LoRA-Leak: Membership Inference Attacks Against LoRA Fine-tuned Language Models

Delong Ran[1], Xinlei He[2], Tianshuo Cong[1](✉), Anyu Wang[1], Qi Li[1], and Xiaoyun Wang[1]
[1]Tsinghua University, [2]Hong Kong University of Science and Technology (Guangzhou)

***Abstract*—Language Models (LMs) typically adhere to a "pre-training and fine-tuning" paradigm, where a universal pre-trained model can be fine-tuned to cater to various specialized domains. Low-Rank Adaptation (LoRA) has gained the most widespread use in LM fine-tuning due to its lightweight computational cost and remarkable performance. Because the proportion of parameters tuned by LoRA is relatively small, there might be a misleading impression that the LoRA fine-tuning data is invulnerable to Membership Inference Attacks (MIAs). However, we identify that utilizing the pre-trained model can induce more information leakage, which is neglected by existing MIAs. Therefore, we introduce *LoRA-Leak*, a holistic evaluation framework for MIAs against the fine-tuning datasets of LMs. *LoRA-Leak* incorporates fifteen membership inference attacks, including ten existing MIAs, and five improved MIAs that leverage the pre-trained model as a reference. In experiments, we apply *LoRA-Leak* to three advanced LMs across three popular natural language processing tasks, demonstrating that LoRA-based fine-tuned LMs are still vulnerable to MIAs (e.g., 0.775 AUC under conservative fine-tuning settings). We also applied *LoRA-Leak* to different fine-tuning settings to understand the resulting privacy risks. We further explore four defenses and find that only dropout and excluding specific LM layers during fine-tuning effectively mitigate MIA risks while maintaining utility. We highlight that under the "pre-training and fine-tuning" paradigm, the existence of the pre-trained model makes MIA a more severe risk for LoRA-based LMs. We hope that our findings can provide guidance on data privacy protection for specialized LM providers.**

***Index Terms*—Language Model, Membership Inference, LoRA Fine-tuning, Privacy.**

I. Introduction

LANGUAGE Models (LMs) have been extensively utilized in a variety of Natural Language Processing (NLP) tasks, including legal advise [1], scientific research [2], etc. Despite the belief that pre-trained LMs such as ChatGPT [3] and Llama [4] have exhibited the rudiments of Artificial General Intelligence (AGI), their data-driven nature results in sub-optimal performance in specialized domains [5]. To address this issue, the general paradigm for tailoring LMs to downstream tasks consists of two steps: *pre-training* and *fine-tuning*. The pre-training process aims to learn rich language features and structures from a massive corpus in an unsupervised way, forming the *pre-trained models* capable of mastering general language patterns. Consequently, these pre-trained models can serve as a remarkable starting point and further fine-tuned on vertical domains in a supervised way, resulting in the *specialized models* that are adept at diverse downstream tasks.

(✉)Correspondence to: Tianshuo Cong (congtianshuo@gmail.com).

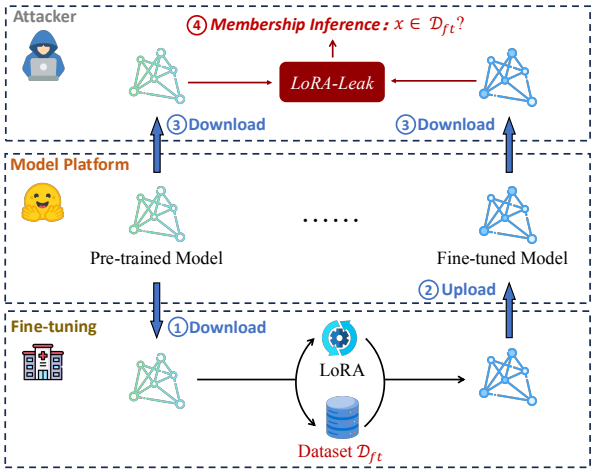

Fig. 1: Overview of *LoRA-Leak*. *LoRA-Leak* aggregates information from the specialized fine-tuned model and its pre-trained model to launch more powerful MIAs against LMs.

For example, codeLlama [5] and AstroLLaMA [6] are the fine-tuned variants of Llama-2 [4] on programming domain and astronomy domain, respectively.

The performance of the fine-tuned specialized LMs hinges on two key factors: the fine-tuning algorithms and the fine-tuning datasets (see Figure 1). Given the current scale of parameters in LMs, the computational cost of full-parameter fine-tuning is prohibitively expensive (e.g., a 16-bit full-tuning for Llama-7B requires 60GB of GPU memory [7]). This limitation has spurred the rapid development of Parameter-Efficient Fine-Tuning (PEFT) [8]. Specifically, Low-Rank Adaptation (LoRA) [9] is the state-of-the-art (SOTA) Parameter-Efficient Fine-Tuning (PEFT) framework with the highest usage. By the end of 2023, there were over 12,000 LoRA models on Hugging Face, with some receiving over one hundred thousand downloads per month [10]. Since LoRA only trains side-loaded rank-decomposition matrices while keeping the model backbones frozen, it only needs 16GB to fine-tune Llama-7B, which can be further reduced to 6GB through its 4-bit quantization version qLoRA [11]. Moreover, a high-quality fine-tuning dataset is also paramount as it serves as the specialized knowledge source and determines the upper limit of the model's capability. Considering the potential presence of privacy-sensitive information within fine-tuning datasets, such as those in financial and medical domains, a comprehensive assessment for privacy leakage associated with fine-tuning

datasets is of vital importance.

Membership Inference Attacks (MIAs) [12], in which attackers aim to determine if a specific sample was part of the training data of a target model, pose a persistent privacy threat to machine learning models. Recently, with the emergence of an increasing number of specialized LMs in the open-source model zoo (e.g., Huggingface[1]), conducting MIAs against the fine-tuning datasets of the LoRA-based fine-tuned LMs has become a prominent research focus. Since LoRA only fine-tunes a small subset of the model's parameters, recent studies suggest that their fine-tuning datasets are invulnerable to MIA [13], [14]. However, the current research fails to recognize that the publicly accessible pre-trained model could introduce additional privacy threats.

Our Work. We propose *LoRA-Leak*, a holistic evaluation framework for measuring the vulnerability of LoRA-based fine-tuned LMs against MIAs. To comprehensively assess the privacy risks of the LoRA fine-tuning dataset, we formulate three *Research Questions* (***RQs***):

- ***RQ1***: Is MIA still a serious privacy threat for LoRA-based fine-tuned LMs?
- ***RQ2***: Can incorporating pre-trained model information lead to the design of more potent MIAs?
- ***RQ3***: What LoRA fine-tuning strategies can mitigate the threat of MIAs?

To answer ***RQ1***, we propose *LoRA-Leak*, a comprehensive framework for MIA against LoRA finetuning, incorporating fifteen diverse attack methods (see Table I). Subsequently, we assess the effectiveness of MIAs in *LoRA-Leak* against nine LoRA fine-tuned LMs, developed using three widely used LMs and three practical fine-tuning datasets under a conservative setting to prevent overfitting. Our experimental results demonstrate that *LoRA-Leak* can achieve high AUC scores against LoRA-based fine-tuned LMs. For instance, the AUC scores against Llama-2 model fine-tuned on AG News [15], OAsst [16], and MedQA [17] are 0.765, 0.721, and 0.775, respectively.

To demonstrate the necessity of introducing pre-trained models for answering ***RQ2***, we compare the effectiveness of different MIAs the performance of various MIAs with and without using the pre-trained model as a reference. We observe that the calibration from the pre-trained model can consistently amplify the privacy risk (see Table II). For a more in-depth analysis, we discuss the impact of different kinds of reference models [12]. As Figure 4 shows, introducing other kinds of reference models cannot achieve the optimal attack results as introducing pre-trained models.

To address ***RQ3***, we comprehensively discuss various fine-tuning settings. We first analyze the influence of fine-tuning hyperparameters on the attack effect, such as the fine-tuning epoch and the selection of LoRA fine-tuning modules. Furthermore, we discuss four potential defenses. We first explore three traditional defense strategies, i.e., dropout, weight decay, and differential privacy (DP), in which only dropout can mitigate the risk of MIAs while preserving utility. Furthermore, in Section VI-D, we demonstrate that fine-tuning excluding specific modules can also mitigate privacy risks.

In summary, our contributions are as follows:

- We introduce *LoRA-Leak*, a comprehensive evaluation framework on MIAs against LoRA-based fine-tuned LMs.
- We propose that introducing pre-trained models into the inference attacking pipeline can effectively amplify the privacy risks.
- We explore four defenses and find that dropout and fine-tuning excluding specific layers can mitigate the threat of *LoRA-Leak*.

[1]https://huggingface.co/models.

II. Preliminaries

A. Causal Language Model (CLM)

A causal language model (CLM) is designed for next-token prediction tasks over a token space $\mathcal{T}$. It first takes a sequence of tokens $x_{1:n} \in \mathcal{T}^n$ as input and transforms each token x_i into a continuous embedding $\boldsymbol{e}_i$. These embeddings are then fed into a decoder-only transformer $\mathcal{M}$, which produces the probability of each token $x_{n+1} \in \mathcal{T}$ being the next token, i.e., $\boldsymbol{p}_n = \langle \Pr(x_{n+1}|x_{1:n}) \rangle_{x_{n+1} \in \mathcal{T}}$.

The primary purpose of such a model is to generate reasonable text completions for user inputs by repeatedly performing next-token predictions. This is achieved by training the model on a collection of token sequences in an autoregressive manner: For each training sample $x_{1:n} \in \mathcal{T}^n$, the objective is to minimize the model's perplexity (PPL) on that sequence, defined in the form of average cross-entropy loss as

$$\mathcal{L}(x_{1:n}; \mathcal{M}) = -\frac{1}{n-1} \sum_{i=1}^{n-1} \log p_{i, x_{i+1}}. \tag{1}$$

B. Low-Rank Adaptation (LoRA)

Low-Rank Adaptation (LoRA) is one of the most widely used Parameter-Efficient Fine-Tuning (PEFT) algorithms for model fine-tuning [10]. For a pre-trained model $\mathcal{M}_{pt}$, this method selects only a subset of layers for fine-tuning. For each selected layer, it freezes the pre-trained weight $W_i \in \mathbb{R}^{d \times k}$ and introduces two additional decomposition matrices $\Delta_i = (A_i, B_i) \in \mathbb{R}^{d \times r} \times \mathbb{R}^{r \times k}$ to fine-tune, where $r \ll \min(d, k)$ is the hyperparameter of rank. In the resulting fine-tuned model $\mathcal{M}_{ft}$, its layer is then represented as:

$$W_i' = W_i + A_i B_i. \tag{2}$$

C. Membership Inference Attack

Membership inference (MI) is a privacy game where an adversary $\mathcal{A}$, given access to a machine learning model $\mathcal{M}$, aims to determine whether a specific record x is part of the model's training dataset $\mathcal{D}$, i.e., $\mathcal{A}(x; \mathcal{M}) \rightarrow \{0, 1\}$. The adversary wins if and only if $\mathcal{A}(x; \mathcal{M}) = \mathbb{I}[x \in \mathcal{D}]$. To achieve this, the adversary will choose a score function $\mathcal{S}(x; \mathcal{M}) \rightarrow \mathbb{R}$ and a threshold $\tau \in \mathbb{R}$. Finally, the adversary determines membership based on the rule $\mathcal{A}(x; \mathcal{M}) = \mathbb{I}[S(x; \mathcal{M}) > \tau]$.

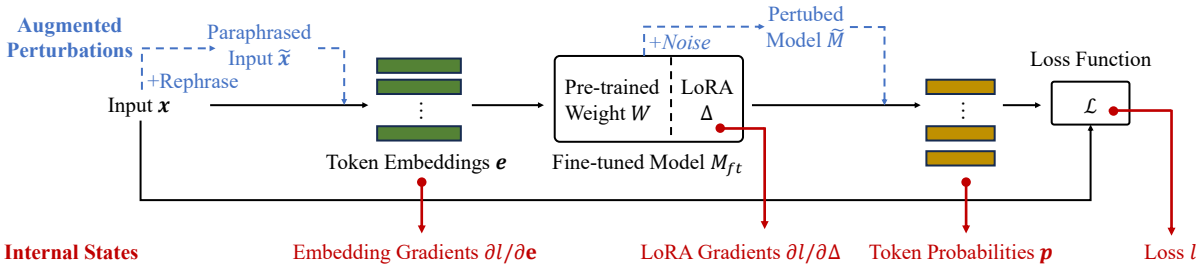

Fig. 2: The pipeline of *LoRA-Leak*. The sample x is fed to the target model to infer its membership. During the forward and back propagation, the internal states of the model can be perturbed as the dotted line indicates. Some internal states of the model can be extracted to initiate attacks as the red arrow indicates. These signals can be further calibrated by pre-trained models to obtain more effective MIAs.

TABLE I: The list of fifteen MIAs integrated within *LoRA-Leak*. *LoRA-Leak* incorporates eight well-established MIAs, and refines six of these leveraging the pre-trained model as a reference.

Attack	Internal State	Augmented Perturbation	Used to attack LoRA's $\mathcal{D}_{ft}$?	Calibrated by referencing $\mathcal{M}_{pt}$?
LOSS [18]	l	-	✓ ([13])	✓ ([19])
zlib [20]	l	-	×	× (already calibrated by zlib)
Neighborhood [21]	l	$\tilde{x}$	✓ ([22])	✓ **(Ours)**
SPV [22]	l	-	✓	× (already calibrated by $\mathcal{M}_{sp}$)
MoPe [23]	l	$\tilde{\mathcal{M}}_{ft}$	×	✓ **(Ours)**
Min-K% [24]	h	-	×	✓ **(Ours)**
Min-K%++ [25]	h	-	×	✓ **(Ours)**
GradNorm$_\theta$ [26]	$\partial l/\partial \Delta$	-	×	× (no correspondence for $\mathcal{M}_{pt}$)
GradNorm$_x$ [26]	$\partial l/\partial e$	-	×	✓ **(Ours)**

III. Related Work

A. MIAs against Neural Networks

Membership inference attack (MIA) was initially proposed to detect the training sample of image classification tasks in artificial neural networks [27]. It utilizes shadow datasets to train multiple shadow models as proxies to mimic the behavior of the target model. The prediction results of these shadow models are then collected to train a binary classifier, which learns the characteristics of membership samples. Nasr et al. [28] extend this approach to the white-box setting, where the model's internal gradients are also available. Based on this additional feature, the classifier could achieve a more accurate result. However, as the model's complexity increases, it becomes impractical to select effective and learnable features for training the classifier.

Another approach is to identify an explicit intermediate signal from the model to determine membership. For example, using the prediction correctness [29] or prediction confidence [30] as a metric for membership. The Likelihood Ratio Attack (LiRA) [12] is one of the most effective metric-based MIAs, which calibrates the membership signal by comparing the prediction confidence between the target model and the shadow models. We extend the idea to develop MIAs against fine-tuned language models.

Despite proposing various attacks, there is also theoretical research on the mechanics of membership inference. Yeom et al. [31] highlight that the primary threat of membership inference is due to overfitting. Additionally, Bentley et al. [32] further quantitatively associate the risk of membership inference with the target model's generalization gap. Therefore, we report the model's generalization gap to measure its inherent vulnerability to membership inference.

B. MIAs against Pre-trained Language Models

With the advent of Pre-trained Language Models (PLM), the membership inference attacks against their pre-training corpora are surpassing. Earlier research primarily adapted from the MIAs against neural networks, such as the LOSS attack [18] and the LiRA attack [19]. Recently, several attacks that specifically exploit the characteristics of LMs have been proposed. For example, the difference of token probabilities inspires Min-K% [24] and Min-K%++ [25], while MoPe [23] exploits the smoothness around the training points. However, the credibility of these attacks is hindered by their faulty evaluation method [33]. Because PLMs lack transparency regarding their training corpus, these works use corpora from before and after the model's knowledge cut-off date as a proxy for ground truth. Consequently, there is a distribution shift between members and non-members, allowing even classifiers without access to the model to achieve high accuracy [34]. As for the benchmark with a fair membership split, these attacks degrade to a nearly random performance against PLMs [35]. Therefore, we explore whether those attacks against PLMs can be effective against the fine-tuning corpus in LoRA fine-tuning under a fair membership split.

C. MIA against Fine-tuned Language Models

There are also MIAs against LLM's fine-tuning samples. The neighborhood attack [21] calibrates the loss signal by the average loss of rephrased samples. MIA-SPV [22] further

enhances this attack with LiRA by comparing the calibrated loss from the target model with the calibrated loss from a self-prompted shadow model. Although this attack has been reported to achieve good performance, it is important to note that their target models were trained for 10 epochs, making them inherently vulnerable to membership inference (See Section V-B). Consequently, many MIAs could achieve similar performance, rendering the advantage of this method unclear. Moreover, this attack is computational-intensive for generating self-prompt samples.

Although existing MIAs against fine-tuned LLMs are delicate and promising, they often overlook the significance of the pre-trained model. The reference attack [36], a simple yet effective approach, draws inspiration from the LiRA attack by using the pre-trained model as a shadow model to calibrate the loss signal of the target model. This suggests that referencing pre-trained models could further amplify the privacy risks associated with LoRA fine-tuned LMs. Therefore, we extend this concept to fill the gaps in all existing MIAs by calibrating their signals with their pre-trained model.

LoRA has already been the most used language model fine-tuning algorithm [10], but the studies for its privacy risks are still incomplete. Wen et al. [13] reported that LoRA fine-tuning is invulnerable to MIAs. However, their work only employs the LiRA attack [19], which may not fully reveal the MIA risks of LoRA fine-tuning. In our work, we comprehensively explore the privacy risks evoked by LoRA with fifteen MIAs against different fine-tuning settings and defense strategies.

Recently, Liu et al. [14] proposed *PreCurious*, a framework designed to amplify membership inference risks in fine-tuned language model by poisoning its pre-trained model. However, their threat model is strong because the victim must use the corrupted model provided by the adversary. In this work, we assume the victim uses the official open-source pre-trained model that is also accessible to the adversary. This is a more practical threat model in LoRA fine-tuning scenarios that still significantly amplifies privacy risks.

IV. LoRA-Leak

In this section, we introduce *LoRA-Leak*, a comprehensive framework for MIAs against LoRA fine-tuning. We begin by outlining the threat model of *LoRA-Leak*. Next, we present a systematic taxonomy that identifies the essential components of MIAs to categorize all existing MIAs within this framework. This framework proposes a neglected attacking surface that utilizing the pre-trained model as a reference could further enhance existing MIAs.

A. Threat Model

In our threat model, the victim is a model fine-tuner aiming to build a specialized CLM $\mathcal{M}_{ft}$ for a downstream task using their private dataset $\mathcal{D}_{ft}$. To achieve this, the victim first obtains a renowned PLM $\mathcal{M}_{pt}$ that is publicly released by a benign party, such as OpenAI's GPT-2 or Meta's Llama-2. Subsequently, the victim fine-tunes $\mathcal{M}_{pt}$ on $\mathcal{D}_{ft}$ using LoRA to get the resulting model $\mathcal{M}_{ft}$. Finally, the victim publicly releases the fine-tuned LoRA model $\mathcal{M}_{ft}$ in a model zoo such as Hugging Face.

Adversary's Goal. The adversary's goal is to infer whether a record x belongs to the fine-tuning dataset $\mathcal{D}_{ft}$ of the target model $\mathcal{M}_{ft}$. Note that the adversary aims to infer the membership in the victim's private fine-tuning dataset $\mathcal{D}_{ft}$, rather than the pre-training dataset that is used to train $\mathcal{M}_{pt}$.

Adversary's Knowledge. The adversary has full knowledge of the final fine-tuned LoRA model $\mathcal{M}_{ft}$ but does not know its fine-tuning details such as the fine-tuning hyperparameters. The attacker also does not know any information about the fine-tuning or pre-training datasets, even their domains. However, we additionally assume the adversary has full knowledge of the pre-trained model from which the target model was fine-tuned. This assumption is based on the fact that LoRA models must be used with their pre-trained model. Consequently, the name of the pre-trained model is typically specified in the target LoRA model's metadata or model card, allowing the adversary to effortlessly obtain this PLM by referencing the name.

Adversary's Capability. We assume that the adversary possesses sufficient GPU resources for model inference and backpropagation. Since the adversary fully possesses the fine-tuned model and its pre-trained model, that means they can self-host these models. As a result, the adversary has white-box access to the models, including all internal states during inference, such as sample loss, predicted token probabilities, and gradients, etc. However, the adversary cannot interfere with the pre-training and fine-tuning process, nor can they poison the victim's pre-trained model or fine-tuning dataset, as these actions typically require sophisticated supply-chain attacks. Some literature [22] proposed a gray-box scenario, where the adversary is limited to access partial internal states, such as the loss of the sample or prediction probabilities of each token. While these attacks appear to operate in a more constrained scenario, current inference APIs, such as OpenAI's Developer Platform[2] and Hugging Face's Inference Endpoints[3], do not provide any internal states that aligns their assumption. Therefore, we focus on the white-box scenario to fully expose the threat of membership inference in a passive setting.

B. Holistic Framework for MIAs against LMs

As discussed in Section II-C, the essence of MIA lies in selecting an appropriate score function $\mathcal{S}(x;\mathcal{M})$ that effectively differentiates between members and nonmembers. Here, we identify the key components involved in the design of $\mathcal{S}(x;\mathcal{M})$, including *Intermediate States*, *Augmented Perturbations*, and *Referenced Calibrations*. We will demonstrate how this holistic framework can encompass existing MIAs and lead to our proposed enhancements.

Internal States. To calculate $\mathcal{S}(x;\mathcal{M})$, the text sample x is fed into the fine-tuned model $\mathcal{M}$ for forward or backpropagation.

[2]https://platform.openai.com/docs/api-reference/chat/create
[3]https://huggingface.co/docs/inference-endpoints/index

During this process, the adversary can collect various *internal states* as their knowledge base for initiating the attack. As illustrated in Figure 2, there are four internal states that correlate with the sample's membership status. The *loss of sample* is denoted as $\mathcal{L}(x; M)$. Empirically, members tend to have smaller loss value than nonmembers, leading to the well-known LOSS attack [18], whose score function is defined as

$$S_{loss}(x; \mathcal{M}) = -\mathcal{L}(x; M). \tag{3}$$

The *predicted next-token probabilities for position* i are denoted as $\boldsymbol{p}_i = \langle \Pr(x_{i+1}|x_{1:i})\rangle_{x_{i+1}\in\mathcal{T}}$, which represents the model's confidence that each token being the next-token given on all previous tokens. The Min-K% [24] attack utilizes the fact that the model is less likely to predict words in the membership sentence with low probabilities. Therefore, this attack selects K% of tokens with the lowest predicted probabilities and calculates the score function as the average log likelihood of these selected tokens, i.e.,

$$\mathcal{S}_{\text{Min-K\%}}(x; \mathcal{M}) = -\frac{\sum_{x_{i+1}\in\text{Min-K\%}(x)} \log p_{i,x_{i+1}}}{|\text{Min-K\%}(x)|}. \tag{4}$$

The Min-K%++ [25] attack further utilizes the probability of the nonmember tokens, resulting in the score function

$$\mathcal{S}_{\text{Min-K\%++}}(x; \mathcal{M}) = -\frac{\sum_{x_{i+1}\in\text{Min-K\%}(x)} \frac{\log p_{i,x_{i+1}} - \mu(\log \boldsymbol{p}_i)}{\sigma(\log \boldsymbol{p}_i)}}{|\text{Min-K\%}(x)|}. \tag{5}$$

Another internal state is *the gradients of the fine-tuned model with respect to the sample loss*, i.e., $\partial\mathcal{L}/\partial\Delta(x)$. Because these gradients tend to be smaller for members, Wang et. al. [26] proposed using the norm of these gradients as the score function, i.e.,

$$\mathcal{S}_{\text{GradNorm}_\theta}(x; \mathcal{M}) = -\|\partial\mathcal{L}/\partial\Delta(x)\|. \tag{6}$$

They also suggested that *the gradients on the input embeddings* ($\partial\mathcal{L}/\partial\boldsymbol{e}$) could serve as an approximation of $\partial\mathcal{L}/\partial\Delta$, resulting in the score function

$$\mathcal{S}_{\text{GradNorm}_x}(x; \mathcal{M}) = -\|\partial\mathcal{L}/\partial\boldsymbol{e}(x)\|. \tag{7}$$

Augmented Perturbations. In addition to performing standard forward and backward propagation, the adversary may introduce *augmented perturbations* into the pipeline to collect internal states of nonmembers. There are two approaches to introduce perturbations, as represented by the dashed line in Figure 2. The neighborhood attack [21] perturbs the sample x into N paraphrased samples $\tilde{x}_1, \ldots, \tilde{x}_N$ by replacing random words with predicted ones using mask-filling models like BERT [37] or T5 [38]. The losses of these paraphrased samples are then collected to calculate the score function as

$$\mathcal{S}_{\text{Nei}}(x; \mathcal{M}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(\tilde{x}_i; \mathcal{M}) - \mathcal{L}(x; \mathcal{M}). \tag{8}$$

The MoPe attack [21] adds Gaussian noise to the model parameters, resulting N perturbed models $\tilde{\mathcal{M}}_1, \ldots, \tilde{\mathcal{M}}_N$. The sample's loss on the perturbed models are then collected to calculate the score function as

$$\mathcal{S}_{\text{MoPe}}(x; \mathcal{M}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(x; \tilde{\mathcal{M}}_i) - \mathcal{L}(x; \mathcal{M}). \tag{9}$$

Referenced Calibrations. In addition to using internal states for membership inference, the adversary can employ external references to calibrate the score function. For instance, Carlini et al. [20] proposed calibrating the sample loss with the entropy of input data evaluated by the zlib compressor, i.e.,

$$\mathcal{S}_{\text{zlib}}(x; \mathcal{M}) = |\text{zlib}(x)| - S_{LOSS}(x; \mathcal{M}). \tag{10}$$

Additionally, other models can serve as effective references for calibration. For example, the LiRA attack [19] utilizes the pre-trained model as a reference to calibrate the fine-tuned model's LOSS score function, i.e.,

$$\mathcal{S}_{\text{LiRA}}(x; \mathcal{M}) = S_{LOSS}(x; \mathcal{M}_{pt}) - S_{LOSS}(x; \mathcal{M}). \tag{11}$$

The SPV-MIA attack [22] trains a self-prompted model $\mathcal{M}_{sp}$ to calibrate the neighborhood score function, i.e.,

$$\mathcal{S}_{\text{SPV}}(x; \mathcal{M}) = S_{Nei}(x; \mathcal{M}_{sp}) - S_{Nei}(x; \mathcal{M}). \tag{12}$$

C. Pre-trained Model Calibration

As highlighted in Table I, while LoRA has emerged as the most widely adopted fine-tuning algorithm for LMs, only a limited number of existing MIAs have been explored against LoRA fine-tuning datasets. Moreover, most attacks do not calibrate their scores with external references. Given that the pre-trained model of the target LoRA model is publicly accessible, we propose *pre-trained model calibration* to the performance of existing MIAs without incurring additional costs. This technique leverages the pre-trained model as a reference to recalibrate the MIA scores.

Formally, let $\mathcal{S}$ denote a score function of an MIA targeting the LoRA fine-tuned model $\mathcal{M}$. We introduce a calibrated score function, which utilizes the pre-trained model $\mathcal{M}pt$ as a reference, defined as:

$$\mathcal{S}_{\text{pt-ref}}(x; \mathcal{M}) = \mathcal{S}(x; \mathcal{M}_{pt}) - \mathcal{S}(x; \mathcal{M}). \tag{13}$$

Here, $\mathcal{S}(x; \mathcal{M}_{pt})$ estimates $\Pr(x \in \mathcal{D}_{pt}; \mathcal{M}_{pt})$, which serves as the *a priori* probability of the membership status of x. Conversely, $\mathcal{S}(x; \mathcal{M})$ estimates $\Pr(x \in \mathcal{D}_{ft}; \mathcal{M})$. Therefore, this score function captures the variation in confidence before and after fine-tuning. Compared to the original score function derived solely from the fine-tuned model, the calibrated one can make the membership status of members and nonmembers more distinguishable.

However, among all existing MIAs, only the LOSS attack has been enhanced using this calibration technique, as demonstrated in Equation (11). Consequently, despite the scoring functions that already utilize reference calibration (S_{zlib}, S_{SPV}) and the score function incompatible to $\mathcal{M}_{pt}$ (GradNorm_θ), we can enhance five MIAs by Equation (13), including $S_{\text{Min-k\%}}$, $S_{\text{Min-k\%++}}$, S_{GradNorm_x}, S_{Nei}, and S_{MoPe}.

Ultimately, as shown in Table I, our proposed *LoRA-Leak* framework encompasses a total of fifteen MIA attacks, including five newly introduced MIAs.

V. EVALUATION

In this section, we will evaluate all membership inference attacks in *LoRA-Leak*. First, we will outline our experimental settings and considerations for membership inference evaluation. Next, we will analyze their applicability to fine-tuning datasets and demonstrate the superiority of our pre-trained model calibration method. Finally, we will discuss the privacy risk of LoRA fine-tuning under different hyperparameters using *LoRA-Leak*.

A. Experimental Settings

Models. We select three representative open-source LMs as our pre-trained models for fine-tuning, i.e., GPT-2 XL [39], Pythia-2.8B [40], and Llama-2 7B [4], with parameter sizes ranging from 1.5 billion, 2.8 billion, and 7 billion, respectively. The chosen pre-trained models have been extensively used for LoRA fine-tuning.

Datasets. We focus on the following three downstream tasks to fine-tune LMs and evaluate the MIA risks.

- **AG News** [15] is a text classification task that categorizes news articles into four classes based on their titles and contents. We leverage this dataset to simulate scenarios where LLMs are fine-tuned to adherent to specific output formats.
- **Open Assistant Conversations (OAsst)** [16] is a textual dataset containing multi-round conversations between users and AI chatbots in real-world scenarios. In our evaluation, we utilize its subset, the OpenAssistant TOP-1 Conversation Threads dataset [41], which is particularly suitable for fine-tuning LLMs into general-purpose chatbots. The conversations of this dataset are converted to ChatML format [42].
- **MedQA** [17] is an English single-choice Question Answering (QA) task for medical exams. Each question has five options to choose from. We leverage MedQA to mimic the scenarios where LLMs are fine-tuned on sensitive datasets for domain adaptation.

For the AG News and OAsst tasks, we randomly select 10,000 items to construct the fine-tuning dataset $\mathcal{D}_{ft}^{\mathrm{AG}}$ and $\mathcal{D}_{ft}^{\mathrm{OA}}$, respectively. Similarly, for the MedQA task, we randomly select 8,000 samples to build $\mathcal{D}_{ft}^{\mathrm{Med}}$. Additionally, for each of these three tasks, we randomly select 1,000 samples distinct from their respective $\mathcal{D}_{ft}$ to form their validation datasets $\mathcal{D}_{val}$ (e.g., $\mathcal{D}_{val}^{\mathrm{AG}}$).

To evaluate the effectiveness of MIAs, for each task, we randomly select 512 items from its $\mathcal{D}_{ft}$ as the members to infer. Meanwhile, we randomly select 512 items that are disjoint from its $\mathcal{D}_{ft} \cup \mathcal{D}_{val}$ as non-members.

Metrics. We focus on three key metrics: PPL@val, GAP, and AUC. These metrics assess the performance of the fine-tuned models, the susceptibility of fine-tuned models to MIAs, and the effectiveness of MIAs, respectively.

- **Model Utility:** Perplexity (PPL) reflects the model's uncertainty regarding a given text. We leverage the model's PPL on the *validation* set (denoted as PPL@val) as an indicator of its specialized utility. PPL can be calculated through Equation (1). A lower PPL@val suggests that the model exhibits better utility.
- **Overfitting Level:** The generalization gap (GAP) is defined as the perplexity difference between the fine-tuning dataset and the validation dataset, i.e.,

$$\mathrm{GAP} = \mathrm{PPL@val} - \mathrm{PPL@ft}. \tag{14}$$

  Since membership inference threat is primarily due to overfitting [31], we use this metric to evaluate the overfitting level of the fine-tuned models on the fine-tuning datasets, thereby reflecting their hardnesses against MIAs. A lower GAP value (around zero) implies the model is less overfitting, making membership inference more challenging and practical.
- **Effectiveness of MIAs:** We use the Area Under the Receiver Operating Characteristic Curve (AUC) to evaluate the effectiveness of MIAs. A higher AUC indicates that an MIA is more effective at distinguishing between members and non-members. The reason to use AUC lies in the fact that all attacks in *LoRA-Leak* predict a score of membership rather than making a hard decision. Therefore, varying thresholds yield the dynamic change of false-positive rates and true-positive rates, and AUC could capture this characteristic.

TABLE II: The AUC of different MIAs against models fine-tuned from Llama-2. We highlight the results of the pt-referenced attacks.

Attacks	AG News	OAsst	MedQA
zlib	0.640	0.530	0.575
$\mathrm{GradNorm}_{\theta}$	0.669	0.546	0.600
LOSS	0.648	0.530	0.600
↪+*Pre*	0.705	0.583	0.609
Neighborhood	0.613	0.500	0.551
↪+*Pre*	0.718	0.646	0.646
Min-K%	0.664	0.550	0.635
↪+*Pre*	0.731	0.595	0.674
Min-K%++	0.735	0.687	0.689
↪+*Pre*	0.765	0.721	0.775
MoPe	0.635	0.508	0.560
↪+*Pre*	0.731	0.561	0.640
$\mathrm{GradNorm}_{x}$	0.650	0.567	0.624
↪+*Pre*	0.679	0.588	0.613

B. Effectiveness of LoRA-Leak

Setups. In this section, we fine-tune all pre-trained models for 10 epochs. For each epoch, we record the perplexity on the fine-tuning and validation sets to monitor the extent of overfitting. Additionally, we perform *LoRA-Leak* within each epoch and report the best AUC among the eight non-referenced MIAs and six referenced MIAs. The experimental results are shown in Figure 3.

Utility of Target Models. According to Figure 3, we could observe that LoRA fine-tuning effectively reduces the perplexity

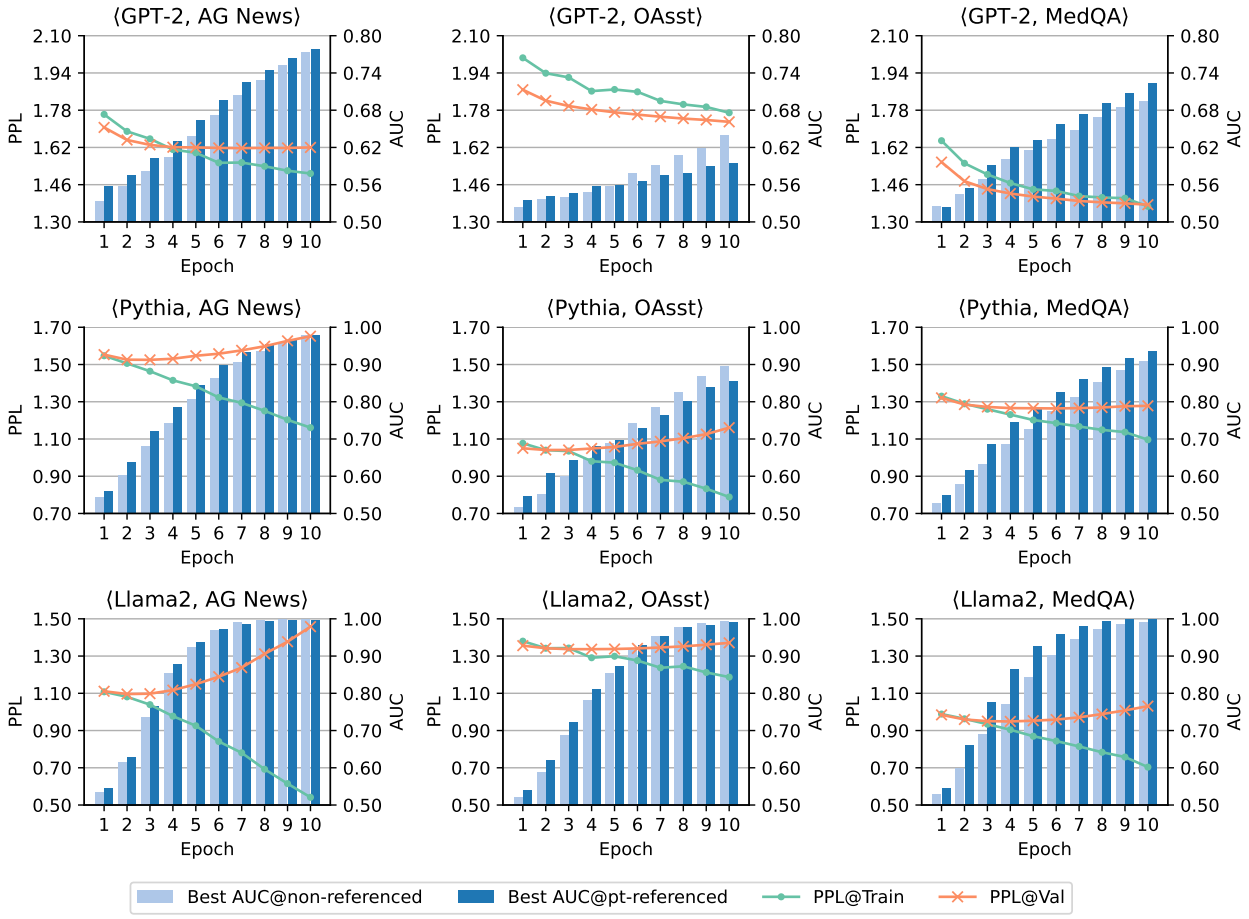

Fig. 3: The perplexity of training and validation dataset as the fine-tuning epoch increases, as well as the best AUC achieved for non-referenced and pt-referenced MIAs.

on training samples, indicating that it helps the model memorize information about the training data. However, as the number of epochs increases, the perplexity on validation samples first decreases and then starts to increase. Consequently, the model's performance on downstream tasks initially improves and then declines.

Relationship of Overfitting Level to MIA Risks. We notice that the perplexity gap between the fine-tuning set and the validation set increases as fine-tuning progresses, indicating that the model's overfitting level intensifies. Consequently, the effectiveness of all MIAs also increase with this growing gap. This observation highlights the significant impact of overfitting on the risk of membership inference. Notably, the best AUC can approach 1.0 when the model is trained for 10 epochs, especially for Llama-2. However, the advantage of non-referenced MIAs versus pre-trained model-referenced MIAs varies. For instance, among all models fine-tuned on MedQA, the pre-trained model-referenced MIA consistently outperforms the non-referenced MIA. Conversely, for models fine-tuned on OAsst, the pre-trained model-referenced MIA performs well when the model has not been severely overfitted, but the non-referenced MIA gains an advantage as overfitting intensifies. This is because the scale of the membership score shrinks as fine-tuning progresses, making the calibrated score less applicable.

Practical Considerations for Evaluating *LoRA-Leak*. As illustrated above, higher epochs can lead to overfitting, enhancing the effectiveness of MIAs and achieving higher reported metrics. However, in practical scenarios, the model will be fine-tuned for optimal performance on downstream tasks rather than for the lowest perplexity on the training set. Moreover, as shown in Figure 6 of the Supplementary Material, all attacks will achieve similarly high AUCs as overfitting intensifies, making their effectiveness indistinguishable. Therefore, we believe that the setting of 3 epochs, where the validation perplexity is relatively low, is a fair representation of the real-world risk of LoRA fine-tuning.

Non-referenced MIAs' Performance. We first perform the non-referenced MIAs as baselines. Here we fix the epoch number to 3. The AUC for these MIAs aginst Llama-2 are reported in Table II, and the AUC achieved for other models are reported in Table IX of the Supplementary Material. Our findings reveal that the attacks utilize more information could infer membership better, such as logits for Min-K and Min-K%++ and gradients for GradNorm$_\theta$ and GradNorm$_x$. Specifically, Min-K%++ consistently acts as the most effective attack for models fine-tuned from Pythia and Llama-

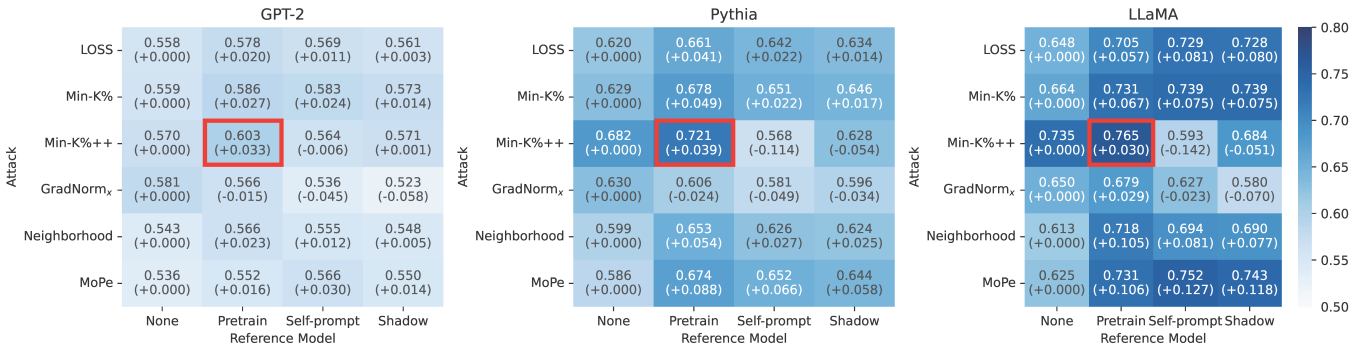

Fig. 4: The AUC achieved for three models trained on AG News, using different reference models to enhance the non-referenced MIAs. We box the highest AUC value in red.

2. However, for GPT-2, Min-K%++ only outperforms other attacks on the MedQA dataset, while GradNorm$_x$ is more effective on the AG News and OAsst datasets. Notably, some attacks perform worse than the LOSS attack, including the zlib attack, Neighborhood attack, and MoPe attack, even though they perform well on inferring membership of pre-training data [20], [21], [23]. We hypothesize that this discrepancy is due to the specific characteristics of fine-tuning data and LoRA modules: (1) Fine-tuning data often involves domain-specific knowledge rather than general corpus data. Consequently, it tends to have high entropy and hard to paraphrase, affecting the effectiveness of zlib and Neighborhood attacks. (2) The compactness of LoRA parameters makes them sensitive to perturbations, potentially impacting the effectiveness of MoPe attacks.

Pt-referenced MIAs' Performance. We further perform pt-referenced MIAs against all nine target fine-tuned LMs. As Table II and Table IX shows, we could observe that using pre-trained models as references can enhance the performance of each attack compared with the corresponding baselines in most cases. For instance, introducing pre-trained Llama-2 further enhances the Min-K%++ attack from 0.689 to 0.775 on MedQA. Additionally, even though MoPe is not the best attack among non-referenced MIAs, its pt-referenced variant performs as the best attack against GPT-2 and Pythia on OAsst dataset.

Takeaways: Using the corresponding pre-trained model as a reference can amplify the effectiveness of existing MIAs and serve as a more powerful tool for privacy auditing in the context of LoRA fine-tuning.

C. Impact of Different Reference Models

Despite using pre-trained models as a reference, some MIAs utilize other models for comparison. For instance, the LiRA attack adjusts the loss signal by comparing it to the loss of a shadow model fine-tuned on a dataset with similar distributions [12]. Additionally, Fu et al. [12] propose constructing a shadow model by prompting the target model itself. In this section, we aim to assess the effectiveness of these different reference models.

Setups. We first construct shadow models following [12] by fine-tuning three pre-trained models on the TLDR News dataset [43], which shares similar domains with AG News. Additionally, we create self-prompt models following [22] by using the first 16 words of the TLDR News dataset to prompt the target model, followed by fine-tuning three pre-trained models on the resulting corpus. We fix the downstream task as AG News. The AUC results are presented in Figure 4.

Results. Our evaluation reveals that both shadow models and self-prompt models enhance the claimed baseline attacks. However, they are generally less effective than using the pre-trained model as a reference. For example, across all attacks, using the pre-trained model as the reference could enhance Min-K%++-Ref$_{\text{pt}}$ and achieve the most effective MIA among all attacks while using another model as the reference even degrades its AUC. Moreover, we observe that some attacks, such as the Min-K% attack, Neighbourhood attack, and MoPe attack can be boosted by any reference models, resulting in AUC increases of 0.067 to 0.127 for Llama-2. Furthermore, other reference models require additional datasets and fine-tuning efforts, while the pre-trained model is naturally obtainable within the context of LoRA fine-tuning.

Takeaways: The pre-trained model is the most effective reference model with the most convenience in the context of LoRA fine-tuning.

D. Impact of Fine-tuning Modules

Setups. Considering that LoRA adapters can be added to different modules of LMs, in this part, we aim to discuss the impact of different fine-tuning choices on the effectiveness of MIAs. We fix the fine-tuning dataset to AG News and keep all other hyperparameters the same as in Appendix C.

Results. As shown in Table III, we could observe that fine-tuning different modules results in varying susceptibility to MIAs. Specifically, excluding `attention layers` (qkv) and `downscale layers` (d) from the feed-forward layers has minimal impact on the AUC of the best *LoRA-Leak* attacks. However, excluding the `upscale layer` (u) during fine-tuning could significantly reduce the best MIA AUC.

TABLE III: Discussion on fine-tuning modules. Here we report the MIA AUC results (Non-referenced→pt-referenced). We highlight the three lowest AUC results. The fine-tuning dataset is AG News.

Model	Fine-tuning Module	r	α	MIA AUC
GPT-2	qkv, o, u, d	4	8	0.581→**0.603**
	qkv, o, u	5	10	0.587→**0.608**
	qkv, o, d	5	10	0.533→**0.566**
	u, d	6	12	0.579→**0.614**
	qkv, o	10	20	0.536→**0.572**
	u	12	24	0.595→**0.626**
	d	12	24	0.530→**0.559**
	qkv	16	32	0.545→**0.573**
Pythia	qkv, o, u, d	4	8	0.682→**0.721**
	qkv, o, u	5	10	0.682→**0.723**
	qkv, o, d	5	10	0.574→**0.601**
	u, d	6	12	0.676→**0.717**
	qkv, o	10	20	0.576→**0.609**
	u	12	24	0.685→**0.728**
	d	12	24	0.553→**0.584**
	qkv	16	32	0.579→**0.609**
Llama-2	qkv, o, u, d, g	4	8	0.735→**0.765**
	qkv, o, u, g	4	8	0.675→**0.702**
	qkv, o, u, d	4	8	0.659→**0.685**
	qkv, o, d, g	4	8	0.681→**0.706**
	qkv, o, d	6	12	0.603→**0.620**
	u, d, g	6	12	0.727→**0.755**
	qkv, o	8	16	0.588→**0.601**
	u, g	9	18	0.710→**0.739**
	qkv	11	22	0.583→**0.597**
	u	18	36	0.664→**0.692**
	g	18	36	0.680→**0.706**
	d	18	36	0.574→**0.585**

For instance, for all three models, the fine-tuning modules corresponding to the three lowest MIA AUC experimental results did not involve the `upscale layer`. Meanwhile, we also evaluate whether excluding certain modules during the fine-tuning process would affect model performance. When we only fine-tune the `downscale layers`, the results of PPL@val are 1.680, 1.543, and 1.110 for GPT-2, Pythia, and Llama-2, respectively, while the results will be 1.631, 1.525, and 1.098 when we include all layers. This means the impact on model performance is relatively low when excluding certain modules. Notably, regardless of whether specific modules are excluded, incorporating information from pre-trained models consistently enhances the effectiveness of MIAs.

Takeaways: Fine-tuning different modules results in varying susceptibility to MIAs. Specifically, including the upscale layers (u) during LoRA fine-tuning makes the model more vulnerable to MIAs.

VI. Defenses

In this section, we discuss several potential defense mechanisms against *LoRA-Leak*. We first discuss three typical fine-tuning techniques that could potentially defend against MIAs, including *dropout*, *weight decay*, and *differential privacy*. Additionally, based on our findings from Section V-D, we further explore a novel defense approach that excludes vulnerable layers during LoRA fine-tuning.

Given that Llama-2 exhibits the highest susceptibility to membership inference and has become the most widely deployed pre-trained model in the real world, we focus on Llama-2 throughout this section. For ease of discussion, we explore the effectiveness of each method independently, while keeping other irrelevant settings consistent with those in Appendix C.

A. Dropout

Definition. Dropout [44] is a traditional technique used to mitigate overfitting in deep learning models. It works by randomly deactivating partial neurons and their corresponding connections during each training step. By doing so, the dependence between neurons is reduced, which in turn decreases the risk of MIAs. We leverage the dropout rate, denoted as $\eta \in [0, 1]$, to determine the proportion of trainable neurons to be dropped. The experimental results are shown in Table VIII. The AUC results for each specific MIA are shown in Figure 7.

Results. We first could observe that the fine-tuned models become less vulnerable to MIAs as the dropout rate η increases. Simultaneously, the overall performance of the models remains relatively stable even at $\eta = 0.95$, where the best MIA AUC could achieve a reduction of at most 0.154 (i.e., on the AG News dataset). One extreme case is that, when $\eta = 0.99$, the AUC of the best MIA on the OAsst dataset could even decrease to 0.543, however, the model utility also degrades. Considering that dropout also incurs negligible overhead in terms of computational cost, it is advisable to incorporate dropout with the LoRA fine-tuning process. Nevertheless, it's worth noting that introducing the pre-trained model as a reference could still result in a stronger attack even when leveraging dropout.

Takeaways: Combining dropout with LoRA can mitigate the risk of membership inference, especially in the context of a high dropout rate. Such mitigation will not compromise model utility significantly as long as the dropout rate remains within a reasonable bound.

B. Weight Decay

Definition. Krogh et al. [45] propose that adding a penalty term for large weights to the original training loss function $\mathcal{L}$ can improve the generalization of machine learning models, thereby alleviating overfitting. Given that membership inference is primarily influenced by overfitting, we thus investigate the effectiveness of weight decay as a defense. To this end, the modified loss function $\mathcal{L}_{decay}$ for the LoRA fine-tuning can be defined as

$$\mathcal{L}_{decay} = \mathcal{L} + \frac{\lambda}{2}\|\Delta\|^2, \tag{15}$$

where Δ represents the weights of the LoRA modules, and λ is a hyperparameter that represents the weight decay rate.

Setups. We employ the Adam optimizer with the decoupled weight decay (AdamW) proposed by Loshchilov et al. [46] to fine-tune Llama-2 on three datasets. During fine-tuning,

TABLE IV: Results on weight decay defense. We report the best MIA AUC results (non-referenced→pt-referenced) with varying weight decay rates (λ).

λ	AG News			OAsst			MedQA		
	PPL@val	GAP	Best AUC	PPL@val	GAP	Best AUC	PPL@val	GAP	Best AUC
w/o	1.098	0.059	0.735→**0.765**	1.041	0.007	0.687→**0.721**	0.950	0.021	0.689→**0.775**
10^{-4}	1.099	0.060	0.736→**0.764**	1.043	0.008	0.687→**0.722**	0.953	0.023	0.689→**0.778**
10^{-3}	1.098	0.058	0.733→**0.764**	1.043	0.008	0.685→**0.721**	0.952	0.024	0.688→**0.776**
10^{-2}	1.099	0.059	0.736→**0.763**	1.042	0.008	0.688→**0.723**	0.952	0.023	0.689→**0.775**
10^{-1}	1.098	0.059	0.737→**0.766**	1.043	0.008	0.687→**0.721**	0.953	0.024	0.690→**0.777**

TABLE V: Results on differential privacy defense. We report the best MIA AUC results (non-referenced→pt-referenced) with varying privacy budgets (ϵ).

ϵ	AG News			OAsst			MedQA		
	PPL@val	GAP	Best AUC	PPL@val	GAP	Best AUC	PPL@val	GAP	Best AUC
w/o	1.098	0.059	0.735→**0.765**	1.041	0.007	0.687→**0.721**	0.950	0.021	0.689→**0.775**
0.1	1.969	-0.031	0.521→**0.531**	1.255	-0.054	0.508→**0.510**	1.473	-0.004	**0.516**→**0.516**
1.0	1.256	-0.015	0.519→**0.527**	1.093	-0.039	**0.520**→0.506	1.134	-0.010	**0.515**→0.506
10	1.223	-0.011	0.521→**0.539**	1.080	-0.034	0.524→**0.525**	1.090	-0.009	**0.527**→0.520

TABLE VI: Results on excluding vulnerable layers-based defense. We report the best MIA AUC results (non-referenced→pt-referenced) with excluding varying layers.

Target Module	r	α	AG News			OAsst			MedQA		
			PPL@val	GAP	Best AUC	PPL@val	GAP	Best AUC	PPL@val	GAP	Best AUC
qkv, o, u, g, d	4	8	1.099	0.059	0.735 →**0.765**	1.043	0.009	0.687→**0.721**	0.952	0.024	0.689 →**0.775**
qkv, o, u, d	4	8	1.098	0.037	0.659→**0.685**	1.043	-0.003	0.632→**0.672**	0.958	0.013	0.633→**0.710**
qkv, o, g, d	4	8	1.097	0.041	0.681→**0.706**	1.043	-0.001	0.645→**0.679**	0.957	0.017	0.652→**0.734**
qkv, o, d	6	12	1.098	0.020	0.603→**0.620**	1.045	-0.016	0.595→**0.628**	0.969	0.003	0.592→**0.654**

we vary the weight decay rate λ from 10^{-4} to 10^{-1}, while also considering models without any weight decay. The results of the best MIA AUC and model utilities are summarized in Table IV. The AUC results of each MIA are shown in Figure 8.

Results. Our results reveal that applying weight decay has no significant impact on the model in terms of both MIA risks and performance. For instance, all the best AUC fluctuates less than 0.010 compared to the model without any weight decay. Additionally, the perplexity on the validation set increases by at most 0.003.

Takeaways: In the context of LoRA fine-tuning, weight decay cannot work for mitigating membership inference risks. This conclusion aligns with [47], which demonstrates that weight decay can even exacerbate the risk of MIAs for Convolutional Neural Networks (CNNs).

C. Differential Privacy (DP)

Definition. Differential Privacy (DP) [48] is a classical privacy protection technique that provides rigorous indistinguishability for a single entry. Since DP offers provable protection in terms of dataset privacy, it has become a natural defense against MIAs. Specifically, a randomized algorithm $\mathcal{M}$ with output space O achieves ϵ-differential privacy if

$$\Pr[\mathcal{M}(\mathcal{D}) \in O] \leq e^{\epsilon} \Pr[\mathcal{M}(\mathcal{D}') \in O] + \delta$$

holds for any two adjacent databases $\mathcal{D}$ and $\mathcal{D}'$ that differ only at one entry. The parameter ϵ represents the privacy budget. A smaller ϵ indicates stronger privacy guarantees.

Setups. We employ DPLoRA [49] implemented by DP-Transformers [50] to fine-tune Llama-2, varying the privacy budget ϵ from 0.1 to 10. The results related to DP are summarized in Table V. Additionally, the training and validation loss during the fine-tuning process is depicted in Figure 11, and the AUC results of each MIA are shown in Figure 9.

Results. We conduct the analysis from two perspectives: the effectiveness of mitigating MIA and the impact on model performance. Regarding the effectiveness of DP, we observe that introducing DP can significantly reduce the susceptibility to MIAs: across all three datasets, the best AUCs among all MIAs drop to $\sim$0.5 after applying DP, whereas the best AUCs are above 0.7 without DP. As the privacy budget decreases, the AUC of MIAs slightly decreases. However, we notice that a smaller privacy budget does not provide substantial gains of MIA protection compared to larger ϵ. For instance, on the AG News dataset, using DP with $\epsilon = 0.1$ only decreases the best MIA AUC by 0.008. Moreover, as Figure 9 shows, non-referenced attacks occasionally outperform the pt-referenced attacks, but the differences remain marginal. This behavior suggests that DP effectively renders all attacks akin to random guessing. Furthermore, though DP is effective in relieving *LoRA-Leak*, the enhanced privacy comes at a significant cost to

model utility. Even with a large privacy budget (e.g., $\epsilon = 10$), models still experience noticeable performance degradation, which deteriorates further as ϵ decreases (see Figure 11). Besides the performance hit, we observe that the loss converges more slowly after applying DP with $\epsilon = 1.0$ and $\epsilon = 10$, and it converges at an even slower rate when $\epsilon = 0.1$. Additionally, DP introduces substantial computational overhead, i.e., fine-tuning on the AG News, OAsst, and MedQA datasets with DP takes 31×, 7×, and 11× longer runtime, respectively, compared to fine-tuning without DP.

Takeaways: While DP achieves nearly perfect defense against *LoRA-Leak*, its performance impact and computational cost make it impractical for real-world deployment.

D. Excluding Vulnerable layers

In light of the findings from Section V-D, we identify that some modules are the key contributors to membership inference vulnerability, thus we regard excluding these vulnerable layers as a new defense strategy for LoRA fine-tuning.

Setups. Recall that fine-tuning the `up` (u) and `gate` (g) layers of Llama-2 can amplify the risks of MIAs. To mitigate this, we fine-tune Llama-2 across all three datasets using all LoRA modules (i.e., qkv, o, u, g, d) excluding u, g, or both of them. Note that we adjust their rank r to ensure that the numbers of the tuned parameters remain roughly the same across all models. The best MIA AUC results are summarized in Table VI. The AUC results of each MIA when excluding certain layers are shown in Figure 10.

Results. First, we can observe that excluding just one of the vulnerable modules can only reduce the best MIA AUC by 0.041 to 0.080, though removing the *gate* layers has a more pronounced impact compared to removing the *up* layers. When both vulnerable modules are excluded, the best AUC values decrease significantly by 0.121 to 0.145. Compared to dropout defense, excluding one of the modules roughly corresponds to the impact of a dropout rate with $\eta = 0.85$, and excluding both modules is similar to a dropout rate with $\eta = 0.95$. However, pt-referenced MIAs still consistently perform as the most effective attack. Regarding model utility, excluding these layers has minimal impact when fine-tuning on the AG News and OAsst datasets. However, there is a moderate performance downgrade when fine-tuning on MedQA. This discrepancy may be due to the specific correlation between the ability of medical knowledge and these excluded modules.

Takeaways: Excluding the vulnerable layers provides a practical defense against MIAs. However, performance degradation may occur when model knowledge is correlated with the excluded modules.

VII. Limitations

Closed-source Language Models. Nowadays, closed-source large models such as ChatGPT have progressively made fine-tuning capabilities available through APIs for data uploads [51]. However, the output information of closed-source LLMs is insufficient to initiate MIAs. For instance, OpenAI API [51] and HuggingFace Serverless Inference API [52] do not offer access to the loss on inputs, which is crucial for executing the *so-called* black-box MIAs [18]. Additionally, the fine-tuning algorithms used by closed-source models are not publicly available, and users cannot flexibly design hyperparameters. Therefore, our work focuses on attacking open-source language models. We leave the design of label-only membership inference attacks [53], [54] against language models as our future work.

The Scale of The Target Models. Our evaluation includes a diverse set of models ranging in parameter size from 125M to 13B. While we acknowledge that experimenting with even larger language models would be an exciting opportunity to explore more advanced capabilities, such an endeavor would exceed our current technical resources. Moreover, given the scope of our study, we believe that the additional insights gained from models with significantly larger parameter sizes would be marginal, particularly in relation to the increased computational and infrastructural demands they would impose. Therefore, we chose to focus on models within this parameter range, which strikes an optimal balance between technical feasibility and the value of the insights.

VIII. Conclusion

In this paper, we propose *LoRA-Leak*, a comprehensive evaluation framework for MIAs against LMs. In *LoRA-Leak*, we consider eight non-referenced MIAs and six pt-referenced MIAs, which provides a systematic quantification for membership leakage. By conducting experiment on three practical datasets with three different pre-trained language models, we present the superiority of the pt-referenced MIA attacks, which achieves the best performance among existing MIAs. Meanwhile, to demonstrate the generality of our insights, we further fine-tune the LMs with LoRA variants, and launch *LoRA-Leak* against these fine-tuned models. We find that DoRA will slightly increase of risk of MIA, while qLoRA could mitigate MIA with the degrade of performance. Additionally, we discuss four defenses and find that excluding specific LM layers and dropout can mitigate privacy risks. We hope our work can benefit the community by presenting comprehensive insights for auditing the privacy risks of LMs, and provide valuable insights for selecting the optimal setting of LoRA fine-tuning to mitigate the risk of MIA.

REFERENCES

[1] D. Cheng, S. Huang, and F. Wei, "Adapting large language models via reading comprehension," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=y886UXPEZ0

[2] T. D. Nguyen, Y.-S. Ting, I. Ciuca, C. O'Neill, Z.-C. Sun, M. Jabłońska, S. Kruk, E. Perkowski, J. Miller, J. J. J. Li, J. Peek, K. Iyer, T. Rozanski, P. Khetarpal, S. Zaman, D. Brodrick, S. J. Rodriguez Mendez, T. Bui, A. Goodman, A. Accomazzi, J. Naiman, J. Cranney, K. Schawinski, and R. Raileanu, "AstroLLaMA: Towards specialized foundation models in astronomy," in *Proceedings of the Second Workshop on Information Extraction from Scientific Publications*, T. Ghosal, F. Grezes, T. Allen, K. Lockhart, A. Accomazzi, and S. Blanco-Cuaresma, Eds. Bali, Indonesia: Association for Computational Linguistics, Nov. 2023, pp. 49–55. [Online]. Available: https://aclanthology.org/2023.wiesp-1.7

[3] OpenAI, "Chatgpt," https://openai.com/index/chatgpt/, 2022.

[4] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, "Llama 2: Open foundation and fine-tuned chat models," 2023.

[5] B. Roziere, J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. E. Tan, Y. Adi, J. Liu, T. Remez, J. Rapin *et al.*, "Code llama: Open foundation models for code," *arXiv preprint arXiv:2308.12950*, 2023.

[6] T. D. Nguyen, Y.-S. Ting, I. Ciucă, C. O'Neill, Z.-C. Sun, M. Jabłońska, S. Kruk, E. Perkowski, J. Miller, J. Li, J. Peek, K. Iyer, T. Różański, P. Khetarpal, S. Zaman, D. Brodrick, S. J. R. Méndez, T. Bui, A. Goodman, A. Accomazzi, J. Naiman, J. Cranney, K. Schawinski, and UniverseTBD, "Astrollama: Towards specialized foundation models in astronomy," 2023. [Online]. Available: https://arxiv.org/abs/2309.06126

[7] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, Z. Luo, Z. Feng, and Y. Ma, "Llamafactory: Unified efficient fine-tuning of 100+ language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Bangkok, Thailand: Association for Computational Linguistics, 2024. [Online]. Available: http://arxiv.org/abs/2403.13372

[8] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, and B. Bossan, "Peft: State-of-the-art parameter-efficient fine-tuning methods," https://github.com/huggingface/peft, 2022.

[9] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[10] T. Dong, M. Xue, G. Chen, R. Holland, Y. Meng, S. Li, Z. Liu, and H. Zhu, "The philosopher's stone: Trojaning plugins of large language models," 2024. [Online]. Available: https://arxiv.org/abs/2312.00374v2

[11] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *arXiv preprint arXiv:2305.14314*, 2023.

[12] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramèr, "Membership inference attacks from first principles," in *2022 IEEE Symposium on Security and Privacy (SP)*, 2022, pp. 1897–1914.

[13] R. Wen, T. Wang, M. Backes, Y. Zhang, and A. Salem, "Last one standing: A comparative analysis of security and privacy of soft prompt tuning, lora, and in-context learning," 2023.

[14] R. Liu, T. Wang, Y. Cao, and L. Xiong, "Precurious: How innocent pre-trained language models turn into privacy traps," 2024.

[15] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf

[16] A. Köpf, Y. Kilcher, D. von Rütte, S. Anagnostidis, Z.-R. Tam, K. Stevens, A. Barhoum, N. M. Duc, O. Stanley, R. Nagyfi, S. ES, S. Suri, D. Glushkov, A. Dantuluri, A. Maguire, C. Schuhmann, H. Nguyen, and A. Mattick, "Openassistant conversations - democratizing large language model alignment," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23. Red Hook, NY, USA: Curran Associates Inc., 2024.

[17] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, and P. Szolovits, "What disease does this patient have? a large-scale open domain question answering dataset from medical exams," *arXiv preprint arXiv:2009.13081*, 2020.

[18] A. Jagannatha, B. P. S. Rawat, and H. Yu, "Membership inference attack susceptibility of clinical language models," 2021. [Online]. Available: https://arxiv.org/abs/2104.08305

[19] F. Mireshghallah, K. Goyal, A. Uniyal, T. Berg-Kirkpatrick, and R. Shokri, "Quantifying privacy risks of masked language models using membership inference attacks," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 8332–8347. [Online]. Available: https://aclanthology.org/2022.emnlp-main.570

[20] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson *et al.*, "Extracting training data from large language models," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2633–2650.

[21] J. Mattern, F. Mireshghallah, Z. Jin, B. Schoelkopf, M. Sachan, and T. Berg-Kirkpatrick, "Membership inference attacks against language models via neighbourhood comparison," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 11 330–11 343. [Online]. Available: https://aclanthology.org/2023.findings-acl.719

[22] W. Fu, H. Wang, C. Gao, G. Liu, Y. Li, and T. Jiang, "Membership inference attacks against fine-tuned large language models via self-prompt calibration," in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37. Curran Associates, Inc., 2024, pp. 134 981–135 010. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/f36ad694188bb4c4bbbd61e2038e069e-Paper-Conference.pdf

[23] M. Li, J. Wang, J. Wang, and S. Neel, "MoPe: Model perturbation based privacy attacks on language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 13 647–13 660. [Online]. Available: https://aclanthology.org/2023.emnlp-main.842

[24] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, and L. Zettlemoyer, "Detecting pretraining data from large language models," 2023.

[25] J. Zhang, J. Sun, E. Yeats, Y. Ouyang, M. Kuo, J. Zhang, H. Yang, and H. Li, "Min-k%++: Improved baseline for detecting pre-training data from large language models," *arXiv preprint arXiv:2404.02936*, 2024.

[26] J. G. Wang, J. Wang, M. Li, and S. Neel, "Pandora's white-box: Increased training data leakage in open llms," 2024.

[27] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," in *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2017, pp. 3–18.

[28] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning," in *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2019, pp. 1021–1035.

[29] K. Leino and M. Fredrikson, "Stolen memories: Leveraging model memorization for calibrated White-Box membership inference," in *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 1605–1622. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/leino

[30] L. Song, R. Shokri, and P. Mittal, "Privacy risks of securing machine learning models against adversarial examples," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 241–257. [Online]. Available: https://doi.org/10.1145/3319535.3354211

[31] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, 2018, pp. 268–282.

[32] J. W. Bentley, D. Gibney, G. Hoppenworth, and S. K. Jha, "Quantifying membership inference vulnerability via generalization gap and other model metrics," 2020. [Online]. Available: https://arxiv.org/abs/2009.05669

[33] M. Meeus, I. Shilov, S. Jain, M. Faysse, M. Rei, and Y.-A. de Montjoye, "Sok: Membership inference attacks on llms are

rushing nowhere (and how to fix it)," 2024. [Online]. Available: https://arxiv.org/abs/2406.17975

[34] D. Das, J. Zhang, and F. Tramèr, "Blind baselines beat membership inference attacks for foundation models," 2024. [Online]. Available: https://arxiv.org/abs/2406.16201

[35] M. Duan, A. Suri, N. Mireshghallah, S. Min, W. Shi, L. Zettlemoyer, Y. Tsvetkov, Y. Choi, D. Evans, and H. Hajishirzi, "Do membership inference attacks work on large language models?" 2024. [Online]. Available: https://arxiv.org/abs/2402.07841

[36] F. Mireshghallah, A. Uniyal, T. Wang, D. Evans, and T. Berg-Kirkpatrick, "Memorization in nlp fine-tuning methods," 2022. [Online]. Available: https://arxiv.org/abs/2205.12506

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[38] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[39] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[40] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff *et al.*, "Pythia: A suite for analyzing large language models across training and scaling," in *International Conference on Machine Learning*. PMLR, 2023, pp. 2397–2430.

[41] OpenAssistant, "Openassistant top-1 conversation threads," 2023. [Online]. Available: https://huggingface.co/datasets/OpenAssistant/oasst_top1_2023-08-25

[42] OpenAI, "Chat markup language," https://github.com/openai/openai-python/blob/284c1799070c723c6a553337134148a7ab088dd8/chatml.md, 2020.

[43] J. Belveze, "Tl;dr news dataset," 2023. [Online]. Available: https://huggingface.co/datasets/JulesBelveze/tldr_news

[44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[45] A. Krogh and J. Hertz, "A simple weight decay can improve generalization," *Advances in neural information processing systems*, vol. 4, 1991.

[46] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[47] Y. Kaya, S. Hong, and T. Dumitras, "On the effectiveness of regularization against membership inference attacks," 2020.

[48] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, S. Halevi and T. Rabin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.

[49] D. Yu, S. Naik, A. Backurs, S. Gopi, H. A. Inan, G. Kamath, J. Kulkarni, Y. T. Lee, A. Manoel, L. Wutschitz, S. Yekhanin, and H. Zhang, "Differentially private fine-tuning of language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=Q42f0dfjECO

[50] L. Wutschitz, H. A. Inan, and A. Manoel, "dp-transformers: Training transformer models with differential privacy," https://www.microsoft.com/en-us/research/project/dp-transformers, August 2022.

[51] OpenAI, "Openai api reference," 2024, accessed: 2024-09-05. [Online]. Available: https://platform.openai.com/docs/api-reference/chat

[52] H. Face, "Hugging face api inference documentation," 2024, accessed: 2024-09-05. [Online]. Available: https://huggingface.co/docs/api-inference/index

[53] C. A. C. Choo, F. Tramèr, N. Carlini, and N. Papernot, "Label-Only Membership Inference Attacks," in *International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 1964–1974.

[54] Z. Li and Y. Zhang, "Membership Leakage in Label-Only Exposures," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2021, pp. 880–895.

[55] L. von Werra, Y. Belkada, L. Tunstall, E. Beeching *et al.*, "Trl: Transformer reinforcement learning," https://github.com/huggingface/trl, 2020.

[56] S.-Y. Liu, C.-Y. Wang, H. Yin, P. Molchanov *et al.*, "Dora: Weight-decomposed low-rank adaptation," 2024. [Online]. Available: https://arxiv.org/abs/2402.09353

[57] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3045–3059. [Online]. Available: https://aclanthology.org/2021.emnlp-main.243

[58] H. Liu, D. Tam, M. Muqeeth, J. Mohta *et al.*, "Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave *et al.*, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 1950–1965. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/0cde695b83bd186c1fd456302888454c-Paper-Conference.pdf

APPENDIX

DATASET EXAMPLES

A. AG News

> Below is a news article. Please classify it under one of the following classes (World, Business, Sports, Sci/Tech).
>
> ### Article: Bangladesh paralyzed by strikes Opposition activists have brought many towns and cities in Bangladesh to a halt, the day after 18 people died in explosions at a political rally.
>
> ### Class: World

B. OAsst

> <|im_start|>user
> Tell me a knock-knock joke.<|im_end|>
> <|im_start|>assistant
> Knock knock!<|im_end|>
> <|im_start|>user
> Who's there?<|im_end|>
> <|im_start|>assistant
> Boo.<|im_end|>

C. MedQA

> Please answer the letter of option truthfully.
>
> ### Question: Which of the following compounds is most responsible for the maintenance of appropriate coronary blood flow??
>
> ### Options: 'A': 'Epinephrine', 'B': 'Norepinephrine', 'C': 'Histamine', 'D': 'Nitric oxide', 'E': 'VEGF'
>
> ### Answer: D

DEFAULT FINE-TUNING SETTINGS OF LORA

We leverage the Supervised Fine-Tuning (SFT) command provided by the Transformer Reinforcement Learning (TRL) library [55] for LoRA fine-tuning. The LoRA modules are added on all linear layers of the pre-trained model, including both the self-attention modules and the feed-forward modules of the transformer. These modules are configured with a rank (r) of 4, and their scaling factor (α) is set to be twice of r. Each model is trained with a batch size of 16 and a dropout rate of 5%. The default fine-tuning epoch number is 3. We employ the AdamW optimizer [46], with a fixed learning rate of 10^{-4} without using the weight decay technique. The text sequences are truncated to a maximum of 1024 tokens, without resorting to sequence packing or input masking techniques. All other hyper-parameters are set as the default values of the script.

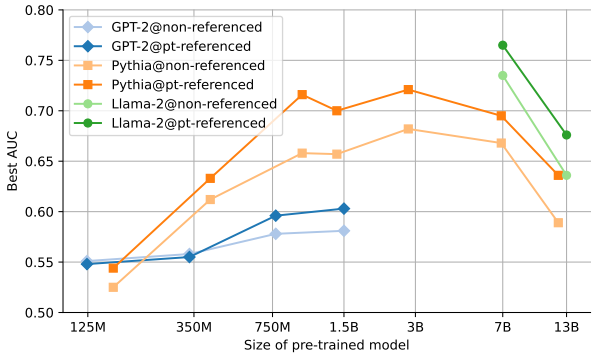

Fig. 5: The best AUC scores for non-referenced and pt-referenced MIAs on models fine-tuned on AG News, across various pre-trained model sizes.

IMPACT OF LORA VARIANTS

Setups. Given that multiple variants of LoRA have been developed to date, their associated privacy threats also urgently require assessment. We here explore two popular LoRA variants, i.e., Weight-Decomposed Low-Rank Adaptation (DoRA) [56] and qLoRA [11]. Specifically, we consider the Int8 quantization and FP4 quantization settings for qLoRA. The pre-trained model in this part is Llama-2. We evaluate both the membership inference risks and the performance of these LoRA variants.

Results. As Table VII illustrates, using DoRA to fine-tune models slightly increases the effectiveness of *LoRA-Leak* by 0.004 to 0.008. However, the fine-tuned models experience a slight decrease in overall performance. For qLoRA, the effectiveness of *LoRA-Leak* decreases by -0.001 to 0.014 for Int8 quantization, and FP4 quantization leads to a further decrease of 0.024 to 0.030. Additionally, qLoRA fine-tuning also exhibits performance degradation. Therefore, we could conclude that lower-precision quantization for qLoRA reduces vulnerability but at the cost of greater performance degradation.

Takeaways: Introducing pre-trained models can enhance the effectiveness of MIA attacks across different LoRA variants, indicating the presence of general privacy vulnerabilities within the LoRA paradigm.

IMPACT OF THE SIZE OF PRE-TRAINED MODEL

To investigate the MIA risks associated with varying scales of pre-trained language models, we further fine-tune different sizes of pre-trained models on the AG News dataset using LoRA, including GPT-2 (124M, 335M, 774M, and 1.5B), Pythia (160M, 410M, 1B, 1.4B, 2.8B, 6.9B, and 12B), and Llama-2 (7B and 13B). We follow the same fine-tuning settings as in Appendix C while selecting different rank r so as to maintain approximately the same number of tuned parameters. As illustrated in Figure 5, the MIA risk increases as the size of the pre-trained model scales up to around one billion (1B) parameters but decreases for models larger than

TABLE VII: Results on LoRA variants. We report the best MIA AUC (non-referenced→pt-referenced). The model is Llama-2.

Method	AG News			OAsst			MedQA		
	PPL@val	GAP	Best AUC	PPL@val	GAP	Best AUC	PPL@val	GAP	Best AUC
LoRA	1.098	0.059	0.735→**0.765**	1.041	0.006	0.687→**0.721**	0.950	0.021	0.689→**0.775**
DoRA	1.099	0.061	0.743→**0.769**	1.043	0.009	0.689→**0.725**	0.958	0.029	0.698→**0.783**
qLoRA (Int8)	1.100	0.057	0.722→**0.751**	1.044	0.007	0.683→**0.722**	0.951	0.021	0.686→**0.773**
qLoRA (FP4)	1.104	0.055	0.700→**0.735**	1.054	0.008	0.662→**0.695**	0.955	0.022	0.670→**0.751**

TABLE VIII: Results on dropout defense. We report the best MIA AUC results (non-referenced→pt-referenced) with the varying dropout rates (η).

η	AG News			OAsst			MedQA		
	PPL@val	GAP	Best AUC	PPL@val	GAP	Best AUC	PPL@val	GAP	Best AUC
w/o	1.099	0.059	0.735→**0.764**	1.043	0.009	0.689→**0.724**	0.952	0.024	0.692→**0.779**
0.05	1.098	0.059	0.736→**0.765**	1.041	0.007	0.687→**0.721**	0.950	0.021	0.689→**0.775**
0.25	1.098	0.056	0.730→**0.758**	1.043	0.007	0.681→**0.715**	0.953	0.023	0.689→**0.776**
0.45	1.097	0.051	0.714→**0.741**	1.043	0.005	0.673→**0.711**	0.953	0.022	0.682→**0.769**
0.65	1.097	0.046	0.697→**0.724**	1.042	0.001	0.656→**0.693**	0.954	0.020	0.676→**0.759**
0.85	1.096	0.033	0.661→**0.686**	1.042	-0.007	0.626→**0.663**	0.955	0.016	0.655→**0.736**
0.95	1.098	0.014	0.595→**0.610**	1.043	-0.018	0.580→**0.611**	0.959	0.009	0.621→**0.691**
0.99	1.108	-0.003	0.543→**0.553**	1.050	-0.030	0.523→**0.543**	0.970	-0.006	0.555→**0.602**

this range. This is because as the number of model parameters increases, the model's expressive power strengthens, leading to an increased degree of overfitting and thus "memorizing" more details of the fine-tuning data. This "memorization" heightens the risk of the model being susceptible to MIAs. However, once the model parameters reach a certain scale, the risk of overfitting may actually diminish, particularly when the model becomes more regularized or exhibits improved generalization capabilities. Nevertheless, under all circumstances, pt-referenced MIAs expose the MIA risk more thoroughly than non-referenced MIAs in terms of best AUC scores.

OTHER PEFT METHODS

While our work primarily focuses on LoRA-based methods, we discuss the applicability of *LoRA-Leak* to other PEFT methods in this section. We fine-tune the Llama-2 model on AG News using both Prompt Tuning [57] and IA3 [58]. For Prompt Tuning, we set the number of virtual prompts as 20 tokens, initialized with "Predict the topic of this news is World, Sports, Business or Sci/Tech". The best AUCs for non-referenced and pt-reference attacks are 0.511 and 0.546, respectively. For IA3, we add the learned vectors to all attention and feed-forward layers. The best AUC for non-referenced and pt-reference attacks are 0.517 and 0.551, respectively. Overall, the risks associated with these PEFT methods remain relatively minor. This may be due to the very small parameter sizes tuned. For instance, Prompt Tuning and IA3 only tuned 82k and 1.5M parameters, respectively, which is far less than LoRA with 10M parameters. Therefore, although *LoRA-Leak* does expose more MIA risks in these PEFT methods, their vulnerability to MIA remains minimal.

TABLE IX: The AUC of different MIAs against models fine-tuned from GPT-2 and Pythia. We highlight the results of the pt-referenced attacks.

Attack	GPT-2			Pythia		
	AG News	OAsst	MedQA	AG News	OAsst	MedQA
zlib	0.549	0.514	0.536	0.609	0.520	0.536
GradNorm$_\theta$	0.568	0.512	0.545	0.626	0.533	0.590
LOSS	0.558	0.501	0.544	0.620	0.508	0.579
↪+*Pre*	0.578	0.524	0.547	0.661	0.555	0.583
Neighborhood	0.543	0.496	0.519	0.599	0.497	0.536
↪+*Pre*	0.566	0.523	0.542	0.653	0.592	0.585
Min-K%	0.559	0.510	0.548	0.629	0.525	0.592
↪+*Pre*	0.586	0.531	0.558	0.678	0.564	0.611
Min-K%++	0.570	0.528	0.569	0.682	0.599	0.633
↪+*Pre*	0.603	0.517	0.566	0.721	0.606	0.686
MoPe	0.536	0.519	0.550	0.586	0.514	0.558
↪+*Pre*	0.552	0.546	0.592	0.674	0.643	0.650
GradNorm$_x$	0.581	0.540	0.555	0.630	0.547	0.602
↪+*Pre*	0.566	0.539	0.541	0.606	0.576	0.599

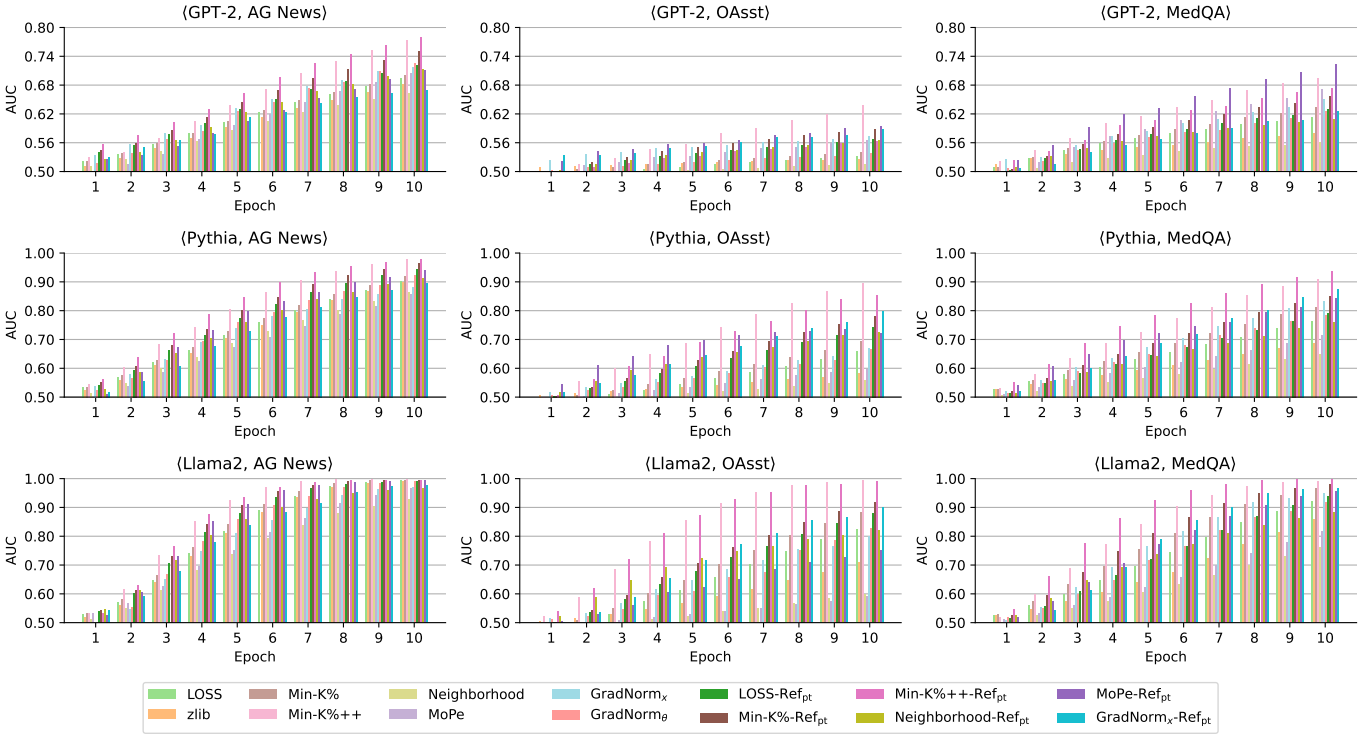

Fig. 6: The AUC of various MIAs when fine-tuning the pre-trained models with different epochs.

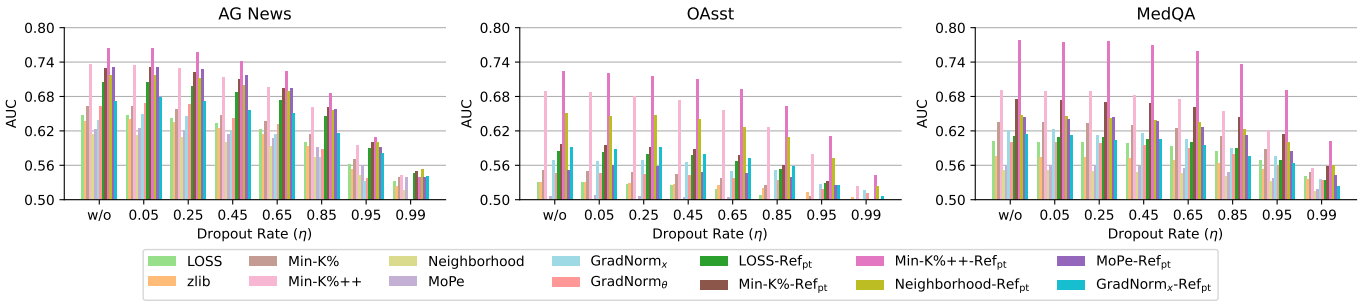

Fig. 7: The AUC of various MIAs against the fine-tuned Llama-2 with varying dropout rates (η).

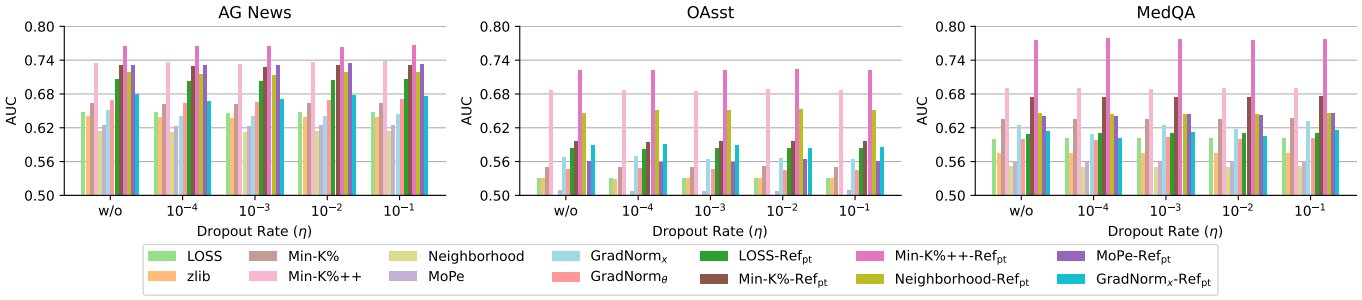

Fig. 8: The AUC of various MIAs against the fine-tuned Llama-2 with varying weight decay rates (α).

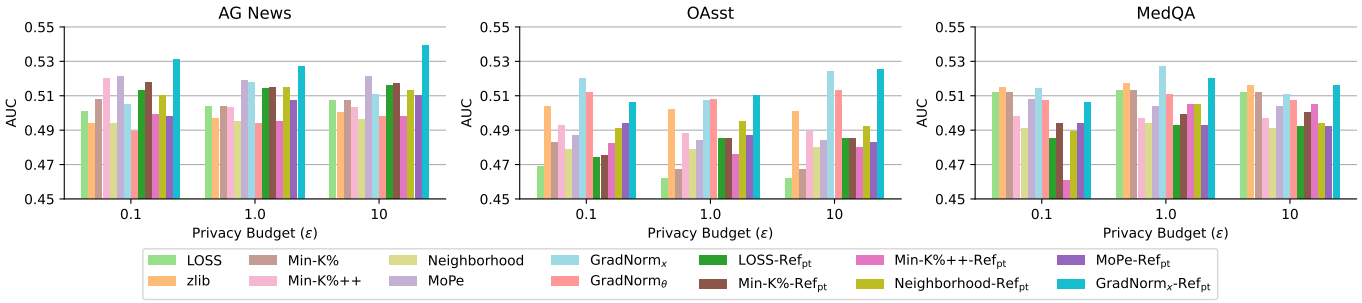

Fig. 9: The AUC of various MIAs against the fine-tuned Llama-2 with varying privacy budgets (ϵ).

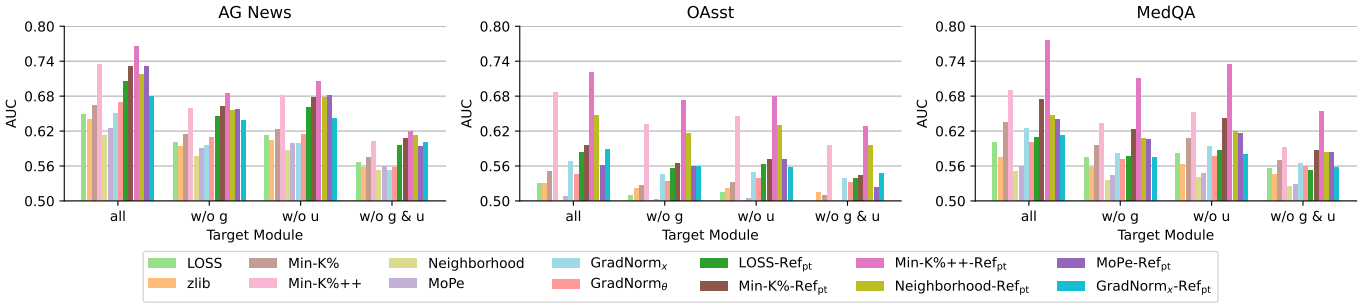

Fig. 10: The AUC of various MIAs against the fine-tuned Llama-2 with and without the *up* and *gate* layers.

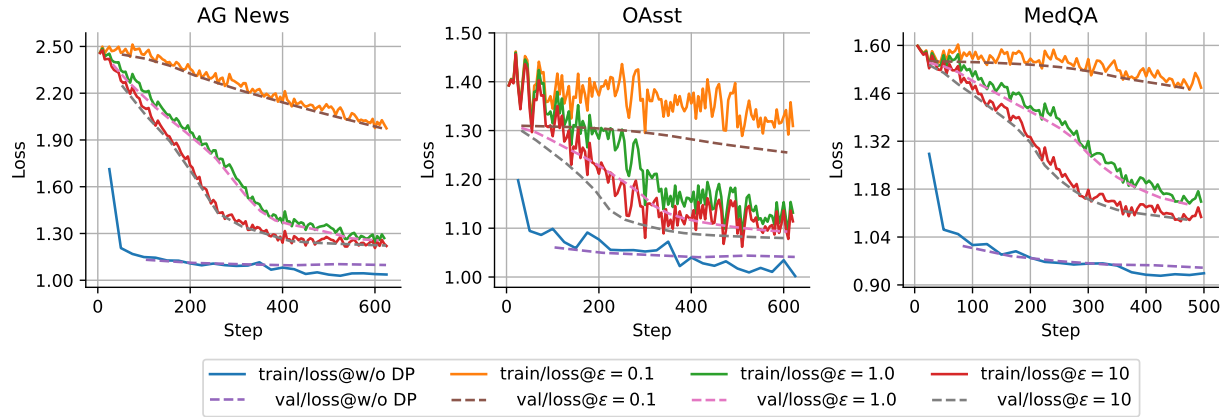

Fig. 11: The training and validation loss during Llama-2 LoRA fine-tuning across three datasets when applying differential privacy with varying privacy budgets ϵ.