Protecting Privacy against Membership Inference Attack with LLM Fine-tuning through Flatness

Tiejin Chen[1], Longchao Da[1], Huixue Zhou[2], Pingzhi Li[3],

Kaixiong Zhou[4], Tianlong Chen[3], Hua Wei[1]

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 promising privacy protection. 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 private & non-private weights copies. We empirically demonstrate that our framework Privacy-Flat outperforms vanilla private training baseline while protecting privacy from membership inference attacks (MIA). Comprehensive experiments of both black-box and white-box scenarios are conducted to demonstrate the effectiveness of our proposal in enhancing generalization. The code link is provided at `https://github.com/tiejin98/Privacy_Flatness`.

1 Introduction

Large Language Models (LLMs) such as GPT-4 [37] and Llama 2 [47] have become integral in various real-world applications, including story generation [61, 53], AI agents [32, 10], chatbots [29] and sim-to-real learning [9]. 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 [4, 33], which often includes personal details like emails [19], phone numbers and addresses [4]. There are also LLMs trained especially for clinical and medical usage with highly sensitive data [54]. The leakage of such information from LLMs may cause a severe privacy issue. The leakage of such information from LLMs may cause a severe privacy issue.

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) [1] 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 [24, 14]. Moreover, most current works focus on improving privacy for white-box LLMs, which have limited applicability to closed-source LLMs in real-world scenarios. Therefore, there is an urgent call for pioneering efforts to design effective algorithms in black-box privacy-protection optimization.

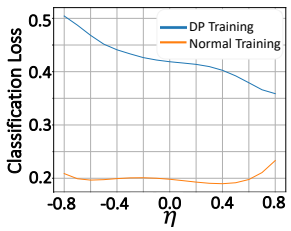
To understand this performance gap, we examine the loss landscape of DP-trained models compared to the ones from non-private training. As shown in Figure 1, it illustrates the analysis with the following formula:

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

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 LLMs with good privacy?

[1]Arizona State University.
Email: {tchen169, longchao, hua.wei}@asu.edu.
[2]University of Minnesota Twin Cities.
Email: zhou1742@umn.edu.
[3]University of North Carolina at Chapel Hill.
Email: {pingzhi, tianlong}@cs.unc.edu.
[4]North Carolina State University.
Email: kzhou22@ncsu.edu.



	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).

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 [23], 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 that protect private information.

• 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.

• 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 Related Work

Learnable Prompts for LLMs: Prompt-based learning has gained traction, initially focusing on discrete, task-specific prompts [41]. The shift to continuous, learnable prompts (soft prompts) has led to improved performance [21, 27]. Unlike traditional prompt tuning, prefix-tuning [23] and P-tuning V2 [26] incorporate prompts at each transformer layer. For prompt tuning or prefix-tuning, zeroth-order optimization (ZO) methods like ZO-SGD [45] are employed for black-box settings without requiring knowing the parameters of the original model. MEZO, introduced by Malladi et al. [31], optimizes ZO-SGD for LLM fine-tuning with lower memory needs. While other works also explored black-box optimization methods for both discrete [7] and soft prompts [46], they do not investigate the issue of privacy leakage in the model training.

Privacy Leakage in LLMs: The potential of Large Language Models (LLMs) to memorize training data poses privacy risks [33, 6, 20]. Such memorization enables the extraction of private information or even direct reconstruction of training data [39, 19, 4, 59, 17]. Studies have demonstrated the feasibility of recovering keywords or predicting training words from sentence embeddings using auxiliary datasets [38, 44]. A notable advancement was made by Li et al. [22], who introduced an attack model to enhance the efficacy of attacks on sentence embeddings. Recent comprehensive analyses, including those on GPT-4, further underline the seriousness of this issue [50, 37]. In this paper, we employ the Membership Inference Attack (MIA) [5, 55, 42] to evaluate LLMs' vulnerability to privacy leakage issues. To mitigate the privacy leakage in LLMs, DP-SGD [1] and its variants [34, 12] have been applied to fine-tuning of LLMs [56, 3, 16, 24, 14, 30, 15]. Compared with previous methods, Privacy-Flat cannot provide strict DP guarantee while Privacy-Flat can still have good privacy.

3 Methods

In this paper, we mainly focus on the DP-SGD [1] 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

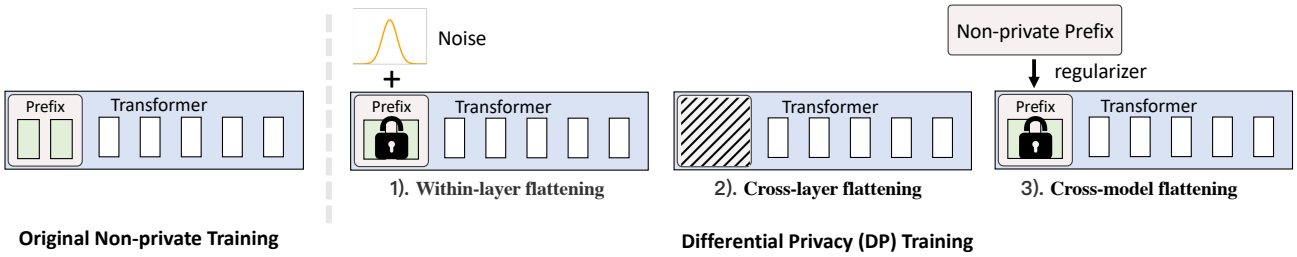


Figure 2: Our methods improve the flatness of the weight loss landscape from three aspects: (1) Within-layer flattening, where a perturbation-aware min-max optimization is utilized to encourage the loss flatness within the weight space of each LLM layer. (2) Cross-layer flattening, where a sparse prefix-tuning 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.

total training steps, ϵ, and δ.

3.1 Enhancing Flatness in White-box Setting

It is notorious that DP-SGD often sacrifices a larger degree of model accuracy to gain the required data privacy. In this work, we propose to investigate this trade-off from a novel perspective, i.e., comparing the metric of model flatness before and after DP training. As shown in Figure 1, LLMs under DP training are prone to converge to sharp local minima, where the loss value increases quickly in the neighborhood around model weights. In other words, a slight perturbation in the model weights will lead to poor generalization in unseen data. Many previous work has revealed the strong correlation between sharp local minima and unacceptable accuracy in vision and natural language processing [8, 13, 2].

To balance between privacy and accuracy, we propose a flatness-aware framework, termed as Privacy-Flat. 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. Many pioneering works have been explored to regularize the layer-wise independent weights, among which adversarial weight perturbation (AWP) shows superior results [52]. AWP flattens the weight loss landscape and aims to improve adversarial robustness, whereas we adopt it with the intuition that the negative impact of DP-SGD noise for model accuracy could be lowered.

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 [23]. AWP updates the model weights with two gradient backpropagation steps:

$$\begin{aligned} \mathbf{v} &= \arg\max_{\mathbf{v}} \mathcal{L}(\mathcal{D}; \mathbf{w}+\mathbf{v}); \\ \mathbf{w} &\leftarrow (\mathbf{w}+\mathbf{v}) - \eta \nabla_{\mathbf{w}+\mathbf{v}} \mathcal{L}(\mathcal{D}; \mathbf{w}+\mathbf{v}) - \mathbf{v}. \end{aligned} \tag{3.1}$$

The first step seeks perturbation gradient $\mathbf{v}$ via gradient ascent, which represents the case of worst loss centered around the current weights $\mathbf{w}$. After adversarially applying the perturbation gradient on the model (i.e., $\mathbf{w}+\mathbf{v}$), the second step updates the model weights with another complete forward and backward pass. In this way, the weight loss landscape has a smaller curvature at the final learned weights, which in turn shrinks the accuracy loss.

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 $\mathbf{v}$. This ensures the correct location of the adversarial gradient.

Cross-layers Weight Flattening. Beyond the regular weight regularization, we manipulate prefix weights in LLMs to further improve flatness via considering their cross-layer dependencies. In particular, the prefix tuning adds the differential parameters in every layer of LLMs: 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:*

$$\text{Sharpness} := \frac{\max_{\mathbf{w}' \in \mathcal{C}_\eta} (\mathcal{L}(\mathbf{w}') - \mathcal{L}(\mathbf{w}))}{(1 + \mathcal{L}(\mathbf{w}))^2}.$$

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 the lowest sharpness. First, with the prefix initialization at all the layers of LLMs, we can compute its sharpness value. Next, we remove one prefix layer each time and obtain:

$$\mathbf{w}_{-i} = [\mathbf{w}_1, .., \mathbf{w}_{i-1}, \mathbf{w}_{i+1}, ..], i = 1 \cdots, n. \tag{3.2}$$

For each prefix detaching, we calculate the corresponding sharpness of the remaining model parameters. The prefix layer where its removal 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. We get all the sharpness results right after the same random initialization and do not require fine-tuning. After this greedy procedure, LLMs are appended with the sparse prefixes only at the chosen layers and used for DP-SGD.

Cross-models Weight Flattening. Recall that private training inevitably results in a sharper loss landscape than that of normal training. One of the intuitive ways to generalize the private model is to regularize it with the normal counterpart via knowledge distillation [18]. 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 the loss function to force the weight closeness between $\mathbf{w}$ and $\mathbf{w}_{\text{nor}}$:

$$\mathcal{L}_g = \|\mathbf{w} - \mathbf{w}_{\text{nor}}\|_2. \tag{3.3}$$

Therefore, the final loss function will be:

$$\mathcal{L}_f = \mathcal{L}(\mathcal{D}|\mathbf{w}) + \lambda\mathcal{L}_g, \tag{3.4}$$

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. Then the final loss is trained with DP-SGD or its variants. Finally, we summarize our training pipeline for white-box setting in Algorithm 1.

Algorithm 1 Privacy-Flat on White-box 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 $\mathbf{w}_{-i}$ in Equation (3.2)
      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

3.2 Analysis of Sharpness over Landscape

We calculate the sharpness over the landscape for proposed methods on SST-2 by integrating the proposed weight flattening method with DP-trained prefix tuning. The results in Figure 3 show that all our proposed three weight flattening methods flatten the weight loss landscape. This matches the design intuition that Privacy-Flat smooths the sharp local minima of LLMs comprehensively by three aspects. Later in Section 4, we will validate how Privacy-Flat improves the performance with competitive privacy by enhancing the loss landscape.

3.3 Enhancing Flatness in Black-box Setting

While LLMs of interest are often black boxes, i.e., their weights are not accessible for training, in this section, we extend our framework to the black-box settings.

To deal with black-box settings of neural networks,

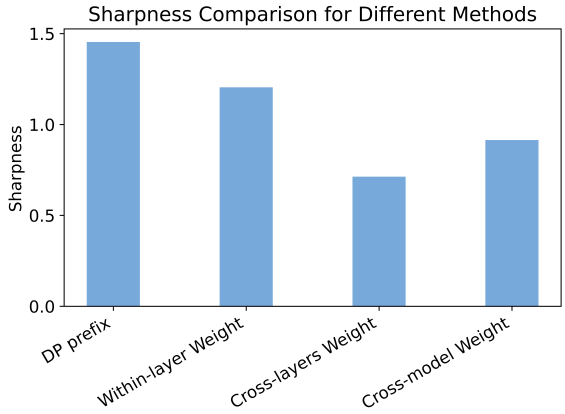


Figure 3: Sharpness for DP trained prefix tuning plus our proposed three weight flattening methods on SST-2. Our proposed model has a flatter loss landscape.

the zeroth-order (ZO) optimizers [31] are often used to estimate the gradient of neural networks using output differences without any backpropagation. To enable private training for black-box LLMs, DPZero [58] was proposed: Let $\mathbf{g}$ represent the noise sampled from Gaussian distribution $\mathcal{N}(0, \sigma^2)$, the gradient will be updated with the equation:

(3.5)
$$\hat{\nabla}\mathcal{L}(\mathbf{w};\mathcal{B}) = (cilp(\frac{\mathcal{L}(\mathbf{w}+\varepsilon\mathbf{z};\mathcal{B}) - \mathcal{L}(\mathbf{w}-\varepsilon\mathbf{z};\mathcal{B})}{2\varepsilon}) + \mathbf{g})\mathbf{z}.$$

Here, $\mathbf{z}$ is a random noise sampled from standard Gaussian distribution, ε is the perturbation scale and $\mathcal{B}$ represents the batch data. We will focus on the DPZero framework in our paper.

In the black-box setting, we consider improving through non-private duplication. Compared with the white-box setting, $\mathbf{w}_{nor}$ is also trained with the black-box setting. Note that in the black-box setting, it is impractical to improve the cross-layer weight flatness since we do not have access to the internal weights of each layer in LLMs. It is also difficult to enhance 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 the $\mathbf{v}$ [60]. Though ablating these two components, we empirically found that our Privacy-Flat still delivers the outperforming accuracy with privacy.

3.4 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 show that under the framework of DP training, 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.

4 Experiments

4.1 Experimental Settings

Datasets To assess the effectiveness of our proposed model, Privacy-Flat, we explore two principal NLP tasks, i.e., text classification and text generation, across 7 datasets: (1) For *text classification*, we engage with datasets from the GLUE benchmark [49]: SST-2 [43] for sentiment classification; MNLI [51] and QNLI [49] for sentence pair classification; QQP and TREC [48] for topic classification. (2) For *text generation*, we utilize E2E [36] and DART [35] for table-to-text generation. This selection of datasets allows us to comprehensively evaluate Privacy-Flat across a spectrum of linguistic tasks and complexities.

Setups In the white-box setting, we mainly use Roberta-base [28] and BERT [11] for encoder-only architectures and GPT-2 [40] for decoder-only architectures. In the black-box setting, we adopt Roberta-base. For the DP-SGD framework, we follow the common practice of setting the privacy budget as $\epsilon = [3, 8]$ and $\delta = \frac{1}{2|\mathcal{D}|}$ for all settings. Here ϵ and δ are only used for determining the noise level in Privacy-Flat. All experiments were run on a single RTX 4090 with 64GB memory on Ubuntu 22.04.

Training 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 1. For Privacy-Flat, we set the regularization weight λ in Equation (3.4) to 0.01 for all experiments.

Baselines For white-box settings, we mainly compare with full fine-tuning and prefix tuning under non-private and DP training; for black-box settings, we compare with prompt-tuning.

4.2 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. Following existing work [57, 16], we evaluate the privacy risks empirically by membership inference attack (MIA), which targets judging whether a data sample belongs to a training set or not. In this paper, we consider a simple but efficient

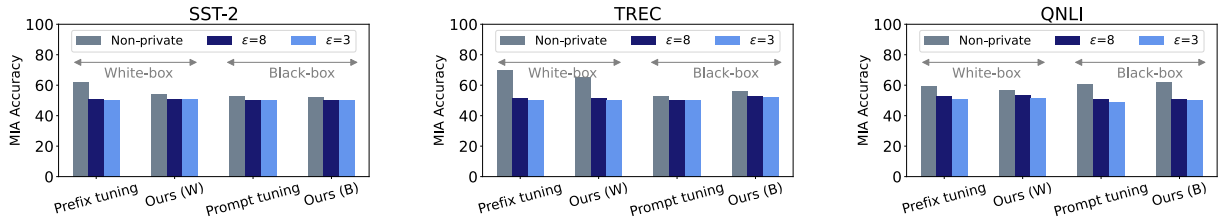


Figure 4: Comparison of MIA accuracy under both white-box and black-box settings across text classification datasets. The lower the accuracy, the lower the privacy risk. The results show that our proposed method will not affect the privacy protection for both white-box and black-box settings.

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

Table 1: Detailed hyperparameters for DP training and normal training on MNLI and E2E.

loss-based MIA [55], which considers the samples with a loss lower than a threshold as the training dataset. 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. A model with higher accuracy in MIA indicates higher privacy risks since the successes mean that the attackers may be able to reveal information about the data used to train the model. From the results in Figure 4, we have the following observations:

(1) Compared with non-private training ($\epsilon = \infty$), all DP baselines show lower accuracies against MIA, indicating better protection. This matches with existing literature that DP training lowers privacy risks [14, 25].

(2) Under the same privacy budget, Privacy-Flat shows very similar MIA accuracies with DP-trained prefixes, indicating that Privacy-Flat does not hurt the privacy protection. We prove that though our method cannot have a strict DP guarantee, our method still maintains good privacy.

4.3 Evaluation in Classification and Generation

We conduct experiments in both black-box and white-box settings. We report test accuracy in classification tasks, and BLEU and ROUGE-L for generation tasks.

4.3.1 White-box Setting

Text Classification 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 2, we provide the results of the experiment for Roberta-base with different tasks. We have the following observations:

(1) Privacy-Flat can increase the performance of DP prefix tuning significantly, and even outperforms full fine-tuning. Compared with DP prefix tuning, Privacy-Flat improves at most 8.39% on QNLI and at least 2.5% on SST-2. Though Privacy-Flat is not the best performance on BERT, Privacy-Flat still shows an improvement over prefix tuning, enjoying a much lower memory cost than full fine-tuning. This is because Privacy-Flat considers three flattening aspects that can mitigate the negative impact of DP-SGD and achieve a better trade-off between privacy and performance.

(2) Privacy-Flat can also work well across different base models. We train Privacy-Flat on Bert-base with the same tasks for Roberta-base. Similar performances as in Roberta-base are also found in BERT: Privacy-Flat outperform DP prefix tuning in all settings and bridge the gap between non-private models ($\epsilon = \infty$) and DP-trained models.

(3) Though prefix tuning can outperform full fine-tuning in some tasks under Roberta, there is no consistent winner between DP prefix tuning and DP full fine-tuning considering their performance on all datasets, which is coherent with the conclusion made in the

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 2: 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 privacy budget is highlighted. The results show that Privacy-Flat can increase the performance of DP-SGD-trained LLMs for various text classification tasks.

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 privacy budgets is highlighted. Privacy-Flat performs consistently better than strictly DP-trained methods on text generation.

previous work [24]. In comparison, Privacy-Flat achieves consistently the best performance under Roberta-base.

Text Generation For the table-to-text generation, where LLMs are asked to generate the natural language description for the given table entry. We adopt the decoder-only GPT2 for this task and the results are shown in Table 3. We have the following observations:

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

(2) For tasks with different difficulties, Privacy-Flat shows competitive or better performances. In simple tasks (E2E dataset), 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. However, 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.

4.3.2 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. We compare with the following baseline methods for prompt tuning: (1) non-private prompt tuning with zeroth order optimization method MEZO [31], (2) DP prompt tuning with zeroth order optimization method DPZero [58]. The results are shown in Table 4 across different datasets with Roberta-base. We have the following observations:

(1) Compared with the white-box setting, the bridge between the non-private model and the DP-trained model in the black-box setting becomes bigger than the bridge white-box setting in general, indicating further effort should be made to improve the stability in the black-box setting.

(2) Privacy-Flat remains comparable performance for the QQP dataset and consistently improves the performance in all other datasets. Despite the difficulties of black-box settings in calculating the gradients, Privacy-Flat still shows better accuracy under the privacy setting with better flatness.

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 with different levels of privacy.

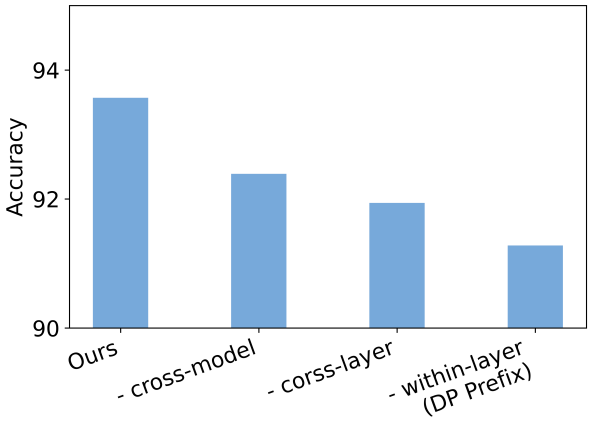


Figure 5: Influences of gradually removing different flatness methods on the classification performance w.r.t. accuracy under SST-2 dataset on Roberta-base. The higher, the better. The results show that each part of Privacy-Flat helps increase the performance.

4.4 Ablation Study To test how much each part of Privacy-Flat contributes to the final results, we conduct ablation studies to show the performance while gradually removing our methods. Specifically, we conduct experiments on SST-2 and Roberta-base. Figure 5 shows the performance of variants of our method. We can see that each component will help the performance, indicating the effectiveness of the proposed flattening methods. Note that our method will downgrade to DP-trained prefix tuning when all three aspects are removed.

4.5 Sensitivity Analysis In this section, we focus on the sensitivity on different λ. The regularization factor

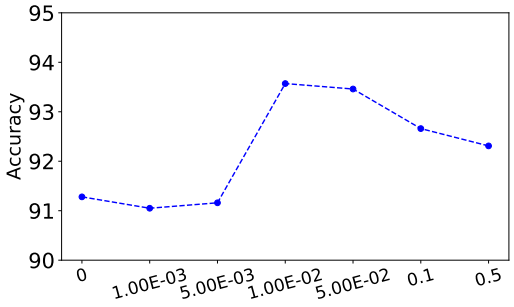


Figure 6: Influences of different values of factor λ on the classification performance w.r.t. accuracy under SST-2 dataset on Roberta-base. The higher, the better.

in Equation (3.4) 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.

5 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.

Acknowledgements

The work was partially supported by NSF awards #2421839, NAIRR #240120, #2431516. This work used AWS through Amazon Research Awards and the CloudBank project supported by National Science Foundation grant #1925001. Pingzhi Li and Tianlong Chen are supported by NIH OT2OD038045-01 and UNC SDSS Seed Grant. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies. We thank OpenAI for providing us with API credits under the Researcher Access program.

References

[1] 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*, pages 308–318, 2016.

[2] Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning*, pages 639–668. PMLR, 2022.

[3] Rohan Anil, Badih Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. Large-scale differentially private bert. *arXiv preprint arXiv:2108.01624*, 2021.

[4] 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)*, pages 2633–2650, 2021.

[5] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

[6] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

[7] Lichang Chen, Jiuhai Chen, Tom Goldstein, Heng Huang, and Tianyi Zhou. Instructzero: Efficient instruction optimization for black-box large language models. *arXiv preprint arXiv:2306.03082*, 2023.

[8] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. *arXiv preprint arXiv:2106.01548*, 2021.

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

[10] 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*, 2023.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*, 2019.

[13] Jiawei Du, Hanshu Yan, Jiashi Feng, Joey Tianyi Zhou, Liangli Zhen, Rick Siow Mong Goh, and Vincent YF Tan. Efficient sharpness-aware minimization for improved training of neural networks. *arXiv preprint arXiv:2110.03141*, 2021.

[14] 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*, pages 2665–2679, 2023.

[15] Haonan Duan, Adam Dziedzic, Nicolas Papernot, and Franziska Boenisch. Flocks of stochastic parrots: Differentially private prompt learning for large language models. *arXiv preprint arXiv:2305.15594*, 2023.

[16] Christophe Dupuy, Radhika Arava, Rahul Gupta, and Anna Rumshisky. An efficient dp-sgd mechanism for large scale nlu models. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4118–4122. IEEE, 2022.

[17] Adel Elmahdy and Ahmed Salem. Deconstructing classifiers: Towards a data reconstruction attack against text classification models. *arXiv preprint arXiv:2306.13789*, 2023.

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

[19] 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.

[20] Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas

Carlini. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv preprint arXiv:2210.17546*, 2022.

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

[22] Haoran Li, Mingshi Xu, and Yangqiu Song. Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. *arXiv preprint arXiv:2305.03010*, 2023.

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

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

[25] Yansong Li, Zhixing Tan, and Yang Liu. Privacy-preserving prompt tuning for large language model services. *arXiv preprint arXiv:2305.06212*, 2023.

[26] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.

[27] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *AI Open*, 2023.

[28] 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.

[29] 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.

[30] Lingjuan Lyu, Xuanli He, and Yitong Li. Differentially private representation for nlp: Formal guarantee and an empirical study on privacy and fairness. *arXiv preprint arXiv:2010.01285*, 2020.

[31] 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.

[32] 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.

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

[34] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017.

[35] 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.

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

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

[38] Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1314–1331. IEEE, 2020.

[39] Rahil Parikh, Christophe Dupuy, and Rahul Gupta. Canary extraction in natural language understanding models. *arXiv preprint arXiv:2203.13920*, 2022.

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

[41] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

[42] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks

against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[43] 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*, pages 1631–1642, 2013.

[44] Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 377–390, 2020.

[45] James C Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 1992.

[46] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pages 20841–20855. PMLR, 2022.

[47] 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.

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

[49] 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.

[50] Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. *arXiv preprint arXiv:2306.11698*, 2023.

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

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

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

[54] 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*, 2022.

[55] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[56] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*, 2021.

[57] Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Large scale private learning via low-rank reparametrization. In *International Conference on Machine Learning*, pages 12208–12218. PMLR, 2021.

[58] 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.

[59] Ruisi Zhang, Seira Hidano, and Farinaz Koushanfar. Text revealer: Private text reconstruction via model inversion attacks against transformers. *arXiv preprint arXiv:2209.10505*, 2022.

[60] 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.

[61] 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.