On Membership Inference Attacks in Knowledge Distillation

Ziyao Cui[*] **Minxing Zhang**[*] **Jian Pei**
Department of Computer Science
Duke University
Durham, NC 27705
{richard.cui, minxing.zhang, j.pei}@duke.edu

Abstract

Nowadays, Large Language Models (LLMs) are trained on huge datasets, some including sensitive information. This poses a serious privacy concern because privacy attacks such as Membership Inference Attacks (MIAs) may detect this sensitive information. While knowledge distillation compresses LLMs into efficient, smaller student models, its impact on privacy remains underexplored. In this paper, we investigate how knowledge distillation affects model robustness against MIA. We focus on two questions. First, how is private data protected in teacher and student models? Second, how can we strengthen privacy preservation against MIAs in knowledge distillation? Through comprehensive experiments, we show that while teacher and student models achieve similar overall MIA accuracy, teacher models better protect member data, the primary target of MIA, whereas student models better protect non-member data. To address this vulnerability in student models, we propose 5 privacy-preserving distillation methods and demonstrate that they successfully reduce student models' vulnerability to MIA, with ensembling further stabilizing the robustness, offering a reliable approach for distilling more secure and efficient student models. Our implementation source code is available at `https://github.com/richardcui18/MIA_in_KD`.

1 Introduction

Large Language Models (LLMs) have achieved significant success. The vast amount of pretraining data used to develop these models is one of the most important reasons for this impressive breakthrough. However, while the amount of pretraining data used to train LLMs has quickly expanded, it has also raised numerous privacy concerns [37, 40]. Given the huge volume of pretraining data, it is impossible to filter out sensitive information, such as copyrighted materials [10, 21] and personally identifiable information [22, 33]. As a result, LLMs may inadvertently memorize private content, and it is possible for attackers to infer this sensitive information from the pretraining dataset, using methods such as Membership Inference Attack (MIA).

There have been numerous efforts investigating how to effectively detect pretraining data in LLMs using MIA [4, 7, 40, 42]. However, the privacy implications of LLMs typically focus on individual models in isolation. Meanwhile, in order to reduce the size and the number of arithmetic operations of LLMs, model compression techniques like knowledge distillation have been proposed and applied to many state-of-the-art LLMs including Llama, Gemma, and BERT [18, 19, 26, 31, 32, 35]. Nevertheless, despite their wide adoption, the privacy implications for these compressed (student) models, in particular how they compare to their original counterparts (teacher models), remain underexplored.

[*] Both authors contributed equally to this research.

Preprint. Under review.

In this work, we investigate how knowledge distillation affects model robustness, as measured using MIA accuracy. By gathering 12 different pairs of teacher and student models and 3 different MIA methods, we found that, although the overall MIA accuracies between teacher and student models are similar, teacher models tend to protect pretraining data points better than student models, while student models are better at identifying which data points are not in pretraining datasets. As the ability to identify pretraining data points is more valuable than identifying those that are not [4], we conclude that teacher models protect pretraining data better and thus are more robust to MIA.

Building on this observation, to strengthen privacy preservation in knowledge distillation and make student models more robust to MIA, we develop 5 privacy-preserving distillation methods, including 3 targeting the distillation process and 2 focusing on the post-distillation process. We also propose an ensemble of privacy-preserving distillation methods to reduce the variance of privacy-preserving distillation performance. Our comprehensive experiments show that all of our privacy-preserving distillation methods, including ensemble, lead to more robust student models.

To the best of our knowledge, we are the first to propose privacy-preserving distillation methods to mitigate MIA vulnerabilities in knowledge distillation. We are also the first to conduct an extensive comparison of model vulnerability to MIAs in knowledge distillation across multiple teacher-student model pairs. While Jagannatha et al. [17] explored this with a single BERT-based teacher-student pair focusing on medical datasets, our study evaluates 12 different model pairs.

The remainder of this paper is organized as follows. The problem is formally defined in Section 2. A discussion of related work is given in Section 3. In Section 4, the methodology for comparing MIA accuracy between student and teacher models and the privacy-preserving distillation methods are presented. In Section 5, experimental results and analysis are presented to compare teacher and student models and to show the optimality of the proposed privacy-preserving distillation methods. Conclusion and future work are finally discussed in Section 6.

2 Problem Definition

Consider a LLM $\mathcal{N}$ and a dataset $\mathcal{D}$ used to train $\mathcal{N}$. A **membership inference attack (MIA) method** M takes a target data point d as input and aims to determine whether $d \in \mathcal{D}$. For simplicity, denote by $M(\mathcal{N}, d)$ the attacker's prediction, where $M(\mathcal{N}, d) = 1$ if the MIA method M predicts that d belongs to the training dataset of $\mathcal{N}$ and 0 otherwise. In practice, M may compute a confidence score $M'(\mathcal{N}, d)$ and use a threshold τ to make the final prediction of the membership of d, that is, $M_\tau(\mathcal{N}, d) = \mathbb{1}[M'(\mathcal{N}, d) > \tau]$, where $\mathbb{1}$ is the indicator function.

In knowledge distillation, a teacher model $\mathcal{T}$ is used as a guiding framework to transfer knowledge to a student model $\mathcal{S}$, where the student model $\mathcal{S}$ is learned to mimic the performance of the teacher $\mathcal{T}$ [41]. Let $\mathcal{D}_t$ and $\mathcal{D}_s$ be the training dataset for $\mathcal{T}$ and $\mathcal{S}$, respectively. Given a data point $d \in \mathcal{D}_t$, since the teacher model $\mathcal{T}$ is trained using d and knowledge from $\mathcal{T}$ is transferred to $\mathcal{S}$ during knowledge distillation, we consider d to also be used for training $\mathcal{S}$, thus $d \in \mathcal{D}_s$. Therefore, $\mathcal{D}_t \subseteq \mathcal{D}_s$. It follows that $\mathcal{D}_t$ is the training dataset for both $\mathcal{T}$ and $\mathcal{S}$, and we denote $\mathcal{D}_t = \mathcal{D}$ for simplicity.

Moreover, let $\mathcal{D}'$ be a dataset that was not used to train $\mathcal{T}$ or $\mathcal{S}$. $\mathcal{D}'$ can be obtained using several methods, such as selecting a dataset that was released after $\mathcal{T}$ and $\mathcal{S}$.

We define the accuracy of a MIA method M with respect to threshold τ on a teacher model $\mathcal{T}$ as

$$A(\mathcal{T}, M_\tau, \mathcal{D}, \mathcal{D}') = \frac{1}{2} \cdot \Big[\frac{\sum_{x \in \mathcal{D}} \mathbb{1}[M_\tau(\mathcal{T}, x) = 1]}{|\mathcal{D}|} + \frac{\sum_{y \in \mathcal{D}'} \mathbb{1}[M_\tau(\mathcal{T}, y) = 0]}{|\mathcal{D}'|} \Big] \tag{1}$$

where $\mathcal{D}$ and $\mathcal{D}'$ denote the training and non-training datasets for the teacher model $\mathcal{T}$, respectively, $|\cdot|$ denotes the size of the dataset, and M_τ denotes the MIA method with respect to threshold τ that is used to attack $\mathcal{T}$.

Similarly, define the accuracy of a MIA method M with respect to threshold τ on a student model $\mathcal{S}$ as

$$A(\mathcal{S}, M_\tau, \mathcal{D}, \mathcal{D}') = \frac{1}{2} \cdot \Big[\frac{\sum_{x \in \mathcal{D}} \mathbb{1}[M_\tau(\mathcal{S}, x) = 1]}{|\mathcal{D}|} + \frac{\sum_{y \in \mathcal{D}'} \mathbb{1}[M_\tau(\mathcal{S}, y) = 0]}{|\mathcal{D}'|} \Big] \tag{2}$$

where $\mathcal{D}$ and $\mathcal{D}'$ are the same training and non-training datasets in Equation 1, and M_τ denotes the MIA method with respect to threshold τ that is used to attack the student model $\mathcal{S}$.

The core problem investigated in this paper is whether student models are more vulnerable to MIA attacks than teacher models, that is, $A(\mathcal{T}, M_{\mathcal{T}}, \mathcal{D}, \mathcal{D}') < A(\mathcal{S}, M_{\mathcal{T}}, \mathcal{D}, \mathcal{D}')$, and how teacher models can inject privacy knowledge during distillation to enhance student robustness against MIA.

3 Related Work

3.1 MIA Methods

Initially proposed by Shokri et al. [28], MIA aims to detect whether a given data point belongs to a model's training dataset. While MIA has been investigated in various domains, such as diffusion models [5] and multi-layer perceptrons [39], MIA in LLMs presents unique challenges. For example, LLMs often lack publicly-available training data [1, 14, 36], making it difficult to validate MIA performance due to the absence of ground-truth membership labels. Moreover, many modern LLMs use single-epoch training frameworks on massive datasets, which makes memorization, a key factor in traditional MIA, difficult [6, 27]. Recently, MIA methods in LLM have attracted much interest. Yeom et al. [42] proposed a loss-based approach for attacking member data, observing that models generally have a lower loss for member data than non-member data. Carlini et al. [7] improved this approach by normalizing the loss by zlib compression size. A reference-model-based attack method was developed by Carlini et al. [4], where two sets of shadow models are trained using datasets with and without the target sample, and the prediction is made through a likelihood ratio test between the two sets of shadow models. More recently, Xie et al. [40] proposed ReCaLL, which compares the relative conditional log-likelihood to calculate the confidence score for membership classification.

However, the existing work primarily focused on applying MIA to one model in isolation, without considering how MIA behaves differently across interconnected models, such as in a knowledge distillation setting where teacher and student models share a learned relationship. For example, while Yeom et al. [42] proposed loss-based attacks and Carlini et al. [7] introduced zlib-normalized loss, these methods were not designed to compare privacy trade-offs in knowledge distillation. Our work bridges this gap by systematically quantifying how knowledge distillation affects models' vulnerability to MIA and leveraging these insights to develop privacy-preserving distillation techniques.

3.2 Knowledge Distillation

Knowledge distillation [3, 16] is a model compression technique that leverages a larger teacher model as a guiding framework to train a smaller student model. During the distillation process, the student model is trained with a combined objective of minimizing the distillation loss, a loss function reflecting the differences between teacher and student models, and the student model's actual loss, reflecting the difference between student model output and ground truth [13, 41]. There has been much work applying knowledge distillation to LLMs. Sanh et al. [26] proposed DistilBERT by applying knowledge distillation to BERT in the pre-training and optional fine-tuning stages, where a linear combination of distillation loss and supervised training loss is used to train the student model. Similarly, Song et al. [29], Liang et al. [20], and Zhang et al. [43] also proposed different knowledge distillation methods focusing on mimicking the teacher model's output distribution. Another common approach to knowledge distillation leverages the finding that intermediate representations learned by the teacher models could be used as helpful hints to train and achieve a better final performance of the student model [25]. Sun et al. [30] and Jiao et al. [18] applied this idea and proposed a knowledge distillation framework using a layer-wise distillation strategy, where the student model mimics the teacher model's hidden state behaviors on multiple intermediate layers.

While previous studies primarily focus on maintaining or improving model performance during knowledge distillation, we examine knowledge distillation from a privacy perspective. More specifically, we investigate how the distillation process affects the vulnerability of teacher and student models to MIA. Instead of focusing exclusively on how the teacher model can guide the student models to better performance, we leverage the teacher model's attack vulnerability as valuable privacy information to be transferred to student models during the distillation process. As knowledge distillation becomes increasingly adopted in LLMs, shifting the focus from performance-only distillation to privacy-aware distillation is crucial to ensure responsible LLM development and usage.

3.3 Relating MIA With Knowledge Distillation

There are very few studies investigating MIA in a knowledge distillation setting. To the best of our knowledge, only Jagannatha et al. [17] provided empirical evidence that BERT has higher privacy leakage than DistilBERT. However, the study [17] is limited in scope as it only compares one pair of teacher-student models and the dataset is limited to the US hospital datasets and electronic health records. We hope to expand the scope and include more pairs of teacher and student models in this work to better understand how knowledge distillation contributes to model robustness, as well as how models could become more robust under a knowledge distillation setting.

4 Methodology

In this section, we present our methodology for investigating the privacy implications of knowledge distillation under MIAs. In Section 4.1, we outline an approach to quantify, between teacher and student models, the change in model vulnerability to MIAs. Then, we develop 5 novel privacy-preserving distillation methods in Section 4.2 that target both the distillation process and post-distillation inference. Finally, we introduce an ensemble approach in Section 4.3 that combines these methods to achieve more robust and reliable privacy protection for the knowledge distillation process.

4.1 Comparing MIA Accuracy Between Student and Teacher Models

Our methodology of testing whether the student models are more vulnerable to MIA attacks than teacher models (i.e. $A(\mathcal{T}, M_\tau, \mathcal{D}, \mathcal{D}') < A(\mathcal{S}, M_\tau, \mathcal{D}, \mathcal{D}')$) works in three steps. First, we gather teacher models $\mathcal{T}$ with known training datasets $\mathcal{D}$ and open-source distilled student models $\mathcal{S}$. Based on the release dates of $\mathcal{T}$ and $\mathcal{S}$, we select a dataset $\mathcal{D}'$ that is released later than both $\mathcal{T}$ and $\mathcal{S}$ as the non-member dataset. Finally, we apply MIA methods to $\mathcal{T}$ and $\mathcal{S}$ on $\mathcal{D} \cup \mathcal{D}'$ and calculate $A(\mathcal{T}, M_\tau, \mathcal{D}, \mathcal{D}')$ and $A(\mathcal{S}, M_\tau, \mathcal{D}, \mathcal{D}')$.

In addition to the overall accuracy comparison, we also compute the standard deviation and 95% confidence intervals to decide whether the difference in accuracy is statistically significant. Specifically, given that the accuracy is an average of two independent binomial proportions, the standard deviation for the teacher model can be written as:

$$\begin{aligned} SD(\mathcal{T}, M_\tau, \mathcal{D}, \mathcal{D}') &= \sqrt{\frac{1}{4} \cdot \Big[\text{Var}(TPR) + \text{Var}(TNR)\Big]} \\ &= \frac{1}{2} \cdot \sqrt{\frac{TPR(1-TPR)}{|\mathcal{D}|} + \frac{TNR(1-TNR)}{|\mathcal{D}'|}} \end{aligned} \tag{3}$$

where $TPR = \frac{\sum_{x \in \mathcal{D}} \mathbb{1}[M_\tau(\mathcal{T},x)=1]}{|\mathcal{D}|}$ and $TNR = \frac{\sum_{y \in \mathcal{D}'} \mathbb{1}[M_\tau(\mathcal{T},y)=0]}{|\mathcal{D}'|}$ are the two components of the accuracy definition.

It follows that the 95% confidence interval for the teacher model is:

$$CI_{95\%}(\mathcal{T}, M_\tau, \mathcal{D}, \mathcal{D}') = A(\mathcal{T}, M_\tau, \mathcal{D}, \mathcal{D}') \pm 1.96 \times SD(\mathcal{T}, M_\tau, \mathcal{D}, \mathcal{D}') \tag{4}$$

The standard deviation and confidence interval expressions for student models are in a similar form.

We further compare the difference in vulnerability between member and non-member data across teacher and student models, which allows us to determine whether teacher or student models demonstrate consistent advantages in protecting member data, the primary target for MIAs. We also aim to understand how knowledge distillation affects different aspects of model vulnerability to MIAs by comparing results across different model architectures and MIA methods. The details on the 12 different pairs of teacher and student models considered in the experiment, along with the training and non-training datasets, will be discussed in Section 5.1.1. We will also outline our approach for balancing the sizes of $\mathcal{D}$ and $\mathcal{D}'$ to ensure fair classification evaluation.

4.2 Privacy-Preserving Distillation Methods

We now explore ways to enhance the robustness of student models against MIA. Our approach addresses the problem from two perspectives. First, we propose 3 in-distillation methods to strengthen robustness during the distillation process. Then, we introduce 2 post-distillation methods to further improve resistance to MIA after distillation is complete.

4.2.1 In-Distillation Approaches

Bottleneck Carlini et al. [7] showed that larger models are prone to memorizing the training data, thus making them more vulnerable to MIAs. Therefore, one possible approach to make models more robust against MIA is to directly change the model architecture so that the model becomes smaller. One common approach to do so is through adding factorized embedding parameterization, or "bottleneck" architecture [19, 31], into the feed-forward network. Specifically, rather than directly projecting from the hidden dimension H to the standard intermediate size I (typically $4H$), we first project the representation downwards into a lower-dimensional bottleneck space with dimensionality B (where $B \ll H$) before expanding to I. By doing so, the number of parameters needed changes from $O(H \cdot I)$ to $O(H \cdot B + B \cdot I)$, which is a significant reduction if $H \gg B$. For example, in a typical BERT architecture, where $H = 768$ and $I = 3072$, implementing a bottleneck dimensionality of $B = 192$ would give a four-time reduction in projection layer parameters needed. This change in model architecture not only improves computational efficiency but also limits the model's ability to memorize details of training samples, thus making the model less vulnerable to MIAs.

Replacing Layer Normalization Sun et al. [31] showed that replacing layer normalization can simplify the model and reduce model latency. Moreover, the mean and variance calculations in layer normalization may inadvertently memorize training data. To address this potential privacy vulnerability, we replace the layer normalization with a simpler, element-wise linear transformation:

$$\texttt{NoNorm}(\mathbf{h}) = \gamma \circ \mathbf{h} + \beta$$

where $\gamma, \beta \in \mathbb{R}^n$, n is the number of channels, $\mathbf{h}$ is the hidden state vector, and $\circ$ is the Hadamard product. There are several advantages to this modification of the normalization function. First, the `NoNorm` function does not calculate the mean and variance of the inputs, which eliminates the possibility of leaking privacy information of the training data that could be exploited in MIAs. Second, as demonstrated by Sun et al. [31], replacing the layer normalization with `NoNorm` would optimize the inference latency of LLMs. This optimization suggests a simpler model that is less likely to be prone to memorization, thus becoming theoretically less vulnerable to MIAs.

Replacing GELU Activation With ReLU Previous research suggested that, in addition to replacing layer normalization, replacing GELU activation [15] with the simpler ReLU activation function [23] can also make the model simpler and reduce model latency [31]. Therefore, we replace GELU activation with ReLU as another privacy-preserving distillation method.

4.2.2 Post-distillation Approaches

Red-List Sampling Besides changes to student model architecture, we can also enhance student robustness by injecting privacy knowledge from the teacher model. More specifically, by applying MIA on the teacher model, we split the training dataset $\mathcal{D}$ into two subsets: $\mathcal{D}_v$, the "vulnerable" subset including all the data points in $\mathcal{D}$ that are successfully attacked by the MIA, and $\mathcal{D}_{nv}$, the "non-vulnerable" subset including those data points where MIA failed. That is, $\mathcal{D}_v = \{M(\mathcal{T}, x) = 1 | x \in \mathcal{D}\}$ and $\mathcal{D}_{nv} = \{M(\mathcal{T}, x) = 0 | x \in \mathcal{D}\}$ ($\mathcal{D}_v \cup \mathcal{D}_{nv} = \mathcal{D}$ and $\mathcal{D}_v \cap \mathcal{D}_{nv} = \emptyset$).

The intuition is that these two subsets should encode privacy knowledge of the teacher model, so if we leverage this privacy knowledge during distillation, the student model might be able to learn privacy information from the teacher model. Our idea here is that, the tokens in the vulnerable subset $\mathcal{D}_v$, which includes all the data points that are being successfully attacked in the teacher model, should be suppressed in the student model so that the student model generates those easy-to-attack tokens less often, thus making it more robust.

Formally, the original softmax probability is

$$p_i = \frac{\exp(\mathbf{z}_i)}{\sum_{j=1}^{|V|} \exp(\mathbf{z}_j)}$$

where $\mathbf{z}_i$ be the student model's original logit for token i, and V is the full vocabulary.

Let V_{vul} denote the set of vulnerable tokens from the vulnerable subset $\mathcal{D}_v$. If token i belongs to V_{vul}, then we penalize its logit and probability by a penalization factor δ so that the updated probability is

$$p_i^{red} = \begin{cases} \frac{\exp(\mathbf{z}_i - \delta)}{\sum_{j=1}^{|V|} \exp(\mathbf{z}_j - \delta \cdot \mathbb{1}[j \in V_{vul}])} & \text{if } i \in V_{vul} \\ \frac{\exp(\mathbf{z}_i)}{\sum_{j=1}^{|V|} \exp(\mathbf{z}_j - \delta \cdot \mathbb{1}[j \in V_{vul}])} & \text{if } i \notin V_{vul} \end{cases}$$

Temperature Scaling We can also increase the temperature for the vulnerable tokens so that their probability distribution is more spread out, making the generation more random on those tokens. Effectively, this penalizes the vulnerable tokens by suppressing their logits. Specifically, the updated logit in this case becomes

$$\mathbf{z}_i^{temp} = \begin{cases} \frac{\mathbf{z}_i}{T} & \text{if } i \in V_{vul} \\ \mathbf{z}_i & \text{if } i \notin V_{vul} \end{cases}$$

where T is the temperature hyperparameter. Thus the updated probability is

$$p_i^{temp} = \frac{\exp(\mathbf{z}_i^{temp})}{\sum_{j=1}^{|V|} \exp(\mathbf{z}_j^{temp})}$$

4.3 Ensembling Privacy-Preserving Distillation Methods

Different privacy-preserving distillation methods may result in different levels of robustness against MIA, and there is no one single method that can guarantee to offer the best protection across all models against all MIA methods. Therefore, we explore ensemble as a way to aggregate the various privacy-preserving distillation methods and reduce the variance and instability that individual methods may have. Moreover, using only one privacy-preserving distillation method may be more easily attacked by attackers who exploit a single vulnerability. Ensembling multiple methods together makes it harder for attackers to specialize attacks targeting one particular type of defense.

Let $P = \{P_1, P_2, \ldots, P_k\}$ be a set of k privacy-preserving distillation methods. Given a teacher model $\mathcal{T}$ and a student model $\mathcal{S}$, denote by $\mathcal{S}_{P_i}$ the student model distilled using the i-th privacy-preserving distillation method. Given a data point d, for each privacy-preserving distillation method P_i and the associated student model $\mathcal{S}_{P_i}$, we can obtain an MIA predicted label $M_\tau(\mathcal{S}_{P_i}, d)$ with threshold τ. Let $\mathcal{O}(\mathcal{S}, M_\tau, d, P)$ denote the number of privacy-preserving distillation methods in P applied on the student model $\mathcal{S}$ such that their MIA predicted label $M_\tau(\mathcal{S}_{P_i}, d) = 1$. Then,

$$\mathcal{O}(\mathcal{S}, M_\tau, d, P) = \sum_{i=1}^{k} \mathbb{1}[M_\tau(\mathcal{S}_{P_i}, d) = 1]$$

Similarly, define $\mathcal{Z}(\mathcal{S}, M_\tau, d, P)$ to be the number of privacy-preserving distillation methods in P applied on the student model $\mathcal{S}$ such that their MIA predicted label $M_\tau(\mathcal{S}_{P_i}, d) = 0$

$$\mathcal{Z}(\mathcal{S}, M_\tau, d, P) = \sum_{i=1}^{k} \mathbb{1}[M_\tau(\mathcal{S}_{P_i}, d) = 0]$$

Finally, let $M_{\text{ensemble}}(\mathcal{S}, M_\tau, d, P)$ be the ensembled MIA predicted label of data point d over the privacy-preserving distillation methods P applied on the student model $\mathcal{S}$, that is,

$$M_{\text{ensemble}}(\mathcal{S}, M_\tau, d, P) = \mathbb{1}[\mathcal{O}(\mathcal{S}, M_\tau, d, P) > \mathcal{Z}(\mathcal{S}, M_\tau, d, P)]$$

Similarly, the MIA accuracy in this case is defined as:

$$\begin{aligned} & A_{\text{ensemble}}(\mathcal{S}, M_\tau, \mathcal{D}, \mathcal{D}', P) \\ = & \frac{1}{2} \cdot \left[\frac{\sum_{x \in \mathcal{D}} \mathbb{1}[M_{\text{ensemble}}(\mathcal{S}, M_\tau, x, P) = 1]}{|\mathcal{D}|} + \frac{\sum_{y \in \mathcal{D}'} \mathbb{1}[M_{\text{ensemble}}(\mathcal{S}, M_\tau, y, P) = 0]}{|\mathcal{D}'|} \right] \end{aligned}$$

5 Experimental Results

In this section, we conduct comprehensive experiments exploring the effect of knowledge distillation on model vulnerability to MIA and the effectiveness of the privacy-preserving distillation methods.

5.1 Experimental Setup

5.1.1 Comparing MIA Accuracy Between Teacher and Student Models

Models and Datasets For this experiment, we choose 5 sets of models with known training datasets and open-source distilled models available: **BERT** [9], with the bookcorpus dataset [44] as the training dataset and DistilBERT [26], TinyBERT [18], ALBERT [19], and MobileBERT [31] as the student models; **GPT2** [24], with the WebText dataset [24] as the training dataset and DistilGPT2 [26] as the student model; **Llama 360M** [36], with the ArXiv dataset that was a part of the Pile dataset [12] as the training dataset and BabyLlama [35] as the student model; **Llama 3.1 8B** [11], with the ArXiv dataset that was a part of the Pile dataset [12] as the training dataset and Llama 3.2 3B [11], Llama 3.2 1B [11], and MambaInLlama [38] as the stduent models; **Gemma 2 27B** [34], with the WikiMIA dataset [27] as the training dataset and Gemma 2 9B [34], Gemma 2 2B [34], and Gemma 2 2B distilled [32] as the student models.

For non-member data points used in testing, we use the built-in non-member data points as a part of the ArXiv and WikiMIA datasets for Llama 360M, Llama 3.1 8B, and Gemma 2 27B models. For BERT and GPT2, we used the WebInstructSub-prometheus dataset [8] since it was published in May 2024, which is later than the release date of both BERT and GPT2, thus it must not be used for training BERT and GPT2. To ensure balanced classification testing, we truncate both member and non-member datasets to the size of the smaller dataset, as existing works did [40].

MIA Methods We use 3 MIA methods to attack the models described above: **ReCaLL** [40], which considers the change in conditional log-likelihood when prefixing the target input texts with non-member context to classify member and non-member data; **Loss** [42], which uses input loss as the confidence score to classify member and non-member data; **Zlib** [7], which normalizes input loss with zlib entropy as the confidence score. Since ReCaLL is only applicable to autoregressive language models, we will only apply methods Loss and Zlib to all BERT-based models.

Implementation Details As explained in Section 2, each MIA method gives a confidence score for each target data point, and a decision threshold is needed for classification. As this is not the main focus of this paper, we choose thresholds randomly for each MIA method with the guarantee that at least $t\%$ of the data points is in each of the prediction classes (member and non-member) for both the teacher and student models, where t is a pre-specified parameter that is fixed for all models and MIA methods in the experiment. In our study, t is chosen as 20.

For computational efficiency, we perform MIA on each model on a subset containing 100,000 data points from the training dataset only. For those training datasets with less than 100,000 data points, we use the full dataset. We perform an ablation study in Appendix A to show that MIA accuracy is generally stable when using test subsets of different sizes. With an A5000 GPU, testing on a subset of 100,000 data points takes about 30 minutes per model.

5.1.2 Privacy-Preserving Distillation Methods

Models and Datasets For this experiment, we choose 4 teacher-student model pairs where the code for knowledge distillation is publicly available: **BERT** [9] trained on the bookcorpus dataset [44] with DistilBERT [26], TinyBERT [18], and SpikingBERT [2] as the student models; **GPT2** [24] trained on the WebText dataset [24] with DistilGPT2 [26] as the student model.

Implementation Details For each teacher-student model pair, we distill new student models using the approaches outlined in Section 4.2 and perform the 3 MIA methods described in Section 5.1.1 to each new student model. Note that ReCaLL is not applicable to BERT.

Similar to the reasons described in Section 5.1.1, we distill student models and perform MIA on a subset containing 100,000 data points. For consistency, we distill the available student models (DistilBERT, TinyBERT, SpikingBERT, and DistilGPT2) from scratch using the 100,000 data points.

Table 1: Comparison of MIA accuracy between teacher and student models. Each cell shows teacher model / student model accuracy. **Bolded** entries denote the lower accuracy between each teacher-student model pair for each MIA method; Underlined entries denote that the difference is statistically significant.

Teacher Model	Student Model	ReCaLL	Loss	Zlib
BERT	DistilBERT	–	0.103 / **0.059**	**0.120** / 0.169
	TinyBERT	–	**0.103** / 0.696	0.120 / **0.091**
	ALBERT	–	**0.103** / 0.385	**0.120** / 0.165
	MobileBERT	–	0.103 / **0.101**	**0.120** / 0.189
GPT2	DistilGPT2	0.719 / **0.571**	**0.264** / 0.421	**0.832** / 0.859
Llama 360M	BabyLlama	**0.491** / 0.515	**0.494** / 0.496	0.507 / **0.503**
Llama 3.1 8B	Llama 3.2 3B	**0.490** / 0.491	**0.499** / 0.518	**0.510** / 0.512
	Llama 3.2 1B	**0.490** / 0.513	**0.499** / 0.514	**0.510** / 0.514
	MambaInLlama	**0.490** / 0.521	**0.499** / 0.511	**0.510** / 0.511
Gemma 2 27B	Gemma 2 9B	0.606 / 0.606	0.541 / **0.509**	0.569 / **0.541**
	Gemma 2 2B	**0.606** / 0.624	0.541 / **0.528**	0.569 / **0.541**
	Gemma 2 2B Distilled	0.606 / **0.569**	**0.541** / 0.550	0.569 / **0.555**

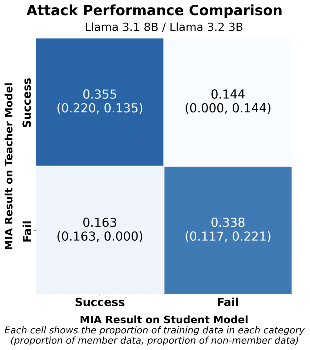

Figure 1: Attack performance for Llama 3.1 8B as teacher model and Llama 3.2 3B as student model, under the Loss MIA method.

5.2 Results and Analysis

5.2.1 Teacher-Student Model Comparison

Table 1 shows MIA accuracy comparison between teacher and student models. For the difference to be considered statistically significant, the confidence intervals of the teacher and student models should not overlap. The complete results table with confidence intervals are shown in Table 6 in Appendix B. In general, we find no consistent pattern of MIA vulnerability between teachers and students, with most differences in MIA accuracy not statistically significant. However, in the 10 statistically significant cases, teacher models show lower MIA accuracy in 7 instances, suggesting they may be more robust than student models.

The difference between teacher and student models under MIA becomes clearer when we look at the distribution of attack results. Figure 1 shows the attack performance matrix for one model pair: Llama 3.1 8B (teacher) and Llama 3.2 3B (student). A "successful" MIA result means the predicted label (member or non-member) matches the true label; a "fail" means it does not. Each cell includes the percentage of test data it represents, with parentheses showing the breakdown between member and non-member data. For example, the top-left cell shows that 35.5% of test data was successfully attacked by both models – 22.0% of it was member data and 13.5% was non-member data.

Figure 1 shows all 14.4% of test data successfully attacked on the teacher model but not the student model was non-member data, while all 16.3% of test data successfully attacked on the student model but not the teacher model was member data. Despite similar overall MIA accuracy, the teacher model better protects member data, while the student model better protects non-member data. As Carlini et al. [4] emphasizes, reliably attacking member data is more critical in MIA than high overall accuracy. Thus, the teacher model offers stronger privacy protection for sensitive member data, making it more robust than the student model. Similar trends appear in other model pairs with statistically insignificant differences.

5.2.2 Privacy-Preserving Distillation Methods

Table 2 shows the MIA accuracy for each privacy-preserving distillation method applied to 4 different student models. In almost all cases, the student model variant with the lowest MIA accuracy is one of the proposed methods and not the original model. The only cases where the original models achieve the lowest MIA accuracy are TinyBERT and SpikingBERT attacked with Zlib, in which cases the accuracies are very similar across privacy-preserving distillation methods. For TinyBERT attacked with Zlib, the best MIA accuracy is achieved by three methods; for SpikingBERT attacked with Zlib, the original model is only 0.3% better than the bottleneck method, the second best-performing privacy-preserving distillation method in this case. This shows that our privacy-preserving distillation methods can indeed lower MIA accuracy, leading to more privacy-preserved student models.

Table 2: Summary of privacy-preserving distillation results. Entries with an asterisk* and **bolded** entries denote the highest and lowest accuracy, respectively, for each student model and MIA method pair across all privacy-preserving distillation methods. "None" represents the original student model.

Teacher Model	Student Model	MIA Method	**Privacy-Preserving Distillation Method**						
			None	ReLU	Norm	Bottleneck	Red List	Temp.	Ensemble
BERT	DistilBERT	Loss	0.990*	0.967	0.877	**0.674**	0.989	0.987	0.986
		Zlib	0.370	0.376*	**0.065**	0.130	0.124	0.197	0.108
	TinyBERT	Loss	0.462	0.462	0.474	0.589*	**0.329**	0.367	0.346
		Zlib	**0.101**	**0.101**	0.106	0.102	0.112*	**0.101**	0.103
	SpikingBERT	Loss	0.423	0.398	0.507*	0.407	0.298	0.312	**0.279**
		Zlib	**0.095**	0.100	0.100	0.098	0.112*	0.101	0.101
GPT2	DistilGPT2	ReCaLL	0.571	**0.417**	0.435	0.681*	0.455	0.455	0.519
		Loss	0.509*	0.326	0.443	0.418	0.289	**0.199**	0.242
		Zlib	0.859	**0.858**	0.900*	0.895	0.884	0.887	0.889

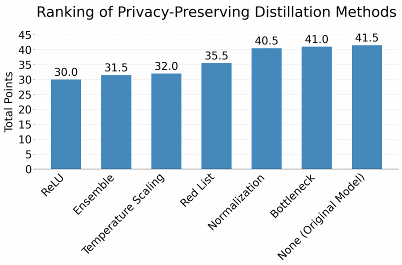

Figure 2: Ranking of privacy-preserving distillation methods with aggregated points. Method ReLU performs the best out of all methods, and the original student model performs the worst.

Table 3: Updated privacy-preserving distillation results with ensemble of only 3 methods: ReLU, temperature scaling, and Red List. **Bolded** entries indicate the privacy-preserving distillation method with the lowest MIA accuracy for each student model and MIA method pair across all privacy-preserving distillation methods.

Teacher Model	Student Model	Privacy-Preserving Distillation Method	**MIA Accuracy**		
			ReCaLL	Loss	Zlib
BERT	DistilBERT	None (Original Model)	–	0.990	0.370
		Ensemble (5 methods)	–	**0.986**	**0.108**
		Ensemble (3 methods)	–	0.990	0.132
	TinyBERT	None (Original Model)	–	0.462	**0.101**
		Ensemble (5 methods)	–	0.346	0.103
		Ensemble (3 methods)	–	**0.326**	0.102
	SpikingBERT	None (Original Model)	–	0.423	**0.095**
		Ensemble (5 methods)	–	**0.279**	0.101
		Ensemble (3 methods)	–	**0.279**	0.101
GPT2	DistilGPT2	None (Original Model)	0.571	0.509	**0.859**
		Ensemble (5 methods)	0.519	0.242	0.889
		Ensemble (3 methods)	**0.482**	**0.224**	0.882

5.2.3 Ranking Privacy-Preserving Distillation Methods

To directly analyze the advantage of each privacy-preserving distillation method, we employ a ranking analysis. For each student model and MIA method pair, we assign 1 point to the privacy-preserving distillation method that gives the lowest MIA accuracy, 2 points to the method that gives the second-lowest MIA accuracy, and so on. When multiple methods achieve identical accuracy, they share the average of the consecutive point values they would have collectively occupied in the ranking order.

As illustrated in Figure 2, ReLU performs the best out of all the privacy-preserving distillation methods, obtaining 6 points lower than the average score. Ensemble gives the second-best performance, which is 4.5 points lower than the average score. This suggests ensembling creates a more robust student model through majority voting, reducing variance across privacy-preserving distillation methods by mitigating vulnerabilities where individual methods may be susceptible to MIA on different data points, models, or MIA attacks. The original model performs the worst, which suggests that all privacy-preserving distillation methods indeed could lower MIA accuracy.

Moreover, since bottleneck and normalization perform worse, we update the ensemble to include only temperature scaling, red list sampling, and ReLU. As indicated in Table 3, ensembling with only the 3 best privacy-preserving distillation methods gives an average MIA accuracy of 0.391, which is 0.006 lower than ensembling with 5 models and 0.096 lower than the original model.

Table 4: MIA accuracy for student models distilled using combinations of $\mathcal{D}_v$ and $\mathcal{D}_{nv}$ across three different test sets. Entries marked with an asterisk* and **bolded** entries indicate the highest and lowest MIA accuracy per attack method, respectively.

Student Model	Distillation Distribution	**Full** ($\mathcal{D}$)		**Vulnerable Only** ($\mathcal{D}_v$)		**Non-vulnerable Only** ($\mathcal{D}_{nv}$)	
		Loss	Zlib	Loss	Zlib	Loss	Zlib
DistilBERT	$\mathcal{D}$	0.678	**0.000**	**0.002**	**0.002**	0.686	**0.000**
	$\mathcal{D}_{nv}$	**0.202**	0.733	0.289*	0.852*	0.738*	0.621
	$\mathcal{D}_v$	0.666	0.682	0.180	0.748	**0.543**	0.711
	$\mathcal{D}_v + 5\%\mathcal{D}_{nv}$	0.819	0.813*	0.258	0.779	0.617	0.797*
	$\mathcal{D}_v + 10\%\mathcal{D}_{nv}$	0.826*	0.787	0.269	0.334	0.733	**0.000**
TinyBERT	$\mathcal{D}$	0.627*	**0.000**	**0.002**	**0.002**	0.662*	**0.000**
	$\mathcal{D}_{nv}$	**0.476**	0.502	0.662	0.653	0.230	0.212
	$\mathcal{D}_v$	0.505	0.497	0.688	0.646	**0.198**	0.334*
	$\mathcal{D}_v + 5\%\mathcal{D}_{nv}$	0.500	0.450	0.700	0.590	0.201	0.328
	$\mathcal{D}_v + 10\%\mathcal{D}_{nv}$	0.502	0.503*	0.726*	0.693*	0.201	0.255
SpikingBERT	$\mathcal{D}$	0.736	0.339*	0.500	0.444*	0.801	0.603
	$\mathcal{D}_{nv}$	0.748*	0.332	0.533*	0.434	0.898*	**0.467**
	$\mathcal{D}_v$	0.652	**0.257**	0.473	**0.337**	0.650	0.727
	$\mathcal{D}_v + 5\%\mathcal{D}_{nv}$	**0.643**	0.315	**0.467**	0.412	**0.645**	0.709
	$\mathcal{D}_v + 10\%\mathcal{D}_{nv}$	0.658	0.315	0.480	0.412	0.660	0.770*
DistilGPT2	$\mathcal{D}$	**0.221**	0.845*	0.292*	**0.000**	**0.230**	0.687*
	$\mathcal{D}_{nv}$	0.356*	0.752	**0.201**	0.587	0.416*	0.379
	$\mathcal{D}_v$	0.344	0.787	0.222	0.539	0.388	0.378
	$\mathcal{D}_v + 5\%\mathcal{D}_{nv}$	0.339	**0.321**	0.235	0.500	0.379	**0.369**
	$\mathcal{D}_v + 10\%\mathcal{D}_{nv}$	0.337	0.805	0.242	0.694*	0.378	0.395

5.2.4 Using Subsets for Distillation Training

In addition to the 5 proposed privacy-preserving distillation methods and the ensemble method, we consider another approach that leverages the teacher model's privacy knowledge during distillation, which is to use combinations of $\mathcal{D}_v$ and $\mathcal{D}_{nv}$ for distillation training. Specifically, we distill new student models using 5 different subsets of training data: full dataset ($\mathcal{D}$), the vulnerable subset $\mathcal{D}_v$, the non-vulnerable subset $\mathcal{D}_{nv}$, the vulnerable subset plus 5% of the non-vulnerable subset, and the vulnerable subset plus 10% of the non-vulnerable subset. To examine the effectiveness of this approach, we test each of the 5 models on the full dataset $\mathcal{D}$, the vulnerable set $\mathcal{D}_v$, and the non-vulnerable set $\mathcal{D}_{nv}$. Note that even when student models are distilled using only $\mathcal{D}_v$ or $\mathcal{D}_{nv}$, the full dataset $\mathcal{D}$ should be considered their training data since their teacher model is trained on $\mathcal{D} = \mathcal{D}_v \cup \mathcal{D}_{nv}$. Students are trained using both the teacher's knowledge and the explicit subset $\mathcal{D}/\mathcal{D}_v/\mathcal{D}_{nv}$.

The complete result is shown in Table 4. In general, we would expect models explicitly distilled using $\mathcal{D}_v$ or $\mathcal{D}_{nv}$ only to achieve higher MIA accuracy when tested on $\mathcal{D}_v$ or $\mathcal{D}_{nv}$, respectively. However, MIA accuracy across testing sets shows no clear pattern and largely depends on the student model. For DistilBERT and TinyBERT, distillation using the entire $\mathcal{D}$ yields the lowest MIA accuracy (0.228 and 0.216, respectively). For SpikingBERT, distillation using only $\mathcal{D}_v$ gives the lowest accuracy of 0.516, 3.12% below the average across all distillation distributions. For DistilGPT2, distillation using $\mathcal{D}_v + 5\%\mathcal{D}_{nv}$ achieves the lowest accuracy of 0.357, 6.34% below the overall average MIA accuracy.

6 Conclusion and Future Work

In this paper, we investigate the impact of knowledge distillation on privacy in LLMs against MIA. Analyzing 12 teacher–student model pairs, we find that teacher models better protect member data, making them more robust to MIA, while student models better protect non-member data. This suggests that distillation may unintentionally weaken privacy. To address this, we propose 5 privacy-preserving distillation methods – 3 during and 2 after distillation – and show through extensive experiments that they reduce MIA risk across various architectures. Ensembling these methods further improves consistency and robustness, supporting safer use of compressed LLMs in sensitive domains like healthcare and finance.

Future work should explore why teacher models better protect member data and how distillation alters memorization in LLMs. Understanding why our methods enhance student robustness may reveal key principles for designing privacy-aware distillation. Linking empirical results with theory will strengthen the foundation for secure model compression.

7 Ethical Statement

This paper investigates privacy risks in knowledge distillation and proposes methods to make models more robust to MIAs. We recognize that while our work aim to improve privacy protections in LLMs, the techniques could potentially be misused to identify model vulnerabilities. However, we believe the benefits of developing more robust, privacy-preserving models outweigh these risks.

References

[1] AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card* 1 (2024), 1.

[2] Malyaban Bal and Abhronil Sengupta. 2024. Spikingbert: Distilling bert to train spiking language models using implicit differentiation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 38. 10998–11006.

[3] Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 535–541.

[4] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*. IEEE, 1897–1914.

[5] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. 2023. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*. 5253–5270.

[6] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. 2023. Quantifying Memorization Across Neural Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/forum?id=TatRHT_1cK

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

[8] chargoddard. 2024. WebInstructSub-prometheus. https://huggingface.co/datasets/chargoddard/WebInstructSub-prometheus

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.

[10] André Vicente Duarte, Xuandong Zhao, Arlindo L. Oliveira, and Lei Li. 2024. DE-COP: Detecting Copyrighted Content in Language Models Training Data. In *Proceedings of the 41st International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.), Vol. 235. PMLR, 11940–11956. https://proceedings.mlr.press/v235/duarte24a.html

[11] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The Llama 3 Herd of Models. *CoRR* abs/2407.21783 (2024). https://doi.org/10.48550/arXiv.2407.21783

[12] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *arXiv preprint arXiv:2101.00027* (2020).

[13] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision* 129, 6 (2021), 1789–1819.

[14] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).

[15] Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016).

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[17] Abhyuday Jagannatha, Bhanu Pratap Singh Rawat, and Hong Yu. 2021. Membership inference attack susceptibility of clinical language models. *arXiv preprint arXiv:2104.08305* (2021).

[18] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 4163–4174. https://doi.org/10.18653/v1/2020.findings-emnlp.372

[19] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1eA7AEtvS

[20] Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. 2020. MixKD: Towards Efficient Distillation of Large-scale Language Models. *Arxiv preprint* (1 Nov. 2020).

[21] Matthieu Meeus, Shubham Jain, Marek Rei, and Yves-Alexandre de Montjoye. 2024. Did the neurons read your book? document-level membership inference for large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*. 2369–2385.

[22] Maximilian Mozes, Xuanli He, Bennett Kleinberg, and Lewis D Griffin. 2023. Use of llms for illicit purposes: Threats, prevention measures, and vulnerabilities. *arXiv preprint arXiv:2308.12833* (2023).

[23] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 807–814.

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

[25] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. FitNets: Hints for Thin Deep Nets. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). `http://arxiv.org/abs/1412.6550`

[26] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR* abs/1910.01108 (2019). arXiv:1910.01108 `http://arxiv.org/abs/1910.01108`

[27] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting Pretraining Data from Large Language Models. In *The Twelfth International Conference on Learning Representations*. `https://openreview.net/forum?id=zWqr3MQuNs`

[28] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.

[29] Kaitao Song, Hao Sun, Xu Tan, Tao Qin, Jianfeng Lu, Hongzhi Liu, and Tie-Yan Liu. 2020. Light{PAFF}: A Two-Stage Distillation Framework for Pre-training and Fine-tuning. `https://openreview.net/forum?id=B1xv9pEKDS`

[30] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient Knowledge Distillation for BERT Model Compression. 4314–4323. `https://doi.org/10.18653/v1/D19-1441`

[31] Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 2158–2170. `https://doi.org/10.18653/v1/2020.acl-main.195`

[32] Syed-Hasan-8503. 2024. Gemma-2-2b-it-distilled. `https://huggingface.co/Syed-Hasan-8503/Gemma-2-2b-it-distilled`

[33] Xinyu Tang, Richard Shin, Huseyin A Inan, Andre Manoel, Fatemehsadat Mireshghallah, Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, and Robert Sim. 2023. Privacy-preserving in-context learning with differentially private few-shot generation. *arXiv preprint arXiv:2309.11765* (2023).

[34] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118* (2024).

[35] Inar Timiryasov and Jean-Loup Tastet. 2023. Baby Llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell (Eds.). Association for Computational Linguistics, Singapore, 279–289. `https://doi.org/10.18653/v1/2023.conll-babylm.24`

[36] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[37] Cheng Wang, Yiwei Wang, Bryan Hooi, Yujun Cai, Nanyun Peng, and Kai-Wei Chang. 2024. Con-ReCall: Detecting Pre-training Data in LLMs via Contrastive Decoding. *arXiv preprint arXiv:2409.03363* (2024).

[38] Junxiong Wang, Daniele Paliotta, Avner May, Alexander Rush, and Tri Dao. 2024. The mamba in the llama: Distilling and accelerating hybrid models. *Advances in Neural Information Processing Systems* 37 (2024), 62432–62457.

[39] Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. 2022. On the Importance of Difficulty Calibration in Membership Inference Attacks. In *International Conference on Learning Representations*. `https://openreview.net/forum?id=3eIrli0TwQ`

[40] Roy Xie, Junlin Wang, Ruomin Huang, Minxing Zhang, Rong Ge, Jian Pei, Neil Zhenqiang Gong, and Bhuwan Dhingra. 2024. ReCaLL: Membership Inference via Relative Conditional Log-Likelihoods. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 8671–8689. `https://doi.org/10.18653/v1/2024.emnlp-main.493`

[41] Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116* (2024).

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

[43] Rongzhi Zhang, Jiaming Shen, Tianqi Liu, Jialu Liu, Michael Bendersky, Marc Najork, and Chao Zhang. 2023. Do Not Blindly Imitate the Teacher: Loss Perturbation for Knowledge Distillation. `https://openreview.net/forum?id=FILleBqk31S`

[44] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*. 19–27.

A Ablation Study on the Effect of Testing Subsets on MIA Accuracy

To show that only testing models on a subset of 100,000 data points is reasonable, we conducted an ablation study on the effect of size of test sets on MIA accuracy. The results are shown in Table 5.

The results show that the MIA accuracy is quite stable across different subset sizes. The mean for the loss method is 0.090 with a standard deviation of 0.015, and the mean for the zlib method is 0.063 with a standard deviation of 0.049. Since the size of the testing set is not a significant factor in the MIA accuracy, using a subset of 100,000 data points is a reasonable choice for our study.

Table 5: MIA accuracy for BERT using test sets of different sizes.

Subset Size	MIA Accuracy	
	Loss	Zlib
1000	0.068	0.134
5000	0.094	0.027
10000	0.075	0.030
30000	0.107	0.035
50000	0.097	0.036
100000	0.101	0.117

Table 6: Comparison of MIA accuracy between teacher and student models with 95% confidence intervals. **Bolded** entries denote the lower accuracy between each teacher-student model pair for each MIA method; Underlined entries denote that the difference between teacher and student accuracy is statistically significant.

Teacher Model	Student Model	MIA Method	Teacher Accuracy	Student Accuracy
BERT	DistilBERT	Loss	0.103 [0.101, 0.104]	**0.059** [0.058, 0.060]
		Zlib	**0.120** [0.119, 0.121]	0.169 [0.168, 0.171]
	TinyBERT	Loss	**0.103** [0.101, 0.104]	0.696 [0.694, 0.698]
		Zlib	0.120 [0.119, 0.121]	**0.091** [0.090, 0.092]
	ALBERT	Loss	**0.103** [0.101, 0.104]	0.385 [0.383, 0.387]
		Zlib	**0.120** [0.119, 0.121]	0.165 [0.164, 0.167]
	MobileBERT	Loss	0.103 [0.101, 0.104]	**0.101** [0.099, 0.102]
		Zlib	**0.120** [0.119, 0.121]	0.189 [0.187, 0.190]
GPT2	DistilGPT2	ReCaLL	0.719 [0.717, 0.721]	**0.571** [0.569, 0.573]
		Loss	**0.264** [0.262, 0.265]	0.421 [0.419, 0.423]
		Zlib	**0.832** [0.830, 0.834]	0.859 [0.857, 0.860]
Llama 360M	BabyLlama	ReCaLL	**0.491** [0.470, 0.513]	0.515 [0.497, 0.533]
		Loss	**0.494** [0.472, 0.516]	0.496 [0.475, 0.518]
		Zlib	0.507 [0.487, 0.526]	**0.503** [0.482, 0.523]
Llama 3.1 8B	Llama 3.2 3B	ReCaLL	**0.490** [0.469, 0.512]	0.491 [0.469, 0.513]
		Loss	**0.499** [0.477, 0.521]	0.518 [0.499, 0.537]
		Zlib	**0.510** [0.488, 0.531]	0.512 [0.490, 0.534]
	Llama 3.2 1B	ReCaLL	**0.490** [0.469, 0.512]	0.513 [0.491, 0.534]
		Loss	**0.499** [0.477, 0.521]	0.514 [0.495, 0.533]
		Zlib	**0.510** [0.488, 0.531]	0.514 [0.493, 0.535]
	MambaInLlama	ReCaLL	**0.490** [0.469, 0.512]	0.521 [0.502, 0.539]
		Loss	**0.499** [0.477, 0.521]	0.511 [0.493, 0.529]
		Zlib	**0.510** [0.488, 0.531]	0.511 [0.491, 0.531]
Gemma 2 27B	Gemma 2 9B	ReCaLL	0.606 [0.541, 0.670]	0.606 [0.544, 0.667]
		Loss	0.541 [0.476, 0.607]	**0.509** [0.443, 0.576]
		Zlib	0.569 [0.507, 0.631]	**0.541** [0.481, 0.601]
	Gemma 2 2B	ReCaLL	**0.606** [0.541, 0.670]	0.624 [0.560, 0.688]
		Loss	0.541 [0.476, 0.607]	**0.528** [0.464, 0.591]
		Zlib	0.569 [0.507, 0.631]	**0.541** [0.476, 0.607]
	Gemma 2 2B Distilled	ReCaLL	0.606 [0.541, 0.670]	**0.569** [0.503, 0.634]
		Loss	**0.541** [0.476, 0.607]	0.550 [0.488, 0.613]
		Zlib	0.569 [0.507, 0.631]	**0.555** [0.492, 0.618]

B Teacher-Student Model Comparison Results With Confidence Intervals

In Section 5.2.1, we present Table 1 with MIA accuracy across teacher-student model pairs and whether the accuracy is statistically significant. Table 6 gives the full results including the 95% confidence intervals for each attack using Equation 4. Based on the confidence intervals, we consider the accuracy difference between the teacher and student models to be statistically significant if their confidence intervals do not overlap.