Privacy-Preserving and Federated Fine-Tuning of Language Models

Sajjad Ghiasvand
ECE Department
UC Santa Barbara
Santa Barbara, USA
sajjad@ucsb.edu

Wei-Yuan Su
ECE Department
UC Santa Barbara
Santa Barbara, USA
wei-yuansu@ucsb.edu

***Abstract*—Federated Learning (FL) has been integrated with parameter-efficient fine-tuning (PEFT) methods for Large Language Models (LLMs) to protect the privacy of distributed users at the edge. However, even with FL, privacy leakage remains a concern. To address this, we incorporate differential privacy into federated fine-tuning of LLMs. Through extensive experiments, we evaluate the performance of two PEFT methods—Low-Rank Adaptation (LoRA) and Adapters—and demonstrate that, in our specific problem setup, Adapters can outperform LoRA, despite LoRA being generally favored for its efficiency in standard settings.**

***Index Terms*—Language Models, Federated Learning, Differential Privacy, Parameter-Efficient Fine-Tuning.**

I. Introduction

Large Language Models (LLMs) like GPT-4 [1], LLaMA [2], and BERT [3] have transformed artificial intelligence, making tasks like translation and summarization [4] more advanced with the help of Transformer architectures [5]. These models can be fine-tuned for specific applications, such as toxicity classification [6], using specialized datasets [7]. However, their massive size—often containing billions of parameters—makes full fine-tuning expensive and prone to overfitting. To solve this, Parameter-Efficient Fine-Tuning (PEFT) methods, including Adapters [8], Prompt-Tuning [9], LoRA [10], and LoRETTA [11], have been developed. These techniques update only a small portion of the model's parameters while keeping the rest fixed, significantly reducing computational costs without sacrificing performance [12]. Among these methods, LoRA and Adapters stand out as a particularly efficient and flexible approach, making it the main focus of our study.

Most PEFT methods assume that LLM fine-tuning happens on a single machine or client. However, in practice, sensitive datasets—such as medical or legal records—are often spread across multiple devices [13], [14]. Due to privacy concerns, centralizing this data is not a viable option, making it essential to develop fine-tuning methods that adapt LLMs while preserving data privacy. Federated Learning (FL) addresses this challenge by keeping data on local devices throughout training. Rather than sending raw data to a central server, FL allows clients to update model parameters locally and share only aggregated updates, such as gradients or parameters, which are then merged into a global model [15]. As a result, FL has been widely adopted in PEFT approaches [16]–[18].

While FL provides an effective framework for fine-tuning LLMs on distributed, privacy-sensitive data, it remains susceptible to privacy leakage. Differential Privacy (DP), a well-established technique for enforcing strong privacy guarantees [19], [20], has been widely adopted to enhance privacy in FL settings. Prior works [21], [22] have applied DP-SGD—a popular privacy-preserving optimization method [23]—to federated fine-tuning of LLMs, primarily with LoRA. However, studies suggest that LoRA and DP are inherently misaligned, as injecting noise into the low-rank matrices A and B significantly slows convergence and degrades model accuracy [21], [24].

In this work, we study the federated and differentially private fine-tuning of LLMs via Adapters which is largely unexplored in the literature. Leveraging two widely used models from the BERT family, we demonstrate that Adapters achieve faster convergence and higher final accuracy on the GLUE benchmark [25] compared to LoRA. The remainder of the paper is structured as follows: Section II reviews PEFT methods, federated fine-tuning, and DP techniques. Section III provides an overview of LoRA and Adapter-based fine-tuning, DP-SGD, federated fine-tuning, and LoRA's incompatibility with DP. Section IV details our proposed algorithm, followed by performance evaluations in Section V.

II. Related Work

A. Parameter-Efficient Fine-Tuning (PEFT)

PEFT methods can generally be categorized into three distinct types [26]. The first category is Additive PEFT, where a small set of trainable parameters is incorporated into the model, and only these parameters are updated during training. Techniques like Serial Adapters [8], Parallel Adapters [27], Prefix-tuning [28], and Prompt-tuning [9] all belong to this group, and our approach follows a similar strategy. The second category is Selective PEFT, which focuses on tuning a specific subset of the model's existing parameters. Methods such as BitFit [29], U-Diff pruning [30], and PaFi [31] fall under this approach. Finally, Reparameterized PEFT introduces a low-rank parameterization of pre-trained weights for fine-tuning, exemplified by techniques such as LoRA [10] and DoRA [32].

In this work, we focus primarily on Serial Adapters (referred to as Adapters) and LoRA.

B. Federated Fine-Tuning

In their studies, [16], [17] evaluate and compare various PEFT methods, including Adapters, LoRA, Prompt Tuning, and BitFit, within the context of federated learning. Several adaptations of LoRA have been introduced to enhance its efficiency in highly heterogeneous federated settings. For instance, SLoRA [33], [34] modifies the initialization process to better handle data heterogeneity, while HetLoRA [35] and FlexLoRA [36] dynamically adjust LoRA ranks per client to account for system heterogeneity. More recently, FLoRA [37] introduces slack matrices A and B for all clients and multiplies the resulting matrices to mitigate interference caused by the FedAvg algorithm.

To reduce communication overhead in federated LoRA, [38] propose sparse fine-tuning techniques. Meanwhile, FFA-LoRA [21] and RoLoRA [39] aim to enhance model accuracy in heterogeneous environments while minimizing the number of trainable parameters. Additionally, FedTT [40] integrates tensorized adapters for federated fine-tuning, significantly reducing trainable parameters and improving communication efficiency.

Despite these advancements in addressing communication and data heterogeneity challenges, limited research has explored integrating DP with federated fine-tuning. Notably, [21], [22] examine DP in the context of LoRA, leaving a gap in DP-integrated approaches for other PEFT methods.

C. Differential Privacy and FL

In the context of federated learning, differential privacy protection is typically structured in two tiers based on whether the federated aggregation server is trusted by the clients. The first scenario assumes the server is trusted, with model updates sent to it without concerns for privacy. In this case, privacy is ensured through randomization in the final aggregated model output on the server side [41]. A more robust privacy approach eliminates the assumption of a trustworthy server, ensuring that the updates shared by each client are differentially private from the outset [42], [43]. This paper adopts this stronger privacy approach, guaranteeing that all information shared (i.e., model parameter updates) from local clients to the server complies with DP. Leveraging DP's intrinsic properties—such as parallel and sequential composition and resistance to post-processing [19], [23], [42]—the final model inherently satisfies global differential privacy.

III. Preliminaries

A. Fine-Tuning via Low-Rank Adaptation

LoRA is a fine-tuning method designed to efficiently adapt large pre-trained models while significantly reducing the number of trainable parameters. Instead of updating the entire weight matrix $W_0 \in \mathbb{R}^{d\times k}$, LoRA introduces a low-rank decomposition by reparameterizing the weight update as

$$W_0 + \Delta W = W_0 + BA,$$

where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$. By choosing a small rank $r \ll \min(d,k)$, LoRA reduces memory and computational costs while preserving the expressiveness of fine-tuning. During training, the pre-trained weights W_0 remain frozen, and only the newly introduced low-rank matrices A and B are updated. This approach is motivated by the observation that the intrinsic dimensionality of weight updates in large models is often much lower than their actual size, making low-rank adaptation an effective strategy.

B. Fine-Tuning via Adapters

Adapter-based fine-tuning, proposed by [8], modifies the architecture of a pre-trained model by introducing new adapter layers after each attention and feed-forward layer. These adapter layers function as bottleneck layers with residual connections. Specifically, given an input x, an adapter layer A operates as follows:

$$A(x) = U(\tau(D(x))) + x,$$

where $D \in \mathbb{R}^{r\times d}$ is a down-projection affine transformation, $U \in \mathbb{R}^{d\times r}$ is an up-projection affine transformation, and τ is a non-linear activation function. Here, d represents the input dimension, and $r \ll d$ controls the adapter's bottleneck size. By choosing a small r, adapter tuning significantly reduces the number of trainable parameters compared to full fine-tuning. During fine-tuning, the pre-trained model parameters remain frozen, and only the adapter layers, along with layer normalization parameters, are updated.

C. Fine-Tuning via DP-SGD

We start by recalling the formal definition of differential privacy.

Definition 3.1: A randomized algorithm $\mathcal{A}$ satisfies (ε, δ)-differential privacy if, for any two neighboring datasets D and D' that differ by at most one user's data, and for any subset $\mathcal{S}$ of possible outputs, the following holds:

$$\Pr[\mathcal{A}(D) \in \mathcal{S}] \leq e^{\varepsilon} \Pr\left[\mathcal{A}\left(D'\right) \in \mathcal{S}\right] + \delta.$$

Here, ε controls the privacy loss, with smaller values ensuring stronger privacy guarantees, while δ allows for a small probability of the condition being violated.

DP-SGD: Differentially Private Stochastic Gradient Descent (DP-SGD) [23] is an adaptation of the standard SGD algorithm that incorporates differential privacy guarantees. It accomplishes this by modifying the gradient computation process in two crucial ways:

- **Gradient Clipping:** To limit the influence of any individual data point on the model updates, each per-sample gradient is constrained to a predefined maximum norm C.
- **Noise Injection:** After computing the sum of clipped gradients over a batch $\mathcal{B}$ drawn from the dataset $\mathcal{D}$, Gaussian noise $z \sim \mathcal{N}\left(0, C^2\sigma^2 I\right)$ is added before performing the parameter update. This step obscures the precise contribution of any single data point, ensuring privacy.

The perturbed gradient used for model updates is computed as:

$$\bar{g} = \frac{\sum_{i \in \mathcal{B}} \text{Clip}(\nabla f, C) + z}{|\mathcal{B}|},$$

where f is the loss function and the noise scale σ is calibrated based on privacy parameters ϵ and δ, the total number of iterations T, and the sampling probability $q = |\mathcal{B}|/|\mathcal{D}|$, following standard privacy accounting mechanisms.

We now introduce fine-tuning via DP-SGD, as proposed by [44].

Consider a loss function $f(W_{\mathrm{PT}}, \theta)$, where W_{PT} represents the pre-trained weights, and θ denotes a set of additional trainable parameters. Crucially, the dimensionality of θ is much smaller than that of W_{PT}, i.e., $\dim(\theta) \ll \dim(W_{\mathrm{PT}})$, ensuring minimal parameter overhead during fine-tuning.

Fine-tuning is performed using DP-SGD on the additional parameters θ while keeping W_{PT} fixed. This method retains the knowledge embedded in the pre-trained model while enabling efficient adaptation to new tasks with significantly fewer trainable parameters.

D. Federated Fine-Tuning

Federated fine-tuning is a decentralized learning framework that enables a central server to collaboratively refine a global model with a network of N distributed clients, denoted as $\mathcal{C} = \{c_1, \ldots, c_N\}$. The objective is to optimize the trainable parameters θ by minimizing the following global objective function:

$$\min_{\theta} f(W_{PT}, \theta) = \frac{1}{N} \sum_{i=1}^{N} f_i(W_{PT}, \theta; \mathcal{D}_i),$$

where W_{PT} represents the fixed pre-trained model parameters shared across all clients, and each local objective function $f_i(\cdot)$ is expressed as:

$$f_i(W_{PT}, \theta) = \frac{1}{|\mathcal{B}_i|} \sum_{j=1}^{M} \ell_i(W_{PT}, \theta; x_{ij}),$$

where $\ell_i(\cdot)$ is the local loss function and $\mathcal{D}_i = \{x_{ij}\}_{j=1}^{M}$ represents the dataset specific to client c_i.

A widely adopted approach in federated learning is the FedAvg algorithm [15]. In each training round, client c_i initializes its local model with the global parameters from the previous round $\theta^{(t)}$ and performs local training using stochastic gradient descent (SGD) for K local updates:

$$\theta_i^{(t)+k+1} \longleftarrow \theta_i^{(t)+k} - \eta \nabla f_i(W_{PT}, \theta_i^{(t)+k}),$$

where η is the learning rate and $\theta_i^{(t)+k}$ represents the local model parameters of client c_i at communication round t and local update step k. Once local training is complete, clients transmit their updated model parameters to the server, which then aggregates the updates to form the new global model:

$$\theta^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} \theta_i^{(t)+K}.$$

This iterative process continues until convergence, allowing the global model to be progressively refined through decentralized client updates while ensuring data privacy.

E. LoRA: Discordance with DP

In this section, we discuss the problem of LoRA and DP. Then, in Section IV, we explain how fine-tuning via Adapters can help mitigate this issue. Under DP-SGD, the updates for matrices A and B in LoRA are perturbed by noise terms ξ_A and ξ_B, respectively. In this way, we can write

$$(B + \xi_B)(A + \xi_A)x = BAx + B\xi_A x + \xi_B Ax + \xi_B \xi_A x.$$

The first-order noise term, $\xi_B Ax + B\xi_A x$, is expected and occurs even in full FT with DP-SGD. However, the second-order noise term, $\xi_B \xi_A$, causes noise amplification, leading to further performance degradation in LoRA-based methods [21]. This issue is exacerbated in FL, as individual client updates deviate even further from the ideal global update.

IV. Proposed Algorithm

In this section, we first define and describe the differentially private and federated fine-tuning algorithm in Alg. 1, which can be applied to any fine-tuning method, including Adapters and LoRA. We then explain why fine-tuning via Adapters may offer advantages.

A. Algorithm description

The algorithm operates over T communication rounds, where each client i performs K local updates on its trainable parameters while ensuring differential privacy.

- **Initialization:** We initialize the hyperparameters, including the number of local updates K, learning rate η, the pre-trained language model W_{PT} (RoBERTa or DeBERTa), the trainable layers θ, the number of clients N, noise scale σ, and privacy budget (ϵ, δ). The server then randomly initializes the global model's trainable layers θ and distributes them among the clients. Each client loads the pre-trained model W_{PT} and tokenizer to preprocess the input text (Lines 1-3 of Alg. 1).
- **Local Training at Each Client:** In each communication round, each client updates the model locally before sending updates to the server. During training, the client performs K local updates using its private dataset. In each local step, a mini-batch B_i is randomly sampled from the client's dataset D_i, and the gradients of the loss function are computed for the trainable parameters θ_i. Using gradient clipping and noise injection can achieve differential privacy by obscuring individual data contributions. The differentially private gradient $\bar{g}_i^{(t,k)}$ is computed by aggregating all clipped gradients and adding Gaussian noise. The model parameters are then updated using these gradients (Lines 4-11 of Alg. 1).

- **Model Transmission:** After completing K local updates, each client sends its differentially private model parameters $\theta_i^{(t,K)}$ to the central server. The client update process is completed for this communication round (Lines 12-13 of Alg. 1).
- **Federated Aggregation:** The global model aggregates client updates using Federated Averaging, meaning that the new global model is computed as the mean of client updates. The updated global model is redistributed to all clients for the next round (Line 14 of Alg. 1).
- **Global Model Update and Distribution:** The process continues until the model converges, achieving optimal performance while maintaining privacy guarantees. The whole training process is completed. The final model is deployed for downstream tasks (Line 15 of Alg. 1).

Algorithm 1 Differentially Private and Federated Fine-Tuning

```
1: Input: Number of clients $N$, global model $\theta$, privacy budget $(\epsilon, \delta)$, learning rate $\eta$, number of local updates $K$
2: Initialize: Server initializes global model $\theta^{(0)}$ randomly
3: Loading pre-trained model and tokenizer at each client
4: for each communication round $t = 1$ to $T$ do
5:     for user $i$ do
6:         for each local update step $k = 1$ to $K$ do
7:             Sample batch: $\mathcal{B}_i \sim \mathcal{D}_i$
8:             Compute gradients: $\nabla f_i(W_{PT}, \theta_i^{(t)+k}; \mathcal{B}_i)$
9:             Clip gradients and Add noise:
                   $$\bar{g}_i^{(t)+k} = \frac{\sum_{i \in \mathcal{B}_i} \text{Clip}(\nabla f_i, C) + \mathcal{N}(0, C^2\sigma^2 I)}{|\mathcal{B}_i|}$$
10:            Update model parameters:
                   $$\theta_i^{(t)+k+1} \longleftarrow \theta_i^{(t)+k} - \eta \bar{g}_i^{(t)+k}$$
11:        end for
12:        Each user $\theta_i^{(t)+K}$ to the central server
13:    end for
14:    Global Model Update:
           $$\theta^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} \theta_i^{(t)+K}$$
       Server sends $\theta^{(t+1)}$ to all clients
15: end for
```

B. Fine-Tuning via Adapters

In this section, we present our intuition behind why Adapters should perform better in federated and DP fine-tuning. When applying DP-SGD to Adapters, we have

$$(W_{\text{up}} + \xi_u)\, \tau\left((W_{\text{down}} + \xi_d)\, x\right).$$

The activation function τ in the above expression is typically the ReLU activation function. ReLU maps negative inputs to zero, which, to some extent, helps mitigate the noise amplification problem discussed in Section III-E. We hypothesize that, unlike the standard non-DP setting where LoRA generally outperforms Adapters (as shown in Table 2 of [40] and Table 1 of [11]), Adapters can achieve superior performance under DP constraints. We will empirically validate this hypothesis in the next section.

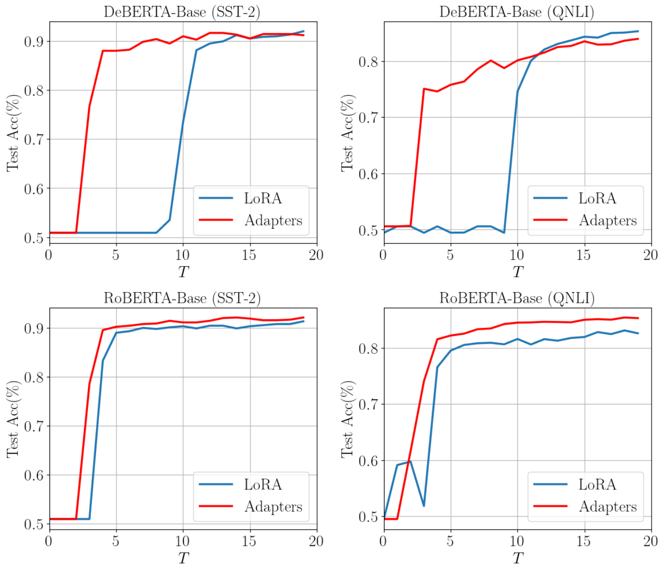

Fig. 1: Convergence speed for SST-2 and QNLI datasets using DeBERTA-Base and RoBERTA-Base models.

V. Numerical Results

We conduct extensive experiments to evaluate the performance of the proposed algorithms on two language models: RoBERTa-base [45] and DeBERTa-base [46]. Our experiments utilize the MRPC, SST-2, QNLI, QQP, and MNLI datasets from the Generalized Language Understanding Evaluation (GLUE) benchmark [25]. We use the full training dataset for each task and report the best validation accuracy.

A. Convergence Speed

In this section, we compare the convergence speed of LoRA and Adapter methods in our setting on the SST-2 and QNLI datasets using DeBERTa-Base and RoBERTa-Base models. We set $N = 10$, $K = 5$, $T = 20$, $\epsilon = 6.7$, $\delta = 1e - 5$, and $C = 2$. As shown in Fig. 1, fine-tuning with Adapters achieves faster convergence for both models and datasets. Notably, for the DeBERTa-Base model, the convergence is significantly faster.

B. Comparison of Accuracies

To further compare the accuracy of fine-tuning using Adapters versus LoRA, we present results under varying privacy budgets on the SST-2, QNLI, and MNLI datasets. The results are provided in Tables I and II. As shown, on average, fine-tuning with Adapters achieves better performance with both DeBERTa-Base and RoBERTa-Base models.

TABLE I: Comparative analysis of Adapters and LoRA methods using the RoBERTa-Base model.

Priv. Budget	Model & Method	SST-2	QNLI	MNLI	Avg.
$\epsilon = 1$	RoBERTa-Base (LoRA)	90.96	82.57	76.99	83.51
	RoBERTa-Base (Adapter)	**91.74**	**83.82**	**77.82**	**84.46**
$\epsilon = 3$	RoBERTa-Base (LoRA)	91.29	83.40	**77.21**	83.97
	RoBERTa-Base (Adapter)	**92.76**	**84.89**	75.97	**84.54**
$\epsilon = 6.7$	RoBERTa-Base (LoRA)	91.74	84.38	**78.18**	84.77
	RoBERTa-Base (Adapter)	**92.19**	**85.56**	77.17	**84.97**
$\epsilon = inf$	RoBERTa-Base (LoRA)	**95.20**	**92.50**	-	**93.85**
	RoBERTa-Base (Adapter)	94.98	92.20	-	93.59

TABLE II: Comparative analysis of Adapters and LoRA methods using the DeBERTa-Base model.

Dataset	Model & Method	$\epsilon = 0.5$	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 6.7$	$\epsilon = inf$	Avg.
SST-2	DeBERTa-Base (LoRA)	90.11	90.91	**91.93**	92.05	**96.43**	92.29
	DeBERTa-Base (Adapter)	**90.23**	**91.14**	91.59	**92.16**	**96.43**	**92.31**
QNLI	DeBERTa-Base (LoRA)	-	75.06	81.84	**85.30**	**94.20**	84.10
	DeBERTa-Base (Adapter)	-	**81.12**	**82.42**	83.95	94.14	**85.41**

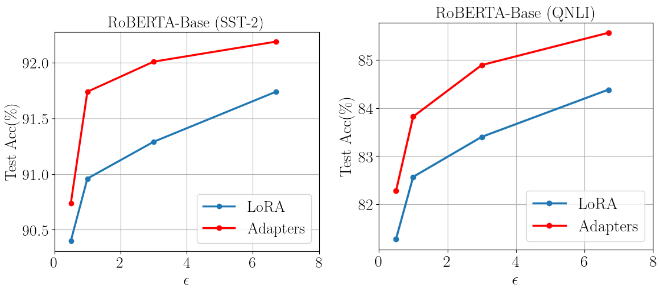

Fig. 2: Test accuracy versus ϵ for SST-2 and QNLI datasets using RoBERTA-Base model.

VI. Conclusion

In this work, we explored the intersection of federated learning, differential privacy, and parameter-efficient fine-tuning for large language models. While prior research has primarily focused on applying LoRA in differentially private federated fine-tuning, we demonstrated that Adapters offer a compelling alternative. Through extensive empirical evaluations, we showed that Adapters can mitigate the convergence and accuracy issues associated with LoRA in differentially private settings and achieve superior performance. These findings suggest that Adapters can be a more effective choice for privacy-preserving distributed training of LLMs. Future research could further optimize Adapter-based approaches, explore alternative PEFT methods in privacy-sensitive scenarios, and investigate their deployment in real-world applications.

A. Limitations and Future Work

Due to constraints in resources and time, the results presented in Section V are based on a single experimental run, which may limit reproducibility and statistical significance. Future work should include multiple runs with different random seeds to ensure robustness.

Federated learning introduces challenges due to data heterogeneity. Unlike traditional centralized training, where data is typically independently and identically distributed (IID), federated learning often involves non-IID data, meaning client datasets may follow different distributions. This variation can lead to conflicting local model updates, making it difficult for the global model to generalize effectively. Future research can explore ways to optimize Adapters to better handle non-IID data and improve performance across diverse clients.

This study primarily focuses on classification tasks, leaving the effectiveness of Adapter-based fine-tuning on generative tasks unexplored. Extending evaluations to tasks such as text generation and summarization could provide deeper insights into the adaptability of Adapters in broader NLP applications.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

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

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[4] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[5] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[6] H. E. Oskouie, C. Chance, C. Huang, M. Capetz, E. Eyeson, and M. Sarrafzadeh, "Leveraging large language models and topic modeling for toxicity classification," *Workshop on Computing, Networking and Communications (CNC)*, pp. 123–127, 2025.

[7] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[8] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International conference on machine learning*. PMLR, 2019, pp. 2790–2799.

[9] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 3045–3059.

[10] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[11] Y. Yang, J. Zhou, N. Wong, and Z. Zhang, "LoRETTA: low-rank economic tensor-train adaptation for ultra-low-parameter fine-tuning of large language models," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024, pp. 3161–3176.

[12] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen *et al.*, "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature Machine Intelligence*, vol. 5, no. 3, pp. 220–235, 2023.

[13] A. Manoel, M. d. C. H. Garcia, T. Baumel, S. Su, J. Chen, R. Sim, D. Miller, D. Karmon, and D. Dimitriadis, "Federated multilingual models for medical transcript analysis," in *Conference on Health, Inference, and Learning*. PMLR, 2023, pp. 147–162.

[14] O. B. Shoham and N. Rappoport, "Federated learning of medical concepts embedding using behrt," *arXiv preprint arXiv:2305.13052*, 2023.

[15] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

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

[17] T. Fan, Y. Kang, G. Ma, W. Chen, W. Wei, L. Fan, and Q. Yang, "Fate-llm: A industrial grade federated learning framework for large language models," *arXiv preprint arXiv:2310.10049*, 2023.

[18] H. Zhao, W. Du, F. Li, P. Li, and G. Liu, "Fedprompt: Communication-efficient and privacy-preserving prompt tuning in federated learning," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[19] C. Dwork, "Differential privacy," in *International colloquium on automata, languages, and programming*. Springer, 2006, pp. 1–12.

[20] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[21] Y. Sun, Z. Li, Y. Li, and B. Ding, "Improving LoRA in privacy-preserving federated learning," *arXiv preprint arXiv:2403.12313*, 2024.

[22] X.-Y. Liu, R. Zhu, D. Zha, J. Gao, S. Zhong, M. White, and M. Qiu, "Differentially private low-rank adaptation of large language model using federated learning," *ACM Transactions on Management Information Systems*, 2024.

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

[24] R. Singhal, K. Ponkshe, R. Vartak, L. R. Varshney, and P. Vepakomma, "Fed-sb: A silver bullet for extreme communication efficiency and performance in (private) federated lora fine-tuning," *arXiv preprint arXiv:2502.15436*, 2025.

[25] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.

[26] Z. Han, C. Gao, J. Liu, S. Q. Zhang *et al.*, "Parameter-efficient fine-tuning for large models: A comprehensive survey," *arXiv preprint arXiv:2403.14608*, 2024.

[27] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, "Towards a unified view of parameter-efficient transfer learning," *arXiv preprint arXiv:2110.04366*, 2021.

[28] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *arXiv preprint arXiv:2101.00190*, 2021.

[29] 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.

[30] D. Guo, A. M. Rush, and Y. Kim, "Parameter-efficient transfer learning with diff pruning," *arXiv preprint arXiv:2012.07463*, 2020.

[31] B. Liao, Y. Meng, and C. Monz, "Parameter-efficient fine-tuning without introducing new latency," *arXiv preprint arXiv:2305.16742*, 2023.

[32] S.-Y. Liu, C.-Y. Wang, H. Yin, P. Molchanov, Y.-C. F. Wang, K.-T. Cheng, and M.-H. Chen, "Dora: Weight-decomposed low-rank adaptation," *arXiv preprint arXiv:2402.09353*, 2024.

[33] S. Babakniya, A. R. Elkordy, Y. H. Ezzeldin, Q. Liu, K.-B. Song, M. El-Khamy, and S. Avestimehr, "SLoRA: federated parameter efficient fine-tuning of language models," *arXiv preprint arXiv:2308.06522*, 2023.

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

[35] Y. J. Cho, L. Liu, Z. Xu, A. Fahrezi, M. Barnes, and G. Joshi, "Heterogeneous lora for federated fine-tuning of on-device foundation models," in *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*, 2023.

[36] J. Bai, D. Chen, B. Qian, L. Yao, and Y. Li, "Federated fine-tuning of large language models under heterogeneous language tasks and client resources," *arXiv preprint arXiv:2402.11505*, 2024.

[37] 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.

[38] K. Kuo, A. Raje, K. Rajesh, and V. Smith, "Federated lora with sparse communication," *arXiv preprint arXiv:2406.05233*, 2024.

[39] S. Chen, Y. Ju, H. Dalal, Z. Zhu, and A. J. Khisti, "Robust federated finetuning of foundation models via alternating minimization of LoRA," in *Workshop on Efficient Systems for Foundation Models II@ ICML2024*.

[40] S. Ghiasvand, Y. Yang, Z. Xue, M. Alizadeh, Z. Zhang, and R. Pedarsani, "Communication-efficient and tensorized federated fine-tuning of large language models," *arXiv preprint arXiv:2410.13097*, 2024.

[41] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv preprint arXiv:1710.06963*, 2017.

[42] Z. Li, B. Ding, C. Zhang, N. Li, and J. Zhou, "Federated matrix factorization with privacy guarantee," *Proceedings of the VLDB Endowment*, vol. 15, no. 4, 2021.

[43] N. Wu, F. Farokhi, D. Smith, and M. A. Kaafar, "The value of collaboration in convex machine learning with differential privacy," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 304–317.

[44] D. Yu, S. Naik, A. Backurs, S. Gopi, H. A. Inan, G. Kamath, J. Kulkarni, Y. T. Lee, A. Manoel, L. Wutschitz *et al.*, "Differentially private fine-tuning of language models," in *International Conference on Learning Representations*.

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

[46] P. He, X. Liu, J. Gao, and W. Chen, “Deberta: Decoding-enhanced bert with disentangled attention,” in *International Conference on Learning Representations*, 2020.