DP-FedLoRA: Privacy-Enhanced Federated Fine-Tuning for On-Device Large Language Models

Honghui Xu*, Shiva Shrestha*, Wei Chen†, Zhiyuan Li†, Zhipeng Cai‡
*Department of Information Technology, Kennesaw State University, Marietta, USA
Email: hxu10@kennesaw.edu, sshres16@students.kennesaw.edu
†Nexa AI, Cupertino, USA. Email: {alexchen, zack}@nexa.ai
‡Department of Computer Science, Georgia State University, Atlanta, USA. Email: zcai@gsu.edu

***Abstract*—As on-device large language model (LLM) systems become increasingly prevalent, federated fine-tuning enables advanced language understanding and generation directly on edge devices; however, it also involves processing sensitive, user-specific data, raising significant privacy concerns within the federated learning framework. To address these challenges, we propose DP-FedLoRA, a privacy-enhanced federated fine-tuning framework that integrates LoRA-based adaptation with differential privacy in a communication-efficient setting. Each client locally clips and perturbs its LoRA matrices using Gaussian noise to satisfy (ϵ, δ)-differential privacy. We further provide a theoretical analysis demonstrating the unbiased nature of the updates and deriving bounds on the variance introduced by noise, offering practical guidance for privacy-budget calibration. Experimental results across mainstream benchmarks show that DP-FedLoRA delivers competitive performance while offering strong privacy guarantees, paving the way for scalable and privacy-preserving LLM deployment in on-device environments.**

***Index Terms*—Large Language Model, Differential Privacy, Federated Fine-tuning**

I. Introduction

The increasing deployment of on-device large language models (LLMs) has brought powerful language understanding and generation capabilities directly to edge devices [1], [2]. To adapt these large models to diverse user environments and device-specific tasks, federated fine-tuning has emerged as a popular solution [3], [4]. Federated learning (FL) enables decentralized model adaptation by allowing edge devices to train on their local data while only sharing model updates [5], [6]. Specifically, combined with parameter-efficient fine-tuning (PEFT) techniques, this federated learning framework allows even resource-constrained devices to fine-tune powerful LLMs collaboratively [3], [7].

However, this federated training paradigm introduces serious privacy risks, even though raw data never leaves the device. In particular, membership inference attacks (MIAs) have shown that adversaries, especially a semi-honest central server, can exploit the shared model updates to infer whether specific user data was used in training [8], [9]. These vulnerabilities pose significant threats to user confidentiality, especially when dealing with private text, medical records, or behavioral logs processed by on-device LLMs [10], [11].

To bridge this gap, we propose a privacy-enhanced federated fine-tuning framework designed specifically for on-device LLM ecosystems. Our approach integrates Low-Rank Adaptation (LoRA), a parameter-efficient fine-tuning technique, with differential privacy (DP) in a communication-efficient federated learning setup. We introduce a novel algorithm, DP-FedLoRA, which enables each edge client to locally fine-tune a low-rank adaptation of the global LLM using private data, while preserving privacy through calibrated noise injection and norm clipping. These privacy-enhanced updates are then securely aggregated at a central server using a structured stacking mechanism that reconstructs a global adaptation layer without exposing individual client contributions, which ensures privacy protection against membership inference attacks on-device LLMs.

Beyond practical design, we provide a theoretical analysis of the privacy-utility trade-off introduced by Gaussian noise in our DP-FedLoRA framework. To be specific, we show that the injected noise introduces no bias in expectation but contributes additional variance, which can be bounded analytically in terms of the model size and noise scale. These bounds offer explicit guidelines for tuning the privacy parameters without severely degrading model performance. Finally, we validate our proposed method through comprehensive experiments on real-world LLM benchmarks. Our results demonstrate that DP-FedLoRA delivers performance comparable to existing federated fine-tuning methods, while providing strong privacy guarantees across LLM-enabled edge devices. To sum up, the key contributions of this work are summarized as follows:

- We propose the first privacy-enhanced federated fine-tuning framework for on-device LLM ecosystems.
- We design a noise-injected aggregation mechanism that preserves the LoRA structure and accommodates heterogeneous clients with varying adaptation ranks.
- We provide a theoretical analysis of the expectation and variance of model updates under the Gaussian mechanism in DP, offering practical guidance for balancing privacy guarantees and learning performance.
- Comprehensive experiments are conducted across mainstream benchmarks to validate the effectiveness of our DP-FedLoRA.

The remainder of this paper is organized as follows. We review related work in Section II and present the necessary preliminaries in Section III. The threat model is described in

Section IV, and the implementation details of our proposed DP-FedLoRA framework are provided in Section V. We then present the theoretical analysis in Section VI. Experimental results are discussed in Section VII, and we conclude the paper in Section VIII.

II. Related Work

In this section, we review related work on mainstream parameter-efficient fine-tuning techniques for LLMs, as well as recent advances in federated fine-tuning of LLMs.

A. Parameter-Efficient Fine-tuning of LLMs

LLMs pose significant challenges in fine-tuning due to their immense size and resource requirements. To address this, parameter-efficient fine-tuning (PEFT) methods [12], [13] have emerged as practical alternatives, enabling efficient adaptation with minimal trainable parameters. (i) Early approaches like BitFit [14] focus on tuning only bias terms, offering comparable performance to full fine-tuning with drastically reduced computation. (ii) Adapter tuning inserts small trainable modules between transformer layers, allowing efficient task adaptation without modifying the base model [15], [16]. LoRA (Low-Rank Adaptation) [17], [18] further improves efficiency by decomposing weight updates into the product of two low-rank matrices, enabling fine-tuning with reduced memory overhead. Its flexibility and strong performance have led to widespread adoption and motivated variants such as AdaLoRA, which dynamically adjusts the parameter budget based on training needs. (iii) Other recent efforts explore optimizing LoRA across dimentions such as layer selection [19], initialization [20], and merging strategies [21]. Our work adopts LoRA as the default PEFT method due to its simplicity, strong empirical performance across diverse downstream tasks, and compatibility with distributed training frameworks.

B. Federated Fine-Tuning of LLMs

Federated fine-tuning of LLMs offers a promising solution for preserving data privacy while enabling collaborative learning across decentralized datasets, particularly in on-device LLM systems where heterogeneous edge devices [3], [4]. (i) Early works such as FedIT integrated LoRA-based PEFT into the federated learning framework, demonstrating its practicality [22]. However, its limited support for heterogeneous LoRA configurations restricts scalability in real-world deployments. (ii) Subsequent approaches attempted to address this by zero-padding LoRA modules [23], [24], incurring extra computation and communication overhead, and by separately averaging LoRA's decomposed matrices, which introduced aggregation noise. (iii) Other research has improved LoRA's utility in FL through sparse initialization [25], SVD-based heterogeneity mitigation [26], and communication-efficient optimization [27]. While these methods improve efficiency and generalization, federated training remains vulnerable to privacy threats such as membership inference attacks [8], [9], despite keeping raw data on-device. Integrating differential privacy (DP) into the federated fine-tuning process is therefore a critical direction to provide protection guarantee against such leakage risks.

III. Preliminary

Our proposed DP-FedLoRA framework builds upon the federated LoRA model. In this section, we first present the formulation of federated LoRA as a foundation before introducing our DP-FedLoRA approach.

A. Low-Rank Adaptation (LoRA)

LoRA fine-tunes large pre-trained models by introducing a low-tank decomposition into specific weight matrices without updating the original weights [28]. Let $W \in \mathbf{R}^{m\times n}$ be a pre-trained weight matrix in a neural network layer. Instead of directly updating W, LoRA keeps it frozen and adds a trainable low-rank matrix ΔW, such that the update weight becomes $W' = W + \Delta W$. The low-rank matrix ΔW is parameterized as $\Delta W = BA$, where $B \in \mathbf{R}^{m\times r}$ and $A \in \mathbf{R}^{r\times n}$. This factorization drastically reduces the number of trainable parameters from $m \times r$ to $(rd + rk)$. Finally, the fine-tuned weight W' will be

$$W' = W + BA. \tag{1}$$

In the forward pass, for an input vector $x \in \mathbf{R}^n$, the transformed output becomes $y = Wx + BAx$, where Wx is computed using the frozen base model weights, and BAx is the low-rank adaptation term learned during fine-tuning. This approach maintains the integrity of the pre-trained model while allowing task-specific adaptation with a small number of additional parameters.

B. Federated LoRA

Federated LoRA scheme provides a privacy-aware and communication-efficient mechanism for fine-tuning LLMs [22]. Supposing that in a federated learning framework consisting of K clients, each client holds a private dataset $\mathbf{D}_k$ and collaboratively fine-tunes a shared pre-trained LLM. The base weight matrix $W \in \mathbf{R}^{m\times n}$ in the LLM remains frozen across all clients. Instead of updating W directly, each client $k \in \{1, 2, \cdots, K\}$ learns a low-rank adaptation $\Delta W_k = B_k A_k$, where $B_k \in \mathbf{R}^{m\times r_k}$ and $A_k \in \mathbf{R}^{r_k\times n}$. This approach ensures minimal parameter overhead on resource-constrained edge devices. During local training, each client computes the adapted weight as

$$W'_k = W + B_k A_k, \tag{2}$$

where the matrices B_k and A_k are the parameters trained based on its local data $\mathbf{D}_k$. After a predefined number of local epochs, each client sends its learned low-rank matrices B_k and A_k to central server. The server performs a secure aggregation by using the stacking operation symbolized by $\bigoplus$. Then, we can formalize the aggregation of LoRA modules. The sum of the products of K LoRA module pairs is equivalent to the product of their stacked matrices:

$$\sum_{k=1}^{K} B_k A_k = (B_1 \bigoplus \cdots \bigoplus B_K)(A_1 \bigoplus \cdots \bigoplus A_K), \tag{3}$$

where $(B_1 \bigoplus \cdots \bigoplus B_K)$ indicates each B_k is horizontally stacked to the right one before it, and $(A_1 \bigoplus \cdots \bigoplus A_K)$ indicates that each A_k is vertically stacked below the preceding one. Once aggregated, the central server broadcasts the updated $B \in \mathbf{R}^{m \times \sum_{k=1}^{K} r_k}$ and $A \in \mathbf{R}^{\sum_{k=1}^{K} r_k \times n}$ to all clients, allowing them to update their models as

$$W'_k = W + BA, \tag{4}$$

where $B = (B_1 \bigoplus \cdots \bigoplus B_K)$ and $A = (A_1 \bigoplus \cdots \bigoplus A_K)$. The aforementioned federated learning framework enables each client to adapt the model to its local context while contributing to a globally improved adaptation layer, making it ideal for real-time, on-device LLMs across distributed, resource-limited devices.

IV. Threat Model

In federated learning, although raw data never leaves the client side, the exchanged model updates can still leak sensitive information. One critical threat is the Membership Inference Attack (MIA), where an adversary attempts to infer whether a particular data sample was used in the local training of a participating client. We consider a semi-honest server that follows the federated protocol but may attempt to perform MIA on received updates. Specifically, in the LoRA-based federated setting, each client $k \in \{1, \ldots, K\}$ computes a low-rank adaptation $\Delta W_k = B_k A_k$, where $B_k \in \mathbb{R}^{m \times r_k}$ and $A_k \in \mathbb{R}^{r_k \times n}$, based on its private dataset D_k. These matrices are transmitted to a central server for aggregation. In MIA scenario, let D_k and D'_k be two neighboring datasets differing in at most one sample. The goal of the attacker is to distinguish whether the received update ΔW_k was generated using D_k or D'_k. Formally, this can be described as distinguishing between the outputs of two mechanisms:

$$\mathcal{M}(D_k) = \Delta W_k = B_k A_k; \quad \mathcal{M}(D'_k) = \Delta W'_k = B'_k A'_k. \tag{5}$$

An effective MIA [29], [30] implies that the attacker can exploit differences in ΔW_k to infer the presence of specific data points, which violates data confidentiality and causes data privacy leakage.

V. DP-FedLoRA

In this work, to enhance privacy in federated LoRA and defend against membership inference attacks, we incorporate differential privacy into the training process, making it difficult for MIA to distinguish between the outputs of the two mechanisms defined in Eq. 5. Each client $k \in \{1, 2, \cdots, K\}$ holds local data $\mathbf{D}_k$ and independently fine-tunes the low-rank adaptation matrices $B_k \in \mathbf{R}^{m \times r_k}$ and $A_k \in \mathbf{R}^{r_k \times n}$, while keeping the base model weight matrix $W \in \mathbf{R}^{m \times n}$ frozen. After local training, each client obtains an update $\Delta W_k = B_k A_k$ which is not shared directly to preserve privacy. Instead, we ensure that the shared updates satisfy (ϵ, δ)-differential privacy through noise perturbation. Before transmitting the local LoRA matrices to the server, each client clips their updates to control sensitivity and then adds Gaussian noise. Specifically, for each client k, the matrices B_k and A_k are first individually clipped such that:

$$||B_k||_F \leq C_{B_k}, ||A_k||_F \leq C_{A_k}, \tag{6}$$

where $|| \cdot ||_F$ denotes the Frobenius norm, and C_{B_k}, C_{A_k} are predetermined clipping thresholds. After clipping, noise sample from isotropic Gaussian distributions is added to each matrix:

$$\tilde{B_k} = B_k + \mathcal{N}(0, \sigma^2_{B_k}\mathbf{I}); \tag{7}$$

$$\tilde{A_k} = A_k + \mathcal{N}(0, \sigma^2_{A_k}\mathbf{I}); \tag{8}$$

where $\sigma^2_{B_k}$ and $\sigma^2_{A_k}$ are noise variances calibrated to the desired privacy budget (ϵ, δ) and $\mathbf{I}$ denotes the identity matrix.

To be specific, the differential privacy guarantee is characterized by privacy budget (ϵ, δ). For the Gaussian mechanism, the standard result states that a mechanism with l_2-sensitivity S and Gaussian noise of standard deviation σ satisfies (ϵ, δ)-DP if

$$\sigma \geq S \cdot \sqrt{2\log(1.25/\delta)}/\epsilon. \tag{9}$$

In our case, the sensitivity of each matrix is bounded by the Frobenius norm due to clipping: $S_{B_k} = C_{B_k}$ and $S_{A_k} = C_{A_k}$. Therefore, the noise levels must

$$\sigma_{B_k} \geq C_{B_k} \cdot \sqrt{2\log(1.25/\delta)}/\epsilon_{B_k}; \tag{10}$$

$$\sigma_{A_k} \geq C_{A_k} \cdot \sqrt{2\log(1.25/\delta)}/\epsilon_{A_k}; \tag{11}$$

where ϵ_{B_k} and ϵ_{A_k} are the target privacy budgets for $\tilde{B_k}$ and $\tilde{A_k}$, respectively.

These perturbed matrices $\tilde{B_k}$ and $\tilde{A_k}$ are then sent to the server for aggregation. Instead of directly summing the client updates as in traditional federated averaging, we apply a structured stacking operation to combine the low-rank matrices across clients. More specifically, each client sends its clipped and noise-perturbed matrices $\tilde{B_k} \in \mathbf{R}^{m \times r_k}$ and $\tilde{A_k} \in \mathbf{R}^{r_k \times n}$ to the central server. The server then constructs the global low-rank adaptation matrices $\tilde{B} \in \mathbf{R}^{m \times r}$ and $\tilde{A} \in \mathbf{R}^{r \times n}$, where $r = \sum_{k=1}^{K} r_i$, by horizontally stacking the $\tilde{B_k}$s and vertically stacking the $\tilde{A_k}$s as follows:

$$\tilde{B} = (\tilde{B_1} \bigoplus \cdots \bigoplus \tilde{B_K}); \tag{12}$$

$$\tilde{A} = (\tilde{A_1} \bigoplus \cdots \bigoplus \tilde{A_K}). \tag{13}$$

This aggregation strategy preserves the low-rank structure of each client's update while allowing flexible integration of variable-rank adaptations across clients. The resulting global update $\tilde{B}\tilde{A}$ represents a unified adaptation module that encodes contributions from all clients while maintaining privacy through injected noise. This update is broadcast back to each client, allowing them to locally reconstruct the global fine-tuned model as:

$$W'_k = W + \tilde{B}\tilde{A}. \tag{14}$$

By structuring the aggregation in this way, our proposed DP-FedLoRA maintains model compatibility across clients while enabling scalable, privacy-preserving federated learning. The pseudocode of the DP-FedLoRA algorithm is described in Algorithm 1.

Algorithm 1 DP-FedLoRA: Differentially Private Federated LoRA Fine-Tuning

```
Require: Pre-trained base model weight $W \in \mathbb{R}^{m\times n}$, number of clients $K$, noise scales $\sigma_{B_k}, \sigma_{A_k}$, clipping thresholds $C_{B_k}, C_{A_k}$, local data $\{\mathbf{D}_k\}_{k=1}^{K}$
Ensure: Global LoRA update $W' = W + \tilde{B}\tilde{A}$
  for each client $k = 1$ to $K$ in parallel do
      Initialize $B_k \in \mathbb{R}^{m\times r_k}$, $A_k \in \mathbb{R}^{r_k \times n}$
      Train $B_k$, $A_k$ on local data $\mathbf{D}_k$ with $W$ frozen
      Clip: $B_k \leftarrow B_k \cdot \min(1, \frac{C_{B_k}}{\|B_k\|_F})$
      Clip: $A_k \leftarrow A_k \cdot \min(1, \frac{C_{A_k}}{\|A_k\|_F})$
      Add Gaussian noise: $\tilde{B_k} = B_k + \mathcal{N}(0, \sigma_{B_k}^2 \mathbf{I})$
      Add Gaussian noise: $\tilde{A_k} = A_k + \mathcal{N}(0, \sigma_{A_k}^2 \mathbf{I})$
      Send $\tilde{B_k}$ and $\tilde{A_k}$ to server
  end for
  Server aggregates:
    $\tilde{B} \leftarrow (\tilde{B_1} \bigoplus \cdots \bigoplus \tilde{B_K})$      ▷ Horizontal stacking
    $\tilde{A} \leftarrow (\tilde{A_1} \bigoplus \cdots \bigoplus \tilde{A_K})$      ▷ Vertical stacking
  Broadcast $\tilde{B}$ and $\tilde{A}$ to all clients
  for each client $k = 1$ to $K$ do
      Update local model: $W_k' = W + \tilde{B}\tilde{A}$
  end for
```

VI. Analysis of Noise in DP-FedLoRA

In this section, we present an expectation and variance analysis of the impact of Gaussian noise on LoRA-based federated fine-tuning within the DP-FedLoRA framework, providing practical guidance for privacy-budget calibration.

A. Expectation Analysis of Gaussian Noise in DP

For simplicity, in DP-FedLoRA, we can treat $\tilde{B} = B + \beta$ and $\tilde{A} = A + \alpha$, where $B \in \mathbf{R}^{m\times r}$ and $A \in \mathbf{R}^{r\times n}$ are the clean low-rank components obtained from local fine-tuning, and $\beta \in \mathbf{R}^{m\times r}$, $\alpha \in \mathbf{R}^{r\times n}$ are Gaussian noise matrices sampled from isotropic distributions:

$$\beta \sim \mathcal{N}(0, \sigma_\beta^2 \mathbf{I}), \alpha \sim \mathcal{N}(0, \sigma_\alpha^2 \mathbf{I}), \tag{15}$$

which are general notations for the additive noise in the aggregated LoRA matrices at the server side after receiving and stacking the clients' noise-injected updates. Assume that β and α are mutually independent and also independent of B and A. Then we investigate the expectation difference between the noise-injected low-rank matrix product $\tilde{B}\tilde{A}$ and the original product BA in the following.

To understand how the injected noise influences the fine-tuning process, we analyze the expectation of the difference between the noise-perturbed and original updates. Specifically, we aim to compute $\mathbb{E}[\tilde{B}\tilde{A}] - \mathbb{E}[BA]$, which becomes $\mathbb{E}[(B+\beta)(A+\alpha)] - \mathbb{E}[BA]$ after substituting the perturbed matrices.

Expanding the product yields $(B+\beta)(A+\alpha) = BA + B\alpha + \beta A + \beta\alpha$. Taking the expectation of both sides, we have:

$$\mathbb{E}[(B+\beta)(A+\alpha)] = \mathbb{E}[BA] + \mathbb{E}[B\alpha] + \mathbb{E}[\beta A] + \mathbb{E}[\beta\alpha]. \tag{16}$$

Given that both α and β are independent zero-mean Gaussian noise matrices and are independent of the model parameters B and A, the cross terms vanish. Specifically,

$$\mathbb{E}[B\alpha] = B \cdot \mathbb{E}[\alpha] = 0; \tag{17}$$

$$\mathbb{E}[\beta A] = \mathbb{E}[\beta] \cdot A = 0; \tag{18}$$

$$\mathbb{E}[\beta\alpha] = 0, \tag{19}$$

due to the independence and zero-mean properties. As a result, we conclude that $\mathbb{E}[\tilde{B}\tilde{A}] = \mathbb{E}[BA]$, and hence the expected update difference is zero:

$$\mathbb{E}[\tilde{B}\tilde{A}] - \mathbb{E}[BA] = 0. \tag{20}$$

This analysis shows that the noise-injected update remains unbiased in expectation. Although individual updates may deviate due to the stochastic noise, the average behavior of the aggregated updates aligns with the original non-private adaptation. Thus, the differential privacy mechanism in DP-FedLoRA does not introduce bias into the global model updates, maintaining the performance of federated fine-tuning.

B. Variance Analysis of Gaussian Noise in DP

While the expectation of the noise-injected low-rank adaptation $\tilde{B}\tilde{A}$ remains unbiased with respect to the original adaptation BA, the introduction of Gaussian noise inevitably increases the variance in the learned model update. To understand this impact, we analyze the total variance introduced by the noise. We start by expanding the noise-injected adaptation as:

$$\mathrm{Var}[\tilde{B}\tilde{A}] = \mathrm{Var}[BA + B\alpha + \beta A + \beta\alpha]. \tag{21}$$

Assuming that the noise terms α and β are independent, zero-mean Gaussian matrices and are also independent of A and B, the cross-covariance terms in the variance expansion vanish. This simplifies the total variance to the sum of variances of individual components:

$$\mathrm{Var}[\tilde{B}\tilde{A}] = \mathrm{Var}[B\alpha] + \mathrm{Var}[\beta A] + \mathrm{Var}[\beta\alpha]. \tag{22}$$

First, we consider $\mathrm{Var}[B\alpha]$. Since $\alpha \sim \mathcal{N}(0, \sigma_\alpha^2 \mathbf{I})$ and B is fixed, the resulting variance is linearly proportional to the Frobenius norm of B:

$$\mathrm{Var}[B\alpha] \leq m\sigma_\alpha^2 \cdot \|B\|_F^2. \tag{23}$$

Next, for $\mathrm{Var}[\beta A]$, since $\beta \sim \mathcal{N}(0, \sigma_\beta^2 \mathbf{I})$ and A is fixed, the resulting variance is similarly:

$$\mathrm{Var}[\beta A] \leq n\sigma_\beta^2 \cdot \|A\|_F^2. \tag{24}$$

Lastly, for the composite term $\mathrm{Var}[\beta\alpha]$, the product of two independent Gaussian matrices yields a variance that depends on both noise scales and the shared inner dimension r of the low-rank decomposition:

$$\mathrm{Var}[\beta\alpha] \leq \sigma_\beta^2 \sigma_\alpha^2 \cdot mnr. \tag{25}$$

By combining these three terms, we obtain a total bound for the variance of the noise-injected update:

$$\mathrm{Var}[\tilde{B}\tilde{A}] \leq m\sigma_\alpha^2 \cdot \|B\|_F^2 + n\sigma_\beta^2 \cdot \|A\|_F^2 + \sigma_\beta^2 \sigma_\alpha^2 \cdot mnr. \tag{26}$$

This bound highlights that the noise level σ must be carefully chosen in relation to the Frobenius norms of the adaptation matrices B and A, as well as the overall model dimensions, to avoid excessive update variance. Additionally, the compound error term $\sigma_\beta^2\sigma_\alpha^2 \cdot mnr$ reveals how the structure of LoRA amplifies variance in high-rank and high-dimension models.

Based on the above analysis of updates in DP-FedLoRA, we can understand how the model's loss is affected by using noisy updates instead of clean ones. Since the added noise is zero-mean and independent, it does not alter the expected output, keeping the average prediction unbiased. However, it introduces additional fluctuations, increasing the variance in model performance. This variance is influenced by the noise scale, model size, and the structure of the low-rank adaptation. Larger models or higher noise levels result in greater prediction variability.

VII. Experiment

In this section, we first describe our experimental setup and then present comprehensive results to demonstrate the effectiveness of our proposed DP-FedLoRA framework, along with key findings derived from its evaluation. The code for all our experiments is available at: https://github.com/ahahnut/DP-FedLoRA.

A. Experiment Settings

We describe our experimental settings from three aspects: datasets, baseline methods, and training details.

1) Datasets: We use the Alpaca-GPT-4 dataset for training, which is generated using GPT-4 via the Self-Instruct framework. During training, we simulate a federated learning environment with 20 clients and randomly sample 2 clients per round. These selected clients collectively hold a total of 20,000 data samples. For evaluation, we consider close-ended benchmarks only. The close-ended benchmarks include MMLU (knowledge) [31], BBH (reasoning) [32], and CRASS (counterfactual reasoning) [33].

2) Baselines: To gain deeper insights into the performance of existing federated learning (FL) baselines in the context of LLMs and to establish a more comprehensive evaluation framework, we implement seven representative FL algorithms. Specifically, we integrate FedAvg [34], FedProx [35], SCAFFOLD [36], FedAvgM [37], FedAdagrad [38], FedYogi [39], and FedAdam [40]. Among these, FedProx and SCAFFOLD are designed to address data heterogeneity by incorporating local model correction mechanisms. In contrast, FedAvgM, FedAdagrad, FedYogi, and FedAdam introduce server-side momentum or adaptive optimization to stabilize global model updates.

3) Default Training Details: For setting a consistent baseline across different experimental settings, a quantized LLaMA-2-7B model is used to boost memory and computation efficiency. The federated fine-tuning process is carried out over 200 rounds of communication using a single NVIDIA A100 GPU. Every participant client of this effort undergoes 10 rounds of local updates per communication round, using the AdamW optimizer. A cosine learning rate schedule is followed through rounds, along with a linear decay of the learning rate from 5×10^{-5} to 1×10^{-6}. The maximum length of the sequence of inputs is limited to 512 tokens, while each client has a local batch size of 16. To enable parameter-efficient fine-tuning, Low-Rank Adaptation (LoRA) is used with a rank of 32 and scaling factor $\alpha = 64$.

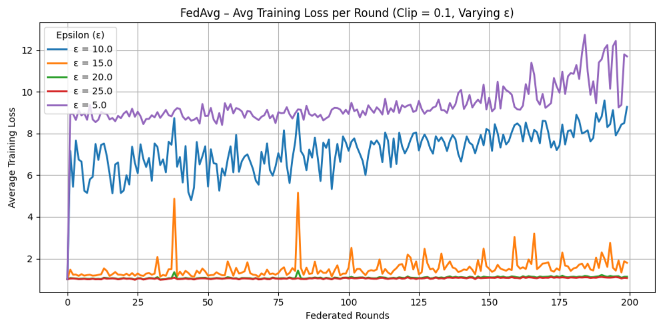
B. Ablation Study

We performed the ablation study only on the FedAvg algorithm, sweeping the differential privacy parameter ε and the clip norm which is illustrated in Fig.1. This choice was motivated by the fact that similar phenomena emerged for other optimization algorithms and hence the choice of FedAvg was representative enough to analyze. The main goal of this study was to analyze how differential privacy hyperparameters affect model convergence and overall model stability during training.

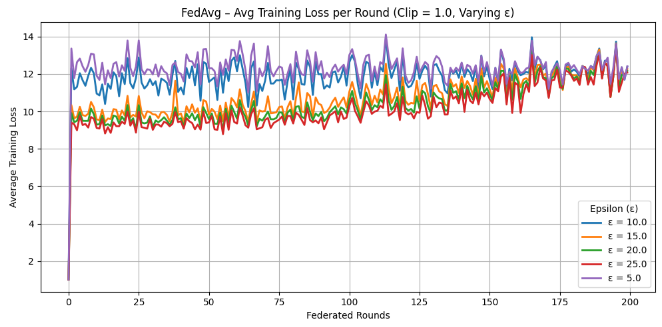
As shown in Fig.1a, when the value of ε varied while keeping a minimum and fixed clipping norm of 0.1, an increment in values of ε consistently produced improved convergence behavior, which is reflected by lower average losses throughout all iterations of training. Among all settings tested, an ε value of 25.0 resulted in the most stable performance curve with the lowest average loss curve, indicating that stricter privacy constraints (i.e., a larger value of ε) help maintain more relevant information from gradients even when there is greater differential privacy noise. On the contrary, using stricter privacy parameters (e.g., ε= 5.0 or 10.0) resulted in significantly noisier and less stable performance measures, thus proving the trade-off between intrinsic privacy and model effectiveness. Besides, Fig. 1b shows the same comparison for clip norm 1.0, which exhibits a larger average training loss across all values of ε when compared to the configuration with a clip norm of 0.1.

Furthermore, we also sought to further test the impact of clipping's norm gradient by keeping ε to a fixed value of 15.0 and varying clip levels from 0.1 to 1.0, as shown in Fig. 1c. Results indicated a significant reduction in performance to correlate with rising clipping norms, together with a parallel increase in loss values for all test conditions; namely, the lower norm—clip = 0.1 had improved regulated gradients and better overall convergence. As illustrated in Fig. 1d, a careful examination between ε = 15.0 and ε = 25.0 for clip = 0.1 substantiated these findings: not only did the higher setting of ε produce a lower final training loss, but it showed fewer and less frequent instabilities throughout training for lower ε.

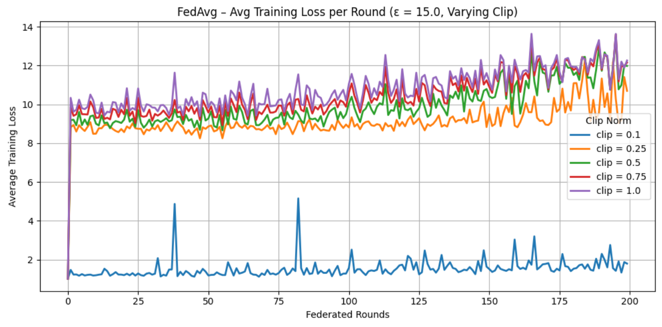
To sum up, the above results indicate that our setting of ε = 25.0 and clip = 0.1 provides the best balance between privacy and performance under the conditions of our experimental model. Due to its improved convergence behavior and continued lowering of loss measures, we adopted this setting as the default for all subsequent differential privacy experiments under our investigation.



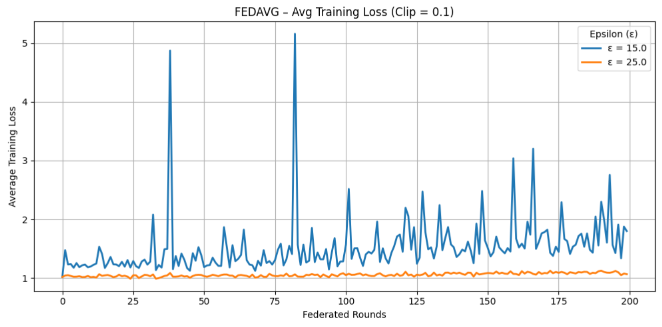
(a) Varying ε, Clip = 0.1



(b) Varying ε, Clip = 1.0

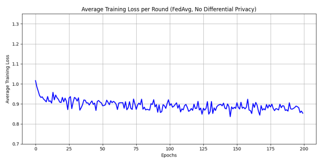


(c) Varying Clip Norm, $\epsilon = 15.0$

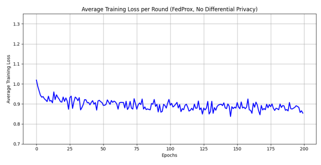


(d) $\epsilon = 15.0$ vs $\epsilon = 25.0$ (Clip = 0.1)

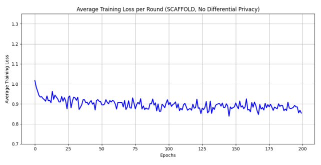
Fig. 1: Results of Ablation Study



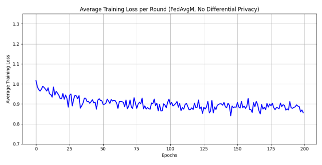
(a) FedAvg



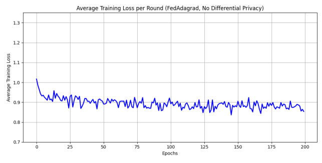
(b) FedProx



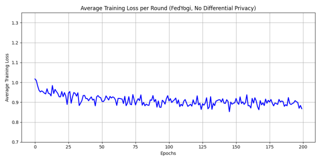
(c) SCAFFOLD



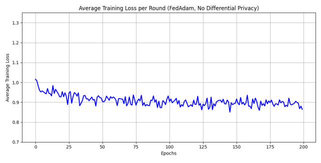
(d) FedAvgM



(e) FedAdagrad

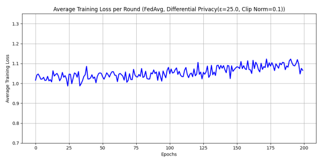


(f) FedYogi

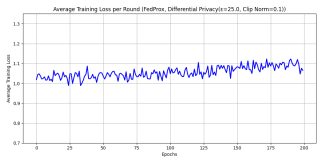


(g) FedAdam

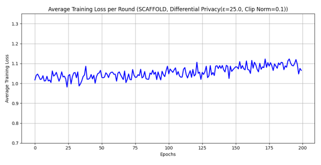
Fig. 2: Training Loss Curves of FedLLMs without Differential Privacy



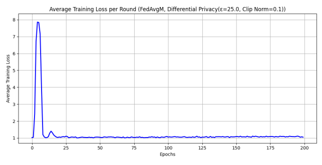
(a) FedAvg



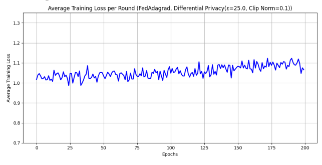
(b) FedProx



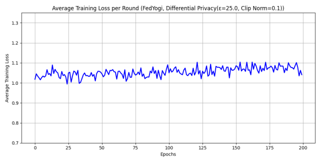
(c) SCAFFOLD



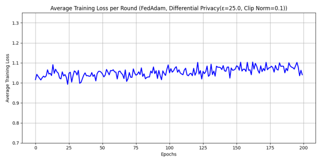
(d) FedAvgM



(e) FedAdagrad



(f) FedYogi



(g) FedAdam

Fig. 3: Training Loss Curves of DP-FedLLMs with Our Proposed DP-FedLoRA ($\epsilon = 25.0$, Clip = 0.1)

C. Performance Evaluation: FedLLM vs. DPFedLLM

To assess the effectiveness of our DPFedLoRA framework, we compare the training loss and downstream close-end benchmark accuracy of models trained with and without differential privacy for seven federated optimization algorithms: FedAvg, FedProx, SCAFFOLD, FedAvgM, FedAdagrad, FedYogi, and FedAdam. The differential privacy configuration is $\epsilon = 25.0$ with a clipping norm of 0.1 for all experiments.

1) Training Loss Curves: Table I presents the average training loss after 200 communication rounds for both FedLLM and our **DPFedLLM**, which are trained using the proposed DP-FedLoRA framework, across all federated learning algorithms. As expected, differential privacy leads to an increase in the mean training loss due to the additive noise and clipping of the gradients, which caps the learning potential. However, the reduction in performance is relatively moderate for all of the algorithms, especially in SCAFFOLD, indicating their robustness under privacy constraints. Moreover, the training loss curves in Fig. 2 and 3 are used to further illustrate this trend, showing the progression of loss across 200 epochs without and with differential privacy, respectively. Due to gradient clipping and added Gaussian noise, the DP-FedLoRA models exhibit slower convergence and higher loss values overall.

As seen in Fig. 3d, when operating under the differential privacy setting, FedAvgM shows an elevated training loss in the first rounds—up to about 8.0—before converging sharply to about 1.0. Such an effect is not seen in other methods. Such an anomaly can be explained by the combination of the effect of momentum and the added differential privacy noise, coupled with aggressive clipping (clip norm = 0.1). Since prior gradients are accumulated by FedAvgM, the effect of noisy gradients in the first rounds can be compounded, causing divergence-like behavior. Even though this spike does not lead to divergence, the model eventually converges. However, this effect suggests that FedAvgM is possibly more sensitive to privacy parameters compared to other methods.

TABLE I: Training Loss Comparison of FedLLM and DPFedLLM after 200 Communication Rounds (ε = 25.0, Clip = 0.1)

Algorithm	**FedLLM Loss**	**DPFedLLM Loss**
FedAvg	0.8362	0.9868
FedProx	0.8370	0.9890
SCAFFOLD	0.8384	0.9815
FedAvgM	0.8400	1.0003
FedAdagrad	0.8364	0.9866
FedYogi	0.8524	0.9942
FedAdam	0.8504	0.9929

2) Evaluation on Close-ended Benchmarks: Table II summarizes the performance results, in terms of test scores, across different federated learning methods in both non-private (Non-DP) and differentially private (DP) (*i.e.*, our proposed DP-FedLoRA framework) setups over three prominent close-end benchmarking tasks: MMLU, BBH, and CRASS. These benchmarking tasks aim to measure a wide range of skills, from factual knowledge to reasoning abilities, and commonsense reasoning. Differential privacy seems to have little effect on model efficacy, as indicated by the relatively small decreases in MMLU and BBH performance, with average score reductions of approximately 4–5% across all algorithms. For example, the MMLU value for FedAvg goes from 44.64% (Non-DP) to 42.45% (DP), while that for BBH drops from 38.96% to 38.52%. this implies that private training methodologies can be used without affecting the model in scenarios that require reasoning knowledge.

In contrast, a more prominent reduction in performance is reported on the CRASS benchmark when differential privacy (DP) is introduced, suggesting that commonsense reasoning tasks are especially vulnerable to interference introduced by differential privacy. Still, this reduction is within a range (i.e., from 40.90% to 25.00%, i.e., FedProx), which suggests that differentially private federated learning should be improved in the counterfactual reasoning LLMs. To conclude, our findings show our proposed DP-FedLoRA can be an effective method to fine-tune LLMs while protecting data privacy.

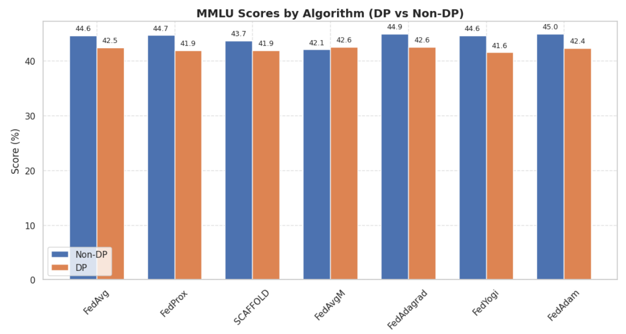
TABLE II: Evaluation Scores of Federated Learning Algorithms with and without Differential Privacy(DP)

Algorithm	**Privacy**	**MMLU**	**BBH**	**CRASS**
FedAvg	Non-DP	44.64	38.96	38.09
FedAvg	DP	42.45	38.52	22.73
FedProx	Non-DP	44.70	38.79	40.90
FedProx	DP	41.92	37.33	25.00
SCAFFOLD	Non-DP	43.68	39.88	29.54
SCAFFOLD	DP	41.92	37.70	25.00
FedAvgM	Non-DP	42.07	38.29	38.64
FedAvgM	DP	42.57	37.48	20.45
FedAdagrad	Non-DP	44.94	39.67	31.82
FedAdagrad	DP	42.58	37.07	22.28
FedYogi	Non-DP	44.63	38.52	34.09
FedYogi	DP	41.57	37.70	22.73
FedAdam	Non-DP	44.95	38.63	36.36
FedAdam	DP	42.35	38.59	20.45

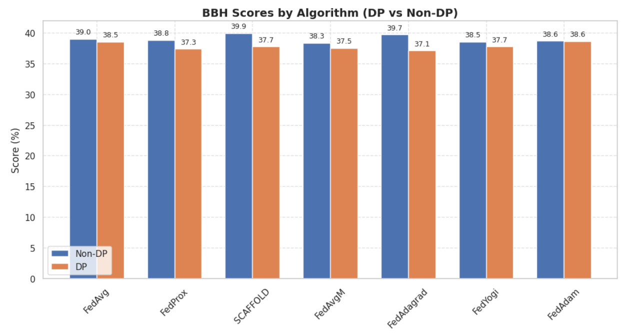
D. Impact of Rank in DP-FedLoRA

To investigate how the Low-Rank Adaptation (LoRA) rank affects the behavior of noise introduced in our proposed DP-FedLoRA framework, experiments were performed with FedAvg as the base aggregation algorithm with a LLaMA2-7B model, having the privacy budget ε constant at 25.0, the clipping norm being 0.1, and the LoRA alpha α being 128. The only tunable hyperparameter was the LoRA rank, tested at the values $\{8, 16, 32, 64, 128\}$ using the aggregation method of FedAvg. During these experimental processes, statistical properties of the noise-influenced updates were tracked, with a specific focus on their expectation and variance across the communication rounds.

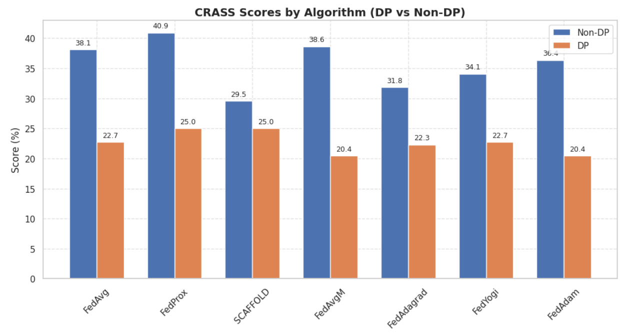
In Fig. 5, our experimental results show that the expectation difference ($\mathbb{E}[\tilde{\Delta}]$) between the noise-injected and original updates remained near zero throughout the training process across all LoRA ranks as indicated in the Eq.(20), thus establishing the evidence that the differential privacy mechanism introduced unbiased noise, which is independent



(a) MMLU Score

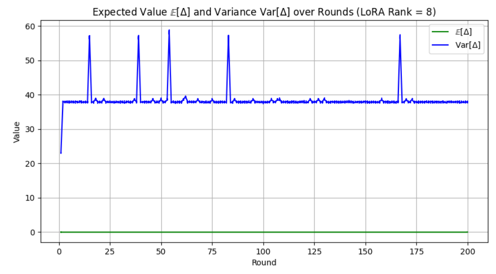


(b) BBH Score

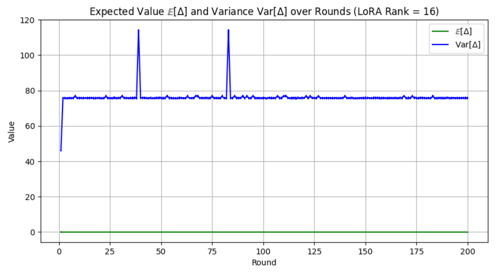


(c) CRASS Score

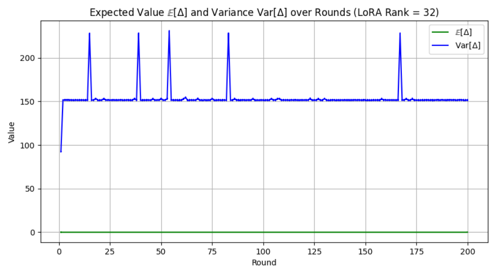
Fig. 4: Performance scores of FedLLMs and DPFedLLMs for MMLU, BBH, and CRASS benchmarks



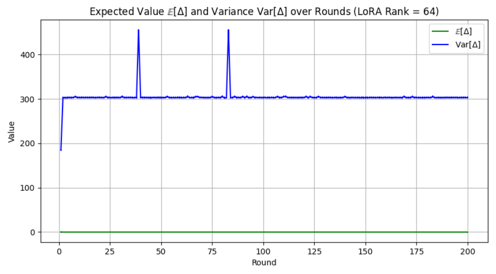
(a) Rank 8



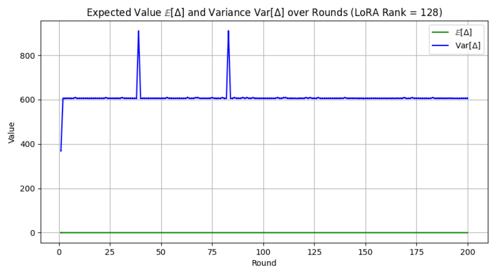
(b) Rank 16



(c) Rank 32



(d) Rank 64



(e) Rank 128

Fig. 5: Expectation Value $\mathbb{E}[\tilde{\Delta}]$ and Variance $\mathrm{Var}[\tilde{\Delta}]$ of Noisy updates in DP-FedLoRA with Various Ranks

of LoRA configuration. This also confirms that average update behaviors are consistent with those derived for non-private learning, thereby preserving the correctness of the federated optimization trajectory in the limit.

Besides, as illustrated in Fig. 5, the variance ($\mathrm{Var}[\tilde{\Delta}]$) of the updates showed an upward trend corresponding to the rank of LoRA. To be specific, the variances observed were around 37 at rank 8, 75 at rank 16, 150 at rank 32, 300 at rank 64, and 600 at rank 128. This regular monotonic increase in variance is consistent with the previously derived theoretical upper bound, particularly the term $\sigma_\beta^2\sigma_\alpha^2 \cdot mnr$ from Eq.(26), showing linear growth with the inner rank r. Therefore, higher rank adaptations naturally increase the scale of the noise, regardless of the noise scale and clipping parameter constancy.

Notably, while there is a common trend of variance growth with respect to LoRA rank, variations and occasional spikes in variance in Fig. 5 were also identified throughout training across all ranks. Such spikes are identified as nonsystematic; that is, they can be caused by differences in the gradient distributions between clients, particularly when the participating clients have heterogeneous data or model states. In federated learning setups, such non-IID scenarios where clients hold diverse and potentially skewed data distributions, can easily cause occasional spikes in both sensitivity and variability of norms, which in turn affects the amount of noise injected in each iteration even in cases where clipping mechanisms are used. Thus, such spikes are viewed as a natural effect of the dynamic, decentralized, and data-heterogeneous nature of federated optimization in differential privacy settings.

From Table III, we can also conclude that while our proposed DP-FedLoRA framework promises unbiased updates, an increase in the adaptation rank significantly raises the noise variance. This trade-off between expressiveness and stability requires careful adjustment to maintain an efficient balance between privacy and utility in practical applications.

TABLE III: Converged Expectation Value $\mathbb{E}[\tilde{\Delta}]$ and Variance $\mathrm{Var}[\tilde{\Delta}]$ of Noisy updates in DP-FedLoRA with Various Ranks

LoRA Rank	**Expectation** ($\mathbb{E}[\tilde{\Delta}]$)	**Variance** ($\mathrm{Var}[\tilde{\Delta}]$)
8	-2.38×10^{-8}	37.88
16	-4.27×10^{-9}	75.75
32	1.04×10^{-8}	151.47
64	1.26×10^{-8}	302.92
128	-1.36×10^{-8}	605.82

E. Impact of Parameter Size in DP-FedLoRA

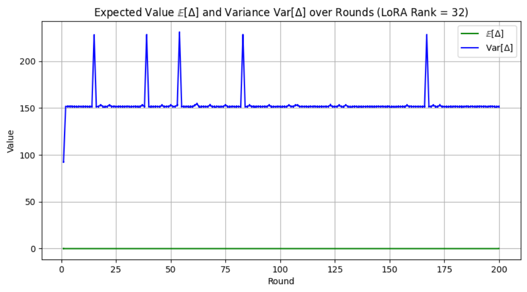
To evaluate the effect of model size on the statistical properties of noise during federated fine-tuning, we performed a comparative study of both the LLaMA-2-7B and LLaMA-2-13B under the same training setup. This setup used the FedAvg aggregation method, had a constant LoRA rank of 32, a privacy budget of $\varepsilon = 25.0$, a clipping norm of 0.1, and a LoRA scaling factor $\alpha = 128$. All other settings remained unchanged.

As shown in Fig.6, $\mathbb{E}[\tilde{\Delta}]$ values in all communication rounds indicates the addition of noise is unbiased regardless of the model size. However, the variance represented as $\mathrm{Var}[\tilde{\Delta}]$ shows a considerable difference. The LLaMA-2-13B model always shows a higher variance than the 7B model. This increase is consistent with the theoretical bound concerning the matrix dimensions m and n in the term $\sigma_\beta^2 \sigma_\alpha^2 \cdot mnr$, which argues that larger models inherently increase the noise variance while keeping the same differential privacy mechanism.

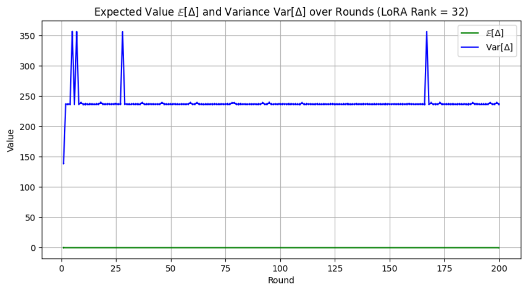
Similarly, as shown in Table IV, we observe empirical results that confirm our theoretical findings in Section VI: the expectation remains invariant to model size, while the variance increases with larger model sizes.

VIII. Conclusion

In this paper, we proposed DP-FedLoRA, a privacy-enhanced federated fine-tuning framework for on-device LLMs deployed on edge devices. Our approach combines LoRA-based parameter-efficient adaptation with differential privacy to safeguard sensitive local data while preserving the performance of federated LLMs. To be specific, we introduced a structured noise injection and aggregation mechanism that enforces differential privacy on client updates and supports heterogeneous adaptation ranks. Additionally, we provided a theoretical analysis demonstrating the unbiased nature and bounded variance of noise-injected updates, offering practical guidance for privacy-budget calibration in federated fine-tuning. Finally, extensive experiments on real-world LLM benchmarks validate that DP-FedLoRA achieves strong privacy guarantees with minimal performance loss, presenting a scalable and effective solution for privacy-preserving LLM deployment in edge devices.



(a) LLaMA-2-7B (DP-FedLoRA with Rank = 32)



(b) LLaMA-2-13B (DP-FedLoRA with Rank = 32)

Fig. 6: Expectation $\mathbb{E}[\tilde{\Delta}]$ and Variance $\mathrm{Var}[\tilde{\Delta}]$ of Noisy Updates in Our DP-FedLoRA over training rounds with Different Base Models and Fixed LoRA Rank

TABLE IV: Converged Expectation $\mathbb{E}[\tilde{\Delta}]$ and Variance $\mathrm{Var}[\tilde{\Delta}]$ of Noisy Updates in Our DP-FedLoRA over training rounds with Different Base Models and Fixed LoRA Rank

Base Model	**Expectation** ($\mathbb{E}[\tilde{\Delta}]$)	**Variance** ($\mathrm{Var}[\tilde{\Delta}]$)
LLaMA-2 7B	1.04×10^{-8}	151.47
LLaMA-2 13B	2.39×10^{-9}	237.50

Acknowledge

This work was partly supported by the National Science Foundation of U.S. (2416872, 2244219, 2315596, 2146497).

References

[1] Y. Zheng, Y. Chen, B. Qian, X. Shi, Y. Shu, and J. Chen, "A review on edge large language models: Design, execution, and applications," *ACM Computing Surveys*, vol. 57, no. 8, pp. 1–35, 2025.

[2] J. Xu, Z. Li, W. Chen, Q. Wang, X. Gao, Q. Cai, and Z. Ling, "On-device language models: A comprehensive review," *arXiv preprint arXiv:2409.00088*, 2024.

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

[4] H. Amini, M. J. Mia, Y. Saadati, A. Imteaj, S. Nabavirazavi, U. Thakker, M. Z. Hossain, A. A. Fime, and S. Iyengar, "Distributed llms and multimodal large language models: A survey on advances, challenges, and future directions," *arXiv preprint arXiv:2503.16585*, 2025.

[5] E. T. M. Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2983–3013, 2023.

[6] P. Bellavista, L. Foschini, and A. Mora, "Decentralised learning in federated deployment environments: A system-level survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 1, pp. 1–38, 2021.

[7] N. Yan, Y. Su, Y. Deng, and R. Schober, "Federated fine-tuning of llms: Framework comparison and research directions," *arXiv preprint arXiv:2501.04436*, 2025.

[8] T. Nguyen, P. Lai, K. Tran, N. Phan, and M. T. Thai, "Active membership inference attack under local differential privacy in federated learning," *arXiv preprint arXiv:2302.12685*, 2023.

[9] J. Zhang, M. Li, S. Zeng, B. Xie, and D. Zhao, "A survey on security and privacy threats to federated learning," in *2021 International conference on networking and network applications (NaNA)*. IEEE, 2021, pp. 319–326.

[10] H. Kibriya, W. Z. Khan, A. Siddiqa, and M. K. Khan, "Privacy issues in large language models: a survey," *Computers and Electrical Engineering*, vol. 120, p. 109698, 2024.

[11] M. Ali, A. Arunasalam, and H. Farrukh, "Understanding users' security and privacy concerns and attitudes towards conversational ai platforms," *arXiv preprint arXiv:2504.06552*, 2025.

[12] Y. Xin, J. Yang, S. Luo, H. Zhou, J. Du, X. Liu, Y. Fan, Q. Li, and Y. Du, "Parameter-efficient fine-tuning for pre-trained vision models: A survey," *arXiv preprint arXiv:2402.02242*, 2024.

[13] L. Xu, H. Xie, S.-Z. J. Qin, X. Tao, and F. L. Wang, "Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment," *arXiv preprint arXiv:2312.12148*, 2023.

[14] E. B. Zaken, S. Ravfogel, and Y. Goldberg, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," *arXiv preprint arXiv:2106.10199*, 2021.

[15] R. He, L. Liu, H. Ye, Q. Tan, B. Ding, L. Cheng, J.-W. Low, L. Bing, and L. Si, "On the effectiveness of adapter-based tuning for pretrained language model adaptation," *arXiv preprint arXiv:2106.03164*, 2021.

[16] S. M. Siddiqui, M. A. Sheikh, M. Aleem, and K. R. Singh, "Comparative analysis of efficient adapter-based fine-tuning of state-of-the-art transformer models," *arXiv preprint arXiv:2501.08271*, 2025.

[17] 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.

[18] S. Wang, L. Yu, and J. Li, "Lora-ga: Low-rank adaptation with gradient approximation," *Advances in Neural Information Processing Systems*, vol. 37, pp. 54 905–54 931, 2024.

[19] C. Gao, K. Chen, J. Rao, B. Sun, R. Liu, D. Peng, Y. Zhang, X. Guo, J. Yang, and V. Subrahmanian, "Higher layers need more lora experts," *arXiv preprint arXiv:2402.08562*, 2024.

[20] S. Hayou, N. Ghosh, and B. Yu, "The impact of initialization on lora finetuning dynamics," *Advances in Neural Information Processing Systems*, vol. 37, pp. 117 015–117 040, 2024.

[21] H. Chen, R. Li, B. Zhu, Z. Wang, and L. Chen, "Iteris: Iterative inference-solving alignment for lora merging," *arXiv preprint arXiv:2411.15231*, 2024.

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

[23] Z. Xu, S. Tong, P. Xie, and J. Wang, "From demodulation to decoding: Toward complete lora phy understanding and implementation," *ACM Transactions on Sensor Networks*, vol. 18, no. 4, pp. 1–27, 2023.

[24] S. Chen, O. Tavallaie, N. Nazemi, and A. Y. Zomaya, "Rbla: Rank-based-lora-aggregation for fine-tuning heterogeneous models in flaas," Sep 2024. [Online]. Available: https://arxiv.org/abs/2408.08699

[25] P. Wu, K. Li, T. Wang, Y. Dong, V. C. M. Leung, and F. Wang, "Fedfmsl: Federated learning of foundation models with sparsely activated lora," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 15 167–15 181, 2024.

[26] S. Chen, O. Tavallaie, N. Nazemi, X. Chen, and A. Y. Zomaya, "Autorank: Mcda based rank personalization for lora-enabled distributed learning," 2024. [Online]. Available: https://arxiv.org/abs/2412.15553

[27] N.-H. Nguyen, T.-A. Nguyen, T. Nguyen, V. T. Hoang, D. D. Le, and K.-S. Wong, "Towards efficient communication and secure federated recommendation system via low-rank training," in *Proceedings of the ACM Web Conference 2024*. New York, NY, USA: Association for Computing Machinery, 2024, p. 3940–3951. [Online]. Available: https://doi.org/10.1145/3589334.3645702

[28] J. Zhao, T. Wang, W. Abid, G. Angus, A. Garg, J. Kinnison, A. Sherstinsky, P. Molino, T. Addair, and D. Rishi, "Lora land: 310 fine-tuned llms that rival gpt-4, a technical report," 2024. [Online]. Available: https://arxiv.org/abs/2405.00732

[29] Y. He, B. Li, L. Liu, Z. Ba, W. Dong, Y. Li, Z. Qin, K. Ren, and C. Chen, "Towards label-only membership inference attack against pre-trained large language models," 2025. [Online]. Available: https://arxiv.org/abs/2502.18943

[30] Y. Hu, Z. Li, Z. Liu, Y. Zhang, Z. Qin, K. Ren, and C. Chen, "Membership inference attacks against vision-language models," 2025. [Online]. Available: https://arxiv.org/abs/2501.18624

[31] Y. Wang, X. Ma, G. Zhang, Y. Ni, A. Chandra, S. Guo, W. Ren, A. Arulraj, X. He, Z. Jiang, T. Li, M. Ku, K. Wang, A. Zhuang, R. Fan, X. Yue, and W. Chen, "Mmlu-pro: A more robust and challenging multi-task language understanding benchmark," 2024. [Online]. Available: https://arxiv.org/abs/2406.01574

[32] M. Kazemi, B. Fatemi, H. Bansal, J. Palowitch, C. Anastasiou, S. V. Mehta, L. K. Jain, V. Aglietti, D. Jindal, P. Chen, N. Dikkala, G. Tyen, X. Liu, U. Shalit, S. Chiappa, K. Olszewska, Y. Tay, V. Q. Tran, Q. V. Le, and O. Firat, "Big-bench extra hard," 2025. [Online]. Available: https://arxiv.org/abs/2502.19187

[33] J. Frohberg and F. Binder, "Crass: A novel data set and benchmark to test counterfactual reasoning of large language models," 2022. [Online]. Available: https://arxiv.org/abs/2112.11941

[34] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," 2021. [Online]. Available: https://arxiv.org/abs/2104.11375

[35] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2020. [Online]. Available: https://arxiv.org/abs/1812.06127

[36] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," 2021. [Online]. Available: https://arxiv.org/abs/1910.06378

[37] J. Sun, X. Wu, H. Huang, and A. Zhang, "On the role of server momentum in federated learning," 2023. [Online]. Available: https://arxiv.org/abs/2312.12670

[38] S. Cao, H. Wu, X. Wu, R. Ma, D. Wang, Z. Han, and W. Zhang, "Fedda: Resource-adaptive federated learning with dual-alignment aggregation optimization for heterogeneous edge devices," *Future Gener. Comput. Syst.*, vol. 163, p. 107551, 2025. [Online]. Available: https://doi.org/10.1016/j.future.2024.107551

[39] G. A. Baumgart, J. Shin, A. Payani, M. Lee, and R. R. Kompella, "Not all federated learning algorithms are created equal: A performance evaluation study," 2024. [Online]. Available: https://arxiv.org/abs/2403.17287

[40] L. Ju, T. Zhang, S. Toor, and A. Hellander, "Accelerating fair federated learning: Adaptive federated adam," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 1017–1032, 2024.