FLoRG: Federated Fine-tuning with Low-rank Gram Matrices and Procrustes Alignment

Anonymous authors
Paper under double-blind review

Abstract

Parameter-efficient fine-tuning techniques such as Low-rank Adaptation (LoRA) enable large language models (LLMs) to adapt to downstream tasks efficiently. Federated learning (FL) further facilitates this process by enabling collaborative fine-tuning across distributed clients without sharing private data. However, the use of two separate low-rank matrices in LoRA for federated fine-tuning introduces two types of challenges. The first challenge arises from the error induced by separately aggregating those two low-rank matrices. The second challenge occurs even when the product of two low-rank matrices is aggregated. The server needs to recover factors via matrix decomposition, which is non-unique and can introduce decomposition drift. To tackle the aforementioned challenges, we propose FLoRG, a federated fine-tuning framework which employs a single low-rank matrix for fine-tuning and aggregates its Gram matrix (i.e., the matrix of inner products of its column vectors), eliminating the aggregation error while also reducing the communication overhead. FLoRG minimizes the decomposition drift by introducing a Procrustes alignment approach which aligns the decomposed matrix between consecutive fine-tuning rounds for consistent updates. We theoretically analyze the convergence of FLoRG and prove that adopting the Procrustes alignment results in a tighter convergence bound. Experimental results across multiple LLM fine-tuning benchmarks demonstrate that FLoRG outperforms four state-of-the-art baseline schemes in the downstream task accuracy and can reduce the communication overhead by up to 82%.

1 Introduction

Large language models (LLMs) (Zhao et al., 2023; Achiam et al., 2023; Touvron et al., 2023; Grattafiori et al., 2024) have achieved state-of-the-art performance across a wide range of natural language processing tasks. However, their massive scale introduces critical challenges in terms of computation cost, memory consumption, and adaptability to downstream tasks. To address these concerns, low-rank adaptation (LoRA) (Hu et al., 2022; Hayou et al., 2024; Kopiczko et al., 2024; Liu et al., 2024) has emerged as an effective approach. In particular, the LoRA module employs a fine-tuning matrix $\Delta\mathbf{W}$ with two low-rank matrices $\mathbf{B}$ and $\mathbf{A}$ into the pretrained model $\mathbf{W}^0$ as $\mathbf{W} = \mathbf{W}^0 + \Delta\mathbf{W} = \mathbf{W}^0 + \mathbf{BA}$. Thus, it enables task-specific adaptation while only updating $\Delta\mathbf{W}$. This approach reduces both memory usage and computation overhead compared to full-model fine-tuning significantly. However, fine-tuning still favors domain-specific data at scale. Such data is typically distributed across multiple clients and therefore requires collaborative fine-tuning. To resolve this issue, federated learning (FL) (McMahan et al., 2017; Li et al., 2020) provides a privacy-preserving framework for collaborative model training, where multiple clients fine-tune a shared global model without exposing their raw data. Combining LoRA with FL is therefore highly appealing: clients can collaboratively fine-tune a model by locally performing lightweight training via LoRA modules and uploading the low-rank updates for global aggregation.

The conventional works (Zhang et al., 2024; Fang et al., 2024; Zhang et al., 2023b; Wu et al., 2024) propose federated fine-tuning with LoRA which enables each client n to transmit its low rank matrices $\mathbf{B}_n$ and $\mathbf{A}_n$ to the central server. Afterwards, the central server aggregates $\mathbf{B}_n$ and $\mathbf{A}_n$ separately and then broadcasts the two aggregated matrices back to each client for performing fine-tuning in the subsequent rounds. In this case, the updated LoRA module after the model aggregation can be expressed as $(\frac{1}{N}\sum_n \mathbf{B}_n) \times (\frac{1}{N}\sum_n \mathbf{A}_n)$. This approach introduces a challenge: *The aggregation*

is fundamentally biased, because the true update should be $\frac{1}{N}\sum_n(\mathbf{B}_n\mathbf{A}_n)$, which is different from $(\frac{1}{N}\sum_n \mathbf{B}_n) \times (\frac{1}{N}\sum_n \mathbf{A}_n)$. This mismatch introduces a systematic aggregation error which affects the global model update in every fine-tuning round. As the number of rounds increases, the aggregation error is stacked, which will degrade the fine-tuning performance.

To alleviate the error induced by model aggregation, some works (Yan et al., 2024; Bai et al., 2024; Yan et al., 2025) calculate the product of $\mathbf{B}_n\mathbf{A}_n$ and perform aggregation at the central server. Then, the central server performs decomposition to this aggregated matrix to recover two low-rank matrices for the next fine-tuning round. While this approach avoids the error induced by separately aggregating matrices $\mathbf{B}_n$ and $\mathbf{A}_n$, it introduces another non-trivial challenge: *decomposition is generally non-unique.* In particular, the rank of the LoRA module is typically much smaller than the dimension of the input or output of the parameter matrix. This rank deficiency can lead to multiple decompositions. In addition, when the aggregated matrix has eigenvalue multiplicities, many valid decompositions exist. As a result, choosing different matrix decompositions fundamentally changes two low-rank matrices. It incurs a drift in the parameter subspace and changes the direction of the model update for the subsequent fine-tuning round. This drift will stack as the fine-tuning proceeds and degrade the fine-tuning performance. Furthermore, direct decomposition (e.g., eigendecomposition) may incur rank mismatch since the rank of the aggregated matrix may be different from the local low-rank matrices.

Based on the aforementioned discussions, we focus on addressing the following question: *Is there a federated fine-tuning approach which eliminates the error induced by separate model aggregation while minimizing the drift induced by matrix decomposition?*

To address this question, we start by rethinking what to aggregate in federated fine-tuning. In particular, as LoRA involves matrix multiplication, either separate model aggregation or matrix decomposition is unavoidable. Therefore, one of our key insights is to reparameterize the LoRA module with a single low-rank matrix. FLoRG aggregates the corresponding Gram matrices to achieve an unbiased aggregation with a low communication overhead. Furthermore, another insight is to propose a Procrustes alignment approach to the decomposed matrix to stabilize the fine-tuning while preserving its Gram matrix, thereby mitigating drift caused by the non-uniqueness of matrix decomposition.

Designing such a framework is challenging due to the following unexplored questions: (i) *How can we design a low-rank parameterization which adapts to any pretrained matrix shape while supporting error-free aggregation?* (iii) *How to optimize the Procrustes alignment matrix to minimize the decomposition drift?* (ii) *How to characterize the overall convergence rate of FLoRG under nonconvex losses, and disentangle the impact of Procrustes alignment on the convergence?* In this work, we make the following contributions to address the aforementioned questions:

- We propose FLoRG, which replaces two low-rank matrices in LoRA with a single low-rank matrix. By leveraging a shared semi-orthogonal basis, FLoRG adapts to parameter matrices with arbitrary shapes. Each client only updates the single low-rank matrix, and the server aggregates the corresponding Gram matrix. FLoRG eliminates the aggregation error by turning the bilinear server-side aggregation into a linear operation. By transmitting a single matrix instead of two matrices, FLoRG is communication-efficient when compared with federated LoRA schemes.
- We propose a Procrustes alignment approach after matrix decomposition to preserve the aggregated Gram matrix while aligning the decomposed matrix across rounds to mitigate the decomposition drift and address the rank mismatch issue. In particular, we solve an optimization problem to minimize the inter-round decomposition drift via a Frobenius-norm objective. The closed-form optimal solution projects the decomposed matrix onto a target r-rank subspace without changing its Gram matrix.
- We theoretically analyze the convergence rate of our proposed FLoRG in the nonconvex loss setting. In particular, incorporating our proposed Procrustes alignment zeros out the Procrustes alignment drift term, thereby resulting in a tighter convergence bound.
- We conduct extensive experiments on GLUE (Wang et al., 2018) with MRPC, QQP, MNLI, QNLI, WNLI, and RTE datasets. We compare our proposed FLoRG with four state-of-the-art baseline schemes, including FedIT (Zhang et al., 2024), FeDeRA (Yan et al., 2024), FFA-LoRA (Sun et al., 2024), and FedSA-LoRA (Guo et al., 2025). Results show that

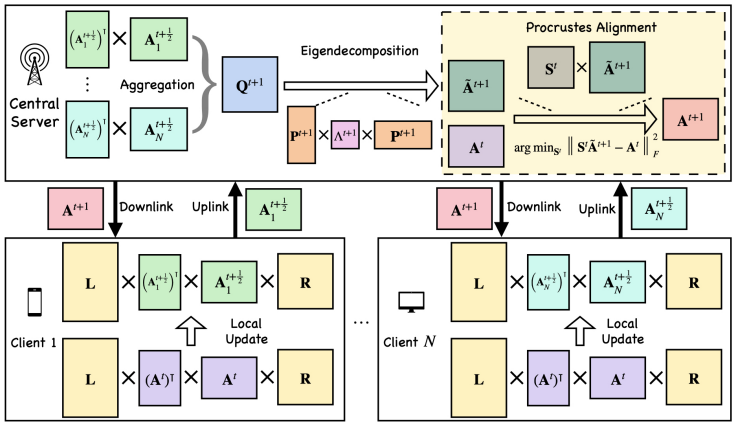

Figure 1: System model of our proposed FLoRG.

our proposed FLoRG achieves a higher testing accuracy than the baseline schemes under different settings and reduces the communication overhead by up to 82%.

2 Related Works

LoRA: Low-rank adaptation (LoRA) (Zhang et al., 2024) introduces two low-rank matrices to the pretrained model to perform parameter-efficient fine-tuning. Multiple LoRA variants have been proposed (Zhang et al., 2023a; Dettmers et al., 2023; Hayou et al., 2024; Kopiczko et al., 2024; Liu et al., 2024; Zhao et al., 2024; Bensaïd et al., 2025). For example, in (Zhang et al., 2023a), the authors proposed a LoRA framework to adaptively allocate the parameter budget among weight matrices based on the importance score. In (Zhao et al., 2024), the authors projected the gradient matrix into a low-rank form to perform efficient fine-tuning. In (Bensaïd et al., 2025), the authors reformulated the low-rank adaptation with a single matrix. While the aforementioned works have shown an improvement in the performance, they are primarily designed for centralized settings. Domain-specific data are often possessed by a number of distributed clients, which motivates the incorporation of FL.

Federated Fine-tuning with LoRA: LoRA has been incorporated into FL to enable collaborative fine-tuning across distributed clients. The conventional federated fine-tuning works (Zhang et al., 2024; Fang et al., 2024; Zhang et al., 2023b; Wu et al., 2024; Long et al., 2024; Cho et al., 2024; Byun & Lee, 2025) directly aggregate two low-rank matrices separately to obtain the global model. Some works (Babakniya et al., 2023; Yan et al., 2024; Bai et al., 2024; Yan et al., 2025) aggregate the LoRA modules (i.e., the products of two matrices) and then perform matrix decomposition to recover two low-rank matrices. In addition, the authors in (Wang et al., 2024) proposed a stacking-based approach to aggregate the low-rank matrices. The authors in (Sun et al., 2024) proposed to freeze matrix $\mathbf{A}$ and only update matrix $\mathbf{B}$. The authors in (Guo et al., 2025) proposed to let the clients locally update matrix $\mathbf{B}$ and only share matrix $\mathbf{A}$ for aggregation.

3 Methodology

3.1 FLoRG

In this section, we propose FLoRG, a federated fine-tuning framework. FLoRG employs a single low-rank matrix and aggregates Gram matrices to eliminate the error induced by the separate aggregation as in conventional LoRA. FLoRG performs Procrustes alignment to the decomposed matrix to minimize the decomposition drift. A schematic illustration is shown in Fig. 1.

We consider a central server and N clients. Let $\mathcal{T} = \{1, 2, \ldots, T\}$ and $\mathcal{N} = \{1, 2, \ldots, N\}$ denote the set of T fine-tuning rounds and the set of N clients, respectively. At the beginning of the first fine-tuning round, each client $n \in \mathcal{N}$ has the same pretrained weight matrix $\mathbf{W}^0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$. Note that

$\mathbf{W}^0$ is kept frozen and will not be updated during fine-tuning. We consider that each client $n \in \mathcal{N}$ has a local dataset $\mathcal{D}_n$. Let $\boldsymbol{\xi}_n \sim \mathcal{D}_n$ denote a mini-batch of training samples. Let $F_n(\mathbf{W}^t; \boldsymbol{\xi}_n)$ denote the local loss function of client $n \in \mathcal{N}$ on $\boldsymbol{\xi}_n$ with model $\mathbf{W}^t$ in the t-th fine-tuning round. We denote the expected loss of client n with model $\mathbf{W}^t$ as $F_n(\mathbf{W}^t) = \mathbb{E}_{\boldsymbol{\xi}_n \sim \mathcal{D}_n}[F_n(\mathbf{W}^t; \boldsymbol{\xi}_n)]$. Let $\nabla F_n(\mathbf{W}^t; \boldsymbol{\xi}_n)$ denote the local stochastic gradient of $\mathbf{W}^t$ of client n in the t-th fine-tuning round. The learning procedure of FLoRG can be summarized as the following steps.

(i) Local Update with Low-rank Gram Matrices: Different from LoRA, FLoRG uses a single matrix $\mathbf{A} \in \mathbb{R}^{r \times k}$ for fine-tuning, where $r \ll \min\{d_{\text{in}}, d_{\text{out}}\}$ denotes the rank of $\mathbf{A}$. Let $k = \min\{d_{\text{in}}, d_{\text{out}}\}$. Matrices $\mathbf{L} \in \mathbb{R}^{d_{\text{out}} \times k}$ and $\mathbf{R} \in \mathbb{R}^{k \times d_{\text{in}}}$ are initialized and shared across all clients. Both matrices $\mathbf{L}$ and $\mathbf{R}$ are semi-orthogonal, i.e., $\mathbf{L}^\intercal \mathbf{L} = \mathbf{I}_k$ and $\mathbf{R}\mathbf{R}^\intercal = \mathbf{I}_k$. Note that matrices $\mathbf{L}$ and $\mathbf{R}$ remain unchanged during fine-tuning. The fine-tuning matrix in the t-th fine-tuning round is given by

$$\Delta \mathbf{W}^t = \mathbf{L}\mathbf{Q}^t\mathbf{R} = \mathbf{L}(\mathbf{A}^t)^\intercal \mathbf{A}^t \mathbf{R}, \quad t \in \mathcal{T}, \tag{1}$$

where $\mathbf{Q}^t$ denotes the square Gram parameter matrix in the t-th fine-tuning round. By leveraging $\mathbf{L}$ and $\mathbf{R}$, FLoRG is compatible with parameter matrices of any dimensions. Thus, the full model can be expressed as $\mathbf{W}^t = \mathbf{W}^0 + \Delta \mathbf{W}^t$.

In the t-th fine-tuning round, the central server broadcasts $\mathbf{A}^t$ to all clients. Client $n \in \mathcal{N}$ updates $\mathbf{A}^t$ using its local dataset. Let $\nabla_{\mathbf{A}} F_n(\mathbf{W}^t; \boldsymbol{\xi}_n) \in \mathbb{R}^{r \times k}$ denote the gradient of the low-rank matrix, which is given by

$$\nabla_{\mathbf{A}} F_n(\mathbf{W}^t; \boldsymbol{\xi}_n) = \mathbf{A}^t \left(\mathbf{H}_n^t + (\mathbf{H}_n^t)^\intercal\right), \quad n \in \mathcal{N}, t \in \mathcal{T}, \tag{2}$$

where $\mathbf{H}_n^t = \mathbf{L}^\intercal \nabla F_n(\mathbf{W}^t; \boldsymbol{\xi}_n)\mathbf{R}^\intercal$. Client n performs stochastic gradient descent to update matrix $\mathbf{A}^t$. We define η as the learning rate. Let $\mathbf{A}_n^{t+\frac{1}{2}}$ denote the locally updated low-rank matrix of client n in the t-th fine-tuning round, which is given by

$$\mathbf{A}_n^{t+\frac{1}{2}} = \mathbf{A}^t - \eta \nabla_{\mathbf{A}} F_n(\mathbf{W}^t; \boldsymbol{\xi}_n), \quad n \in \mathcal{N}, t \in \mathcal{T}. \tag{3}$$

(ii) Model Aggregation: When compared with the conventional federated LoRA schemes, in which clients must upload both locally-updated low-rank matrices, client n in FLoRG only needs to transmit $\mathbf{A}_n^{t+\frac{1}{2}}$ to the central server, thereby reducing the per-round uplink communication overhead by more than a half. After receiving the locally updated parameter matrices from all clients, the central server performs model aggregation with respect to $\mathbf{A}_n^{t+\frac{1}{2}}$ as

$$\mathbf{Q}^{t+1} = \frac{1}{N} \sum_{n \in \mathcal{N}} \left(\mathbf{A}_n^{t+\frac{1}{2}}\right)^\intercal \mathbf{A}_n^{t+\frac{1}{2}}, \quad t \in \mathcal{T}, \tag{4}$$

where $\mathbf{Q}^{t+1} \in \mathbb{R}^{k \times k}$ is the aggregated Gram matrix at the end of the t-th fine-tuning round. Due to the Gram matrix design, aggregating $\left(\mathbf{A}_n^{t+\frac{1}{2}}\right)^\intercal \mathbf{A}_n^{t+\frac{1}{2}}$ is linear and preserves the positive semi-definite (PSD) property. Therefore, the central server can obtain the true aggregated matrix. This removes the bilinear inconsistency induced by aggregating matrices $\mathbf{B}_n$ and $\mathbf{A}_n$ separately as in conventional federated LoRA schemes (i.e., aggregation error). Let r' denote the rank of $\mathbf{Q}^{t+1}$, which satisfies

$$r' = \text{rank}(\mathbf{Q}_n^{t+\frac{1}{2}}) \leq \text{rank}(\mathbf{Q}^{t+1}) \overset{\text{(a)}}{\leq} \min\{k, Nr\}. \tag{5}$$

where inequality (a) follows from the subadditivity property of rank operator.

(iii) Decomposition with Procrustes Alignment: Since $\mathbf{Q}^{t+1}$ is a square PSD Gram matrix, the central server performs eigendecomposition to $\mathbf{Q}^{t+1}$ as follows:

$$\mathbf{Q}^{t+1} = (\mathbf{P}^{t+1})^\intercal \Lambda^{t+1} \mathbf{P}^{t+1}, \quad t \in \mathcal{T}, \tag{6}$$

where $\mathbf{P}^{t+1} \in \mathbb{R}^{r' \times k}$. $\Lambda^{t+1} \in \mathbb{R}^{r' \times r'}$ denotes the eigenvalue matrix of $\mathbf{Q}^{t+1}$. Thus, a canonical decomposition which satisfies $(\tilde{\mathbf{A}}^{t+1})^\intercal \tilde{\mathbf{A}}^{t+1} = \mathbf{Q}^{t+1}$ is

$$\tilde{\mathbf{A}}^{t+1} = (\Lambda^{t+1})^{\frac{1}{2}} \mathbf{P}^{t+1}, \quad t \in \mathcal{T}, \tag{7}$$

where $\tilde{\mathbf{A}}^{t+1}$ is a valid decomposition of $\mathbf{Q}^{t+1}$. However, directly applying $\tilde{\mathbf{A}}^{t+1}$ for the subsequent fine-tuning round (i.e., $\mathbf{A}^{t+1} := \tilde{\mathbf{A}}^{t+1}$) may yield sub-optimal performance since the decomposition yields two challenges: **non-unique decomposition** and **rank mismatch**.

First, the above expression of matrix $\tilde{\mathbf{A}}^{t+1}$ provides a canonical representation of the decomposition of $\mathbf{Q}^{t+1}$, which is non-unique. This is because for any matrix with orthogonal columns $\mathbf{O} \in \mathbb{R}^{r' \times r'}$, we have $(\mathbf{O}\tilde{\mathbf{A}}^{t+1})^\intercal \mathbf{O}\tilde{\mathbf{A}}^{t+1} = \mathbf{Q}^{t+1}$. Furthermore, although the decomposition guarantees that $(\tilde{\mathbf{A}}^{t+1})^\intercal \tilde{\mathbf{A}}^{t+1}$ preserves the aggregated Gram matrix, there exist many such decompositions due to the non-uniqueness of decompositions (e.g., Cholesky decomposition or SVD) when the rank is deficient or the eigenvalues have multiplicities. However, this non-uniqueness affects the future fine-tuning. From the update rule in eqn. (2), each client performs local fine-tuning in the $(t+1)$-th fine-tuning round using the gradient which depends explicitly on $\mathbf{A}^{t+1}$. As a result, different decompositions of $\mathbf{Q}^{t+1}$ may yield different $\mathbf{A}^{t+1}$ which result in divergent gradient paths. These paths may potentially lead to unstable fine-tuning dynamics across rounds.

Second, as stated in eqn. (5), the rank of the decomposed matrix may be different from that of the original one (i.e., $r' \neq r$). In such cases, we need to recover a matrix of the target rank r for consistency across fine-tuning rounds.

The aforementioned challenges motivate reparameterizing matrix $\tilde{\mathbf{A}}^{t+1}$ to stabilize the subsequent fine-tuning process while enforcing the target rank r. To this end, we propose Procrustes alignment to project $\tilde{\mathbf{A}}^{t+1}$ onto the r-rank subspace. Let $\mathbf{S}^t \in \mathbb{R}^{r \times r'}$ denote this Procrustes alignment matrix in the t-th fine-tuning round. The matrix after projection is denoted as $\mathbf{S}^t\tilde{\mathbf{A}}^{t+1}$. In particular, Procrustes alignment minimizes the Frobenius norm between the matrix after projection $\mathbf{S}^t\tilde{\mathbf{A}}^{t+1}$ and $\mathbf{A}^t$, which minimizes the drift caused by the non-uniqueness of matrix decomposition. We then formulate the following optimization problem in the t-th fine-tuning round:

$$\mathcal{P}_1: \quad \underset{\mathbf{S}^t}{\text{minimize}} \quad \left\|\mathbf{S}^t\tilde{\mathbf{A}}^{t+1} - \mathbf{A}^t\right\|_F^2 \tag{8a}$$

$$\text{subject to} \quad (\mathbf{S}^t)^\intercal \mathbf{S}^t = \mathbf{I}_{r'}, \tag{8b}$$

where $\mathbf{I}_{r'} \in \mathbb{R}^{r' \times r'}$ denotes the identity matrix. To solve problem $\mathcal{P}_1$, we first convert the objective function (8a) into the following form:

$$\begin{aligned}\left\|\mathbf{S}^t\tilde{\mathbf{A}}^{t+1} - \mathbf{A}^t\right\|_F^2 &= \operatorname{Tr}\left((\mathbf{A}^t)^\intercal \mathbf{A}^t\right) + \operatorname{Tr}\left((\tilde{\mathbf{A}}^{t+1})^\intercal (\mathbf{S}^t)^\intercal \mathbf{S}^t \tilde{\mathbf{A}}^{t+1}\right) - 2\operatorname{Tr}\left(\mathbf{A}^t(\tilde{\mathbf{A}}^{t+1})^\intercal (\mathbf{S}^t)^\intercal\right) \\ &= \left\|\mathbf{A}^t\right\|_F^2 + \left\|\tilde{\mathbf{A}}^{t+1}\right\|_F^2 - 2\operatorname{Tr}\left(\mathbf{A}^t(\tilde{\mathbf{A}}^{t+1})^\intercal (\mathbf{S}^t)^\intercal\right).\end{aligned} \tag{9}$$

Since $\|\mathbf{A}^t\|_F^2$ and $\|\tilde{\mathbf{A}}^{t+1}\|_F^2$ have been determined after matrix decomposition at the end of the t-th fine-tuning round, problem $\mathcal{P}_1$ is equivalent to the following problem:

$$\mathcal{P}_2: \quad \underset{\mathbf{S}^t}{\text{maximize}} \quad \operatorname{Tr}\left(\mathbf{A}^t(\tilde{\mathbf{A}}^{t+1})^\intercal (\mathbf{S}^t)^\intercal\right) \tag{10a}$$

$$\text{subject to} \quad \text{constraint (8b)}.$$

Then, we present the following theorem to obtain the optimal solution to problems $\mathcal{P}_2$ and $\mathcal{P}_1$, with the proof provided in Appendix A.1.

Theorem 1. (Optimal Procrustes Alignment Matrix) *We denote the SVD of* $\mathbf{A}^t(\tilde{\mathbf{A}}^{t+1})^\intercal$ *as* $\mathbf{A}^t(\tilde{\mathbf{A}}^{t+1})^\intercal = \mathbf{U}^{t+1}\mathbf{\Sigma}^{t+1}(\mathbf{V}^{t+1})^\intercal$*, where* $\mathbf{U}^{t+1} \in \mathbb{R}^{r \times r'}$ *and* $\mathbf{V}^{t+1} \in \mathbb{R}^{r' \times r'}$ *have orthogonal columns. Let* $\mathbf{\Sigma}^{t+1} = \operatorname{diag}(\sigma_1^{t+1}, \sigma_2^{t+1}, \ldots, \sigma_{r'}^{t+1}) \in \mathbb{R}^{r' \times r'}$ *denote the diagonal matrix of eigenvalues of* $\mathbf{A}^t(\tilde{\mathbf{A}}^{t+1})^\intercal$*. The optimal solution* $\mathbf{S}^{t,\star}$ *to problems* $\mathcal{P}_2$ *and* $\mathcal{P}_1$ *satisfies*

$$\mathbf{S}^{t,\star} = \mathbf{U}^{t+1}(\mathbf{V}^{t+1})^\intercal. \tag{11}$$

The main benefits of our proposed Procrustes alignment are two-fold. First, it resolves the issue due to non-unique decomposition. In particular, the Procrustes alignment approach selects, among all valid decompositions of $\mathbf{Q}^{t+1}$, the one which is nearest to that in the last fine-tuning round (i.e., $\mathbf{A}^t$) in Frobenius norm. Thus, it stabilizes the gradients in the subsequent fine-tuning rounds. Second, it

addresses the rank mismatch issue of the decomposed matrix by using a semi-orthogonal Procrustes alignment matrix to project $\tilde{\mathbf{A}}^{t+1}$ with rank r' onto the target r-rank subspace.

After the central server has determined matrix $\mathbf{S}^{t,\star}$, it calculates the low-rank matrix for the $(t+1)$-th training round as $\mathbf{A}^{t+1} = \mathbf{S}^{t,\star}\tilde{\mathbf{A}}^{t+1}$. Thus, the full model in the $(t+1)$-th round can be obtained as

$$\mathbf{W}^{t+1} = \mathbf{W}^0 + \mathbf{L}(\mathbf{A}^{t+1})^\intercal \mathbf{A}^{t+1}\mathbf{R}. \tag{12}$$

Then, the central server broadcasts the matrix $\mathbf{A}^{t+1}$ to all clients on the downlink. Note that the proposed FLoRG can effectively reduce the per-round communication overhead by more than half when compared with traditional federated LoRA schemes. The workflow of our proposed FLoRG is presented in Appendix A.2.

3.2 Theoretical Analysis

In this section, we analyze the convergence rate of our proposed FLoRG. Without loss of generality, we consider nonconvex loss functions in our analysis. We first present the following assumptions which are widely used in the literature (e.g., (Li et al., 2020; Wang et al., 2020; Guo et al., 2025)).

Assumption 1. (L-Smoothness (Li et al., 2020; Wang et al., 2020)) *The loss function of each client n is continuously differentiable and L-smooth. That is, for arbitrary two matrices $\mathbf{W}^t$ and $\mathbf{W}^{t+1}$, we have $f_n(\mathbf{W}^{t+1}) \leq f_n(\mathbf{W}^t) + \langle \nabla f_n(\mathbf{W}^t), \mathbf{W}^{t+1} - \mathbf{W}^t \rangle_F + \frac{L}{2}\|\mathbf{W}^{t+1} - \mathbf{W}^t\|_F^2$, $t \in \mathcal{T}$, $n \in \mathcal{N}$.*

Assumption 2. (Bounded Gradient (Li et al., 2020; Wang et al., 2020)) *The local stochastic gradient of $\mathbf{A}$ is upper-bounded, i.e., $\mathbb{E}_{\boldsymbol{\xi}_n \sim \mathcal{D}_n}\left[\|\nabla_{\mathbf{A}} F_n(\mathbf{W}^t; \boldsymbol{\xi}_n)\|_F^2\right] \leq \psi$, $t \in \mathcal{T}$, $n \in \mathcal{N}$.*

Assumption 3. (Bounded Parameter Space (Guo et al., 2025)) *The Frobenius norm of model parameter matrices $\mathbf{A}^t$ and $\tilde{\mathbf{A}}^t$ are upper-bounded by two positive constants $C_{\mathbf{A}}$ and $\tilde{C}_{\mathbf{A}}$, respectively, i.e., $\|\mathbf{A}^t\|_F \leq C_{\mathbf{A}}$, $\|\tilde{\mathbf{A}}^t\|_F \leq \tilde{C}_{\mathbf{A}}$, $t \in \mathcal{T}$, $t \in \mathcal{T}$.*

In addition, we present two lemmas to facilitate our convergence analysis, with the proof provided in Appendices A.3 and A.4, respectively.

Lemma 1. *We denote $\mathbf{H}^t = \frac{1}{N}\sum_{n\in\mathcal{N}} \mathbf{H}_n^t = \frac{1}{N}\sum_{n\in\mathcal{N}} \mathbf{L}^\intercal \mathbf{G}_n^t \mathbf{R}^\intercal$. Let $\lambda_{\min}(\mathbf{X})$ denote the smallest positive eigenvalue of matrix $\mathbf{X}$. For any matrices $\mathbf{A}^t$ and $\mathbf{H}^t$, we have*

$$\left\langle \mathbf{H}^t,\ (\mathbf{A}^t)^\top \mathbf{A}^t\big(\mathbf{H}^t + (\mathbf{H}^t)^\top\big) + \big(\mathbf{H}^t + (\mathbf{H}^t)^\top\big)(\mathbf{A}^t)^\top \mathbf{A}^t \right\rangle_F \geq 4\,\lambda_{\min}\big((\mathbf{A}^t)^\top \mathbf{A}^t\big)\ \left\|\mathbf{H}^t\right\|_F^2,\ \ t \in \mathcal{T}. \tag{13}$$

Lemma 2. *Let $\mathbf{S}^t$ denote an arbitrary Procrustes alignment matrix in the t-th fine-tuning round. We define $\Delta_{\text{proc}}^{t+1} = \left\|\mathbf{S}^t\tilde{\mathbf{A}}^{t+1} - \mathbf{A}^t\right\|_F^2 - \left\|\mathbf{S}^{t,\star}\tilde{\mathbf{A}}^{t+1} - \mathbf{A}^t\right\|_F^2 \geq 0$. The difference of two Procrustes alignment matrices is bounded, i.e.,*

$$\left\|\mathbf{S}^t - \mathbf{S}^{t,\star}\right\|_F^2 \leq \frac{\Delta_{\text{proc}}^{t+1}}{\lambda_{\min}\left(\tilde{\mathbf{A}}^{t+1}(\mathbf{A}^t)^\intercal\right)}, \quad t \in \mathcal{T}. \tag{14}$$

Now, we present the convergence rate of our proposed FLoRG in the following theorem, with the proof provided in Appendix A.5.

Theorem 2. (Convergence Rate of FLoRG) *We denote the optimal model as $\mathbf{W}^\star$. Under Assumptions 1 − 3 and Lemmas 1 − 2, if the learning rate satisfies $\eta < 8\min_{t\in\mathcal{T}}\{\lambda_{\min}\left((\mathbf{A}^t)^\intercal \mathbf{A}^t\right)\} - 1$, then the convergence rate of our proposed FLoRG is bounded by*

$$\frac{1}{T}\sum_{t\in\mathcal{T}} \left\|\nabla f(\mathbf{W}^t)\right\|_F^2 \leq \underbrace{\frac{f(\mathbf{W}^1) - f(\mathbf{W}^\star)}{T\Omega}}_{\text{Initial optimality gap}} + \underbrace{\frac{\eta^2\psi^2}{2\Omega} + \frac{3L\eta^2\psi\left(\eta^2\psi + 2C_{\mathbf{A}}^2\right)}{2\Omega}}_{\text{Residual bias term}}$$
$$+ \underbrace{\frac{1}{T\Omega}\sum_{t\in\mathcal{T}} \frac{2\eta\psi\tilde{C}_{\mathbf{A}}^2\Delta_{\text{proc}}^t}{N\lambda_{\min}\left(\tilde{\mathbf{A}}^t(\mathbf{A}^{t-1})^\intercal\right)}}_{\text{Procrustes alignment drift term}}, \tag{15}$$

where $\Omega = 4\eta \min_{t\in\mathcal{T}}\{\lambda_{\min}\left((\mathbf{A}^t)^\intercal \mathbf{A}^t\right)\} - \frac{\eta^2}{2} - \frac{\eta}{2}$.

Based on the theoretical analysis, we can observe that the convergence rate of our proposed FLoRG depends on three terms. The first term is the initial optimality gap, which diminishes as the number of fine-tuning rounds T increases. The second term is the non-diminishing residual bias term. The third term is the Procrustes alignment drift term, which captures the impact of the Procrustes alignment on the convergence rate. When the Procrustes alignment is applied, Δ^t_{proc} becomes zero. Hence, the third term becomes zero, under which we can achieve a tighter bound and improve the convergence rate.

4 Experimental Results

4.1 Experiment Setup

Base Models, Datasets, and Baseline Schemes: We choose OPT-125M (Zhang et al., 2022) and RoBERTa-large (Liu et al., 2019) as two base models with different scales. In particular, OPT-125M and RoBERTa-large have 125M and 355M parameters. We choose GLUE (Wang et al., 2018) as a benchmark with MRPC, QQP, MNLI, QNLI, WNLI, and RTE datasets. We compare the performance of our proposed FLoRG with the following baseline schemes:

- **FedIT** (Zhang et al., 2024): Each client transmits the locally updated LoRA matrices $\mathbf{B}$ and $\mathbf{A}$ to the central server. The central server aggregates $\mathbf{B}$ and $\mathbf{A}$ separately.
- **FeDeRA** (Yan et al., 2024): Each client transmits locally updated LoRA matrices $\mathbf{B}$ and $\mathbf{A}$ to the central server. The central server aggregates $\mathbf{BA}$ and performs SVD to obtain the updated matrices $\mathbf{B}$ and $\mathbf{A}$.
- **FFA-LoRA** (Sun et al., 2024) Each client freezes matrix $\mathbf{A}$ and only updates matrix $\mathbf{B}$. The central server performs aggregation on matrix $\mathbf{B}$.
- **FedSA-LoRA** (Guo et al., 2025): Each client locally updates matrices $\mathbf{B}$ and $\mathbf{A}$ but only shares matrix $\mathbf{A}$ for aggregation.

Implementation Details: To present the learning performance, we show the average testing accuracy of all clients. To characterize the incurred communication overhead, we present the total number of parameters transmitted between all clients and the central server. To model the data heterogeneity across clients' local datasets, we use the Dirichlet distribution $\text{Dir}(\rho)$ to create non-independent and identically distributed (non-iid) data partitioning. In particular, $\rho > 0$ controls the degree of non-iidness across clients' local datasets. A lower value of ρ indicates a higher degree of data heterogeneity. In the ablation studies, we choose RoBERTa-large as the base model. Unless stated otherwise, we set $\eta =$ 5e-5, $\rho = 0.5$, $N = 20$, and $r = 4$.

4.2 Testing Accuracy

In this section, we compare the testing accuracy of different schemes under different base models and datasets. Due to the space limit, we present the results under MNLI, QNLI, WNLI, and RTE datasets. Results in Table 1 show that our proposed FLoRG outperforms the baseline schemes under those four datasets. In particular, on OPT-125M, FLoRG improves the testing accuracy over the strongest baseline by 2.77 on MNLI, 0.86 on QNLI, 2.66 on WNLI, and 1.52 on RTE. On RoBERTa-large, the margins are 0.49 on MNLI, 0.75 on QNLI, 2.73 on WNLI, and 1.31 on RTE. Additional experimental results are presented in Appendix A.6. These results validate the superiority of FLoRG.

4.3 Comparison of the Communication Overhead

In this section, we compare the communication overhead incurred under different baseline schemes. We use the QNLI dataset to conduct the experiments. Results in Table 2 show that to achieve the target test accuracy, our proposed FLoRG uses a much lower total number of transmitted model parameters when compared with the baselines. On OPT-125M and RoBERTa-large, FLoRG achieves up to 78% and 82% reduction in the number of transmitted parameters, respectively. This demonstrates that FLoRG can significantly reduce the communication overhead.

Table 1: Comparison of the testing accuracy across different baseline schemes.

Base Model	Dataset	FLoRG	FedIT	FeDeRA	FFA-LoRA	FedSA-LoRA
OPT-125M	MNLI	**87.20**	79.19	80.99	83.67	84.43
	QNLI	**89.69**	84.35	86.55	87.80	88.83
	WNLI	**65.41**	58.30	59.11	62.75	62.60
	RTE	**68.77**	61.26	64.32	65.89	67.25
RoBERTa-large	MNLI	**91.39**	84.76	88.20	89.12	90.90
	QNLI	**92.44**	87.63	89.91	90.82	91.69
	WNLI	**66.34**	59.19	61.97	63.55	63.61
	RTE	**71.41**	64.11	66.97	68.62	70.10

Table 2: Comparison of the total number of transmitted parameters (in millions) to achieve the target accuracy. Symbol "$-$" means that the target accuracy cannot be achieved.

Base Model	Target Acc.	FLoRG	FedIT	FeDeRA	FFA-LoRA	FedSA-LoRA
OPT-125M	80.00	**4.1M**	18.9M	12.3M	7.9M	10.5M
	85.00	**5.3M**	–	21.1M	13.7M	18.1M
RoBERTa-large	80.00	**4.3M**	23.4M	15.1M	9.7M	12.9M
	85.00	**7.2M**	40.6M	25.8M	16.7M	22.1M

4.4 Ablation Studies

Impact of the Procrustes alignment In this subsection, we study the impact of our proposed Procrustes alignment on the learning performance. Results in Table 3 show that by applying Procrustes alignment, our proposed FLoRG yields a consistent improvement in terms of the testing accuracy. On OPT-125M, Procrustes alignment provides an improvement of 3.40 on MRPC, 2.86 on QQP, 6.27 on MNLI, 2.97 on QNLI, 5.60 on WNLI, and 4.45 on RTE, respectively. On RoBERTa-large, Procrustes alignment provides an improvement of 3.37 on MRPC, 2.84 on QQP, 2.46 on MNLI, 3.86 on QNLI, 4.34 on WNLI, and 4.31 on RTE, respectively, whereas FLoRG without Procrustes alignment can only achieve a comparable testing accuracy to FeDeRA, as shown in Table 1. It showcases the importance of our proposed Procrustes alignment to improve the fine-tuning performance.

Table 3: Comparison of the testing accuracy of FLoRG with and without Procrustes alignment.

Base Model	FLoRG	MRPC	QQP	MNLI	QNLI	WNLI	RTE
OPT-125M	w/ Procrustes alignment	**86.54**	**88.71**	**87.20**	**89.69**	**65.41**	**68.77**
	w/o Procrustes alignment	83.14	85.85	80.93	86.72	59.81	64.32
RoBERTa-large	w/ Procrustes alignment	**89.87**	**91.27**	**91.39**	**92.48**	**66.41**	**71.40**
	w/o Procrustes alignment	86.50	88.43	88.93	88.62	62.07	67.09

Impact of the Rank In this subsection, we vary r to demonstrate the impact of rank on the fine-tuning performance. In particular, we conduct experiments under $r = 2, 4, 8$, respectively. We present the results under the WNLI and RTE datasets. Results in Table 4 show that our proposed FLoRG outperforms the baseline schemes under different rank settings, demonstrating the robustness of our proposed FLoRG under various ranks. Additional experimental results can be found in Appendix A.6.

Robustness to the Data Heterogeneity In this subsection, we study the impact of the degree of data heterogeneity across clients' local datasets on the fine-tuning performance. We present the results under the WNLI and RTE datasets. It can be observed in Table 5 that under different degrees of data heterogeneity, our proposed FLoRG outperforms the baseline schemes. In addition, as the degree of data heterogeneity increases (i.e., ρ decreases), the improvement over the baseline

Table 4: Comparison of the testing accuracy under different ranks.

Rank	Dataset	FLoRG	FedIT	FeDeRA	FFA-LoRA	FedSA-LoRA
$r = 2$	WNLI	**60.55**	55.57	56.30	57.50	59.14
	RTE	**65.82**	58.41	61.19	62.88	64.30
$r = 4$	WNLI	**66.34**	59.19	61.97	63.55	63.61
	RTE	**71.41**	64.11	66.97	68.62	70.10
$r = 8$	WNLI	**68.83**	61.70	63.52	65.10	66.47
	RTE	**72.10**	64.78	66.99	68.02	70.61

Table 5: Comparison of the testing accuracy under different degrees of data heterogeneity.

Non-IIDness	Dataset	FLoRG	FedIT	FeDeRA	FFA-LoRA	FedSA-LoRA
$\rho = 0.1$	WNLI	**60.12**	53.07	54.21	56.14	57.74
	RTE	**65.30**	55.60	59.19	61.20	60.75
$\rho = 0.5$	WNLI	**66.34**	59.19	61.97	63.55	63.61
	RTE	**71.41**	64.11	66.97	68.62	70.10
$\rho = 1$	WNLI	**67.83**	61.70	63.52	64.33	65.61
	RTE	**72.21**	66.90	68.71	70.40	71.78

schemes also increases, showcasing the robustness and superiority of our proposed FLoRG under heterogeneous data settings. Additional experimental results are presented in Appendix A.6.

Matrix Initialization for Matrices $\mathbf{L}$ and $\mathbf{R}$ In this subsection, we show the impact of the initialization of matrices $\mathbf{L}$ and $\mathbf{R}$ on the learning performance. In particular, we compare our proposed semi-orthogonal initialization with Kaiming initialization (He et al., 2015) and SVD initialization (Boutsidis & Gallopoulos, 2007). Results in Table 6 show that the semi-orthogonal approach outperforms the other two approaches in most cases, demonstrating the effectiveness of our proposed initialization approach for matrices $\mathbf{L}$ and $\mathbf{R}$.

Table 6: Comparison of the testing accuracy under different initialization schemes.

Base Model	Initialization	MRPC	QQP	MNLI	QNLI	WNLI	RTE
OPT-125M	Semi-orthogonal	**86.54**	88.71	**87.20**	**89.69**	**65.41**	68.77
	Kaiming	84.35	**88.90**	85.23	87.73	62.29	**69.31**
	SVD	86.41	87.69	83.19	88.74	64.37	67.69
RoBERTa-large	Semi-orthogonal	**89.87**	**91.27**	91.39	**92.48**	**66.41**	**71.40**
	Kaiming	87.68	92.34	89.11	91.57	64.19	70.32
	SVD	88.70	90.37	**91.49**	91.45	65.08	**71.40**

5 Conclusion

In this work, we proposed FLoRG, a federated fine-tuning framework with low-rank Gram matrices. In particular, FLoRG features a single low-rank matrix instead of two low-rank matrices as in the conventional LoRA module. By transmitting this matrix and aggregating the corresponding Gram matrix, FLoRG eliminates the error induced by separately aggregating two matrices and significantly reduces the per-round communication overhead. Moreover, we proposed a Procrustes-fixing approach to reparameterize the decomposed low-rank matrix after model aggregation. We theoretically analyzed the convergence rate of our proposed FLoRG framework and characterized the impact of our proposed Procrustes alignment on the convergence. Experimental results show that our proposed FLoRG framework achieves a higher testing accuracy and can reduce the communication overhead by up to 82% when compared with four baseline schemes.

REFERENCES

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

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

Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. Federated fine-tuning of large language models under heterogeneous tasks and client resources. In *Proc. Advances Neural Info. Process. Syst. (NeurIPS)*, Dec. 2024.

David Bensaïd, Noam Rotstein, Roy Velich, Daniel Bensaïd, and Ron Kimmel. SingLoRA: Low rank adaptation using a single matrix. *arXiv preprint arXiv:2507.05566*, Jul. 2025.

Christos Boutsidis and Efstratios Gallopoulos. SVD based initialization: A head start for nonnegative matrix factorization. *Pattern recognition*, 41(4):1350–1362, Sep. 2007.

Yuji Byun and Jaeho Lee. Towards federated low-rank adaptation of language models with rank heterogeneity. In *Proc. Conf. Nations Americas Assoc. Comput. Linguistics (NAACL))*, Apr 2025.

Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, and Gauri Joshi. Heterogeneous loRA for federated fine-tuning of on-device foundation models. In *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Nov. 2024.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Proc. Proc. Advances Neural Info. Process. Syst. (NeurIPS)*, Dec. 2023.

Zihan Fang, Zheng Lin, Zhe Chen, Xianhao Chen, Yue Gao, and Yuguang Fang. Automated federated pipeline for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2404.06448*, Apr. 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, Jul. 2024.

Pengxin Guo, Shuang Zeng, Yanran Wang, Huijie Fan, Feifei Wang, and Liangqiong Qu. Selective aggregation for low-rank adaptation in federated learning. In *Proc. Int. Conf. Learn. Representations (ICLR)*, Apr. 2025.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *Proc. int. Conf. Mach. Learn. (ICML)*, Jul. 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imageNet classification. In *Proc. IEEE Int. Conf. Compute. Vis. (ICCV)*, Dec. 2015.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. In *Proc. Int. Conf. Learn. Representations (ICLR)*, Apr. 2022.

Dawid J Kopiczko, Tijmen Blankevoort, and Yuki M Asano. VeRA: Vector-based random matrix adaptation. In *Proc. Int. Conf. Learn. Representations (ICLR)*, May 2024.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of FedAvg on non-IID data. In *Proc. Int. Conf. Learn. Representations (ICLR)*, Apr. 2020.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *Proc. int. Conf. Mach. Learn. (ICML)*, Jul. 2024.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pre-training approach. *arXiv preprint arXiv:1907.11692*, Jul. 2019.

Guodong Long, Tao Shen, Jing Jiang, Michael Blumenstein, et al. Dual-personalizing adapter for federated foundation models. In *Proc. Advances Neural Info. Process. Syst. (NeurIPS)*, Dec. 2024.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, Fort Lauderdale, FL, Apr. 2017.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving LoRA in privacy-preserving federated learning. In *Proc. Int. Conf. Learn. Representations (ICLR)*, May 2024.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, Apr. 2018.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Proc. Advances Neural Info. Process. Syst. (NeurIPS)*, Dec. 2020.

Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. FLoRA: Federated fine-tuning large language models with heterogeneous low-rank adaptations. In *Proc. Advances Neural Info. Process. Syst. (NeurIPS)*, Dec. 2024.

Feijie Wu, Zitao Li, Yaliang Li, Bolin Ding, and Jing Gao. FedBiOT: LLM local fine-tuning in federated learning without full model. In *Proc. ACM SIGKDD Conf. on Knowl. Discovery Data Mining (KDD)*, Aug. 2024.

Yunlu Yan, Chun-Mei Feng, Wangmeng Zuo, Rick Siow Mong Goh, Yong Liu, and Lei Zhu. Federated residual low-Rank adaptation of large language models. In *Proc. Int. Conf. Learn. Representations (ICLR)*, Apr. 2025.

Yuxuan Yan, Qianqian Yang, Shunpu Tang, and Zhiguo Shi. FeDeRA: Efficient fine-tuning of language models in federated learning leveraging weight decomposition. In *Proc. Advances Neural Info. Process. Syst. (NeurIPS)*, Dec. 2024.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the federated GPT: Federated instruction tuning. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Apr. 2024.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *Proc. Int. Conf. Learn. Representations (ICLR)*, May 2023a.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, May 2022.

Zhuo Zhang, Yuanhang Yang, Yong Dai, Qifan Wang, Yue Yu, Lizhen Qu, and Zenglin Xu. FedPETuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In *Proc. Annu. Meeting Assoc. of Comput. Linguistics (ACL)*, Jul. 2023b.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-efficient llm training by gradient low-rank projection. In *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2024.

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