PrivLoRA: Enhancing Privacy in LoRA-Based Fine-Tuning of Large Language Models for Federated Learning

Bayan Alzahrani
Computer Science
Colorado School of Mines
Golden, Colorado
balzahrani@mines.edu

Dejun Yang
Computer Science
Colorado School of Mines
Golden, Colorado
djyang@mines.edu

***Abstract*—Large Language Models (LLMs) are advanced neural networks pre-trained on massive datasets, enabling them to perform a wide range of natural language processing tasks with exceptional generalization and adaptability. Federated learning (FL) enables fine-tuning of LLMs across distributed clients while preserving data privacy, making it ideal for sensitive domains with diverse datasets. However, full fine-tuning (FFT) of LLMs in FL environments presents significant challenges, including high communication overhead, data privacy concerns, and low performance with heterogeneous client data. This paper introduces Private Low-Rank Adaptation (PrivLoRA), a novel privacy-preserving FL algorithm designed to address these challenges. PrivLoRA combines Low-Rank Adaptation (LoRA), an efficient parameter-efficient fine-tuning (PEFT) technique, with a novel layer-wise fine-tuning strategy and Singular Value Decomposition (SVD)-based initialization to mitigate performance degradation in highly heterogeneous data settings. The proposed framework incorporates advanced privacy mechanisms, including homomorphic encryption (HE) and differential privacy (DP), to ensure robust data protection while maintaining strong model performance. Extensive experiments demonstrate that PrivLoRA achieves superior accuracy and communication efficiency compared to state-of-the-art baselines, particularly in heterogeneous data environments. This research offers a significant advancement for privacy-preserving, communication-efficient LLM fine-tuning in collaborative and sensitive applications.**

***Index Terms*—Large language models, differential privacy, federated learning, LoRA**

I. Introduction

The rapid growth in the parameter scale of LLMs has significantly enhanced their generalization capabilities and introduced new application possibilities. Pre-trained LLMs have shown impressive performance even in zero-shot settings [1]. Despite these advancements, LLMs often require fine-tuning for optimal performance on specialized downstream tasks such as law, finance, and healthcare. However, fine-tuning LLMs in sensitive domains requires protecting data privacy while enabling collaborative model improvements. FL [2] offers a promising solution, allowing models to be trained across distributed devices or organizations without sharing raw data, thus minimizing direct data exposure. However, FL presents new challenges: 1) High communication costs are a significant challenge in FL, particularly as the sizes of LLMs have recently expanded dramatically—from 330 million parameters in BERT [3] to 540 billion in PaLM [4]. This growth has made full fine-tuning (FFT) highly resource-intensive, particularly in FL environments where client resources are often limited. 2) Potential privacy risks of sensitive information exposure through model updates [5], [6]. Addressing these challenges is crucial for enabling privacy-preserving and communication-efficient fine-tuning of LLMs in collaborative environments. Parameter-efficient fine-tuning (PEFT) methods [7] have been introduced as a training approach that enables LLMs to adapt to specific tasks by tuning only a subset of parameters rather than the entire model, significantly reducing resource demands. However, one of the biggest challenges in PEFT is that, as the level of data heterogeneity increases, so does the performance gap between FFT and PEFT methods [8], [9]. Among PEFT techniques, LoRA [10] has shown particular promise for FL due to its reduced parameter requirements and efficiency. Yet, it struggles with performance degradation when client data is highly non-IID (Independent and Identically Distributed). A primary challenge in such environments is the slow convergence rate, which results from the initialization process of the LoRA blocks. Specifically, when matrices A and B are initialized with zero and random Gaussian distributions, the model struggles to adapt effectively across diverse client datasets [9], [11]. To address all these challenges, we introduce Private Low-Rank Adaptation (PrivLoRA), an FL algorithm that ensures privacy while significantly reducing communication costs for LLMs. This research explores the following core question: How can we ensure data privacy in LLM fine-tuning using a practical federated learning approach that minimizes communication overhead while maintaining performance comparable to FFT, especially in highly heterogeneous data environments? PrivLoRA addresses three primary issues:

- Ensuring Data Privacy: In the first stage of our scheme, we utilize HE [12] to securely perform computations on encrypted data without revealing sensitive information.

For the second stage, PrivLoRA integrates DP using a Gaussian mechanism [13], which introduces noise into weight updates during training, ensuring that minor changes in individual data points have a minimal impact on the model output. This mechanism prevents attackers from inferring private data details, and we demonstrate that PrivLoRA adheres to (ε, δ)-differential privacy, a robust privacy standard.

- Enhancing Performance in Highly Heterogeneous Data: To mitigate the impact of non-iid data on LoRA's performance, we introduce a layer-wise fine-tuning approach with SVD initialization [14]. This strategy enhances performance by combining lightweight layer-wise fine-tuning on the client side with SVD-based initialization of the LoRA matrices A and B on the server side.
- Reducing Communication Overhead: Studies have shown that within federated learning, PEFT can potentially replace FFT without sacrificing performance and with notably reduced communication costs [15], [16]. We focus on LoRA [10] as it stands out among PEFT techniques for its ability to achieve similar or better performance than FFT with fewer tunable parameters, making it highly suitable for LLMs in diverse environments [8], [17].

PrivLoRA is designed to enable efficient, privacy-preserving federated fine-tuning of LLMs, offering a robust solution for privacy-sensitive applications in diverse and collaborative settings. In summary, this article contributes in the following ways:

- We introduce PrivLoRA, a privacy-preserving scheme for fine-tuning LLMs in FL environments. This approach utilizes HE and DP to protect sensitive data while ensuring effective and efficient fine-tuning, significantly reducing the risk of privacy breaches.
- We propose a novel layer-wise fine-tuning strategy combined with SVD-based initialization of LoRA matrices A and B. This two-step process enhances convergence and model performance, particularly in highly heterogeneous data settings, ensuring robust adaptability for real-world FL applications.
- We leverage LoRA, a PEFT technique, to replace traditional FFT in FL. To further enhance efficiency, we implement an alternating update mechanism where, in each communication round, only one LoRA matrix is updated while the other remains frozen. This innovative strategy achieves comparable performance to standard LoRA fine-tuning while effectively halving communication costs, addressing a critical challenge in FL deployment.
- We conducted extensive experiments to evaluate PrivLoRA across various datasets. The results demonstrate that our approach outperforms other privacy-preserving LoRA baselines in model performance while maintaining equivalent privacy guarantees.

The remainder of the paper is structured as follows: Section II covers the preliminaries. Section III introduces the proposed PrivLoRA scheme, followed by a privacy analysis in Section IV. Evaluations are detailed in Section V, and Section VI concludes the paper.

II. Preliminaries

A. Large Language Models

LLMs are advanced deep-learning models with billions to trillions of parameters, designed to understand and generate natural language [18]. They are built on the Transformer architecture, introduced by Vaswani et al. [19] in 2017, which utilizes parallel processing and attention mechanisms to significantly enhance the efficiency of sequential data processing. Transformers have driven the development of models such as GPT [20], BERT [3], and other modern LLMs. LLMs have three key stages: First, they undergo pre-training on vast, diverse datasets, such as internet text, books, and academic corpora, to acquire a broad understanding of language. Second, fine-tuning is performed on smaller, task-specific datasets to adapt to specialized domains. Finally, the inference stage, where the fine-tuned model generates predictions or decisions, such as automated responses or data analysis, effectively leverages its learned knowledge to address real-world problems [21]. This research addresses privacy issues in the fine-tuning stage of LLMs, where the model is adapted to specific tasks using private datasets. These datasets are often sourced from sensitive domains, such as medicine or law, raising critical concerns about protecting confidential information during the fine-tuning process [6], [22], [23].

B. Transformer Architecture

The transformer architecture is a deep-learning model introduced by Vaswani et al. [19] and is widely used in natural language processing tasks. It consists of key components: an input embedding layer that converts tokens into vector representations with positional encodings, an encoder with multi-head self-attention and feed-forward sub-layers to capture relationships between tokens, and a decoder with masked self-attention, multi-head attention, and feed-forward sub-layers for generating contextually appropriate outputs. The final output layer transforms the decoder output into probabilities in the vocabulary, allowing the model to generate the final sequence of tokens. Transformers are well known for their ability to capture global interactions in inputs through self-attention layers [24]–[26]. The self-attention mechanism operates in two steps: first, it computes weight coefficients using the query and key vectors, and second, it applies these weights to the value vectors to produce the output. Recent studies highlight that self-attention layers play a more critical role than other layers, demonstrating greater robustness to distribution shifts and improving model performance on heterogeneous data distributions [27]–[30]. These layers enable the model to evaluate the importance of different parts of the input data, making them highly effective in capturing complex relationships and dependencies, even in diverse and heterogeneous datasets [19], [31]. Inspired by these insights, we hypothesize that fine-tuning self-attention layers before LoRA initialization can effectively address data

heterogeneity in LLMs, and we conduct a detailed empirical analysis to evaluate this hypothesis.

C. Low-Rank Adaptation

LoRA [10] is one of the most popular PEFT methods for pre-trained LLMs. This method is based on the observation that, when adapting pre-trained LLMs to specific tasks, the intrinsic dimensionality of the weight update matrices is much lower than their original dimensions [32]. Consequently, projecting these matrices into a lower-dimensional space does not result in significant information loss. The key idea behind LoRA is to avoid fine-tuning the entire pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, which is computationally expensive and requires substantial GPU memory for LLMs. Instead, LoRA decomposes ΔW into two small matrices $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$, such that:

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

Where $r \ll \min(d, k)$, A and B are initialized with a random Gaussian distribution and zero, respectively, α represents the scaling factor that controls the strength of updates. The number of parameters in LoRA is $r \times (d + k)$, which is significantly smaller than $d \times k$.

III. Proposed PrivLoRA Scheme

A. PrivLoRA Algorithm

LoRA is vulnerable to membership inference attacks [33], where an adversary can infer whether a particular data point was included in the training dataset, potentially leading to privacy leakage. To defend against this, we adopt an honest-but-curious model, ensuring that any shared information between local clients and the server satisfies privacy guarantees via DP [13]. In addition to robust privacy protection, PrivLoRA enhances performance by utilizing layer-wise fine-tuning with SVD-based initialization [14].

In centralized learning, LoRA has demonstrated consistent and promising performance across various tasks, often achieving results comparable to those of FFT [8], [17]. This trend is also observed in FL when the data distribution is homogeneous. However, when data is highly non-IID, LoRA may struggle to achieve the same level of performance as FFT or experience slower convergence [8], [9]. One of the primary factors contributing to this issue is the initialization of the LoRA blocks. According to Hu et al. [10], LoRA initializes the B matrix with zeros and the A matrix with random Gaussian values. While this initialization strategy is effective in centralized settings with large, well-distributed datasets, it may not be suitable for FL, particularly when the data across clients is diverse and heterogeneous.

To address these challenges, we propose PrivLoRA, a privacy-preserving, PEFT approach designed to enhance LoRA for FL. Our method focuses on ensuring privacy protection, improving the initialization process, and enhancing model performance, particularly in non-iid data scenarios, all while reducing communication costs.

Algorithm 1: Overview of the PrivLoRA Algorithm

Input: T: FL rounds, N: Total number of nodes, W: Model weights, SA_Layer: self-attention layers, E: number of epochs, J: Total number of decryption nodes, r: LoRA rank parameter, σ: noise scale, C: clipping norm, η: learning rate.

Output: $[A, B]_t$: Updated model LoRA matrices after t rounds.

//Stage 1:

if $t = 1$ **then**

  for $n = 1$ **to** N **do**

    $\Delta W_n \leftarrow \text{train}(W_0, SA\_Layers, E)$;

    $[\![\Delta W_n]\!] \leftarrow \text{Enc}(pk, \Delta W_n)$;

  $[\![Agg\_\Delta W]\!] \leftarrow \text{aggregate}([\![\Delta W_1, \ldots, \Delta W_N]\!])$;

  for $j = 1$ **to** J **do**

    $[Agg\_\Delta W]_j \leftarrow \text{PartDec}(sk_j, [\![Agg\_\Delta W]\!])$;

  $Agg\_\Delta W \leftarrow \text{FullDec}([Agg\_\Delta W]_1, \ldots, [Agg\_\Delta W]_J)$;

  $W_1 \leftarrow W_0 + Agg\_\Delta W$;

  $[A, B]_1 \leftarrow \text{SVD}(W_1, r)$;

//Stage 2:

else

  for $t = 2$ **to** T **do**

    for $n = 1$ **to** N **do**

      if $t\%2 == 1$ **then**

        $g(B_t^n) \leftarrow \text{trainLoRA}([B]_{t-1}, E)$;

        $g(B_t^n) \leftarrow \frac{g(B_t^n)}{\max(1, \|g(B_t^n)\|_2/C)}$;

        $\tilde{g}(B_t^n) \leftarrow g(B_t^n) + N(0, \sigma^2 C^2 I)$;

        $\tilde{B}_t^n \leftarrow B_t^n - \eta(\tilde{g}(B_t^n))$;

        $\tilde{B}_t \leftarrow \text{aggregate}(\tilde{B}_t^{(1,\ldots,N)})$;

      else

        $g(A_t^n) \leftarrow \text{trainLoRA}([A]_{t-1}, E)$;

        $g(A_t^n) \leftarrow \frac{g(A_t^n)}{\max(1, \|g(A_t^n)\|_2/C)}$;

        $\tilde{g}(A_t^n) \leftarrow g(A_t^n) + N(0, \sigma^2 C^2 I)$;

        $\tilde{A}_t^n \leftarrow A_t^n - \eta(\tilde{g}(A_t^n))$;

        $\tilde{A}_t \leftarrow \text{aggregate}(\tilde{A}_t^{(1,\ldots,N)})$;

In PrivLoRA, we have N nodes, each holding a local dataset D_n with S_n samples, which are never shared with the server or other nodes. The total number of samples across all nodes is denoted as $S = \sum_{i=1}^{N} S_i$. Initially, each node shares the same model weights, which are the pre-trained LLM weights W_0. To ensure privacy, the updates are subject to differential privacy using the DP-SGD algorithm. Privacy guarantees are controlled by parameters such as the noise scale σ, the clipping norm C, the number of iterations T, the learning rate η, and the batch size b.

As outlined in Algorithm 1 and illustrated in Fig. 1, the PrivLoRA algorithm consists of two main stages:

1) Stage 1: Layer-Wise Fine-Tuning with SVD-Based Initialization: This stage combines lightweight layer-wise fine-tuning on the client side with the SVD-based initialization of

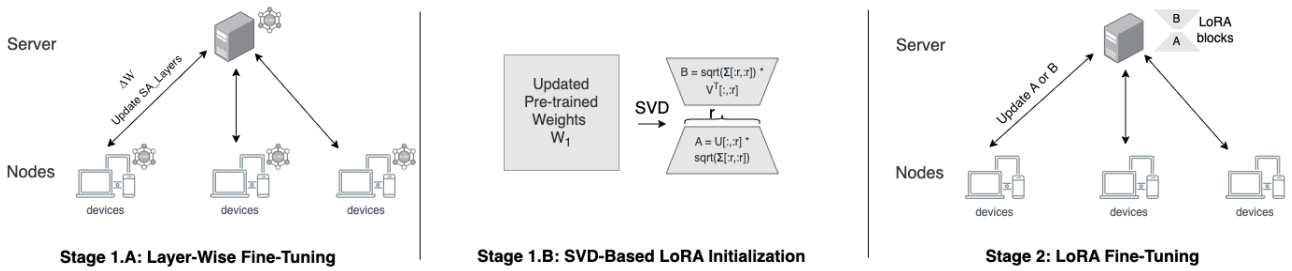

Fig. 1. An overview of the PrivLoRA scheme.

the LoRA matrices A and B on the server side.

- Layer-Wise Fine-Tuning: Each node receives the initial pre-trained LLM weight W_0 from the server and performs layer-wise fine-tuning on a specific layer (such as self-attention layers) for a small number of epochs (e.g., three epochs). After fine-tuning, each node encrypts the weight updates (ΔW) using the PK algorithm, and the encrypted updates are sent back to the server.
- SVD-Based LoRA Initialization: Upon receiving encrypted updates from multiple nodes, the server aggregates the fine-tuned layer updates and sends the encrypted aggregated result to a set of nodes t. Each node computes its decryption share and returns it to the server. Once all decryption shares are received, the server decrypts the aggregated ciphertext and updates the pre-trained model weights with the aggregated result ($W_1 = W_0 + \Delta W$). The server then performs SVD on the adjusted weight matrix W_1, which is used to initialize the LoRA matrices A and B. These matrices are then distributed to the nodes to serve as the starting point for LoRA fine-tuning. The decomposition of W_1 through SVD [34] is represented as:

$$W_1 \xrightarrow{\text{SVD}} U\Sigma V^T. \tag{2}$$

Then the weight matrices A and B are initialized with a rank r as follows:

$$A = \sqrt{\Sigma[:r,:r]} \cdot V^T[:,:r],$$
$$B = U[:,:r] \cdot \sqrt{\Sigma[:r,:r]}.$$

2) Stage 2: LoRA Fine-Tuning: At this stage, nodes fine-tune the model using the initialized matrices A and B, while the global model parameters W_0 remain fixed during the fine-tuning process. Nodes alternate the fine-tuning of matrices A and B in odd and even rounds. The **PrivLoRA algorithm** is as follows:

1) **Broadcast:** The server sends the initialized LoRA weights A and B to all nodes.
2) **Local Training:** Every node n selects S_n samples from its local dataset D_n and uses the clipped gradient to update the local weights A_t^n or B_t^n based on the communication round. In odd communication rounds, all nodes freeze A_t^n and update B_t^n. In even communication rounds, nodes freeze B_t^n and update A_t^n.
3) **Add Noise:** Gaussian noise $e_n \sim \mathcal{N}(0, \sigma^2)$ is added to the weights A_t^n or B_t^n. The noise weights are denoted as:
$$\tilde{A}_t^n = A_t^n + e_n, \quad \tilde{B}_t^n = B_t^n + e_n.$$
4) **Upload:** Each node n uploads its updated weights $\tilde{A}_t^n$ or $\tilde{B}_t^n$ to the server.
5) **Aggregate:** The server aggregates the noised weights $\tilde{A}_t^n$ or $\tilde{B}_t^n$ from all nodes and computes the weighted average as follows:
   - **Odd communication round:**
   $$\frac{1}{N}\sum_{n=1}^{N} \tilde{B}_t^n = \frac{1}{N}(\tilde{B}_t^1 + \tilde{B}_t^2 + \cdots + \tilde{B}_t^N).$$
   The server distributes the aggregated $\tilde{B}_t$ back to all nodes.
   - **Even communication round:**
   $$\frac{1}{N}\sum_{n=1}^{N} \tilde{A}_t^n = \frac{1}{N}(\tilde{A}_t^1 + \tilde{A}_t^2 + \cdots + \tilde{A}_t^N).$$
   The server distributes the aggregated $\tilde{A}_t$ back to all nodes.

IV. Privacy Analysis

A. PrivLoRA Algorithm (Stage 1)

In the first stage, tMK-CKKS [12] provides robust security against colluding and semi-honest nodes and servers, effectively preventing information leakage from ciphertexts. The security of PrivLoRA during this stage relies on the RLWE problem [35], a well-established cryptographic framework known for its resilience, even against quantum computing-based attacks.

B. PrivLoRA Algorithm (Stage 2)

As outlined in stage 2 of Algorithm 1, PrivLoRA satisfies (ϵ, δ)-differential privacy as follows:

The proof follows standard techniques from the differential privacy literature, particularly those in Abadi et al. [13] for moments accountants and the properties of DP. The FL setup involves N nodes, each with a disjoint local dataset D_n, collectively forming the global dataset D. Each client locally updates the LoRA matrices (A and B) using DP-SGD. The DP mechanism is applied during the local training phase using

DP-SGD, with privacy noise scaled by σ, ensuring per-step privacy. For each node n, where q represents the sampling ratio, T is the total number of iterations, ϵ, and δ, the noise scale σ is computed as:

$$\sigma = O\left(\frac{q\sqrt{T\log(1/\delta)}}{\epsilon}\right) \tag{3}$$

The training step in PrivLoRA involves three key steps:

1) **Locally updating trainable LoRA matrices A and B using DP-SGD:** DP-SGD ensures that local updates satisfy (ϵ, δ)-DP by adding Gaussian noise with variance σ. The privacy loss for each step is composed using the moments accountant as described in Abadi et al. [13].
2) **Aggregating updates on the server using FedAvg:** The server averages client updates using FedAvg. This is a post-processing operation, which does not introduce additional privacy loss, as per the post-processing theorem of DP.
3) **Repeating Steps (1) and (2) for T communication rounds:** Over T communication rounds, the privacy losses are composed using the moments accountant as in Abadi et al. [13].

Finally, the moments accountant results are converted into (ϵ, δ)-DP parameters using Theorem 1 from Abadi et al. [13], providing the final guarantee for the entire training process.

V. Evaluation

A. Evaluation Setup

We evaluated the performance of our system on downstream tasks using RoBERTa [36] and LLaMA [37] models. We conducted experiments focusing on natural language understanding (NLU) tasks. We used the GLUE [38] benchmark, specifically on MNLI, SST-2, QNLI, and QQP using RoBERTa-Large (355M), a robust and versatile model widely adopted in research [36]. We used the pre-trained model available in the HuggingFace library [39]. For LLaMA-7B, we evaluated the model on the SST-2 dataset.

All experiments are conducted using NVIDIA A100 GPUs. We perform each experiment with five different random seeds and report the average of the top 3 results. For a cross-silo federated setting, we assume a total of 50 clients, with 10 participants per communication round. The learning rate is set to $\eta = 5e^{-4}$, and client data is randomly split to enforce strong data heterogeneity across clients.

To implement the baseline methods, we utilize FederatedScope-LLM [40], a federated framework designed for fine-tuning LLMs. To ensure fair comparisons, we use a consistent batch size $b = 32$, communication rounds = 500, and 6 local update steps across all experiments. The LoRA adapters are added to both the attention and feed-forward layers with a rank $r = 8$.

For privacy-preserving settings, we use the following parameters: $\delta = 1e^{-5}$, privacy budget $\epsilon = 3$, and clipping threshold $C = 10$. Based on the sampling rate, total number of iterations, and privacy requirement (ϵ, δ), we employ the privacy accountant from Opacus [41] to calculate the noise scale σ for all our experiments.

TABLE I
Accuracy, communication cost (# trainable parameters), and computational cost (training time in min.) of different algorithms for the SST-2 dataset on the LLaMA-7B model.

Systems	Accuracy	Comm. Cost	Comp. Cost
DP-LoRA	85.35	8.40 M	72
FFA-LoRA	86.67	1.60 M	40
PrivLoRA	88.51	1.60 M	47

B. Baselines

Our primary focus is on the accuracy of the final fine-tuned LLM model. We also evaluate the communication cost and training time for each algorithm. Our comparison covers three methods across four GLUE datasets.

DP-LoRA [42]: In each iteration t, the node n fine-tunes and adds noise to A and B using the DP-SGD algorithm and sends the noisy updates $\tilde{A}_t^n$ and $\tilde{B}_t^n$ to the server. The server then aggregates the updates by calculating $\tilde{A}_t = \text{FedAvg}(\tilde{A}_t^n)$ and $\tilde{B}_t = \text{FedAvg}(\tilde{B}_t^n)$.

FFA-LoRA [43]: In each iteration t, node n fine-tunes and adds noise to B while keeping A fixed locally. Consequently, the server only aggregates $\tilde{B}_t^n$ by calculating $\tilde{B}_t = \text{FedAvg}(\tilde{B}_t^n)$.

C. Comparative Evaluation

In this section, we conducted a comprehensive evaluation of PrivLoRA, DP-LoRA, and FFA-LoRA across three key performance metrics: accuracy, communication cost, and computational cost, using the SST-2 dataset on the LLaMA-7B model. The results are summarized in Table I.

As shown in Table I, PrivLoRA consistently outperforms both DP-LoRA and FFA-LoRA, achieving higher accuracy. This can be attributed to Stage 1 of PrivLoRA, which enables more efficient updates and improved convergence in heterogeneous and privacy-constrained environments. In terms of communication efficiency, Table I illustrates the comparative communication costs of each method. PrivLoRA and FFA-LoRA exhibit similar communication overhead, both of which are lower than DP-LoRA. This is because PrivLoRA and FFA-LoRA update only one LoRA matrix per round, resulting in fewer updates per round. On the other hand, DP-LoRA transmits updates for both matrices, leading to higher communication demands. The PrivLoRA approach achieves stronger model performance than both methods while incurring only half the communication costs associated with LoRA.

Table I also provides an analysis of the computational cost of each method. While the computational cost of PrivLoRA is slightly higher than that of FFA-LoRA due to the additional operations in Stage 1, this stage incurs a one-time overhead of 5.24 minutes, primarily attributed to HE and SVD initialization. However, since this stage is executed only once, it has a negligible impact on the overall training efficiency. Moreover, the SVD decomposition in this stage is efficiently handled by

the server. The two-stage structure of PrivLoRA enables more targeted fine-tuning, reducing the number of training iterations needed to reach convergence. In fact, DP-LoRA's dual matrix updates, as well as FFA-LoRA's limited fine-tuning of matrix B, result in higher computational requirements to achieve the same performance as PrivLoRA.

Overall, as summarized in Table I, PrivLoRA outperforms DP-LoRA and FFA-LoRA in terms of accuracy, making it a more suitable choice for privacy-preserving and resource-efficient fine-tuning of LLMs in FL.

D. Ablation Studies and Design Validation

To validate our design choices, we conducted targeted experiments isolating two key components: (a) self-attention versus feed-forward fine-tuning in Stage 1, and (b) the complete two-stage approach versus Stage 2-only fine-tuning. The experiments were performed on the RoBERTa-Large model using the MNLI dataset under a severely non-IID data setting.

TABLE II
ABLATION STUDY: DESIGN CHOICE VALIDATION

Variant	Accuracy
Stage 1: Layer Selection	
Self-Attention FT	73.81%
Feed-Forward FT	67.22%
Stage Configuration	
Full PrivLoRA (Stage 1+2)	73.81%
Stage 2 Only	66.47%

1) Self-Attention vs. Feed-Forward Fine-Tuning: As shown in Table II, fine-tuning self-attention layers in Stage 1 leads to a **6.59%** accuracy improvement over fine-tuning feed-forward layers. This confirms that self-attention layers provide a more effective foundation for SVD initialization in heterogeneous federated environments.

2) Two-Stage vs. Single-Stage Approach: The complete PrivLoRA scheme (Stage 1 + Stage 2) outperforms Stage 2-only fine-tuning by **7.34%**, demonstrating that our layer-wise fine-tuning with SVD initialization significantly improves downstream task performance.

Taken together, these results strongly validate our core design choices: prioritizing self-attention layers for Stage 1 fine-tuning and adopting a two-stage approach that combines robust initialization with efficient LoRA adaptation, leading to clear performance gains.

E. The Effect of Parameters

1) Effect of Data Heterogeneity: Table III presents the performance of the RoBERTa model on the MNLI, SST-2, QNLI, and QQP datasets, highlighting the impact of data heterogeneity across different methods. PrivLoRA consistently outperforms DP-LoRA and FFA-LoRA, achieving higher accuracy in both IID and non-IID scenarios. This improvement can be attributed to the first stage of our algorithm, where a lightweight fine-tuning of the LLM's self-attention layers is performed for a small number of epochs. Then, LoRA is initialized using an SVD technique based on the fine-tuned model. This process enables the model to effectively adapt to diverse client datasets, mitigating the negative effects of heterogeneity and facilitating more efficient convergence.

TABLE III
PERFORMANCE COMPARISON ACROSS DIFFERENT DATA DISTRIBUTIONS

Data Distribution	Method	MNLI	SST-2	QQP	QNLI
IID	DP-LoRA	72.54	87.14	73.83	79.50
	FFA-LoRA	75.13	87.58	74.96	80.64
	PrivLoRA	**77.49**	**89.27**	**75.55**	**82.20**
Mild Het.	DP-LoRA	68.73	82.35	69.93	75.07
	FFA-LoRA	73.18	83.67	70.61	76.23
	PrivLoRA	**75.97**	**88.51**	**75.06**	**81.11**
Severe Het.	DP-LoRA	62.15	78.10	67.40	71.42
	FFA-LoRA	68.65	82.22	69.12	73.37
	PrivLoRA	**73.81**	**87.24**	**74.83**	**77.70**

TABLE IV
PERFORMANCE COMPARISON ACROSS DIFFERENT PRIVACY BUDGETS

Privacy Budget	Method	MNLI	SST-2	QQP	QNLI
Non-Private	DP-LoRA	77.24	88.14	76.83	82.74
	FFA-LoRA	75.46	86.58	73.37	80.38
	PrivLoRA	**78.11**	**90.27**	**76.55**	**84.20**
$\epsilon = 6$	DP-LoRA	73.79	85.42	72.54	77.61
	FFA-LoRA	74.95	84.10	71.12	78.14
	PrivLoRA	**76.93**	**89.72**	**75.82**	**82.38**
$\epsilon = 3$	DP-LoRA	68.73	82.35	69.93	75.07
	FFA-LoRA	73.18	83.67	70.61	76.23
	PrivLoRA	**75.97**	**88.51**	**75.06**	**81.11**
$\epsilon = 1$	DP-LoRA	65.27	79.63	67.80	71.45
	FFA-LoRA	72.43	81.11	69.82	73.28
	PrivLoRA	**74.12**	**87.97**	**74.34**	**78.49**

2) Effect of Different Privacy Budgets: Table IV demonstrates that PrivLoRA outperforms DP-LoRA and FFA-LoRA in all tasks. Unlike DP-LoRA, which applies noise to both LoRA matrices in every round and updates them simultaneously, PrivLoRA utilizes an alternating update mechanism. This allows the model to optimize one matrix at a time, preserving more useful information in each round while minimizing the impact of noise. Similarly, FFA-LoRA exhibits weaker performance because it exclusively fine-tunes matrix B while keeping matrix A fixed throughout all rounds, making it less resilient under stricter privacy constraints.

VI. CONCLUSION AND FUTURE WORK

This paper introduces PrivLoRA, an FL framework addressing the critical challenges of privacy preservation, communication efficiency, and performance degradation in fine-tuning LLMs. By integrating LoRA with novel strategies such as layer-wise fine-tuning, SVD-based initialization, and an alternating update mechanism, PrivLoRA enhances model performance, particularly in non-IID client data settings, while significantly reducing communication costs. Additionally, the incorporation of HE and DP ensures robust protection of sensitive information throughout the training process. Extensive experiments across diverse datasets validate the effectiveness of PrivLoRA, demonstrating superior accuracy and efficiency compared to existing privacy-preserving fine-tuning methods. Future work may explore further extensions of PrivLoRA to optimize computational efficiency for real-world deployment.

REFERENCES

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

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.

[4] S. Narang and A. Chowdhery, "Pathways language model (palm): Scaling to 540 billion parameters for breakthrough performance," *Google AI Blog*, 2022.

[5] S. Kim, S. Yun, H. Lee, M. Gubri, S. Yoon, and S. J. Oh, "Propile: Probing privacy leakage in large language models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 20 750–20 762, 2023.

[6] G. Sebastian, "Privacy and data protection in chatgpt and other ai chatbots: Strategies for securing user information," *International Journal of Security and Privacy in Pervasive Computing*, vol. 15, no. 1, 2023.

[7] N. Ding, Y. Qin, W. Yang *et al.*, "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature Machine Intelligence*, vol. 5, no. 3, pp. 220–235, 2023.

[8] Z. Zhang, Y. Yang, Y. Dai, Q. Wang, Y. Yu, L. Qu, and Z. Xu, "Fedpetuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models," in *Annual Meeting of the Association of Computational Linguistics 2023*. Association for Computational Linguistics (ACL), 2023, pp. 9963–9977.

[9] S. Babakniya, A. R. Elkordy, Y. H. Ezzeldin, Q. Liu, K.-B. Song, M. EL-Khamy, and S. Avestimehr, "Slora: Federated parameter efficient fine-tuning of language models," in *International Workshop on FL in the Age of Foundation Models in Conjunction with NeurIPS*, 2023.

[10] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, vol. 1, no. 2, p. 3, 2022.

[11] Y. Yan, Q. Yang, S. Tang, and Z. Shi, "Federa: Efficient fine-tuning of language models in federated learning leveraging weight decomposition," *arXiv preprint arXiv:2404.18848*, 2024.

[12] W. Du, M. Li, L. Wu, Y. Han, T. Zhou, and X. Yang, "A efficient and robust privacy-preserving framework for cross-device federated learning," *Complex & Intelligent Systems*, vol. 9, pp. 4923–4937, 2023.

[13] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[14] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[15] G. Sun, M. Mendieta, T. Yang, and C. Chen, "Exploring parameter-efficient fine-tuning for improving communication efficiency in federated learning," *arXiv preprint arXiv:2210.01708*, 2022.

[16] Z. Zhang, Y. Yang, Y. Dai, L. Qu, and Z. Xu, "When federated learning meets pre-trained language models' parameter-efficient tuning methods," *arXiv preprint arXiv:2212.10025*, 2022.

[17] Y. Mao, Y. Ge, Y. Fan, W. Xu, Y. Mi, Z. Hu, and Y. Gao, "A survey on lora of large language models," *Frontiers of Computer Science*, vol. 19, no. 7, p. 197605, 2025.

[18] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, vol. 1, no. 2, 2023.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[20] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training." USA, 2018.

[21] B. Yan, K. Li, M. Xu, Y. Dong, Y. Zhang, Z. Ren, and X. Cheng, "On protecting the data privacy of large language models (llms): A survey," *arXiv preprint arXiv:2403.05156*, 2024.

[22] C. Peris, C. Dupuy *et al.*, "Privacy in the time of language models," in *Proceedings of the sixteenth ACM international conference on web search and data mining*, 2023.

[23] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on large language model (llm) security and privacy: The good, the bad, and the ugly," *High-Confidence Computing*, p. 100211, 2024.

[24] D. W. Romero and J.-B. Cordonnier, "Group equivariant stand-alone self-attention for vision," in *International Conference on Learning Representations*, 2020.

[25] Y. Hao, L. Dong, F. Wei, and K. Xu, "Self-attention attribution: Interpreting information interactions inside transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 12 963–12 971.

[26] J. Ji, M. Wang, X. Zhang, M. Lei, and L. Qu, "Relation constraint self-attention for image captioning," *Neurocomputing*, pp. 778–789, 2022.

[27] Y. Sun and H. Ochiai, "Homogeneous learning: Self-attention decentralized deep learning," *IEEE Access*, vol. 10, pp. 7695–7703, 2022.

[28] L. Qu, Y. Zhou, Y. Liang, F. Wang, E. Adeli, L. Fei-Fei, and D. Rubin, "Rethinking architecture design for tackling data heterogeneity in federated learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 061–10 071.

[29] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit, "Understanding robustness of transformers for image classification," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 231–10 241.

[30] Q. Guo, Y. Qi, S. Qi, D. Wu, and Q. Li, "Fedmcsa: Personalized federated learning via model components self-attention," *Neurocomputing*, vol. 560, p. 126831, 2023.

[31] T. Ashraf, F. Bin Afzal Mir, and I. A. Gillani, "Transfed: A way to epitomize focal modulation using transformer-based federated learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 554–563.

[32] A. Aghajanyan, S. Gupta, and L. Zettlemoyer, "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)*, 2021, pp. 7319–7328.

[33] Z. Luo, X. Xu, F. Liu, Y. S. Koh, D. Wang, and J. Zhang, "Privacy-preserving low-rank adaptation for latent diffusion models," *CoRR*, 2024.

[34] F. Meng, Z. Wang, and M. Zhang, "Pissa: Principal singular values and singular vectors adaptation of large language models," *Advances in Neural Information Processing Systems*, pp. 121 038–121 072, 2024.

[35] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2010, pp. 1–23.

[36] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

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

[38] A. Wang, A. Singh *et al.*, "Glue: A multi-task benchmark and analysis platform for natural language understanding," in *International Conference on Learning Representations*, 2018.

[39] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Huggingface's transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.

[40] W. Kuang, B. Qian, Z. Li, D. Chen, D. Gao, X. Pan, Y. Xie, Y. Li, B. Ding, and J. Zhou, "Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 5260–5271.

[41] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao *et al.*, "Opacus: User-friendly differential privacy library in pytorch," in *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021.

[42] X. Liu, R. Zhu *et al.*, "Differentially private low-rank adaptation of large language model using federated learning," *ACM Transactions on Management Information Systems*, vol. 16, no. 2, 2025.

[43] Y. Sun, Z. Li, Y. Li, and B. Ding, "Improving lora in privacy-preserving federated learning," in *The Twelfth International Conference on Learning Representations*, 2024.