Promoting Data and Model Privacy in Federated Learning through Quantized LoRA

JianHao Zhu, Changze Lv, Xiaohua Wang, Muling Wu, Wenhao Liu, Tianlong Li, Zixuan Ling, Cenyuan Zhang, Xiaoqing Zheng*, Xuanjing Huang
School of Computer Science, Fudan University, Shanghai, China
{zhujh22}@m.fudan.edu.cn
{zhengxq,xjhuang}@fudan.edu.cn

Abstract

Conventional federated learning primarily aims to secure the privacy of data distributed across multiple edge devices, with the global model dispatched to edge devices for parameter updates during the learning process. However, the development of large language models (LLMs) requires substantial data and computational resources, rendering them valuable intellectual properties for their developers and owners. To establish a mechanism that protects both data and model privacy in a federated learning context, we introduce a method that only requires distributing a quantized version of the model's parameters during training. This method enables accurate gradient estimations for parameter updates while preventing clients from accessing a model with performance comparable to the centrally hosted one. Moreover, we combine this quantization strategy with LoRA, a popular and parameter-efficient fine-tuning method, to significantly reduce communication costs in federated learning. The proposed framework, named FEDLPP, successfully ensures both data and model privacy in federated learning. Additionally, the learned central model exhibits good generalization and can be trained in a resource-efficient manner.

1 Introduction

As large language models (LLMs) (Radford et al., 2019; Brown et al., 2020; Zhang et al., 2022; Chowdhery et al., 2023; Workshop et al., 2022; Zeng et al., 2022; Achiam et al., 2023; Touvron et al., 2023) continue to advance, their applications are proliferating across various fields, including healthcare (Ge et al., 2020), finance (Long et al., 2020), and the mobile keyboard (Ji et al., 2019). These applications often involve training LLMs on data that is distributed across multiple clients or edge devices, with stringent privacy constraints on this data. Federated Learning (FL) (Konecnỳ et al., 2016; McMahan et al., 2017; Yang et al., 2019; Kairouz et al., 2021) has emerged as a promising paradigm in these scenarios, allowing models to be trained on decentralized data without transferring raw data.

In traditional FL, the main focus has been ensuring data privacy. The central server sends the global model to clients, who update the model parameters locally and then send the updates back to the server. However, as LLMs become more sophisticated and valuable, the models themselves become critical intellectual property (IP) that also requires protection. This necessity is especially clear when LLMs are commercialized as paid services, where unauthorized access to the models could significantly undermine the interests of the developers and owners.

There are two ways to maintain intellectual property. One traditional method is to apply for a patent to protect the unique design from being copied. Another method is through secrecy (such as a unique recipe). In our scenario, where almost all generative language models currently use transformer architecture(Vaswani et al., 2017), the network weights are akin to a unique recipe. We protect the intellectual property of the owner by safeguarding the model privacy.

This new scenario in FL indicates that clients, as custodians of private data, are restricted from accessing the model on the server. Similarly, the server, as the owner of the FL product, is not authorized to collect data dispersed among different clients. This raises the question: Is there a framework that can simultaneously protect both data and model privacy ?

Unfortunately, existing FL frameworks mainly focus on data privacy, neglecting model privacy, leaving the model's intellectual property vulnerable to potential breaches. Take FEDAVG, the most widely used FL algorithm, as an example, clients

* Corresponding author.

Findings of the Association for Computational Linguistics: EACL 2024, pages 10501–10512

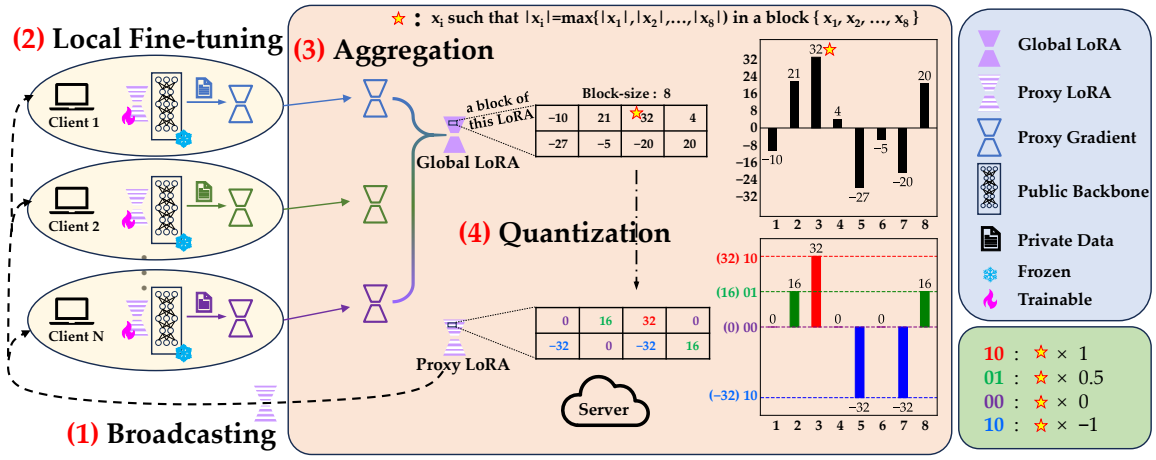

Figure 1: Visualization of the proposed method. To prevent clients (or edge devices) from fully accessing a global model, the central server broadcasts only the quantized version of the global LoRA's parameters during each communication round. The two bar charts visually demonstrate the quantization process, where eight distinct parameter values are mapped into four categories (highlighted in red, green, purple and blue).

have access to the global model maintained by the server, which the server intends to commercialize. It will pose a significant risk to the global model's intellectual property and the interests of this model's owners. Hence, there is a crucial need for mechanisms that ensure "LLM Privacy Protection", which means preventing clients (or edge devices) from obtaining a model that performs comparably to the final global model during the FL process.

To the best of our knowledge, only one existing study, FEDSP (Dong et al., 2023), has attempted to address LLM privacy protection in FL. The FEDSP approach involves constructing a proxy model at the onset of the FL process, which clients download from the server rather than the actual global LLM, avoiding the need to share the global LLM. However, FEDSP necessitates the server to have access to a labeled dataset that is identically and independently distributed (iid) to the clients' data, which is often impractical in real-world FL. Therefore, we need an algorithm that does not rely on such a dataset.

Recognizing these limitations, we introduce a novel approach, FEDLPP (FL with LLM Privacy Protection) , to address the dual challenges of protecting both data and model privacy, without requiring any auxiliary dataset for the server. As shown in Fig 1, our proposed framework leverages quantization techniques and parameter-efficient fine-tuning (PEFT) (Houlsby et al., 2019; Lester et al., 2021; Li and Liang, 2021; Hu et al., 2021; Zaken et al., 2021; Wu et al., 2024) strategies to ensure that clients can access only a quantized version of the model's parameters. This prevents clients from obtaining a model that performs comparably to the central one while still allowing accurate gradient estimations for parameter updates. Additionally, by combining this quantization strategy with LoRA (Hu et al., 2021), we significantly reduce communication costs in FL.

In summary, our framework, FEDLPP, effectively ensures data and model privacy in FL contexts. The learned central model exhibits strong generalization capabilities and can be trained in a resource-efficient manner, addressing the dual privacy challenges in FL. Our contributions are as follows:

- **Privacy Protection for Both Models and Data**. The proposed algorithm enables effective learning without requiring model owners to open-source their models or data owners to share their data, achieves mutual confidentiality between the server and clients.
- **Excellent Performance**. Experiments on four text generation datasets demonstrate the great performance improvement of our method compared to the baseline. Our approach achieves model privacy protection without a significant drop in performance.
- **Low Communication and Computation Demands**. Our method is also applicable in scenarios with limited computational and communication resources, making it suitable for real-world applications.

- **No Need for Auxiliary Datasets**. Our approach eliminates the requirement for an auxiliary dataset to be provided to the server, making it more practical for real-world applications where such datasets are often difficult to obtain.

2 Related Work

2.1 Federated Learning

Federated learning (FL)(Konecnỳ et al., 2016; McMahan et al., 2017; Yang et al., 2019; Kairouz et al., 2021) is a distributed machine learning approach that enables model training across multiple devices or clients while keeping data decentralized and preserving privacy. FL has gained significant attention for addressing privacy concerns in sensitive data scenario. Research in FL has primarily focused on challenges (Wen et al., 2023)such as privacy and security challenges(Bogdanov et al., 2008; Geyer et al., 2017; Cai et al., 2020), communication challenges(Shahid et al., 2021) and heterogeneity challenges(Wang et al., 2020). Notable works in FL include FEDAVG(McMahan et al., 2017), FEDPROX(Li et al., 2020), and FEDGAN(Rasouli et al., 2020), with FEDAVG(McMahan et al., 2017) being a well-recognized algorithm.

2.2 LLMs-Privacy-Protection in Federated Learning

The previously mentioned FEDSP(Dong et al., 2023) is the only notable study that has attempted to achieve LLM privacy protection in FL. FEDSP involves the server constructing a proxy model that is similar enough to the global model before federated learning starts, which is then deployed to the clients. In each communication round of FEDSP, the server broadcasts a soft prompt to the clients, which can be inserted into either the global model or the proxy model. Clients train this soft prompt through prefix-tuning (Li and Liang, 2021) and upload the updates of the soft prompt back to the server, which then aggregates these updates for the next iteration. Therefore, in FEDSP, clients only have access to the proxy model, thereby protecting the global model's privacy.

However, due to significant differences between the proxy model and the global model, the soft prompts trained by clients may not align with the global model. To address this, FEDSP uses distillation techniques during proxy model construction to ensure it resembles the global model closely. Additionally, in each training round, the server fine-tunes the aggregated soft prompt to align it with the global model. Both steps require an auxiliary dataset, which is often impractical in real-world FL contexts.

2.3 Quantization for Federated Learning

Research on quantization techniques in FL has predominantly focused on reducing communication bandwidth or computational overheads, without addressing their potential application for LLM privacy protection. A milestone in this field, FEDPAQ (Reisizadeh et al., 2020), introduces a FL framework with quantization, where clients upload quantized local updates to the server. In contrast, we apply the quantization operation to the LoRA parameters sent by the server to the clients.

2.4 Parameter-Efficient Fine-Tuning for Federated Learning

Parameter-Efficient Fine-Tuning (PEFT)(Houlsby et al., 2019; Lester et al., 2021; Li and Liang, 2021; Hu et al., 2021; Zaken et al., 2021; Wu et al., 2024) has gained significant attention in recent years, with techniques such as Prefix Tuning (Li and Liang, 2021) and LoRA (Hu et al., 2021) demonstrating effective strategies for reducing model parameters while maintaining performance.

Leveraging these features, the PEFT methods can reduce communication overhead and alleviate the training burden on individual clients in FL. In the field of Natural Language Processing (NLP), FedPETuning (Zhang et al., 2023) provides a benchmark for a comprehensive evaluation of PEFT methods for LLMs under FL settings. Our work also demonstrates how PEFT methods contribute to LLM privacy protection.

3 Methods

3.1 Preliminary

In a cross-device scenario with N clients, where client i owns a private dataset $\mathcal{D}_i$, the standard Federated Learning (FL) considers training the weight matrix $\mathbf{W}$ by minimizing the loss (empirical risk):

$$\min_{\mathbf{W}} \mathcal{L}(\mathbf{W}) = \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{M} \mathcal{L}_i(\mathbf{W}) \tag{1}$$

where $\mathcal{L}_i(\cdot)$ is the local loss function on $\mathcal{D}_i$, ,$M = \sum_{i=1}^{N} |\mathcal{D}_i|$, $\mathbf{W} \in \mathcal{R}^{d \times k}$.

To reduce the communication overhead in the FL course, we use Low-Rank Adaptation technology

Dataset	FL Scenario	#Train	#Validation	#Test	N	C
E2E	large-scale cross-device	42,061	4,672	46,93	25	5
E2E	cross-silo	42,061	46,72	4,693	5	5
ViGGO	cross-silo	5,103	714	1,083	5	5
DART	large-scale cross-device	62,659	2,768	5,097	50	5
DialogSum	large-scale cross-device	12,460	500	1,500	20	5

Table 1: Statistics for different datasets. The symbol N denotes the total number of clients and C denotes the actual number of clients participating in each communication round. Different values of C are used to simulate scenarios like unresponsive clients or synchronization errors in real-world settings.

Method	BLEU	NIST	METEOR	ROUGE-L	CIDEr	Data Privacy	Model Privacy
FedLPP							
⌞ Global	**34.60**	**6.06**	**31.43**	**51.32**	**1.70**	✓	✓
⌞ Proxy	32.46	5.12	29.72	50.31	1.52		
FedSP	26.42	3.65	25.88	44.42	1.21	✗	✓
⌞(w/o Server Train)	0.14	0.20	2.91	8.57	0.00	✓	✓
⌞(w/o Client Train)	29.64	4.85	27.51	46.74	1.38	✗	✓
FedAvg + LoRA	36.04	6.58	33.64	53.01	1.90	✓	✗

Table 2: Results of the proposed method compared to the baselines. For FEDLPP, we use bold formatting to highlight the better one between Global model and Proxy model. Our proposed method FedLPP achieves significant improvement over the baseline method FedSP, while simultaneously protecting both the model and the data.

to decompose $\mathbf{W}$ into a frozen pre-trained weight matrix $\mathbf{W}_0 \in \mathcal{R}^{d\times k}$ and trainable delta matrices $\mathbf{B} \in \mathcal{R}^{d\times r}$ and $\mathbf{A} \in \mathcal{R}^{r\times k}$, where r is the rank of LoRA:

$$\mathbf{W} = \mathbf{W}_0 + \mathbf{BA}, \tag{2}$$

Here, $\mathbf{W}_0$ is open-source and available online for both the server and the clients, while $\mathbf{B}$ and $\mathbf{A}$ contain task-specific parameters enriched with proprietary knowledge after the FL course. Therefore, the privacy of $\mathbf{B}$ and $\mathbf{A}$, which are the final product of the FL course, is crucial for maintaining their owners' commercial interests. This means the server should not directly share $\mathbf{B}$ and $\mathbf{A}$ with clients, while still needing to collect updates of these matrices from them to enable effective FL.

$$\mathbf{X}^{t+1} = \mathbf{X}^t + \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{M} \Delta \mathbf{X}_i^t \tag{3}$$

In Equation 3, $\mathbf{X}$ can represent either $\mathbf{B}$ or $\mathbf{A}$, depending on the context, and $t \in \{1, 2, ..., T\}$ denotes the t-th communication round of the FL course, where T is the total number of communication rounds.

3.2 Computing Proxy LoRA Matrices by Quantization

In traditional deep neural networks (DNNs), full-precision model parameters are typically stored in 32-bit floating-point format. To preserve the privacy of the global LoRA matrices $\mathbf{X} \in \mathcal{R}^{a\times b}$, FedLPP no longer sends the exact values of $\mathbf{X}$ to the clients. Instead, the server computes low-precision versions $\mathbf{X} = Q(\mathbf{X})$ as proxy LoRA matrices which are available to clients. The specific calculation process of $Q(\cdot)$ is as follows.

First, following QLoRA(Dettmers et al., 2024), after selecting the desired bit-width w , we choose $2^{(w-1)} - 2$ numbers from the interval $(-1, 0)$ and $2^{(w-1)} - 1$ numbers from the interval $(0, 1)$. we then combine these numbers with -1, 0, and 1, resulting in a total of 2^w numbers. We refer to these numbers as "standard numbers" and denote them in sorted order as $V = \{v_0, v_1, ..., v_{2^w-1}\}$, where $v_0 = -1$, $v_{2^{(w-1)}-1} = 0$, and $v_{2^w-1} = 1$.

Second, we combine consecutive floating-point numbers in matrix $\mathbf{X}$ into blocks of size s, ensuring the spatial continuity[1] of $\mathbf{X}$. For each block X_l, we identify the floating-point number with the largest absolute value, denoted as z_l. Thus, the normalized version of the l-th block X_l can be obtained as follows:

$$X_l' = \frac{X_l}{z_l} \tag{4}$$

Thus, the floating-point numbers in X_l' fall

[1]Here, 'spatial continuity' refers to the uninterrupted sequence of floating-points numbers in the flattened representation of matrix $\mathbf{X}$.

within the interval $(-1, 1)$, which matches the value range of the 'standard numbers.' Consequently, we can replace each floating-point number in $X_l^{'}$ with the closest 'standard number,' resulting in a new block of the same size, denoted as $\tilde{X}_l^{'}$. This new block serves as an approximate version of $X_l^{'}$.

Next, we define $\tilde{X}_l = z_l \tilde{X}_l^{'}$, which is also an approximate version of X^l because $X^l = z_l X_l^{'}$. Finally, by arranging all $\lfloor \frac{ab}{s} \rfloor$ blocks $\tilde{X}_l$ in their original order, we can reconstruct an approximate version of the original matrix $\mathbf{X}$ as $\tilde{\mathbf{X}}$.

It is important to note that the computed $\tilde{\mathbf{X}}$ incurs some information loss compared to $\mathbf{X}$. If we broadcast $\tilde{\mathbf{X}}$ as a proxy LoRA matrix to the clients instead of $\mathbf{X}$, the clients will not obtain a model with performance comparable to that of the server's model. However, since $0 \in V$, we can conclude that the transformation from $\mathbf{X}$ to $\tilde{\mathbf{X}}$ preserves the signs of all floating-point numbers in $\mathbf{X}$. As a result, the proxy gradients computed by clients using these proxy LoRA matrix will provide a good estimate of the true gradients. Therefore, our constructed proxy LoRA matrix has the potential to achieve both data privacy protection and LLM privacy protection simultaneously.

3.3 Client-Side Local Fine-tuning

Upon receiving the trainable proxy LoRA matrix $\tilde{\mathbf{X}}^t$ in the t-th round, the selected client i combines it with the frozen backbone $\mathbf{W}_0$ to form a local model. Subsequently, using $\mathcal{L}_i$ and local private data $\mathcal{D}_i$, the local model undergoes training to obtain proxy update $\Delta\tilde{\mathbf{X}}_i^t$, which is then sent back to the server. The central server employs secure aggregation algorithms(Bonawitz et al., 2016) to compute $\mathbf{X}^{t+1}$ for the next round. This process iterates continuously until the communication round t reaches the upper limit T.

4 Experiments

To test if our algorithm maintains good FL performance while preserving both model and data privacy, we conducted extensive experiments in this section, including performance comparisons (see Sec 4.4) . Additionally, we conducted an ablation analysis of FEDLPP , including different FL scenarios (see Sec 4.6) , different quantization levels (see Sec 4.5) and scaling (see Sec 4.6) .

4.1 Models, Datasets and Metrics

We conduct our experiments with two popular language models, namely GPT2-XL and GPT2-Medium (Radford et al., 2019), utilizing four datasets: E2E (Novikova et al., 2017), VIGGO (Juraska et al., 2019), DART (Nan et al., 2020), and DIALOGSUM (Chen et al., 2021).

The E2E dataset comprises a collection of table-to-text generation data for training end-to-end natural language generation systems in the restaurant domain. The VIGGO dataset is also a table-to-text generation dataset, but it is designed for generalizable and conversational dialogue act types. DART is another open-domain table-to-text generation dataset, while DIALOGSUM is specifically designed for dialogue summarization.

Referring to the original papers for these datasets, we use five metrics to evaluate the quality of the text generated by our model: BLEU (Papineni et al., 2002), NIST (Belz and Reiter, 2006), METEOR (Banerjee and Lavie, 2005), ROUGE-L (Lin, 2004), and CIDEr (Vedantam et al., 2015). More details on how we use these four datasets in the FL setting are shown in Table 1.

4.2 Baselines

We use the following four methods as baselines to compare with our proposed FedLPP:

- **FEDSP** (Dong et al., 2023): This is the only framework that focuses on addressing the issue of LLM privacy protection in FL.
- **FEDSP (w/o Server Train)**: A variant of FedSP where the server cannot use a labeled dataset for supervised fine-tuning in each communication round. In this setting, FedSP can still utilize the unlabeled dataset for the knowledge distillation (KD) process to create a proxy model at the start of its FL process. This variant helps to verify whether the performance of FEDSP derives from the labeled information of the auxiliary dataset deployed on the server.
- **FEDSP (w/o Client Train)**: Another variant of FedSP where clients do not participate in the FL, meaning that the server only trains the model on its labeled dataset. This variant helps to verify how much of FEDSP's performance is attributed to the labeled information of the auxiliary dataset deployed on the server.

Method	BLEU	NIST	METEOR	ROUGE-L	CIDEr	Data Privacy	Model Privacy
FEDLPP ($w = 1$)							
⌞ Global	**51.81**	**6.66**	**0.33**	**60.30**	**1.29**	✓	✓
⌞ Proxy	51.54	6.15	0.33	59.94	1.24		
FEDLPP ($w = 2$)							
⌞ Global	**54.93**	**7.70**	**0.37**	**61.65**	**1.40**	✓	✓
⌞ Proxy	52.05	6.23	0.35	60.75	1.22		
FEDLPP ($w = 3$)							
⌞ Global	54.62	7.71	**0.41**	**62.33**	**1.45**	✓	✗
⌞ Proxy	**54.97**	**7.74**	0.38	61.65	1.42		
FEDAVG + LoRA	54.88	7.71	0.41	63.24	1.50	✓	✗

Table 3: Results obtained from the proposed method with varying quantization levels.With an appropriate quantization level, specifically w=2, FEDLPP achieves both model and data protection while maintaining high performance

FL Scenario	Method	BLEU	NIST	METEOR	ROUGE-L	CIDEr	Data Privacy	Model Privacy
Cross-Silo	FEDLPP							
	⌞ Global	**54.75**	**7.68**	**0.41**	**62.66**	**1.47**	✓	✓
	⌞ Proxy	53.42	7.56	0.38	61.12	1.36		
	FEDAVG+LoRA	54.99	7.73	0.41	63.32	1.51	✓	✗
Large-Scale Cross-Device	FEDLPP							
	⌞ Global	**54.93**	**7.70**	**0.37**	**61.65**	**1.40**	✓	✓
	⌞ Proxy	52.05	6.23	0.35	60.75	1.22		
	FEDAVG+LoRA	54.88	7.71	0.41	63.24	1.50	✓	✗

Table 4: Results obtained from the proposed method in different FL scenarios. After transitioning from a Cross-Silo to a Large-Scale Cross-Device FL scenario,the performance of FEDLPP has not been significantly affected, and it continues to effectively protect the model.

- **FEDAVG + LoRA**: Uses the FEDAVG algorithm to train LoRA adapters while keeping the backbone frozen. This is also a variant of our method that doesn't quantify the LoRA matrices before broadcasting them to the clients. Since this method does not consider LLM privacy protection (i.e., clients can have unrestricted access to the server's LoRA adapters), we only consider it a potential upper bound for comparison with the FEDLPP algorithm. This variant has also been considered in FedPETuning (Zhang et al., 2023).

4.3 Implementation Details

For fair and reasonable comparisons, we conducted hyperparameter searches for each dataset and method. We selected the best model based on the loss from the validation set and reported metrics on the test set.

We primarily evaluated the performance of FEDLPP under three quantization levels, namely bit-width $w \in \{1, 2, 3\}$, with a consistent block size of 256. As for FedSP, the prefix length was chosen from $\{40, 80, 160\}$, and the number of layers in the proxy model was chosen from $\{1, 4, 8\}$. For all methods, the learning rate was chosen from $\{1e{-}4, 3e{-}4, 1e{-}3\}$, and the batch size was set to 16. The training epochs in each round were chosen from $\{1, 3, 5\}$, and the total number of communication rounds was set to 100. We implemented the proposed approach, FedLPP, and baseline methods based on Hugging Face Transformers (Wolf et al., 2020). All experiments were conducted on a single server equipped with four NVIDIA GeForce RTX 3090 GPUs, each with 24 GB of RAM.

4.4 Performance Comparison

In Table 2, we present comparisons between our method and the baselines. For all methods, we select the best models among all communication rounds as the final model for the FL course. For our approach, we report both the best global model on the server and the best proxy model available to clients. We only consider FEDLPP to have effectively protected the global model when the global model outperforms the proxy model in all metrics.

Model	Method	BLEU	NIST	METEOR	ROUGE-L	CIDEr	Data Privacy	Model Privacy
GPT-2 Medium	FEDLPP							
	⌞ Global	**54.93**	**7.70**	**0.37**	**61.65**	**1.40**	✓	✓
	⌞ Proxy	52.05	6.23	0.35	60.75	1.22		
	FEDAVG+LoRA	54.88	7.71	0.41	63.24	1.50	✓	✗
GPT-2 XL	FEDLPP							
	⌞ Global	**54.81**	**7.76**	**0.39**	**61.34**	**1.41**	✓	✓
	⌞ Proxy	51.58	7.03	0.35	59.51	1.23		
	FEDAVG+LoRA	55.52	7.77	0.41	63.01	1.50	✓	✗
Llama-2 7B	FEDLPP							
	⌞ Global	**55.80**	**7.37**	**0.42**	**63.92**	**1.55**	✓	✓
	⌞ Proxy	52.22	7.31	0.39	60.87	1.45		
	FEDAVG+LoRA	55.86	7.97	0.42	64.33	1.57	✓	✗

Table 5: Results obtained from the proposed method using models of varying sizes.

Table 2 shows the average performance across the four datasets.

Based on Table 2, we can draw the following conclusions from the table. Our method shows significant improvements over FEDSP across four datasets and maintains comparable performance to FEDAVG+LoRA. As FEDAVG+LoRA can be regarded as a variant of FEDLPP without model protection, we can infer from the results that our additional efforts to protect the model did not lead to a significant decrease in performance.

Additionally, we compare the global model with the proxy model in the FEDLPP algorithm. Analyzing the first two rows in Table 2, we find that the global model on the server consistently outperforms the proxy model across all five metrics, ensuring that FEDLPP indeed achieves the goal of protecting the commercial interests of the owners of large models. As long as the best model is not disclosed to any participating client, the model owned by the server will consistently maintain an advantage over each client, thereby protecting the fundamental interests of the model owner.

It is noted that our method is compatible with secure aggregation algorithms (Bonawitz et al., 2016), and we do not need a labeled dataset on the server which could compromise client data privacy, unlike FEDSP. Therefore, our method offers client data security comparable to FEDAVG in this regard.

However, the analysis of the baseline method, FEDSP, and its two variants confirms that the performance of FEDSP mainly arises from fine-tuning on the server's dataset. Without referencing the aggregated information from clients, the FEDSP server can train models with performance even better than the original FEDSP, but when FEDSP no longer fine-tunes on such a dataset on the server, the algorithm cannot proceed. Thus, the significance of this server dataset for FEDSP is clear.

4.5 Impact of Different Quantization Levels

In the FedLPP framework, there exists a trade-off between LLM privacy protection and performance. This trade-off depends on the level of quantization: a higher level of quantization (a smaller w) results in greater deviation of the proxy LoRA received by the clients from the global LoRA, resulting in increased protection but also a greater bias in the computed proxy gradients, which impacts final performance. In this study, we examine three different quantization bit widths: $\{1, 2, 3\}$, and the results are presented in Table 3. It is evident that choosing a bit level of 2 achieves better performance while ensuring LLM privacy protection.

If the quantization level is excessively low (w is big), the information loss in the proxy model received by the client will be insufficient to ensure model privacy protection. As shown in the table, when $w = 3$, the proxy model's performance even surpasses that of the global model in some metrics, which implies that the server fails to guarantee the model's privacy. This conclusion is consistent with the perspectives of (Jin et al., 2024) and (Marchisio et al., 2024).

4.6 Impact of Different FL Scenarios and Model Scaling

To evaluate the effectiveness of FEDLPP across different FL scenarios, we primarily considered two FL scenarios: the cross-silo FL scenario (Kairouz

et al., 2021) and the large-scale cross-device FL scenario (Lai et al., 2022). In the cross-silo scenario, the server selects all clients for training in each communication round, while in large-scale cross-device scenarios, the data held by local clients is scarcer; additionally, not every client participates in each learning round. This scenario more closely reflects real-world FL situations where, due to communication constraints or synchronization issues, the server cannot always receive responses from every client. Therefore, it is necessary to test whether the FedLPP algorithm can operate effectively under these more challenging conditions. In this experiment, we used VIGGO to simulate cross-silo FL scenarios, while DART and DIALOGSUM were used to simulate large-scale cross-device FL scenarios. Additionally, we used E2E to simultaneously simulate both scenarios to facilitate a clear comparison and investigate whether the effectiveness of FEDLPP is influenced by the FL scenario factor. The performance comparison of FEDLPP under two different data splits is shown in Table 4.

Furthermore, we extended our model from GPT2-Medium to larger models, including GPT2-XL and Llama2-7B (Touvron et al., 2023). To determine whether FedLPP can be effectively utilized with larger models, the results on the E2E dataset are shown in Table 5.

5 Conclusion

In this study, we presented FEDLPP, a novel federated learning framework designed to tackle the dual challenges of data privacy and model confidentiality. By integrating a quantization strategy with LoRA, FEDLPP facilitates effective updates to model parameters while significantly reducing communication overhead. Our framework ensures that each client participates in the learning process without accessing a full-performance model, thereby protecting the intellectual property rights of model developers. The results demonstrate that FEDLPP not only maintains robust privacy protections but also enables the creation of a generalized model with lower resource consumption. Moreover, FEDLPP is especially well-suited for scenarios where data privacy and intellectual property rights are crucial, offering a practical solution in the rapidly evolving landscape of federated learning.

Limitations

While FEDLPP introduces significant advancements in FL, several limitations warrant further investigation. First, the quantization process, although effective in reducing the model size and protecting intellectual property, may introduce quantization noise, potentially affecting the learning accuracy and convergence rate. Future work could explore adaptive quantization techniques to mitigate this issue. Secondly, the integration of LoRA is primarily tested under controlled conditions; its efficacy across diverse network architectures and heterogeneous data distributions remains to be fully evaluated. Addressing these limitations could enhance the applicability of FEDLPP across a broader range of FL scenarios and contribute to its adoption in industry-standard practices.

Reproducibility Statement

The authors have diligently worked to guarantee the reproducibility of the empirical findings presented in this paper. To facilitate reproducibility, the source code for the proposed method has been submitted alongside the paper, and we intend to make the source code publicly available on GitHub upon acceptance.

References

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

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of nlg systems. In *11th conference of the european chapter of the association for computational linguistics*, pages 313–320.

Dan Bogdanov, Sven Laur, and Jan Willemson. 2008. Sharemind: A framework for fast privacy-preserving computations. In *Computer Security-ESORICS 2008: 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008. Proceedings 13*, pages 192–206. Springer.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2016.

Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Xingjuan Cai, Yun Niu, Shaojin Geng, Jiangjiang Zhang, Zhihua Cui, Jianwei Li, and Jinjun Chen. 2020. An under-sampled software defect prediction method based on hybrid multi-objective cuckoo search. *Concurrency and Computation: Practice and Experience*, 32(5):e5478.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. Dialogsum: A real-life scenario dialogue summarization dataset. *arXiv preprint arXiv:2105.06762*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Chenhe Dong, Yuexiang Xie, Bolin Ding, Ying Shen, and Yaliang Li. 2023. Tunable soft prompts are messengers in federated learning. *arXiv preprint arXiv:2311.06805*.

Suyu Ge, Fangzhao Wu, Chuhan Wu, Tao Qi, Yongfeng Huang, and Xing Xie. 2020. Fedner: Privacy-preserving medical named entity recognition with federated learning. *arXiv preprint arXiv:2003.09288*.

Robin C Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

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

Shaoxiong Ji, Shirui Pan, Guodong Long, Xue Li, Jing Jiang, and Zi Huang. 2019. Learning private neural language modeling with attentive aggregation. In *2019 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

Renren Jin, Jiangcun Du, Wuwei Huang, Wei Liu, Jian Luan, Bin Wang, and Deyi Xiong. 2024. A comprehensive evaluation of quantization strategies for large language models. *arXiv preprint arXiv:2402.16775*.

Juraj Juraska, Kevin K Bowden, and Marilyn Walker. 2019. Viggo: A video game corpus for data-to-text generation in open-domain conversation. *arXiv preprint arXiv:1910.12129*.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210.

Jakub Konecnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 8.

Fan Lai, Yinwei Dai, Sanjay Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha Madhyastha, and Mosharaf Chowdhury. 2022. Fedscale: Benchmarking model and system performance of federated learning at scale. In *International conference on machine learning*, pages 11814–11827. PMLR.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Guodong Long, Yue Tan, Jing Jiang, and Chengqi Zhang. 2020. Federated learning for open banking. In *Federated Learning: Privacy and Incentive*, pages 240–254. Springer.

Kelly Marchisio, Saurabh Dash, Hongyu Chen, Dennis Aumiller, Ahmet Üstün, Sara Hooker, and Sebastian Ruder. 2024. How does quantization affect multilingual llms? *arXiv preprint arXiv:2407.03211*.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, et al. 2020. Dart: Open-domain structured data record to text generation. *arXiv preprint arXiv:2007.02871*.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Mohammad Rasouli, Tao Sun, and Ram Rajagopal. 2020. Fedgan: Federated generative adversarial networks for distributed data. *arXiv preprint arXiv:2006.07228*.

Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. 2020. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International conference on artificial intelligence and statistics*, pages 2021–2031. PMLR.

Osama Shahid, Seyedamin Pouriyeh, Reza M. Parizi, Quan Z. Sheng, Gautam Srivastava, and Liang Zhao. 2021. Communication efficiency in federated learning: Achievements and challenges. *Preprint*, arXiv:2107.10996.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Alex Wang. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Cong Wang, Yuanyuan Yang, and Pengzhan Zhou. 2020. Towards efficient scheduling of federated mobile devices under computational and statistical heterogeneity. *IEEE Transactions on Parallel and Distributed Systems*, 32(2):394–410.

Jie Wen, Zhixia Zhang, Yang Lan, Zhihua Cui, Jianghui Cai, and Wensheng Zhang. 2023. A survey on federated learning: challenges and applications. *International Journal of Machine Learning and Cybernetics*, 14(2):513–535.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Muling Wu, Wenhao Liu, Xiaohua Wang, Tianlong Li, Changze Lv, Zixuan Ling, Jianhao Zhu, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. 2024. Advancing parameter efficiency in fine-tuning via representation editing. *arXiv preprint arXiv:2402.15179*.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

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

Zhuo Zhang, Yuanhang Yang, Yong Dai, Qifan Wang, Yue Yu, Lizhen Qu, and Zenglin Xu. 2023. 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*, pages 9963–9977. Association for Computational Linguistics (ACL).

A Selection of V

Regarding the specific choice of V, please refer to the appendix. The selection of V directly determines the quality of quantization. Based on QLoRA(Dettmers et al., 2024), it is observed that the model parameters follow a normal distribution. It is preferable for the standard numbers in V to be densely distributed in the middle range close to 0, while sparsely distributed towards the ends of the intervals at 1 and -1. This ensures a relatively even distribution of model parameters allocated to different standard numbers, leading to an optimal estimation of the original parameter matrix under the constraint of limited standard numbers. Specifically, when w=2, the selection of V in the FedLPP algorithm is $[-1.00, 0.00, 0.33, 1.00]$, and $V = [-1.00, -0.47, -0.21, 0.00, 0.16, 0.33, 0.56, 1.00]$ for $w = 3$, In particular, when $w = 1$, we choose V as $[-1.00, 0.00, 1.00]$.

B Generalizability to other types of tasks

In addition to conducting experiments on the generation task, we also performed experiments on datasets from other task types to validate the effectiveness of our approach. Specifically, we conducted experiments using the GPT-2 (Medium) model on the SST-2 and RTE (Wang, 2018) datasets. The hyperparameter settings and experimental results are shown in Table 6 and Table 7

C Future Work

Building upon the foundational success of FEDLPP, several avenues for future research are evident to further refine and expand the capabilities of our FL framework. One immediate area of exploration involves the optimization of the quantization strategy to balance model performance with privacy preservation more effectively. Advanced techniques such as dynamic quantization or mixed-precision training could be employed to enhance model accuracy without compromising privacy.

Additionally, expanding the compatibility of FEDLPP with various neural network architectures, including more complex models like GANs or transformers, could significantly broaden its applicability. Investigating the framework's effectiveness in these contexts will help in addressing the diverse needs of practical applications in different sectors such as healthcare, finance, and telecommunications.

Another promising direction is the exploration of hybrid approaches that combine FEDLPP with other privacy-preserving techniques such as differential privacy or secure multi-party computation. Such combinations could offer layered security features, thereby providing stronger guarantees against potential privacy breaches.

Further, the development of resource management strategies to efficiently handle the computational and communication overheads in FEDLPP would be crucial, especially for deployment in edge computing scenarios. Optimizing these aspects will ensure that the benefits of FL can be realized even in resource-constrained environments.

Lastly, conducting large-scale empirical studies to validate the framework's efficacy across different real-world datasets and environments would provide deeper insights into its practical implications and limitations. This would not only solidify the theoretical advancements made but also guide the practical implementations of FL systems.

	RTE	SST-2
N	5	5
C	5	5
BATCH-SIZE	64	64
LR	1e-3,1e-4,3e-4	1e-3,1e-4,3e-4
LORA RANK	1,2,4,8	1,2,4,8
BIT-WIDTH	1,2,3	1,2,3
COMMUNICATION ROUND	100	100
LOCAL EPOCH	1	1

Table 6: Hyperparameter settings of different datasets.

Method	RTE	SST-2
FEDLPP		
⌞ Global	**75.64**	**96.53**
⌞ Proxy	72.56	95.30
FEDAVG + LORA	76.53	96.90

Table 7: Experimental results of different methods on the RTE and SST-2 datasets.