PRIVACY-PRESERVING FINE-TUNING OF LARGE LANGUAGE MODELS THROUGH FLATNESS

Tiejin Chen
Arizona State University
tiejin@asu.edu

Longchao Da
Arizona State University
longchao@asu.edu

Huixue Zhou
University of Minnesota
zhou1742@umn.edu

Pingzhi Li
The University of North Carolina at Chapel Hill
pingzhi@cs.unc.edu

Kaixiong Zhou
North Carolina State University
kzhou22@ncsu.edu

Tianlong Chen
University of North Carolina at Chapel Hill
tianlong@cs.unc.edu

Hua Wei
Arizona State University
hua.wei@asu.edu

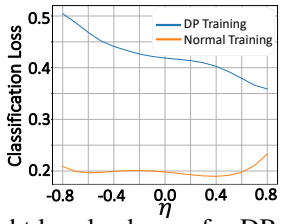
ABSTRACT

The privacy concerns associated with the use of Large Language Models (LLMs) have grown dramatically with the development of pioneer LLMs such as ChatGPT. Differential Privacy (DP) techniques that utilize DP-SGD are explored in existing work to mitigate their privacy risks at the cost of generalization degradation. Our paper reveals that the flatness of DP-SGD trained models' loss landscape plays an essential role in the trade-off between their privacy and generalization. We further propose a holistic framework Privacy-Flat to enforce appropriate weight flatness, which substantially improves model generalization with competitive privacy preservation. It innovates from three coarse-to-grained levels: Perturbation-aware min-max optimization within a layer, flatness-guided sparse prefix-tuning across layers, and weight knowledge distillation between DP & non-DP weights copies. We empirically demonstrate that our framework Privacy-Flat outperforms vanilla DP training baseline while preserving strong privacy by the evaluation of membership inference attacks. Comprehensive experiments of both black-box and white-box scenarios are conducted to demonstrate the effectiveness of our proposal in enhancing generalization.

1 INTRODUCTION

Large Language Models (LLMs) such as GPT-4 (OpenAI, 2023) and Llama 2 (Touvron et al., 2023) have become popular in various real-world applications, including story generation (Zhou et al., 2023; Yang et al., 2022a), AI agents (Mialon et al., 2023; Da et al., 2023b), chatbots (Luo et al., 2022) and sim-to-real learning (Da et al., 2023a). Despite their widespread use, these models raise significant privacy concerns. Previous studies have shown that LLMs can memorize and potentially leak sensitive information from their training data (Carlini et al., 2021; Mireshghallah et al., 2022), which often includes personal details like emails (Huang et al., 2022), phone numbers and addresses (Carlini et al., 2021). There are also LLMs trained especially for clinical and medical usage with highly sensitive data (Yang et al., 2022b). The leakage of such information from LLMs may cause privacy issues.

Differential Privacy (DP) has emerged as a key method for protecting data privacy in LLMs, yet sacrificing the generalization ability. Specifically, techniques such as Differentially Private Stochastic Gradient Descent (DP-SGD) (Abadi et al., 2016) have been employed to improve the trade-off between privacy and performance. However, there remains a noticeable performance gap between DP-trained models and standard models in both full fine-tuning and parameter-efficient training settings (Li et al., 2021; Du et al., 2023). Moreover, most current works focus on improving privacy for white-box LLMs, which have limited applicability to closed-source LLMs in real-world scenarios.



	MIA Accuracy (↓)	Classification Accuracy (↑)
DP Training	**50.46**	91.28
Normal Training	62.11	**94.15**

Figure 1: Left: Weight loss landscape for DP-trained LLMs and normal (non-private) training on SST-2. The DP-trained model has a sharper loss landscape. Right: The privacy-performance trade-off for DP-trained LLMs: Compared with normal trained models, the DP-trained model has lower privacy risks (better privacy) under Membership Inference Attack (MIA), while it shows lower classification accuracy (worse performance).

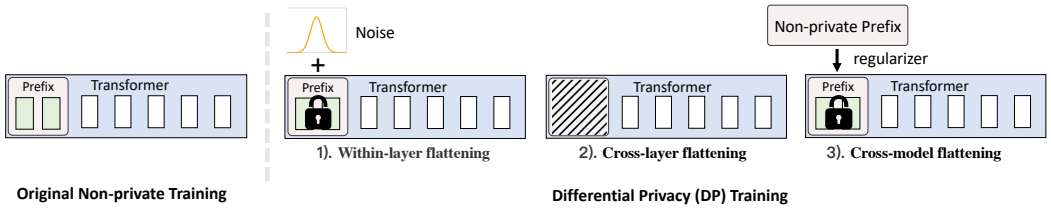


Figure 2: Our methods improve the flatness of the weight loss landscape from three aspects: (1) Within-layer flattening, where a perturbation-aware min-max optimization is utilized to encourage the loss flatness within the weight space of each LLM layer. (2) Cross-layer flattening, where a sparse prefix-tuning algorithm guides layer selection with a flatness-ware indicator. (3) Cross-model flattening, where non-private prefixes are used to guide DP-SGD training through weight knowledge distillation regularization.

Therefore, there is an urgent call for pioneering efforts to design effective algorithms in black-box privacy-preserving optimization.

To understand this performance gap, we examine the loss landscape of DP-trained models compared to the ones from non-private training. As shown in Figure 1, it illustrates the analysis with the following formula:

$$f(\eta) = \mathcal{L}(\mathcal{D} \mid \mathbf{w} + \eta \cdot \mathbf{d}),$$

where $\mathcal{D}$ and $\mathbf{w}$ represent the dataset and model weights, respectively, and $\mathbf{d}$ is a random noise sampled from a standard Gaussian distribution and η is the magnitude. It reveals that DP-trained models tend to have a sharper (*i.e.*, less flatness) loss landscape with respect to model weights. Then, a natural question comes:

Q: Does the Loss Flatness Affect the Privacy and Performance Trade-off in DP-trained LLMs?

If so, could we take one step further — improving performance with competitive privacy by appropriately enhancing the loss landscape's flatness? We present a holistic framework, consisting of three novel strategies to promote weight-level flatness from three coarse-to-grained perspectives:

▷ *Within-layer flattening*. We introduce a perturbation-aware min-max optimization to encourage the loss landscape flatness within the weight space of each LLM layer.

▷ *Cross-layer flattening*. We propose a sparse prefix-tuning algorithm to facilitate the landscape flatness across LLM layers (Li & Liang, 2021), where a flatness-ware indicator will guide the sparse layer selection.

▷ *Cross-model flattening*. We design a novel approach using non-private prefixes to guide DP-SGD training through knowledge distillation regularization with non-private weights, aiming to improve the flatness in the whole weight space of LLMs.

Our main contributions can be summarized as follows:

• We conduct pioneering efforts to investigate the critical role of weight flatness in DP-trained LLMs. We show that appropriately enforced weight flatness improves the performance of LLMs with DP-SGD.

• We propose a holistic framework named **Privacy-Flat** to promote weight flatness in three coarse-to-grained levels, including perturbation-ware mix-max optimization on weights within a layer, flatness-guided sparse prefix-tuning on weights across layers, and weight knowledge distillation between Privacy-Flat & non-private weight copies. Our experimental results show that under DP-SGD, our framework can have good privacy empirically.

• We make pioneering efforts to propose effective privacy-preserving algorithms for closed-source large language models with tailored black-box optimization.

• Comprehensive experiments in both black-box and white-box settings are conducted to show that our proposed methods can bridge the notorious gap between non-private LLMs and LLMs with good privacy. For example, on the text classification dataset QNLI, Privacy-Flat even outperforms non-private full fine-tuning.

2 METHODS

In this paper, we mainly focus on the DP-SGD (Abadi et al., 2016) and its variants for providing privacy even without a strict DP guarantee. ϵ and δ are the privacy budgets for DP-SGD where small values of ϵ and δ indicate strong privacy protection. DP-SGD algorithm could be realized via three interleaved steps: clipping per-sample gradient, sampling a random noise $z \sim N(0, \sigma^2 I)$, and adding z to the accumulated clipped gradient. The variance parameter σ^2 is determined by several factors including total training steps, ϵ, and δ.

2.1 ENHANCING FLATNESS IN WHITE-BOX SETTING

To mitigate the negative impact of private training, we propose a flatness-aware framework, termed as Privacy-Flat, to enhance the accuracy-privacy trade-off. Specifically, considering a multi-layer white-box model, we smooth the sharp local minima of LLMs comprehensively from three perspectives, including within-layer, cross-layer, and cross-model weight flattening.

Within-layer Weight Flattening. We mainly adjust adversarial weight perturbation (AWP) (Wu et al., 2020) to flatten the weight loss landscape. In detail, Let $\mathbf{w}$ represent the trainable parameters in LLMs, and let $\mathcal{D}$ represent the training dataset. Typically in prefix tuning of LLMs, $\mathbf{w}$ is given by the appending learnable tokens at each layer (Li & Liang, 2021). AWP updates the model weights with two gradient backpropagation steps: First, $\mathbf{v} = arg\max_{\mathbf{v}} \mathcal{L}(\mathcal{D}; \mathbf{w} + \mathbf{v})$; Second, $\mathbf{w} \leftarrow (\mathbf{w} + \mathbf{v}) - \eta \nabla_{\mathbf{w}+\mathbf{v}} \mathcal{L}(\mathcal{D}; \mathbf{w} + \mathbf{v}) - \mathbf{v}$. We tailor AWP to DP-SGD with two critical changes. First, we only consider applying the adversarial perturbation gradients in the first T rounds of training, following which the normal model updating is turned on. With this procedure, we can save the external time cost of adversarial computation while guiding the model towards a smooth loss region. Second, during the initial T rounds, the required noises in DP-SGD are only added to the final gradient $\nabla_{\mathbf{w}+\mathbf{v}} \mathcal{L}(\mathcal{D}; \mathbf{w} + \mathbf{v})$, instead of the process of computing $\mathbf{v}$.

Cross-layers Weight Flattening. Given a n-layer LLMs, prefix weights $\mathbf{w}_i$ are appended at the i-th layer and we have $\mathbf{w} = [\mathbf{w}_1, ..., \mathbf{w}_n]$. However, as the prefix added to a layer influences its following output, the flatness of the weight loss landscape is determined by where the prefix modules are added. Thus we explore how to quickly quantify the model sharpness and how to adopt it for controlling the positions of prefix layers.

Definition 1 (Prefix Sharpness)**.** *Given prefix parameters $\mathbf{w}'$ within a box in parameter space $\mathcal{C}_\eta$ with sides of length $\eta > 0$, centered around a minima of interest at parameters $\mathbf{w}$, the sharpness of loss $\nabla\mathcal{L}(\mathbf{w})$ at $\mathbf{w}$ is defined as:* $\text{Sharpness} := \frac{\max_{\mathbf{w}' \in \mathcal{C}_\eta} \big(\mathcal{L}(\mathbf{w}') - \mathcal{L}(\mathbf{w})\big)}{1 + \mathcal{L}(\mathbf{w})}$.

In practice, we approximate the above prefix sharpness by sampling prefix weights $\mathbf{w}'$: $\mathbf{w}' \in \{\mathbf{w} - \eta \nabla \mathcal{L}(\mathbf{w}|\mathcal{D}) | \eta \in [0, 1]\}$. Based on the sharpness definition, we design a greedy solution to gradually eliminate the prefix layers and keep those resulting in lowest sharpness. First, with the prefix initialization at all the layers of LLMs, we can compute the its sharpness value. Next we remove one prefix layer each time and calculate the corresponding sharpness of remaining model parameters. The prefix layer where its removing is associated with the lowest sharpness will be permanently deleted. We will continue this loop until the remaining prefixes meet our sparse requirement or the sharpness metric does not decrease.

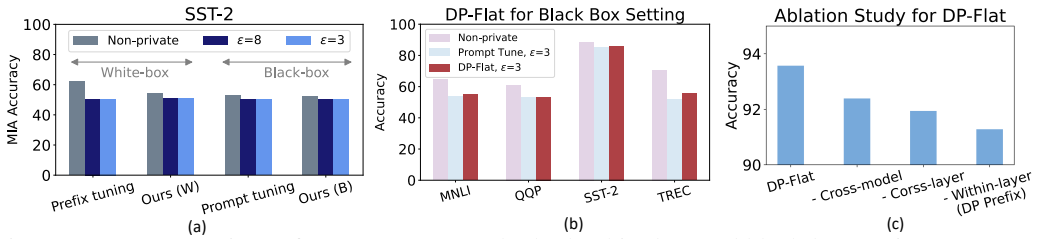


Figure 3: (a): Comparison of MIA accuracy under both white-box and black-box settings on SST-2; (b): Performance of Privacy-Flat compared with baseline methods in black-box settings; (c): Ablation study on SST-2 where we gradually remove our methods.

Cross-models Weight Flattening. Recall that DP-SGD inevitably results in a sharper loss landscape than that of normal training. One of the intuitive ways to generalize private model is to regularize it with the normal counterpart via knowledge distillation (Gou et al., 2021). For this purpose, given parameters $\mathbf{w}$ fine-tuned with DP framework, we create their duplicates $\mathbf{w}_{\text{nor}}$ using the same network architecture and initialization but fine-tuning them normally. We then define a new term of loss function to force the weight closeness between $\mathbf{w}$ and $\mathbf{w}_{\text{nor}}$: $\mathcal{L}_g = \|\mathbf{w} - \mathbf{w}_{\text{nor}}\|_2$ Therefore, the final loss function will be: $\mathcal{L}_f = \mathcal{L}(\mathcal{D}|\mathbf{w}) + \lambda\mathcal{L}_g$, where $\mathcal{L}$ can be any loss function in general, such as cross-entropy loss for sentence classification tasks, and λ is the balancing factor for regularization. It is minimized using the DP-SGD framework to achieve both data privacy protection and the desired accuracy. Finally, we summarize our training pipeline for the white-box setting in the Appendix Appendix A.3. and show how each part of Privacy-Flat helps to reduce the sharpness in Appendix A.4.

2.2 Enhancing Flatness in Black-box Setting

While LLMs of interest are oftentimes black boxes, i.e., their weights are not accessible for training. in this section, we extend our Privacy-Flat framework to the black-box settings. And we will utilize zeroth-order (ZO) optimizer (Malladi et al., 2023) to estimate the gradient and DPZero (Zhang et al., 2023) to provide the DP guarantee. Implementing zeroth-order optimization is not enough for considering application in the real-world scenario since the weights of black-box models are not accessible. In this case, only manipulating input embedding is more practical. Therefore, in the black-box setting, we adopt Prompt Tuning (Lester et al., 2021), which only adds learnable tokens before input embedding.

In the black-box setting, we only consider improving through non-private duplication. Compared with the white-box setting, $\mathbf{w}_{nor}$ is also trained with the black-box setting. We don't consider enhancing the within-layer weight flatness since the min-max training framework with zeroth order optimization suffers from the high variance of an additional gradient estimation to compute $\mathbf{v}$ (Zhang et al., 2022).

2.3 Discussion

Since Privacy-Flat does not consider the DP framework every time like generating model perturbation gradient $\mathbf{v}$, Privacy-Flat cannot provide a strict DP guarantee. Though our method cannot provide a strict DP guarantee, we prove that under the framework of DP-SGD, our method can still have good privacy in the experimental parts and thus improve the trade-off between accuracy and privacy. We leave the theoretical proof of why Privacy-Flat can still maintain good privacy in future work.

3 Experiments

Experimental Settings In this paper, we consider use text classification datasets from GLUE (Wang et al., 2018): SST-2 (Socher et al., 2013), MNLI (Williams et al., 2017), QNLI (Wang et al., 2018), QQP and TREC (Voorhees et al., 1999). For text generation tasks, we mainly consider table2text generation with E2E (Novikova et al., 2017) and DART dataset (Nan et al., 2020). And we mainly consider Roberta-base (Liu et al., 2019) and consider the privacy budget $\epsilon = [3, 8]$ and $\delta = \frac{1}{2|\mathcal{D}|}$.

3.1 Empirical Evaluation of Privacy Risks

In this section, we conduct experiments to show that Privacy-Flat shows a similar capability in privacy-preserving as vanilla DP training. To measure the privacy-preserving ability, we apply a loss-based membership inference attack to different models with SST-2 and Roberta-base. More detailed

descriptions of the MIA setting can be found in the Appendix A.2. From the results in Figure 3 (a), we can see that Privacy-Flat show similar MIA accuracies with DP-trained prefixes and lower the privacy risks a lot compared with non-private training.

Method	Roberta-base					BERT				
	MNLI	QNLI	SST-2	QQP	TREC	MNLI	QNLI	SST-2	QQP	TREC
Non-private ($\epsilon = \infty$)										
Full Fine-tuning	85.95	91.06	**94.68**	**88.05**	93.00	**83.09**	**88.94**	**91.85**	**90.17**	92.60
Prefix Tuning	**86.12**	**91.59**	94.15	87.79	91.40	79.95	86.34	91.62	89.25	**96.00**
$\epsilon = 3$										
Full Fine-tuning	80.95	86.03	92.08	83.61	79.00	**72.57**	**81.70**	87.50	**81.46**	**73.60**
Prefix Tuning	79.03	83.70	91.28	80.13	78.40	60.07	65.15	81.19	71.99	48.40
Privacy-Flat	**84.12**	**90.72**	**93.57**	**86.05**	**82.20**	65.32	71.02	**88.53**	74.68	47.80
$\epsilon = 8$										
Full Fine-tuning	81.42	86.03	92.18	83.61	85.40	**73.64**	**82.37**	88.30	**81.92**	**80.60**
Prefix Tuning	79.56	84.64	91.51	81.02	86.80	62.72	67.62	82.34	72.46	61.80
Privacy-Flat	**85.30**	**91.29**	**94.03**	**87.13**	**90.60**	67.42	72.08	**89.56**	74.29	70.20

Table 1: Performance of our weight flattening methods with baselines for the sentence classification task w.r.t accuracy on white-box settings across different language models. The higher, the better. The **best** performance under the same DP training is highlighted. The results show that Privacy-Flat can increase the performance of DP-trained LLMs for various text classification tasks.

3.2 Evaluation in Results

We conduct experiments under both black-box and white-box settings to show the performance of Privacy-Flat in classification and generation tasks.

White-box Setting We first explore whether Privacy-Flat can bridge the gap between private models and non-private models ($\epsilon = \infty$) in a white-box setting. In Table 1, we provide the results. Privacy-Flat can increase the performance of models trained with private prefix tuning a lot across different datasets and model architectures and beat all DP full-tuning models for Roberta-base. Privacy-Flat can even beat the non-private setting for QNLI when $\epsilon = 8$. Privacy-Flat also shows an improvement for table2text generation and the detailed results are shown in Appendix A.5.

Black-box Setting In this section, we test Privacy-Flat in the black-box setting where we can only manipulate input embedding. Therefore, instead of prefix tuning, only prompt tuning could be implemented. The results are shown in Figure 3 (b) across different datasets with Roberta-base, which shows that compared with white-box setting, black-box setting is much harder than white-box setting. However, Privacy-Flat can still improve the performances of private models due to the flat loss landscape in most tasks.

Ablation Study on Different Flatness Aspects We conduct ablation studies to show the performance while gradually removing our methods on SST-2 as shown in Figure 3 (c). Each component will enhance the performance while maintaining the privacy guarantee, indicating the effectiveness of the proposed flattening methods.

4 Conclusion

In this paper, we address the challenge of balancing privacy with performance in Large Language Models. We introduce a novel framework aimed at enhancing the flatness of the loss landscape in DP-SGD-trained models, proposing strategies at three levels: within-layer flattening, cross-layer flattening, and cross-model flattening. Our approach provides a better balance between privacy and performance, as well as offering pioneering solutions for privacy-preserving algorithms in closed-source settings. Our comprehensive experiments demonstrate significant performance improvements across different tasks in both black-box and white-box settings while maintaining good privacy.

Acknowledgement The work was partially supported by NSF award #2153311. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Longchao Da, Minchiuan Gao, Hao Mei, and Hua Wei. Llm powered sim-to-real transfer for traffic signal control. *arXiv preprint arXiv:2308.14284*, 2023a.

Longchao Da, Kuanru Liou, Tiejin Chen, Xuesong Zhou, Xiangyong Luo, Yezhou Yang, and Hua Wei. Open-ti: Open traffic intelligence with augmented language model. *arXiv preprint arXiv:2401.00211*, 2023b.

Minxin Du, Xiang Yue, Sherman SM Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2665–2679, 2023.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. Are large pre-trained language models leaking your personal information? *arXiv preprint arXiv:2205.12628*, 2022.

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

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

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Bei Luo, Raymond YK Lau, Chunping Li, and Yain-Whar Si. A critical review of state-of-the-art chatbot designs and applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(1):e1434, 2022.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *arXiv preprint arXiv:2305.17333*, 2023.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*, 2023.

Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David Evans, and Taylor Berg-Kirkpatrick. Memorization in nlp fine-tuning methods. *arXiv preprint arXiv:2205.12506*, 2022.

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

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

OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. URL https://arxiv.org/abs/2303.08774.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Ellen M Voorhees et al. The trec-8 question answering track report. In *Trec*, volume 99, pp. 77–82, 1999.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.

Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. Re3: Generating longer stories with recursive reprompting and revision. *arXiv preprint arXiv:2210.06774*, 2022a.

Xi Yang, Aokun Chen, Nima PourNejatian, Hoo Chang Shin, Kaleb E Smith, Christopher Parisien, Colin Compas, Cheryl Martin, Mona G Flores, Ying Zhang, et al. Gatortron: A large clinical language model to unlock patient information from unstructured electronic health records. *arXiv preprint arXiv:2203.03540*, 2022b.

Liang Zhang, Kiran Koshy Thekumparampil, Sewoong Oh, and Niao He. Dpzero: Dimension-independent and differentially private zeroth-order optimization. *arXiv preprint arXiv:2310.09639*, 2023.

Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jinfeng Yi, Mingyi Hong, Shiyu Chang, and Sijia Liu. How to robustify black-box ml models? a zeroth-order optimization perspective. *arXiv preprint arXiv:2203.14195*, 2022.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Peng Cui, Tiannan Wang, Zhenxin Xiao, Yifan Hou, Ryan Cotterell, and Mrinmaya Sachan. Recurrentgpt: Interactive generation of (arbitrarily) long text. *arXiv preprint arXiv:2305.13304*, 2023.

A EXPERIMENT SETUPS

A.1 HYPERPARAMETERS

Different tasks and methods require different parameters. For example, full fine-tuning requires a much smaller learning rate while prefix tuning needs a much larger learning rate. Besides, tasks like table-to-text generation require a small learning with a large training epoch. The only fixed hyperparameter is the batch size. We set the batch size to 1024 for all settings with gradient accumulation. Detailed hyperparameters for MNLI and E2E can be found in Table 2. For Privacy-Flat, we set the regularization weight λ to 0.01 for all experiments. For the DP-SGD, we follow the common practice to set the privacy budget as $\epsilon = [3, 8]$ and $\delta = 1e - 5$ for all settings,

Methods	Learning Rate	Training Epoch
Non private-MNLI		
Full Fine-tuning	5e-5	5
Prefix Tuning	0.01	20
Privacy-Flat	0.01	20
DP setting-MNLI		
Full Fine-tuning	5e-4	5
Prefix Tuning	0.01	20
Privacy-Flat	0.01	20
Non private-E2E		
Full Fine-tuning	2e-3	15
Prefix Tuning	5e-4	30
Privacy-Flat	5e-4	30
DP setting-E2E		
Full Fine-tuning	2e-3	15
Prefix Tuning	5e-4	100
Privacy-Flat	5e-4	100

Table 2: Detailed hyperparameters for DP training and normal training on MNLI and E2E.

A.2 SETTINGS FOR MEMBERSHIP INFERENCE ATTACK

We evaluate the privacy risks empirically by membership inference attack (MIA) using Likelihood Ratio test (LiRA) (Mireshghallah et al., 2022). For SST-2, because of the distributional bias between the training and test sets, we filter the training set to include samples with more than 20 tokens, in which case only 15 test samples are eliminated. The data filtering can avoid undesired high MIA accuracy due to the lack of short samples in test sets. Then we compute the loss for all samples in $\hat{\mathcal{D}}$ and rank every sample by its loss. We label all the samples with 1% lowest loss as training data and compute the success rate of MIA only on samples with 1% lowest loss. Note that a model that preserves more privacy indicates that the success rate of MIA is closer to 50% because if attackers get an MIA success rate below 50%, they could use reverse results to implement attacks. The results are reported in Figure 3 under text classification datasets with both white-box and black-box settings.

A.3 ALGORITHM

We provide the detailed algorithm in Algorithm 1.

A.4 HOW EACH PART OF PRIVACY-FLAT INFLUENCE THE FLATNESS

To test whether our proposed methods can help to flatten the loss landscape, we compute the sharpness for DP prefix tuning and DP prefix tuning with the proposed methods. The results shown in Figure 5 shown that even with only one (and each one) method added to the DP prefix tuning, it can help to reduce the sharpness of weight loss landscape.

Algorithm 1 Privacy-Flat White-box DP training pipeline

Input: λ, η,warm-up epochs E, DP training total epochs T_{dp}, normal training epochs T_{nor}, elimination rounds R, random initialization prefix $\mathbf{w} = [\mathbf{w}_1, ..., \mathbf{w}_n]$.
if *Cross-layers Weight Flattening* **then**
  for $r = 1$ **to** R **do**
    $S_{\min} = \infty$
    $P = 0$
    for $i = 1$ **to** n **do**
      Get $\mathbf{w}_{-i}$
      Compute sharpness S for $\mathbf{w}_{-i}$
      if $S < S_{\min}$ **then**
        $S_{\min} = S$, $P = i$
      end if
    end for
    $\mathbf{w} \leftarrow \mathbf{w}_{-P}$
  end for
end if
$\mathbf{w}_{nor} = \mathbf{w}$
for $t = 1$ **to** T_{nor} **do**
  $\mathbf{w}_{\text{nor}} \leftarrow \mathbf{w}_{\text{nor}} - \eta \nabla_{\mathbf{w}_{\text{nor}}} \mathcal{L}(\mathcal{D}|\mathbf{w}_{\text{nor}})$
end for
for $t = 1$ **to** T **do**
  if $t <= E$ **and** *Within-layer Weight Flattening* **then**
    Compute $\mathbf{v}$
    $\mathcal{L}_f = \mathcal{L}(\mathcal{D}|\mathbf{w} + \mathbf{v})$
  else
    $\mathcal{L}_f = \mathcal{L}(\mathcal{D}|\mathbf{w})$
  end if
  if *Cross-model Weight Flattening* **then**
    $\mathcal{L}_f = \mathcal{L}_f + \lambda \|\mathbf{w} - \mathbf{w}_{\text{nor}}\|_2$
  else
    $\mathcal{L}_f = \mathcal{L}_f$
  end if
  Update $\mathbf{w}$ with $\mathcal{L}_f$ and DP-Adam
end for

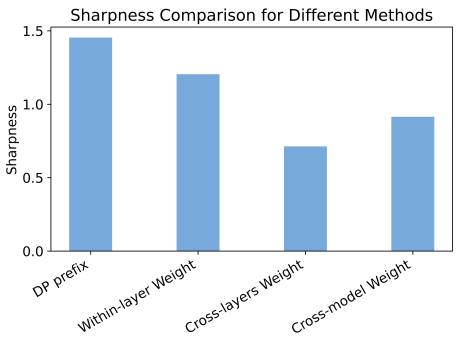


Figure 4: Sharpness for DP trained prefix tuning plus our proposed three weight flattening methods on SST-2. Our proposed model has a flatter loss landscape.

A.5 Detailed Experimental Results

We provide the detailed results for generation task in Table 3. we get the following observations:

(1) Privacy-Flat outperforms DP-trained models across all datasets in private training settings. With the same privacy budget ϵ, Privacy-Flat consistently performs the best.

(2) The performance of DP training models increases higher privacy budget ϵ, while Privacy-Flat achieve competitive performance with DP prefix tuning methods with higher ϵ. This indicates that Privacy-Flat can provide a strong utility for conservative privacy budgets.

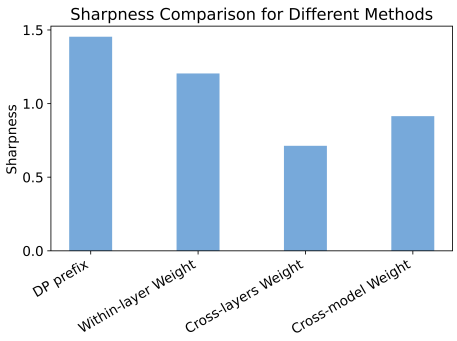


Figure 5: Sharpness for DP trained prefix tuning plus our proposed three weight flattening methods on SST-2. Our proposed model has a flatter loss landscape.

(3) For tasks with different difficulties, Privacy-Flat shows competitive or better performances. In simple tasks (E2E dataset), the gap between DP-trained models and non-private models ($\epsilon = \infty$) is small. When $\epsilon = 8$, Privacy-Flat can even compete with prefix tuning with non-private training. For difficult tasks (DART dataset), the performance gap between the non-private model and the DP-trained model becomes much larger. The performance of DP prefix tuning can compete or become even better than DP full fine-tuning, indicating the advantages of full fine-tuning rely on the easy dataset.

We also show our detailed results for black-box setting across different datasets in Table 4 to show that Privacy-Flat has a great performance.

Method	E2E		DART	
	BLEU	ROUGE-L	BLEU	ROUGE-L
Non-private ($\epsilon = \infty$)				
Full Fine-tuning	**66.59**	**69.54**	**43.16**	**57.85**
Prefix Tuning	64.79	68.24	37.08	53.35
$\epsilon = 3$				
Full Fine-tuning	60.3	65.31	30.75	51.69
Prefix Tuning	58.2	64.51	30.26	51.43
Privacy-Flat	**62.13**	**65.84**	**33.14**	**52.40**
$\epsilon = 8$				
Full Fine-tuning	62.9	66.69	32.92	53.43
Prefix Tuning	62.7	67.19	33.45	53.45
Privacy-Flat	**64.30**	**67.22**	**37.06**	**53.49**

Table 3: Comparison of our weight smooth methods with baselines for the table-to-text task on GPT2 and white-box settings. The higher, the better. The **best** performance under the same DP training is highlighted. Privacy-Flat performs consistently better than other DP-trained methods on various text generation tasks.

A.6 Sensitivity on Different λ

The regularization factor λ balances the flattening with knowledge distillation and DP training. As is shown in Figure 6, when we use knowledge distillation, Privacy-Flat performs better Privacy-Flat without knowledge distillation. Note that when $\lambda = 0$, our method will not consider cross-model flattening. In this paper, we set λ as $1e^{-2}$ as it performs the best empirically.

Method	Roberta-base			
	MNLI	QQP	SST-2	TREC
	Non-private ($\epsilon = \infty$)			
Prompt Tuning with MEZO	64.51	60.93	88.46	70.61
	$\epsilon = 3$			
Prompt Tuning with DPZero	53.99	**53.41**	85.2	52.14
Privacy-Flat	**55.07**	53.22	**86.12**	**55.46**
	$\epsilon = 8$			
Prompt Tuning with DPZero	55.41	**53.51**	86.35	53.02
Privacy-Flat	**57.13**	53.42	**87.38**	**56.44**

Table 4: Comparison of our flattening methods with baselines for the sentence classification task on black-box setting. The higher, the better. The **best** performance under the same DP training is highlighted. Under the black-box setting, only prompt tuning could be implemented. Privacy-Flat achieves competitive performance under different text classification tasks.

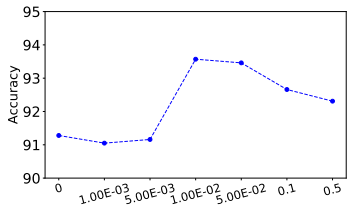


Figure 6: Influences of different values of factor λ on the classification performance w.r.t. accuracy under SST-2 dataset on Roberta-base. The higher, the better.