FedALT: Federated Fine-Tuning through Adaptive Local Training with Rest-of-World LoRA

Jieming Bian[1*], **Lei Wang**[1*], **Letian Zhang**[2], **Jie Xu**[1]
[1]University of Florida, Gainesville, FL, USA
[2]Middle Tennessee State University, Murfreesboro, TN, USA
jieming.bian@ufl.edu, leiwang1@ufl.edu, letian.zhang@mtsu.edu, jie.xu@ufl.edu

Abstract

Fine-tuning large language models (LLMs) in federated settings enables privacy-preserving adaptation but suffers from cross-client interference due to model aggregation. Existing federated LoRA fine-tuning methods, primarily based on FedAvg, struggle with data heterogeneity, leading to harmful cross-client interference and suboptimal personalization. In this work, we propose **FedALT**, a novel personalized federated LoRA fine-tuning algorithm that fundamentally departs from FedAvg. Instead of using an aggregated model to initialize local training, each client continues training its individual LoRA while incorporating shared knowledge through a separate Rest-of-World (RoW) LoRA component. To effectively balance local adaptation and global information, FedALT introduces an adaptive mixer that dynamically learns input-specific weightings between the individual and RoW LoRA components, drawing conceptual foundations from the Mixture-of-Experts (MoE) paradigm. Through extensive experiments on NLP benchmarks, we demonstrate that FedALT significantly outperforms state-of-the-art personalized federated LoRA fine-tuning methods, achieving superior local adaptation without sacrificing computational efficiency.

Introduction

Large language models (LLMs) (Kenton and Toutanova 2019; Brown et al. 2020; Raffel et al. 2020; Touvron et al. 2023a; Zhou et al. 2024; Zeng et al. 2022; Touvron et al. 2023b) have demonstrated exceptional capabilities in language understanding and generation, enabling a wide range of applications. Adapting pretrained LLMs to specialized domains or further enhancing their performance requires fine-tuning with task-specific data (Minaee et al. 2024). However, full fine-tuning, which involves updating all model parameters, is computationally expensive and often impractical for real-world deployment. To address this limitation, several parameter-efficient fine-tuning (PEFT) (Ding et al. 2023; Fu et al. 2023; Liu et al. 2022; Li and Liang 2021; Lester, Al-Rfou, and Constant 2021) methods have been proposed, with low-rank adaptation (LoRA) (Hu et al. 2021) being one of the most widely used. LoRA reduces the number of trainable parameters by integrating low-rank matrices into the model, significantly reducing computational costs.

Fine-tuning LLMs necessitates substantial data volumes for adaptation to downstream tasks; however, this requisite data often resides across multiple institutions where privacy regulations and security protocols prohibit direct data sharing. Federated Learning (FL) facilitates collaborative fine-tuning without raw data exposure, establishing itself as an essential paradigm for privacy-preserving adaptation (Bian et al. 2025a). Nonetheless, existing federated LoRA fine-tuning methodologies (Zhang et al. 2024; Sun et al. 2024; Wang et al. 2024c; Bian et al. 2025b) predominantly presuppose a singular global model, neglecting the inherent data heterogeneity among participating clients. Such a homogeneous modeling approach frequently yields suboptimal performance and introduces fairness disparities, particularly in real-world deployments where client data exhibits significant variance in volume, domain, and distributional characteristics. While limited research on personalized federated LoRA fine-tuning (Hao et al. 2025; Qi et al. 2024) exists, these approaches primarily address label heterogeneity cases. However, in realistic LLM fine-tuning contexts, clients' learning objectives naturally diverge due to diverse business imperatives (Yao et al. 2022). Personalized federated LoRA fine-tuning addressing more realistic task-heterogeneous scenarios (Yang et al. 2024a) remains insufficiently investigated in the current literature.

Federated Averaging (FedAvg) (McMahan et al. 2017) is perhaps the most widely used FL framework, serving as the foundation for numerous follow-up works that introduce various optimizations. Its core principle is to aggregate locally trained client models into a global model, which then initializes the next round of local training. Most personalized FL algorithms (Yang et al. 2024a; Qi et al. 2024; Tan et al. 2022; Deng, Kamani, and Mahdavi 2020; Collins et al. 2021; Fallah, Mokhtari, and Ozdaglar 2020), despite their implementation differences, build upon the FedAvg framework by incorporating personalization mechanisms—and personalized federated LoRA fine-tuning follows the same paradigm. For example, the recently proposed FedDPA (Yang et al. 2024a) algorithm personalizes LoRA by introducing both global and local LoRA components. The global LoRA component is trained using FedAvg, while the local LoRA component is trained independently on each client's private data. This local adaptation can occur either after the global LoRA stabilizes or alternatedly

*These authors contributed equally.

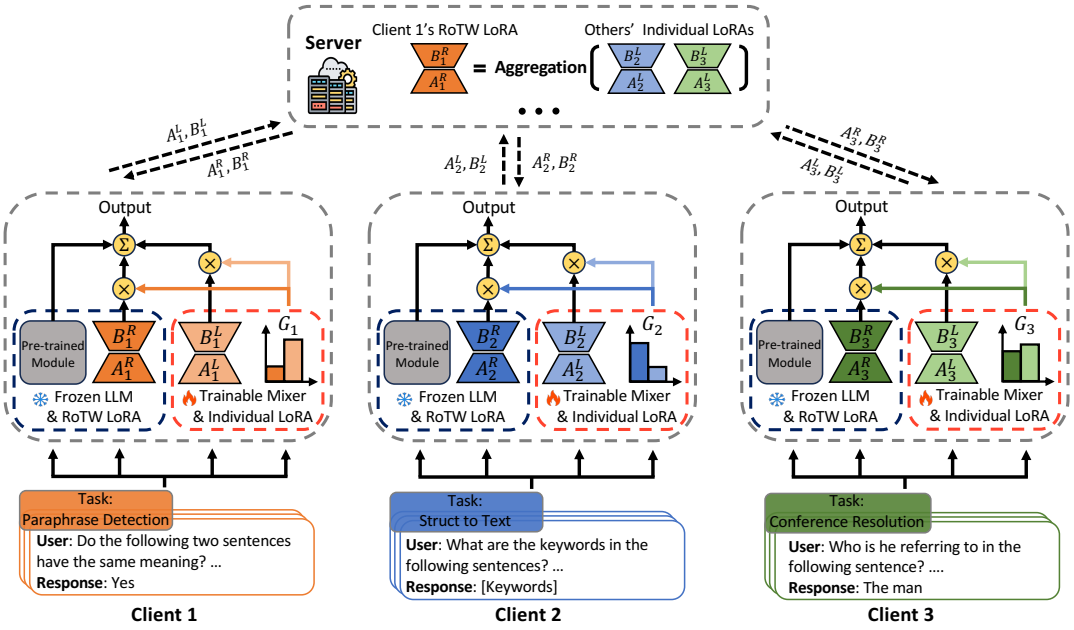


Figure 1: Illustration of FedALT. Instead of directly aggregating local LoRA modules from each client using FedAvg, FedALT introduces a frozen RoW LoRA component to transmit shared global knowledge while preserving client-specific adaptations through Individual LoRA. The adaptive mixer dynamically combines the RoW LoRA and Individual LoRA.

with global LoRA training, in which case the final output is computed as a weighted combination of both components using a pre-defined weighting factor. However, our motivational study reveals that FedAvg-based approaches may degrade client-specific performance due to harmful cross-client interference. We identify two fundamental limitations of FedAvg-based methods like FedDPA in heterogeneous FL settings. **First**, they suffer from harmful cross-client interference due to reliance on global aggregation, which can negate individual client improvements achieved during local fine-tuning. **Second**, these methods lack an effective mechanism for balancing global and local information, relying on fixed or predefined weighting factors, which can be suboptimal for clients poorly represented by the global model.

To address these critical limitations, we propose a novel personalized federated LoRA fine-tuning approach called FedALT (short for Federated Fine-tuning through **A**daptive **L**ocal **T**raining with Rest-of-World LoRA.). Distinctly different from FedAvg-based methods, FedALT does not aggregate local models to initialize training rounds. Instead, each client continues training on it own previously trained local models, supplemented by a frozen **Rest-of-World (RoW) LoRA** component. The RoW LoRA captures global knowledge without updating during local training rounds, thus avoiding interference with client-specific adaptations. To leverage RoW LoRA effectively, we introduce an **adaptive mixer**, inspired by Mixture-of-Experts (MoE) (Jordan and Jacobs 1994), which dynamically and optimally combines global and local information for each client's data sample. Our contributions are as follows:

- We highlight that personalized federated fine-tuning for clients with **heterogeneous tasks** has been insufficiently explored and identify critical limitations in existing methods that build upon the FedAvg paradigm.
- We propose FedALT, a novel personalization framework featuring a frozen RoW LoRA and an adaptive mixer, effectively addressing these limitations by dynamically balancing global and local model adaptations.
- We perform comprehensive experiments on two LLMs (Bloom (Le Scao et al. 2023) and Llama 2 (Touvron et al. 2023b)) using the Flan benchmark (Chung et al. 2024), demonstrating that FedALT significantly outperforms existing federated LoRA fine-tuning methods.

Related Work

Parameter-Efficient Fine-Tuning

As the size of LLMs continues to grow, full parameter fine-tuning has become increasingly computationally and storage-intensive (Hadi et al. 2023). To address this issue, Parameter-Efficient Fine-Tuning (PEFT) methods (Ding et al. 2023; Fu et al. 2023; Han et al. 2024; Liu et al. 2022; Li and Liang 2021; Lester, Al-Rfou, and Constant 2021) have been developed to significantly reduce the number of trainable parameters. PEFT techniques introduce a limited set of additional trainable parameters to enhance model performance while keeping the majority of pre-trained parameters frozen. Among all existing PEFT methods, the most prominent approach is LoRA (Hu et al. 2021), which employs low-rank matrices to approximate the pre-trained weight matrix, updating only the low-rank components. LoRA has became a standard method for adapting LLMs like Llama under limited computational resources (Lermen, Rogers-Smith, and

Ladish 2023). Several works have been proposed to improve LoRA (Tian et al. 2024; Lermen, Rogers-Smith, and Ladish 2023; Wang et al. 2023; Sheng et al. 2023; Liu et al. 2023). A recent development, Hydra-LoRA (Tian et al. 2024), employs an asymmetric structure with a shared matrix for all samples and distinct matrices for each intrinsic component, thereby enhancing domain adaptation. While these advancements primarily focus on centralized learning settings, our work extends the application of PEFT methods to federated LLM fine-tuning, addressing the unique challenges posed by distributed and heterogeneous client data.

Federated Learning

The foundational work, FedAvg (McMahan et al. 2017), addresses privacy and communication efficiency by aggregating local model to train a shared global model. Since then, numerous studies (Liu et al. 2025a; Wang et al. 2024b; Yang et al. 2024b; Liu et al. 2025c; Bian et al. 2024; Zhang et al. 2025; Liu et al. 2025b, 2024) have focused on tackling various challenges within FL settings. One of the key challenges is data heterogeneity, which makes it difficult to train a single shared global model that performs well across all clients. To address this, Personalized Federated Learning (PFL) (Tan et al. 2022; Deng, Kamani, and Mahdavi 2020; Collins et al. 2021; Fallah, Mokhtari, and Ozdaglar 2020) has emerged as an approach to adapt the global model to the specific needs of each client. Most PFL research has focused on addressing statistical differences in data distributions (e.g., label distribution skew) and has been applied primarily to smaller, standard models. In contrast, our work addresses the unique challenges posed by heterogeneous clients with diverse tasks by leveraging LLM fine-tuning within the FL framework.

Federated Fine-Tuning

Several studies (Cho et al. 2024; Wang et al. 2024c; Kuang et al. 2024; Wu et al. 2024) have explored federated fine-tuning approaches. (Kuang et al. 2024) proposed federated full parameters fine-tuning, while (Sun et al. 2022) introduced federated fine-tuning with PEFT using prefix-tuning. (Zhang et al. 2024) was the first study to apply LoRA in a federated context. (Bai et al. 2024; Cho et al. 2024) focused on federated fine-tuning where clients have different LoRA ranks, while (Wang et al. 2024c) proposed strategies for LoRA initialization to achieve better performance. Another studies (Bian et al. 2025b; Sun et al. 2024) addressed server aggregation bias in LoRA-based federated fine-tuning.

The most relevant works to ours focus on personalized federated fine-tuning with LoRA (Yang et al. 2024a; Qi et al. 2024). FedDPA (Yang et al. 2024a) introduced global and local LoRA components but sufferd from interference during global LoRA training and struggled to balance the two phases effectively. PF2LoRA (Hao et al. 2025) employed a bilevel framework to combine a shared adapter with client-specific adapters but faced interference issues during the aggregation of the shared adapter. FDLoRA (Qi et al. 2024) utilized personalized LoRA modules to initialize federated training, combining them with a global LoRA module via adaptive fusion; however, it relied on server-side datasets and was prone to interference during global LoRA aggregation, limiting its effectiveness in heterogeneous settings.

Preliminaries and Motivations

Low-Rank Adaptation (LoRA)

Consider a pre-trained model with parameters $\mathbf{W}_0 \in \mathbb{R}^{l\times d}$, where $\mathbf{W}_0$ represents the fixed parameters of the model, and $\mathbf{\Delta W} \in \mathbb{R}^{l\times d}$ denotes the trainable update matrix applied during fine-tuning. Here, d is the input dimension and l is the output dimension. Instead of updating all elements in $\mathbf{\Delta W}$, LoRA decomposes $\mathbf{\Delta W}$ into two low-rank matrices $\mathbf{A} \in \mathbb{R}^{r\times d}$ and $\mathbf{B} \in \mathbb{R}^{l\times r}$, where $r \ll \min(d, l)$. This decomposition allows the fine-tuning process to focus on the significantly smaller low-rank matrices $\mathbf{A}$ and $\mathbf{B}$ instead of the full matrix $\mathbf{\Delta W}$. Consequently, the total number of trainable parameters is reduced from $d \times l$ to $r \times (d+l)$. The model parameters after fine-tuning are given by:

$$\mathbf{W} = \mathbf{W}_0 + \mathbf{\Delta W} = \mathbf{W}_0 + \mathbf{BA}, \tag{1}$$

where $\mathbf{A}$ is typically initialized with random Gaussian values, while $\mathbf{B}$ is initialized to zero.

Heterogeneous Federated Fine-Tuning

Assume there are K institutions (clients), each possessing a distinct local training dataset $\mathcal{D}_k = \{X_k, Y_k\}$, where k indexes a client. In the context of LLM fine-tuning, it is common for clients to have datasets corresponding to entirely different tasks (e.g., one client may have data for text summarization, while another client has data for sentiment analysis), rather than simply exhibiting statistical differences as in conventional FL.

In this scenario, each client k aims to leverage FL to extract useful information while fine-tuning a model tailored to its specific task. The goal is for each client to learn an individualized fine-tuned model $\mathbf{W}_k$ that best fits its local dataset. The objective can be formulated as:

$$\min_{\mathcal{W}} \frac{1}{K} \sum_{k=1}^{K} L_k(X_k, Y_k, \mathbf{W}_k), \tag{2}$$

where $L_k(\cdot)$ represents the loss function for client k, and $\mathcal{W}$ denotes the set of personalized models $\{\mathbf{W}_k\}_{k=1}^{K}$.

Motivational Study

Federated fine-tuning operates in highly heterogeneous environments where client tasks can differ significantly. To understand the effects of such heterogeneity, we conducted a motivational study that highlights both the benefits and challenges of applying FL in these settings.

Fact 1: *FL Can Transfer Useful Knowledge but Also Introduces Harmful Interference.*

In FL settings, the data collected by each client can be highly diverse (Huang, Ye, and Du 2022; Mendieta et al. 2022; Wang et al. 2024a). We conducted a motivational experiment using the Flan dataset (Chung et al. 2024), selecting 8 tasks and creating 8 clients, each specializing in a single task. To demonstrate potential interference in federated fine-tuning, we compared two baseline methods. The first

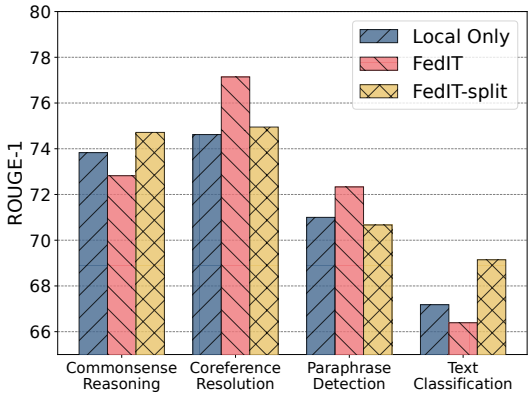


Figure 2: Motivational study results.

method, *FedIT* (Zhang et al. 2024), combines FedAvg with LoRA fine-tuning. The second method is *Local Only* LoRA fine-tuning, where each client fine-tunes its LoRA locally without any communication or information sharing between clients. The results, shown in fig. 2, support our first observation. On one hand, federated fine-tuning using FedIT achieves better performance than local fine-tuning on tasks Coreference Resolution and Paraphrase Detection, confirming that FL can enhance learning by enabling clients to benefit from shared knowledge. On the other hand, for tasks Commonsense Reasoning and Text Classification, FedIT underperforms compared to individual local fine-tuning. This dual outcome underscores both the promise and the pitfalls of federated fine-tuning in heterogeneous settings.

Fact 2: *FedAvg Paradigm Struggles to Mitigate Interference*

Having established the existence of harmful interference, we next explored potential mitigation strategies. In centralized LoRA fine-tuning, HydraLoRA (Tian et al. 2024) found that splitting a single large LoRA into multiple smaller LoRAs reduces cross-domain interference. However, FL introduces additional challenges, as data domains are distributed across clients. To test whether multiple smaller LoRAs could mitigate interference in FL, we implemented FedAvg with multiple parallel LoRAs (i.e. FedIT-split). However, as shown in fig. 2, its performance remained comparable to FedIT (FedAvg with a single large LoRA), indicating that interference persisted. This result is expected: in centralized settings, interference occurs within a shared dataset and can be mitigated by partitioning LoRAs. In FL, interference arises primarily during server aggregation, and simply introducing multiple LoRAs does not prevent the aggregated model from disrupting local adaptations. Consequently, direct adaptations of centralized interference-mitigation strategies fail to address the core issue in FL.

Proposed Method: FedALT

Our motivational studies reveal that harmful interference in FL arises from server aggregation and the use of the aggregated model to initialize subsequent training rounds. In the FedAvg paradigm, a global LoRA serves as a shared initialization point, allowing clients to fine-tune locally on their respective datasets. After fine-tuning, the updated LoRAs are averaged on the server and redistributed to clients as the new initialization. However, when client data differ significantly, aggregation disrupts individual adaptations, often negating improvements from local training and hindering personalization. Can cross-client interference be mitigated? The simplest and most direct solution is local fine-tuning using only a client's own dataset. This approach eliminates interference entirely but fails to leverage shared knowledge from other clients. To address this trade-off, we propose a novel personalized federated LoRA fine-tuning algorithm that diverges from the FedAvg paradigm, prioritizing local training while selectively integrating useful global information.

Our method introduces two key innovations: 1. Each client maintains two LoRA modules: an **Individual LoRA**, which captures locally learned information and is updated exclusively during the client's fine-tuning, and a **Rest-of-World (RoW) LoRA**, which aggregates information from all other clients. Crucially, the RoW LoRA remains fixed during local fine-tuning, ensuring that aggregation does not interfere with a client's adaptation. 2. An **adaptive mixer** dynamically learns the optimal **input-specific** weighting between the Individual LoRA and RoW LoRA components to balance personalization and global knowledge integration. In the following of this section, we detail the design and functionality of these components. The theoretical convergence analysis is provided in the supplementary material.

Individual and Rest-of-World (RoW) LoRA

Each client k maintains two types of LoRAs: an Individual LoRA, denoted by $\mathbf{A}_k^L/\mathbf{B}_k^L$, and a RoW LoRA, denoted by $\mathbf{A}_k^R/\mathbf{B}_k^R$. The Individual LoRA captures information obtained locally by the client and is isolated from the information of other clients. In contrast, the RoW LoRA represents the average of the Individual LoRAs from all other clients and remains frozen during the client's local fine-tuning. Specifically, at the end of each round, client k updates its Individual LoRA $\mathbf{A}_k^L/\mathbf{B}_k^L$ through local training. The RoW LoRA for client k is then computed as:

$$\mathbf{A}_k^R = \frac{1}{K-1}\sum_{m\neq k}\mathbf{A}_m^L, \quad \mathbf{B}_k^R = \frac{1}{K-1}\sum_{m\neq k}\mathbf{B}_m^L, \quad (3)$$

After clients upload their local LoRAs to the server, the server computes the RoW LoRA for each client individually and distributes it back to them. Alternatively, the server can calculate the global average of all clients' individual LoRAs and distribute this average back to the clients. Each client then determines its RoW LoRA by subtracting its individual LoRA from the global average LoRA.

It is crucial to emphasize that the RoW LoRA is not further updated on clients using their local datasets. Instead, it is only updated through averaging at the end of each learning round. Training the RoW LoRA on local datasets is unnecessary and potentially counterproductive, as subsequent averaging could cancel out any improvements. Furthermore, skipping local training for the RoW LoRA reduces the computational workload on clients by half.

Dynamic Mixture-of-Experts Mechanism

A natural question arises: why not simply add the RoW LoRA directly to the pre-trained model, i.e., $\mathbf{W}_0 \leftarrow \mathbf{W}_0 + \mathbf{B}_k^R\mathbf{A}_k^R$, since the RoW LoRA remains fixed during local training? This approach would allow each client to maintain only a single Individual LoRA, saving memory space. However, there are two key reasons why this is not desirable:

1. **Potential Contamination of the Pre-trained Model**: If the RoW LoRA underperforms, adding it directly to the pre-trained model risks contaminating it, making further corrections difficult.
2. **Loss of Flexibility**: Directly incorporating the RoW LoRA enforces a fixed weight, preventing dynamic adjustment between the Individual LoRA and shared information for different inputs. This lack of flexibility is problematic because different inputs may benefit variably from the local model versus the globally averaged model. A "one-size-fits-all" approach often fails to perform optimally across diverse inputs.

To address these challenges, we propose a dynamic weighting mechanism for combining the Individual LoRA and the RoW LoRA, enabling input-specific flexibility. Specifically, we introduce a mixer to dynamically adjust the contributions of the two LoRAs for each input. During the forward pass of each local epoch, the model's output is computed as:

$$y = \mathbf{W}_0 x + \alpha_k(x)\mathbf{B}_k^L\mathbf{A}_k^L x + \big(1 - \alpha_k(x)\big)\mathbf{B}_k^R\mathbf{A}_k^R x,$$

where $\alpha_k(x)$ determines for client k the contribution weight of the Individual LoRA, and $1 - \alpha_k(x)$ determines the contribution of the RoW LoRA.

Given the complexity of modern LLMs, where LoRA modules are inserted at multiple layers, learning $\alpha(x)$ is non-trivial. Inspired by (Jordan and Jacobs 1994), we adopt a Mixture-of-Experts (MoE) approach to compute $\alpha(x)$. Specifically, we introduce a mixer $\mathbf{G}_k \in \mathbb{R}^{2\times d}$ which is a dense layer with trainable weights (a transformation matrix), followed by a softmax function, to learn the weighting:

$$\alpha(x),\ 1 - \alpha(x) = \text{softmax}(\mathbf{G}_k x). \tag{4}$$

Importantly, the MoE mixers are personalized for each client, tailored to their specific domain tasks, and are not averaged across clients. This personalization ensures that the weight adjustments reflect each client's unique data distribution and learning objectives.

Algorithm Workflow

By combining the two LoRA modules (Individual LoRA and RoW LoRA) with dynamic weights, our proposed method achieves a balance between leveraging useful information from other clients and isolating harmful interference. To better illustrate the FedALT training process, we provide an overview of the server and client operations at round t.

Server Side. After local fine-tuning in round t, the server receives each client k's Individual LoRA modules ($\mathbf{A}_k^L/\mathbf{B}_k^L$). Upon receiving all clients' Individual LoRA modules, the server updates and computes the RoW LoRA modules for each client k based on eq. (3). After calculating the RoW LoRA modules for client k, the server broadcasts these updated RoW LoRA to the respective client.

Client Side. Each client k replaces its current RoW LoRA modules with the new RoW LoRA modules ($\mathbf{A}_k^R/\mathbf{B}_k^R$) received from the server. It then begins local fine-tuning for round $t+1$ using its local dataset. During the local fine-tuning process, the client updates the parameters in its Individual LoRA modules ($\mathbf{A}_k^L/\mathbf{B}_k^L$) and the mixer $\mathbf{G}_k$, while the RoW LoRA modules ($\mathbf{A}_k^R/\mathbf{B}_k^R$) and the pre-trained model ($\mathbf{W}_0$) remain fixed. Once the local fine-tuning is complete, the updated Individual LoRA is uploaded to the server, while $\mathbf{G}_k$ stays local to the client.

Experiments

Datasets. Our training datasets are derived from Flan (Chung et al. 2024), a comprehensive benchmark comprising over 60 NLP datasets across more than 12 diverse task types. To simulate client heterogeneity in the main experiments, we assume 8 clients, each assigned a distinct NLP task dataset. Detailed descriptions, including task distributions and client assignments, are provided in supplementary material.

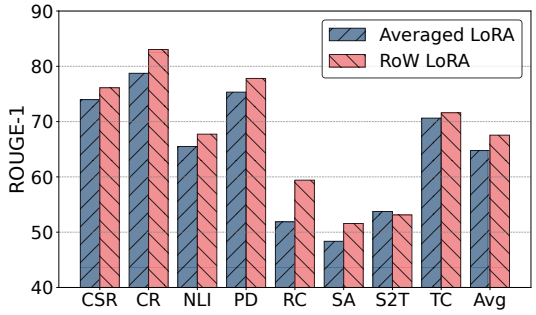
Pre-trained Model and Baselines. We utilize LLaMA2-7B (Touvron et al. 2023b) and Bloom-560M (Le Scao et al. 2023) as the pre-trained models for all baseline and proposed fine-tuning methods. To evaluate the performance of our proposed method, we compare it with the state-of-the-art methods in federated fine-tuning with LoRA, including **FedIT** (Zhang et al. 2024), **FFA-LoRA** (Sun et al. 2024). Additionally, we compare it with state-of-the-art methods focused on personalized federated fine-tuning with LoRA, such as **FedDPA** (Yang et al. 2024a), **PF2LoRA** (Hao et al. 2025), **FedSA** (Guo et al. 2024), and **FDLoRA** (Qi et al. 2024). We also include the **Local Only** baseline, where clients fine-tune their models independently without federated collaboration. Detailed descriptions of these baseline methods are provided in the supplementary material.

Performance Comparisons

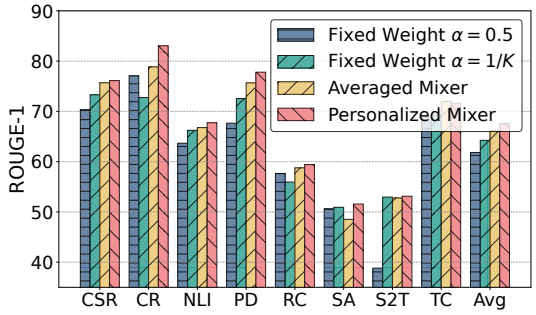
We compare the performance of FedALT, with other baseline methods based on each client's performance on its specific test dataset, aligned with its training dataset. As shown in table 1, which reports the results on LLaMA2-7B pre-trained models, our proposed method achieves the best overall performance compared to all baselines, with the average score exceeding that of the best-performing baseline, FDLoRA, by 2.38. FedALT consistently delivers superior performance across most tasks, outperforming the baselines by at least 1.61 on the Natural Language Inference task and 2.02 on the Reading Comprehension task. We find that on the Struct-to-Text, the Local Only method achieves the highest performance. This is likely due to the unique nature of this task, making it difficult to benefit from shared global knowledge. Furthermore, we observe that FedIT underperforms compared to Local Only on four client tasks, indicating the presence of harmful interference during aggregation. Personalized federated LoRA fine-tuning methods also fall short compared to FedALT, likely because they fail to ef-

Methods	Commonsense Reasoning	Coreference Resolution	Natural Language Inference	Paraphrase Detection	Reading Comprehension	Sentiment Analysis	Struct to Text	Text Classification	Average
LLaMA2-7B									
Local	73.83	74.62	58.73	71.00	55.82	46.56	**55.14**	67.18	62.86
FedIT	72.82	77.14	59.53	72.33	53.35	46.76	49.23	66.39	62.19
FFA-LoRA	66.96	68.48	50.41	69.79	49.69	44.44	43.29	61.15	56.78
FedSA	72.68	78.24	64.46	76.33	54.32	42.71	53.57	65.47	63.47
FedDPA	74.81	81.88	62.92	76.33	55.91	47.86	52.02	65.42	64.64
PF2LoRA	74.13	77.55	64.17	**78.33**	55.36	48.65	53.44	63.90	64.44
FDLoRA	**76.29**	75.60	66.12	75.67	57.39	49.85	52.85	67.59	65.17
FedALT	76.12	**83.04**	**67.73**	77.80	**59.41**	**51.57**	53.12	**71.60**	**67.55**
Bloom-560M									
Local	52.13	39.20	37.55	62.67	29.45	42.91	**41.33**	49.99	44.40
FedIT	50.24	40.08	38.51	67.00	27.98	40.34	38.95	51.52	44.33
FFA-LoRA	46.07	35.05	33.57	66.12	25.52	37.13	35.43	44.63	40.44
FedSA	55.79	41.65	41.12	69.33	27.90	**43.72**	41.27	55.03	46.98
FedDPA	55.43	**41.88**	41.80	69.80	29.74	41.89	39.54	55.85	46.99
PF2LoRA	54.58	40.16	37.40	70.08	25.78	42.36	37.90	58.67	45.87
FDLoRA	53.65	40.22	37.27	68.89	29.95	42.89	35.39	56.05	45.54
FedALT	**56.39**	41.27	**44.45**	**70.67**	**30.62**	43.04	39.24	**59.03**	**48.09**

Table 1: Performance comparison with baseline methods across different models.



(a) Impact of RoW LoRA. Isolating client-specific information is more effective than using globally averaged updates.



(b) Impact of dynamic weighting via MOE. The personalized mixer outperforms fixed or shared weighting.

Figure 3: Ablation studies of FedALT.

fectively isolate harmful interference during the server aggregation step. Despite their designs for local/global LoRA interaction, these methods update LoRA modules using aggregated global information, which may carry conflicting information from other clients.

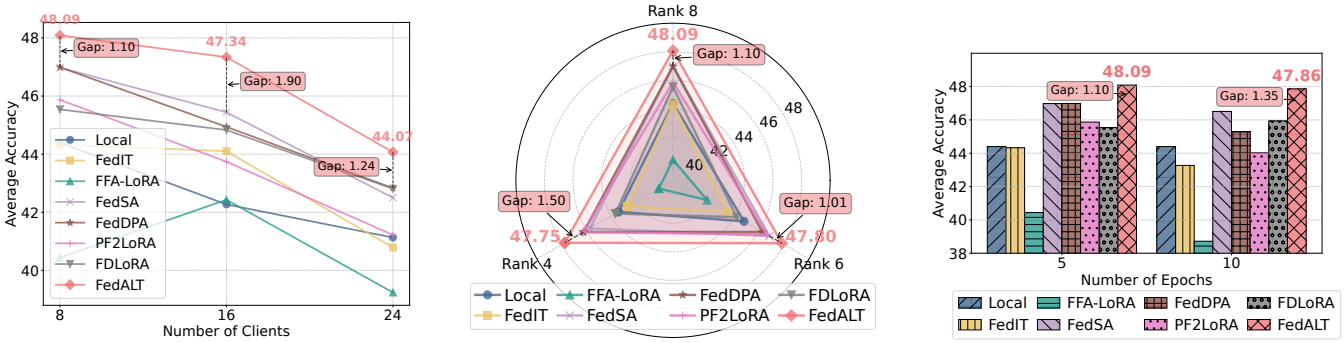
To further evaluate the robustness of our proposed FedALT, we conduct experiments using the Bloom-560M model. As shown in table 1, FedALT consistently outperforms all baseline methods across the NLP tasks. These empirical results validate that conventional FedAvg-based approaches suffer from harmful cross-client parameter interference during aggregation. In contrast, FedALT's adaptive strategy, which balances local specialization and global knowledge, effectively alleviates this issue. The consistent performance gains observed with both LLaMA2-7B and Bloom-560M confirm the generalizability of our approach across diverse model families and parameter scales.

Ablation Study

Impact of Decoupling LoRA Training. In our proposed method, only the Individual LoRA modules are updated locally during client fine-tuning, while the RoW LoRA remains frozen and is updated exclusively through server-side aggregation. This design mitigates harmful cross-client interference by preventing local updates to the shared global component. To evaluate the effectiveness of this decoupled training strategy, we compare FedALT with two alternative methods. The first baseline, *RoW-Update (rank = 4)*, fine-tunes both the Individual and RoW LoRA modules locally, each with a rank of 4 to maintain the same total number of trainable parameters. The second baseline, *RoW-Update (rank = 8)*, also fine-tunes both LoRA modules locally, but assigns a rank of 8 to each, resulting in a higher overall parameter count. For fair comparison, both baselines use the dynamic mixer. The key distinction lies in whether the RoW LoRA is updated locally. As shown in table 2, while the second method outperforms the first due to its increased capacity, both alternatives underperform to FedALT. Notably, even with more trainable parameters, the second method fails to match FedALT's performance, highlighting the benefits of decoupling local updates from the global component.

Impact of RoW LoRA To evaluate the impact of using the RoW LoRA instead of the global average of all Local LoRAs (which includes each client's Individual LoRA), we conduct an experiment. As shown in fig. 3a, leveraging the RoW LoRA significantly improves performance compared to using the global average. This result underscores the importance of isolating client-specific information from the aggregated global knowledge.

Impact of Dynamic Weighting via MOE. To enable



(a) Effect of number of clients. (b) Effect of LoRA rank. (c) Effect of local epochs.

Figure 4: Sensitivity Analyses of FedALT under different training configurations.

Methods	Same Trainable	Same Inference	Avg. Perf.
RoW-Update (rank = 4)	✓	×	63.59
RoW-Update (rank = 8)	×	✓	65.75
FedALT	–	–	**67.55**

Table 2: Impact of decoupling LoRA training.

input-specific flexibility, our proposed method adopts a MOE mechanism to dynamically adjust the contributions of the Individual LoRA and RoW LoRA for each data sample. We conduct experiments to evaluate the efficiency of this dynamic weighting by comparing it to two fixed weighting strategies. In the first fixed strategy, we set the contribution weight to a fixed value of $\alpha = 0.5$, meaning that the Individual LoRA and RoW LoRA contribute equally. In the second fixed strategy, we set the contribution weight to $\alpha = 1/K$, where K is the number of clients. The results in fig. 3b validate that our proposed dynamic weighting mechanism outperforms both fixed strategies, demonstrating its ability to enhance performance. Additionally, our dynamic weighting is implemented through the personal $\mathbf{G}_k$ mixer, which is maintained locally by each client and is not aggregated during the training process. To further investigate this design choice, we conducted experiments where $\mathbf{G}_k$ is averaged across clients. The results confirm that retaining a personalized $\mathbf{G}_k$ for each client yields optimal performance, emphasizing the importance of input-specific and client-specific adjustments.

Sensitivity Analysis

Effect of Number of Clients. In our main experiments, we demonstrated the effectiveness of the proposed method using 8 clients. Here we extend our evaluation by conducting additional experiments with 16 and 24 clients, using BLOOM as the foundation model. For the 16-client setting, we distributed each original dataset from the 8-client configuration between two clients, resulting in each client having approximately half the data samples compared to the 8-client scenario, while maintaining the same 8 distinct task types across the federation. The results in fig. 4a show that FedALT consistently maintains its performance advantage even as the number of clients increases, outperforming the strongest baseline by 1.90 at least. Note that FedALT remains efficient due to its architectural design: each client maintains only its Individual LoRA and a single RoW LoRA, rather than storing separate components for every other client. Additionally, the server-side computation of the RoW LoRA remains lightweight, as it involves only simple averaging operations.

Effect of LoRA Rank. We evaluate the impact of different LoRA ranks $r \in \{4, 6, 8\}$ using BLOOM as the pretrained model, keeping all other settings consistent with the main experiment. As shown in fig. 4b, the proposed method consistently outperforms baseline methods across all rank values, demonstrating its adaptability and robustness.

Effect of Local Epochs. We investigate the effect of varying local training epochs (5 and 10), while keeping all other configurations unchanged. As presented in fig. 4c, our method outperforms all baselines, further highlighting its robustness to different local training epochs.

Additional experiments on the **effect of data heterogeneity** can be found in the supplementary material.

Conclusion

In this work, we introduced FedALT, a personalized federated LoRA fine-tuning framework that departs from conventional FedAvg paradigms. Our key innovations include a decoupled training scheme with Individual and frozen Rest-of-World (RoW) LoRA components that mitigates harmful cross-client interference in heterogeneous settings. Additionally, we introduce an adaptive mixer inspired by Mixture-of-Experts that dynamically balances local and global information on an input-specific basis. Through comprehensive evaluation across diverse NLP tasks, FedALT demonstrates superior performance compared to state-of-the-art baselines while maintaining computational efficiency. Future research directions include investigating clustering-based approaches for maintaining multiple RoW LoRA components, and extending our framework to multi-modal foundation models.

Acknowledgments

The work of Jieming Bian, Lei Wang and Jie Xu is partially supported by NSF under grants 2433886, 2505381 and 2515982. The work of Letian Zhang is partially supported by NSF under grant 2348279 and also supported by MTSU Stark Land project.

References

Bai, J.; Chen, D.; Qian, B.; Yao, L.; and Li, Y. 2024. Federated fine-tuning of large language models under heterogeneous language tasks and client resources. *arXiv e-prints*, arXiv–2402.

Bian, J.; Peng, Y.; Wang, L.; Huang, Y.; and Xu, J. 2025a. A survey on parameter-efficient fine-tuning for foundation models in federated learning. *arXiv preprint arXiv:2504.21099*.

Bian, J.; Wang, L.; Yang, K.; Shen, C.; and Xu, J. 2024. Accelerating hybrid federated learning convergence under partial participation. *IEEE Transactions on Signal Processing*.

Bian, J.; Wang, L.; Zhang, L.; and Xu, J. 2025b. LoRA-FAIR: Federated LoRA Fine-Tuning with Aggregation and Initialization Refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 3737–3746.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Cho, Y. J.; Liu, L.; Xu, Z.; Fahrezi, A.; and Joshi, G. 2024. Heterogeneous lora for federated fine-tuning of on-device foundation models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 12903–12913.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70): 1–53.

Collins, L.; Hassani, H.; Mokhtari, A.; and Shakkottai, S. 2021. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, 2089–2099. PMLR.

Deng, Y.; Kamani, M. M.; and Mahdavi, M. 2020. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*.

Ding, N.; Qin, Y.; Yang, G.; Wei, F.; Yang, Z.; Su, Y.; Hu, S.; Chen, Y.; Chan, C.-M.; Chen, W.; et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3): 220–235.

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*.

Fu, Z.; Yang, H.; So, A. M.-C.; Lam, W.; Bing, L.; and Collier, N. 2023. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 12799–12807.

Guo, P.; Zeng, S.; Wang, Y.; Fan, H.; Wang, F.; and Qu, L. 2024. Selective Aggregation for Low-Rank Adaptation in Federated Learning. *arXiv preprint arXiv:2410.01463*.

Hadi, M. U.; Qureshi, R.; Shah, A.; Irfan, M.; Zafar, A.; Shaikh, M. B.; Akhtar, N.; Wu, J.; Mirjalili, S.; et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

Han, Z.; Gao, C.; Liu, J.; Zhang, J.; and Zhang, S. Q. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Hao, J.; Wu, Y.; Payani, A.; Lee, M.; and Liu, M. 2025. Personalized Federated Fine-tuning for Heterogeneous Data: An Automatic Rank Learning Approach via Two-Level LoRA. *arXiv preprint arXiv:2503.03920*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Huang, W.; Ye, M.; and Du, B. 2022. Learn from others and be yourself in heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10143–10153.

Jordan, M. I.; and Jacobs, R. A. 1994. Hierarchical mixtures of experts and the EM algorithm. *Neural computation*, 6(2): 181–214.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1. Minneapolis, Minnesota.

Kuang, W.; Qian, B.; Li, Z.; Chen, D.; Gao, D.; Pan, X.; Xie, Y.; Li, Y.; Ding, B.; and Zhou, J. 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*, 5260–5271.

Le Scao, T.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A. S.; Yvon, F.; Gallé, M.; et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Lermen, S.; Rogers-Smith, C.; and Ladish, J. 2023. Lora fine-tuning efficiently undoes safety training in llama 2-chat 70b. *arXiv preprint arXiv:2310.20624*.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Li, X. L.; and Liang, P. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Liu, H.; Tam, D.; Muqeeth, M.; Mohta, J.; Huang, T.; Bansal, M.; and Raffel, C. A. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35: 1950–1965.

Liu, J.; Liu, Y.; Shang, F.; Liu, H.; Liu, J.; and Feng, W. 2025a. Improving Generalization in Federated Learning with Highly Heterogeneous Data via Momentum-Based Stochastic Controlled Weight Averaging. In *Forty-second International Conference on Machine Learning*.

Liu, J.; Shang, F.; Liu, Y.; Liu, H.; Li, Y.; and Gong, Y. 2024. Fedbcgd: Communication-efficient accelerated block coordinate gradient descent for federated learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2955–2963.

Liu, J.; Shang, F.; Tian, Y.; Liu, H.; and Liu, Y. 2025b. Consistency of Local and Global Flatness for Federated Learning. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, 3875–3883. New York, NY, USA: Association for Computing Machinery. ISBN 9798400720352.

Liu, J.; Shang, F.; Zhu, K.; Liu, H.; Liu, Y.; and Liu, J. 2025c. FedAdamW: A Communication-Efficient Optimizer with Convergence and Generalization Guarantees for Federated Large Models. arXiv:2510.27486.

Liu, Q.; Wu, X.; Zhao, X.; Zhu, Y.; Xu, D.; Tian, F.; and Zheng, Y. 2023. Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications. *arXiv preprint arXiv:2310.18339*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Mendieta, M.; Yang, T.; Wang, P.; Lee, M.; Ding, Z.; and Chen, C. 2022. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8397–8406.

Minaee, S.; Mikolov, T.; Nikzad, N.; Chenaghlu, M.; Socher, R.; Amatriain, X.; and Gao, J. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Qi, J.; Luan, Z.; Huang, S.; Fung, C.; Yang, H.; and Qian, D. 2024. FDLoRA: Personalized Federated Learning of Large Language Model via Dual LoRA Tuning. *arXiv preprint arXiv:2406.07925*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140): 1–67.

Sheng, Y.; Cao, S.; Li, D.; Hooper, C.; Lee, N.; Yang, S.; Chou, C.; Zhu, B.; Zheng, L.; Keutzer, K.; et al. 2023. S-lora: Serving thousands of concurrent lora adapters. *arXiv preprint arXiv:2311.03285*.

Sun, G.; Khalid, U.; Mendieta, M.; Yang, T.; Wang, P.; Lee, M.; and Chen, C. 2022. Conquering the communication constraints to enable large pre-trained models in federated learning. *arXiv preprint arXiv:2210.01708*.

Sun, Y.; Li, Z.; Li, Y.; and Ding, B. 2024. Improving loRA in privacy-preserving federated learning. *arXiv preprint arXiv:2403.12313*.

Tan, A. Z.; Yu, H.; Cui, L.; and Yang, Q. 2022. Towards personalized federated learning. *IEEE transactions on neural networks and learning systems*, 34(12): 9587–9603.

Tian, C.; Shi, Z.; Guo, Z.; Li, L.; and Xu, C. 2024. HydraLoRA: An Asymmetric LoRA Architecture for Efficient Fine-Tuning. *arXiv preprint arXiv:2404.19245*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wang, L.; Bian, J.; Zhang, L.; Chen, C.; and Xu, J. 2024a. Taming Cross-Domain Representation Variance in Federated Prototype Learning with Heterogeneous Data Domains. *arXiv preprint arXiv:2403.09048*.

Wang, Y.; Fu, H.; Kanagavelu, R.; Wei, Q.; Liu, Y.; and Goh, R. S. M. 2024b. An aggregation-free federated learning for tackling data heterogeneity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26233–26242.

Wang, Y.; Lin, Y.; Zeng, X.; and Zhang, G. 2023. Multilora: Democratizing lora for better multi-task learning. *arXiv preprint arXiv:2311.11501*.

Wang, Z.; Shen, Z.; He, Y.; Sun, G.; Wang, H.; Lyu, L.; and Li, A. 2024c. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. *arXiv preprint arXiv:2409.05976*.

Wu, F.; Li, Z.; Li, Y.; Ding, B.; and Gao, J. 2024. Fedbiot: Llm local fine-tuning in federated learning without full model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3345–3355.

Yang, Y.; Long, G.; Shen, T.; Jiang, J.; and Blumenstein, M. 2024a. Dual-Personalizing Adapter for Federated Foundation Models. *arXiv preprint arXiv:2403.19211*.

Yang, Z.; Zhang, Y.; Zheng, Y.; Tian, X.; Peng, H.; Liu, T.; and Han, B. 2024b. FedFed: Feature distillation against data heterogeneity in federated learning. *Advances in Neural Information Processing Systems*, 36.

Yao, L.; Gao, D.; Wang, Z.; Xie, Y.; Kuang, W.; Chen, D.; Wang, H.; Dong, C.; Ding, B.; and Li, Y. 2022. A benchmark for federated hetero-task learning. *arXiv preprint arXiv:2206.03436*.

Zeng, A.; Liu, X.; Du, Z.; Wang, Z.; Lai, H.; Ding, M.; Yang, Z.; Xu, Y.; Zheng, W.; Xia, X.; et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Zhang, J.; Vahidian, S.; Kuo, M.; Li, C.; Zhang, R.; Yu, T.; Wang, G.; and Chen, Y. 2024. Towards building the federatedGPT: Federated instruction tuning. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6915–6919. IEEE.

Zhang, L.; Chen, B.; Bian, J.; Wang, L.; and Xu, J. 2025. FedEL: Federated Elastic Learning for Heterogeneous Devices. *arXiv preprint arXiv:2509.16902*.

Zhou, C.; Li, Q.; Li, C.; Yu, J.; Liu, Y.; Wang, G.; Zhang, K.; Ji, C.; Yan, Q.; He, L.; et al. 2024. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *International Journal of Machine Learning and Cybernetics*, 1–65.