Membership Inference Attacks by Exploiting Loss Trajectory

Yiyong Liu
CISPA Helmholtz Center for Information Security

Zhengyu Zhao
CISPA Helmholtz Center for Information Security

Michael Backes
CISPA Helmholtz Center for Information Security

Yang Zhang
CISPA Helmholtz Center for Information Security

ABSTRACT

Machine learning models are vulnerable to membership inference attacks in which an adversary aims to predict whether or not a particular sample was contained in the target model's training dataset. Existing attack methods have commonly exploited the output information (mostly, losses) solely from the given target model. As a result, in practical scenarios where both the member and non-member samples yield similarly small losses, these methods are naturally unable to differentiate between them. To address this limitation, in this paper, we propose a new attack method, called TrajectoryMIA, which can exploit the membership information from the whole training process of the target model for improving the attack performance. To mount the attack in the common black-box setting, we leverage knowledge distillation, and represent the membership information by the losses evaluated on a sequence of intermediate models at different distillation epochs, namely *distilled loss trajectory*, together with the loss from the given target model. Experimental results over different datasets and model architectures demonstrate the great advantage of our attack in terms of different metrics. For example, on CINIC-10, our attack achieves at least 6× higher true-positive rate at a low false-positive rate of 0.1% than existing methods. Further analysis demonstrates the general effectiveness of our attack in more strict scenarios.[1]

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies** → **Machine learning**;

KEYWORDS

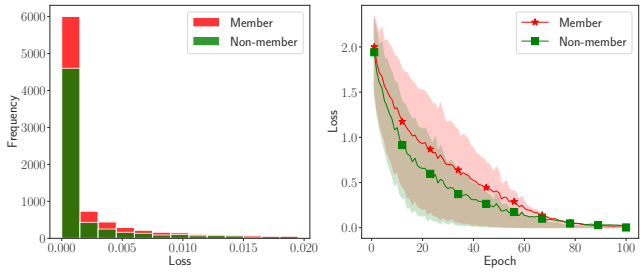
membership inference; loss trajectory; knowledge distillation

ACM Reference Format:
Yiyong Liu, Zhengyu Zhao, Michael Backes, and Yang Zhang. 2022. Membership Inference Attacks by Exploiting Loss Trajectory. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22), November 7–11, 2022, Los Angeles, CA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3548606.3560684

[1]Our code is available at https://github.com/DennisLiu2022/Membership-Inference-Attacks-by-Exploiting-Loss-Trajectory.

CCS '22, November 7–11, 2022, Los Angeles, CA, USA

ACM ISBN 978-1-4503-9450-5/22/11…$15.00
https://doi.org/10.1145/3548606.3560684



(a) Loss Distribution **(b) Loss Trajectory**

Figure 1: (a) the loss distribution of all target member and non-member samples on the target model; (b) the *loss trajectory* for specific member and non-member samples that have similarly small (< 0.02) losses on the target model. Using loss trajectory can help differentiate between these specific member and non-member samples.

1 INTRODUCTION

Recent machine learning (ML) tasks have involved sensitive data such as healthy records in model training. However, prior studies [5, 12, 40] have shown that most of the training data can be memorized by the ML models, which incurs the risk of privacy leakage. Membership inference attack (MIA) [40] is one of the privacy attacks against ML models whereby an adversary aims to infer whether or not a target sample was used to train a specific ML model. As of today, MIA is the de facto standard for evaluating ML models' privacy risks.

In order to infer the membership of a given target sample, most of the current MIA methods have used its losses (or posteriors) obtained from the target model as their inputs [37, 40, 52]. The general assumption of these MIAs is that member samples have overall smaller losses than non-member samples [52]. These MIAs are effective in terms of average-case metrics (e.g. balanced accuracy and ROC-AUC); however, they cannot differentiate between member and non-member samples that have similar losses, while the fact is most non-member samples have similarly small losses as member samples, as illustrated in Figure 1a. This is also the reason that the current MIAs suffer a relatively high false-positive rate [4]. In this paper, we investigate whether an adversary can leverage other signals to improve membership inference performance, in particular, reducing the attack's false-positive rate.

Although a non-member sample might have a small loss like a member sample on a target model, since it does not participate in the training process of the model, it might exhibit a different *loss trajectory*, i.e., its losses evaluated on the target model at its different training epochs, than a member sample. Our general hypothesis is that an adversary can exploit a sample's loss trajectory to differentiate between member and non-member samples. As can be seen from Figure 1b, there are indeed substantial differences between the loss trajectory of member vs. non-member samples even when both have similarly small losses on the target model (the losses at its last epoch). Specifically, the loss trajectory of non-member samples is lower than that of member samples. This can be explained based on the hypothesis of sample hardness [4, 15, 36, 46, 48], which suggests these non-member samples with small losses are essentially easy samples while member samples are gradually learned through the whole training process to eventually reach similarly small losses. See more detailed discussion in Section 3.2.

In this paper, we propose a novel membership inference attack against machine learning models, namely TrajectoryMIA, which leverages target samples' loss trajectory to differentiate members from non-members. We focus on machine learning classifiers in the image domain as most of the MIAs are evaluated in the same setting. However, we emphasize that TrajectoryMIA is general and can be directly applied to ML models in other domains.

To mount TrajectoryMIA, the first step is to obtain the target sample's loss trajectory. However, in practical scenarios, the adversary can only observe the final trained target model instead of all the intermediate models during the target model's training process. To solve this, we leverage knowledge distillation [21]. Concretely, the adversary first performs a black-box model distillation to the target model to obtain a distilled model. During the process, they keep all the intermediate versions of the distilled model locally. Here, different versions correspond to different training epochs. Then, the adversary evaluates the target sample's loss on each of the intermediate distilled models to obtain the target sample's loss trajectory, namely *distilled loss trajectory*. In the end, the attack model, which is a membership classifier, takes as input the target sample's distilled loss trajectory as well as its loss on the original target model to infer the sample's membership status. Note that TrajectoryMIA only needs to perform model distillation once and keeps on reusing the distilled intermediate models for all target samples at inference time.

We evaluate TrajectoryMIA on a comprehensive suite of benchmark datasets, with extensive comparisons to other advanced attack methods. Following the recent recommendation on evaluating MIAs [4], we mainly focus on the metric that measures *True-Positive Rate at low False-Positive Rate* (TPR at low FPR), but also report results in terms of other average-case metrics, including balanced accuracy and ROC-AUC. Experimental results show that TrajectoryMIA is able to achieve 5.3% TPR at 0.1% FPR on the CINIC-10 dataset, at least 6× better than other considered advanced attacks. Furthermore, we evaluate the attack performance at a more fine-grained level by calculating the TPR at low FPR for separate groups of target samples that have varied loss values on the target model. The evaluation results demonstrate that TrajectoryMIA achieves strong performance in all the settings. We conduct extensive ablation studies to analyze the impact of different important factors on the attack success of TrajectoryMIA, e.g., the size of the dataset used for training the target model and for the knowledge distillation, as well as the number of epochs used in the distillation process. We further explore TrajectoryMIA in more strict scenarios with relaxed assumptions about the knowledge of the adversary, including different architectures between the target model and local models, and data distribution shift. Finally, we provide additional insights into understanding the characteristics of TrajectoryMIA, by discussing the importance of its main components. In general, this paper makes the following contributions:

- We take the first step to exploit the information from the training process of the target model to conduct membership inference attacks, and propose a novel attack method TrajectoryMIA based on knowledge distillation.
- We demonstrate that TrajectoryMIA consistently outperforms other advanced attack methods in common scenarios, but also in more strict scenarios with relaxed assumptions.
- We provide in-depth analyses about the impact of each component of TrajectoryMIA and other important factors on the attack performance.

Roadmap. In Section 2, we introduce the preliminary knowledge about machine learning, membership inference attacks, and knowledge distillation. Section 3 presents the threat model, design intuition, and the details of our attack method. We conduct extensive experiments to show the effectiveness of our attack in Section 4, and the impact of important factors on the attack performance in Section 5. In Section 6, we provide an in-depth analysis on the impact of each component in our attack. We discuss the related work in Section 7 and conclude the paper in Section 8.

2 PRELIMINARY

2.1 Machine Learning

For machine learning classification tasks, a learned neural network $\mathcal{M}_\theta$ is a function that maps each data sample from a dataset $\mathbf{X}$ to its class/label in a label set $\mathbf{Y}$. Given a sample $\mathbf{x}$, its output from $\mathcal{M}_\theta$, denoted as $p = \mathcal{M}_\theta(\mathbf{x})$, is a vector that represents the prediction posteriors of the sample over different pre-defined classes. In order to train a ML model, a loss function $\mathcal{L}(y, p)$ is defined to determine the error between a sample's prediction posteriors and its corresponding label. Cross-entropy loss is one of the most common loss functions used for classification tasks, and it is defined as:

$$\mathcal{L}_{CE}(y, p) = -\sum_{i=1}^{k} y_i \log p_i \tag{1}$$

where k is the total number of classes. y_i equals 1 only if the sample belongs to class i and otherwise 0, and p_i is the i-th element of the prediction posteriors. The model training is implemented to minimize the empirical loss by stochastic gradient descent:

$$\theta_{i+1} \leftarrow \theta_i - \epsilon \sum_{(x,y)\in\mathcal{B}} \nabla_\theta \mathcal{L}(y, \mathcal{M}_{\theta_i}(\mathbf{x})) \tag{2}$$

where $\mathcal{B}$ is a small batch of training samples and ϵ is the learning rate for iteratively updating the parameters θ of the neural network. The model will be trained for multiple epochs (times that the entire

training set is passed to the model) in order to achieve a well-generalized model. Normally, the intermediate models at different epochs can be preserved and the training process can be stopped at a specific epoch.

2.2 Membership Inference Attacks

The objective of membership inference attacks is to identify whether or not a target sample exists in the training set of the given model. First proposed by [40], MIA has drawn great attention in various scenarios [6, 7, 16, 20, 30, 31, 42, 56]. What makes the MIA so important is its connection to privacy leakage due to the increasingly sensitive data for training ML models and also its simplicity to be deployed.

Definition of MIA. Concretely, given a target sample $\mathbf{x}$, a trained ML model $\mathcal{M}_\theta$ and some external knowledge of the adversary, denoted by $\mathcal{I}$, membership inference attacks $\mathcal{A}$ can be defined by the following function:

$$\mathcal{A} : \mathbf{x}, \mathcal{M}_\theta, \mathcal{I} \rightarrow \{0, 1\} \tag{3}$$

Here, 0 means $\mathbf{x}$ is not a member from $\mathcal{M}_\theta$'s training set and 1 means $\mathbf{x}$ is a member. Most of the current MIA attacks are based on training a binary classifier, which we follow as well in this paper.

Adversary's Knowledge. Basically, the adversary is assumed to have black-box access to the target model, that is, they can only obtain the posteriors output by the target model for each query sample. In addition, they are able to leverage an auxiliary dataset that comes from the same distribution as the training set of the target model. In this way, they can train a shadow model to mimic the behavior of the target model and take as input the posteriors output by the shadow model for training a binary classifier, namely attack model. The trained attack model is then used to infer the membership of any given target sample. To make the attack more efficient, some studies [37, 52] use the output from the target model directly as a signal to predict the membership status without training any shadow model.

Recently, MIA has also been explored in other scenarios, including white-box [31] and label-only [8, 28, 34]. The former scenario assumes the adversary has full access to the target model, which contains the model architecture and parameters. The latter scenario considers a more strict setting in which the adversary can only get the hard label predicted by the target model. As a result, the adversary mounts the attack by perturbing the sample to change its predicted label, and then measures the magnitude of the perturbation. If it is larger than a predefined threshold, the adversary will consider the sample as a member and otherwise a non-member.

Defense Against MIA. The overfitting level of the ML model is one of the major factors that influence the success of the MIA as demonstrated in [37, 40]. For this reason, general regularization techniques, such as Dropout [44] and confidence penalty [33], can be used to defend against MIAs. Besides, knowledge distillation is another effective tool for mitigating MIAs, such as PATE [32] and DMP [39]. MemGuard [22] obfuscates the output from the target model to reduce the information that can be leveraged by the adversary. However, it fails when a label-only MIA is conducted. DP-SGD [3], one popular application of differential privacy (DP) in machine learning, provides a provable privacy guarantee for defending against membership inference attacks. It can achieve strong protection against MIAs, but with a sacrifice in severe accuracy degradation for the original classification tasks.

2.3 Knowledge Distillation

Knowledge distillation (KD) represents a class of methods that train a smaller student model to have better performance by learning based on the output of a larger teacher model. The key idea of KD is that the soft information (i.e. output posteriors) from a larger teacher model contains a lot more information than the hard, ground-truth label. Here we adopt the most classical KD framework proposed by Hinton et al. [21]. Given any input $\mathbf{x}$, the corresponding output from the teacher model is essentially a vector of posteriors, denoted as $p^t = [p_1^t, \cdots, p_C^t]$, C is the number of classes, and the output from the student model is p^s. Normally, these posteriors are calculated through a softmax function, but in order to make them softer so as to extract more information from the teacher model, it is modified to:

$$\tilde{p}_i^t = \frac{\exp^{s_i^t/\tau}}{\sum_j \exp^{s_j^t/\tau}} \tag{4}$$

where s_i^t is the logit value before the softmax function for the i-th class and τ is the temperature used for controlling the softness level. To learn a student model through distillation, one only needs to submit a set of samples to the target model and obtains their posteriors. Then, the student model is trained on the samples supervised by their posteriors using the loss that is a linear combination of the typical Cross-entropy $\mathcal{L}_{cls}$ and the knowledge distillation loss $\mathcal{L}_{KD}$:

$$\mathcal{L} = \alpha \mathcal{L}_{cls} + (1 - \alpha)\mathcal{L}_{KD} \tag{5}$$

Here, $\mathcal{L}_{KD}$ is calculated between the soft posteriors output from the teacher model $\tilde{p}^t$ and student model $\tilde{p}^s$ by Kullback-Leibler divergence loss:

$$\mathcal{L}_{KD}(\tilde{p}^t, \tilde{p}^s) = \sum_{i=1}^{k} \tilde{p}_i^t \log \frac{\tilde{p}_i^t}{\tilde{p}_i^s} \tag{6}$$

In this paper, we aim to learn a student model that is as similar to the teacher model as possible, and thus we set $\alpha = 0$ and $\tau = 1$. Note that knowledge distillation in this setting is similar to model stealing techniques [47].

3 ATTACK METHODOLOGY

In this section, we introduce the methodology of TrajectoryMIA. We start by defining the threat model. Then, we introduce the design intuition of our attack and explain why it works. Finally, the detailed pipeline of our attack is provided to illustrate how to conduct TrajectoryMIA with knowledge distillation in practical scenarios with only black-box access to the target model.

3.1 Threat Model

In this paper, we focus on the commonly-adopted, black-box scenario of MIAs, in which the adversary only has access to the posterior output from the target model. For a given target model, we assume that the adversary has an auxiliary dataset $\mathcal{D}^a$ that comes from the same distribution as the target model's training

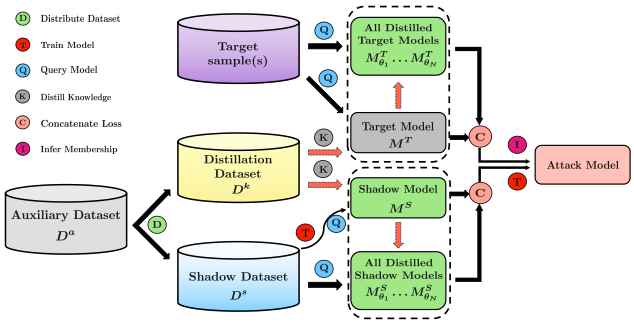


Figure 2: The working pipeline of our method. Different from the conventional MIA pipeline, our method uses a proportion of the auxiliary dataset as the knowledge distillation dataset for obtaining distilled target/shadow models at their different epochs, and each sample is represented by the concatenation of the loss from the target/shadow model and the distilled loss trajectories from the distilled target/shadow models.

set $\mathcal{D}^t$. This follows the standard setting of most of the advanced MIAs [4, 37, 40, 43, 48]. Both the data used to train the shadow model and used to distill the target/shadow model for obtaining the corresponding *distilled loss trajectory* are sampled from this auxiliary dataset. Furthermore, we assume the adversary knows the architecture of the target model. Later in Section 5, we show that these two assumptions on the adversary's knowledge about the training data distribution and the architecture of the target model can be relaxed.

3.2 Design Intuition

As ML models are trained to minimize the losses from their training samples, it is assumed that member samples are on average more likely to have smaller losses than non-member samples, which is also due to the overfitting of the model. Thus, losses (or posteriors) are used as common signals by most current methods to conduct membership inference. Although these loss-based attacks are effective in terms of the average-case metrics such as balanced accuracy and ROC-AUC; they actually fail to differentiate between member samples and non-member samples when both of them have small losses. We show it in Figure 1a, the loss distribution between member samples and non-member samples from the target model, that most non-member samples indeed get similar small losses as member samples. This causes a high false-positive rate in most existing MIAs and renders them unreliable in real scenarios. To this end, we propose to utilize other stronger signals to enhance membership inference and especially aim to reduce the attack's false-positive rate.

The main idea behind our attack is to leverage the major difference between member samples and non-member samples, that is the former indeed participates in the training of the target model while the latter does not. Our general hypothesis is that non-member samples should have a distinct changing pattern on the losses evaluated at each training epoch, namely *loss trajectory*, compared to member samples. As illustrated in Figure 1b, for those member and non-member samples that have similarly small losses from the model at its last training epoch (i.e., the given target model), they behave quite differently along the training process regarding the changing pattern of the loss trajectory. This difference has a close connection to sample hardness. Concretely, we show that non-member samples are indeed easier samples in terms of two existing sample hardness metrics. For the metric that defines the hardness as the epoch when the model does not change the prediction [15, 46], the result for members vs. non-members is 35.4 vs. 26.1. For the metric that defines the hardness as the loss from arbitrary reference models [4, 36, 48], the result is 0.006 vs. 0.005. Thus, loss trajectory can provide a more detailed profile for data samples and can be used as a stronger signal to conduct MIA.

3.3 Attack Method

Inspired by the differences in the loss trajectory between the member and non-member samples, we propose a new membership inference attack, called TrajectoryMIA. In order to mount TrajectoryMIA, the adversary needs to get the loss trajectory from the training process of the target model. However, in practical scenarios, the adversary only has black-box access to the target model, i.e., only the target model at its last training epoch is directly accessible. To address this issue, we leverage knowledge distillation. Specifically, the adversary first conducts a standard model distillation to the target model and gets a distilled model. By doing this, the adversary has full control of the distillation process and can preserve the distilled target models at different epochs. After distillation, the adversary can evaluate any given target sample on all intermediate distilled models to acquire its loss trajectory, which we call *distilled loss trajectory*. Finally, the attack model takes as input this distilled loss trajectory together with the loss obtained from the original target model to infer the membership.

The detailed pipeline of TrajectoryMIA is illustrated in Figure 2. It consists of four stages: shadow model training, model distillation, attack model training, and membership inference. In particular, the model distillation stage is newly introduced by our TrajectoryMIA, and the other three stages follow the common MIA pipeline except that the input to the attack model is different.

Shadow Model Training. As aforementioned, the adversary has an auxiliary dataset $\mathcal{D}^a$ drawn from the same distribution as the training dataset $\mathcal{D}^t$ of the target model $\mathcal{M}^T$. This auxiliary dataset is split into two disjoint subsets. One subset is used as the shadow dataset $\mathcal{D}^s$ to train a shadow model $\mathcal{M}^S$ in conjunction with classical training techniques. Following the common practice, we use the same architecture of the target model to build our shadow model.

Model Distillation. The other subset of the auxiliary dataset is used as the knowledge distillation dataset $\mathcal{D}^k$ to distill the trained shadow model and the target model in order to obtain their distilled loss trajectory. Specifically, in order to make the distilled model more similar to the original model (target model or trained shadow model), we only use the posteriors output from the original model and distilled model to calculate a Kullback-Leibler divergence loss in the distillation process, and do not consider the ground truth label. One thing needs to mention, for the shadow model, although we already have the loss trajectory from its actual training process, we still conduct the distillation to it in order to make sure its distilled

Table 1: Training and testing accuracy for all model architectures on different datasets.

Target Model	CIFAR-10		CIFAR-100		GTSRB		CINIC-10	
	Train acc	Test acc	Train acc	Test acc	Train acc	Test acc	Train acc	Test acc
MibileNetV2	97.2%	70.2%	100.0%	32.4%	100.0%	78.4%	98.8%	52.0%
VGG-16	100.0%	82.6%	99.9%	47.5%	100.0%	84.1%	99.9%	65.8%
ResNet-56	98.3%	76.2%	94.1%	42.2%	100.0%	88.1%	97.4%	60.7%
WideResNet-32	93.6%	77.1%	94.5%	45.1%	100.0%	85.4%	90.5%	62.8%

loss trajectory is better aligned with the distilled loss trajectory from the target model. This is reasonable as we cannot access the actual loss trajectory from the target model and the distilled loss trajectory of the target model may behave differently. More discussion regarding the advantage of this specific design choice can be found in Section 6. The distillation process for our shadow model can be derived from Equation 2 and Equation 6 as:

$$\theta_{i+1} \leftarrow \theta_i - \epsilon \sum_{(x,y)\in\mathcal{D}^k} \nabla_\theta \mathcal{L}_{KD}(\mathcal{M}^S(\mathbf{x}), \mathcal{M}^S_{\theta_i}(\mathbf{x})) \tag{7}$$

where $i = 1, 2\cdots, N$ as N is the number of epochs for model distillation, meaning that we can obtain N intermediate shadow models. Similarly, we use the same knowledge distillation dataset $\mathcal{D}^k$ to distill the target model and get N distilled target models in total.

Attack Model Training. The adversary trains an attack model on the shadow dataset $\mathcal{D}^s$ as common MIA methods do, but the only difference is that the input to the attack model becomes the concatenation of the loss trajectory from all distilled shadow models and the loss from the original shadow model:

$$\hat{\mathbf{x}} = \mathcal{L}(\mathcal{M}^S_{\theta_1}(\mathbf{x}), y) \oplus \cdots \oplus \mathcal{L}(\mathcal{M}^S_{\theta_N}(\mathbf{x}), y) \oplus \mathcal{L}(\mathcal{M}^S(\mathbf{x}), y) \tag{8}$$

Here $\hat{\mathbf{x}}$ is the input to the attack model and the corresponding label is 1 when $\mathbf{x}$ is used for training the shadow model and otherwise 0.

Membership Inference. Finally, the adversary can infer the membership of each given target sample by feeding the concatenation of its losses obtained from $N+1$ target models (including one original target model and all N distilled target models) to the trained attack model.

For label-only attacks, we do not have the posteriors but only the hard labels predicted by the target model. Thus, we can view the predicted hard label as the ground truth to calculate the Cross-entropy loss instead of the Kullback-Leibler divergence loss in the distillation process. Accordingly, the loss from the target model will be replaced by the HopSkipJump boundary distance, and the same is done for the shadow model as we want to better align the training and testing phase of the attack model.

4 EVALUATION

In this section, we conduct extensive experiments to evaluate our TrajectoryMIA on diverse model architectures and benchmarking datasets, with comparisons to other representative attack baselines. We focus on the commonly-adopted, black-box attack scenario, but also explore the more challenging, label-only scenario.

4.1 Experimental Setup

Datasets. For the main experiments, we consider the following four image datasets:

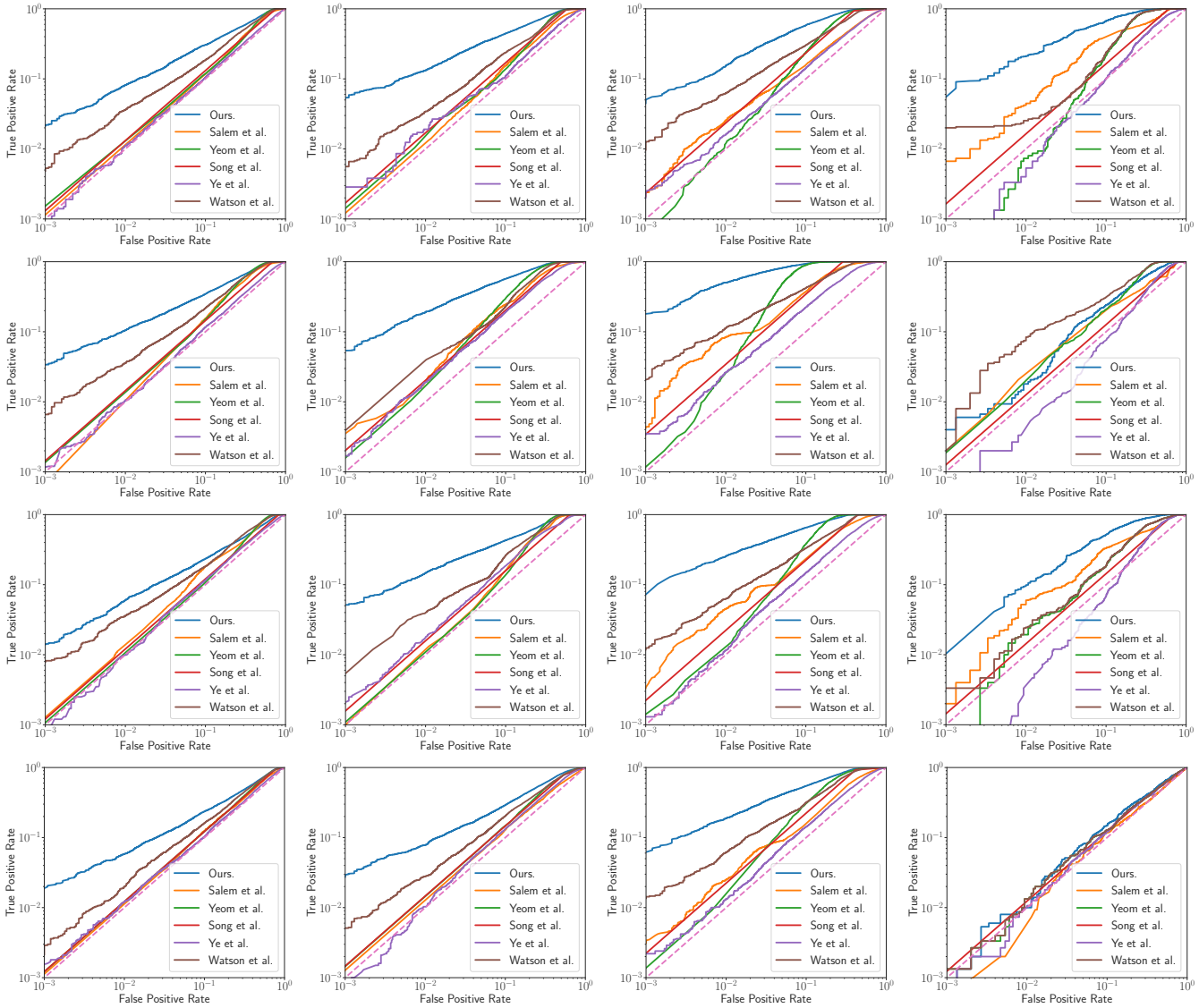
- **CIFAR-10 [1].** The CIFAR-10 is a benchmark dataset used for classification tasks, which has totally 60000 images with 10 classes and each class has the same number of samples. Each sample has a size of 32×32×3.
- **CINIC-10 [9].** CINIC-10 is an extension of CIFAR-10 via the addition of downsampled ImageNet [10] images for the same classes in CIFAR-10. The number of classes is 10 as well but with 270000 images in total. The size of each sample is 32×32×3.
- **CIFAR-100 [1].** Similar to CIFAR-10 dataset, CIFAR-100 also contains 60000 images with a size of 32×32×3. And it has 100 classes with 600 images for each class.
- **GTSRB [2].** German Traffic Sign Recognition Benchmark (GTSRB) is a classification benchmark with 51839 images for all 43-category traffic signs. Since the size of each sample varies, we resize them to 32×32×3.

We also show the effectiveness of our approach on the following three datasets beyond the image domain:

- **Purchase.** The Purchase is a simplified dataset based on Kaggle's "acquire valued shoppers" dataset (with 197324 records), where each record has 600 binary features. Following Shokri et al. [40], we use it to conduct a 100-classes classification task.
- **Location [50].** The Location is a "check-in" dataset in the Foursquare social network. We use the same method in [40] to filter out the whole dataset and get 5010 records with 30 classes in total, and each record has 446 binary features.
- **News.** The News (20 Newsgroup) dataset is a commonly used dataset for text classification. The dataset consists of 20000 newsgroup documents categorized into 20 classes. We follow [37] to preprocess the dataset in our experiments.

For each dataset, the number of samples for constructing the training and testing sets of target and shadow models ($\mathcal{D}^t_{train}$, $\mathcal{D}^t_{test}$, $\mathcal{D}^s_{train}$, and $\mathcal{D}^s_{test}$, respectively) are the same and the remaining data samples are used for knowledge distillation dataset $\mathcal{D}^k$. The details of data splitting for different datasets can be found in [29].

Models. For image datasets, we consider four popular neural network architectures including ResNet-56 [17], MobileNetV2 [38], VGG-16 [41], and WideResNet-32 [54] for the target, shadow, and distilled models. And for the other three datasets, we apply a 2-layer MLP. We use SGD with a learning rate of 0.1, Nesterov momentum

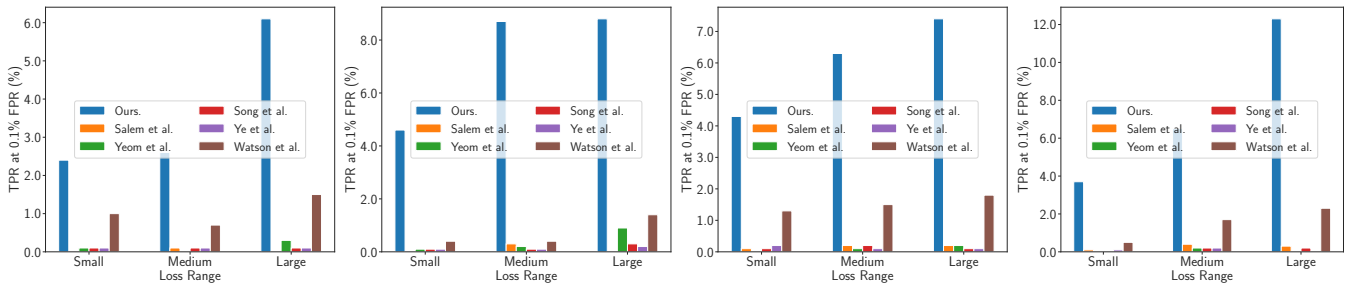


(a) CIFAR-10 (b) CINIC-10 (c) CIFAR-100 (d) GTSRB

Figure 3: ROC curves for attacks on four different datasets and four model architectures (from top to bottom: ResNet-56, MobileNetV2, VGG-16, and WideResNet-32).

Table 2: Attack performance of different attacks against ResNet-56 trained on four datasets. Additional results for the other three model architectures with a similar pattern can be found in our technical report [29].

Attack	TPR at 0.1% FPR				Balanced accuracy				AUC			
method	CIFAR-10	CINIC-10	CIFAR-100	GTSRB	CIFAR-10	CINIC-10	CIFAR-100	GTSRB	CIFAR-10	CINIC-10	CIFAR-100	GTSRB
Salem et al. [37]	0.1%	0.0%	0.2%	0.3%	0.610	0.623	0.577	0.677	0.628	0.646	0.612	0.755
Yeom et al. [52]	0.1%	0.2%	0.1%	0.0%	0.647	0.705	0.772	0.797	0.646	0.755	0.804	0.818
Song et al. [43]	0.1%	0.2%	0.1%	0.0%	0.650	0.707	0.773	0.681	0.644	0.728	0.804	0.820
Ye et al. [51]	0.0%	0.1%	0.2%	0.0%	0.527	0.603	0.578	0.606	0.531	0.632	0.605	0.618
Watson et al. [48]	0.5%	0.6%	0.9%	1.5%	0.631	0.698	0.727	0.798	0.677	0.735	0.778	0.822
Ours	**2.1%**	**5.3%**	**4.9%**	**7.3%**	**0.650**	**0.730**	**0.800**	**0.839**	**0.724**	**0.819**	**0.886**	**0.914**



(a) CIFAR-10 (b) CINIC-10 (c) CIFAR-100 (d) GTSRB

Figure 4: TPR at 0.1% FPR of different attacks for ResNet-56 trained on four datasets for samples with different ranges of losses obtained from the target model. Here we consider three loss ranges, 'small': [0.0,0.02), 'medium': [0.02,0.2), and 'large': [0.2,+∞]. Additional results for the other model architectures with a similar pattern can be found in our technical report [29].

of 0.9 and a cosine learning rate schedule for optimization. We also adopt data augmentations for images in order to make the models more generalized. All the models are trained from 20 to 150 epochs in terms of the model size and dataset complexity. For the attack model, we train a 4-layer MLP.

Metrics. We consider the following evaluation metrics:

- **Full Log-scale ROC.** It is the commonly-used Receiver Operating Characteristic (ROC) curve comparing the ratio of true-positives to false-positives, but reported in logarithmic scale for emphasizing the low FPR regime [4]
- **TPR at Low FPR.** It summarizes the attack performance at a single low false-positive rate for quick evaluation [4]. We also take a step further to apply this metric to separate groups of samples that have different levels of loss obtained from the target model.
- **Balanced Accuracy and AUC.** They are two widely used average-case metrics to measure the performance for binary classification tasks, including most previous MIAs [30, 37, 48, 51]. Here the "Balanced" means that the number of the member and non-member samples is the same. Since they are not the most suitable metrics for evaluating MIAs, we adopt them here just for completeness.

Attack Baselines. We mainly compare our new MIA method with five representative existing methods [37, 43, 48, 51, 52] as the baselines. Among them, Salem et al. [37] utilize posteriors to conduct the attack while Yeom et al. [52] leverage the loss from target model; Song et al. [43] propose a metric-based method without the use of attack model; Watson et al. [48] and Ye et al. [51] both consider sample hardness where the former use reference models and the latter leverage distilled models. To ensure a fair comparison, those methods that involve model training have access to the same auxiliary dataset, and use a single shadow/reference model like ours. Please refer to Section 7.1 for more descriptions of these five methods.

4.2 Experimental Results

Here, we show the detailed attack results in the black-box setting with a comparison to 5 advanced baseline methods; among them, we compare our attack to a state-of-the-art method LiRA as well. Besides, we apply our attack in the label-only scenario and also attack the model defended with DP-SGD. Table 1 reports the performance of the models we attack.

Table 3: Attack performance of our attack and LiRA (online version) [4] for ResNet-56 trained on CINIC-10.

Metric	LiRA [4]	Ours
TPR at 0.1% FPR	5.0%	**5.3%**
Balanced accuracy	0.713	**0.730**
AUC	0.793	**0.819**

Black-box Attacks. We first evaluate different attacks in the black-box scenario. As can be seen from Figure 3, our method consistently achieves the best performance on the low-FPR regime. The TPR at 0.1% FPR further confirms this conclusion, as shown in Table 2. Regarding the two aggregate metrics, balanced accuracy and AUC, we can also observe that our method strictly dominates all the baselines.

We then evaluate different attacks at a more fine-grained level by looking at the TPR at 0.1% FPR on separate groups of samples that have different levels of loss values from the target model. Here we consider three levels of loss values, i.e., 'small': [0.0,0.02), 'medium': [0.02,0.2), and 'large': [0.2,+∞). For the GTSRB dataset, we run the experiments for five times and take the average because some loss ranges may suffer from unstable results due to the limited number of data samples. As demonstrated in Figure 4, our attack outperforms all other baselines across all different loss ranges. The above experimental observations also hold for the other three model architectures (MobileNetV2, VGG-16, and WideResNet-32) and three datasets (Purchase, Location, and News), for which the results can be found in our technical report [29].

Comparison with LiRA. Besides the evaluation recommendation on the low-FPR regime, Carlini et al. [4] also introduce a new attack method called Likelihood Ratio Attack (LiRA), which achieves the state-of-the-art attack performance. For LiRA, the adversary trains N shadow models, of which half are IN models (trained with target sample $(\mathbf{x}, y)$ and the other half are OUT models (trained without

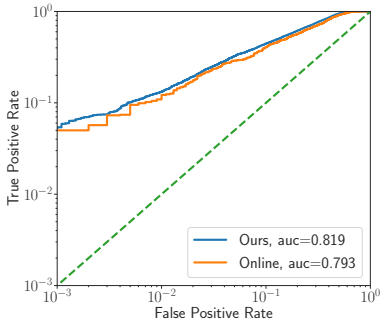


Figure 5: ROC curves of our attack compared with LiRA (online version) [4] for ResNet-56 trained on CINIC-10.

Table 4: TPR at 0.1% FPR of different attacks on CINIC-10 in the label-only setting.

Model Architecture	Li & Zhang [28]	Watson et al. [48]	Ours
MobileNetV2	0.2%	0.2%	**0.8%**
VGG-16	0.2%	0.1%	**0.3%**
ResNet-56	**0.3%**	0.2%	**0.3%**
WideResNet-32	0.0%	0.0%	**0.1%**

Table 5: Balanced acc of different attacks on CINIC-10 in the label-only setting.

Model Architecture	Li & Zhang [28]	Watson et al. [48]	Ours
MobileNetV2	0.782	0.761	**0.828**
VGG-16	0.735	0.728	**0.806**
ResNet-56	0.713	0.676	**0.733**
WideResNet-32	0.615	0.595	**0.667**

$(\mathbf{x}, y)$). Then Gaussians are fitted to the confidences of the IN and OUT models on $(\mathbf{x}, y)$. Finally, the confidence of $(\mathbf{x}, y)$ from the target model will be used to conduct a parametric Likelihood-ratio test.

Following the original work of LiRA, 256 shadow models (128 IN models and 128 OUT models) are trained. We find that using fewer shadow models results in worse attack performance, and using only (128) out models also leads to a low performance, 2.1%. To ensure a fair comparison, LiRA also queries the target model once for each sample instead of multiple times, the same as in our attack. It can be seen from Table 3 that LiRA achieves comparable (but a bit lower) attack performance to ours, and the detailed comparison in terms of ROC curve in Figure 5 further confirms this. However, LiRA is not practically feasible due to the necessity of training N shadow models for each given target sample at inference time. In contrast, our attack requires no inference-time model training but only queries to obtain the corresponding loss trajectory.

Label-only Attacks. When the target model only returns a hard prediction, that is a single label, rather than a continuous-valued output like posteriors, membership inference can still be conducted by label-only attacks. Here we use the CINIC-10 dataset and ResNet-56 to evaluate our attack in the label-only scenario. We compare

Table 6: Attack performance of our attack against DP-SGD for ResNet-56 trained on CINIC-10.

Noise Multiplier (σ)	C = 10 ϵ	Model acc	Attack acc	TPR at 0.1% FPR
0.0	∞	0.520	0.560	0.2%
0.2	> 1000	0.485	0.544	0.2%
0.5	> 100	0.438	0.528	0.1%
1.0	8	0.332	0.512	0.1%

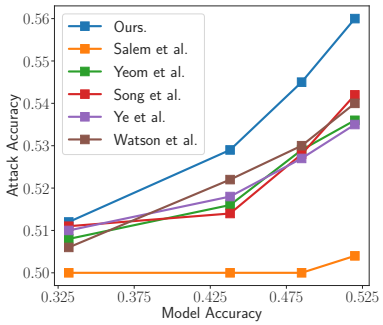


Figure 6: The trade-off between model accuracy and attack accuracy of different attacks against DP-SGD with different noise multiplier for ResNet-56 trained on CINIC-10.

the result to the original boundary-attack in [28] and label-only attacks with calibration in [48] as other baselines do not take this scenario into consideration.

Table 4 and Table 5 report the attack performance between TrajectoryMIA and other baselines. Similar to the results in Table 2, label-only attacks can also benefit from loss trajectory with increased TPR at 0.1% FPR and balanced attack accuracy. However, the performance increase is observably smaller compared to the results in score-based attacks. One possible reason is that solely relying on the hard predicted labels from the target model, the *distilled loss trajectory* contains much less information than using the posteriors; besides, due to the same reason, the loss from the original model is replaced by the corresponding HopSkipJump boundary distance, which is not as reliable as using the loss to infer the membership.

Attacking DP-SGD. Differential Privacy (DP) [11] is a widely used mechanism to defend machine learning models against different privacy attacks [24, 25, 42]. Essentially, it provides a bound on the ability to distinguish two neighboring datasets that differ in the presence of one data sample, which has a close connection to our problem of membership inference.

Here we adopt the popular mechanism DP-SGD [3] to evaluate our attack under the DP-based defense. We fix the clip bound C to 10 and change the noise multiplier from 0.0 to 1.0 to control the privacy budget ϵ. As can be seen from Table 6 and Figure 6, applying DP in the training process achieves strong defense effects against all attacks, although our attack still outperforms others across all privacy budgets. However, DP also reduces the classification accuracy heavily even when $C = 10$, $\sigma = 0.0$, and $\epsilon = \infty$.

Table 7: The impact of the knowledge distillation set size for ResNet-56 trained on CINIC-10. The accuracy of the target model is 60.7 %.

Knowledge distillation set size	Distilled model acc	Attack acc	Attack TPR at 0.1% FPR
20000	63.5%	0.713	1.8%
70000	63.4%	0.720	2.2%
120000	63.5%	0.729	3.5%
170000	63.9%	0.725	2.8%
220000	63.6%	0.730	5.3%

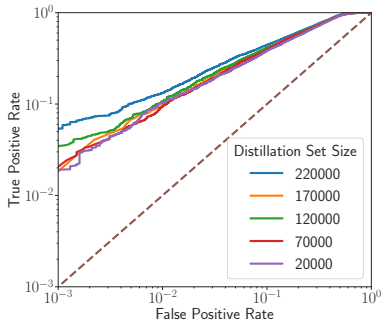


Figure 7: ROC curves of our attack with different knowledge distillation set size for ResNet-56 trained on CINIC-10.

This trade-off between defense strength and classification accuracy makes DP not feasible in practical scenarios.

5 ABLATION STUDY

In this section, we analyze the impact of several important factors on attack performance. We first discuss the impact of the size of the knowledge distillation dataset as well as the number of epochs used in distillation for getting the distilled loss trajectory. We then explore the impact of the overfitting level of the target model by changing the size of the target model training set. Finally, we relax the two major assumptions about the adversary, namely the data distributions of the auxiliary dataset $\mathcal{D}^a$ and the architectures of the target model.

5.1 Knowledge Distillation Set Size

For knowledge distillation, a generally important factor that influences the distillation performance is the size of the distillation dataset. Here we explore the impact of this factor on our attack performance. We follow the same settings in Section 4.2 for training the target model and the distilled models, except that the size of the knowledge distillation dataset $\mathcal{D}^k$ is varied from 20000 to 220000.

Table 7 and Figure 7 summarize the results. As expected, when the adversary is able to acquire a larger set of auxiliary data, the attack performance could be improved both in TPR at 0.1% FPR and balanced accuracy. Differently, the classification accuracy of different distilled models remains very similar. This difference indicates that the membership information distilled from the training process does not have a direct connection to the functionality (as represented by the classification accuracy) of the distilled model, which we will discuss more in Section 6.

5.2 Number of Knowledge Distillation Epochs

The number of epochs for knowledge distillation will influence both the computational cost in the distillation process and the input dimension to the attack model. Thus, it is crucial to figure out how many epochs are necessary in the distillation process to obtain the *distilled loss trajectory*.

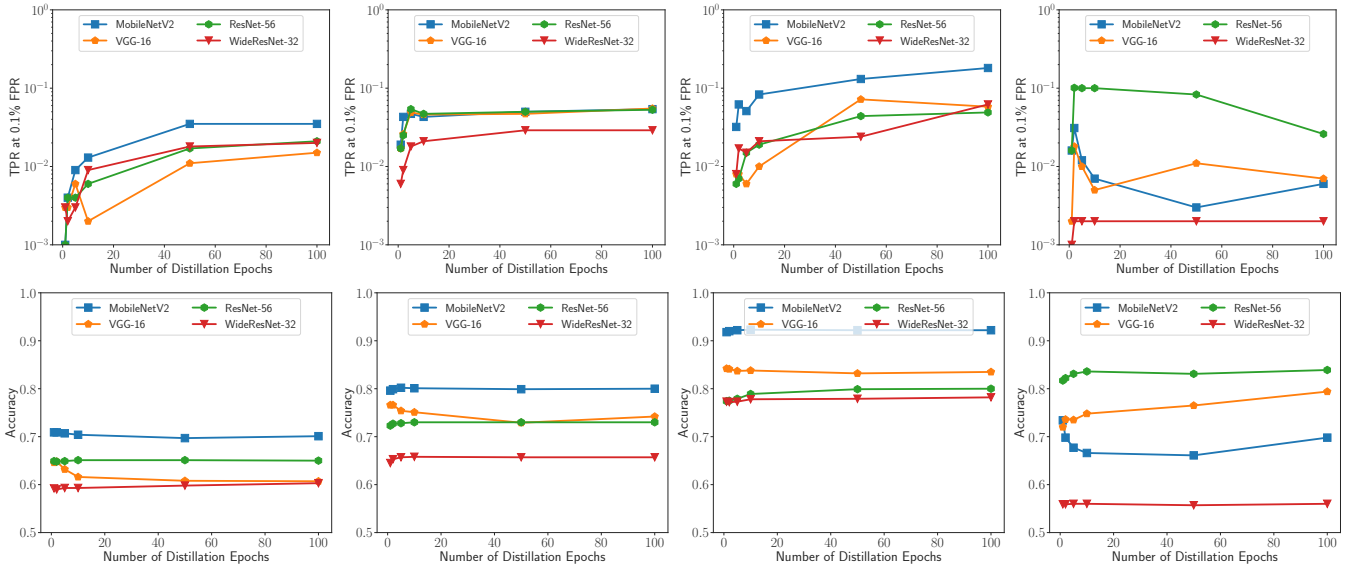
We can see from Figure 8 that distilling the target model for more epochs indeed increases the attack TPR at low FPR whereas has little impact on the attack accuracy across all different datasets and model architectures. However, although more distillation epochs can give us a better attack performance, the larger the number of epochs is, the lower the marginal benefits will be, thus we need to find a trade-off between them. Take CIFAR-10 and CINIC-10 for example, the number of epochs for the distilled models trained on these two datasets to reach a similar classification accuracy as the target model are around fifty and five respectively, and interestingly, just using the same epochs for distillation can also make the attack performance near the best. Another extreme case is the GTSRB dataset where if we continue the distillation for too many epochs after the distilled model already reaches the time point to have comparable functionality as the target model, it even degrades the attack performance. Thus, this time point can be used as a strong signal for us to stop the distillation so as to save the computational cost and keep a considerable attack performance at the same time.

Actually, there is an interrelationship between the size of the knowledge distillation set and the epochs needed in the distillation process, where more distillation data samples mean the model can be distilled quickly. Thus, we go a step further to investigate the connection between each other. As demonstrated in Figure 9, first, both larger distillation set size ($\mathcal{D}^k$) and more distillation epochs can improve the attack performance. More interestingly, regarding the TPR at 0.1% FPR, using 220000 distillation data samples with only 1 epoch can have comparable performance as using 20000 samples in distillation with 100 epochs, although the number of distillation epochs in the latter setting is 100 times larger than the former one. This indicates that the impact of distillation dataset size seems more significant. This is reasonable as a larger size of the distillation set can help the distilled model to be more representative and the *distilled loss trajectory* can obtain more information from the original one while more epochs contribute a little especially after the distilled model already has a similar classification accuracy as the original model.

5.3 Overfitting Level of the Target Model

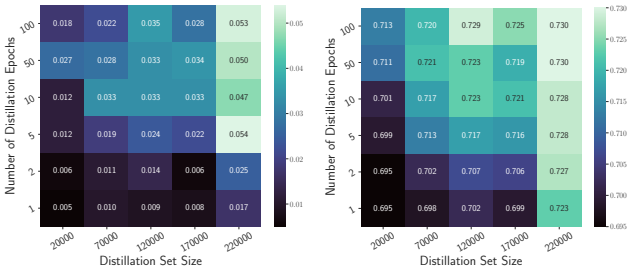
There is a general consensus that the success of MIAs is relevant to the overfitting level of the target model [37, 40]. Here we represent the overfitting level by the training and testing accuracy gap and control it by varying the training set size. Specifically, we vary the size of $\mathcal{D}^t_{train}$, $\mathcal{D}^t_{test}$, $\mathcal{D}^s_{train}$ and $\mathcal{D}^s_{test}$ from 10000 to 30000, while fixing the size of the knowledge distillation dataset $\mathcal{D}^k$ as 150000.

As can be seen from Table 8, as expected, a larger training set size incurs a lower training testing accuracy gap, indicating a lower



(a) CIFAR-10 (b) CINIC-10 (c) CIFAR-100 (d) GTSRB

Figure 8: The impact of number of knowledge distillation epochs on TPR at 0.1% FPR (top) and balanced accuracy (bottom).



(a) TPR at 0.1% FPR (b) Accuracy

Figure 9: The impact of both knowledge distillation set size and number of distillation epochs on attack success rate for ResNet-56 trained on CINIC-10.

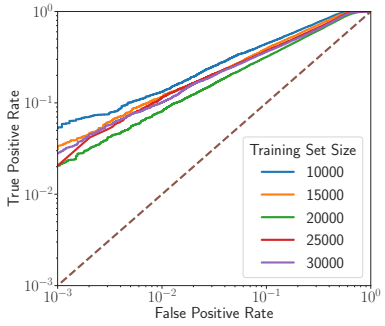


Figure 10: ROC curves for ResNet-56 trained on CINIC-10 with different training set size.

overfitting level. Figure 10 evaluates the corresponding attack performance, and we can observe that the overfitting problem does make the target model more vulnerable to our attack. However, even when the size of the training set is increased to 30000 and the model is well-generalized, our attack can still secure a good attack performance (2.8% in terms of TPR at 0.1% FPR), which remains much better than other attack baselines that are mounted on more overfitted target models with 10000 training samples (cf. Table 2).

5.4 Disjoint Datasets

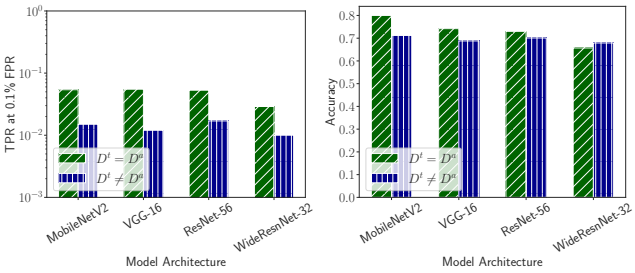
In this section, we will relax the assumption that the adversary has the auxiliary dataset from the same distribution as the training dataset of the target model. We compare two distribution settings:

- $\mathbb{D}^t = \mathbb{D}^a$, which means the dataset for training the target model, shadow model, and distilled models (target and shadow) are from the same distribution, i.e., the CINIC-10 dataset.
- $\mathbb{D}^t \neq \mathbb{D}^a$, which means the dataset for training the target model dataset and the auxiliary dataset are from different distributions. Specifically, the target model is trained on the CIFAR-10 portion of CINIC-10 but the adversary only accesses the ImageNet portion as the auxiliary dataset.

Figure 11 shows that a distribution shift between the training dataset of the target model and the auxiliary dataset will indeed decrease the attack performance in most cases. This can be explained from two aspects. On the one hand, the shadow dataset is different from the target dataset, which may lead to different functionality of the shadow model and target model. On the other hand, the knowledge distillation dataset is also different from the target dataset,

Table 8: The impact of the overfitting level of target model for ResNet-56 trained on CINIC-10.

Training set size	Training testing acc gap	Attack Acc	Attack TPR at 0.1% FPR
10000	0.339	0.730	5.3%
15000	0.300	0.708	3.2%
20000	0.249	0.659	2.3%
25000	0.234	0.685	2.1%
30000	0.214	0.674	2.8%



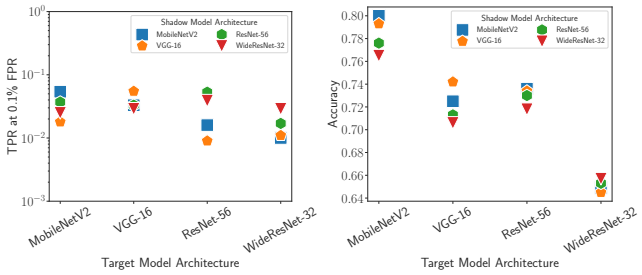
(a) TPR at 0.1% FPR **(b) Accuracy**

Figure 11: The impact of data distribution shift between the target model training set and auxiliary dataset for different models trained on CINIC-10.

which may cause the distilled loss trajectory to behave more different from the actual one. As a result, less membership information can be extracted to train the attack model.

5.5 Model Architecture

After validating the impact of dataset distribution shift, here we focus on another assumption on the knowledge of the adversary about the architecture of the target model. We vary the architectures of the target model, shadow model, and distilled models while keeping the architectures of the shadow model and distilled models the same since both of them are under the full control of the adversary locally. As can be seen from Figure 12, our attack performs the best when the shadow model and distilled models have the same architecture as the target model. In addition, using models from the kindred family of model architectures (e.g., ResNet-56 and WideResNet-32) will lead to a similar attack performance compared to using exactly the same architecture. When the architectures are totally different, although the attack performance will decrease, the result is still better than those achieved by other baselines using the same architecture for the target model and shadow model (or reference model), as reported in Section 4. One possible reason could be that although the target model uses a different architecture, the same architecture of the distilled target and shadow models still achieve a close enough loss trajectory. Overall, the performance of our attack in the above harder settings with relaxed assumptions remains better than that of other attacks achieved in easier settings (as shown in Table 2).



(a) TPR at 0.1% FPR **(b) Accuracy**

Figure 12: The impact of the architecture differences between the target model and local (shadow and distilled) models trained on CINIC-10.

6 DISCUSSION

In this section, we discuss in more detail the characteristics of our attack. We first analyze the different roles of the distilled loss trajectory and the loss from the target model in our attack. We then provide evidence that it is indeed better to use the loss trajectory from the distilled shadow models rather than directly using that from the actual training process of the shadow model.

Distilled Loss Trajectory. Here we show the impact of the distilled loss trajectory to the attack performance. Comparing the result of $Loss_n$ and $Loss_1$ in Table 9, using the distilled loss trajectory can achieve more than 10× TPR at 0.1% FPR than solely using the loss from the last distilled model. In addition, our original attack can also have a considerable improvement over the improved variant of the attack in [51], where we concatenate the loss from the original model and the last distilled model together. A more comprehensive picture of this observation is shown by the ROC curves in Figure 13, where our attack achieves consistently best TPR across the whole range of FPR.

Loss from Original Models. Apart from the *distilled loss trajectory*, our attack also concatenates the loss from the original models (the target model and the trained shadow model). We argue that it is due to the difference between the functionality distillation and membership information distillation. More concretely, although the classification accuracy of the last distilled model can be similar to or even better than the original model as illustrated in Table 7, it omits some information about membership, thus there is still membership information that can be extracted from the loss of the original models. As can be seen from Table 9, when concatenating the loss from the original model, our attack is substantially improved, e.g., TPR at 0.1% FPR increases from 1.7% to 5.3%. Similarly, if we leave out the loss from the original model for the last distilled loss used in [51], the attack is substantially degraded, e.g., TPR at 0.1% FPR decreases from 1.2% to 0.1%.

Distilled Shadow Models. As mentioned before, we are able to obtain the actual loss trajectory from the shadow model as it is trained locally, but we still use the distilled one. Table 10 and Figure 14 support this specific design by showing that using the distilled loss trajectory can indeed lead to better attack performance than directly using the actual loss trajectory. For example, the

Table 9: The impact of incorporating the loss from the target model ($Loss_t$) into the loss trajectory ($Loss_n$ for using losses from all distilled models, and $Loss_1$ for using only the last one). The model is ResNet-56 trained on CINIC-10.

Metric	$Loss_1$ [51]	$Loss_1$+$Loss_t$	$Loss_n$	$Loss_n$+$Loss_t$ (ours)
TPR at 0.1% FPR	0.1%	1.2%	1.7%	**5.3%**
Accuracy	0.603	0.726	0.633	**0.730**

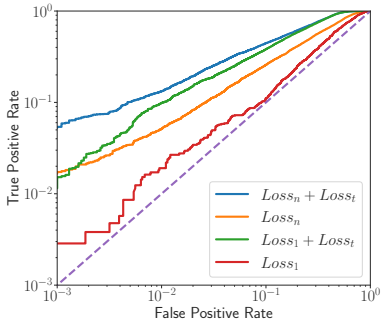


Figure 13: The attack performance showing the effect of each component in our attack, the name of the four methods are the same as in Table 9.

distilled loss trajectory yields about 2× higher TPR at 0.1% FPR than the actual loss trajectory.

This observation is understandable because directly using the actual loss trajectory makes the loss trajectory not well aligned with the distilled loss trajectory from the target model. This worse alignment is due to the fact that the *distilled loss trajectory* could be similar to the actual one yet not the same. For instance, when the distillation dataset is larger than the shadow dataset, the distilled model needs fewer epochs to reach a similar classification accuracy as the original shadow model. Thus if we use the actual loss trajectory to train the attack model, the different patterns in the *distilled loss trajectory* from the target model will incur a dissatisfied membership prediction.

7 RELATED WORK

7.1 Membership Inference Attacks

Currently, membership inference attack (MIA) has gaining attentions for quantifying the privacy risks of machine learning models [16, 18, 19, 26, 27, 31, 37, 40, 43, 52]. Shokri et al. [40] propose the first membership inference attack in black-box settings. They train multiple shadow models to mimic the behavior of the target model and use the posteriors obtained from these shadow models to train multiple attack models. Salem et al. [37] later relax the assumptions made in [40], and only train one shadow model without the knowledge of architecture and data distribution used in target model. Song et al. [43] introduce a metric-based attack without training any attack model. Similarly, Yeom et al. [52] assume that the adversary knows the target model's training dataset distribution and size, and conducts membership inference by relying on the samples' loss.

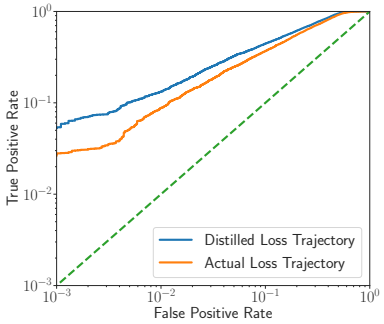


Figure 14: ROC curves of our attack with distilled loss trajectory vs. actual loss trajectory for the shadow model.

Table 10: Attack performance of our method with distilled loss trajectory vs. actual loss trajectory for the shadow model.

Metric	Actual	Distilled
TPR at 0.1% FPR	2.8%	**5.3%**
Acc	0.693	**0.730**
AUC	0.794	**0.819**

Recently, more studies begin to focus on addressing one major problem shared by most MIAs, which is the high false-positive rate. Ye et al. [51] design a model-dependent and sample-dependent MIA via knowledge distillation, which trains multiple distilled models to approximate the samples from the retrained model distribution. Sablayrolles et al. [36] introduce a calibration term, computed by the loss from both the models trained with and without the target sample, in order to calibrate the loss from the target model. Similarly, Watson et al. [48] view the loss from reference models (trained without the target sample) as the sample's hardness, and a smaller difference between the loss from the target model and the sample's hardness implies that the target sample is more likely to be a non-member. Carlini et al. [4] go a step further from [36] to develop a Likelihood Ratio Attack. By taking advantage of the logit scaling, Gaussian likelihood, and multiple queries, this attack can achieve high TPR at low FPR.

7.2 Knowledge Distillation

The notion of transferring knowledge from larger models (teacher models) to smaller ones (student models) emerges quite early. After forming a framework under the name of knowledge distillation (KD), this line of work is extended by either finding new approaches for KD or applying KD in different domains. For the former direction, Romero et al. [35] use additional linear projection to train a thinner and deeper student model. Zagoruykoet et al. [55] adopt attention mechanism to advance the performance of knowledge distillation. Yim et al. [53] propose a new method that adds additional losses to enhance the performance but also speeds up the optimization process. Xu et al. [49] utilize conditional adversarial networks to learn a loss function for KD. Regarding the applications of knowledge distillation, many studies have explored KD in

other domains, such as semi-supervised learning [45], sequence modeling [23] and multi-modal learning [14].

Our work is more related to a specific direction of knowledge distillation, that is self-distillation [13]. In this direction, the teacher model and student model have identical model architectures, and the distillation is used to improve the performance of the student over the teacher. However, there is still a core difference between this direction and our work, that is we adopt the knowledge distillation to extract the membership information represented by the loss trajectory, but care less about the general performance of the model.

8 CONCLUSION

In this paper, we take the first step to exploit the information from the training process of the target model to conduct membership inference. We demonstrate that the *loss trajectory*, i.e., losses of the sample evaluated on the target model at its different training epochs, can be used to represent such membership information. Specifically, we propose a new attack method, call TrajectoryMIA, that leverages knowledge distillation to extract the loss trajectory information of the target model with only black-box access. Our extensive experiments demonstrate the state-of-the-art performance of TrajectoryMIA, especially on a practically meaningful metric that measures the true-positive rate at a low false-positive rate. We also show the general advantage of our attack over existing methods for separate groups of target samples that have different loss values. Additional analyses provide evidence on the importance of each component of our TrajectoryMIA for its attack success. For future work, one promising direction could be exploring more fine-grained modeling of loss trajectories beyond simply using the whole trajectory as the input feature.

ACKNOWLEDGMENTS

We thank all anonymous reviewers for their constructive comments and our shepherd Simon Oya for helping improving our paper. This work is partially funded by the Helmholtz Association within the project "Trustworthy Federated Data Analytics" (TFDA) (funding number ZT-I-OO1 4).

REFERENCES

[1] https://www.cs.toronto.edu/~kriz/cifar.html.
[2] http://benchmark.ini.rub.de/?section=gtsrb.
[3] Martin Abadi, Andy Chu, Ian Goodfellow, Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 308–318. ACM, 2016.
[4] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership Inference Attacks From First Principles. *CoRR abs/2112.03570*, 2021.
[5] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *USENIX Security Symposium (USENIX Security)*, pages 267–284. USENIX, 2019.
[6] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting Training Data from Large Language Models. *CoRR abs/2012.07805*, 2020.
[7] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 343–362. ACM, 2020.
[8] Christopher A. Choquette Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. Label-Only Membership Inference Attacks. In *International Conference on Machine Learning (ICML)*, pages 1964–1974. PMLR, 2021.
[9] Luke Nicholas Darlow, Elliot J. Crowley, Antreas Antoniou, and Amos J. Storkey. CINIC-10 is not ImageNet or CIFAR-10. *CoRR abs/1810.03505*, 2018.
[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.
[11] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography Conference (TCC)*, pages 265–284. Springer, 2006.
[12] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1322–1333. ACM, 2015.
[13] Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born-Again Neural Networks. In *International Conference on Machine Learning (ICML)*, pages 1602–1611. PMLR, 2018.
[14] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross Modal Distillation for Supervision Transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2827–2836. IEEE, 2016.
[15] Guy Hacohen, Leshem Choshen, and Daphna Weinshall. Let's Agree to Agree: Neural Networks Share Classification Order on Real Datasets. In *International Conference on Machine Learning (ICML)*, pages 3950–3960. PMLR, 2020.
[16] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. LOGAN: Evaluating Privacy Leakage of Generative Models Using Generative Adversarial Networks. *Privacy Enhancing Technologies Symposium*, 2019.
[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.
[18] Xinlei He, Zheng Li, Weilin Xu, Cory Cornelius, and Yang Zhang. Membership-Doctor: Comprehensive Assessment of Membership Inference Against Machine Learning Models. *CoRR abs/2208.10445*, 2022.
[19] Xinlei He, Rui Wen, Yixin Wu, Michael Backes, Yun Shen, and Yang Zhang. Node-Level Membership Inference Attacks Against Graph Neural Networks. *CoRR abs/2102.05429*, 2021.
[20] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte Carlo and Reconstruction Membership Inference Attacks against Generative Models. *Privacy Enhancing Technologies Symposium*, 2019.
[21] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the Knowledge in a Neural Network. *CoRR abs/1503.02531*, 2015.
[22] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 259–274. ACM, 2019.
[23] Yoon Kim and Alexander M. Rush. Sequence-Level Knowledge Distillation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1317–1327. ACL, 2016.
[24] Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer. Thieves on Sesame Street! Model Extraction of BERT-based APIs. In *International Conference on Learning Representations (ICLR)*, 2020.
[25] Mathias Lécuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified Robustness to Adversarial Examples with Differential Privacy. In *IEEE Symposium on Security and Privacy (S&P)*, pages 656–672. IEEE, 2019.
[26] Klas Leino and Matt Fredrikson. Stolen Memories: Leveraging Model Memorization for Calibrated White-Box Membership Inference. In *USENIX Security Symposium (USENIX Security)*, pages 1605–1622. USENIX, 2020.
[27] Zheng Li, Yiyong Liu, Xinlei He, Ning Yu, Michael Backes, and Yang Zhang. Auditing Membership Leakages of Multi-Exit Networks. *CoRR abs/2208.11180*, 2022.
[28] Zheng Li and Yang Zhang. Membership Leakage in Label-Only Exposures. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 880–895. ACM, 2021.
[29] Yiyong Liu, Zhengyu Zhao, Michael Backes, and Yang Zhang. Membership Inference Attacks by Exploiting Loss Trajectory. *CoRR abs/2208.14933*, 2022.
[30] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting Unintended Feature Leakage in Collaborative Learning. In *IEEE Symposium on Security and Privacy (S&P)*, pages 497–512. IEEE, 2019.
[31] Milad Nasr, Reza Shokri, and Amir 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)*, pages 1021–1035. IEEE, 2019.
[32] Nicolas Papernot, Martin Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *International Conference on Learning Representations (ICLR)*, 2017.

[33] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing Neural Networks by Penalizing Confident Output Distributions. In *International Conference on Learning Representations (ICLR)*, 2017.

[34] Shadi Rahimian, Tribhuvanesh Orekondy, and Mario Fritz. Sampling Attacks: Amplification of Membership Inference Attacks by Repeated Queries. *CoRR abs/2009.00395*, 2020.

[35] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for Thin Deep Nets. In *International Conference on Learning Representations (ICLR)*, 2015.

[36] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs Black-box: Bayes Optimal Strategies for Membership Inference. In *International Conference on Machine Learning (ICML)*, pages 5558–5567. PMLR, 2019.

[37] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2019.

[38] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520. IEEE, 2018.

[39] Virat Shejwalkar and Amir Houmansadr. Membership Privacy for Machine Learning Models Through Knowledge Transfer. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 9549–9557. AAAI, 2021.

[40] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *IEEE Symposium on Security and Privacy (S&P)*, pages 3–18. IEEE, 2017.

[41] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[42] Congzheng Song and Vitaly Shmatikov. Auditing Data Provenance in Text-Generation Models. In *ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pages 196–206. ACM, 2019.

[43] Liwei Song and Prateek Mittal. Systematic Evaluation of Privacy Risks of Machine Learning Models. In *USENIX Security Symposium (USENIX Security)*. USENIX, 2021.

[44] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 2014.

[45] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1195–1204. NIPS, 2017.

[46] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An Empirical Study of Example Forgetting during Deep Neural Network Learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[47] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing Machine Learning Models via Prediction APIs. In *USENIX Security Symposium (USENIX Security)*, pages 601–618. USENIX, 2016.

[48] Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. On the Importance of Difficulty Calibration in Membership Inference Attacks. *CoRR abs/2111.08440*, 2021.

[49] Zheng Xu, Yen-Chang Hsu, and Jiawei Huang. Training Shallow and Thin Networks for Acceleration via Knowledge Distillation with Conditional Adversarial Networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[50] Dingqi Yang, Daqing Zhang, and Bingqing Qu. Participatory Cultural Mapping Based on Collective Behavior Data in Location-Based Social Networks. *ACM Transactions on Intelligent Systems and Technology*, 2016.

[51] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, and Reza Shokri. Enhanced Membership Inference Attacks against Machine Learning Models. *CoRR abs/2111.09679*, 2021.

[52] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In *IEEE Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

[53] Junho Yim, Donggyu Joo, Ji-Hoon Bae, and Junmo Kim. A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7130–7138. IEEE, 2017.

[54] Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 2016.

[55] Sergey Zagoruyko and Nikos Komodakis. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In *International Conference on Learning Representations (ICLR)*, 2017.

[56] Minxing Zhang, Zhaochun Ren, Zihan Wang, Pengjie Ren, Zhumin Chen, Pengfei Hu, and Yang Zhang. Membership Inference Attacks Against Recommender Systems. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 864–879. ACM, 2021.