Towards Robust and Efficient Federated Low-Rank Adaptation with Heterogeneous Clients

Anonymous ACL submission

Abstract

Federated fine-tuning for Large Language Models (LLMs) has recently gained attention due to the heavy communication overhead of transmitting large model updates. Low Rank Adaptation (LoRA) has been proposed as a solution, yet its application in federated learning is complicated by discordance in aggregation. Existing methods addressing this discordance often suffer from performance degradation at low ranks in heterogeneous data settings. In response, we introduce LoRA-A^2 (Low Rank Adaptation with Alternating freeze and Adaptive rank selection), which demonstrates robustness in challenging settings with low ranks and high data heterogeneity. Our experimental findings reveal that LoRA-A^2 maintains performance even under extreme heterogeneity and low rank conditions, achieving up to a 99.8% reduction in uploaded parameters compared to full fine-tuning without compromising performance. This adaptive mechanism boosts robustness and communication efficiency in federated fine-tuning, enabling the practical deployment of LLMs in resource-constrained environments.

1 Introduction

Large Language Models (LLMs), exemplified by ChatGPT (OpenAI, 2024), Llama (Dubey et al., 2024) and others, represent a hallmark of the current era. These models are being widely applied in real-world scenarios by fine-tuning them on various task-specific datasets (Dodge et al., 2020). With the expansion of edge devices, the potential to leverage rich, privacy-sensitive data for fine-tuning LLMs has shifted the focus toward federated fine-tuning. Despite its potential, this is often infeasible due to the large size of LLMs, which require extensive computational and communication resources from local devices.

Parameter-Efficient Fine-Tuning (PEFT) methods (Lester et al., 2021; Liu et al., 2022) are increasingly being explored in the context of federated fine-tuning. Among these, Low-Rank Adaptation (LoRA) (Hu et al., 2021) is particularly noteworthy for its significant reduction in number of communicated parameters. However, naive application of LoRA in Federated Learning (FL) (McMahan et al., 2017) environment comes with several challenges such as aggregation discordance. Although several solutions have been proposed, they often remain vulnerable to high heterogeneity and low ranks due to a limited parameter space, making it difficult to reduce rank size for communication efficiency in realistic FL scenarios.

To address this, we introduce LoRA-A^2 (**Lo**w **R**ank **A**daptation with **A**lternating freeze and **A**daptive rank selection), which is robust to both high heterogeneity and low ranks. LoRA-A^2 incorporates two main strategies: alternating freeze, which switches between freezing LoRA modules B and A in each round, and adaptive rank selection, which identifies and updates only important ranks in LoRA modules. We conduct experiments across various rank sizes and heterogeneity levels, comparing our algorithm with multiple baselines. Through the experiments, we reveal the vulnerabilities of existing methods and highlight the robustness of LoRA-A^2 in challenging conditions, providing analyses of the reasons for its robustness. Additionally, we empirically demonstrate that our approach achieves performance comparable to or exceeding that of full fine-tuning, while uploading less than 0.2% of parameters to the server.

Our contributions can be summarized as follows:

- We address the vulnerabilities of previous federated LoRA methods in high heterogeneity and low-rank settings, and propose a novel algorithm, LoRA-A^2, which demonstrates robustness in these challenging conditions.
- Our algorithm effectively reduces communication costs, achieving a 99.8% reduction in uploaded parameters compared to federated

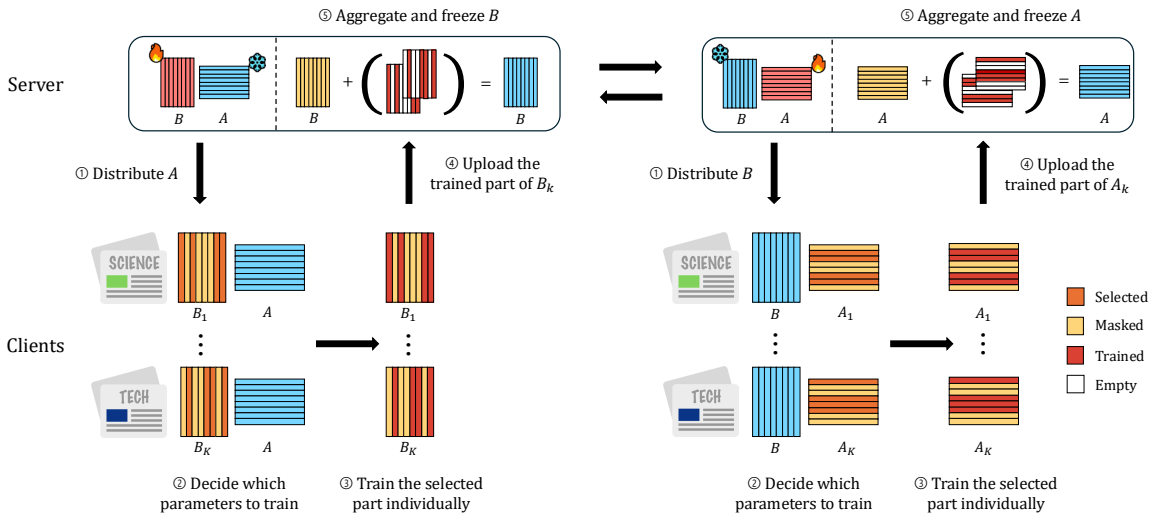


Figure 1: An overview of the proposed method, LoRA-A^2. It alternately trains B and A of the LoRA adapters, with each client training only a subset of the downloaded parameters. LoRA-A^2 is free from several issues for using LoRA in FL, which are discussed in Section 3. A detailed explanation of the method is provided in Section 4.

full fine-tuning, while maintaining or surpassing its performance.

- We provide visualization on adaptive rank selection process and a thorough empirical exploration on how important ranks are efficiently trained and transmitted.

2 Related Works

LoRA with adaptive rank selection LoRA (Hu et al., 2022) is a widely used PEFT method for LLMs. It tries to approximate the updated part of the pre-trained model with two smaller size of matrices. This approach is inspired by previous studies (Li et al., 2018; Aghajanyan et al., 2021), which suggest that newly learned parameters for adaptation lie within a low dimensional subspace.

AdaLoRA (Zhang et al., 2023) assumes a scenario where the total parameter budget is limited. It adaptively selects the rank for each LoRA adapter under this constraint, with a criterion for rank selection based on singular values of the updated part.

ALoRA (Liu et al., 2024) utilizes a router for each LoRA adaptor. The router determines which part of each LoRA adaptor should be either turned on or off, enabling efficient fine-tuning via pruning. Similarly, DoRA (Mao et al., 2024) re-splits LoRA into smaller groups of LoRAs. During the training session, it estimates the importance of each small LoRA, allowing the parts with less contribution across the whole LoRA to be pruned. Our research extends this adaptive rank selection in centralized learning so that each client adaptively selects different ranks suitable for their own dataset.

Federated learning with LoRA As training LLMs on mobile devices becomes feasible, fine-tuning LLMs via FL has recently gained attention. In line with this trend, using LoRA for federated fine-tuning (Babakniya et al., 2023; Kuo et al., 2024; Wang et al., 2024), is also being considered. However, simply adopting LoRA for FL presents several obstacles, which are discussed in Section 3.

HetLoRA (Cho et al., 2023) assumes that each client may have different computational power, which is a common scenario in FL. Based on this assumption, it allows each client to use a LoRA adapter of varying sizes. Zero-padding is then applied to equalize the LoRA sizes for aggregation.

Sun et al. (2024) point out that aggregating the two matrices of a LoRA adapter separately cannot fully approximate the original LoRA adapter. Based on this finding, they propose FFA-LoRA, which addresses this issue by freezing half of each LoRA throughout the entire fine-tuning session.

FlexLoRA (Bai et al., 2024) aggregates the product of two matrices that make up each LoRA adapter and then decomposes the aggregated parameters back into two smaller matrices via singular value decomposition. This approach allows FlexLoRA to overcome the challenges addressed by HetLoRA and FFA-LoRA, respectively, though at the cost of increased computational cost on the server-side for the decomposition process.

3 Problem Formulation

Low rank adaptation Because LLMs have billions of parameters, fine-tuning them for specific domains demands significant computational power, which may be infeasible in many situations. To address this issue, PEFT techniques such as LoRA (Hu et al., 2022) have recently gained attention, as they can effectively reduce the number of parameters that need to be trained. Specifically, when fine-tuning a pre-trained weight matrix $W_0 \in \mathbb{R}^{d_1 \times d_2}$ to obtain W, LoRA achieves this by decomposing ΔW, the update of the weight matrix, into smaller matrices $B \in \mathbb{R}^{d_1 \times r}$ and $A \in \mathbb{R}^{r \times d_2}$:

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

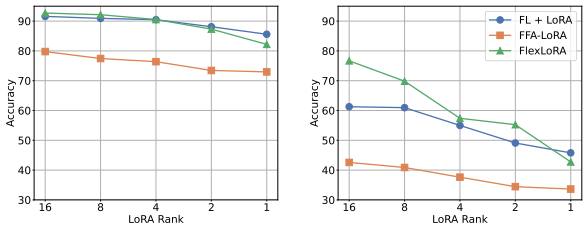
where $r \ll \{d_1, d_2\}$ denotes the rank of LoRA. With this approximation, the number of trainable parameters is reduced from $d_1 \cdot d_2$ to $r \cdot (d_1 + d_2)$.

Federated LoRA and discordance problem Consider a global pre-trained model W_0 and a set of clients $\{1, 2, \cdots, K\}$. The objective in federated fine-tuning is to update W_0 to obtain a model W that is suitable for local datasets $\mathcal{D}_k$. However, fine-tuning LLMs is very expensive for local devices in terms of both computation and communication, as billions of parameters must be trained and transmitted in each round.

LoRA presents a promising approach in FL for reducing communication costs, as only low rank module B and A are trained and transmitted, allowing the number of communicated parameters to be linearly reduced by the rank r of LoRA modules. However, the straightforward application of LoRA in FL introduces a significant issue known as discordance (Sun et al., 2024), primarily due to aggregation algorithms. In methods like FedAvg (McMahan et al., 2017), where each weight is aggregated individually, discordance occurs between the actual and aggregated parameters. That is,

$$\begin{aligned} \sum_{k=1}^{K} w_k \Delta W_k &= \sum_{k=1}^{K} w_k B_k A_k \\ &\neq \left(\sum_{k=1}^{K} w_k B_k \right) \left(\sum_{k=1}^{K} w_k A_k \right) \end{aligned} \tag{2}$$

in general, where $\sum_{k=1}^{K} w_k = 1$ with $w_k \geq 0$ for all $k \in [K]$. One might consider aggregating $\Delta W_k = B_k A_k$ directly to eliminate the discordance, but this approach entails decomposing $\Delta W = \sum_{k=1}^{K} w_k \Delta W_k$ back into B and A for the next round, which is computationally unstable and non-trivial.



(a) $Dir(0.1)$ (b) $Dir(0.01)$

Figure 2: Accuracy of previous Federated LoRA methods across different rank sizes in heterogeneous data settings.

Limited parameter space in low rank and high data heterogeneity This discrepancy can be effectively addressed by either freezing the LoRA module A, as suggested by Sun et al. (2024), or employing SVD decomposition, as outlined by Bai et al. (2024). However, Figure 2 illustrates that the accuracy of these approaches decreases significantly at lower ranks in the presence of high heterogeneity. We attribute this decline primarily to the restricted parameter space imposed by LoRA. A limited training parameter space constrains the optimization capabilities for complex federated learning tasks, and a restricted aggregation parameter space exacerbates conflicts among clients. A detailed analysis of this limited parameter space is provided in Appendix C.

4 Proposed Method

To tackle the identified challenges, we propose a novel framework called **Lo**w **R**ank **A**daptation with **A**lternating freeze and **A**daptive rank selection for federated learning, or LoRA-A^2, for communication efficient FL with LoRA. LoRA-A^2 adaptively selects LoRA ranks for training and communication. And it trains and transmits only the selected part of each adaptor in an alternating way.

4.1 Alternating Freeze

LoRA-A^2 efficiently addresses the issue of discordance by employing a simple alternating freeze technique to train the LoRA modules B and A. Instead of solely training module B while keeping module A frozen permanently, as suggested by FFA-LoRA (Sun et al., 2024), LoRA-A^2 alternates between the two: LoRA module A is frozen during even rounds, while module B is frozen during

odd rounds. This method preserves the optimization space while effectively resolving discordance. Specifically, when freezing A, we have

$$\begin{aligned}\Delta W &= \sum_{k=1}^{K} (w_k B_k)\, A \\ &= \sum_{k=1}^{K} (w_k B_k A_k) = \sum_{k=1}^{K} (w_k \Delta W_k)\,,\end{aligned} \tag{3}$$

and the same applies when freezing B. In this way, LoRA-A^2 trains both B and A, ensuring that A does not remain the same as its initial value.

To further enhance the effect of alternating optimization, we adopt different learning rates for B and A, inspired by LoRA+ (Hayou et al., 2024). Figure 6 demonstrates the effectiveness of alternating freeze and learning rate adjustment.

4.2 Adaptive Rank Selection

Furthermore, we propose an adaptive rank selection method designed to reduce the number of transmitted parameters while preserving the training and aggregation parameter space. This approach selects important LoRA ranks to match local communication rank budget r_i out of global LoRA adapter with rank r_G adaptively based on the local dataset. We mainly focus on communication cost for uploading parameters to the server as it is known that upload bandwidth is generally much slower than download bandwidth and is the major part of communication cost (Konečný et al., 2017; Suresh et al., 2017; Kairouz et al., 2021). The adaptive rank selection process provides two key benefits: it minimizes client conflicts by allowing different clients to choose different LoRA ranks in high heterogeneity, and reallocates rank resources from unimportant LoRA modules to modules that require more fine-tuning which is especially effective when communication rank budget is small.

To quantify which ranks are more important, we introduce our novel criterion $S_{m,i}$, as follows:

$$\begin{aligned}S_{m,i}^{B_k} &= \|\Delta B_{k[:,i]} A_{[i,:]}\|_F \\ S_{m,i}^{A_k} &= \|B_{[:,i]} \Delta A_{k[i,:]}\|_F\end{aligned} \tag{4}$$

Our criterion is based on the Frobenius norm of each rank's contribution $(C_{m,i})$ to the change in ΔW, represented as $\Delta W_k^{t+1} - \Delta W_k^t = \sum(\Delta B_{k[:,i]} A_{[i,:]}) = \sum C_{m,i}$. This criterion captures the impact of each rank on model updates, considering the interaction between the updated and frozen LoRA modules. This approach is better suited for LoRA modules than simpler magnitude-based criteria, $||\Delta B_{k[:,i]}||$ or $||\Delta A_{k[i,:]}||$, as it explicitly accounts for the interplay between the updated and frozen modules, which is a critical factor in our alternating freeze strategy. The ablation study in Table 6 empirically supports the superiority of this criterion.

After computing $S_{m,i}^{B_k}$ or $S_{m,i}^{A_k}$ for each module m, we select top-$(r_i \cdot N)$ LoRA ranks from a total of $r_G \cdot N$ based on the scores across the entire model, where N denotes the number of target modules across all the layers of the base model. We refer to the set of selected ranks of client k as $\mathcal{R}_k$.

Once the ranks are selected, each client defines LoRA module mask $M_k^{(m)}$ for the module m to be

$$\begin{aligned}M_{k[:,i]}^{(m)} &= \begin{cases}\mathbf{1}_{d_1}^T & \text{if } i \in \mathcal{R} \\ \mathbf{0}_{d_1}^T & \text{otherwise}\end{cases}, \\ M_{k[i,:]}^{(m)} &= \begin{cases}\mathbf{1}_{d_2} & \text{if } i \in \mathcal{R} \\ \mathbf{0}_{d_2} & \text{otherwise}\end{cases},\end{aligned} \tag{5}$$

which is producted element-wise to the updated part of B_k (or A_k). That is, before each backpropagation, LoRA-A^2 calculates

$$\begin{aligned}\Delta {B_k}^{(m)} &\leftarrow \Delta {B_k}^{(m)} \odot {M_k}^{(m)} \\ \Delta {A_k}^{(m)} &\leftarrow \Delta {A_k}^{(m)} \odot {M_k}^{(m)}\end{aligned} \tag{6}$$

for each B_k (or A_k), where the notation $\odot$ stands for the Hadamard product. After each local training, each client uploads $B_k \odot M_k$ (or $A_k \odot M_k$), resulting in sparsification and reducing the number of uploaded parameters. Then, the server aggregates the uploaded ones, which are again added to the B_k (or A_k) saved two rounds before. Algorithm 1 and 2 provide the pseudocode of LoRA-A^2.

4.3 Theoretical Insights

In this section, we provide a brief theoretical analysis of the parameter spaces relevant to previous methods and our proposed LoRA-A^2 framework. To substantiate our approach, we introduce the following proposition:

Proposition 1. For a model W, consider LoRA-based FL algorithms which update r rank parameters per round. Let $\Omega_{\mathcal{A}}$ denote the space of all possible parameter values that an algorithm $\mathcal{A} \in \{\text{FFA-LoRA}, \text{FL+LoRA}, \text{FlexLoRA}, \text{LoRA-A}^2\}$ can make. Then, we have $\Omega_{\text{FFA-LoRA}} \subsetneq \Omega_{\text{FL + LoRA}} = \Omega_{\text{FlexLoRA}} \subset \Omega_{\text{LoRA-A}^2}$.

Algorithm 1 LoRA-A^2

Initialize $\Delta W = BA$ with $B \in \mathbb{R}^{d_1 \times r_G}$ and $A \in \mathbb{R}^{r_G \times d_2}$ for each LoRA adaptor
for $t = 1, 2, \cdots, T$ **do**
  Sample participants $\mathcal{K}^{(t)} \subseteq [K]$ for round t
  $w_k = |\mathcal{D}_k| / \left(\sum_{k=1}^{K} |\mathcal{D}_k|\right)$
  if $t \,\%\, 2 = 1$ **then**
    for $k = 1, 2, \cdots, K$ in parallel **do**
      $B_k^{(t+1)} = \text{LocalTraining}(B^{(t)}, t)$
      $B^{(t+1)} = B^{(t)} + \sum_{k=1}^{K} w_k B_k^{(t+1)}$
      $A^{(t+1)} = A^{(t)}$
    end for
  else
    for $k = 1, 2, \cdots, K$ in parallel **do**
      $A_k^{(t+1)} = \text{LocalTraining}(A^{(t)}, t)$
      $A^{(t+1)} = A^{(t)} + \sum_{k=1}^{K} w_k A_k^{(t+1)}$
      $B^{(t+1)} = B^{(t)}$
    end for
  end if
end for

Algorithm 2 LocalTraining

[Rank Selection]
Calculate importance scores following (4)
Define the mask M_k following (5)
[Local Training]
if $t \,\%\, 2 = 1$ **then**
  $B_k^{(t;\, e-1)} = B^{(t)}$
  for $e = 1, 2. \cdots, E$ **do**
    $\Delta B_k^{(t;\, e)} = B_k^{(t;\, e-1)} - B_k^{(t;\, e-1)}$
    $\Delta B_k^{(t;e)} = \Delta B_k^{(t;\, e)} \odot M_k$
    Backpropagate $\Delta B_k^{(t;\, e+1)}$
  end for
  Return: $B_k^{(t;\, E)}$
else
  for $e = 1, 2. \cdots, E$ **do**
    $\Delta A_k^{(t;\, e)} = A_k^{(t;\, e-1)} - A_k^{(t;\, e-1)}$
    $\Delta A_k^{(t;e)} = \Delta A_k^{(t;\, e)} \odot M_k$
    Backpropagate $\Delta A_k^{(t;e+1)}$
  end for
  Return: $A_k^{(t;\, E)}$
end if

The proof for the proposition is provided in Appendix D.

Our algorithm is designed to adaptively select the relevant training and aggregation parameter spaces while concurrently reducing the number of parameters that are updated.

5 Experiments

In this section, we evaluate the performance of our algorithm against existing FL methods combined with LoRA across various heterogeneity settings and datasets. We assess performance based on accuracy and the total number of uploaded parameters.

5.1 Experimental Settings

Across all experiments, we utilize RoBERTa-base (Liu et al., 2019) pre-trained model as the base model. For fine-tuning, we choose BANKING77 (Casanueva et al., 2020) and 20 Newsgroups (Lang, 1995) datasets for fine-tuning the base model. These datasets are chosen for their ability to simulate a controlled level of data heterogeneity using Dirichlet distribution (Hsu et al., 2019). Dataset statistics for different levels of heterogeneity are reported in Appendix A.

Unless otherwise stated, we trained 30 local clients, assuming a full participation setting, i.e., $\mathcal{K}^{(t)} = [K]$ for all $t \in [T]$. The clients were trained for 50 rounds with 5 local epochs. Detailed hyperparameters for experiments are specified in Appendix B.

For baselines, we adopt four methods that utilize LoRA for federated fine-tuning: FL + LoRA, FFA-LoRA (Sun et al., 2024), FlexLoRA (Bai et al., 2024), and HetLoRA (Cho et al., 2023), where FL + LoRA stands for the naive implementation of LoRA in FedAvg (McMahan et al., 2017).

5.2 Main Results

We compare our algorithm with the baseline methods under various data heterogeneity settings in BANKING77 and 20 Newsgroups datasets to demonstrate that our algorithm outperforms previous federated LoRA fine-tuning methods across different non-IID settings and LoRA ranks.

Robustness of LoRA-A^2 in low ranks and high heterogeneity Table 1 highlights the vulnerability of previous methods under conditions of high heterogeneity and low ranks. The accuracy of baseline methods declines significantly as rank decreases, whereas our algorithm maintains its performance, achieving up to a 23% accuracy advantage. This suggests that reducing LoRA ranks is challenging for previous methods under realistic heterogeneous data conditions. Also, Our algorithm

Method	BANKING77 Dataset			20 Newsgroups Dataset			Communicated Parameters*
	$Dir(0.5)$	$Dir(0.1)$	$Dir(0.01)$	$Dir(0.5)$	$Dir(0.1)$	$Dir(0.01)$	
FL (w/o LoRA)	$92.76_{\pm 0.30}$	$90.29_{\pm 0.73}$	$67.58_{\pm 0.44}$	$70.93_{\pm 1.04}$	$68.82_{\pm 0.69}$	$64.41_{\pm 0.30}$	186B
FL + LoRA$_{(\text{Rank}=8)}$	$92.80_{\pm 0.24}$	$90.47_{\pm 0.53}$	$60.96_{\pm 1.47}$	$\underline{70.44}_{\pm 0.28}$	$\underline{67.33}_{\pm 0.18}$	$43.90_{\pm 1.08}$	1.99B
FFA-LoRA$_{(\text{Rank}=8)}$	$87.20_{\pm 0.57}$	$77.44_{\pm 1.28}$	$40.88_{\pm 1.04}$	$67.00_{\pm 0.67}$	$61.27_{\pm 0.71}$	$37.34_{\pm 0.30}$	0.991B
FlexLoRA$_{(\text{Rank}=8)}$	$\mathbf{93.35}_{\pm 0.24}$	$\mathbf{92.14}_{\pm 0.25}$	$\underline{69.84}_{\pm 0.65}$	$\mathbf{70.59}_{\pm 0.22}$	$\mathbf{68.10}_{\pm 0.38}$	$\mathbf{60.41}_{\pm 1.54}$	1.99B
Ours$_{(\text{Rank}=8)}$	$\underline{93.24}_{\pm 0.27}$	$\underline{91.61}_{\pm 0.39}$	$\mathbf{70.13}_{\pm 1.22}$	$70.26_{\pm 0.21}$	$67.12_{\pm 0.22}$	$\underline{54.50}_{\pm 1.44}$	1.31B
FL + LoRA$_{(\text{Rank}=4)}$	$\underline{92.86}_{\pm 0.08}$	$88.11_{\pm 0.88}$	$54.99_{\pm 0.59}$	$70.33_{\pm 0.12}$	$\underline{67.29}_{\pm 0.19}$	$43.12_{\pm 2.67}$	0.991B
FFA-LoRA$_{(\text{Rank}=4)}$	$86.90_{\pm 1.14}$	$76.38_{\pm 0.61}$	$37.63_{\pm 0.80}$	$67.75_{\pm 0.45}$	$61.25_{\pm 0.26}$	$36.04_{\pm 0.80}$	0.497B
FlexLoRA$_{(\text{Rank}=4)}$	$92.71_{\pm 0.31}$	$\underline{90.53}_{\pm 0.70}$	$\underline{57.38}_{\pm 1.30}$	$70.05_{\pm 0.14}$	$\mathbf{68.00}_{\pm 0.33}$	$\underline{50.50}_{\pm 2.09}$	0.991B
Ours$_{(\text{Rank}=4)}$	$\mathbf{93.22}_{\pm 0.24}$	$\mathbf{91.43}_{\pm 0.63}$	$\mathbf{69.63}_{\pm 1.52}$	$\mathbf{70.28}_{\pm 0.32}$	$67.12_{\pm 0.60}$	$\mathbf{53.04}_{\pm 1.68}$	0.888B
FL + LoRA$_{(\text{Rank}=2)}$	$91.97_{\pm 0.43}$	$85.59_{\pm 1.13}$	$49.08_{\pm 0.56}$	$\mathbf{70.14}_{\pm 0.13}$	$65.40_{\pm 0.31}$	$39.07_{\pm 2.23}$	0.497B
FFA-LoRA$_{(\text{Rank}=2)}$	$84.65_{\pm 1.05}$	$73.44_{\pm 0.88}$	$34.44_{\pm 2.15}$	$68.12_{\pm 0.47}$	$61.57_{\pm 0.38}$	$36.65_{\pm 0.52}$	0.249B
FlexLoRA$_{(\text{Rank}=2)}$	$\underline{92.22}_{\pm 0.50}$	$87.31_{\pm 0.27}$	$\underline{55.24}_{\pm 2.19}$	$70.03_{\pm 0.31}$	$66.17_{\pm 1.70}$	$48.23_{\pm 1.73}$	0.497B
Ours$_{(\text{Rank}=2)}$	$\mathbf{93.10}_{\pm 0.07}$	$\mathbf{92.02}_{\pm 0.36}$	$\mathbf{69.40}_{\pm 0.48}$	$\underline{70.12}_{\pm 0.18}$	$\mathbf{67.02}_{\pm 0.26}$	$\mathbf{52.99}_{\pm 2.56}$	0.528B
FL + LoRA$_{(\text{Rank}=1)}$	$\underline{90.61}_{\pm 0.10}$	$\underline{82.24}_{\pm 1.68}$	$\underline{45.78}_{\pm 1.04}$	$69.40_{\pm 0.33}$	$\underline{63.16}_{\pm 0.53}$	$\underline{36.58}_{\pm 0.98}$	0.249B
FFA-LoRA$_{(\text{Rank}=1)}$	$82.51_{\pm 0.53}$	$72.96_{\pm 0.54}$	$33.68_{\pm 0.20}$	$67.73_{\pm 0.30}$	$61.35_{\pm 0.22}$	$34.44_{\pm 0.68}$	0.124B
FlexLoRA$_{(\text{Rank}=1)}$	$90.40_{\pm 0.54}$	$82.20_{\pm 0.74}$	$42.75_{\pm 0.89}$	$\underline{69.53}_{\pm 0.25}$	$62.98_{\pm 1.12}$	$35.54_{\pm 0.68}$	0.249B
Ours$_{(\text{Rank}=1)}$	$\mathbf{93.21}_{\pm 0.13}$	$\mathbf{91.87}_{\pm 0.33}$	$\mathbf{68.88}_{\pm 1.15}$	$\mathbf{70.31}_{\pm 0.24}$	$\mathbf{66.95}_{\pm 0.07}$	$\mathbf{54.84}_{\pm 1.15}$	0.270B

Table 1: Results with RoBERTa-base on BANKING77 and 20 Newsgroups datasets. Smaller α for $Dir(\alpha)$ implies that the simulated setting is more heterogeneous. The best results on each dataset are shown in **bold** and second best is shown by underline. * This column reports the total number of uploaded parameters, averaged across rows.

consistently achieves the highest performance or remains within a 1% margin of the best-performing baselines at ranks 8 and 4 while showing large performance gap in low ranks.

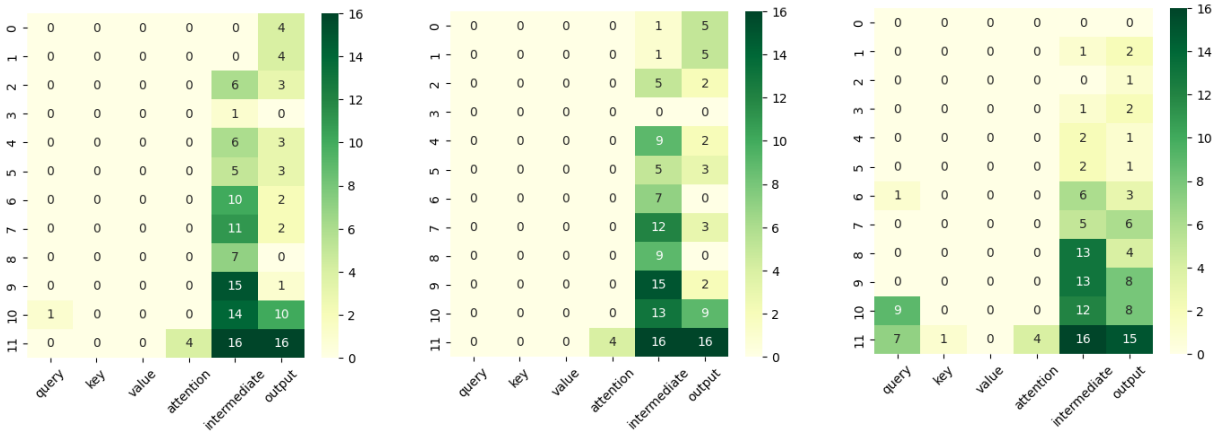
Communication cost reduction by LoRA-A^2 Decreasing LoRA ranks in federated LoRA methods reduces the communication cost linearly. Our algorithm achieves performance comparable to or better than fully fine-tuned models even at rank 1, allowing for up to a 99.8% reduction in communicated parameters with minimal performance loss. This demonstrates that LoRA-A^2 effectively solves the significant communication cost challenges of federated fine-tuning on LLMs.

5.3 Analysis on Adaptive Rank Selection

In this section, we visualize the process of our adaptive rank selection, and explore how we efficiently train and send important ranks, highlighting the robustness of our algorithm in heterogeneous and low rank environments. To simulate extreme cases of both identical and different client distributions, we test our algorithm on a pathological toy dataset using the 20 Newsgroups dataset. In this setup, 20 clients each holds data from only two classes, with consecutive pairs sharing the same classes, while others do not. For instance, clients 0 and 1 have classes "medical" and "space," whereas clients 2 and 3 have "motorcycle" and "religions". Detailed settings are shown in Appendix C.

Robustness to low rank by Adaptive Module Selection In this experiment, our algorithm selects $2 \cdot N^{(m)}$ ranks from a total of $16 \cdot N^{(m)}$ across the whole RoBERTa model, guided by our importance criterion, and visualizes the adaptive selection of modules. Figure 3 illustrates the number of ranks selected for each module in the model during the training. The figure shows that most modules are allocated with zero ranks, indicating either no need for fine-tuning or the insignificance of updates on those modules. This suggests that our adaptive rank selection automatically prunes out modules that do not require additional fine-tuning.

To further justify that our adaptive rank selection adequately selects important modules, we conduct an ablation study on module selection, akin to the approach in AdaLoRA (Zhang et al., 2023) but in a federated environment. Figure 5 displays the model's performance when only specific modules or layers are fine-tuned. The results show that tun-



(a) client 0 (b) client 1 (c) client 2

Figure 3: Visualization on number of selected rank per module. The x-axis shows RoBERTa module types, while the y-axis indicates layer numbers. Experimented on the 20 Newsgroups dataset with a pathological data distribution. Average 2 ranks were selected out of 16 ranks by our adaptive rank selection algorithm.

ing the last layer and intermediate, dense modules leads to better performance, highlighting their importance for fine-tuning. This aligns with our findings, where the last layers and intermediate / output dense modules are automatically selected, demonstrating the algorithm's effectiveness in prioritizing essential modules for additional fine-tuning.

Robustness to data heterogeneity by client clustering Another effect of rank selection is clustering of clients to minimize conflicts among clients with different dataset and enhance cooperation among clients with similar dataset.

Figure 4 (a) illustrates how much local rank parameters are shared among different clients. The figure shows that clients that share data distributions share more rank parameters than the clients who do not share data tends to share less parameters. This trend is also evident at the module level in Figure 3, where clients 0 and 1 select a similar number of ranks for each module, differing from client 2, while retaining the tendency to choose more ranks from the last layers or intermediate and output dense modules. This indicates that clients with similar datasets select the same ranks, promoting cooperative model training, whereas clients with differing data select fewer common ranks, resulting in independent parameter training. Figure 4 (b) further supports this by visualizing the cosine similarity between clients' model updates, showing near 1 for clients with the same classes and near zero for those who do not share data. This underscores the cooperative nature of updates from similar clients while maintaining independence from those with different data, contributing to our algorithm's robustness against data heterogeneity.

# of Ranks	RoBERTa-Large FL+LoRA	FFA-LoRA	FlexLoRA*	Ours
8	80.56	63.08	-	**85.85**
4	78.37	62.07	-	**84.70**
2	75.47	60.70	-	**84.70**
1	72.02	55.97	-	**85.78**

Table 2: Experimental results on RoBERTa-Large model. The level of heterogeneity is $Dir(0.01)$.
* FlexLoRA results could not be reported due to an ill-conditioned matrix issue in SVD decomposition

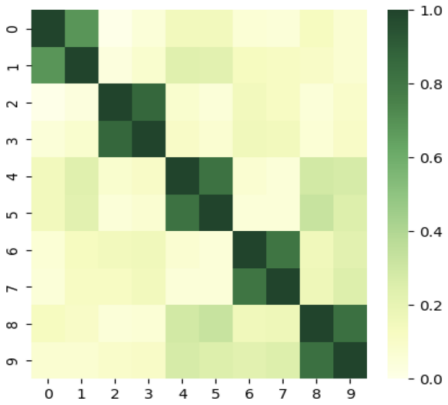
# of Ranks	DistilBERT FL+LoRA	FFA-LoRA	FlexLoRA*	Ours
8	32.58	18.82	51.21	**52.97**
4	36.92	16.73	41.26	**51.24**
2	27.14	15.49	34.05	**49.97**
1	21.59	14.29	21.01	**48.89**

Table 3: Experimental results on DistilBERT (Sanh et al., 2020) model. The level of heterogeneity is $Dir(0.01)$.

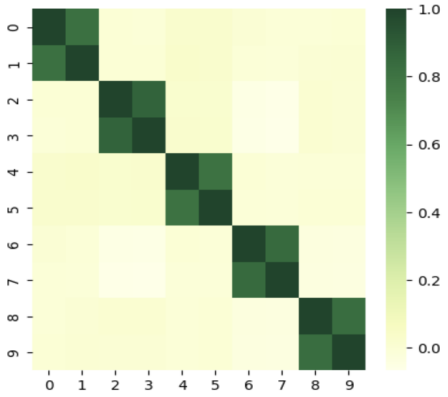
5.4 Ablation Studies

Through these ablation studies, we show empirical evidence for our engineering choices on aggregation tactics and rank selection criteria.

Efficacy of alternating freeze To address the discordance problem in federated LoRA aggregation, we employ an alternating freeze approach that alternately freezes LoRA modules B and A, rather than exclusively freezing module A as in

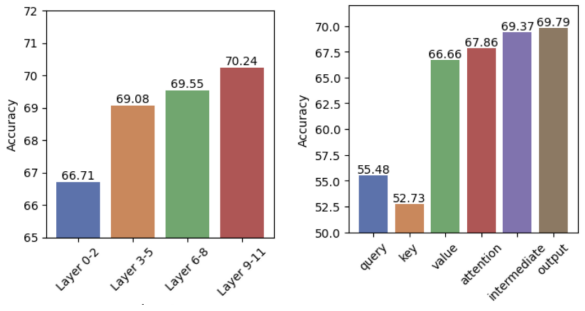


(a) Rank selection similarity



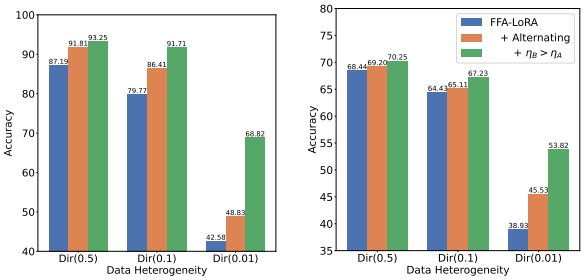
(b) Cosine similarity of local updates

Figure 4: Visualization of similarity between clients. the x and y axes represent individual clients trained on 20 Newsgroups dataset with pathologic data distribution.



(a) Selected layers (b) Selected modules

Figure 5: Ablation analysis on the performance of model when solely fine-tuned on selected layers or types of modules. Experimented on 20 Newsgroups dataset with Dir(0.1) heterogeneity.



(a) BANKING77 (b) 20 Newsgroups

Figure 6: Effect of alternating freeze under varying levels of heterogeneity.

FFA-LoRA (Cho et al., 2023). Furthermore, we set the learning rate of module B, η_B, to be five times that of module A, η_A, inspired by LoRA+ (Hayou et al., 2024). This configuration further enhances overall performance and robustness in highly heterogeneous environments. Figure 6 illustrates the performance difference among these approaches, showing that solely freezing A is less effective under high data heterogeneity, whereas alternating freeze demonstrates greater robustness.

Scalability and generalizability on model structures In evaluating the scalability and generalizability of our algorithm across various model structures, we present the results in Table 2 and Table 3. These tables illustrate the performance of our model when applied to diverse architectures and parameter configurations. The outcomes clearly demonstrate that our algorithm achieves superior performance, even on models with a larger number of parameters or different architectures. This highlights the robust scalability and generalizability of our approach across different model structures.

Additional experiments We also include further experiments addressing resource heterogeneity settings, pathological distributions, as well as investigations into convergence speed and computational overhead in Appendix C.

6 Conclusion

In this work, we tackle the vulnerability of previous methods in high heterogeneity and low ranks by proposing a novel algorithm, LoRA-A^2, which shows robustness in these challenging conditions with alternating freeze and adaptive rank selection. Our approach offers significant improvements in communication efficiency without compromising performance, as demonstrated by a reduction of 99.8% in parameter uploads compared to full fine-tuning. Through extensive experiments, we establish LoRA-A^2 as a superior alternative, providing a practical pathway for efficient and effective federated fine-tuning in diverse and resource-constrained environments.

7 Limitations

LoRA-A^2 shows promising results and we plan to distribute the implementation code with detailed instructions for reproducibility. However, several areas remain open for future exploration.

First, our work mainly focuses on classification tasks, primarily due to computational constraints and the use of Dirichlet distribution to simulate non-IID conditions. However, extending LoRA-A^2 to more complex tasks, such as natural language generation, could offer additional perspectives. Future work with more resources could explore these broader applications.

Second, our experiments are primarily conducted on comparatively smaller language models, such as RoBERTa-base and RoBERTa-large, due to limited computation resources. Applying LoRA-A^2 to larger models, such as LLaMA or GPT-style architectures, could provide an opportunity to test its scalability. Investigating how well the method handles the increased parameter space of these state-of-the-art models could further demonstrate its efficiency.

Finally, due to the limited access to real world datasets, our current results are mainly based on simulated settings. Extensive research on real world dataset, which typically exhibit more diverse types of noise and heterogeneity would help understand performance and robustness of LoRA-A^2 in practical, dynamic environments.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online. Association for Computational Linguistics.

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

Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. 2024. Federated fine-tuning of large language models under heterogeneous tasks and client resources. *Preprint*, arXiv:2402.11505.

Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Hei Li Kwing, Titouan Parcollet, Pedro PB de Gusmão, and Nicholas D Lane. 2020. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Daoyuan Chen, Liuyi Yao, Dawei Gao, Bolin Ding, and Yaliang Li. 2023. Efficient personalized federated learning via sparse model-adaptation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 5234–5256. PMLR.

Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, Matt Barnes, and Gauri Joshi. 2023. Heterogeneous loRA for federated fine-tuning of on-device foundation models. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. 2024. Lora+: Efficient low rank adaptation of large models. *arXiv 2402.12354*.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the effects of non-identical data distribution for federated visual classification. *Preprint*, arXiv:1909.06335.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih

Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. 2021. Advances and open problems in federated learning. *Preprint*, arXiv:1912.04977.

Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2017. Federated learning: Strategies for improving communication efficiency. *Preprint*, arXiv:1610.05492.

Kevin Kuo, Arian Raje, Kousik Rajesh, and Virginia Smith. 2024. Federated lora with sparse communication. *Preprint*, arXiv:2406.05233.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. 2018. Measuring the intrinsic dimension of objective landscapes. *Preprint*, arXiv:1804.08838.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Preprint*, arXiv:2205.05638.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Zequan Liu, Jiawen Lyn, Wei Zhu, Xing Tian, and Yvette Graham. 2024. ALoRA: Allocating low-rank adaptation for fine-tuning large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 622–641, Mexico City, Mexico. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yulong Mao, Kaiyu Huang, Changhao Guan, Ganglin Bao, Fengran Mo, and Jinan Xu. 2024. DoRA: Enhancing parameter-efficient fine-tuning with dynamic rank distribution. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11662–11675, Bangkok, Thailand. Association for Computational Linguistics.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. 2024. Improving loRA in privacy-preserving federated learning. In *The Twelfth International Conference on Learning Representations*.

Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and H. Brendan McMahan. 2017. Distributed mean estimation with limited communication. *Preprint*, arXiv:1611.00429.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*.

A Dataset Statistics

BANKING77 (Casanueva et al., 2020) is an intent classification dataset with 77 fine-grained intents

	$Dir(0.01)$		$Dir(0.1)$		$Dir(0.5)$	
	Train	Test	Train	Test	Train	Test
$\max\|\{\mathcal{D}_k\}\|_{k\in[K]}$	1317	877	911	606	576	383
$\min\|\{\mathcal{D}_k\}\|_{k\in[K]}$	1	1	58	37	151	100
$\max\|\{\mathcal{C}_k\}\|_{k\in[K]}$	5	5	12	12	20	14
$\min\|\{\mathcal{C}_k\}\|_{k\in[K]}$	1	1	5	5	20	12
Number of classes	20					
Number of clients	30					

Table 4: Statistics of 20 Newsgroups datasets.

	$Dir(0.01)$		$Dir(0.1)$		$Dir(0.5)$	
	Train	Test	Train	Test	Train	Test
$\max\|\{\mathcal{D}_k\}\|_{k\in[K]}$	639	212	672	185	473	133
$\min\|\{\mathcal{D}_k\}\|_{k\in[K]}$	50	30	139	43	248	75
$\max\|\{\mathcal{C}_k\}\|_{k\in[K]}$	15	10	34	24	65	52
$\min\|\{\mathcal{C}_k\}\|_{k\in[K]}$	2	2	18	15	37	31
Number of intents	77					
Number of clients	30					

Table 5: Statistics of BANKING77 dataset.

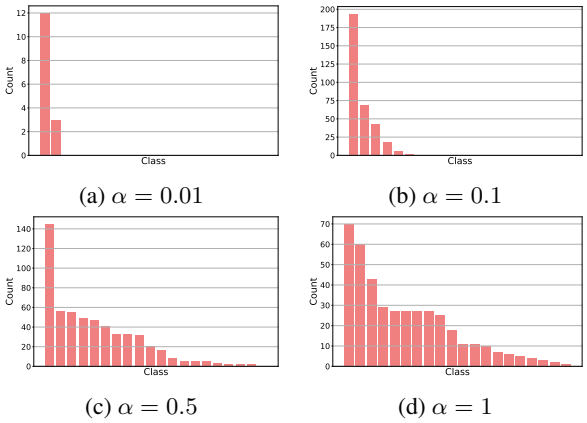
related to the banking domain, comprising 10,003 training samples and 3,080 test samples. 20 Newsgroups (Lang, 1995) is a widely used text classification dataset with 20 classes, each representing a unique topic. It contains 11,314 training samples and 7,532 test samples.

We provide the statistics of two datasets in Table 4 and Table 5, respectively. $\mathcal{D}_k$ and $|\mathcal{C}_k|$ denotes the local dataset of k and the number of unique classes in $\mathcal{D}_k$, respectively. Figure 7 shows the distribution of a local dataset for varying α simulating the Dirichlet distribution.

B Reproducibility

Hyperparameters When training, we use AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate of $\eta = 0.0005$. For LoRA-A^2, since B and A of each LoRA module are optimized separately, we use different learning rates for them. Specifically, $\eta_A = \eta$ is used for A and $\eta_B = 5 \cdot \eta_A$ is used for B, which is inspired by LoRA+ (Hayou et al., 2024). For HetLoRA, $\gamma = 0.99$ is used for the decaying factor as suggested by Cho et al. (2023). When evaluating, we merge the LoRA adapter ΔW with the pre-trained model W_0 using a scaling factor, so that $W_{ft} = W_0 + \frac{16}{r}\Delta W$.

Experiments Settings Without further specification, $K = 30$ clients participate in all experiments. We assume that there are no stragglers, i.e., $\mathcal{K}^{(t)} = K$ for all $t = 1, 2, \cdots, T$, where $T = 50$ represents the total communication round. Each local client trains 5 epochs before each communication round. This simulation setting is constructed



(a) $\alpha = 0.01$ (b) $\alpha = 0.1$ (c) $\alpha = 0.5$ (d) $\alpha = 1$

Figure 7: Local dataset distribution for varying heterogeneity. The 20 Newsgroup datasets of client 0 for each $Dir(\alpha)$ are visualized as an example.

	BANKING77 Dataset $Dir(0.1)$	BANKING77 Dataset $Dir(0.01)$	Communicated Parameters
Importance	$91.29_{\pm 0.76}$	$66.92_{\pm 1.58}$	0.215B
Magnitude	$\underline{91.71}_{\pm 0.23}$	$\underline{68.00}_{\pm 0.57}$	0.651B
Ours	$\mathbf{92.02}_{\pm 0.36}$	$\mathbf{69.40}_{\pm 0.48}$	0.507B

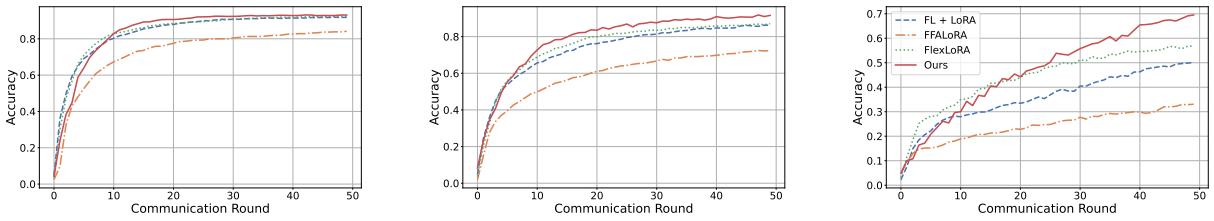
Table 6: Ablation study on scoring functions.

using Flower (Beutel et al., 2020), and all experiments with RoBERTa-base (Liu et al., 2019) are conducted three times to ensure reproducibility.

Base Model We mainly adopt the pre-trained RoBERTa-base (Liu et al., 2019) as the base model for fine-tuning, The base model has approximately 125M parameters, which are all frozen during the fine-tuning phase. And a frozen classifier is added upon the model, following Sun et al. (2024). For Table 2 and 3, we adopt RoBERTa-large and DistilBERT(Sanh et al., 2020), respectively. RoBERTa-large has approximately 355M parameters, and DistilBERT has approximately 82M parameters. All the models are downloaded from HuggingFace Tranformers (Wolf et al., 2020) library.

C Additional Experiments

Client Drift Experiment To thoroughly analyze the impact of data heterogeneity within constrained parameter spaces, we conducted additional experiments that illustrate the local client drift observed in baseline methods operating under these limitations. We quantified the degree of client drift by calculating the "Average Gradient Similarity," de-



(a) $Dir(0.5)$ (b) $Dir(0.1)$ (c) $Dir(0.01)$

Figure 8: Convergence curve of baseline methods in various levels of heterogeneity. Experimented on BANKING77 dataset and the ranks were all set to 2.

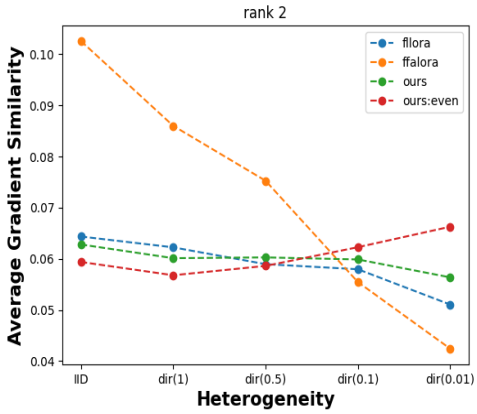


Figure 9: Average Gradient Similarity on various level of heterogeneity. Experimented on 20 Newsgroups dataset and the ranks were all set to 2.

Rank	FL + LoRA	FFA-LoRA	FlexLoRA	Ours
8	$53.80_{\pm 1.44}$	$52.60_{\pm 0.96}$	$\mathbf{60.36}_{\pm 1.15}$	$\underline{58.74}_{\pm 0.95}$
4	$55.03_{\pm 0.43}$	$50.57_{\pm 1.58}$	$\mathbf{59.12}_{\pm 0.98}$	$\underline{58.62}_{\pm 1.51}$
2	$50.40_{\pm 0.77}$	$48.36_{\pm 0.86}$	$\underline{55.46}_{\pm 0.99}$	$\mathbf{59.63}_{\pm 0.59}$
1	$\underline{51.24}_{\pm 3.12}$	$46.92_{\pm 1.30}$	$51.05_{\pm 0.69}$	$\mathbf{59.11}_{\pm 0.88}$

Table 7: Experiments on pathologic settings.

fined as follows:

$$AverageGradientSimilarity = \frac{1}{n^2}\sum_{i}^{n}\sum_{i}^{n}\frac{(\Delta W_i^t - \Delta W_i^{t-1})\cdot(\Delta W_j^t - \Delta W_j^{t-1})}{||\Delta W_i^t - \Delta W_i^{t-1}||\cdot||\Delta W_j^t - \Delta W_j^{t-1}||} \quad (7)$$

The experimental results presented in Figure 9 indicate a rapid decline in average gradient similarity as the level of heterogeneity increases. In contrast, our method demonstrates greater robustness, exhibiting lower client drift even in rounds where only the LoRA module A is updated. These findings are consistent with the results shown in Figure 2 and Table 1, which illustrate that FFA-LoRA experiences the most significant performance decline between the directional settings of 0.1 and 0.01, while our algorithm maintains its effectiveness in heterogeneous environments.

Efficacy of importance criterion As mentioned in Section 4.2, other criteria such as magnitude-based or importance-based scoring functions can be used for selecting ranks. Table 6 shows that our criterion outperforms others, with less communication than the magnitude-based criterion.

Convergence Speed Analysis Figure 8 shows the convergence curve of our algorithm and baseline methods. The figure demonstrates that our algorithm shows similar convergence speed compared to baseline methods in various levels of heterogeneity.

Pathologic Setting Table 7 provides experiments on pathologic setting, which is also used to generate Figure 4 in Section 5.3, to show the efficacy of adaptive rank selection. In this setting, we have $K = 20$ clients. And client $(2k-1)$ and client $(2k)$ exclusively possess half of class $(2k-1)$ and $(2k)$ of 20 Newsgroups datasets, respectively, for $k = 1, 2, \cdots, 10$.

Experiments on Resource Heterogeneity In this section, we assume that each client has a different communication cost budget (Chen et al., 2023). For example, some clients might use smartphones with Wi-Fi, while others may use 3G networks for federation. We aim to allow each client to have its own rank for the LoRA adapter, allowing clients with lower budgets to participate in training. In Table 8, we compare our method with HetLoRA and FlexLoRA, two previous LoRA methods that can handle resource heterogeneity in FL. Here, we assume that there are 5 types of ranks, $\{2^1, 2^2, 2^3, 2^4, 2^5\}$. The ranks are evenly distributed, with 6 clients assigned to each rank. Specifically, $r_k = 2^{k \bmod 6}$ for $k = 1, 2, \cdots, 30$.

	BANKING77 Dataset		Communicated
	$Dir(0.1)$	$Dir(0.01)$	Parameters
HetLoRA	$\underline{86.91}_{\pm 0.43}$	$\underline{68.53}_{\pm 2.14}$	3.09B
FlexLoRA	$73.01_{\pm 0.69}$	$45.41_{\pm 1.60}$	3.09B
Ours	$\mathbf{92.02}_{\pm 0.16}$	$\mathbf{70.67}_{\pm 0.76}$	1.97B

Table 8: Experimental results for the resource heterogeneity setting.

Computational OverHead Regarding computational overhead, our analysis shows that LoRA-A exhibits a 1.17x increase in computation time compared to standard FL+LoRA, slightly higher than FFA-LoRA (0.93x) and FlexLoRA (1x). However, we note that communication time, often the dominant bottleneck in federated learning, is significantly reduced by LoRA-A^2 (upto 99.8% reduction compared to full-finetuning), outweighing the modest increase in computation time.

D Theoretical Proofs

Here's brief proof for the proposition made in section 4.3: Proof) First, since FFA-LoRA freezes all the A_i's permanently, $\Omega_{\text{FFA-LoRA}} = \{B_i\}_{i=1}^N$. Next, since FL + LoRA and FlexLoRA update B_i's and A_i's simultaneously, $\Omega_{\text{FL + LoRA}} = \{(B_i, A_i)\}_{i=1}^N = \Omega_{\text{FlexLoRA}}$. Finally, $\Omega_{\text{LoRA-A}^2} = \left\{\left(\bar{B}_i, \bar{A}_i\right)\right\}_{i=1}^N$, where its subspace $\{B_i\}_{i=1}^N$ or $\{A_i\}_{i=1}^N$ is optimized according to the Alternating freeze and Adaptive rank selection algorithm. Therefore, noting that $r \leq r_G$, we have $\Omega_{\text{FFA-LoRA}} \subsetneq \Omega_{\text{FL + LoRA}} = \Omega_{\text{FlexLoRA}} \subset \Omega_{\text{LoRA-A}^2}$