Cascading and Proxy Membership Inference Attacks

Yuntao Du, Jiacheng Li, Yuetian Chen, Kaiyuan Zhang, Zhizhen Yuan, Hanshen Xiao, Bruno Ribeiro, Ninghui Li
Purdue University
{ytdu, li2829, yuetian, zhan4057, yuan462, hsxiao, ribeirob, ninghui}@purdue.edu

arXiv:2507.21412v1 [cs.CR] 29 Jul 2025

***Abstract*—A Membership Inference Attack (MIA) assesses how much a trained machine learning model reveals about its training data by determining whether specific query instances were included in the dataset. We classify existing MIAs into adaptive or non-adaptive, depending on whether the adversary is allowed to train shadow models on membership queries. In the adaptive setting, where the adversary can train shadow models after accessing query instances, we highlight the importance of exploiting membership dependencies between instances and propose an attack-agnostic framework called Cascading Membership Inference Attack (CMIA), which incorporates membership dependencies via conditional shadow training to boost membership inference performance.**

In the non-adaptive setting, where the adversary is restricted to training shadow models before obtaining membership queries, we introduce Proxy Membership Inference Attack (PMIA). PMIA employs a proxy selection strategy that identifies samples with similar behaviors to the query instance and uses their behaviors in shadow models to perform a membership posterior odds test for membership inference. We provide theoretical analyses for both attacks, and extensive experimental results demonstrate that CMIA and PMIA substantially outperform existing MIAs in both settings, particularly in the low false-positive regime, which is crucial for evaluating privacy risks[1].

I. Introduction

Machine learning (ML) has advanced rapidly over the past decade, with models such as neural networks increasingly being trained on sensitive datasets. This raises a critical need to ensure that these trained models are privacy-preserving. Membership inference attacks (MIAs) [1] quantify the degree to which a model leaks information by predicting whether some instances were part of its training set. Closely connected with Differential Privacy (DP) [2], MIAs have become a widely adopted approach for empirical privacy auditing of ML models [3], [4], and serve as crucial components for more sophisticated attacks [5], [6].

Generally speaking, MIAs exploit discrepancies in a target model's behavior between training samples (*i.e.,* members) and non-training samples (*i.e.,* non-members). These discrepancies are often captured through signals such as output losses. To effectively leverage these signals for membership inference, a prevalent approach is the *shadow training* technique [1], which involves training multiple shadow models on datasets drawn from the same distribution as the target model's training data. These shadow models provide insights into how a model's output on an instance varies depending on whether the instance is included in the training dataset. Shadow-based MIAs [7], [8] have achieved state-of-the-art performance, particularly when evaluated using the increasingly recommended metric [9]: True-Positive Rate (TPR) at a low False-Positive Rate (FPR).

We classify MIAs into two categories: adaptive and non-adaptive (see Section II for a formal definition). In the adaptive setting, which has been extensively studied in prior work [7], [9]–[11], the adversary can train shadow models *after* knowing the query instances to be inferred. For each query instance, the adversary can train shadow *in* models (trained with the instance) and shadow *out* models (trained without the instance), enabling the computation of a membership score that takes advantage of both shadow in and shadow out models. In the non-adaptive setting, which has garnered attention in recent studies [8], [12]–[14], the adversary can only train shadow models *before* learning the query instances. As a result, only shadow *out* models are available to compute the membership scores for query instances.

In this paper, we study MIAs in both settings. For the adaptive setting, we highlight the importance of exploiting membership dependencies between instances for attack, which is largely overlooked by existing adaptive MIAs [7], [9]. Building on the theoretical analysis of joint membership estimation using Gibbs sampling, we introduce Cascading Membership Inference Attack (CMIA), an attack-agnostic framework designed to enhance attack performance through conditional shadow training. CMIA operates by iteratively running a base shadow-based attack to identify highly probable samples, then using their inferred membership to train new shadow models to improve the inference of remaining instances.

In the non-adaptive setting, we propose Proxy Membership Inference Attack (PMIA), which approximates the likelihood that a query instance in the training set using the behaviors of the shadow models' training data as proxies. We also introduce several methods for selecting proxy data at different granularities, including global, class, and instance levels.

We compare the proposed attacks to a wide range of state-of-the-art MIAs and conduct extensive experiments across six benchmark datasets and five model architectures. The experimental results demonstrate that CMIA consistently improves attack performance across all evaluated attack algorithms and

[1] Our code is available at https://github.com/zealscott/MIA.

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9894372-8-3
https://dx.doi.org/10.14722/ndss.2026.250661
www.ndss-symposium.org

datasets, including those previously considered difficult to attack. For instance, CMIA improves LiRA [9] by more than $\mathbf{5}\times$ in the true-positive rate at a 0.001% false-positive rate on MNIST, showing a significant performance boost. Additionally, PMIA significantly outperforms all existing MIAs in the non-adaptive setting while maintaining high efficiency. We also conduct comprehensive ablation studies to evaluate the impact of various components in CMIA and PMIA and assess the effectiveness of existing defenses against the proposed attacks. In summary, we make the following contributions:

- We provide a new formulation of the MIA game in Section II, which allows the clear differentiation of the adaptive and non-adaptive settings for MIA.
- In the adaptive setting, we highlight the importance of modeling membership dependencies and present a theoretical analysis of joint membership estimation using approximate Gibbs sampling. We also introduce CMIA, an attack-agnostic framework that enhances attack performance via conditional shadow training.
- In the non-adaptive setting, we introduce PMIA, a new attack that approximates the membership posterior odds ratio test using proxy data.
- We conduct extensive experiments and demonstrate that CMIA and PMIA consistently outperform all state-of-the-art MIAs in various attack scenarios.

Organization. The rest of this paper is organized as follows. Section II introduces the definitions and threat models for membership inference attacks. We then describe the proposed adaptive attack (*i.e.,* CMIA) in Section III and the non-adaptive attack (*i.e.,* PMIA) in Section IV. Section V presents the experimental results for both attacks. Related work is discussed in Section VI, and the paper concludes in Section VII.

II. Problem Definition and Threat Models

The goal of a membership inference attack (MIA) [1] is to determine whether some instances were included in the training data of a given trained model f_θ. In this paper, we consider the model to be a neural network classifier $f_\theta : \mathcal{X} \to \Delta^m$, where f_θ is a learned function that maps an input data sample $x \in \mathcal{X}$ to a probability distribution over m classes, where Δ^m denotes the m-dimensional simplex. Given a dataset D, we use $f_\theta \leftarrow \mathcal{T}(D)$ to denote that the neural network f_θ, parameterized by weights θ, is learned by applying the training algorithm $\mathcal{T}$ on the training set D.

In the literature [9], [15], [16], membership inference has been defined via a security game in which the adversary is asked to determine the membership of a single instance. However, there exists a disconnection between this definition and the experiments, which train shadow models and determine memberships for a set of query instances (referred to as the membership query set). More importantly, using such a single-instance game definition for MIA, it is difficult to clearly differentiate adaptive versus non-adaptive MIA settings, as will be detailed later. To address these challenges, we define membership inference through the following security game:

Definition 1 (Membership Inference Security Game)**.** *The following game is between a challenger and an adversary that both have access to a data distribution $\mathbb{D}$:*

1) *The challenger samples a training dataset $D \sim \mathbb{D}$, trains a target model $f_\theta \leftarrow \mathcal{T}(D)$ on the dataset D, and grants the adversary query access to the model f_θ.*
2) *The challenger selects two sets: a subset $D_\mathrm{a} \subseteq D$ and a set D_b sampled from $\mathbb{D}$. These two sets are combined to create a query set: $D_\mathrm{query} = D_\mathrm{a} \cup D_\mathrm{b}$, which the challenger then sends to the adversary.*
3) *The adversary responds with a set $D_\mathrm{g} \subseteq D_\mathrm{query}$, which represents that the adversary guesses that instances in D_g are used when training f_θ, and instances in $D_\mathrm{query} \setminus D_\mathrm{g}$ are not used when training f_θ.*

In this paper, we focus on membership inference attacks in black-box scenarios, where the adversary is granted oracle access to the target model f_θ, but is not given the model parameters. That is, the adversary can obtain the output softmax probabilities for any input instance x.

State-of-the-art MIAs [7]–[9], [14] leverage shadow models [1] to analyze how the model's outputs depend on whether specific instances are used in training or not. We assume the adversary constructs a dataset and samples subsets from it to train the shadow models. The above game allows the adversary to access the data distribution $\mathbb{D}$, which they can sample when constructing the dataset. Depending on when the shadow models are trained, we consider two threat models.

Adaptive Setting. In this setting, the adversary is allowed to train shadow models *after* receiving the query set D_query (*i.e.,* after step 2 in Definition 1). Thus, the adversary samples D' from $\mathbb{D}$ and combines it with D_query to create a dataset for shadow training, *i.e.,* $D^\text{adapt}_\text{adv} \leftarrow D' \cup D_\text{query}$. This enables the adversary to train shadow *in* models and shadow *out* models for each query instance, and exploit the behavioral discrepancies to mount an attack.

This setting models several attack situations. One situation is that the adversary is willing to spend substantial computation to train shadow models for specific query instances [7], [9]. Another situation is that the target model's training set is drawn from a dataset that the adversary also possesses, and the adversary tries to learn which specific instances are used [10], [11]. MIAs in the adaptive setting are also advocated for empirical privacy auditing [3], [4] of ML models, where the goal is to assess the extent of worst-case privacy leakage.

Non-Adaptive Setting. In contrast, under the non-adaptive setting, the adversary is only allowed to train shadow models *before* the adversary learns the query set D_query (*i.e.,* before step 2 Definition 1). Thus, the adversary constructs $D^\text{non-adapt}_\text{adv}$ only by sampling from $\mathbb{D}$, *i.e.,* $D^\text{non-adapt}_\text{adv} \sim \mathbb{D}$. For most practical classification tasks, the sizes of $D^\text{non-adapt}_\text{adv}$ and D_query are relatively small compared to the entire data distribution $\mathbb{D}$, meaning the probability of each instance belonging to both datasets is quite low. As a result, the adversary can only observe a model's behavior when the query instance is not part of the training set, thus having access only to shadow *out* models but not shadow *in* models.

This setting models the situation where the adversary is given a sequence of membership queries and needs to answer them without paying the cost of retraining shadow models for each query. Recent studies [8], [12]–[14] focus on this setting, as it presents a more efficient attack scenario.

Discussion. It is worth noting that some studies [8], [9], [14] refer to the adaptive and non-adaptive settings as "online" and "offline", respectively. This terminology may lead to confusion, as the offline setting models queries being processed without training new shadow models, which is similar in spirit to "online algorithms" [17]. Thus, we adopt the terms "adaptive" and "non-adaptive", which are more aligned with well-established concepts in the security and privacy domain (*e.g.,* the (adaptive) chosen-ciphertext attacks [18], [19]). We use these terms throughout this paper for clarity.

Differences from Previous MIA Game. Notably, our membership inference security game, as defined in Definition 1, goes beyond existing studies [9], [15], [16] by using a query set rather than a single instance. This modification offers a few advantages. Firstly, it more accurately reflects the experimental methodologies employed in existing research, where performance evaluations (*e.g.,* TPR at low FPR) are typically conducted using a query set, rather than isolating assessments to each instance. Secondly, this query-set-based approach enables a clear distinction between adaptive and non-adaptive settings, achieved by controlling the timing of shadow model training relative to the receipt of the query set.

Missed Opportunities of Existing MIAs. Our paper is motivated by the observations that existing MIAs did not fully take advantage of the available information in both adaptive and non-adaptive settings.

- *Membership Dependencies in the Adaptive Setting.* xisting adaptive MIAs [7], [9], [10] typically predict the membership of each query instance *independently*. They thus fail to take advantage of the conditional membership dependencies among instances, leading to suboptimal performance.
- *Proxy Shadowing in the Non-Adaptive Setting.* In the non-adaptive setting, the adversary can construct only shadow out models, and lacks the knowledge of models trained with a specific query instance behave on that instance. However, we observe that it is possible to find instances that are similar to the query instances and observe their behavior.

III. Cascading Membership Inference Attack

In this section, we describe the methodology of Cascading Membership Inference Attacks (CMIA), which is an attack-agnostic framework designed for the adaptive setting. We first present a novel attack paradigm, denoted *Joint MIA*, which seeks to *jointly* estimate the membership of *all* query instances. We then provide a detailed description of CMIA.

A. Theoretical Intuition of Joint MIA

We assume that instances in the size-n query set D_{query} are ordered in a canonical order, *i.e.,* $D_{\text{query}} = \{(x_i, y_i) \mid 1 \leq i \leq n\}$. Given target model f_θ trained using dataset D, we define the (unobserved) vector of membership variables as $\mathbf{M} = (M_1, M_2, \ldots, M_n)$, where each M_i is a Bernoulli variable, $M_i := \mathbb{1}[(x_i, y_i) \in D]$. We use $o_\theta = \{(x_i, f_\theta(x_i)) | \forall (x_i, y_i) \in D_{\text{query}}\}$ to denote the output softmax probabilities that the adversary can obtain from D_{query}.

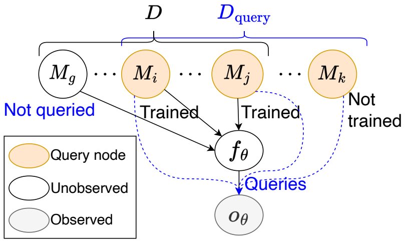


Fig. 1: Statistical dependencies of joint MIA shows that conditioning on output o_θ creates a collider dependency between the membership indicators M_i and M_j, *i.e.,* $M_i \not\perp\!\!\!\perp M_j \mid o_\theta$.

Existing MIAs [7], [9], [14] typically treat each query instance (x_i, y_i) in isolation, computing individual membership probabilities $\Pr(M_i = 1|o_\theta)$ independently. At first glance, this approach appears reasonable, as the data instances in the training set D could be assumed to be independently and identically distributed (i.i.d.), implying that the membership indicators M_i and M_j are marginally independent, *i.e.,* $M_i \perp\!\!\!\perp M_j$, for all $i \neq j$. However, this independence assumption is violated when conditioning on the model's output o_θ over the entire query dataset D_{query}. In this conditional setting, the membership indicators, although independent marginally, become dependent due to the collider effect induced by o_θ, as illustrated in Figure 1. Specifically, conditioning on o_θ renders M_i and M_j conditionally dependent, i.e., $M_i \not\perp\!\!\!\perp M_j \mid o_\theta$.

The practical significance of this dependence becomes evident in the adaptive setting, where the adversary can train shadow models after knowing D_{query}, which overlaps with the target's training dataset D. Existing adaptive MIAs [1], [7], [9]–[11] **do not explore this conditional joint dependence**. By leveraging the joint membership distributions rather than relying on marginal probabilities, an adversary can mount more effective membership inference attacks. We first formulate the definition of Joint MIA via Gibbs sampling.

Joint MIA with Gibbs Sampling (accurate but prohibitively expensive). In order to sample from the joint distribution $\Pr(\mathbf{M}|o_\theta)$, it requires capturing the dependencies between the membership indicators introduced by conditioning on o_θ. The traditional Gibbs sampling procedure updates each variable sequentially [20], with each update based on the conditional distribution of the variable given the current states of all other variables. This iterative process can be expressed formally at any iteration $t \geq 0$ as:

$$M_i^{(t+1)} \sim \Pr(M_i \mid \mathbf{M}_{-i}^{(t+1,t)}, o_\theta), \ \forall i, \qquad (1)$$

where $\mathbf{M}_{-i}^{(t+1,t)} = M_1^{(t+1)}, \ldots, M_{i-1}^{(t+1)}, M_{i+1}^{(t)}, \ldots, M_n^{(t)}$.

We now provide a theoretical result to show that it is possible to perform this joint sampling and converge to the true target stationary distribution $\Pr(\mathbf{M}|o_\theta)$, as well as the almost sure convergence of any metric of success of the attack.

Theorem 1 (Convergence of Joint MIA Gibbs Sampling)**.** *Let $\mathbf{M} = (M_1, M_2, \ldots, M_n)$ be the vector of membership statuses, where $M_i = \mathbb{1}[(x_i, y_i) \in D]$. Let $\pi(\mathbf{M}|o_\theta) = \Pr(\mathbf{M}|o_\theta)$ be the target joint distribution of membership statuses conditioned on the model output o_θ. Consider the Gibbs sampling procedure that iteratively samples at step $t \geq 1$:*

$$M_i^{(t+1)} \sim \Pr(M_i|\mathbf{M}_{-i}^{(t+1,t)}, o_\theta),\ \forall (x_i, y_i) \in D_{query},$$

where $\mathbf{M}_{-i}^{(t+1,t)} = M_1^{(t+1)}, \ldots, M_{i-1}^{(t+1)}, M_{i+1}^{(t)}, \ldots, M_n^{(t)}$. Then:

1) *The sequence of states $M^{(t)}$ forms a Markov chain with stationary distribution $\pi(\mathbf{M}|o_\theta) = \Pr(\mathbf{M}|o_\theta)$.*
2) *For any measurable MIA performance metric L with $\mathbb{E}_\pi[|L(\mathbf{M}, D)|] < \infty$, the sequence*

$$S_T = \frac{1}{T} \sum_{t=1}^{T} L(\mathbf{M}^{(t)}, D)$$

converges almost surely to $\mathbb{E}_\pi[L(\mathbf{M}, D)]$ as $T \to \infty$.

The proof of Theorem 1 is provided in Appendix A and is based on the stationarity of Markov chains and the convergence properties of martingale theory.

Theorem 1 provides a rigorous foundation for the proposed Gibbs sampling approach, confirming its suitability for Joint MIA by establishing both the convergence to the target distribution and the consistency of performance metrics. However, using this approach is computationally prohibitive, as convergence usually requires many iterations. The challenge now lies in devising a practical and effective method for performing a joint membership inference attack (Joint MIA), which benefits from jointly inferring all instances in D_{query} without paying the high cost of full Gibbs sampling.

B. CMIA*: A Fast Approximation of Joint MIA*

We now introduce CMIA, a heuristic that significantly speeds up the Gibbs sampling procedure by approximating it and performing just a single joint sampling step.

Notably, an important hyperparameter of the Gibbs sampling process is the ordering in which variables in D_{query} are sampled, referred to as the *scan order*. The ordering in Equation (1) is known as the systematic scan (also known as deterministic or sequential scan). Interestingly, this ordering can be modified, with significant implications for the convergence rate of Gibbs sampling [21], [22]. The most common strategy is actually random ordering, which shuffles D_{query} after each iteration. In adaptive Gibbs sampling [23], the scan order is adaptively determined.

Our approach (CMIA) also uses adaptive ordering, but we focus exclusively on identifying a single, (highly likely) joint membership sample. This allows for optimizations that would be incorrect or inefficient in standard Gibbs sampling contexts, where the goal is to explore the entire distribution, including less likely observations. More specifically, a key innovation in CMIA is the dynamic reordering of the instances (x_i, y_i) within D_{query} to *prioritize* those with higher membership probabilities (instances we will call them **anchors**). For the initial sample, the instances are rearranged to satisfy $\Pr(M_1 = 1|M_{-1}^{(0,0)}, o_\theta) \geq \Pr(M_i = 1|M_{-i}^{(0,0)}, o_\theta), \forall i$, with ties resolved arbitrarily (in practice, ties and near-ties are sampled jointly as shown in Section III-C). To illustrate the effect of this reordering, consider a scenario where $\Pr(M_1 = 1|M_{-1}^{(0,0)}, o_\theta) \approx 1$, indicating near certainty that (x_1, y_1) is a member of D. In this context, the reordered D_{query} sets the stage for a cascading effect, where subsequent sampling decisions are informed by this initial high confidence. By leveraging this certainty, the process streamlines the exploration of the sample space, focusing on the most promising regions first.

The sampling process proceeds iteratively, with each step refining the ordering of the remaining instances to prioritize those with the highest membership probabilities. Specifically, for the i-th sample, the instances are reordered such that $\Pr(M_i = 1|M_{-i}^{(1,0)}, o_\theta) \geq \Pr(M_j = 1|M_{-j}^{(1,0)}, o_\theta)$ for $i < j$, where $i = 2, \ldots, n$ denotes the current iteration. Ties are again resolved arbitrarily. At each iteration, the process checks if the membership probability for the next instance exceeds a predetermined threshold, in which case the instance is deemed a member, and its membership status is set to $M_i = 1$. Otherwise, the process terminates and the remaining instances are deemed non-members.

This greedy approach identifies a subset of highly probable members while avoiding unnecessary computations for less likely candidates. Empirically, as we see in Section V-B, joint MIA achieves significantly higher attack performance compared to independent inference of prior work.

C. Implementation Details of CMIA

In this section, we describe the implementation details of CMIA. CMIA approximates the sampling procedure outlined in the previous section, and estimates conditional membership probabilities by performing a base shadow-based MIA using conditional shadow models. Specifically, in each iteration, CMIA identifies multiple highly probable instances (*i.e.,* anchors), by conducting membership inference using a base shadow-based MIA. It then utilizes their inferred membership to generate conditional shadow models, which enhance the inference of the remaining instances. This process can be repeated until no additional anchors can be reliably identified. We begin by providing a definition for shadow-based MIA, which will serve as the base attack in CMIA.

Shadow-based MIAs. Shadow-based MIAs [1], [9] train shadow models to imitate the target model's behavior. Formally, shadow-based MIAs are defined as follows:

Definition 2 (Shadow-based MIA)**.** *Let $D_{\text{adv}}^{\text{adapt}}$ be the adversary's dataset in the adaptive setting. A shadow-based MIA constructs multiple shadow dataset-model pairs $\mathcal{P}_{\text{shadow}} = \{(D_{\text{shadow}}^j, f_{\text{shadow}}^j)\}$, where each pair consists of a shadow dataset D_{shadow}^j sampled from $D_{\text{adv}}^{\text{adapt}}$ and the corresponding shadow model f_{shadow}^j trained on the dataset, i.e., $f_{\text{shadow}}^j \leftarrow \mathcal{T}(D_{\text{shadow}}^j)$. The attack model $\mathcal{M}$ (e.g., LiRA [9]) then utilizes these shadow dataset-model pairs to generate a continuous*

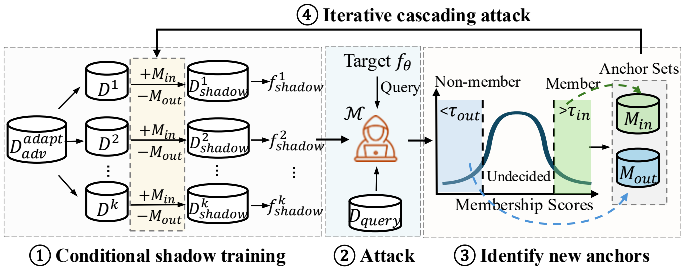


Fig. 2: Demonstration of CMIA. The adversary ① constructs conditional shadow datasets by sampling from $D_{\text{adv}}^{\text{adapt}}$ and incorporating the membership of anchors, ② performs the base attack $\mathcal{M}$, ③ uses the computed membership scores to identify new anchors ($M_{\text{in}}/M_{\text{out}}$), ④ repeats the above processes to enhance the inference of the remaining instances.

membership score to predict the membership of instance (x_i, y_i) *for the target model* f_θ*:*

$$s(x_i, y_i) = \mathcal{M}\left(f_\theta, (x_i, y_i), \mathcal{P}_{\text{shadow}}\right).$$

Note that different MIAs may employ different training algorithms $\mathcal{T}$ to train shadow models. In CMIA, we use these shadow-based MIAs by only modifying the input shadow dataset, while both the training algorithm $\mathcal{T}$ and the attack model $\mathcal{M}$ remain unchanged.

Framework Overview. The pipeline of CMIA is illustrated in Figure 2 and outlined in Algorithm 1. First, the maximum number of cascading iterations K is set. In each iteration k, the adversary constructs shadow datasets by sampling from $D_{\text{adv}}^{\text{adapt}}$ (defined in Section II) and incorporating with anchor sets ($M_{\text{in}}/M_{\text{out}}$). Specifically, samples from M_{in}^{k-1} are included while those from M_{out}^{k-1} are excluded from the sampled set to construct the conditional shadow dataset (line 7). This procedure repeats N times to generate N conditioned shadow dataset-model pairs. Then, the base attack $\mathcal{M}$ is executed using these dataset-model pairs to compute membership scores for all instances in D_{query} (lines 11-15). New anchors are selected based on their membership scores relative to learned decision thresholds (lines 19–23, detailed below). The iterations stop when either the maximum number of iterations is reached or the number of new anchors falls below δ (line 24). Finally, shadow datasets/models from all iterations are used to perform the final attack on all query instances (lines 29-33).

Thresholds Selection. Anchors (*i.e.,* highly probable instances) are selected by comparing membership scores against two decision thresholds: τ_{in} for members and τ_{out} for non-members. To determine these thresholds, for each iteration, we perform the attack on a ground-truthed shadow model and analyze the relationship between membership scores and actual membership. Specifically, we randomly select a shadow model and treat it as the target, then apply the base attack on this model to compute membership scores for instances in $D_{\text{adv}}^{\text{adapt}}$. Since the adversary knows the ground truth about the members (*i.e.,* the training set) of the selected shadow model, we set the decision thresholds by ordering the membership scores and setting: (i) τ_{in} as the highest score among non-members, and (ii) τ_{out} as the 10th lowest score among members. In other words, τ_{in} is chosen to avoid false positives when predicting members, while τ_{out} is selected to tolerate less than 10 false negatives when predicting non-members. This distinction is made because the adversary typically focuses on reliably identifying members rather than non-members [9]. We analyze the impact of thresholds in Section V-D.

Algorithm 1 **Cascading Membership Inference Attack.**

```
Require: Target model $f_\theta$, adversary's dataset $D_{\text{adv}}^{\text{adapt}}$, training
    algorithm $\mathcal{T}$, base MIA algorithm $\mathcal{M}$, membership query
    set $D_{\text{query}}$, number of iterations $K$, stopping criterion $\delta$
 1: $M_{\text{in}}^0 \leftarrow \{\},\ M_{\text{out}}^0 \leftarrow \{\}$                    ▷ initialize anchor sets
 2: for $k = 1$ to $K$ do
 3:     # Step 1: Train conditional shadow models
 4:     $\mathcal{P}_{\text{shadow}}^k \leftarrow \{\}$          ▷ initialize shadow dataset-model set
 5:     for $N$ times do
 6:         $D_{\text{tmp}} \sim D_{\text{adv}}^{\text{adapt}}$                              ▷ sample a dataset
 7:         $D_{\text{shadow}} \leftarrow (D_{\text{tmp}} \setminus M_{\text{out}}^{k-1}) \cup M_{\text{in}}^{k-1}$
 8:         $\mathcal{P}_{\text{shadow}}^k \leftarrow \mathcal{P}_{\text{shadow}}^k \cup \{(D_{\text{shadow}}, \mathcal{T}(D_{\text{shadow}}))\}$
 9:     end for
10:     # Step 2: Attack with conditional shadow models
11:     $\mathcal{S}^k \leftarrow \{\}$                    ▷ initialize membership scores set
12:     for each $(x, y) \in D_{\text{query}}$ do
13:         # compute membership score, defined in Definition 2
14:         $\mathcal{S}^k \leftarrow \mathcal{S}^k \cup \{\mathcal{M}\left(f_\theta, (x, y), \mathcal{P}_{\text{shadow}}^k\right)\}$
15:     end for
16:     # Step 3: Identify anchor samples using membership scores
17:     $M_{\text{in}}^k = M_{\text{in}}^{k-1},\ M_{\text{out}}^k = M_{\text{out}}^{k-1}$
18:     # thresholds selection is detailed in Section III-C
19:     $\tau_{\text{in}}^k, \tau_{\text{out}}^k \leftarrow \texttt{SelectThresholds}(\mathcal{M}, \mathcal{P}_{\text{shadow}}^k, D_{\text{adv}}^{\text{adapt}})$
20:     for each $s^k(x, y) \in \mathcal{S}^k$ do
21:         if $s^k(x, y) > \tau_{\text{in}}^k$ then $M_{\text{in}}^k \leftarrow M_{\text{in}}^k \cup \{(x, y)\}$
22:         else if $s^k(x, y) < \tau_{\text{out}}^k$ then $M_{\text{out}}^k \leftarrow M_{\text{out}}^k \cup \{(x, y)\}$
23:     end for
24:     if $|M_{\text{in}}^t| - |M_{\text{in}}^{k-1}| < \delta$ and $|M_{\text{out}}^t| - |M_{\text{out}}^{k-1}| < \delta$ then
25:         break
26:     end if
27: end for
28: # Perform final attack using all shadow dataset-model pairs
29: $\mathcal{P}_{\text{shadow}} \leftarrow \bigcup_k \mathcal{P}_{\text{shadow}}^k,\ \mathcal{S} \leftarrow \{\}$
30: for each $(x, y) \in D_{\text{query}}$ do
31:     $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathcal{M}\left(f_\theta, (x, y), \mathcal{P}_{\text{shadow}}\right)\}$
32: end for
33: return $\mathcal{S}$
```

Implementation Details. We set the number of the cascading iterations $K = 10$ for all experiments, as we observe that all evaluated attacks exhibit significant improvements within only a few iterations (as shown in Section V-B). The stopping criterion δ is set to 30, meaning that the iteration halts if fewer than 30 new anchors can be selected. Although the threshold selection procedure can be repeated by using different ground-truthed shadow models to obtain an average threshold estimate, we find that it remains robust even with a single randomly chosen shadow model. All above hyperparameters are kept fixed across all evaluated attacks and datasets to demonstrate the robustness of the framework.

IV. Proxy Membership Inference Attack

While CMIA shows impressive performance, it requires knowledge of query instances, and cannot be applied in the non-adaptive setting. In this section, we propose Proxy Membership Inference Attack (PMIA), a new non-adaptive MIA that can respond to arbitrary membership queries without training additional shadow models. We begin by presenting the theoretical intuition behind PMIA and then describe the attack procedure in detail.

A. Theoretical Intuition of Marginal MIA

In the non-adaptive setting, the adversary is given a sequence of membership queries and needs to answer them *independently*, *i.e.,* for instance $(x_i, y_i) \in D_{\text{query}}$ we predict its membership status: $M_i | e_\theta$, where $e_\theta \leftarrow \mathcal{Q}(f_\theta)$ denotes the information the adversary can observe from the model f_θ with the available data. We refer to this type of attack as **Marginal MIA**, as our predictions are only based on $\Pr(M_i | e_\theta)$ regardless of the membership status of other instances in $D_{\text{query}} \setminus \{(x_i, y_i)\}$. We follow [24] and frame the marginal MIA as a Bayesian binary hypothesis testing problem. In the black-box scenario we study, e_θ is the output of f_θ. We now show how to calculate the membership posterior odds ratio test for marginal MIA.

Theorem 2 (A Membership Posterior Odds Test for Marginal MIA)**.** *Let $\mathcal{S}^+_{(x_i,y_i)}$ and $\mathcal{S}^-_{(x_i,y_i)}$ be the sets of all subsets in the support domain $\mathbb{D}$ that include or exclude the query instance (x_i, y_i), respectively. Let $M_i = \mathbb{1}[(x_i, y_i) \in D]$ and $\mathcal{L}(D, e_\theta) = \Pr(\mathcal{Q}(\mathcal{T}(D)) = e_\theta)$ denote the likelihood function, where $\mathcal{T}(D)$ represents the model is trained with dataset D. Given the adversary's observation e_θ, the Bayesian posterior odds test for marginal MIA is defined as:*

$$\mathcal{A}_{\text{odds}}(x_i, y_i) = \mathbb{1}\left[\frac{\Pr\left(M_i = 1 \mid \mathcal{Q}(f_\theta) = e_\theta\right)}{\Pr\left(M_i = 0 \mid \mathcal{Q}(f_\theta) = e_\theta\right)} > 1\right],$$

which can be obtained from

$$\mathcal{A}_{\text{odds}}(x_i, y_i) = \mathbb{1}\left[\frac{\mathbb{E}_{D' \sim \mathcal{S}^+_{(x_i,y_i)}} \mathcal{L}(D', e_\theta)}{\mathbb{E}_{D' \sim \mathcal{S}^-_{(x_i,y_i)}} \mathcal{L}(D', e_\theta)} > \frac{\Pr(M_i = 0)}{\Pr(M_i = 1)}\right]. \tag{2}$$

A detailed proof based on Bayes' rule is provided in Appendix B. Note that the expectations in the formula are taken with respect to datasets sampled uniformly at random from $\mathcal{S}^+_{(x_i,y_i)}$ and $\mathcal{S}^-_{(x_i,y_i)}$, which are the countable collections of all possible datasets that include or exclude the query instance (x, y). The theorem establishes that the membership posterior odds test $\mathcal{A}_{\text{odds}}$ for an adversary involves comparing a special likelihood ratio to a threshold derived from prior probabilities.

Connecting with Evaluation Metrics. For most practical classifiers, the training set D is relative small compared to the entire data distribution, so the prior probability $\Pr(M_i = 1)$ is low for most data points (x_i, y_i), making the threshold $\frac{\Pr(M_i=0)}{\Pr(M_i=1)}$ fairly large. Thus, the adversary must have a significantly high likelihood ratio to infer membership correctly. This statistical analysis helps explain why recent studies [9], [14] emphasize evaluating MIAs on the FPR at low FPR.

Connecting with SOTA MIAs. We observe that most state-of-the-art MIAs, e.g., [9], [14], adhere to the principles of the posterior odds ratio test outlined in Equation (2). For instance, LiRA [9] estimates the likelihood ratio for a query instance by utilizing the (scaled) losses of its shadow *in* and shadow *out* models. However, in the non-adaptive setting, computing the likelihood ratio becomes challenging because the adversary only has access to the shadow *out* models for query instances. As a result, LiRA adopts a one-sided hypothesis test and estimates only the likelihood that the instance is not in the training set. This deviation leads to suboptimal performance, as evidenced by recent studies [8], [13].

PMIA**: Better Approximating the Posterior Odds Test.** Motivated by these insights, we propose a new attack, PMIA, which better aligns with the membership posterior odds test presented above while avoiding the need to train additional shadow models for queries. The key idea is to approximate the likelihood that a query instance is in the training set using the behaviors of the shadow models' training data as proxies. We also explore various proxy-finding strategies at different levels of granularity to further improve the attack's performance.

B. Attack Method

Likelihood Estimators. Building on the analysis in the previous section, our goal is to estimate the likelihood ratio of the query instance for inference. We first follow LiRA [9] and define the output of query instance (x, y) on model f as:

$$\phi(f(x)_y) = \log\left(\frac{f(x)_y}{1 - f(x)_y}\right),$$

where $f(x)_y$ denotes the softmax probability (*aka.* confidence score) of the model f on the instance (x, y). This scaling transformation stabilizes the distribution and, empirically, allows it to be well-approximated by a normal distribution. We then adopt the same likelihood estimators used in LiRA [9]:

$$\mathbb{E}_{D' \sim \mathcal{S}^+_{(x,y)}} \tilde{\mathcal{L}}(D', e_\theta) = p\left(\phi(f_\theta(x)_y) \mid \mathcal{N}(\mu_{\text{in}}, \sigma^2_{\text{in}})\right),$$
$$\mathbb{E}_{D' \sim \mathcal{S}^-_{(x,y)}} \tilde{\mathcal{L}}(D', e_\theta) = p\left(\phi(f_\theta(x)_y) \mid \mathcal{N}(\mu_{\text{out}}, \sigma^2_{\text{out}})\right),$$

where $p\left(\phi(f_\theta(x)_y) \mid \mathcal{N}(\mu, \sigma^2)\right)$ is the probability density function over $\phi(f_\theta(x)_y)$ under a normal distribution with mean μ and variance σ^2. In the adaptive setting, these two normal distributions can be estimated by training shadow *in* models and shadow *out* models on the query instance (x, y) and retrieving the scaled confidence scores from shadow models. However, in the non-adaptive setting, the adversary is not allowed to train shadow models on query instances, meaning that the distribution $\mathcal{N}(\mu_{\text{in}}, \sigma^2_{\text{in}})$ is unavailable. Therefore, we aim to find proxy data D_{proxy} from the adversary's dataset $D^{\text{non-adapt}}_{\text{adv}}$ and use their behaviors (*i.e.,* $\mathcal{N}(\tilde{\mu}_{\text{in}}, \tilde{\sigma}_{\text{in}})$) to approximate the behavior of a query instance $\mathcal{N}$(*i.e.,* $(\mu_{\text{in}}, \sigma^2_{\text{in}})$).

Overview. The workflow of PMIA is outlined in Algorithm 2, consisting of two phases: the prepare and inference phases. In the prepare phase, we follow the standard shadow training technique [1], [9] and train N shadow models by randomly sampling shadow datasets from $D_{\text{adv}}^{\text{non-adapt}}$ (lines 3–7). In the inference phase, we first collect the distribution of confidence scores for the target instance (x, y) when it is *not* in the model's training set (lines 11–13). Next, we select proxy data from $D_{\text{adv}}^{\text{non-adapt}}$ for the query instance and gather their confidence score distributions when they are *in* the shadow models' training sets (lines 15–22). Finally, we estimate the mean and variance for both collected confidence distributions (lines 23–24), query the target model f_θ on (x, y) to output a parametric likelihood ratio $\tilde{\Lambda}$ as the membership score.

Finding Proxy Data. The effectiveness of PMIA relies on selecting appropriate proxy data from $D_{\text{adv}}^{\text{non-adapt}}$ and using their confidence distribution to approximate the likelihood of the query instance being in the training set. We propose three strategies for selecting these proxies at different granularities:

- *Global-level*. We use all instances in the attacker's dataset as the proxy set D_{proxy} and collect their confidence scores when they are part of the shadow model's training set. This produces a global *in* confidence distribution that is used to compute the likelihood for all query instances.
- *Class-level*. Since the adversary knows the label of the query instance, we select samples from $D_{\text{adv}}^{\text{non-adapt}}$ that belong to the same class as the query instance to form D_{proxy}.
- *Instance-level*. We retrieve the top-10 similar samples from $D_{\text{adv}}^{\text{non-adapt}}$ using some similarity measurements (*e.g.,* cosine similarity in embedding space), and use them as D_{proxy}.

We implement all the above strategies and analyze their effectiveness in Section V-D. Despite their simplicity, we find that using these proxy data achieves state-of-the-art performance. We thus leave the exploration of more advanced proxy selection strategies for future work.

Implementation Details. We follow LiRA and train 256 shadow models during the prepare phase. We also apply standard data augmentations to fit n-dimensional spherical Gaussians, which are collected by querying the shadow models n times per sample (n is set as 9 for all experiments). The final membership score, $\tilde{\Lambda}$, is then computed using the likelihood ratio between two multivariate normal distributions. When using the instance-level strategy to select proxy data, we adopt different approaches depending on the data modality. For image datasets, we first embed each image into a 512-dimensional vector using a pretrained CLIP [25] encoder. We then use the Faiss library [26] to retrieve the top-10 most similar samples from the attacker's dataset, based on cosine similarity in this embedding space. For non-image datasets where a powerful pretrained encoder is unavailable, we use Wasserstein distance on the raw data for proxy selection.

Algorithm 2 **Proxy Membership Inference Attack.**

```
Require: Target model $f_\theta$, adversary's dataset $D_{\text{adv}}^{\text{non-adapt}}$,
    training algorithm $\mathcal{T}$, query instance $(x, y) \in D_{\text{query}}$
 1: # Prepare Phase: Shadow Model Training
 2: $\mathcal{D}_{\text{shadow}} \leftarrow \{\}$, $\mathcal{F}_{\text{shadow}} \leftarrow \{\}$
 3: for $N$ times do
 4:     $D_{\text{shadow}} \sim D_{\text{adv}}^{\text{non-adapt}}$                ▷ sample a shadow dataset
 5:     $\mathcal{D}_{\text{shadow}} \leftarrow \mathcal{D}_{\text{shadow}} \cup \{D_{\text{shadow}}\}$
 6:     $\mathcal{F}_{\text{shadow}} \leftarrow \mathcal{F}_{\text{shadow}} \cup \{\mathcal{T}(D_{\text{shadow}})\}$
 7: end for
 8: # Inference Phase: Query on $(x, y)$
 9: $\text{confs}_{\text{in}} \leftarrow \{\}$, $\text{confs}_{\text{out}} \leftarrow \{\}$
10: # collect out confidence scores
11: for each $f_{\text{shadow}} \in \mathcal{F}_{\text{shadow}}$ do
12:     $\text{confs}_{\text{out}} \leftarrow \text{confs}_{\text{out}} \cup \{\phi(f_{\text{shadow}}(x)_y)\}$
13: end for
14: # collect in confidence scores via proxies, see Section IV-B
15: $D_{\text{proxy}} \leftarrow \texttt{FindProxy}(D_{\text{adv}}^{\text{non-adapt}}, (x, y))$ ▷ find a proxy set
16: for each $D_{\text{shadow}}^i \in \mathcal{D}_{\text{shadow}}$ do
17:     for each $(u, v) \in D_{\text{proxy}}$ do
18:         if $(u, v) \in D_{\text{shadow}}^i$ then
19:             $\text{confs}_{\text{in}} \leftarrow \text{confs}_{\text{in}} \cup \{\phi(f_{\text{shadow}}^i(u)_v)\}$
20:         end if
21:     end for
22: end for
23: Compute mean $\tilde{\mu}_{\text{in}}$, and variance $\tilde{\sigma}_{\text{in}}^2$ from $\text{confs}_{\text{in}}$
24: Compute mean $\mu_{\text{out}}$, and variance $\sigma_{\text{out}}^2$ from $\text{confs}_{\text{out}}$
25: $\text{conf}_{\text{obs}} \leftarrow \phi(f_\theta(x)_y)$                ▷ query target model
26: return $\tilde{\Lambda} = \dfrac{p(\text{conf}_{\text{obs}} \mid \mathcal{N}(\tilde{\mu}_{\text{in}}, \tilde{\sigma}_{\text{in}}^2))}{p(\text{conf}_{\text{obs}} \mid \mathcal{N}(\mu_{\text{out}}, \sigma_{\text{out}}^2))}$
```

V. Evaluation

We conduct a comprehensive evaluation of our adaptive (CMIA) and non-adaptive (PMIA) attacks across various datasets and attack settings. We first describe the experimental setup in Section V-A. Next, we evaluate the attack performance in adaptive and non-adaptive settings in Section V-B and Section V-C, respectively. We also analyze the impact of attack components in Section V-D and perform additional analyses and assess their performance against defenses in Section V-E.

A. Experimental Setup

Datasets. We select four image benchmark datasets (*i.e.,* MNIST [27], Fashion-MNIST [28], CIFAR-10 [29], and CIFAR-100 [29]) for our main experiments. The results on two non-image datasets that are commonly used in MIAs (*i.e.,* Purchase and Texas [1]) are reported in Section V-E. A detailed dataset description is provided in Appendix C.

Network Architecture. We consider four widely used neural network architectures for image classification: ResNet50 [30], VGG16 [31], DenseNet121 [32], and MobileNetV2 [33]. To reduce overfitting, we follow the settings of previous studies [7], [9] when training the target models. Specifically, we use the SGD algorithm with a learning rate of 0.1, momentum set to 0.9, and weight decay [34] set to 5×10^{-4}. Additionally, we employ a cosine learning rate schedule [35] for optimization, and apply data augmentation [36] during the training of the target models. The train and validation accuracy of the target model in both settings are reported in Appendix D.

Attack Baselines. We compare our attacks against a broad range of state-of-the-art MIAs in our experiments:

- *Calibration* [37] employs a technique called difficulty calibration, which adjusts the loss of the query instance by calibrating its loss on shadow models as membership scores.
- *Attack-R* [38] compare the loss of query instance on the target model with its loss on shadow out models. The membership score is based on the ratio where the loss on the target model is smaller than the loss on the shadow models.
- *LiRA* [9] exploits behavioral discrepancies in the query instance by modeling its loss distribution as a Gaussian estimate and uses a likelihood ratio test to compute the membership score.
- *Canary* [7] enhances LiRA by using adversarial learning to optimize the query instance for inference.
- *RMIA* [14] calculates the membership score based on the success ratio of pairwise likelihood ratio tests between the query instance and random instances from the population.
- *RAPID* [8] combines both the loss and the calibrated loss [37] to train a neural network for membership inference.

LiRA and Canary use different strategies for adaptive and non-adaptive settings. We distinguish between their versions based on the experimental context. In the non-adaptive setting, in addition to these attacks, we also compare PMIA with the following attacks that do not rely on shadow training:

- *LOSS* [15] uses the loss of the query instance as the score.
- *Entropy* [39] leverages a modified prediction entropy estimation as the membership score.

Evaluation Procedures. For the adaptive setting, we follow [7], [9] split each dataset into two disjoint subsets: D_1 and D_2. We randomly sample 50% of D_1 to train the target model, and D_2 is used as the validation set to prevent overfitting during training. The adversary is provided with the same D_1 to prepare their attack, and the query set D_{query} is also set to be the same (*i.e.,* $D_{\text{adv}}^{\text{adapt}} = D_{\text{query}} = D_1$). For the non-adaptive setting, we follow [8], [13] and divide each dataset into two disjoint subsets, one as the query set and the other as the adversary's dataset, ensuring that the adversary cannot access queries when training shadow models. Each of these subsets is further split into training and validation sets to train the target model and shadow models. The details about the data split are in Table XII and Table XIII.

Evaluation Metrics. Following previous studies [9], [14], we focus on evaluating the attack performance on the low false positive rate regime. Specifically, we use the following evaluation metrics for our experiments:

- *TPR@0.001%FPR.* It directly reflects the extent of privacy leakage of the model by allowing only one (or a few) false positives to compute the true positive rate.
- *TPR@0.1%FPR.* This is a relaxed version of the previous metric, allowing more false-positive samples for evaluation.
- *Balanced Accuracy.* This metric measures how often an attack correctly predicts membership (average case).

Attack Setup. We use the same techniques as LiRA [9] to train shadow models, ensuring that each instance appears in exactly half of the training sets for the shadow models. The performance of shadow-based MIAs depends on the number of shadow models used. Through experimentation, we find that LiRA and Canary benefit from training a larger number (*e.g.,* 256) of shadow models, while other methods plateau after a smaller number of shadow models. Thus, we identify the optimal number of shadow models for each attack method individually. For CMIA, we train the same number of shadow models as the base attack for each cascading iteration. For PMIA, we train a fixed set of 256 shadow models. The hyperparameters are set according to the original implementations, and we report the average metrics over five runs with different random seeds as the final results.

B. Evaluation of CMIA

Main Results. We apply CMIA with six SOTA MIAs across four image datasets. As shown in Table I, CMIA consistently boosts all attacks across datasets, particularly in the low false-positive rate regime. For example, CMIA enhances the performance of LiRA by 5× at a false-positive rate of 0.001% and boosts the Calibration attack by 7× at the same FPR on MNIST. These results represent a substantial advancement, as such datasets were previously considered difficult to attack. In addition, we find that CMIA can elevate previously weak attacks to a level comparable to strong ones. For instance, CMIA boosts the Calibration attack from 0.19% to 0.55% at a false-positive rate of 0.1% on MNIST, nearly matching the performance of LiRA (*i.e.,* 0.56% at the same FPR). The Receiver Operating Characteristic (ROC) curve [40] and results on other model architectures are reported in Appendix E.

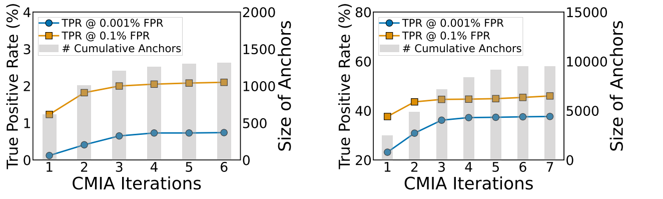
Improvement per Cascading Iteration. We analyze the performance of each iteration in CMIA. Specifically, we use LiRA as the base attack and examine its performance across each iteration, as shown in Figure 3. The results demonstrate a gradual improvement in performance with each additional iteration, while the number of identified anchors increases until reaching the stopping criterion. Notably, the most significant improvements occur during the first few iterations. For example, adding just one cascading iteration increases the true-positive rate from 0.12% to 0.41% on MNIST, with similar improvements observed for CIFAR-100. This pattern likely emerges because identifying anchors is easier in early iterations, whereas it is increasingly difficult to reliably predict memberships for remaining instances.

Efficiency Analysis. While CMIA achieves impressive attack performance, Table III shows that it incurs a significantly higher computational cost than the base attack (*i.e.,* , LiRA). To improve this trade-off, we explore three strategies:

- *Execute with Fewer Iterations.* Most of CMIA's performance gains occur in the early iterations. Thus, reducing the cascading iterations can still achieve strong performance.
- *Optimize the Computational Budget.* The intermediate shadow models of CMIA are primarily used for identifying

TABLE I: Performance comparison of *adaptive* attacks using CMIA on ResNet50 trained on four image datasets (*i.e.,* MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100). For each method, results are provided for both the original attack and the enhanced version using CMIA. %Imp. denotes the relative improvement of CMIA over the baseline. The best result is in bold.

Method		**TPR @ 0.001% FPR**				**TPR @ 0.1% FPR**				**Balanced Accuracy**			
		MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100
Calibration	Base	0.01%	0.52%	0.28%	1.48%	0.19%	2.23%	1.02%	5.51%	51.05%	54.21%	54.62%	61.18%
	CMIA	**0.08%**	**1.24%**	**0.59%**	**3.81%**	**0.55%**	**4.72%**	**3.65%**	**8.52%**	**52.21%**	**55.37%**	**56.13%**	**64.09%**
	%Imp.	700.00%	138.46%	110.71%	157.43%	189.47%	111.66%	257.84%	54.63%	2.27%	2.14%	2.76%	4.76%
Attack-R	Base	0.00%	0.00%	0.21%	1.40%	0.10%	0.00%	1.30%	4.82%	52.15%	57.83%	54.26%	62.13%
	CMIA	**0.00%**	**0.00%**	**0.45%**	**2.01%**	**0.37%**	**0.00%**	**1.95%**	**6.04%**	**52.95%**	**58.48%**	**55.48%**	**63.93%**
	%Imp.	-	-	114.29%	43.57%	270.00%	-	50.00%	25.31%	1.53%	1.12%	2.25%	2.90%
LiRA	Base	0.12%	2.72%	2.64%	23.15%	1.23%	6.28%	8.45%	37.62%	51.26%	58.28%	62.52%	82.05%
	CMIA	**0.77%**	**4.42%**	**3.86%**	**36.74%**	**2.10%**	**8.34%**	**9.71%**	**45.37%**	**52.67%**	**60.91%**	**63.83%**	**84.89%**
	%Imp.	541.67%	62.50%	46.21%	58.70%	70.73%	32.80%	14.91%	20.60%	2.75%	4.51%	2.10%	3.46%
Canary	Base	0.15%	2.95%	2.36%	25.78%	1.28%	6.65%	8.12%	38.25%	53.76%	58.94%	62.60%	83.11%
	CMIA	**0.84%**	**4.73%**	**3.61%**	**37.85%**	**2.48%**	**8.47%**	**9.02%**	**45.96%**	**55.60%**	**61.07%**	**63.81%**	**84.72%**
	%Imp.	460.00%	60.34%	52.97%	46.82%	93.75%	27.37%	11.08%	20.16%	3.42%	3.61%	1.93%	1.94%
RMIA	Base	0.21%	2.05%	1.43%	10.72%	0.96%	4.71%	5.24%	30.13%	52.99%	58.16%	62.05%	80.64%
	CMIA	**0.52%**	**3.56%**	**2.05%**	**14.67%**	**1.62%**	**5.81%**	**6.05%**	**37.51%**	**53.51%**	**60.90%**	**62.49%**	**82.53%**
	%Imp.	147.62%	73.66%	43.36%	36.85%	68.75%	23.35%	15.46%	24.49%	0.98%	4.71%	0.71%	2.34%
RAPID	Base	0.23%	1.31%	0.56%	9.83%	0.79%	3.44%	3.12%	21.69%	52.44%	58.40%	59.58%	75.83%
	CMIA	**0.48%**	**2.45%**	**0.94%**	**11.83%**	**1.24%**	**4.73%**	**4.75%**	**25.90%**	**52.97%**	**58.51%**	**59.77%**	**78.52%**
	%Imp.	108.70%	87.02%	67.86%	20.35%	56.96%	37.50%	52.24%	19.41%	1.01%	0.19%	0.32%	3.55%



(a) MNIST

(b) CIFAR-100

Fig. 3: The impact of number of cascading iterations in CMIA.

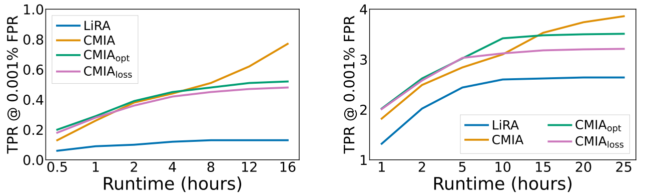
anchors rather than full membership inference. Thus, we can reduce the number of shadow models trained per iteration and increase the iterations to take advantage of the cascading framework (denoted as CMIA$_{\text{opt}}$).

- *Lightweight Attacks for Anchor Selection.* A lightweight attack (*i.e.,* LOSS attack [15]) can be used for anchor selection, followed by a stronger attack (*i.e.,* LiRA) for inference (denoted as CMIA$_{\text{loss}}$).

We evaluate these strategies on MNIST and CIFAR-10 using a cluster with 8 A100 GPUs. As shown in Figure 4, LiRA saturates after 2 and 5 hours (corresponding to 256 shadow models), whereas our proposed methods achieve higher accuracy within the same budget. Notably, CMIA$_{\text{opt}}$ surpasses LiRA while training only 64 shadow models per iteration. Further, CMIA$_{\text{loss}}$ proves effective without requiring any additional model training. Overall, these enhancements allow CMIA to offer a better efficiency and effectiveness trade-off.

C. *Evaluation of* PMIA

Main Results. We compare PMIA with SOTA MIAs in the non-adaptive setting. The results for ResNet50 are shown in Table II, while performance on other model architectures and the ROC curves are provided in Appendix F. It is observed that PMIA achieves the best attack performance across all datasets. For example, it achieves a true-positive rate of 5.90% at a false-positive rate of 0.001% on CIFAR-100, which is at least twice as high as the best-performing baseline (*i.e.,* RMIA). Despite using the same likelihood estimator as LiRA, PMIA significantly outperforms it. This improvement stems from PMIA's use of the approximated likelihood ratio for membership prediction, while LiRA offline relies on a one-sided hypothesis.



(a) MNIST

(b) CIFAR-10

Fig. 4: Efficiency analysis of CMIA with LiRA as the base attack. CMIA$_{\text{opt}}$ optimize the computation efficiency by training 64 shadow models per iteration. CMIA$_{\text{loss}}$ utilizes the LOSS attack [15] to identify anchors for cascading attack.

Efficiency Analysis. We analyze its computational cost in two phases: preparation and inference. In the preparation phase, PMIA trains the same number of shadow models (*e.g.,* 256) as LiRA. While this step is quite expensive, it only needs to be performed once. The more critical phase is inference, where the adversary must respond quickly to any queries. In Table VI, we report the total inference time cost for all attacks on MNIST. PMIA responds faster than RMIA and RAPID, demonstrating strong inference efficiency. As a result, the overall runtime of PMIA (prepare and inference) is only marginally greater than that of LiRA, as shown in Table III.

D. *Ablation Study*

Impact of Hyperparameters for CMIA**.** We vary the number of cascading iterations K and stopping criterion δ in CMIA to investigate its impact. The results on CIFAR-10 are shown in Table IV. As illustrated, increasing the number of iterations

TABLE II: Performance comparison of *non-adaptive* attacks on ResNet50 trained on four image datasets. The %Imp. indicates the relative improvement of PMIA compared to the strongest baseline (underlined).

Method	**TPR @ 0.001% FPR**				**TPR @ 0.1% FPR**				**Balanced Accuracy**			
	MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100
LOSS	0.01%	0.01%	0.00%	0.00%	0.08%	0.09%	0.00%	0.00%	52.81%	61.51%	63.35%	78.20%
Entropy	0.01%	0.01%	0.00%	0.00%	0.08%	0.10%	0.00%	0.21%	52.80%	61.16%	63.08%	78.05%
Calibration	0.05%	0.06%	0.08%	1.08%	0.34%	0.45%	1.03%	2.83%	52.51%	55.10%	57.96%	66.10%
Attack-R	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	52.62%	58.47%	63.46%	77.36%
LiRA	0.09%	0.05%	0.03%	0.98%	0.30%	0.67%	0.78%	8.56%	50.54%	53.11%	58.97%	73.25%
Canary	0.11%	0.08%	0.03%	1.78%	0.30%	1.02%	0.77%	7.35%	51.01%	53.79%	58.77%	73.93%
RMIA	0.17%	0.05%	0.41%	2.73%	0.51%	1.25%	2.60%	6.64	52.78%	58.96%	62.72%	77.53%
RAPID	0.09%	0.15%	0.15%	1.16%	0.45%	0.44%	1.34%	3.14%	52.05%	58.42%	61.39%	78.49%
PMIA	**0.31%**	**0.17%**	**1.20%**	**5.90%**	**1.01%**	**2.80%**	**3.29%**	**11.5%**	**52.87%**	**61.56%**	**64.34%**	**80.4%**
%Imp.	82.35%	13.33%	192.68%	116.12%	98.04%	124.00%	26.54%	34.35%	0.11%	0.08%	1.39%	2.43%

TABLE III: Runtime comparison of the proposed methods, measured in hours on a cluster of 8 A100 GPUs. CMIA is measured using LiRA as the base attack.

	MNIST	**FMNIST**	**C-10**	**C-100**	**Purchase**	**Texas**
LiRA	2.25	3.52	5.82	10.20	1.02	1.15
CMIA	13.30	17.53	34.83	71.48	5.14	5.93
PMIA	2.30	3.58	5.85	10.40	1.09	1.18

TABLE IV: Impact of the hyperparamters K and δ in CMIA.

K	δ	**TPR @ 0.001%FPR**	**TPR @ 0.1%FPR**	**Balanced Accuracy**
5	15	3.10%	8.90%	62.71%
5	30	3.18%	8.92%	62.84%
5	60	2.97%	8.65%	62.58%
10	15	**3.88%**	9.67%	63.80%
10	30	3.86%	**9.71%**	**63.83%**
10	60	3.12%	9.03%	62.77%

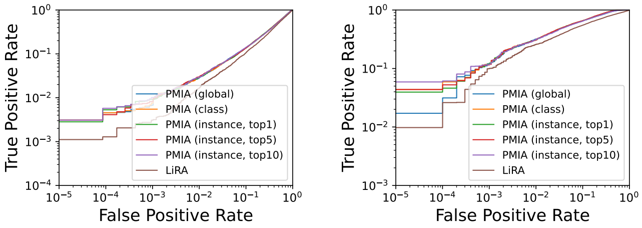
K generally improves performance. A strict stopping criterion (*e.g.,* $\delta = 60$) tends to halt the cascading process prematurely, leading to degraded results. On the other hand, a relaxed criterion (*e.g.,* $\delta = 15$) would increase cascading iterations with marginal gains. Overall, we find that $K = 10$ and $\delta = 30$ provide a good trade-off between effectiveness and efficiency.

Impact of Thresholds Selection for CMIA. We vary the threshold τ_{out} and investigate its impact. Specifically, we rank the membership scores and select the r-th lowest score among the members as τ_{out}, where r presents the tolerance level. We then vary r and report the performance and anchor size for MNIST and CIFAR-100. As shown in Table V, performance initially improves as r increases, but drops as it increases further. This happens because increasing the tolerance allows for more anchors for generating conditional shadow models, but also increases false negatives. The optimal trade-off occurs around $r = 10$, where sufficient anchor samples are selected for shadow training without introducing too many errors. We use the strictest possible selection for τ_{in} (*i.e.,* no false positives are allowed) because allowing false positives would directly degrade the performance on metrics like TPR at low FPR.

Impact of Selecting Proxy Data. In PMIA, we consider three proxy selection strategies at the global, class, and instance levels. Figure 5 illustrates the performance of these proxy selection strategies on the MNIST and CIFAR-100 datasets. All approaches significantly outperform LiRA, highlighting the importance of selecting proxies for attacks. We observe a notable performance improvement when using proxy data from the same class, compared to using the entire adversary's dataset as proxies. In CIFAR-100, performance improves when using similar images as proxy data; however, this improvement is not observed in MNIST. We attribute this discrepancy to the lack of diversity in the MNIST dataset, making the instance-level proxy data less distinct and offering minimal improvement over the class-level proxy. Additionally, we find that instance-level proxying is robust to the number of similar images selected, with comparable performance observed when using the top 1, 5, or 10 most similar images as proxies.

TABLE V: Impact of the threshold parameter r in CMIA. A larger r introduces more false negatives for anchor selections.

r	**# Identified Anchors**		**TPR @ 0.1%FPR**		**TPR @ 0.001%FPR**	
	MNIST	CIFAR-100	MNIST	CIFAR-100	MNIST	CIFAR-100
1	245	1851	0.22%	23.68%	1.23%	38.95%
5	573	3124	0.50%	26.88%	1.82%	41.57%
10	1310	9512	**0.77%**	36.74%	**2.10%**	45.96%
15	1507	13435	0.69%	**38.12%**	1.74%	**48.41%**
20	1863	15941	0.51%	34.31%	1.45%	43.59%
25	2437	17064	0.43%	31.51%	1.20%	41.14%



(a) MNIST

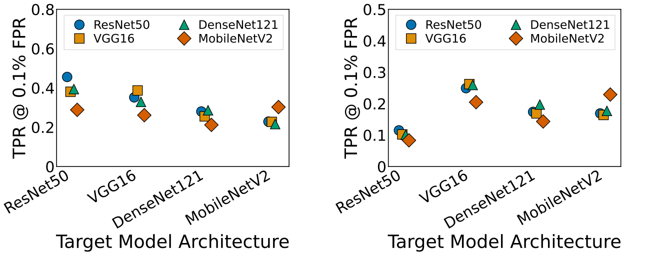
(b) CIFAR-100

Fig. 5: The impact of selecting different proxy data in PMIA. The y-axis range is adjusted to enhance visibility.

Mismatched Model Architecture. We examine how our attack is affected when the attacker is unaware of the exact training procedure used for the target model. Specifically, we explore the impact of varying the target model's architecture on the performance of both CMIA and PMIA attacks. As shown in Figure 6, our attack performs best when the attacker trains shadow models with the same architecture as the target model, and using a different model (*e.g.,* VGG16 instead of ResNet50) has only a minimal effect on the attack's effectiveness. However, we observe a notable decrease in performance when using MobileNetV2. This decrease can be attributed to

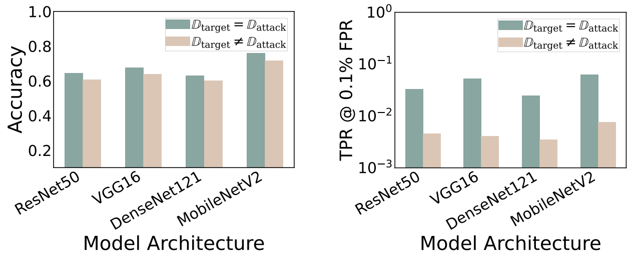
TABLE VI: Membership inference time cost of non-adaptive attacks against a ResNet50 model on MNIST.

Attack Method	LOSS	Entropy	Calibration	Attack-R	LiRA	Canary	RMIA	RAPID	PMIA
Inference Cost/seconds	1.23	2.52	1.85	3.03	10.47	$>400{,}000$	49.5	31.5	**15.8**



(a) CMIA with LiRA as base attack (b) PMIA

Fig. 6: The impact of architecture differences between the target model and the shadow models trained on CIFAR-100.



(a) Balanced Accuracy (b) TPR @ 0.1% FPR

Fig. 7: The impact of distribution shift between the target model training dataset and the attacker's dataset on PMIA.

the inherent differences in the architectures: in MobileNetV2, the number of channels in the feature map increases and then decreases, which contrasts with other model architectures. Similar results have been reported in previous studies [8], [9].

Attack with Distribution Shift. In our experiments so far, we assume that the adversary has access to the same underlying distribution as the target model's training datasets. However, in a real attack, the adversary's data is likely not perfectly aligned with the target's training data. We now explore this more realistic scenario to assess its impact. Specifically, we follow prior work [8], [9] and conduct following experiments:

- $\mathbb{D}_{\text{target}} = \mathbb{D}_{\text{attack}}$. Both the target and shadow models are trained using disjoint subsets of the CIFAR-10 dataset. This follows the non-adaptive setting in our main experiments.
- $\mathbb{D}_{\text{target}} \neq \mathbb{D}_{\text{attack}}$. The target model is trained on a subset of CIFAR-10, while the shadow models use the ImageNet [41] portion of the CINIC-10 [42]. This creates a distribution shift between the target's data and the adversary's data.

We train the same number of shadow models in both settings, and apply the proposed PMIA for attack. Figure 7 shows that the distribution shift between the attacker's and target's training data leads to a noticeable decrease in performance, particularly for TPR at 0.1% FPR. This decrease can be attributed to errors in proxy data selection: when instances in the shadow dataset differ significantly from the query instances, using them as proxies introduces more approximation errors. However, even in this more challenging setting, PMIA still outperforms most baselines on balanced accuracy.

E. Additional Investigations

Attack on Non-image Datasets. We also evaluate the attack performance on two non-image datasets to demonstrate the generality of the proposed attacks. Specifically, we train multilayer perceptrons (MLPs) on the Purchase and Texas datasets [1]. For CMIA, we use LiRA as the base model and apply the cascading framework. For PMIA, we employ the Wasserstein distance to select the top 10 proxy models for each query instance. The attack performance under both adaptive and non-adaptive settings is presented in Table VII.

TABLE VII: Performance comparison on non-image datasets. CMIA employs LiRA as the base attack. CMIA and PMIA outperform state-of-the-art MIAs on both attack settings.

	TPR @ 0.001%FPR		**TPR @ 0.1%FPR**		**Balanced Accuracy**	
	Purchase	Texas	Purchase	Texas	Purchase	Texas
			Adaptive Setting			
LiRA	1.26%	10.28%	11.33%	24.24%	85.95%	90.10%
RMIA	0.41%	5.37%	4.73%	10.63%	84.62%	89.68%
RAPID	0.33%	9.64%	4.16%	15.73%	84.05%	90.05%
CMIA	**2.08%**	**17.34%**	**15.64%**	**27.38%**	**86.36%**	**90.92%**
			Non-Adaptive Setting			
LiRA	0.01%	0.04%	0.06%	0.17%	62.12%	65.03%
RMIA	0.03%	0.12%	0.14%	0.46%	72.36%	80.52%
RAPID	0.03%	0.10%	0.17%	0.49%	73.72%	81.31%
PMIA	**0.05%**	**0.42%**	**2.28%**	**5.72%**	**78.38%**	**87.10%**

As shown, both CMIA and PMIA consistently outperform existing attacks, particularly in TPR at low FPR.

Impact of Join MIA. We experiment to illustrate the impact of joint membership dependence on attack performance. Specifically, we use the MNIST dataset and partition the 60,000 membership queries into the following scenarios: (a) a single set of 60,000 instances, (b) 6 sets with 10,000 instances each, (c) 60 sets, (d) 600 sets, and (e) 60,000 individual queries (*i.e.,* no dependence is exploited). We employ CMIA with LiRA as the base attack and evaluate the overall attack performance under each scenario. As shown in Table IX, the attack performance consistently degrades from (a) to (e), demonstrating the importance of modeling joint membership dependence for achieving strong performance.

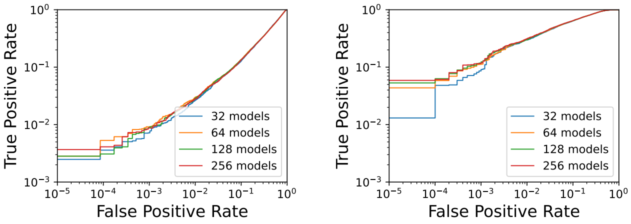
Potential Improvements of CMIA. One potential improvement of CMIA is to adopt a sampling strategy that better aligns with Gibbs sampling. Specifically, for each iteration, we generate conditional shadow datasets by sampling each instance with a probability proportional to its membership score. We implemented this approach (called $\text{CMIA}_{\text{Gibbs}}$) and compared it with CMIA using the same computational budget (*i.e.,* a maximum of 10 iterations). As shown in Table VIII, CMIA consistently outperforms $\text{CMIA}_{\text{Gibbs}}$ across four image datasets. We attribute this to the limited computations: while

TABLE VIII: Performance comparison of CMIA and CMIA$_{\text{Gibbs}}$ (detailed in Section V-E) on a ResNet50 trained on four benchmark datasets. LiRA is used as the base attack for both methods. The maximum number of iterations is set to 10.

	TPR @ 0.001% FPR				**TPR @ 0.1% FPR**				**Balanced Accuracy**			
	MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100
CMIA	0.77%	4.42%	3.86%	36.74%	2.10%	8.34%	9.71%	45.37%	52.67%	60.91%	63.83%	84.89%
CMIA$_{\text{Gibbs}}$	0.35%	3.13%	2.85%	29.52%	1.80%	7.82%	8.98%	40.05%	52.04%	59.58%	63.29%	84.02%

TABLE IX: Impact of joint membership inference on MNIST.

# Queries	**#Instance per query**	**TPR @ 0.001% FPR**	**TPR @ 0.1% FPR**
1	60,000	0.77%	2.10%
6	10,000	0.69%	2.05%
60	1,000	0.45%	1.70%
600	100	0.27%	1.45%
60,000	1	0.12%	1.23%



(a) MNIST (b) CIFAR-100

Fig. 8: The impact of the number of shadow models for PMIA. The y-axis range is adjusted to enhance visibility.

CMIA$_{\text{Gibbs}}$ aligns more closely with the spirit of Gibbs sampling, it requires significantly more iterations to be effective.

Impact of Number of Shadow Models for PMIA. In Figure 8, we vary the number of shadow models from 32 to 256 and find that PMIA is quite robust: the performance drop is less significant compared to LiRA. This robustness stems from using a set of proxy data to approximate the likelihood. By leveraging more data to approximate the confidence score distribution for normal distribution estimation, PMIA remains effective even when trained with fewer shadow models.

Attack Against DP-SGD. Machine learning with differential privacy [43] is an effective defense mechanism against privacy attacks, including MIAs. We follow [8], [9] and assess the effectiveness of the DP-SGD against our attack. Specifically, we fix the clipping norm C as 10 and test CMIA and PMIA on a ResNet50 model trained on CIFAR-10. As shown in Table X, even when only the gradient norm is applied without adding noise, the model's accuracy and the effectiveness of our attack are significantly reduced. We observe that as the noise level increases, the improvement of CMIA compared to the base attack diminishes. This is because CMIA relies on identifying highly probable samples to perform cascading attacks on remaining instances. When selecting reliable anchors becomes difficult, the benefit of the cascading framework decreases. Nevertheless, CMIA consistently outperforms the base attack under all evaluated DP settings. We also evaluate the impact of DP-SGD in the non-adaptive setting in Table XI and observe similar patterns: as the privacy level increases, attack performance drops for all methods, and the gap between attacks narrows. Nevertheless, PMIA still outperforms existing MIAs in the low false-positive regime.

TABLE X: Effectiveness of using DP-SGD against CMIA with different privacy budgets trained on CIFAR-10. σ is the noise multiplier and ε is the privacy budget for DPSGD.

	Clipping norm $C = 10$		Model Acc	TPR @ 0.1% FPR (%)	
	σ	ε		Base (LiRA)	CMIA
No defense	-	-	90.05%	8.45%	**9.71%**
DP-SGD	0	∞	81.86%	1.36%	**2.80%**
	0.2	> 1000	48.62%	0.29%	**0.35%**
	0.5	31	36.45%	0.21%	**0.29%**
	1.0	4	30.48%	0.14%	**0.18%**

TABLE XI: Attack performance of PMIA against a ResNet50 model trained on CIFAR-10 using DP-SGD.

	TPR @ 0.1% FPR (%)		Balanced Accuracy	
	$\sigma = 0.2$	$\sigma = 0.5$	$\sigma = 0.2$	$\sigma = 0.5$
LOSS	0.02%	0.01%	51.36%	51.20%
Entropy	0.05%	0.04%	51.77%	51.62%
Calibration	0.10%	0.12%	51.72%	51.64%
Attack-R	0.00%	0.00%	51.79%	51.60%
LiRA	0.13%	0.11%	50.13%	50.42%
Canary	0.12%	0.10%	50.34%	50.38%
RMIA	0.16%	0.12%	51.77%	**52.10%**
RAPID	0.11%	0.13%	**51.81%**	51.96%
PMIA	**0.24%**	**0.20%**	51.55%	51.83%

VI. Related Work

Neural networks have been shown to be vulnerable to leaking sensitive information about their training data. A variety of attacks [5], [44], [45] have been proposed to quantify the extent of data leakage and assess the associated privacy risks. In this paper, we focus on the membership inference attack (MIA) [1], which predicts whether specific instances were included in the target model's training set. Closedly connected to Differential Privacy [2], [46], MIA has become a widely used tool to audit training data privacy of ML models [3], [4], [47], [48]. The first MIA against ML models was introduced by [1], which also proposed the technique of shadow training. MIAs and shadow training have since been extended to various scenarios, including white-box [10], [24], [49], black-box [11], [16], [39], [50]–[52], label-only access [53], [54], and model distillation [55]. Depending on when the adversary trains shadow models, MIAs can be categorized into two groups:

Adaptive MIAs. In the adaptive setting, the adversary is allowed to perform instance-by-instance analysis of the model's behavior by training additional predictors [10] or conducting hypothesis testing [7], [9], achieving state-of-the-art attack performance. However, these MIAs typically determine the membership of instances independently, which fails to consider the inherent dependencies between instances.

Non-Adaptive MIAs. Some studies [12], [14] argue that adaptive attacks are inefficient due to the need to train shadow models for every batch of queries. In response, a series of non-adaptive MIAs have been proposed to relax this assumption by training shadow models (or forgoing shadow training entirely) before accessing the query set. Several techniques have been introduced for non-adaptive MIAs, such as score functions [15], [39], difficulty calibration [8], [24], [37], loss trajectory [13], [56], quantile regression [12], and hypothesis testing [14], [38]. However, these attacks deviate from the membership posterior odds ratio test, leading to suboptimal attack performance.

Recent research [9], [57] has emphasized evaluating attacks by calculating their true positive rate at a significantly low false positive rate, and most existing MIAs perform poorly under this evaluation paradigm. In Section IV-A we provide theoretical insights into the rationale behind these evaluation metrics. While some literature [58] explores instance vulnerabilities to MIAs, it does not propose an attack that leverages them. In contrast, we theoretically analyze membership dependencies and introduce a framework that exploits these vulnerabilities and dependencies to mount a more powerful attack. Moreover, the dependencies we investigate differ from group-level MIAs [59], [60], which consider the inference target as a set of instances. Instead, our focus lies in exploring the membership dependencies between inference targets.

VII. Discussion & Conclusion

In this paper, we provide a new formulation of the MIA game and categorize existing MIAs into two categories: adaptive and non-adaptive. Guided by theoretical analyses of joint membership estimation and the posterior odds test, we propose two attacks (*i.e.,* CMIA and PMIA), one for each setting. Extensive experiments demonstrate the efficacy of the proposed attacks.

Our work has several limitations. First, the conditional shadow training strategy of CMIA limits its applicability for attacks that do not rely on shadow training. Second, our evaluations mainly focus on classification models, leaving the performance on generative models [61]–[63] unexamined. Nevertheless, our work provides both theoretical insights and practical approaches, offering new directions for MIAs.

Ethics Considerations

This paper focuses on membership inference attacks using publicly available datasets, such as MNIST and CIFAR-10, that do not contain personally identifiable information. We do not use any private or sensitive data, nor do we target specific individuals or proprietary models. Although our proposed attacks demonstrate improved attack performance, we show that established defenses like DP-SGD remain an effective mitigation strategy. We emphasize that the responsible use of such attack methods should be used responsibly to strengthen the security and privacy of ML systems, rather than maliciously exploiting them. We encourage the research community to use our findings in a manner that promotes ethical research and enhances privacy protections for users and data subjects.

References

[1] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*, 2017, pp. 3–18.

[2] C. Dwork, "Differential Privacy," in *Automata, Languages and Programming*, 2006, pp. 1–12.

[3] S. K. Murakonda and R. Shokri, "ML privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning," *arXiv preprint arXiv:2007.09339*, 2020.

[4] S. Song and D. Marn, "Introducing a New Privacy Testing Library in TensorFlow," 2022. [Online]. Available: https://blog.tensorflow.org/2020/06/introducing-new-privacy-testing-library.html

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

[6] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramer, B. Balle, D. Ippolito, and E. Wallace, "Extracting training data from diffusion models," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 5253–5270.

[7] Y. Wen, A. Bansal, H. Kazemi, E. Borgnia, M. Goldblum, J. Geiping, and T. Goldstein, "Canary in a Coalmine: Better Membership Inference with Ensembled Adversarial Queries," in *The Eleventh International Conference on Learning Representations*, 2023.

[8] Y. He, B. Li, Y. Wang, M. Yang, J. Wang, H. Hu, and X. Zhao, "Is Difficulty Calibration All We Need? Towards More Practical Membership Inference Attacks," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 1226–1240.

[9] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," in *2022 IEEE symposium on security and privacy (SP)*. IEEE, 2022, pp. 1897–1914.

[10] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*, 2019, pp. 739–753.

[11] J. Li, N. Li, and B. Ribeiro, "Membership Inference Attacks and Defenses in Classification Models," in *Eleventh ACM Conference on Data and Application Security and Privacy*, 2021, pp. 5–16.

[12] M. Bertran, S. Tang, A. Roth, M. Kearns, J. H. Morgenstern, and S. Z. Wu, "Scalable membership inference attacks via quantile regression," in *Advances in Neural Information Processing Systems*, 2023, pp. 314–330.

[13] H. Li, Z. Li, S. Wu, C. Hu, Y. Ye, M. Zhang, D. Feng, and Y. Zhang, "SeqMIA: Sequential-metric based membership inference attack," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 3496–3510.

[14] S. Zarifzadeh, P. Liu, and R. Shokri, "Low-Cost High-Power Membership Inference Attacks," in *International Conference on Machine Learning*, 2024, pp. 58 244–58 282.

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

[16] B. Jayaraman, L. Wang, K. Knipmeyer, Q. Gu, and D. Evans, "Revisiting membership inference under realistic assumptions," *arXiv preprint arXiv:2005.10881*, 2020.

[17] S. Albers, "Online algorithms: a survey," *Mathematical Programming*, vol. 97, no. 1, pp. 3–26, 2003.

[18] M. Luby, *Pseudorandomness and cryptographic applications*. Princeton University Press, 1996, vol. 1.

[19] D. Bleichenbacher, "Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS# 1," in *Advances in Cryptology—CRYPTO'98: 18th Annual International Cryptology Conference Santa Barbara, California, USA August 23–27, 1998 Proceedings 18*. Springer, 1998, pp. 1–12.

[20] A. E. Gelfand and A. F. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American statistical association*, vol. 85, no. 410, pp. 398–409, 1990.

[21] G. O. Roberts and J. S. Rosenthal, "Surprising convergence properties of some simple Gibbs samplers under various scans," *International Journal of Statistics and Probability*, vol. 5, no. 1, pp. 51–60, 2015.

[22] B. D. He, C. M. De Sa, I. Mitliagkas, and C. Ré, “Scan order in Gibbs sampling: Models in which it matters and bounds on how much,” *Advances in neural information processing systems*, vol. 29, 2016.

[23] K. Łatuszyński, G. O. Roberts, and J. S. Rosenthal, “Adaptive Gibbs samplers and related MCMC methods,” *The Annals of Applied Probability*, vol. 23, no. 1, pp. 66–99, 2013.

[24] A. Sablayrolles, M. Douze, C. Schmid, Y. Ollivier, and H. Jégou, “White-box vs black-box: Bayes optimal strategies for membership inference,” in *International Conference on Machine Learning*, 2019, pp. 5558–5567.

[25] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*, 2021, pp. 8748–8763.

[26] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou, “The Faiss library,” *arXiv preprint arXiv:2401.08281*, 2024.

[27] Y. LeCun, C. Cortes, and C. J. Burges, “The MNIST database of handwritten digits,” http://yann.lecun.com/exdb/mnist/, 1998.

[28] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” *arXiv preprint arXiv:1708.07747*, 2017.

[29] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” 2009.

[30] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[31] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[32] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

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

[35] I. Loshchilov and F. Hutter, “SGDR: Stochastic gradient descent with warm restarts,” *arXiv preprint arXiv:1608.03983*, 2016.

[36] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, “Random erasing data augmentation,” in *Proceedings of the AAAI conference on artificial intelligence*, 2020, pp. 13 001–13 008.

[37] L. Watson, C. Guo, G. Cormode, and A. Sablayrolles, “On the Importance of Difficulty Calibration in Membership Inference Attacks,” in *The Tenth International Conference on Learning Representations*, 2022.

[38] J. Ye, A. Maddi, S. K. Murakonda, V. Bindschaedler, and R. Shokri, “Enhanced membership inference attacks against machine learning models,” in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 3093–3106.

[39] L. Song and P. Mittal, “Systematic evaluation of privacy risks of machine learning models,” in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2615–2632.

[40] S. Sankararaman, G. Obozinski, M. I. Jordan, and E. Halperin, “Genomic privacy and limits of individual detection in a pool,” *Nature genetics*, vol. 41, no. 9, pp. 965–967, 2009.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[42] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey, “Cinic-10 is not imagenet or cifar-10,” *arXiv preprint arXiv:1810.03505*, 2018.

[43] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep learning with differential privacy,” in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[44] M. Fredrikson, S. Jha, and T. Ristenpart, “Model inversion attacks that exploit confidence information and basic countermeasures,” in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[45] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov, “Property inference attacks on fully connected neural networks using permutation invariant representations,” in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, 2018, pp. 619–633.

[46] N. Li, W. Qardaji, D. Su, Y. Wu, and W. Yang, “Membership privacy: A unifying framework for privacy definitions,” in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 889–900.

[47] M. Nasr, S. Songi, A. Thakurta, N. Papernot, and N. Carlin, “Adversary instantiation: Lower bounds for differentially private machine learning,” in *2021 IEEE Symposium on security and privacy (SP)*, 2021, pp. 866–882.

[48] M. Jagielski, J. Ullman, and A. Oprea, “Auditing differentially private machine learning: How private is private sgd?” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 22 205–22 216.

[49] K. Leino and M. Fredrikson, “Stolen memories: Leveraging model memorization for calibrated White-Box membership inference,” in *29th USENIX security symposium (USENIX Security 20)*, 2020, pp. 1605–1622.

[50] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter, and K. Chen, “Understanding membership inferences on well-generalized learning models,” *arXiv preprint arXiv:1802.04889*, 2018.

[51] Y. Liu, R. Wen, X. He, A. Salem, Z. Zhang, M. Backes, E. De Cristofaro, M. Fritz, and Y. Zhang, “ML-Doctor: Holistic risk assessment of inference attacks against machine learning models,” in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 4525–4542.

[52] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, “ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models,” in *26th Annual Network and Distributed System Security Symposium*, 2019.

[53] Z. Li and Y. Zhang, “Membership leakage in label-only exposures,” in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 880–895.

[54] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, “Label-only membership inference attacks,” in *International conference on machine learning*, 2021, pp. 1964–1974.

[55] M. Jagielski, M. Nasr, K. Lee, C. A. Choquette-Choo, N. Carlini, and F. Tramer, “Students parrot their teachers: Membership inference on model distillation,” in *Advances in Neural Information Processing Systems*, 2023, pp. 44 382–44 397.

[56] Y. Liu, Z. Zhao, M. Backes, and Y. Zhang, “Membership Inference Attacks by Exploiting Loss Trajectory,” in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, H. Yin, A. Stavrou, C. Cremers, and E. Shi, Eds., 2022, pp. 2085–2098.

[57] Y. Long, L. Wang, D. Bu, V. Bindschaedler, X. Wang, H. Tang, C. A. Gunter, and K. Chen, “A pragmatic approach to membership inferences on machine learning models,” in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2020, pp. 521–534.

[58] N. Carlini, M. Jagielski, C. Zhang, N. Papernot, A. Terzis, and F. Tramer, “The privacy onion effect: Memorization is relative,” in *Advances in Neural Information Processing Systems*, 2022, pp. 13 263–13 276.

[59] G. Li, S. Rezaei, and X. Liu, “User-Level Membership Inference Attack against Metric Embedding Learning,” in *ICLR 2022 Workshop on PAIR2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2022.

[60] M. Meeus, S. Jain, M. Rei, and Y.-A. de Montjoye, “Did the neurons read your book? document-level membership inference for large language models,” in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 2369–2385.

[61] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, “Logan: Membership inference attacks against generative models,” *arXiv preprint arXiv:1705.07663*, 2017.

[62] J. Duan, F. Kong, S. Wang, X. Shi, and K. Xu, “Are diffusion models vulnerable to membership inference attacks?” in *International Conference on Machine Learning*. PMLR, 2023, pp. 8717–8730.

[63] M. Duan, A. Suri, N. Mireshghallah, S. Min, W. Shi, L. Zettlemoyer, Y. Tsvetkov, Y. Choi, D. Evans, and H. Hajishirzi, “Do membership inference attacks work on large language models?” *arXiv preprint arXiv:2402.07841*, 2024.

[64] D. Williams, *Probability with martingales*. Cambridge university press, 1991.

[65] L. Tierney, “Markov chains for exploring posterior distributions,” *the Annals of Statistics*, pp. 1701–1728, 1994.

[66] J. S. Liu, “Peskun’s theorem and a modified discrete-state Gibbs sampler.” *Biometrika*, vol. 83, no. 3, 1996.

[67] G. O. Roberts and J. S. Rosenthal, "General state space Markov chains and MCMC algorithms," *Probability Surveys*, 2004.

[68] S. P. Meyn and R. L. Tweedie, *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

[69] R. Douc, E. Moulines, P. Priouret, P. Soulier, R. Douc, E. Moulines, P. Priouret, and P. Soulier, *Markov chains: Basic definitions*. Springer, 2018.

[70] S. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*. Cambridge University Press, 2009.

Appendix

A. Proof of Theorem 1

Proof. We prove this theorem by establishing that the Gibbs sampling procedure forms a Markov chain that satisfies the conditions for the Martingale Convergence Theorem [64].

a) Part 1: The Markov chain and its stationary distribution: First, we establish that the Gibbs sampling procedure forms a Markov chain with stationary distribution $\pi(M|o_\theta) = \Pr(M|o_\theta)$.

By construction, the transition from $\mathbf{M}^{(t)}$ to $\mathbf{M}^{(t+1)}$ depends only on the current state $\mathbf{M}^{(t)}$ and not on the previous states, satisfying the Markov property.

Let $\Pr(\mathbf{M}'|\mathbf{M})$ denote the transition probability from state $\mathbf{M}$ to state $\mathbf{M}'$ in one complete iteration of Gibbs sampling. To prove that $\pi(\mathbf{M}|o_\theta)$ is a stationary distribution, we need to verify the detailed balance equation [65]:

$$\pi(\mathbf{M}|o_\theta)\Pr(\mathbf{M}'|\mathbf{M}) = \pi(\mathbf{M}'|o_\theta)\Pr(\mathbf{M}|\mathbf{M}').$$

For Gibbs sampling, where we update one variable M_i at a time, the transition probability is:

$$\Pr(\mathbf{M}'|\mathbf{M}) = \Pr(M_i'|\mathbf{M}_{-i}, o_\theta) \cdot \mathbb{1}[\mathbf{M}'_{-i} = \mathbf{M}_{-i}]$$

where $\mathbf{M}_{-i}$ represents all components except M_i.

Following the results of Liu et al. [66], the construction of Gibbs sampling ensures:

$$\Pr(M_i'|\mathbf{M}_{-i}, o_\theta) = \frac{\pi(M_i', \mathbf{M}_{-i}|o_\theta)}{\sum_{m_i}\pi(m_i, \mathbf{M}_{-i}|o_\theta)} = \pi(M_i'|\mathbf{M}_{-i}, o_\theta)$$

Since the conditional distribution $\pi(M_i|M_{-i}, o_\theta)$ is used for sampling, the detailed balance condition is automatically satisfied [67], and $\pi(M|o_\theta)$ is indeed the stationary distribution of the Markov chain.

b) Part 2: Convergence via Martingale theory: To establish convergence, we apply the Martingale Convergence Theorem as presented in Meryn and Tweedie [68]. Let the performance measure be such

$$|L(\mathbf{M}, D)| < \infty, \quad \mathbf{M} \sim \pi(\mathbf{M}|o_\theta).$$

Let $\mathbf{M}^{(1)} \sim \pi(\mathbf{M}|o_\theta)$ and define the sequence of σ-algebras $\mathcal{F}_t = \sigma(\mathbf{M}^{(1)}, \mathbf{M}^{(2)}, \ldots, \mathbf{M}^{(t)})$, representing the information available up to time t. Following Douc et al. [69], we define:

$$X_t = \mathbb{E}[L(\mathbf{M}^{(t)}, D)|\mathcal{F}_t].$$

This defines a martingale, since:

$$\begin{aligned}\mathbb{E}[X_{t+1}|\mathcal{F}_t] &= \mathbb{E}[\mathbb{E}[L(\mathbf{M}^{(t+1)}, D)|\mathcal{F}_{t+1}]|\mathcal{F}_t] \\ &= \mathbb{E}[L(\mathbf{M}^{(t+1)}, D)|\mathcal{F}_t] = X_t,\end{aligned}$$

since the Markov chain is in steady state. By the Martingale Convergence Theorem [64], X_t converges almost surely to a random variable X_∞ as $t \to \infty$.

We now need to show that X_∞ as $t \to \infty$ converges for any choice of $\mathbf{M}^{(1)}$. Under mild conditions on $\Pr(\mathbf{M}|o_\theta)$, where all query instances have some non-zero probability of being members and non-members, it ensures irreducibility and aperiodicity of the Gibbs sampler [65], [67], and then the Markov chain is ergodic. In this case, following the Ergodic Theorem for Markov chains [70], we have:

$$X_\infty = \mathbb{E}_\pi[L(\mathbf{M}, D)].$$

And for the time average:

$$S_T = \frac{1}{T}\sum_{t=1}^{T} L(\mathbf{M}^{(t)}, D),$$

the Law of Large Numbers for Markov chains [70] ensures:

$$S_T \xrightarrow{a.s.} \mathbb{E}_\pi[L(\mathbf{M}, D)]$$

as $T \to \infty$. This establishes that the Gibbs sampling procedure for joint MIA converges to the correct joint membership distribution, and empirical averages computed from the samples converge almost surely to their expected values under the target distribution. □

B. Proof of Theorem 2

Proof. From a Bayesian perspective, based on the adversary's observation e_θ, we define the posterior probabilities as follows:

$$\begin{aligned}p_{\text{in}}(x_i, y_i) &= \Pr\left(M_i = 1 \mid \mathcal{Q}(f_\theta) = e_\theta\right) \\ &= \Pr\left((x_i, y_i) \in D \mid \mathcal{Q}(f_\theta) = e_\theta\right) \\ &= \Pr\left(D \in \mathcal{S}^{+}_{(x_i,y_i)} \mid \mathcal{Q}(\mathcal{T}(D)) = e_\theta\right), \\ p_{\text{out}}(x_i, y_i) &= \Pr\left(M_i = 0 \mid \mathcal{Q}(f_\theta) = e_\theta\right) \\ &= \Pr\left((x_i, y_i) \notin D \mid \mathcal{Q}(f_\theta) = e_\theta\right) \\ &= \Pr\left(D \in \mathcal{S}^{-}_{(x_i,y_i)} \mid \mathcal{Q}(\mathcal{T}(D)) = e_\theta\right),\end{aligned}$$

where $p_{\text{in}}(x_i, y_i)$ is the posterior probability that (x_i, y_i) is in the training data and $p_{\text{out}}(x_i, y_i)$ is the posterior probability that it is not. The adversary should maximize the posterior probability of correctly identifying whether (x_i, y_i) is part of the training set. Thus, the adversary will use the posterior odds to infer that (x_i, y_i) is in the training set if:

$$\frac{p_{\text{in}}(x_i, y_i)}{p_{\text{out}}(x_i, y_i)} > 1, \tag{3}$$

TABLE XII: Data splits in the adaptive setting.

	MNIST		FMNIST		CIFAR-10		CIFAR-100		Purchase		Texas	
	D_1	D_2	D_1	D_2	D_1	D_2	D_1	D_2	D_1	D_2	D_1	D_2
Size	60k	10k	60k	10k	50k	10k	50k	10k	160k	37k	67k	10k

TABLE XIII: Data splits in the non-adaptive setting.

Dataset	D_{query}		$D_{\text{adv}}^{\text{non-adapt}}$		
	Train	Val	Train	Val	Reference
MNIST	11,667	11,667	11,667	11,667	23,334
FMNIST	11,667	11,667	11,667	11,667	23,334
CIFAR-10	10,000	10,000	10,000	10,000	20,000
CIFAR-100	10,000	10,000	10,000	10,000	20,000
Purchase	32,887	32,887	32,887	32,887	65,774
Texas	11,221	11,221	11,221	11,221	22,443

and it infers that (x_i, y_i) is not in the training set otherwise. By the Bayesian rule, we can express the above ratio as follows:

$$\begin{aligned}
&\frac{\Pr\left(D \in \mathcal{S}^+_{(x_i,y_i)} \mid \mathcal{Q}(\mathcal{T}(D)) = e_\theta\right)}{\Pr\left(D \in \mathcal{S}^-_{(x_i,y_i)} \mid \mathcal{Q}(\mathcal{T}(D)) = e_\theta\right)} \\
=&\frac{\Pr(\mathcal{Q}(\mathcal{T}(D)) = e_\theta, D \in \mathcal{S}^+_{(x_i,y_i)})}{\Pr(\mathcal{Q}(\mathcal{T}(D)) = e_\theta, D \in \mathcal{S}^-_{(x_i,y_i)})} \\
=&\frac{\sum_{S\in\mathcal{S}^+_{(x_i,y_i)}} \Pr(D = S) \cdot \Pr(\mathcal{Q}(\mathcal{T}(S)) = e_\theta)}{\sum_{S\in\mathcal{S}^-_{(x_i,y_i)}} \Pr(D = S) \cdot \Pr(\mathcal{Q}(\mathcal{T}(S)) = e_\theta)} \qquad (4) \\
=&\frac{\Pr(D \in \mathcal{S}^+_{(x_i,y_i)})}{\Pr(D \in \mathcal{S}^-_{(x_i,y_i)})} \times \\
&\frac{\sum_{S\in\mathcal{S}^+_{(x_i,y_i)}} \frac{\Pr(D=S)}{\Pr(D\in\mathcal{S}^+_{(x_i,y_i)})} \Pr(\mathcal{Q}(\mathcal{T}(S)) = e_\theta)}{\sum_{S\in\mathcal{S}^-_{(x_i,y_i)}} \frac{\Pr(D=S)}{\Pr(D\in\mathcal{S}^-_{(x_i,y_i)})} \Pr(\mathcal{Q}(\mathcal{T}(S)) = e_\theta)}.
\end{aligned}$$

We define $\mathcal{L}(D, e_\theta) = \Pr(\mathcal{Q}(\mathcal{T}(D)) = e_\theta)$ to denote the likelihood. Stemming from Equation (3) and Equation (4), the adversary will infer (x_i, y_i) is in the training set if:

$$\frac{\mathbb{E}_{D'\sim\mathcal{S}^+_{(x_i,y_i)}} \mathcal{L}(D', e_\theta)}{\mathbb{E}_{D'\sim\mathcal{S}^-_{(x_i,y_i)}} \mathcal{L}(D', e_\theta)} > \frac{\Pr((x,y) \notin D)}{\Pr((x,y) \in D)}.$$

□

C. Dataset Description

MNIST. MNIST [27] is a benchmark dataset for evaluating handwritten digit classification algorithms. It contains 60,000 grayscale images of 28×28 pixels for training and 10,000 images for testing. The dataset consists of 10 classes, representing digits from 0 to 9.

Fashion-MNIST. Fashion-MNIST (FMNIST) [28] is a dataset designed as a more challenging replacement for MNIST, containing 60,000 grayscale images of size 28×28 pixels for training and 10,000 images for testing. It consists of 10 classes representing different fashion items, such as t-shirts, trousers, and sneakers.

TABLE XIV: Prediction accuracy in the adaptive setting.

Model	MNIST		FMNIST		C-10		C-100	
	Train	Val	Train	Val	Train	Val	Train	Val
ResNet50	100.0%	99.5%	100.0%	91.3%	99.9%	90.1%	99.9%	67.9%
VGG16	100.0%	99.5%	100.0%	92.9%	99.9%	87.1%	99.9%	60.8%
DenseNet121	100.0%	99.5%	99.9%	93.3%	99.9%	89.3%	99.9%	63.4%
MobileNetV2	100.0%	99.4%	100.0%	92.4%	99.9%	88.0%	99.9%	61.0%

TABLE XV: Prediction accuracy in the non-adaptive setting.

Model	MNIST		FMNIST		C-10		C-100	
	Train	Val	Train	Val	Train	Val	Train	Val
ResNet50	100.0%	99.5%	100.0%	94.8%	99.7%	90.4%	92.3%	66.9%
VGG16	100.0%	99.7%	100.0%	95.6%	99.9%	91.7%	99.6%	72.3%
DenseNet121	100.0%	99.6%	100.0%	96.1%	99.9%	90.2%	99.9%	71.5%
MobileNetV2	100.0%	99.5%	100.0%	94.9%	99.9%	97.1%	100.0%	72.4%

CIFAR10. CIFAR-10 [29] is a benchmark dataset for general image classification tasks, containing 60,000 color images of size 32×32 pixels, equally distributed across 10 classes, including animals like cats, dogs, and birds, as well as vehicles like airplanes and trucks.

CIFAR-100. CIFAR-100 [29] is a dataset similar to CIFAR-10 but with a greater level of complexity, as it contains 100 classes instead of 10. It also includes 60,000 color images of size 32×32 pixels, with each class containing 600 images.

Purchase. Purchase [1] is a dataset of shopping records with 197,324 samples of 600 dimensions. Following previous works [1], [9], [13], we cluster these data into 100 classes to train a Multi-Layer Perceptron (MLP) classifier.

Texas. This dataset is designed to predict a patient's primary medical procedure using 6,170 binary features. Following [1], the data comprises records from 67,330 patients, with the 100 most frequent procedures used as classification labels.

D. Accuracy of the Trained Target Model

We report the training and validation accuracies of the target model for both adaptive and non-adaptive settings in Table XIV and Table XV, respectively.

E. Experiments of CMIA on Other Models

Here, we present the experimental results on the remaining three model architectures in the adaptive setting. Specifically, Table XVI shows the results for VGG16, Table XVII presents the results for DenseNet121, and Table XVIII displays the results for MobileNetV2. We also present the ROC curves to better demonstrate the improvement achieved by CMIA. Specifically, we use LiRA as the base attack and demonstrate the improvement across four image datasets in Figure 9.

F. Experiments of PMIA on Other Models

We present the experimental results on the remaining three model architectures in the non-adaptive setting. Specifically, Table XIX shows the results for VGG16, Table XX presents the results for DenseNet121, and Table XXI displays the results for MobileNetV2. We also present the ROC curves of non-adaptive attacks on four datasets, as shown in Figure 10.

TABLE XVI: Performance comparison of *adaptive* attacks using CMIA on VGG16 trained on four image datasets.

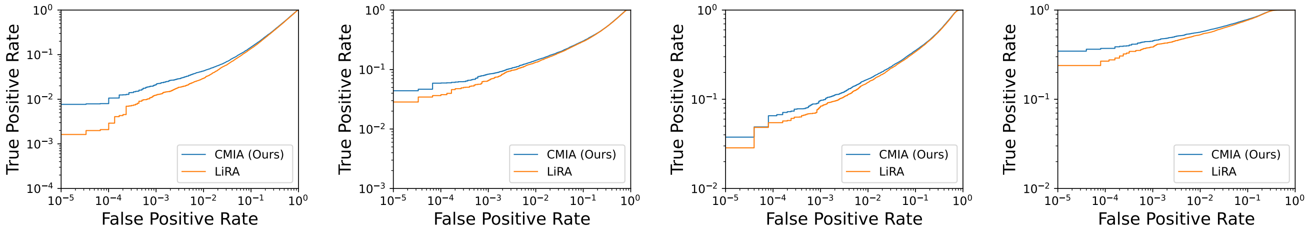
Method		TPR @ 0.001% FPR				TPR @ 0.1% FPR				Balanced Accuracy			
		MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100
Calibration	Base	0.01%	0.69%	0.46%	3.18%	0.40%	1.94%	1.67%	7.15%	50.61%	53.61%	55.17%	63.83%
	CMIA	**0.04%**	**0.94%**	**0.77%**	**5.02%**	**0.69%**	**2.58%**	**1.95%**	**8.82%**	**51.37%**	**53.94%**	**56.52%**	**65.93%**
	%Imp.	300.00%	36.23%	67.39%	57.86%	72.50%	32.99%	16.77%	23.36%	1.50%	0.62%	2.45%	3.29%
Attack-R	Base	0.00%	0.00%	0.00%	0.26%	0.70%	4.08%	5.07%	1.06%	51.48%	57.03%	60.04%	74.79%
	CMIA	**0.00%**	**0.00%**	**0.00%**	**0.49%**	**0.95%**	**4.95%**	**5.77%**	**2.53%**	**51.97%**	**57.74%**	**60.53%**	**75.08%**
	%Imp.	-	-	-	88.46%	35.71%	21.32%	13.81%	138.68%	0.95%	1.24%	0.82%	0.39%
LiRA	Base	0.11%	2.09%	2.01%	12.95%	0.94%	5.06%	7.21%	28.70%	51.24%	58.12%	64.02%	82.07%
	CMIA	**0.79%**	**3.63%**	**3.02%**	**17.88%**	**1.47%**	**5.98%**	**9.18%**	**32.02%**	**52.05%**	**58.88%**	**65.47%**	**82.55%**
	%Imp.	618.18%	73.68%	50.25%	38.07%	56.38%	18.18%	27.32%	11.57%	1.58%	1.31%	2.26%	0.58%
Canary	Base	0.15%	2.16%	2.10%	12.82%	0.94%	5.38%	7.08%	27.53%	51.83%	58.49%	64.33%	81.44%
	CMIA	**0.82%**	**3.99%**	**3.10%**	**18.40%**	**1.51%**	**6.33%**	**9.53%**	**31.90%**	**52.33%**	**59.06%**	**65.95%**	**81.95%**
	%Imp.	446.67%	84.72%	47.62%	43.53%	60.64%	17.66%	34.60%	15.87%	0.96%	0.97%	2.52%	0.63%
RMIA	Base	0.25%	2.07%	0.00%	0.27%	1.02%	3.81%	5.06%	20.49%	51.82%	59.02%	63.82%	80.74%
	CMIA	**0.37%**	**3.11%**	**0.00%**	**0.78%**	**1.94%**	**5.07%**	**7.43%**	**25.80%**	**52.07%**	**59.73%**	**64.05%**	**81.69%**
	%Imp.	48.00%	50.24%	-	188.89%	90.20%	33.07%	46.84%	25.92%	0.48%	1.20%	0.36%	1.18%
RAPID	Base	0.21%	1.10%	0.42%	3.02%	0.76%	3.02%	3.01%	14.03%	51.2%	56.89%	60.01%	76.82%
	CMIA	**0.38%**	**2.53%**	**0.79%**	**4.75%**	**1.05%**	**4.85%**	**5.72%**	**17.63%**	**51.42%**	**57.42%**	**61.04%**	**77.35%**
	%Imp.	80.95%	130.00%	88.10%	57.28%	38.16%	60.60%	90.03%	25.66%	0.43%	0.93%	1.72%	0.69%

TABLE XVII: Performance comparison of *adaptive* attacks using CMIA on DenseNet121 trained on four image datasets.

Method		TPR @ 0.001% FPR				TPR @ 0.1% FPR				Balanced Accuracy			
		MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100
Calibration	Base	0.09%	0.54%	0.53%	2.23%	0.28%	1.53%	1.84%	6.01%	50.45%	52.95%	53.94%	62.1%
	CMIA	**0.18%**	**1.05%**	**0.68%**	**2.64%**	**0.39%**	**2.03%**	**2.07%**	**6.85%**	**51.42%**	**53.49%**	**54.90%**	**62.5%**
	%Imp.	100.00%	94.44%	28.30%	18.39%	39.29%	32.68%	12.50%	13.98%	1.92%	1.02%	1.78%	0.64%
Attack-R	Base	0.00%	0.00%	0.00%	0.00%	0.74%	3.51%	0.04%	0.93%	51.02%	56.14%	59.52%	74.88%
	CMIA	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**1.13%**	**4.62%**	**0.15%**	**2.83%**	**51.95%**	**57.17%**	**60.29%**	**75.63%**
	%Imp.	-	-	-	-	52.70%	31.62%	275.00%	204.30%	1.82%	1.83%	1.29%	1.00%
LiRA	Base	0.23%	2.01%	2.74%	10.05%	1.02%	6.05%	8.03%	24.60%	51.06%	58.46%	63.45%	80.09%
	CMIA	**0.36%**	**2.80%**	**3.33%**	**13.67%**	**1.36%**	**7.18%**	**10.85%**	**28.64%**	**51.50%**	**58.80%**	**63.84%**	**81.68%**
	%Imp.	56.52%	39.30%	21.53%	36.02%	33.33%	18.68%	35.12%	16.42%	0.86%	0.58%	0.61%	1.99%
Canary	Base	0.31%	2.17%	2.77%	10.16%	0.97%	6.28%	8.16%	24.73%	51.53%	58.92%	63.58%	80.12%
	CMIA	**0.39%**	**3.04%**	**3.51%**	**13.48%**	**1.29%**	**7.25%**	**11.01%**	**28.41%**	**51.73%**	**59.00%**	**63.75%**	**81.50%**
	%Imp.	25.81%	40.09%	26.71%	32.68%	32.99%	15.45%	34.93%	14.88%	0.39%	0.14%	0.27%	1.72%
RMIA	Base	0.28%	1.04%	0.54%	0.29%	0.94%	4.06%	3.60%	1.84%	51.04%	58.07%	62.04%	79.11%
	CMIA	**0.33%**	**1.97%**	**0.66%**	**0.77%**	**1.42%**	**4.66%**	**4.15%**	**3.63%**	**51.99%**	**58.86%**	**62.38%**	**79.57%**
	%Imp.	17.86%	89.42%	22.22%	165.52%	51.06%	14.78%	15.28%	97.28%	1.86%	1.36%	0.55%	0.58%
RAPID	Base	0.12%	1.36%	1.05%	5.48%	0.65%	3.75%	3.81%	13.09%	51.39%	57.02%	59.20%	78.01%
	CMIA	**0.37%**	**2.04%**	**1.98%**	**5.96%**	**1.33%**	**4.40%**	**4.63%**	**17.73%**	**51.96%**	**57.77%**	**60.12%**	**79.04%**
	%Imp.	208.33%	50.00%	88.57%	8.76%	104.62%	17.33%	21.52%	35.45%	1.11%	1.32%	1.55%	1.32%

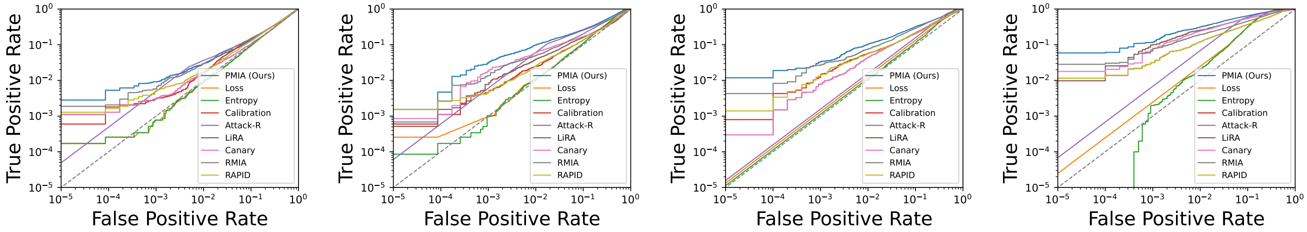
TABLE XVIII: Performance comparison of *adaptive* attacks using CMIA on MobileNetV2 trained on four image datasets.

Method		TPR @ 0.001% FPR				TPR @ 0.1% FPR				Balanced Accuracy			
		MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100
Calibration	Base	0.10%	0.00%	0.43%	1.73%	0.35%	1.62%	1.56%	6.32%	51.35%	54.21%	55.53%	64.3%
	CMIA	**0.42%**	**0.00%**	**0.72%**	**3.03%**	**0.77%**	**1.93%**	**2.00%**	**8.84%**	**51.92%**	**54.83%**	**55.76%**	**65.78%**
	%Imp.	320.00%	-	67.44%	75.14%	120.00%	19.14%	28.21%	39.87%	1.11%	1.14%	0.41%	2.30%
Attack-R	Base	0.00%	0.00%	0.00%	0.14%	0.00%	0.00%	0.92%	2.85%	51.52%	56.56%	58.69%	74.63%
	CMIA	**0.00%**	**0.00%**	**0.00%**	**0.43%**	**0.00%**	**0.00%**	**1.26%**	**4.48%**	**51.96%**	**56.84%**	**58.88%**	**75.12%**
	%Imp.	-	-	-	207.14%	-	-	36.96%	57.19%	0.85%	0.50%	0.32%	0.66%
LiRA	Base	0.18%	1.01%	2.33%	11.52%	1.02%	4.01%	6.01%	23.90%	51.73%	58.62%	62.04%	80.08%
	CMIA	**0.64%**	**2.17%**	**3.02%**	**16.90%**	**1.72%**	**5.36%**	**7.23%**	**30.65%**	**52.24%**	**59.61%**	**62.65%**	**81.69%**
	%Imp.	255.56%	114.85%	29.61%	46.70%	68.63%	33.67%	20.30%	28.24%	0.99%	1.69%	0.98%	2.01%
Canary	Base	0.16%	1.25%	2.37%	12.07%	1.05%	4.04%	6.12%	24.07%	51.77%	58.61%	62.53%	80.17%
	CMIA	**1.53%**	**2.46%**	**3.18%**	**17.74%**	**1.85%**	**5.77%**	**7.35%**	**31.02%**	**52.31%**	**59.99%**	**62.98%**	**81.72%**
	%Imp.	856.25%	96.80%	34.18%	46.98%	76.19%	42.82%	20.10%	28.87%	1.04%	2.35%	0.72%	1.93%
RMIA	Base	0.17%	1.83%	0.10%	3.74%	1.23%	3.05%	1.18%	6.38%	52.43%	59.07%	61.01%	79.64%
	CMIA	**0.43%**	**2.84%**	**0.27%**	**5.06%**	**1.83%**	**4.49%**	**1.99%**	**8.62%**	**52.62%**	**59.82%**	**61.49%**	**80.81%**
	%Imp.	152.94%	55.19%	170.00%	35.29%	48.78%	47.21%	68.64%	35.11%	0.36%	1.27%	0.79%	1.47%
RAPID	Base	0.23%	0.64%	0.32%	5.31%	1.08%	3.01%	2.93%	14.73%	52.16%	58.57%	59.39%	76.28%
	CMIA	**0.41%**	**1.05%**	**0.52%**	**7.29%**	**2.05%**	**4.21%**	**3.40%**	**20.39%**	**52.36%**	**58.90%**	**59.74%**	**77.58%**
	%Imp.	78.26%	64.06%	62.50%	37.29%	89.81%	39.87%	16.04%	38.42%	0.38%	0.56%	0.59%	1.70%



(a) MNIST (b) Fashion-MNIST (c) CIFAR-10 (d) CIFAR-100

Fig. 9: The ROC curves of LIRA and CMIA (base: LiRA) on ResNet50 models trained on four image datasets.



(a) MNIST (b) Fashion-MNIST (c) CIFAR-10 (d) CIFAR-100

Fig. 10: The ROC curves of non-adaptive attack results on ResNet50 models trained on four image datasets.

TABLE XIX: Performance comparison of *non-adaptive* attacks on VGG16 trained on four image datasets.

Method	**TPR @ 0.001% FPR**				**TPR @ 0.1% FPR**				**Balanced Accuracy**			
	MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100
LOSS	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.01%	0.16%	52.18%	59.03%	64.86%	85.68%
Entropy	0.00%	0.00%	0.00%	0.05%	0.00%	0.01%	0.02%	0.23%	52.02%	59.05%	67.56%	88.09%
Calibration	0.07%	0.07%	0.01%	1.53%	0.27%	1.30%	1.60%	4.52%	52.15%	54.79%	58.61%	68.21%
Attack-R	0.00%	0.00%	0.00%	0.00%	0.33%	0.00%	0.00%	0.00%	52.01%	57.60%	64.19%	82.89%
LiRA	0.00%	0.00%	0.00%	2.95%	0.00%	0.29%	1.51%	9.83%	50.41%	51.98%	56.68%	78.51%
Canary	0.00%	0.00%	0.02%	3.58%	0.00%	0.37%	1.93%	10.15%	50.77%	51.34%	56.69%	80.04%
RMIA	0.13%	0.08%	0.24%	5.74%	0.32%	1.78%	3.42%	10.12%	51.94%	57.78%	64.46%	75.53%
RAPID	0.08%	0.04%	0.13%	0.28%	0.25%	0.15%	2.73%	10.78%	52.07%	58.49%	60.88%	82.52%
PMIA	**0.16%**	**0.12%**	**0.65%**	**9.00%**	**0.46%**	**3.09%**	**5.22%**	**26.21%**	**52.27%**	**59.76%**	**67.68%**	**88.93%**
%Imp.	23.08%	50.00%	170.83%	56.79%	39.39%	73.60%	52.63%	143.14%	0.01%	1.20%	0.18%	0.95%

TABLE XX: Performance comparison of *non-adaptive* attacks on DenseNet121 trained on four images datasets.

Method	**TPR @ 0.001% FPR**				**TPR @ 0.1% FPR**				**Balanced Accuracy**			
	MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100
LOSS	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	52.15%	56.55%	61.57%	86.53%
Entropy	0.00%	0.00%	0.00%	0.00%	0.00%	0.15%	0.12%	0.13%	52.07%	56.43%	61.43%	87.61%
Calibration	0.00%	0.12%	0.10%	1.63%	0.51%	0.83%	0.78%	4.70%	52.36%	54.06%	59.78%	69.41%
Attack-R	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	52.15%	56.06%	62.13%	83.46%
LiRA	0.00%	0.13%	0.00%	5.73%	0.60%	0.42%	0.78%	12.05%	51.05%	52.96%	59.56%	79.41%
Canary	0.00%	0.21%	0.00%	5.87%	0.62%	0.51%	0.93%	12.18%	51.12%	53.18%	59.38%	79.80%
RMIA	0.01%	0.23%	0.22%	3.21%	0.82%	1.05%	1.21%	11.83%	52.41%	56.62%	62.06%	76.17%
RAPID	0.01%	0.20%	0.18%	3.05%	0.84%	0.93%	0.92%	10.84%	51.93%	55.17%	60.25%	77.89%
PMIA	**0.19%**	**0.27%**	**0.26%**	**9.75%**	**1.23%**	**1.25%**	**2.45%**	**19.70%**	**52.64%**	**56.70%**	**62.94%**	**89.07%**
%Imp.	1800.00%	17.39%	18.19%	66.10%	48.80%	19.05%	102.48%	61.74%	0.44%	1.14%	1.30%	1.67%

TABLE XXI: Performance comparison of *non-adaptive* attacks on MobileNetV2 trained on four image datasets.

Method	**TPR @ 0.001% FPR**				**TPR @ 0.1% FPR**				**Balanced Accuracy**			
	MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100	MNIST	FMNIST	C-10	C-100
LOSS	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	54.60%	64.45%	72.61%	88.55%
Entropy	0.00%	0.00%	0.00%	0.00%	0.00%	0.01%	0.00%	0.00%	54.61%	64.37%	74.41%	89.92%
Calibration	0.00%	0.16%	0.35%	1.53%	0.53%	0.92%	1.57%	4.81%	53.70%	57.21%	60.71%	69.85%
Attack-R	0.00%	0.00%	0.00%	0.00%	0.83%	0.00%	0.00%	0.00%	52.96%	59.94%	69.43%	84.20%
LiRA	0.00%	0.00%	0.47%	0.15%	0.32%	0.78%	3.71%	12.65%	50.22%	51.92%	62.63%	80.04%
Canary	0.00%	0.02%	0.53%	0.23%	0.35%	0.92%	3.85%	13.73%	50.89%	52.38%	63.61%	81.09%
RMIA	0.13%	0.25%	0.80%	3.18%	0.86%	1.24%	3.12%	13.12%	54.17%	61.62%	68.60%	78.57%
RAPID	0.07%	0.38%	0.52%	2.57%	0.75%	2.01%	2.96%	7.98%	54.61%	62.38%	62.41%	80.72%
PMIA	**0.56%**	**2.25%**	**3.34%**	**6.15%**	**1.15%**	**4.15%**	**6.30%**	**22.89%**	**54.89%**	**64.49%**	**75.81%**	**91.2%**
%Imp.	330.77%	492.11%	317.50%	93.40%	33.72%	106.47%	63.64%	66.72%	0.51%	0.06%	1.88%	1.42%