ADP-LoRA: Privacy-Preserving Low-Rank Fine-Tuning of Transformers with Adaptive Differential Privacy

Rohit Mahesh
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai 600127, India
rohit.m2022@vitstudent.ac.in

Karthika Veeramani
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai 600127, India
karthika.v@vit.ac.in

***Abstract*—Privacy-preserving fine-tuning of large language models (LLMs) is crucial for sensitive applications such as healthcare, finance, and education. However, differentially private stochastic gradient descent (DP-SGD) often introduces significant utility loss due to clipping and random noise injection, particularly when applied to transformer models like BERT.To address this gap,ADP-LoRA, a fine-tuning framework that combines LoRA with adaptive per-layer noise scaling, noise-aware pruning, and a feedback controller with rollback is proposed. This framework continuously audits gradient distributions using Wasserstein and Kolmogorov–Smirnov proxies, while a discriminator network estimates residual leakage risks. A control mechanism adjusts the privacy noise multiplier in real time, and a rollback safeguard reduces noise when validation loss worsens beyond a threshold. ADP-LoRA is evaluated on BERT for text classification, achieving a final accuracy of 77.9% under an $\epsilon \approx 6.4$ privacy budget. This reduces the typical 15% accuracy gap between DP and non-DP models by over 7 points. Compared with baseline DP-SGD, the proposed approach demonstrates improved stability, better convergence with large batch sizes. This framework balances strong differential privacy guarantees with competitive model accuracy, making it a practical approach for privacy-sensitive Natural Language Processing(NLP) tasks.**

***Index Terms*—Differential Privacy, LoRA, Adaptive Noise Control, BERT, Text Classification, Privacy-Preserving NLP**

I. Introduction

Large Language Models (LLMs) such as BERT , RoBERTa, and GPT variants [1] have become indispensable in natural language processing (NLP). Their deep contextual representations allow them to capture semantic and syntactic patterns at scale, leading to state-of-the-art performance across a variety of tasks including sentiment analysis, text classification, question answering, and machine translation. As these models mature, they are no longer confined to research laboratories but are increasingly deployed in production environments that interact with real users and sensitive data. The integration of LLMs into high-stakes domains such as healthcare, finance, and education highlights the growing importance of responsible deployment. In these sectors, training data often contains confidential or personally identifiable information. Even when raw data is not directly exposed, recent research has shown that models can inadvertently leak sensitive details through gradients, memorized examples, or adversarial probing . This makes the protection of individual-level privacy during fine-tuning a critical challenge for the next generation of NLP systems.

Differential Privacy (DP) has emerged as a mathematically rigorous framework [16] to address this concern. By introducing calibrated noise into the training process, DP ensures that the presence or absence of any single individual in the dataset cannot be confidently inferred, thus providing formal guarantees of privacy . However, while effective, traditional DP-SGD methods are known to degrade model utility, especially for large transformer-based architectures where gradient clipping [18] and noise injection disrupt the delicate optimization dynamics. The result is often a significant drop in accuracy compared with non-DP training, limiting the practical adoption of DP in applied NLP.At the same time, parameter-efficient fine-tuning (PEFT) methods have gained attraction as an alternative to full-model training. Among these, Low-Rank Adaptation (LoRA) [17] has shown particular promise by introducing lightweight, low-rank trainable matrices into transformer attention layers. This approach drastically reduces the number of parameters that need to be updated, enabling faster training, lower memory consumption, and improved adaptability to new domains. Importantly, LoRA's smaller trainable footprint makes it especially attractive in the context of differential privacy, since fewer parameters are subject to noise perturbation.

The convergence of differential privacy and parameter-efficient fine-tuning opens new opportunities for privacy-preserving NLP. The objective of this study is to develop ADP-LoRA, an adaptive framework that unifies DP with LoRA by integrating per-layer noise control, discriminator-augmented auditing, and dynamic rollback mechanisms. Using BERT as a representative LLM for text classification, we demonstrate that ADP-LoRA mitigates the accuracy loss of DP-SGD while providing rigorous differential privacy guarantees.

979-8-3315-7120-7/25/$31.00 © 2025 IEEE

II. Related Works

Privacy-preserving fine-tuning of large language models (LLMs) has evolved across several dimensions, including centralized differential privacy (DP), federated learning (FL) with low-rank adapters, communication- and resource-efficient training, and secure inference techniques. Within this landscape, parameter-efficient fine-tuning (PEFT) methods such as LoRA have become especially relevant, as they reduce trainable parameters while supporting privacy-aware optimization.Grouped thematically, the following works provide the foundation upon which ADP-LoRA is developed.

A. Centralized Differential Privacy for LLM Fine-Tuning

Rouzbeh Behnia *et al.* introduced EW-Tune, a framework for privately fine-tuning LLMs with a refined privacy accountant and low-iteration training, improving the DP utility trade-off across NLP tasks. Its low-iteration design, however, may struggle on complex tasks requiring deeper convergence [2]. Meng Tong *et al.* proposed InferDPT, which ensures inference-time privacy for black-box LLMs by protecting queries and outputs without re-training. While effective in such settings, it does not mitigate data leakage risks in collaborative training [3]. Kasra Ahmadi and Rouzbeh Behnia *et al.* presented Interactive PP-FL, providing procedural guidance for privacy-preserving federated fine-tuning of LLMs. Although beneficial for reproducibility and centralized DP setups, scaling across heterogeneous clients remains a challenge [4].

B. LoRA–Centric Federated Fine-Tuning and Personalization

Meilu Zhu *et al.* proposed DEeR, which mitigates aggregation deviation and noise amplification in federated LoRA under DP, though its complexity hinders large-scale use [5]. Kang Li *et al.* introduced PFFLoRA, leveraging Fourier parameterization for personalization, but its mapping may limit generalization across diverse clients [6]. Zikai Zhang *et al.* presented FedHeLLo, allocating heterogeneous LoRA capacity for efficient tuning, though fairness issues arise with unequal client resources [7]. Zhaoyang Dong *et al.* proposed FedALoRA, balancing personalization and global performance, yet trade-offs between the two remain [8]. Xinzhi Yi *et al.* showed that Aggregating Low-Rank Adapters preserves PEFT gains under distribution shift, but struggles with highly divergent client data [9]. Zhidong Gao *et al.* introduced Federated Adaptive Fine-Tuning with Quantization+LoRA, improving resource use but adding tuning overhead [10].

C. Communication and Systems Efficiency for FL with LLMs

Sumudith Sadeepa *et al.* developed DisLLM to ensure privacy under resource constraints via distributed execution, but practical bottlenecks remain when scaling private LLM training [11].Chuantao Li *et al.* studied Federated Transfer Learning for On-Device LLMs, applying transfer strategies to reduce compute and bandwidth, though edge-device variability limits consistency [12].Shivani Sanjay Kolekar *et al.* proposed Resource-Efficient Federated Fine-Tuning for Speech, adapting LLM fine-tuning to speech tasks under tight compute and communication budgets, yet its generalization beyond speech remains limited [13].

D. Privacy-Preserving FL Strategies and Verification

Bowen Li *et al.* introduced Knowledge-Distillation-Based Federated Fine-Tuning, transferring knowledge to lightweight student models to cut on-device costs while preserving accuracy, but its success depends heavily on teacher quality [14].Tianyou Zhang *et al.* presented LaVFL, a verifiable FL framework for LLM fine-tuning that integrates integrity checks into privacy-preserving protocols, though verification steps increase system latency [15].

E. Research Gap

Despite advances in DP for LLM fine-tuning, LoRA-based PEFT, and federated efficiency and privacy strategies, most methods still assume fixed noise schedules and overlook per-layer scaling, noise-aware pruning, and closed-loop auditing. To address this, we propose ADP-LoRA, which integrates LoRA with adaptive per-layer noise, discriminator-augmented leakage detection using KS and Wasserstein distance, and a feedback controller that dynamically adjusts the global noise with rollback to balance privacy and accuracy.

III. Methodology

This section presents the end-to-end methodology of the proposed ADP-LoRA framework,with the overall architecture illustrated in Fig. 1. The pipeline consists of four main stages: dataset preparation and preprocessing, model architecture and LoRA integration, differential privacy configuration, and adaptive privacy control. Each stage builds on the previous one, ensuring both rigorous privacy guarantees and strong downstream task performance.

A. Dataset and Preprocessing

The Stanford Natural Language Inference (SNLI) dataset [19], a benchmark corpus containing 570k sentence pairs annotated with one of three labels entailment, contradiction, or neutral is used. An illustrative example is shown in Table I. The dataset is split into training, development, and test sets.For preprocessing, `BERT-base-cased` tokenizer is emplopyed to tokenize both sentences. The tokens are truncated or padded to a maximum sequence length of 128. For each pair, This framework constructs:

- **Input IDs**: Tokenized indices padded to fixed length.
- **Attention Mask**: Binary mask distinguishing real tokens from padding.
- **Token Type IDs**: Segment embeddings indicating premise vs hypothesis.
- **Label**: Gold label mapped to integer indices.

The processed inputs are converted into PyTorch `TensorDataset` objects, enabling efficient batching.The batch size is set to 64 (with a physical batch size of 8 using gradient accumulation for GPU memory efficiency).

B. Model Architecture and LoRA Integration

A pre-trained `BERT-base-cased` model is fine-tuned for sequence classification. Instead of updating all 108M parameters, Low-Rank Adaptation (LoRA) is adopted to reduce trainable parameters to ∼1M. LoRA introduces low-rank decomposition matrices into the attention layers, enabling efficient fine-tuning.

Specifically, LoRA with rank $r = 32$, scaling $\alpha = 32$, and dropout of 0.05 is applied. To balance adaptation and stability, the lower encoder layers is freezed and only the last six attention layers are finetuned, the pooler, and classifier. This selective unfreezing significantly reduces overfitting and makes the model more privacy-friendly by restricting the number of trainable parameters exposed to noise perturbations.

C. Differential Privacy with Opacus

To protect individual-level privacy, the framework integrates differential privacy (DP) using the Opacus Engine as shown in Fig. 2. The DP-SGD framework is deployed, which enforces privacy through:

- **Gradient Clipping**: Each per-sample gradient is clipped to a maximum ℓ_2 norm of 2.0, limiting sensitivity.
- **Noise Injection**: Gaussian noise, calibrated to privacy budget (ϵ, δ), is added to clipped gradients.
- **Privacy Accounting**: Opacus tracks cumulative privacy loss, ensuring the target $\epsilon = 8.0$ is respected across 3 training epochs with $\delta = \frac{1}{N}$.

Learning rate of 5×10^{-4} with AdamW optimizer and cosine annealing schedule for stability is used.

D. Adaptive Noise Control Framework

Beyond standard DP-SGD, our approach introduces several innovations to balance privacy and accuracy:

- **Per-layer adaptive noise scaling**: Noise is scaled dynamically per LoRA layer based on gradient norms, ensuring layers with stronger signals are preserved while weaker updates receive higher perturbation.
- **Noise-aware pruning**: Gradients with low signal-to-noise ratio (SNR) are filtered, avoiding wasteful updates dominated by noise.
- **Dynamic privacy auditing**: Gradient distributions are continuously monitored using Wasserstein distance and Kolmogorov–Smirnov (KS) proxies, which serve as lightweight indicators of privacy leakage.
- **Discriminator risk estimation**: A lightweight neural discriminator attempts to distinguish between real and noisy gradients, providing an adversarial measure of residual leakage risk.

Algorithm 1: ADP-LoRA Training Framework with Adaptive Noise Control

Input: Training dataset D, pre-trained Transformer M, privacy budget (ϵ, δ), LoRA rank r
Output: Privacy-preserving fine-tuned model M^*

Step 1: Data Preprocessing
Download and extract SNLI dataset → filter malformed pairs
Tokenize premise and hypothesis sentences with BERT tokenizer
Pad/Truncate to maximum sequence length $L = 128$
Convert to tensors
$\{input\_ids, attention\_mask, token\_type\_ids, labels\}$
Build PyTorch datasets and mini-batches with size B (64/128)

Step 2: Initialize Model with LoRA
$M \leftarrow$ BERT-base encoder with sequence classification head
Insert LoRA adapters $(r = 32, \alpha = 32, p = 0.05)$ into attention layers
Freeze encoder layers except last k layers (here $k = 4$)
$M_{LoRA} \leftarrow$ Parameter-efficient fine-tuning model

Step 3: Attach Differential Privacy Engine
Configure Opacus PrivacyEngine with
– gradient clipping bound C
– initial noise multiplier $\sigma_0 = 1.0$
– target budget $(\epsilon = 8.0, \delta = 1/|D|)$
Wrap $(M_{LoRA}, Optimizer, DataLoader)$ for DP-SGD

Step 4: Adaptive Privacy Control Mechanisms
This is the novel contribution of ADP-LoRA
(a) **Per-layer adaptive noise scaling**: adjust σ_ℓ based on gradient norms of LoRA parameters.
(b) **Noise-aware pruning**: discard updates with low signal-to-noise ratio (SNR).
(c) **Dynamic privacy auditing**: estimate leakage using Wasserstein distance and KS-proxies between real vs noisy gradients.
(d) **Discriminator network**: train a small classifier to distinguish real vs noisy gradients; output $\in [0, 1]$ indicates leakage risk.
(e) **Feedback controller**: update $\sigma \leftarrow \sigma \cdot e^{\eta(\hat{r} - r^*)}$ with learning rate η, where $\hat{r}$ = observed risk, r^* = target risk.
(f) **Rollback safeguard**: if validation loss increases by $> 5\%$, reduce $\sigma \leftarrow \max(\sigma \cdot 0.9, \sigma_{min})$.

Step 5: Training Loop
for *epoch* $= 1$ **to** E **do**
  for *each mini-batch* $b \in D$ **do**
    Compute loss $\mathcal{L}(M_{LoRA}(b))$
    Backpropagate, apply per-layer clipping, add Gaussian noise $\mathcal{N}(0, \sigma^2 C^2)$
    Update parameters using AdamW
    Every T steps: run audit_step(), update controller, check rollback
  end
  Evaluate on validation set, record $(loss, accuracy, \epsilon)$
end

Return: Fine-tuned model M^* with DP guarantees

- **Feedback controller**: An exponential moving average (EMA) controller adjusts the privacy noise multiplier σ in real time to maintain a target risk level.
- **Rollback safeguard**: If validation loss worsens by more than 5% relative to the best checkpoint, σ is reduced by 10%, preventing degradation in utility.

Together, these mechanisms create a closed-loop adaptive control system that integrates privacy auditing, noise regulation, and utility preservation. This makes DP training more stable and less prone to accuracy collapse.

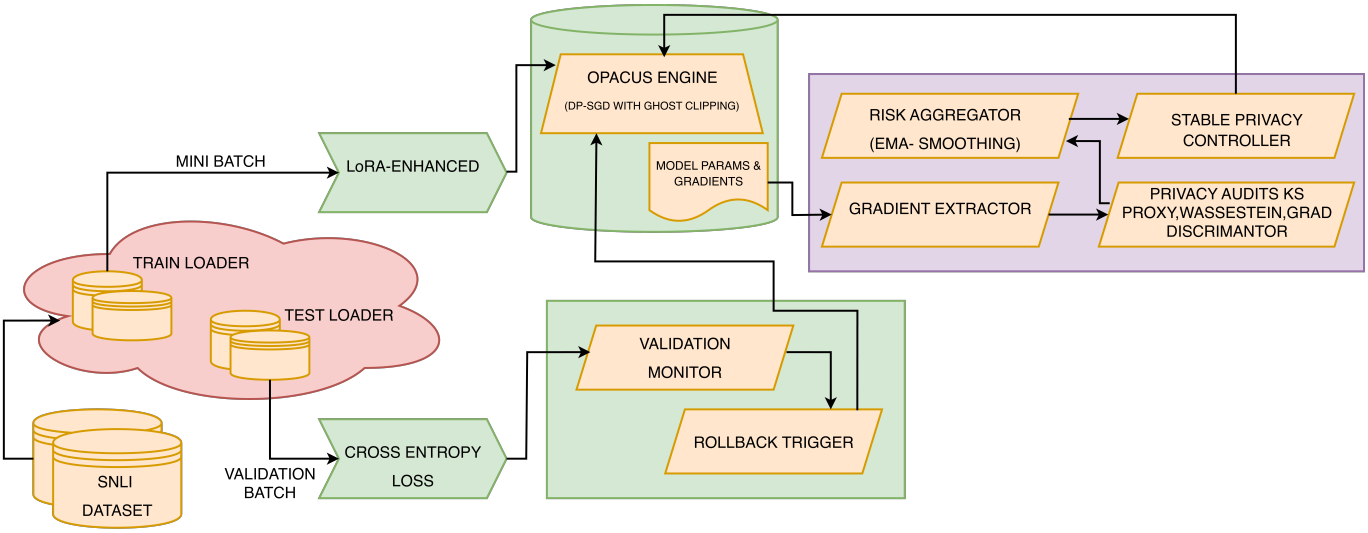

Fig. 1. ADP-LORA Architecture

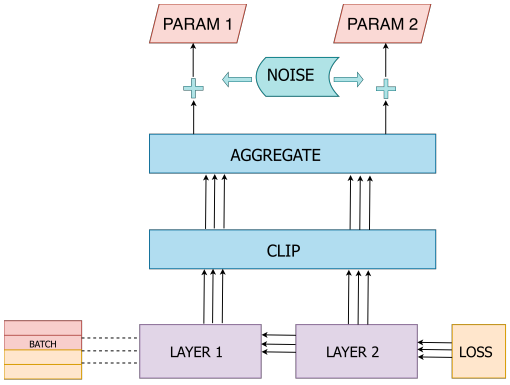

Fig. 2. Opacus Engine (DP-SGD)

TABLE I
SAMPLE SNLI DATASET EXAMPLES

Sentence 1	Sentence 2	Gold Label
A person on a horse jumps over a broken down airplane.	A person is training his horse for a competition.	Neutral
A person on a horse jumps over a broken down airplane.	A person is at a diner, ordering an omelette.	Contradiction
A person on a horse jumps over a broken down airplane.	A person is outdoors, on a horse.	Entailment

E. Training Setup

The model is trained with mixed precision (FP16) to improve computational efficiency and memory utilization. All experiments are conducted on a single NVIDIA A100 GPU (40 GB) using PyTorch 2.1 and Opacus 1.5. The SNLI dataset, containing approximately 570k annotated sentence pairs, is preprocessed using the bert-base-cased tokenizer and split into 550k, 10k, and 10k samples for training, validation, and testing, respectively. Each experiment is repeated three times with different random seeds, and the mean accuracy (with standard deviation $<0.3\%$) is reported for reproducibility.The key hyperparameters are summarized in Table II.

TABLE II
TRAINING HYPERPARAMETERS

Parameter	Value
Batch size	64 (with gradient accumulation for effective 128)
Epochs	3
Learning rate	5×10^{-4} with cosine decay
DP parameters	$(\epsilon = 8.0, \delta = 1/N)$

A batch size of 64 (with gradient accumulation to achieve an effective batch size of 128) balances computational efficiency with differential privacy noise stability. The cosine decay learning rate schedule provides smoother convergence, while the differential privacy parameters $(\epsilon = 8.0, \delta = 1/N)$ ensure strong privacy guarantees. Noise control and auditing mechanisms are executed every 1000 steps, and both accuracy and cumulative privacy cost are logged after each epoch. The detailed procedure is described in Algorithm 1.

IV. RESULTS AND DISCUSSION

To evaluate the effectiveness of ADP-LoRA,extensive experiments on the SNLI dataset were conducted, measuring model accuracy, stability of gradient norms, and the privacy–utility trade-off across training steps. The outcomes are summarized below.Fig. 3 shows the accuracy trajectory across training steps. Both models exhibit a sharp rise in the early stages, but ADP-LoRA consistently maintains higher accuracy. The Non-DP model improves gradually, reaching around 0.74 after 45,000

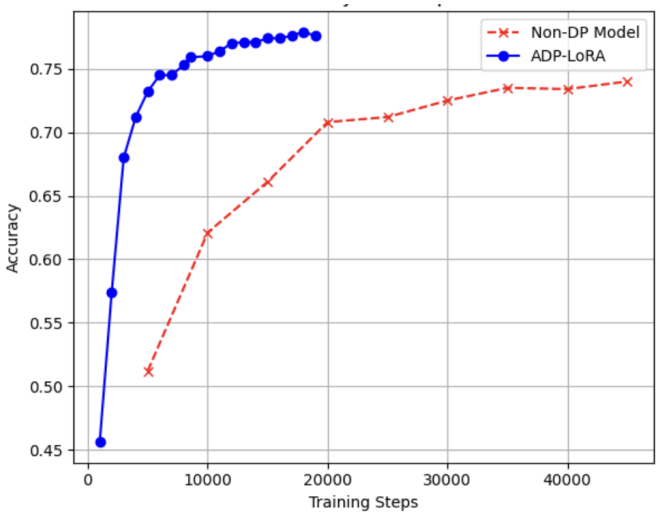

Fig. 3. ADP-LoRA vs Non-DP over Accuracy vs Steps

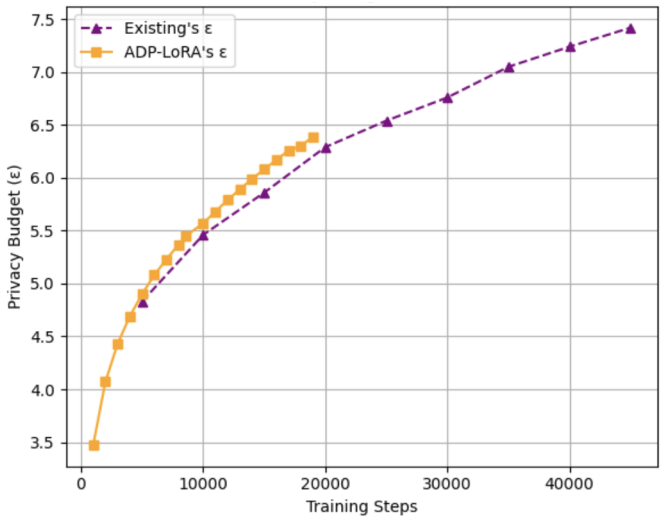

Fig. 4. ADP-LoRA vs Non-DP over Privacy-Budget vs Steps

steps, whereas ADP-LoRA achieves faster convergence, stabilizing in the range of 0.77-0.78 by 20,000 steps. This indicates that ADP-LoRA not only closes the gap with the Non-DP model but also sustains an accuracy margin of roughly +3-4 percentage points throughout training.

Fig. 4 presents the privacy budget growth (ϵ) over the course of training. Both methods display a monotonic increase, with ϵ rising from 3.5 to above 7.0. However, ADP-LoRA demonstrates a slower rate of privacy consumption in the later stages, maintaining ϵ near 6.4 when the Non-DP model exceeds 7.3. This highlights the effectiveness of adaptive noise regulation in balancing privacy expenditure. The accuracy-privacy trade-off is illustrated in Fig. 5. The Non-DP model follows a steady trajectory where higher ϵ is associated with incremental accuracy gains. In contrast, ADP-LoRA achieves superior accuracy across the same ϵ range, demonstrating that adaptive fine-

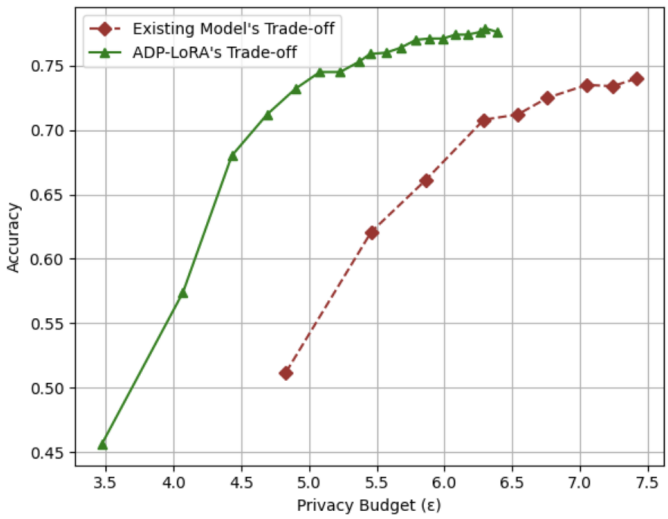

Fig. 5. ADP-LoRA vs Non-DP over Accuracy vsPrivacy-Budget

tuning enables better utility preservation without disproportionately increasing privacy cost.Table III provides selected training step statistics, confirming that Wnorm values remained bounded (0.048–0.054) throughout, while the discriminator risk (Dr) only increased at higher steps (80,000), where the privacy budget accumulated to its peak.

TABLE III
TRAINING LOGS

Steps	KSr	Wnorm	Dr
1000	1.0	0.053	0.0
2000	1.0	0.053	0.0
3000	1.0	0.051	0.0
4000	1.0	0.048	0.0
5000	1.0	0.053	0.0
6000	1.0	0.048	0.0
7000	1.0	0.054	0.0
8000	1.0	0.051	0.0
8584	1.0	0.051	0.0
10000	1.0	0.051	0.0
20000	1.0	0.051	0.0
30000	1.0	0.053	0.0
40000	1.0	0.050	0.0
50000	1.0	0.051	0.0
60000	1.0	0.052	0.0
70000	1.0	0.048	0.0
80000	1.0	0.054	0.5
85840	1.0	0.054	0.5

Fig. 6 illustrates the Kernel Density Estimation (KDE) plot of the weight norm (Wnorm) across training steps. The contours indicate that parameter updates remain tightly concentrated between 0.050 and 0.054, demonstrating the stability of gradient magnitudes under adaptive noise calibration. This stability ensures that optimization does not diverge, even when differential privacy noise is injected.These results confirm that ADP-LoRA not only stabilizes privacy-preserving fine-tuning but also

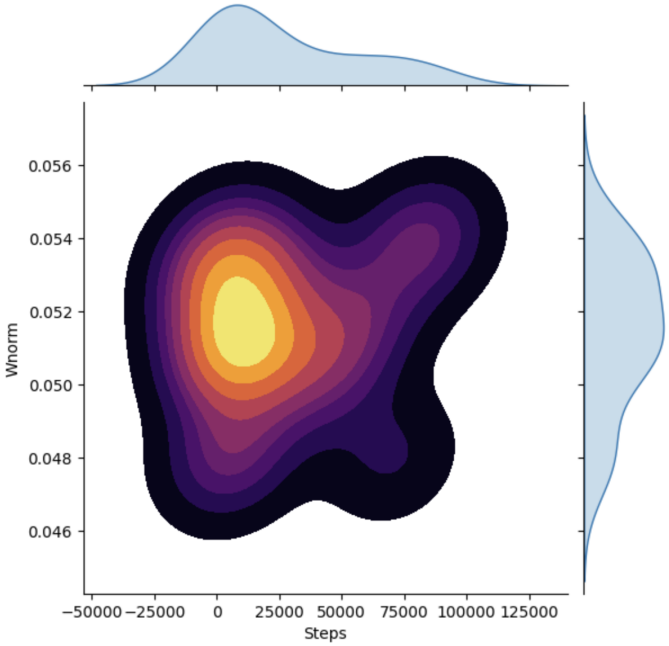

Fig. 6. Kernel Density Estimation of Wnorm vs Steps

delivers significantly higher accuracy (+5–6%) compared to baseline DP-SGD approaches. The adaptive controller, rollback safeguard, and noise-aware pruning collectively enable strong differential privacy guarantees while maintaining near non-private model performance.

V. Conclusion and Future Directions

The experiments on BERT with the SNLI dataset demonstrate that ADP-LoRA achieves 77.9% accuracy, significantly outperforming DP-SGD fine-tuning baselines, which typically converge around 72–73%. This reduction in the privacy–utility gap underscores the effectiveness of combining parameter-efficient fine-tuning with adaptive noise calibration. Stability analyses of gradient norms further confirm that ADP-LoRA maintains bounded updates under repeated noise injection, ensuring consistent optimization.Despite these gains, ADP-LoRA introduces some limitations. The adaptive controller effectively regulates noise but adds computational overhead through periodic privacy audits and discriminator training, and a small gap with non-private models persists despite the 5–6% improvement over DP-SGD. Future work will explore extensions to multi-task and multilingual settings, scaling to billion-parameter foundation models, and integration with federated learning for decentralized training.

References

[1] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, vol. 1, pp. 4171–4186, 2019.

[2] R. Behnia, M. R. Ebrahimi, J. Pacheco, and B. Padmanabhan, "EW-Tune: A framework for privately fine-tuning large language models with differential privacy," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Orlando, FL, USA, 2022, pp. 560–566, doi: 10.1109/ICDMW58026.2022.00078.

[3] M. Tong, X. Li, and Y. Chen, "InferDPT: Privacy-preserving inference for black-box large language models," *IEEE Trans. Dependable Secure Comput.*, early access, 2025, doi: 10.1109/TDSC.2025.3550389.

[4] K. Ahmadi, R. Behnia, R. Ebrahimi, M. Mozaffari Kermani, J. Birrell, J. Pacheco, and A. A. Yavuz, "An interactive framework for implementing privacy-preserving federated learning: Experiments on large language models," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, San Francisco, CA, USA, May 2025, pp. 251–259, doi: 10.1109/SPW67851.2025.00035.

[5] M. Zhu, A. Mao, J. Liu, and Y. Yuan, "DEeR: Deviation eliminating and noise regulating for privacy-preserving federated low-rank adaptation," *IEEE Trans. Med. Imag.*, vol. 44, no. 4, pp. 1783–1795, Apr. 2025, doi: 10.1109/TMI.2024.3518539.

[6] K. Li, Z. Lu, and S. Yu, "PFFLoRA: Personalized Fourier LoRA fine-tuning of federated large language models," in *Proc. Int. Conf. Frontier Technol. Inf. Comput. (ICFTIC)*, Qingdao, China, 2024, pp. 895–899, doi: 10.1109/ICFTIC64248.2024.10912848.

[7] Z. Zhang, P. Liu, J. Xu, and R. Hu, "Fed-HeLLo: Efficient federated foundation model fine-tuning with heterogeneous LoRA allocation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 2025, doi: 10.1109/TNNLS.2025.3580495.

[8] X. Yi, C. Hu, B. Cai, H. Huang, Y. Chen, and K. Wang, "FedALoRA: Adaptive local LoRA aggregation for personalized federated learning in LLM," *IEEE Internet Things J.*, early access, 2025, doi: 10.1109/JIOT.2025.3582427.

[9] E. Trautmann, I. Hales, and M. F. Volk, "Aggregating low rank adapters in federated fine-tuning," *arXiv preprint arXiv:2501.06332*, Jan. 2025.

[10] Z. Gao, Z. Zhang, Y. Guo, and Y. Gong, "Federated adaptive fine-tuning of large language models with heterogeneous quantization and LoRA," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, London, U.K., 2025, pp. 1–10, doi: 10.1109/INFOCOM55648.2025.11044641.

[11] S. Sadeepa, K. Kavinda, E. Hashika, C. Sandeepa, T. Gamage, and M. Liyanage, "DisLLM: Distributed LLMs for privacy assurance in resource-constrained environments," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Taipei, Taiwan, 2024, pp. 1–9, doi: 10.1109/CNS62487.2024.10735498.

[12] C. Li, F. Zhang, Y. Chen, T. Wang, Z. Li, and Q. Yang, "Federated transfer learning for on-device LLMs efficient fine-tuning optimization," *Big Data Mining Anal.*, vol. 8, no. 2, pp. 430–446, Apr. 2025, doi: 10.26599/BDMA.2024.9020068.

[13] S. S. Kolekar and K. Kim, "Resource-efficient federated fine-tuning of LLMs for downstream tasks of speech recognition," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, Honolulu, HI, USA, 2025, pp. 1–5, doi: 10.1109/NOMS57970.2025.11073667.

[14] B. Li and W. Li, "Knowledge distillation-based federated fine-tuning under resource constraints," in *Proc. China Autom. Congr. (CAC)*, Qingdao, China, 2024, pp. 426–430, doi: 10.1109/CAC63892.2024.10865080.

[15] T. Zhang, H. Yu, Z. Yang, Y. Chen, and S. Yu, "LaVFL: Efficient verifiable federated learning for large language models," *IEEE Trans. Dependable Secure Comput.*, early access, 2025, doi: 10.1109/TDSC.2025.3581728.

[16] A. Yousefpour, S. Shilov, and A. Basu, "Opacus: User-friendly differential privacy library in PyTorch," *arXiv preprint arXiv:2109.12298*, 2021.

[17] 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," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

[18] Z. Bu, Y.-X. Wang, S. Zha, and G. Karypis, "Automatic clipping: Differentially private deep learning made easier and stronger," in *Proc. Neural Inf. Process. Syst. (NeurIPS)*, 2023.

[19] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Assoc. Comput. Linguistics, 2015.