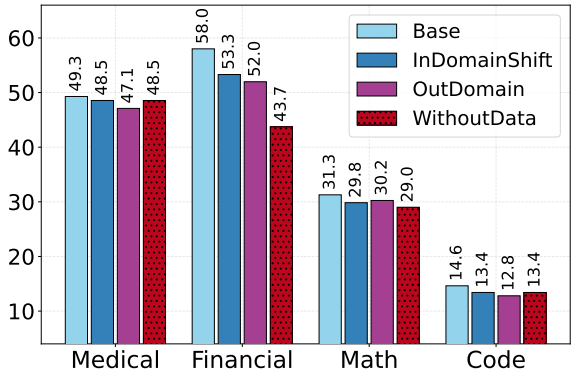


Figure 7: Performance of FedELoRA under different auxiliary dataset settings.

A.3 The Impact of Ranks

We present a performance comparison of FedELoRA with baseline methods for LoRA adapters at different ranks r in Table 5.

A.4 The Impact of Auxiliary Datasets

In this section, we evaluate the generalization ability of our proposed method, FedELoRA, under different auxiliary dataset configurations:

- **FedELoRA-Base**: No distribution shift between the auxiliary dataset and training data.
- **FedELoRA-InDomainShift**: A domain-relevant but distribution-shifted auxiliary dataset is used for adaptation. Specifically, in the medical domain, we use `medical_meadow_wikidoc`(MedAlpaca, 2023), a medical QA dataset; in the financial domain, `fingpt_sentiment_cls` (FinGPT, 2023), a sentiment classification dataset; for mathematical reasoning, `Math-Plus` (TIGER-Lab, 2024); and for code generation, `code_instructions_120k_alpaca` (Iamtarun, 2023).

r	Method	Evaluation Benchmarks					Cost	
		Medical	Financial	Math	Code	Rank	%Params	Commu.
8	FedLoRA	46.92 (4)	57.84 (3)	28.19 (3)	15.24 (2)	3.00	0.062	2.50
	FFA-LoRA	46.92 (4)	40.35 (5)	27.98 (4)	14.02 (5)	4.50	0.031	1.25
	FedSA	47.28 (2)	**58.83** (1)	29.42 (2)	**15.85** (1)	**1.50**	0.062	1.25
	RoLoRA	47.28 (2)	50.33 (4)	27.78 (5)	15.24 (2)	3.25	0.062	1.25
	FedELoRA	**48.19** (1)	57.92 (2)	**29.63** (1)	15.24 (2)	**1.50**	0.074	0.20
16	FedLoRA	46.38 (5)	**59.32** (1)	29.22 (3)	13.41 (5)	3.50	0.124	5.00
	FFA-LoRA	47.10 (3)	46.53 (5)	27.37 (5)	14.63 (2)	3.75	0.062	2.50
	FedSA	46.74 (4)	58.99 (2)	29.42 (2)	14.02 (3)	2.75	0.124	2.50
	RoLoRA	47.28 (2)	56.11 (4)	28.60 (4)	**15.85** (1)	2.75	0.124	2.50
	FedELoRA	**48.01** (1)	57.67 (3)	**30.04** (1)	14.02 (3)	**2.00**	0.144	0.39
32	FedLoRA	46.92 (4)	**60.97** (1)	27.57 (5)	14.02 (4)	3.50	0.248	10.00
	FFA-LoRA	47.10 (2)	54.21 (5)	28.81 (4)	**15.24** (1)	3.00	0.124	5.00
	FedSA	47.10 (2)	60.48 (2)	29.22 (2)	**15.24** (1)	**1.75**	0.248	5.00
	RoLoRA	46.74 (5)	58.17 (3)	29.01 (3)	13.41 (5)	4.00	0.248	5.00
	FedELoRA	**49.28** (1)	58.00 (4)	**31.28** (1)	14.63 (3)	2.25	0.283	0.77
64	FedLoRA	47.28 (2)	**62.71** (1)	27.57 (4)	12.80 (5)	3.00	0.495	20.00
	FFA-LoRA	47.28 (2)	58.50 (5)	26.75 (5)	13.41 (4)	4.00	0.248	10.00
	FedSA	46.92 (4)	60.56 (2)	28.81 (2)	**14.63** (1)	2.25	0.495	10.00
	RoLoRA	46.56 (5)	59.90 (3)	28.19 (3)	**14.63** (1)	3.00	0.495	10.00
	FedELoRA	**49.09** (1)	59.74 (4)	**30.04** (1)	**14.63** (1)	**1.75**	0.563	1.52

Table 5: Performance of FedELoRA and baseline methods with varying LoRA adapter ranks r.

- **FedELoRA-OutDomain**: A general-purpose instruction tuning dataset, `databricks-dolly-15k` (Conover et al., 2023), is used, which is not aligned with the client domain.
- **FedELoRA-WithoutData**: No auxiliary dataset is provided, and the aggregated model is directly used for inference without further fine-tuning.

Figure 7 presents the performance of FedELoRA under various auxiliary dataset configurations across four domains: Medical, Financial, Math, and Code. FedELoRA exhibits strong robustness across various auxiliary data settings. Even with domain shifts, performance remains stable when the data is topically aligned. In contrast, using mismatched or no auxiliary data leads to larger drops, especially in domain-sensitive tasks.