Imitative Membership Inference Attack

Yuntao Du
Purdue University

Yuetian Chen
Purdue University

Hanshen Xiao
Purdue University & NVIDIA Research

Bruno Ribeiro
Purdue University

Ninghui Li
Purdue University

A Membership Inference Attack (MIA) assesses how much a target machine learning model reveals about its training data by determining whether specific query instances were part of the training set. State-of-the-art MIAs rely on training hundreds of shadow models that are independent of the target model, leading to significant computational overhead. In this paper, we introduce Imitative Membership Inference Attack (IMIA), which employs a novel imitative training technique to strategically construct a small number of target-informed imitative models that closely replicate the target model's behavior for inference. Extensive experimental results demonstrate that IMIA substantially outperforms existing MIAs in various attack settings while only requiring less than 5% of the computational cost of state-of-the-art approaches.

1 Introduction

Over the past decade, machine learning (ML) has seen remarkable advances, with models such as neural networks increasingly being trained on sensitive datasets. This trend has raised growing concerns about the privacy risks associated with these models. Membership inference attacks (MIAs) [47] have been proposed to measure the degree to which a model leaks information by determining whether some instances were part of its training set. Closely related to Differential Privacy (DP) [13, 32], MIAs have become a widely adopted technique for empirical auditing of various trustworthy risks in ML models [39,42,50,55,56], and serve as key components for more sophisticated attacks [4, 5, 51].

MIAs in the black-box setting [3, 49, 57, 58] typically exploit the model's behavioral discrepancy between its training instances (*i.e.,* members) and non-training instances (*i.e.,* non-members) for inference. A widely-used strategy to explore the discrepancy is *shadow training* [47], which involves training multiple shadow models on datasets drawn from the same distribution as the target model's training set. For each query instance, the membership scores generated by the shadow models trained without the instance (shadow *out* models) represent the *out* distribution for the instance. In some settings, one also has shadow models that are trained with the instance as a member, and can use the scores generated by these shadow *in* models to compute an *in* distribution. Given a target model, one tries to tell which distribution the score from the target model fits better. MIAs [3, 9, 19, 53, 59] utilizing shadow training have shown strong attack performance.

A critical limitation of existing shadow-based MIAs lies in their **substantial computational overhead**. Training each shadow model incurs non-trivial overhead, and state-of-the-art attacks like LiRA [3] and PMIA [9] require training hundreds of shadow models to estimate the likelihood ratio for inference. This requirement imposes a substantial computational burden, which reduces the feasibility of using these state-of-the-art MIAs for practical privacy auditing [59]. In addition, it also impedes research reproducibility within the privacy community (see Section 2 for further discussion).

We observe that this inefficiency stems from the *target-agnostic* design of shadow training. That is, the current shadow training process fails to take advantage of knowing the target model under attack. As a result, the shadow models learn only the general patterns of members and non-members, rather than the specific behavioral discrepancies of the target. As shown in the top row of Figure 1, this target-agnostic approach causes shadow models to exhibit high predictive variance across instances with varying levels of attack vulnerability. Consequently, existing MIAs need to train a multitude of shadow models to capture this variability, resulting in substantial computational overhead and suboptimal performance.

To address this challenge, we introduce Imitative Membership Inference Attack (IMIA), a novel approach that improves both attack efficiency and effectiveness. At the core of IMIA is *imitative training*, a new shadow training technique that trains *target-informed* imitative models to mimic the target model's behavior. Specifically, we first train a set of imitative *out* models by applying weighted logits (log of output confidence) matching to the target model's outputs, capturing the behavior of the target on non-member instances. We then leverage these models to continue training on specially selected "pivot" instances from the adversary's dataset us-

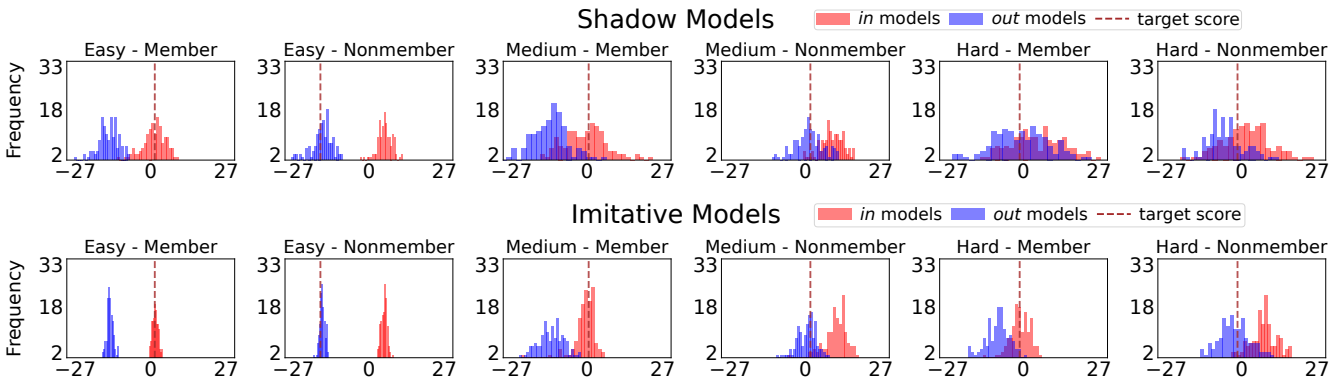


Figure 1: Distributions of membership scores (*i.e.,* scaled confidence scores, defined in Section 3.1) for six CIFAR-100 instances with varying attack difficulty (easy, medium, hard). A larger overlap between *in* (trained with the instance) and *out* (trained without the instance) score distributions indicates greater difficulty in determining membership. The dashed vertical line represents each instance's score on the target model. **Top row**: *target-agnostic* shadow models show high predictive variance (with long tails and wide distributions) for both members and non-members, resulting in significant overlap that hampers reliable inference, especially for hard-to-attack instances. **Bottom row**: *target-informed* imitative models exhibit more stable and well-separated distributions, enabling effective inference across all levels of difficulty. More examples are in the full version.

ing standard cross-entropy loss, yielding a set of imitative *in* models that capture the target model's predictions on member instances. This targeted approach enables us to pair each query instance with a set of *in* and *out* predictions estimated from the target model. As illustrated in the bottom row of Figure 1, imitative models yield significantly more separable and stable estimations, enabling the adversary to better exploit behavioral discrepancies for more powerful inference.

We note that some attacks [15,30,57] employ model distillation to construct shadow models from the target model. However, these methods directly apply the standard distillation technique [20] without tailoring it to capture the membership-related behaviors of the target model. Furthermore, the resulting distillation models can only reflect behaviors on non-member instances, making it infeasible to exploit behavioral discrepancies for effective membership inference. A more detailed discussion of the differences between our approach and distillation-based methods is provided in Section 3.2.

We compare the proposed attack with a broad range of MIAs on six benchmark datasets (4 image and 2 non-image datasets), under both non-adaptive and adaptive settings [9] (also referred to as "offline" and "online" [3]; see Section 2 for details). Experimental results show that IMIA consistently outperforms all evaluated attacks (including distillation-based attacks) while using significantly fewer shadow models, with particularly strong gains in the increasingly advocated non-adaptive setting. For example, IMIA surpasses the state-of-the-art non-adaptive attack (*i.e.,* PMIA [9]) by over **9**$\times$ in true positive rate at a zero false positive rate on Fashion-MNIST, while using only **5%** of its computational resources. Similarly, IMIA outperforms all adaptive attacks with far less computation. We also conduct comprehensive analyses to show the effectiveness of imitative training over shadow training, evaluate the impact of components in IMIA, and assess the effectiveness of existing defenses against the proposed attack. In summary, we make the following contributions:

- We propose IMIA, a new MIA that employs a novel target-informed imitative training technique to effectively capture the target model's behavioral discrepancies for inference.
- We conduct extensive experiments across diverse datasets, models, and attack settings, showing that IMIA consistently outperforms all MIAs with substantially less computation.
- We provide an in-depth analysis of IMIA's effectiveness, showing that its strength lies in imitative training and highlighting its advantages over shadow training.

Organization. The rest of this paper is organized as follows. Section 2 introduces the MIA definitions and threat models. Section 3 describes the proposed attack, IMIA, in detail. Section 4 presents the experimental evaluation. Related work is discussed in Section 5, and the paper concludes in Section 6.

The full version of this paper can be found in the following link: https://arxiv.org/abs/2509.06796.

2 Problem Definition and Threat Models

The goal of a membership inference attack (MIA) [47] is to determine whether some instances were included in the training data of a given model f_θ. In this paper, we consider the model to be a neural network classifier $f_\theta : \mathcal{X} \to \Delta^c$, where f_θ is a learned function that maps an input data sample $x \in \mathcal{X}$ to a probability distribution over c classes, where Δ^c denotes the c-dimensional simplex. Given a dataset D, we denote by $f_\theta \leftarrow \mathcal{T}(D)$ the process of training a model f_θ, parameterized by weights θ, using a training algorithm $\mathcal{T}$ on dataset D.

In early literature [24, 58], membership inference is defined via a security game in which the adversary is asked to determine the membership of a single instance. Recent work [9] refined this definition to more precisely characterize the adversary's capabilities, particularly whether the adversary is allowed to train shadow models using query instances. Inspired by [9], we instantiate the membership inference security game to make it match the typical experimental setting of using half members and half non-members for evaluation, and define the following canonical security game:

Definition 1 (Canonical 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 randomly selects a member set $D_a \subseteq D$ and samples a nonmember set D_b from $\mathbb{D}$, such that $D_b \cap D = \emptyset$ and $|D_a| = |D_b|$. These two sets are combined to create a query set: $D_{\text{query}} = D_a \cup D_b$, which the challenger then sends to the adversary.*
3. *The adversary responds with a set $D_g \subseteq D_{\text{query}}$, which represents that the adversary guesses that instances in D_g are used when training f_θ, and instances in $D_{\text{query}} \setminus D_g$ 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 its parameters. Specifically, the adversary can query the model to obtain softmax output probabilities (using the default softmax temperature of 1) for any input instance. State-of-the-art MIAs [3,9,19,53,59] leverage shadow models [47] to analyze how the model's outputs change in response to discrepancies between its members versus non-members. We assume that the adversary can construct a dataset by sampling from the data distribution $\mathbb{D}$, and use subsets of this dataset to train shadow models. Depending on when the shadow models are trained, we consider the following two threat models.

Non-Adaptive Setting. In this setting, the adversary is only allowed to train shadow models *before* the adversary learns the query set D_{query} (*i.e.,* before step 2 in Definition 1). That is, the adversary constructs an attack dataset $D_{\text{adv}}^{\text{non-adapt}}$ (from which the adversary will sample subsets to train shadow models) by sampling from $\mathbb{D}$, *i.e.,* $D_{\text{adv}}^{\text{non-adapt}} \sim \mathbb{D}$. For most practical classification tasks, the sizes of $D_{\text{adv}}^{\text{non-adapt}}$ 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, for most query instances, the adversary can only observe the model's behavior when the query point is not part of the target model's training set, leading to shadow *out* models available for inference.

This non-adaptive (offline) setting has attracted growing interest from recent studies [2, 19, 30, 59] because it models a realistic attack scenario: the adversary receives a sequence of membership queries and must respond without incurring the cost of retraining shadow models for each query. **In this paper, we also follow these studies and mainly focus on the non-adaptive setting.**

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). The adversary can therefore leverage D_{query} for shadow training. For each query instance, the adversary can train both shadow *in* models (trained with the instance) and shadow *out* models (trained without it), and exploit their behavioral differences for membership inference.

Recent studies [2, 43, 59] criticize this setting as unrealistic in practice, as it requires training new shadow models for each batch of query instances. Nevertheless, we show in Section 3.1 that our proposed method can be adapted to the adaptive setting, achieving state-of-the-art performance while significantly reducing computational overhead.

High Computational Cost in Both Settings. State-of-the-art MIAs in both settings require training a large number of shadow models to achieve their best performance. For instance, LiRA [3] in the adaptive setting and PMIA [9] in the non-adaptive setting both suggest training 256 shadow models. The practical implication of this is severe: a single LiRA run on CIFAR-10 can take around 6 days on a modern A100 GPU using its official implementation. Such high computational demands make it infeasible for practical privacy auditing [59] and difficult to reproduce research results. It has become common practice for subsequent studies [19, 30, 53] to use a computationally cheaper version of LiRA (*e.g.,* with 64 or 128 shadow models) as their strongest baseline for evaluation, acknowledging the original version is too costly to run. However, since LiRA's performance degrades significantly with fewer shadow models in the adaptive setting (as shown in [59] and evidenced in Section 4.2), this compromise leads to skewed evaluations against a weakened baseline.

Assumptions. In this paper, we adopt the assumption used almost universally in the MIA literature [3, 9, 53, 59] that the adversary has access to an oracle that can sample from the data distribution $\mathbb{D}$ (from which the training dataset D is sampled). We also adopt the experimental settings in the MIA literature. Specifically, in the adaptive setting, the adversary is given the membership query set D_{query}, which is constructed using all training instances in D as members and an equal number of non-member instances (both sampled from $\mathbb{D}$), *i.e.,* $D_a = D$ and $|D_b| = |D|$ (the equal-size assumption). In the non-adaptive setting, the adversary can sample a dataset $D_{\text{adv}}^{\text{non-adapt}}$ from $\mathbb{D}$, and that $D_{\text{adv}}^{\text{non-adapt}} \cap D_{\text{query}} = \emptyset$ (the disjoint assumption). We emphasize that the equal-size assumption and the disjoint assumption are used only to align with prior experimental setups; the effectiveness of our attack

does not depend on them. Discussion of attacks without these assumptions is provided in Appendix B.3.

3 Imitative Membership Inference Attack

Motivation of Imitative MIA. Our work is motivated by a core limitation in most MIAs: the high predictive variance of membership signals from their target-agnostic shadow models. These shadow models, trained by randomly sampling from the adversary's dataset, only capture the general patterns of members and non-members across models, failing to account for the specific target model being attacked. To address this, we introduce *imitative training*, which leverages the target model's knowledge to train imitative models that explicitly mimic the target's behavior. This design yields more informative membership signals and, ultimately, more effective attacks. The next subsection describes our approach in detail, and Section 4.3 demonstrate its effectiveness and efficiency.

3.1 Attack Method

We begin by introducing the membership signal, loss function, and pivot data used for imitative training. Next, we outline the imitative training process. Finally, we present the workflow of IMIA, and detail how it can be applied to the adaptive setting.

Scaled Confidence Score. We follow [3] and define the membership signal of query instance (x, y) on model f as:

$$\phi\left(f(x)_y\right) = \log\left(f(x)_y\right) - \log(\max_{y' \neq y} f(x)_{y'}), \tag{1}$$

where $f(x)_y$ denotes the model's output probability for the true class y, and $\max_{y' \neq y} f(x)_{y'}$ is the highest probability among all incorrect classes. This signal captures the model's prediction confidence and is widely used in prior MIAs [3, 9, 53], as it is a more effective indicator of membership than alternatives such as output loss.

Imitation Loss Function. Instead of training shadow models that are independent of the target model, we train imitative models to explicitly imitate the behavior of the target model by aligning their outputs. To achieve this, we introduce the *imitation loss function* that minimizes the discrepancy between the output probabilities of the imitative model and the target model. Formally, for a given instance (x, y), let $f_\psi(x)_i$ and $f_\theta(x)_i$ be the softmax outputs of the imitative model f_ψ and the target model f_θ on the i-th class, respectively. The imitation loss is defined as follows:

$$\mathcal{L}_{\text{imitate}}(x, y, f_\psi, f_\theta) = \sum_{i=1}^{c} w_i(c) \left(\log(f_\psi(x)_i) - \log(f_\theta(x)_i)\right)^2, \tag{2}$$

where c is the total number of classes, and $w_i(c)$ is a class-specific weight that determines the importance of each class:

$$w_i(c) = \begin{cases} \frac{1+\sqrt{c}}{c+2\sqrt{c}}, & \text{if } i = y \text{ or } i = \text{argmax}_{y' \neq y} f_\theta(x)_{y'}, \\ \frac{1}{c+2\sqrt{c}} & \text{else.} \end{cases}$$

Algorithm 1 Imitative Training. We first train an imitative *out* model by matching its output with the target model over T_1 epochs. We then continue training for T_2 epochs on pivot data to obtain an imitative *in* model.

Require: Target model f_θ, imitation dataset D_{imitate}, pivot dataset D_{pivot}, epochs $T_{\text{warmup}}, T_1, T_2$, learning rate η

Initialize a model f_ψ with randomized parameters Ψ
 # train imitative out model
for epoch $= 1, \ldots, T_1$ **do**
  for each batch B in D_{imitate} **do**
    if epoch $\leq T_{\text{warmup}}$ **then**
      # warm-up with cross-entropy loss
      $L \leftarrow \frac{1}{|B|} \sum_{(x,y)\in B} \mathcal{L}_{\text{ce}}(x, y, f_\psi)$
    else
      $L \leftarrow \frac{1}{|B|} \sum_{(x,y)\in B} \mathcal{L}_{\text{imitate}}(x, y, f_\psi, f_\theta)$ ▷ *Eq. (2)*
    end if
    $\psi \leftarrow \psi - \eta \nabla_\psi L$
  end for
  if epoch $= T_{\text{warmup}}$ **then**
    $f \leftarrow f_\psi$ ▷ *warmup checkpoint*
  end if
end for
$f_{\text{out}} \leftarrow f_\psi$ ▷ *save as imitative out model*
 # continue training to obtain imitative in model for pivots
Initialize $f_\psi \leftarrow f$
for epoch $= 1, \ldots, T_2$ **do**
  for each batch B in D_{pivot} **do**
    $L_{\text{ce}} \leftarrow \frac{1}{|B|} \sum_{(x,y)\in B} \mathcal{L}_{\text{ce}}(x, y, f_\psi)$ ▷ *cross-entropy loss*
    $\psi \leftarrow \psi - \eta \nabla_\psi L_{\text{ce}}$
  end for
end for
$f_{\text{in}}^{\text{pivot}} \leftarrow f_\psi$ ▷ *save as imitative in model*
return f_{out}, $f_{\text{in}}^{\text{pivot}}$

Here, the weighting scheme prioritizes the probability corresponding to the ground-truth class, y, and the most likely incorrect class, as predicted by the target model. This design encourages imitative models to focus on the most indicative signals, which aligns with our attack signal in Equation (1).

Our approach can be interpreted as a weighted version of *logits matching* in model distillation [25], with a weighting strategy specifically tailored to membership inference. In Section 4.4, we compare it to the commonly used Kullback-Leibler (KL) divergence [20] and also explore other formulations of the imitation loss to demonstrate its effectiveness. A more detailed discussion of the connections to model distillation is provided in the next subsection.

Selecting Pivot Data. In the non-adaptive setting, the adversary trains models before accessing the query instance, making it infeasible to obtain the *in* behaviors for membership queries. To address this, we construct a pivot dataset D_{pivot} from the adversary's dataset $D_{\text{adv}}^{\text{non-adapt}}$ and use their

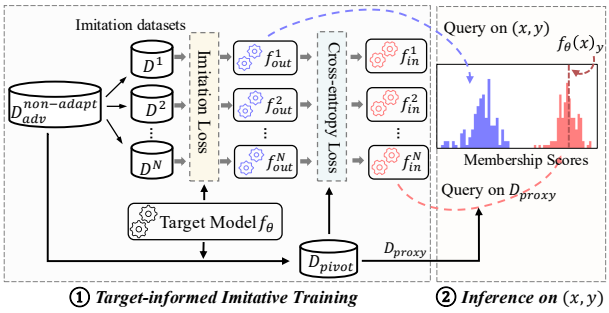


Figure 2: Demonstration of IMIA in the non-adaptive setting. The adversary ① constructs imitative *in* and *out* models by mimicking the behaviors of the target model f_θ, ② performs membership inference on query instance (x, y) using trained imitative models and selected proxy data.

behaviors to approximate the *in* behaviors of query instances. An effective pivot dataset should (i) cover all classes of the data distribution, (ii) facilitate fast convergence of imitative models to improve efficiency. To construct such a dataset, we adopt a simple yet effective heuristic: we query the target model f_θ with all instances from the adversary's dataset, compute their losses, and select the k (set as 100 by default) instances with the lowest losses from each class to form D_{pivot}. Our ablation study in Section 4.4 shows IMIA is robust to this heuristic. While this simple approach works well in our experiments, more sophisticated pivot-selection strategies may further improve performance; we leave this as future work.

Imitative Training. The imitative training procedure is outlined in Algorithm 1. It requires query access to the target model and utilizes two datasets selected from the adversary's data to mimic the target model's behavior: an imitation dataset D_{imitate} and a pivot dataset D_{pivot}. The training proceeds in two stages. In the first stage, we warm up the model using the standard cross-entropy loss for T_{warmup} epochs, after which we switch to the proposed imitative loss and train for T_1 epochs to align the model's outputs with those of the target model on D_{imitate} (lines 3–17). Since query instances are not in the adversary's dataset in the non-adaptive setting, the resulting model serves as an imitative *out* model, imitating how the target model behaves on non-member inputs. In the second stage, we resume training from the warmup checkpoint for T_2 epochs on a pre-selected pivot dataset using the standard cross-entropy loss (lines 19–26). This yields an imitative *in* model for the pivot data, whose behaviors on these instances will serve as proxies to approximate how the target model behaves when a query is in the training set.

Attack Workflow. The workflow of IMIA is outlined in Algorithm 2 and Figure 2, consisting of two phases: the prepare and inference phases. In the prepare phase, we select a pivot set from the adversary's dataset $D_{\text{adv}}^{\text{non-adapt}}$, and train N imitative *in* and imitative *out* models via imitative training (lines

Algorithm 2 Imitative Membership Inference Attack (IMIA) in the non-adaptive setting. The adversary first performs a one-time prepare phase, selecting pivot data and training N imitative models via the proposed imitative training. During inference, the adversary queries each imitative *out* model to obtain its *out* confidence distribution for the query instance, then approximates its *in* confidence distribution using proxies selected from pivot data.

Require: Target model f_θ, adversary's dataset $D_{\text{adv}}^{\text{non-adapt}}$, query instance $(x, y) \in D_{\text{query}}$

1: `# Prepare Phase: Imitative Training`
2: $\mathcal{F}_{\text{out}} \leftarrow \{\}$, $\mathcal{F}_{\text{in}}^{\text{pivot}} \leftarrow \{\}$
3: $D_{\text{pivot}} \leftarrow \texttt{SelectPivot}(f_\theta, D_{\text{adv}}^{\text{non-adapt}})$ ▷ *select pivots*
4: **for** N times **do**
5: $\quad D_{\text{imitate}} \sim D_{\text{adv}}^{\text{non-adapt}}$ ▷ *sample a dataset*
6: $\quad$ *# train imitative models, see Algorithm 1*
7: $\quad f_{\text{out}}, f_{\text{in}}^{\text{pivot}} \leftarrow \texttt{ImitativeTrain}(f_\theta, D_{\text{imitate}}, D_{\text{pivot}})$
8: $\quad \mathcal{F}_{\text{out}} \leftarrow \mathcal{F}_{\text{out}} \cup \{f_{\text{out}}\}$
9: $\quad \mathcal{F}_{\text{in}}^{\text{pivot}} \leftarrow \mathcal{F}_{\text{in}}^{\text{pivot}} \cup \{f_{\text{in}}^{\text{pivot}}\}$
10: **end for**
11: `# Inference Phase: Query on` (x, y)
12: $\mathcal{S}_{\text{out}} \leftarrow \{\}$, $\mathcal{S}_{\text{in}} \leftarrow \{\}$
13: *# collect out confidence scores*
14: **for** each $f_{\text{out}} \in \mathcal{F}_{\text{out}}$ **do**
15: $\quad \mathcal{S}_{\text{out}} \leftarrow \mathcal{S}_{\text{out}} \cup \{\phi(f_{\text{out}}(x)_y)\}$
16: **end for**
17: *# collect in confidence scores via proxies, see Section 3.1*
18: $D_{\text{proxy}} \leftarrow \texttt{FindProxy}(D_{\text{pivot}}, (x, y))$ ▷ *find proxies*
19: **for** each $f_{\text{in}}^{\text{pivot}} \in \mathcal{F}_{\text{in}}^{\text{pivot}}$ **do**
20: $\quad$ **for** each $(u, v) \in D_{\text{proxy}}$ **do**
21: $\quad\quad \mathcal{S}_{\text{in}} \leftarrow \mathcal{S}_{\text{in}} \cup \{\phi(f_{\text{in}}^{\text{pivot}}(u)_v)\}$
22: $\quad$ **end for**
23: **end for**
24: $\bar{s}_{\text{out}} \leftarrow \texttt{mean}(\mathcal{S}_{\text{out}})$
25: $\bar{s}_{\text{in}} \leftarrow \texttt{mean}(\mathcal{S}_{\text{in}})$
26: $s_{\text{obs}} \leftarrow \phi(f_\theta(x)_y)$ ▷ *query target model*
27: **return** $\Lambda = (s_{\text{obs}} - \bar{s}_{\text{out}})^2 - (s_{\text{obs}} - \bar{s}_{\text{in}})^2$

2-10). During the inference, we first collect the *out* distribution of scaled confidence scores for the query instance using the imitative *out* models (lines 13-16). Next, we select a set of proxy instances from the pivot set that are similar to the query and gather their *in* distributions of scaled confidence scores via imitative *in* models (lines 18-23). Finally, we estimate the means for both confidence distributions and compute a non-parametric score Λ based on their distances to the query's confidence score on the target model (lines 24–27).

The computation of the score Λ can be interpreted as a Gaussian likelihood ratio test [3] with fixed variance, which allows us to frame the inference task as a hypothesis testing. Note that the prepare phase can be done before accessing

query instances and only needs to be performed once to answer any queries using the same set of imitative models. During the inference phase, no additional models are trained. The efficiency analysis of each phase is detailed in Section 4.2.

Finding Proxy. During membership inference, we retrieve a set of proxy samples from the pivot dataset (line 18 in Algorithm 2) to approximate the *in* behavior of the query instance. Prior work [9] proposes several proxy-selection heuristics at global, class, and instance granularities. We find that class-level selection consistently offers strong performance while remaining computationally efficient. Therefore, we adopt this strategy: for a given query instance, we retrieve all samples from D_{pivot} belonging to the same class and use their output distributions to approximate the query's *in* behavior.

IMIA in the Adaptive Setting. While IMIA is primarily designed for the non-adaptive setting, it can also be applied to the adaptive attack setting. As shown in Algorithm 3, for a query instance (x,y), we train N pairs of imitative *in* models (trained with (x,y)) and imitative *out* models (trained without (x,y)) on random subsets of the adversary's dataset $D_{\text{adv}}^{\text{adapt}}$ (lines 2–9). This procedure directly learns the target model's behavior when the query instance is in its training set, thus eliminating the need for a pivot dataset and proxy selection. The final score is computed by comparing the query's confidence score on the target model to the mean scores of the imitative *in* and *out* distributions (lines 11-14). In addition, while the algorithm is described for a single query, the procedure can be easily parallelized across all query instances using the same set of imitative models. Specifically, we follow [3] and construct the subsets such that each instance $(x,y) \in D_{\text{query}}$ appears in $N/2$ subsets, and train N imitative models on these subsets. This ensures that the same N imitative models are used to infer the membership for all instances in D_{query}.

3.2 Discussion of IMIA

Non-parametric Modeling. State-of-the-art MIAs [3,9] rely on *parametric modeling* to compute final scores. For instance, LiRA [3] estimates the likelihood ratio by fitting Gaussian distributions to scaled confidence scores. However, accurately estimating the parameters of these distributions requires training a large number of shadow models, and the performance degrades significantly when computational resources are constrained [9,59]. To address this, IMIA adopts a non-parametric view and computes a score by measuring the squared distance between attack signals derived from imitative *in* and imitative *out* models. This design avoids the cost of parameter estimation, enabling IMIA to outperform state-of-the-art MIAs while using less than 5% of their computational cost.

It is worth noting that the core idea behind our approach (*i.e.,* imitative training) is not limited to low-resource settings. As demonstrated in Section 4.2, when computational resources are available (*e.g.,* training 256 models as in LiRA), adapting IMIA to use LiRA's parametric modeling achieves even greater performance. This adaptability highlights the effectiveness and versatility of our imitative training framework across a broad spectrum of computational budgets.

Algorithm 3 Imitative Membership Inference Attack (IMIA) in the adaptive setting. The adversary trains N imitative *in* and imitative *out* models for the query instance.

Require: Target model f_θ, adversary's dataset $D_{\text{adv}}^{\text{adapt}}$, query instance $(x,y) \in D_{\text{query}}$

1: $\mathcal{S}_{\text{out}} \leftarrow \{\}$, $\mathcal{S}_{\text{in}} \leftarrow \{\}$
2: **for** N times **do**
3: $\quad D_{\text{tmp}} \sim D_{\text{adv}}^{\text{adapt}}$ ▷ *sample a dataset*
4: $\quad D_{\text{out}} \leftarrow D_{\text{tmp}} \setminus \{(x,y)\}$
5: $\quad D_{\text{in}} \leftarrow D_{\text{tmp}} \cup \{(x,y)\}$
6: $\quad$ *# train imitative models, see Algorithm 1*
7: $\quad f_{\text{out}}, f_{\text{in}} \leftarrow \texttt{ImitativeTrain}(f_\theta, D_{\text{out}}, D_{\text{in}})$
8: $\quad \mathcal{S}_{\text{out}} \leftarrow \mathcal{S}_{\text{out}} \cup \{\phi(f_{\text{out}}(x)_y)\}$
9: $\quad \mathcal{S}_{\text{in}} \leftarrow \mathcal{S}_{\text{in}} \cup \{\phi(f_{\text{in}}(x)_y)\}$
10: **end for**
11: $\bar{s}_{\text{out}} \leftarrow \texttt{mean}(\mathcal{S}_{\text{out}})$
12: $\bar{s}_{\text{in}} \leftarrow \texttt{mean}(\mathcal{S}_{\text{in}})$
13: $s_{\text{obs}} \leftarrow \phi(f_\theta(x)_y)$ ▷ *query target model*
14: **return** $\Lambda = (s_{\text{obs}} - \bar{s}_{\text{out}})^2 - (s_{\text{obs}} - \bar{s}_{\text{in}})^2$

Connections with Model Distillation. The imitative training procedure in IMIA is conceptually related to model distillation, a technique used in several prior MIAs [15,30,34,57]. However, IMIA differs from these works in several key aspects: (i) *Tailored loss function.* While prior attacks employ standard KL divergence for distillation, IMIA introduces the imitation loss that is specifically designed to distill the structural logit information most indicative of membership. (ii) *Dual behavior modeling.* Previous MIAs that utilize model distillation only model the target's behavior on non-members. In contrast, our two-phase imitative training first creates an imitative *out* model and then fine-tunes it on pivot data to create an imitative *in* model. This second phase explicitly encodes membership signals while preserving the target model's knowledge, allowing the adversary to take advantage of both in and out behaviors to mount a more powerful attack. (iii) *Significant performance gain.* In our experiments, we observe that prior distillation-based MIAs achieve subpar performance. In contrast, IMIA significantly outperforms all existing attacks while requiring substantially less computational overhead.

4 Evaluation

4.1 Experimental Setup

Datasets. We use four image datasets (*i.e.,* MNIST [28], Fashion-MNIST [54], CIFAR-10 [26], and CIFAR-100 [26]) for our main experiments. To demonstrate the generalizability

of IMIA, we also report attack results on two widely used non-image datasets (*i.e.,* Purchase and Texas [47]) in Section 4.5. The dataset descriptions are provided in Appendix A.1.

Network Architecture. We consider four widely used neural network architectures for image classification: ResNet [18], VGG16 [48], DenseNet121 [21], and MobileNetV2 [46]. To make these models compatible with evaluated datasets, we follow [9] and apply several dataset-specific modifications: we adopt ResNet-18 for MNIST and FMNIST, use WideResNet-28 for CIFAR-10 and CIFAR-100, and adjust the input channels of these model architectures to accommodate both grayscale and RGB image datasets. We follow the training configurations used in prior work [3] to mitigate overfitting. Specifically, we use the SGD algorithm with a learning rate of 0.1, momentum of 0.9, and weight decay [27] set to 5×10^{-4}. We also adopt a cosine learning rate schedule [38] for optimization and apply data augmentation techniques [60] during training. Nevertheless, in experiments, we find that some trained models still exhibit overfitting and yield low validation accuracy; in such cases, we rerun these models with different random seeds whenever validation accuracy falls below a threshold. The threshold, along with the accuracy of the target models, is reported in Appendix A.2.

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

- *LOSS* [58] uses the loss of the query instance for inference.
- *Entropy* [49] leverages a modified prediction entropy estimation for membership inference.
- *Calibration* [52] employs a technique called difficulty calibration, which adjusts the query instance's loss by calibrating it on shadow models.
- *Attack-R* [57] compares the loss of the query instance on the target model to its losses on shadow models. The final score is the proportion of shadow models for which the target model yields a lower loss.
- *Attack-D* [57] uses model distillation [20] to train shadow models and computes the same ratio scores as Attack-R.
- *SeqMIA* [30] leverages model distillation to replicate the learning process of the target model, extracting various membership metrics from its loss trajectory for attack.
- *LiRA* [3] adapts different strategies for two attack settings. In the adaptive setting, it trains both shadow *in* and shadow *out* models and applies a likelihood ratio test for inference. In the non-adaptive setting, it uses only shadow *out* models to conduct a one-sided hypothesis test.
- *Canary* [53] improves upon LiRA by employing adversarial optimization to enhance inference performance.
- *GLiRA* [15] enhances LiRA with model distillation to obtain shadow *out* models for a one-sided hypothesis test.
- *RMIA* [59] performs multiple pairwise likelihood ratio tests between the query instance and randomly selected population instances to compute the score.
- *RAPID* [19] trains a neural network for membership inference by combining raw and calibrated loss values [52].
- *PMIA* [9] approximates the posterior odds test by using behaviors of the shadow model's training set as proxies.

While all attacks can be applied in the non-adaptive setting, some methods are either not applicable in the adaptive setting (*i.e.,* GLiRA and PMIA) or exhibit the same performance in both settings (*i.e.,* LOSS, Entropy). Thus, we only evaluate them in the non-adaptive setting. LiRA and Canary employ different strategies for two settings; we distinguish between their versions based on the experimental context. Attack-D, SeqMIA, and GLiRA utilize distillation to train shadow models from the target model; we include them to demonstrate the effectiveness of imitative training over model distillation.

Evaluation Procedures. We divide each dataset into two disjoint subsets: the query set D_{query} and the auxiliary set $D_{\text{auxiliary}}$. The target model is trained on a randomly sampled subset of D_{query}. In the adaptive setting, we follow [3,53] and provide the adversary with the same data as the query set to prepare their attack (*i.e.,* $D_{\text{adv}}^{\text{adapt}} = D_{\text{query}}$). In the non-adaptive setting, consistent with [9, 19, 59], the adversary's dataset is set to the auxiliary set (*i.e.,* $D_{\text{adv}}^{\text{nonadapt}} = D_{\text{auxiliary}}$), ensuring that the adversary cannot access queries when training shadow models. Details about the data split are provided in Table 18.

Evaluation Metrics. We use the following metrics:

- *TPR@0%FPR.* This metric reflects the extent of privacy leakage under the strictest constraint, measuring the highest true positive rate where the attack makes no false positives.
- *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. For baselines, we train the recommended number of shadow models as specified in their papers. For instance, LiRA trains 256 shadow models to estimate the parametric likelihood ratio (see Appendix A.3 for a complete list of shadow model counts). In contrast, IMIA uses a fixed set of $N = 10$ imitative models to highlight its computational efficiency. For imitative training, we first warm up the models for $T_{\text{warmup}} = 50$ epochs on MNIST and Fashion-MNIST, and $T_{\text{warmup}} = 80$ epochs on CIFAR-10 and CIFAR-100. After the warmup, we train the imitative *out* models for $T_1 = 100$ epochs across all datasets, matching the number of epochs used in shadow training [3]. For the imitative *in* models, we continue training for $T_2 = 20$ using pivot data. For CIFAR-10 and CIFAR-100, we follow [3,9,53] by querying each instance with 18 random augmentations to compute the averaged membership score. Regarding hyperparameter configuration, we utilize the default settings from the original baseline implementations without further tuning across all experiments. We acknowledge this as a limitation; while these defaults may not be optimal for every dataset, we maintain this approach as tuning every baseline is computationally prohibitive. The final results are reported as the average metrics over ten runs.

Table 1: Performance comparison of *non-adaptive* attacks on ResNet across four image datasets (*i.e.,* MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100). IMIA significantly outperforms the strongest baseline (underlined, PMIA [9]) while requiring less than **5%** of its computational cost (see Section 4.2 for efficiency comparison). The %Imp. indicates the relative improvement of IMIA compared to the strongest baseline. The best result is in bold.

Method	TPR @ 0% 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%	0.01%	0.05%	0.00%	0.00%	52.87%	60.13%	59.97%	75.11%
Entropy	0.00%	0.00%	0.00%	0.00%	0.07%	0.05%	0.00%	0.00%	53.09%	60.06%	59.87%	74.90%
Calibration	0.18%	0.00%	0.05%	0.18%	0.47%	1.30%	0.74%	3.01%	52.21%	55.31%	57.74%	68.55%
Attack-R	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	52.58%	58.05%	60.02%	72.49%
Attack-D	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	52.58%	59.77%	59.83%	76.60%
SeqMIA	0.05%	0.00%	0.04%	0.05%	0.31%	1.30%	0.51%	3.21%	50.36%	55.34%	57.18%	69.07%
LiRA	0.16%	0.16%	0.30%	0.36%	0.34%	1.00%	0.94%	2.71%	50.43%	52.80%	58.14%	67.41%
Canary	0.10%	0.12%	0.36%	0.43%	0.32%	0.84%	0.93%	2.72%	50.34%	52.70%	58.40%	67.33%
GLiRA	0.07%	0.07%	0.24%	0.11%	0.36%	0.92%	0.90%	3.88%	51.06%	55.57%	55.87%	73.87%
RMIA	0.17%	0.00%	0.03%	0.11%	0.49%	1.14%	0.83%	3.59%	53.20%	58.57%	60.04%	72.78%
RAPID	0.21%	0.00%	0.04%	0.25%	0.48%	1.30%	0.74%	3.09%	52.46%	55.38%	57.79%	68.63%
PMIA	0.30%	0.25%	0.62%	0.89%	0.77%	3.51%	1.84%	5.01%	53.23%	60.51%	60.05%	77.64%
IMIA	**1.01%**	**2.52%**	**1.45%**	**2.10%**	**1.86%**	**5.08%**	**3.42%**	**7.32%**	**54.14%**	**61.22%**	**61.08%**	**79.52%**
%Imp.	236.67%	908.00%	133.87%	135.96%	141.56%	44.73%	85.87%	46.11%	1.71%	1.17%	1.72%	2.42%

Table 2: Performance comparison of *adaptive* attacks on ResNet across four image datasets. IMIA outperforms the strongest baseline (underlined) while requiring less than **5%** of its computational cost. The best result is in bold.

Method	TPR @ 0% 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	0.16%	0.89%	0.19%	2.98%	0.59%	2.10%	0.78%	8.49%	52.26%	55.13%	57.39%	68.91%
Attack-R	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	52.95%	58.84%	60.71%	77.57%
Attack-D	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	52.59%	59.34%	59.62%	75.81%
SeqMIA	0.13%	0.76%	0.11%	1.63%	0.36%	1.82%	0.61%	5.84%	51.92%	52.17%	58.30%	69.15%
LiRA	0.80%	3.85%	1.30%	5.41%	2.01%	6.03%	2.60%	18.03%	54.01%	62.04%	61.03%	81.39%
Canary	0.77%	3.72%	1.31%	5.48%	1.96%	6.17%	2.63%	18.32%	53.97%	61.98%	61.01%	81.47%
RMIA	0.56%	2.05%	0.21%	0.21%	0.91%	3.76%	0.64%	5.10%	53.86%	61.28%	61.08%	79.46%
RAPID	0.46%	1.96%	0.25%	0.42%	0.95%	3.51%	0.82%	5.26%	53.26%	61.42%	60.56%	79.73%
IMIA	**1.33%**	**4.62%**	**2.33%**	**8.52%**	**2.35%**	**7.10%**	**3.61%**	**19.82%**	**54.25%**	**62.43%**	**61.30%**	**82.94%**
%Imp.	66.25%	20.00%	77.86%	55.47%	16.92%	15.07%	37.26%	8.19%	0.44%	0.63%	0.44%	1.80%

4.2 Evaluation of IMIA

In this section, we benchmark IMIA against state-of-the-art (SOTA) non-adaptive and adaptive MIAs, compare with attacks that use model distillation, and analyze its performance under varying computational budgets.

Performance in the Non-Adaptive Setting. We first evaluate IMIA in the non-adaptive setting, and the results on ResNet are shown in Table 1. IMIA consistently outperforms all existing methods across the evaluated metrics. For instance, on MNIST, it achieves a true positive rate (TPR) of 1.01% at zero false positive rate (FPR), which is at least three times as high as the best-performing baseline (*i.e.,* PMIA). The advantage is even more pronounced on Fashion-MNIST, with a 2.52% TPR at the same FPR, over ten times higher than baselines. These results represent a substantial advancement, as such datasets were previously considered hard to attack. Although the improvement in balanced accuracy is relatively modest, this is consistent with prior studies [3, 57, 59]. Moreover, the consensus in the MIA community is that attacks should be evaluated using TPR at low FPR, where IMIA shows a clear advantage. Similar trends are observed on other model architectures; results and Receiver Operating Characteristic (ROC) curves are provided in Appendix B.4.

Performance in the Adaptive Setting. We also evaluate IMIA with SOTA MIAs in the adaptive setting. The results for ResNet are shown in Table 2, while performance on other architectures can be found in Appenix B.4. IMIA consistently achieves the best attack performance across all datasets, surpassing strong attacks like LiRA and Canary. For instance, IMIA achieves nearly double the performance of LiRA on CIFAR-10 at a TPR of 0% FPR, with similar improvements observed across other datasets.

Comparison with Distillation-based MIAs. While the imitative training used in IMIA is conceptually related to model

Table 3: Runtime comparison of leading MIAs and IMIA, measured in hours on a cluster of one A100 GPU. The computation reduction (%) is calculated against the leading attacks in each setting (PMIA for non-adaptive, LiRA for adaptive).

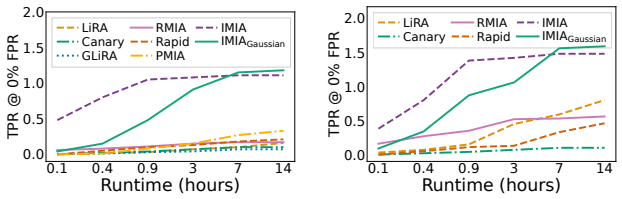
	MNIST	FMNIST	CIFAR-10	CIFAR-100	Purchase	Texas
Non-Adaptive Setting						
LiRA	14.30	14.41	142.32	284.62	2.92	3.22
Canary	22.46	23.34	182.22	336.67	10.59	4.10
GLiRA	8.51	8.65	72.62	144.39	1.53	1.68
RMIA	9.26	9.50	79.79	156.48	1.58	1.74
Rapid	11.21	11.35	88.42	163.35	1.62	1.77
PMIA	14.31	14.43	143.08	287.14	3.01	3.23
IMIA	**0.64**	**0.65**	**6.72**	**13.43**	**0.12**	**0.13**
%Reduction	95.53%	95.50%	95.30%	95.32%	96.01%	95.98%
Adaptive Setting						
LiRA	14.40	14.62	142.43	287.85	2.94	3.38
Canary	22.69	23.51	182.95	337.42	11.03	4.13
RMIA	10.45	10.98	80.02	154.23	1.59	1.75
Rapid	11.34	11.63	89.17	165.73	1.65	1.79
IMIA	**0.72**	**0.73**	**6.96**	**14.08**	**0.14**	**0.16**
%Reduction	95.00%	95.01%	95.11%	95.11%	95.24%	95.27%

Table 4: Runtime breakdown (in hours) of MIAs on MNIST. The preparation cost refers to the time for training shadow (imitative) models, while the inference cost denotes the time for inferring the membership of query instances. GLiRA and PMIA are applicable only in the non-adaptive setting.

	LiRA	Canary	GLiRA	RMIA	RAPID	PMIA	IMIA
Non-Adaptive Setting							
Preparation	14.22	8.57	8.43	8.35	10.59	14.22	**0.62**
Inference	0.08	13.89	0.08	0.91	0.62	0.09	**0.02**
Adaptive Setting							
Preparation	14.26	8.61	-	8.38	10.72	-	**0.71**
Inference	0.14	14.08	-	1.07	0.62	-	**0.01**

distillation, it significantly outperforms distillation-based attacks (*i.e.,* Attack-D, SeqMIA, and GLiRA) in both adaptive and non-adaptive settings. For instance, Attack-D yields negligible TPR at low FPRs, and in the non-adaptive setting, IMIA outperforms SeqMIA and GLiRA by at least a factor of five. We attribute this performance gap to key limitations in existing distillation-based MIAs: (i) These attacks directly apply the standard model distillation process [20], failing to capture the salient membership signals that are important for MIA. (ii) They use distillation to create shadow *out* models, which can only mimic the target's behavior on non-member instances. In contrast, IMIA explicitly caputer membership signals by training both imitative *in* and *out* models, leveraging their behavioral discrepancies for a more effective attack.

Efficiency Analysis. We measure the overall runtime of IMIA in both adaptive and non-adaptive settings on a cluster with a single A100 GPU, comparing it to leading MIAs. We do not include simple baselines (*e.g.,* LOSS attack [58]) for comparison, as their performance is far inferior to IMIA. As shown in Table 3, IMIA requires substantially less computation. For



(a) Non-adaptive setting. (b) Adaptive setting.

Figure 3: Performance on MNIST under varying computational budgets. $\text{IMIA}_{\text{Gaussian}}$ employs a Gaussian likelihood ratio [3] on imitative models to compute the final scores.

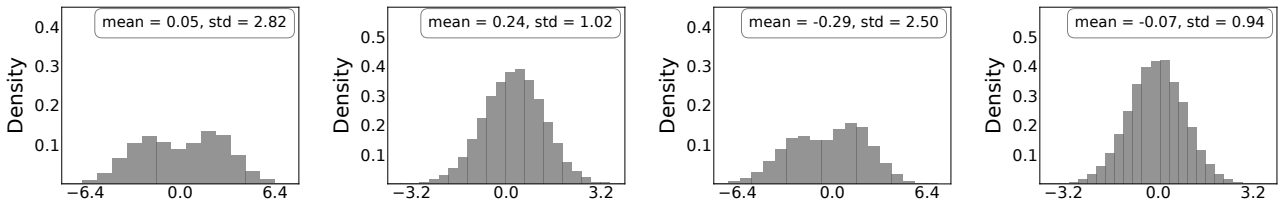
instance, on CIFAR-10, the SOTA baselines (*i.e.,* PMIA and LiRA) require 6 days to perform their attack, whereas IMIA completes in less than 7 hours, a reduction of more than 95%. This efficiency is consistently observed across all datasets.

In Table 4, we also present a runtime breakdown on MNIST, separating the computation spent on preparation (*i.e.,* model training) and inference phases. The results confirm that model training is the bottleneck for most MIAs, and IMIA mitigates this by requiring far fewer models. During inference, while most MIAs are relatively fast, IMIA can even outperform them by computing a simple non-parametric score for inference.

Performance Across Computational Budgets. To evaluate the trade-off between efficiency and attack effectiveness, we benchmark IMIA against leading MIAs using an A100 GPU in both non-adaptive and adaptive settings. We also introduce $\text{IMIA}_{\text{Gaussian}}$, a variant that leverages the same imitative training as IMIA but employs parametric modeling for inference. Specifically, it follows LiRA [3] and fits the scaled confidence scores of imitative models as Gaussian distributions, and uses the likelihood ratio as the membership score.

Figure 3 illustrates the performance under different computational budgets (equivalent to 1 to 256 shadow models). IMIA demonstrates superior performance across all settings and is particularly dominant in low-resource scenarios (*e.g.,* under 3 hours of training). In contrast, state-of-the-art MIAs typically require more computation to achieve their optimal performance. For example, LiRA in the adaptive setting shows substantial improvement when trained for 14 hours (*i.e.,* 256 shadow models), and evaluating it with fewer models would underestimate its efficacy. Additionally, we observe that while $\text{IMIA}_{\text{Gaussian}}$ underperforms IMIA when computational resources are limited, it surpasses IMIA as more computation is available. This further underscores the versatility of the imitative training paradigm: imitative models can effectively mimic the target model's behavior, and increasing their number, in combination with parametric modeling, allows for better capture of subtle behavioral discrepancies between members and non-members, leading to more powerful attacks.

The Stability of Attacks. We also investigate the performance stability of our attack. Specifically, we train 10 target



(a) Residuals of target-agnostic **shadow** models on *members*.
(b) Residual of target-informed **imitative** models on *members*.
(c) Residual of target-agnostic **shadow** models on *non-members*.
(d) Residual of target-informed **imitative** models on *non-members*.

Figure 4: Normalized residual distributions (defined in Section 4.3) of shadow vs. imitative models on CIFAR-100. The residuals of imitative models closely follow the standard normal distribution, indicating they better mimic the target model's behavior.

Table 5: Performance stability on MNIST in the non-adaptive setting. All values are reported as percentages (%).

	TPR @ 0%FPR	**TPR @ 0.1%FPR**	**Balanced Accuracy**
LiRA	0.16 ± 0.10	0.34 ± 0.15	50.43 ± 0.16
Canary	0.10 ± 0.08	0.32 ± 0.12	50.34 ± 0.15
GLiRA	0.07 ± 0.06	0.36 ± 0.08	51.06 ± 0.14
RMIA	0.17 ± 0.04	0.49 ± 0.11	53.20 ± 0.15
RAPID	0.21 ± 0.09	0.48 ± 0.14	52.46 ± 0.10
PMIA	0.30 ± 0.11	0.77 ± 0.18	53.23 ± 0.17
IMIA	$\mathbf{1.01 \pm 0.04}$	$\mathbf{1.86 \pm 0.06}$	$\mathbf{54.14 \pm 0.07}$

models on randomly selected subsets and attack them using leading MIAs and IMIA, then calculate the mean and standard deviation. The results on the non-adaptive setting on MNIST are in Table 5. As observed, existing shadow-based MIAs (*e.g.,* LiRA and PMIA) exhibit high performance variability relative to their mean, whereas our target-informed imitative training approaches show better stability. This demonstrates the reliability and robustness of our attack.

4.3 Effectiveness of Imitative Training

Does Imitative Training Better Mimic the Target Model? We conduct a set of experiments comparing its ability to mimic the target model's behavior against that of shadow training. Specifically, we train 256 shadow models and imitative models in the adaptive setting. We then collect the scaled confidence score distributions for both members and non-members by querying the shadow and imitative models. Prior studies [3, 9, 53] have empirically validated that these score distributions are well-approximated by a Gaussian distribution. Following this approach, we model each distribution as Gaussian and estimate its mean μ and standard deviation σ. Using these parameters, we compute the **normalized residual** of the target model's true score s as $r = \frac{s-\mu}{\sigma}$. If the models' distributions truly replicate the target model's behavior, the residuals r for both members and non-members should follow a standard normal distribution, *i.e.,* $r \sim N(0,1)$.

Figure 4 presents the results for both members and non-members on CIFAR-100. Residuals from imitative models

Table 6: Wasserstein distance between the *in* and *out* distributions of shadow and imitative models. A larger distance indicates that the models can better capture the target model's behavioral discrepancy between members and non-members.

	MNIST	**FMNIST**	**C-10**	**C-100**	**Purchase**	**Texas**
Shadow models	0.01	0.03	0.06	0.21	0.13	0.25
Imitative models	**0.08**	**0.18**	**0.47**	**0.82**	**0.73**	**0.85**

align much more closely with the standard normal distribution than those from shadow models, indicating that imitative models more faithfully reproduce the target model's behavior. In the full version of our paper, we present results for additional image and non-image datasets, along with quantile–quantile plots to assess the alignment with $N(0,1)$ and a likelihood-based analysis to quantify that imitative models better capture the target model's true behavior than shadow models.

Do Imitative Models Enhance Behavioral Discrepancy? Effective imitative models should reliably replicate the behavioral discrepancy between members and non-members of the target model. We quantify this discrepancy by computing the Wasserstein distance between the *in* and *out* distributions of shadow and imitative models, respectively. The results in Table 6 show that the Wasserstein distance of imitative models is significantly larger than that of shadow models across datasets, which indicates that imitative models generate a much more separable signal between members and non-members.

4.4 Ablation Study

Impact of Imitative Training. We evaluate the effectiveness of imitative training by comparing IMIA with two variants: (i) $\text{IMIA}_{\text{distill}}$, which replaces imitative training with standard model distillation using KL divergence as the loss function [20]; and (ii) $\text{IMIA}_{\text{shadow}}$, which substitutes imitative training with target-agnostic shadow training. Table 7 reports results on MNIST and CIFAR-10. Both variants show substantial degradation across all metrics, especially in the low false-positive regime. Using model distillation reduces average performance by over 20%, while replacing our ap-

Table 7: Impact of imitative training in IMIA. We compare against two variants: $\text{IMIA}_{\text{distill}}$, which replaces imitative training with standard model distillation, and $\text{IMIA}_{\text{shadow}}$, which uses target-agnostic shadow training.

	TPR @ 0%FPR		TPR @ 0.1%FPR		Balanced Accuracy	
	MNIST	C-10	MNIST	C-10	MNIST	C-10
Non-Adaptive Setting						
IMIA	**1.01%**	**1.45%**	**1.86%**	**3.42%**	**54.14%**	**61.08%**
$\text{IMIA}_{\text{distill}}$	0.83%	0.96%	1.09%	2.46%	53.13%	60.08%
$\text{IMIA}_{\text{shadow}}$	0.28%	0.54%	0.69%	1.46%	53.20%	59.57%
Adaptive Setting						
IMIA	**1.33%**	**2.33%**	**2.35%**	**3.61%**	**54.25%**	**61.30%**
$\text{IMIA}_{\text{distill}}$	1.04%	1.69%	1.92%	2.78%	53.80%	60.47%
$\text{IMIA}_{\text{shadow}}$	0.63%	1.02%	1.24%	1.78%	53.06%	60.51%

Table 8: Impact of softmax temperature on IMIA.

temperature	TPR @ 0%FPR		TPR @ 0.1%FPR		Balanced Accuracy	
	MNIST	C-10	MNIST	C-10	MNIST	C-10
0	0.72%	0.84%	1.43%	2.03%	54.01%	60.37%
0.5	0.84%	1.12%	1.53%	2.50%	54.08%	60.42%
1	1.01%	**1.45%**	**1.86%**	**3.42%**	54.14%	61.08%
2	**1.04%**	1.43%	1.84%	3.41%	**54.18%**	61.03%

Table 9: Impact of the pivot selection in IMIA on MNIST. Selecting k pivot instances per class with the lowest losses on the target model consistently outperforms random selection.

Selection	k	TPR @ 0%FPR	TPR @ 0.1%FPR	Balanced Accuracy
Loss	50	1.00%	1.84%	51.13%
Random	50	0.89%	1.77%	51.10%
Loss	100	1.01%	1.86%	51.14%
Random	100	0.91%	1.79%	51.12%
Loss	1,000	1.02%	1.85%	51.14%
Random	1,000	0.92%	1.79%	51.11%

Table 10: Different weighting strategies for the imitation loss.

Strategy	TPR @ 0%FPR		TPR @ 0.1%FPR		Balanced Accuracy	
	MNIST	C-10	MNIST	C-10	MNIST	C-10
Non-Adaptive Setting						
uniform	0.85%	1.07%	1.44%	3.14%	53.96%	60.78%
log	0.96%	1.38%	1.79%	3.38%	54.12%	61.05%
square root	**1.01%**	**1.45%**	**1.86%**	**3.42%**	**54.14%**	**61.08%**
linear	0.92%	1.31%	1.75%	3.29%	54.11%	61.02%
Adaptive Setting						
uniform	1.01%	1.98%	2.04%	3.28%	54.16%	61.19%
log	1.29%	**2.35%**	2.28%	3.60%	54.22%	61.27%
square root	**1.33%**	2.33%	**2.35%**	**3.61%**	**54.25%**	**61.30%**
linear	1.27%	2.30%	2.15%	3.48%	54.18%	61.23%

proach with shadow training cuts attack performance by at least half. These results demonstrate that effective imitation should capture the target's membership signals, rather than its general behaviors across models.

We next examine how the softmax temperature of the target model's outputs affects attack performance. This softmax temperature controls the "softness" of the probability distributions used to train the imitative models. A temperature of zero means the imitative models are trained using only the target model's hard labels, while higher temperatures generate smoother distributions that capture more nuanced details of the model's behavior. As shown in Table 8, performance improves when shifting from hard labels to soft labels, with a standard temperature of one typically yielding the best results.

Impact of Pivot Selection. In the non-adaptive setting, the performance of IMIA depends on the selection of a high-quality pivot dataset from $D_{\text{adv}}^{\text{non-adapt}}$ to train imitative *in* models. We investigate two selection strategies: (1) our proposed approach of choosing the k instances per class with the lowest loss on the target model, and (2) a baseline that randomly selects k instances per class. We evaluate both strategies for different values of k, with the results presented in Table 9. As shown, the loss-based selection consistently yields better performance than random selection. Additionally, IMIA is robust to the choice of k, with attack performance remaining stable as k ranges from 50 to 1,000.

Impact of Different Weighting Strategies. We evaluate how different weighting strategies in the imitation loss affect attack performance. We compare three strategies: (i) *uniform*, which assigns equal weight to all logits; (ii) *log*, which assigns weight $\log(c)$ to membership-related logits and 1 to others, where c is the number of classes; (iii) *square root*, which assigns weight $\sqrt{c}$ to membership-related logits (used as the default); (iv) *linear*, which assign weigh c to membership-related logits. The results for both adaptive and non-adaptive settings are presented in Table 10. As shown, ignoring membership signals for the imitative loss (*i.e.,* uniform strategy) leads to a noticeable performance drop. Assigning greater weight to these critical logits generally yields improved results. However, excessively large weights, as seen in the linear strategy, can cause imitative models to focus too narrowly, leading to unstable training and suboptimal performance. Empirically, the square root strategy consistently yields the best results, suggesting it strikes an effective balance.

Impact of Training Epochs. The imitative training consists of two stages: an initial training phase of T_1 epochs for imitative *out* models, followed by an additional T_2 epochs using pivot data to obtain the imitative *in* models. While $T_1 = 100$ is set to match the standard model training, our default of $T_2 = 20$ involves fewer epochs. Here, we analyze the impact of varying T_2 on attack performance. As shown in Table 11, fewer epochs (*e.g.,* $T_2 = 10$) lead to performance degradation in both adaptive and non-adaptive settings, as the model has not fully learned the behaviors of the target model. However, we observe that increasing the number of epochs does not necessarily result in improved performance, particularly in the non-adaptive setting. This may be because training with more epochs using cross-entropy loss causes the model to

Table 11: Impact of imitative training epochs T_2 in IMIA.

T_2	TPR @ 0%FPR		TPR @ 0.1%FPR		Balanced Accuracy	
	MNIST	C-10	MNIST	C-10	MNIST	C-10
Non-Adaptive Setting						
10	0.97%	1.34%	1.72%	3.29%	54.12%	61.01%
20	**1.01%**	**1.45%**	**1.86%**	**3.42%**	54.14%	61.08%
50	0.99%	1.38%	1.75%	3.22%	**51.18%**	**61.13%**
100	0.98%	1.42%	1.77%	3.34%	54.15%	61.07%
Adaptive Setting						
10	1.02%	1.96%	1.94%	2.35%	54.25%	61.33%
20	1.33%	**2.33%**	2.35%	3.61%	54.25%	61.30%
50	1.34%	2.32%	2.37%	**3.69%**	**54.31%**	**61.38%**
100	**1.35%**	2.30%	**2.39%**	3.68%	54.30%	61.34%

Table 12: Impact of different attack signals in IMIA.

Signal	TPR @ 0%FPR		TPR @ 0.1%FPR		Balanced Accuracy	
	MNIST	C-10	MNIST	C-10	MNIST	C-10
Non-Adaptive Setting						
loss	0.41%	0.51%	0.75%	1.24%	53.99%	61.18%
probability	1.01%	1.45%	1.86%	3.42%	54.14%	61.08%
pre-softmax	**1.12%**	**1.57%**	**1.89%**	**3.45%**	**54.22%**	**61.35%**
Adaptive Setting						
loss	0.42%	0.48%	0.65%	1.27%	54.01%	61.22%
probability	1.33%	2.33%	2.35%	3.61%	**54.25%**	61.30%
pre-softmax	**1.35%**	**2.35%**	**2.48%**	**3.70%**	54.20%	**61.31%**

increasingly forget the target-informed behaviors, gradually reverting to the standard shadow training. Therefore, we believe $T_2 = 20$ achieves a good balance, yielding strong attack performance while maintaining training efficiency.

Impact of Different Membership Signals. We analyze how the choice of membership signal affects the performance of IMIA. The signal is used both to train imitative models and to compute final scores for inference. We compare three options: (i) *loss*, which uses the output loss of a query instance; (ii) *probability*, our default approach, which uses the softmax output probabilities; (iii) *pre-softmax*, which uses the outputs of the final layer before softmax. While all black-box MIAs assume access only to probabilities, using pre-softmax activation is also common in the implementation of many attacks [3, 9, 53, 59]. Results for both adaptive and non-adaptive settings are shown in Table 12. As observed, using the prediction loss significantly degrades performance, and using pre-softmax activation instead of probabilities consistently yields better results. We also find that using pre-softmax activation to compute scaled confidence scores is numerically more stable and computationally simpler, as it avoids the logarithmic operation on probability values. These results are consistent with previous studies [3], and we recommend using pre-softmax activation for our attack when available.

Additional Ablation Studies. We also evaluate the robustness of IMIA under varying query budgets, scenarios where the adversary is unaware of the target model's architecture, and cases involving distribution mismatches between the adversary's data and the target's data. Due to space limitations, the results of these experiments are provided in Appendix B.1.

Table 13: Performance comparison on non-image datasets under the *non-adaptive setting*. IMIA outperforms the strongest baseline (underlined) using less than **5%** of its computation.

Method	TPR @ 0%FPR		TPR @ 0.1%FPR		Balanced Accuracy	
	Purchase	Texas	Purchase	Texas	Purchase	Texas
LiRA	0.01%	0.04%	0.06%	0.17%	62.12%	65.03%
Canary	0.01%	0.03%	0.04%	0.19%	62.32%	65.23%
GLiRA	0.02%	0.08%	0.35%	0.24%	66.73%	70.82%
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	<u>0.05%</u>	<u>0.42%</u>	<u>2.28%</u>	<u>5.72%</u>	<u>78.38%</u>	<u>87.10%</u>
IMIA	**0.51%**	**0.82%**	**9.54%**	**10.61%**	**82.45%**	**89.90%**
%Imp.	920.00%	95.24%	318.42%	85.49%	5.19%	3.21%

Table 14: Performance comparsion on non-image datasets under the *adaptive setting*. IMIA outperforms the strongest baseline (underlined) using less than **5%** of its computation.

Method	TPR @ 0%FPR		TPR @ 0.1%FPR		Balanced Accuracy	
	Purchase	Texas	Purchase	Texas	Purchase	Texas
LiRA	1.26%	10.28%	11.33%	24.24%	85.95%	90.10%
Canary	<u>1.28%</u>	<u>10.30%</u>	<u>11.43%</u>	<u>24.36%</u>	<u>85.99%</u>	<u>90.21%</u>
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%
IMIA	**1.98%**	**10.98%**	**16.57%**	**24.88%**	**86.71%**	**90.83%**
%Imp.	54.69%	6.60%	44.97%	2.13%	0.84%	0.69%

4.5 Additional Investigations

Attack on Non-Image Datasets. We evaluate IMIA on two widely-used non-image datasets, Purchase and Texas [47], using a multilayer perceptron (MLP) as the target model and comparing it with leading MIAs. Results in both non-adaptive and adaptive settings are presented in Table 13 and Table 14, respectively. As shown, IMIA consistently outperforms existing attacks, particularly in terms of TPR at low FPR. We notice that all evaluated MIAs perform worse on these non-image datasets than on image datasets like CIFAR-100, despite both having 100 classes. We attribute this discrepancy to the smaller generalization gap in the models trained on non-image data. For instance, our Purchase model shows a modest gap between its training and validation accuracies (97.7% vs. 79.5%), whereas the CIFAR-100 model exhibits a much larger gap indicative of significant overfitting (99.4% vs. 68.9%). Since MIAs typically exploit overfitting for attack, they are less effective against these better-generalized models. Similar observations have been found in prior work [3, 9].

Attack Against DP-SGD. Differentially Private Stochastic Gradient Descent (DP-SGD) [1] is an effective defense mechanism against privacy attacks, including MIAs. Following prior work [3, 19], we evaluate the effectiveness of DP-SGD

Table 15: Effectiveness of using DP-SGD against IMIA on CIFAR-10 in the non-adaptive setting. 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		IMIA
No defense	-	-	90.41%	3.42%
DP-SGD	0	∞	81.46%	1.57%
	0.2	> 1000	49.76%	0.32%
	0.5	31	37.55%	0.15%
	1.0	4	32.10%	0.12%

Table 16: Performance of adaptive attacks against a ResNet model trained on CIFAR-10 using DP-SGD.

	TPR @ 0.1% FPR		Balanced Accuracy	
	$\sigma = 0.2$	$\sigma = 0.5$	$\sigma = 0.2$	$\sigma = 0.5$
Calibration	0.12%	0.11%	50.74%	50.07%
Attack-R	0.00%	0.00%	50.09%	50.06%
Attack-D	0.00%	0.00%	50.13%	50.60%
SeqMIA	0.07%	0.06%	50.05%	50.38%
LiRA	0.17%	0.11%	50.73%	50.22%
Canary	0.19%	0.11%	50.54%	50.28%
RMIA	0.15%	0.10%	50.42%	50.31%
RAPID	0.14%	0.11%	50.81%	50.76%
IMIA	**0.34%**	**0.18%**	**51.05%**	**50.82%**

in defending against IMIA. Specifically, we fix the clipping norm C to 10 and test IMIA on a ResNet model trained on CIFAR-10 under the non-adaptive setting. As shown in Table 15, even applying only gradient clipping (without noise addition) significantly reduces both model accuracy and the effectiveness of our attack. We also assess the impact of DP-SGD in the adaptive setting in Table 16, and observe consistent trends: as the privacy guarantee strengthens, all attacks experience reduced performance, and their performance gaps narrow. Nevertheless, IMIA consistently outperforms existing MIAs, especially in the low false-positive regime.

5 Related Work

Neural networks have been known to be vulnerable to leaking sensitive information about their training datasets. Various attacks [4, 14, 16] have been proposed to quantify the extent of this data leakage and assess the associated risks. In this paper, we focus on the membership inference attack (MIA) [47], which aims to predict whether a specific data instance was included in a target model's training set. MIA has become a widely used tool to audit the data privacy of machine learning models [10, 23, 39, 41, 50]. The first MIA against ML models was introduced by [47], who also proposed the technique of shadow training. MIAs and the shadow training have since been extended to various scenarios, including white-box [29, 40, 44], black-box [24, 31, 35, 36, 45, 49], label-only access [6, 33], and knowledge distillation [22] settings. Existing black-box MIAs can be categorized into the following two groups.

Non-Adaptive MIAs. In this setting, the adversary trains shadow models (or avoids model training entirely) before accessing the membership queries. Several techniques have been introduced, such as score functions [49, 58], difficulty calibration [19, 44, 52], loss trajectory [30, 34], model distillation [15, 57], quantile regression [2], and hypothesis testing [9, 59]. However, these attacks typically rely on target-agnostic shadow training and require training hundreds of shadow models to achieve good performance.

Adaptive MIAs. In the adaptive setting, the adversary can perform instance-by-instance behavioral analysis by training additional predictors [40] or conducting parametric hypothesis testing [3, 53]. A recent work [9] goes beyond individual instance attacks by exploring the membership dependence among query instances. However, these attacks suffer from high computational costs, and recent studies [2, 43, 59] argue that such adaptive attacks are impractical, as they require training new models for each batch of queries.

Model Distillation in MIAs. Model distillation [20, 25] has been used in MIAs in two main approaches: (i) it is used to train more reliable shadow out models for inference, as explored in [15, 57]; (ii) it serves as a tool to simulate the intermediate learning dynamics of the target model, and derive membership signals from this simulated learning process [30, 34]. However, these methods use standard distillation techniques and only model the target's behavior on non-members. In contrast, we propose an imitative training approach tailored for MIAs, which mimics the target model's behavior for both members and non-members, significantly enhancing both attack efficacy and efficiency.

Evaluation of MIAs. Early studies [45, 47, 58] evaluated membership inference as a binary classification task using metrics like balanced accuracy. More recent research [3, 37] has emphasized evaluating attacks by true positive rate (TPR) at a very low false positive rate (FPR), and most MIAs perform poorly under this metric. In line with the current consensus in the MIA literature, we also adopt TPR at low FPRs as the primary evaluation metric in this paper.

6 Conclusion

In this paper, we introduce IMIA, a new MIA built upon a novel imitative training technique to train target-informed imitative models for inference. Imitative training addresses the high variation of shadow models by strategically distilling membership signals from the target model. Extensive experiments in various attack settings demonstrate the effectiveness and efficiency of IMIA. One limitation of our paper is that it mainly focuses on classification models, leaving the generalization to generative models [11, 12, 17] unexplored. Nonetheless, our work provides an efficient and practical attack, opening new directions for future research in MIAs.

Acknowledgments. This work was funded in part by the National Science Foundation (NSF) awards, CNS-2212160, CNS-2504819, CNS-2247794, and CNS-2207204, Amazon Research Award, and CISCO Research Award. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

Ethical Considerations

Stakeholder-Based Analysis. We identify several key stakeholders impacted by this research:

- *Machine Learning Practitioners.* This is our primary audience. We provide ML practitioners with a more effective tool for privacy auditing, allowing them to identify vulnerabilities in their models and improve privacy protections.
- *Data Subjects.* The publication of a more effective MIA could theoretically be used to infer personal information, leading to privacy violations. However, our experiments exclusively use public, non-sensitive datasets, and we do not target any individuals or proprietary models. Moreover, by publishing these attacks and their potential mitigations, as described below, our work helps strengthen the overall privacy and security of the deep learning ecosystem.
- *Adversaries.* Our work could potentially be used by adversaries to exploit privacy vulnerabilities. In recognition of this, we discuss the mitigations below to minimize this risk.

Mitigations. We have taken the following concrete steps to mitigate potential harms:

- *Exclusive Use of Public, Non-Sensitive Data.* Our research was conducted exclusively on public datasets (*e.g.,* CIFAR-10), which do not contain Personally Identifiable Information (PII). We did not use any private or sensitive data, nor did we target specific individuals or proprietary models, ensuring no one's privacy was violated during our research.
- *Validation of Existing Defenses.* We evaluate that established defenses like DP-SGD remain an effective mitigation strategy against IMIA. By demonstrating that practical defenses exist, we lower the privacy risk introduced by our work and provide clear guidance for developers.

Justification for Research and Publication. The vulnerabilities that IMIA exploits are fundamental to current deep learning practices. It is highly likely that adversaries could independently discover similar techniques, and we believe it is crucial to proactively disclose this attack to the research community. By publishing our findings, we enable defenders to address the vulnerabilities in their models before they are exploited in real-world scenarios.

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.

Open Science

To facilitate reproducibility, we have made our research artifacts publicly available. The code repository is hosted at `https://github.com/zealscott/IMIA` and archived at `https://doi.org/10.5281/zenodo.17885393`. The repository contains a detailed `README.md` with step-by-step instructions for environment setup and experiment execution. We also provide automated scripts for training both shadow and imitative models in adaptive and non-adaptive settings. The training process automatically handles the downloading of evaluated datasets. For attack evaluation, a unified interface, `run_attack.py`, allows users to execute IMIA alongside baselines seamlessly. The codebase is modularized for extensibility: model architectures are located in the `models/` directory, attack implementations are in the `attacks/` directory, and hyperparameters are stored in the `config/` directory.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Martin Bertran, Shuai Tang, Aaron Roth, Michael Kearns, Jamie H Morgenstern, and Steven Z Wu. Scalable membership inference attacks via quantile regression. In *Advances in Neural Information Processing Systems*, pages 314–330, 2023.

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

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

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

[6] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974, 2021.

[7] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009.

[9] Yuntao Du, Jiacheng Li, Yuetian Chen, Kaiyuan Zhang, Zhizhen Yuan, Hanshen Xiao, Bruno Ribeiro, and Ninghui Li. Cascading and Proxy Membership Inference Attacks. In *33th Annual Network and Distributed System Security Symposium (NDSS)*, 2026.

[10] Yuntao Du and Ninghui Li. Systematic assessment of tabular data synthesis algorithms. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 3093–3106, 2025.

[11] Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. Are diffusion models vulnerable to membership inference attacks? In *International Conference on Machine Learning*, pages 8717–8730, 2023.

[12] Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*, 2024.

[13] Cynthia Dwork. Differential Privacy. In *Automata, Languages and Programming*, pages 1–12, 2006.

[14] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[15] Andrey V Galichin, Mikhail Pautov, Alexey Zhavoronkin, Oleg Y Rogov, and Ivan Oseledets. Glira: Black-box membership inference attack via knowledge distillation. *IEEE Transactions on Information Forensics and Security*, 2025.

[16] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita 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*, pages 619–633, 2018.

[17] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *arXiv preprint arXiv:1705.07663*, 2017.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Yu He, Boheng Li, Yao Wang, Mengda Yang, Juan Wang, Hongxin Hu, and Xingyu 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*, pages 1226–1240, 2024.

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

[21] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[22] Matthew Jagielski, Milad Nasr, Katherine Lee, Christopher A Choquette-Choo, Nicholas Carlini, and Florian Tramer. Students parrot their teachers: Membership inference on model distillation. In *Advances in Neural Information Processing Systems*, pages 44382–44397, 2023.

[23] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? In *Advances in Neural Information Processing Systems*, volume 33, pages 22205–22216, 2020.

[24] Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. Revisiting membership inference under realistic assumptions. *arXiv preprint arXiv:2005.10881*, 2020.

[25] Taehyeon Kim, Jaehoon Oh, Nakyil Kim, Sangwook Cho, and Se-Young Yun. Comparing Kullback-Leibler Divergence and Mean Squared Error Loss in Knowledge Distillation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 2628–2635, 2021.

[26] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images, 2009.

[27] Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

[28] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1998.

[29] Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated White-Box membership inference. In *29th USENIX security symposium*, pages 1605–1622, 2020.

[30] Hao Li, Zheng Li, Siyuan Wu, Chengrui Hu, Yutong Ye, Min Zhang, Dengguo Feng, and Yang Zhang. SeqMIA: Sequential-metric based membership inference attack. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 3496–3510, 2024.

[31] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. Membership Inference Attacks and Defenses in Classification Models. In *Eleventh ACM Conference on Data and Application Security and Privacy*, pages 5–16, 2021.

[32] Ninghui Li, Wahbeh Qardaji, Dong Su, Yi Wu, and Weining Yang. Membership privacy: A unifying framework for privacy definitions. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 889–900, 2013.

[33] Zheng Li and Yang Zhang. Membership leakage in label-only exposures. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 880–895, 2021.

[34] Yiyong Liu, Zhengyu Zhao, Michael Backes, and Yang Zhang. Membership Inference Attacks by Exploiting Loss Trajectory. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2085–2098, 2022.

[35] Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. ML-Doctor: Holistic risk assessment of inference attacks against machine learning models. In *31st USENIX Security Symposium*, pages 4525–4542, 2022.

[36] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. Understanding membership inferences on well-generalized learning models. *arXiv preprint arXiv:1802.04889*, 2018.

[37] Yunhui Long, Lei Wang, Diyue Bu, Vincent Bindschaedler, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. A pragmatic approach to membership inferences on machine learning models. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 521–534, 2020.

[38] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[39] Sasi Kumar Murakonda and Reza Shokri. ML privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning. *arXiv preprint arXiv:2007.09339*, 2020.

[40] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753, 2019.

[41] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlin. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on security and privacy (SP)*, pages 866–882, 2021.

[42] Michael-Andrei Panaitescu-Liess, Zora Che, Bang An, Yuancheng Xu, Pankayaraj Pathmanathan, Souradip Chakraborty, Sicheng Zhu, Tom Goldstein, and Furong Huang. Can watermarking large language models prevent copyrighted text generation and hide training data? In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 25002–25009, 2025.

[43] Joseph Pollock, Igor Shilov, Euodia Dodd, Yves-Alexandre de Montjoye, and Reviewing Model. Free Record-Level Privacy Risk Evaluation Through Artifact-Based Methods. In *34st USENIX Security Symposium*, 2025.

[44] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pages 5558–5567, 2019.

[45] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *26th Annual Network and Distributed System Security Symposium (NDSS)*, 2019.

[46] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

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

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[49] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium*, pages 2615–2632, 2021.

[50] Shuang Song and David Marn. Introducing a New Privacy Testing Library in TensorFlow. https://blog.tensorflow.org/2020/06/introducing-new-privacy-testing-library.html, 2022.

[51] Yao Tong, Jiayuan Ye, Sajjad Zarifzadeh, and Reza Shokri. How much of my dataset did you use? Quantitative Data Usage Inference in Machine Learning. In *The Thirteenth International Conference on Learning Representations*, 2024.

[52] Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. On the Importance of Difficulty Calibration in Membership Inference Attacks. In *The Tenth International Conference on Learning Representations*, 2022.

[53] Yuxin Wen, Arpit Bansal, Hamid Kazemi, Eitan Borgnia, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Canary in a Coalmine: Better Membership Inference with Ensembled Adversarial Queries. In *The Eleventh International Conference on Learning Representations*, 2023.

[54] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[55] Hanshen Xiao, Zhen Yang, and G Edward Suh. Trustworthy Machine Learning through Data-Specific Indistinguishability. In *Forty-second International Conference on Machine Learning (ICML 2025)*, 2025.

[56] Alexander Xiong, Xuandong Zhao, Aneesh Pappu, and Dawn Song. The Landscape of Memorization in LLMs: Mechanisms, Measurement, and Mitigation. *arXiv preprint arXiv:2507.05578*, 2025.

[57] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3093–3106, 2022.

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

[59] Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-Cost High-Power Membership Inference Attacks. In *International Conference on Machine Learning*, pages 58244–58282, 2024.

[60] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 13001–13008, 2020.

Table 17: Prediction accuracy of used datasets.

Model	MNIST		FMNIST		CIFAR-10		CIFAR-100		Purchase		Texas	
	Train	Val	Train	Val	Train	Val	Train	Val	Train	Val	Train	Val
ResNet	100.0%	99.5%	100.0%	94.8%	99.7%	90.4%	99.4%	68.9%	-	-	-	-
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%	-	-	-	-
MLP	-	-	-	-	-	-	-	-	97.7%	79.5%	96.5%	78.7%

A Experimental Setups

A.1 Dataset Description

MNIST. MNIST [28] 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) [54] contains 60,000 grayscale images for training and 10,000 images for testing. It consists of 10 classes representing different fashion items, such as t-shirts, trousers, and sneakers.

CIFAR10. CIFAR-10 [26] 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 [26] 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 [47] is a dataset of shopping records with 197,324 samples of 600 dimensions. Following previous works [3, 30, 47], we cluster these data into 100 classes to train a Multi-Layer Perceptron (MLP) classifier.

Texas. This dataset is to predict a patient's primary medical procedure using 6,170 binary features. Following [47], the data comprises records from 67,330 patients, with the 100 most frequent procedures used as classification labels.

Table 18: Data partitioning scheme. The query dataset D_{query} and the auxiliary dataset $D_{\text{auxiliary}}$ are strictly disjoint. The target model is trained on a randomly sampled subset from D_{query}. The reference dataset is used by some MIAs [19, 30] to train their attack models.

Dataset	D_{query}		$D_{\text{auxiliary}}$		
	Train	Validation	Train	Validation	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

A.2 Accuracy of the Trained Target Model

We use validation accuracy thresholds to determine whether a trained model needs to be rerun. Specifically, we require a minimum validation accuracy of 0.8 for MNIST and FMNIST, 0.6 for CIFAR-10, and 0.4 for CIFAR-100. If a model fails to meet the corresponding threshold, we retrain it using a different random seed. We report the training and validation accuracies of the target model in Table 17.

A.3 Number of Shadow Models of Baselines

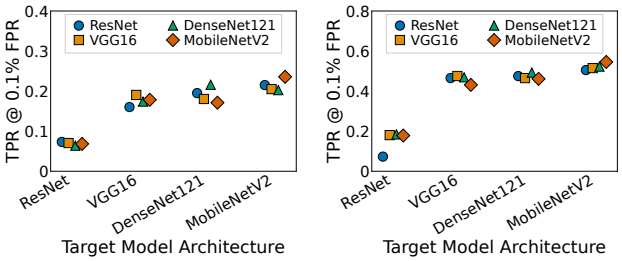
A critical hyperparameter for shadow-based MIAs is the number of trained shadow models. For each baseline method, we adopted the number of shadow models recommended in the original paper, if available. When a number was not specified, we select the maximum number of shadow models until the attack performance saturates to ensure its optimal attack performance. Specifically, we use the following configurations: 10 shadow models for Calibration [52] and SeqMIA [30]; 128 shadow models for Canary [53], GLiRA [15], RMIA [59], RAPID [19]; and 256 shadow models for Attack-R [57], Attack-D [57], LiRA [3], and PMIA [9]. The number of shadow models is consistent across all datasets.

B Additional Experimental Results

B.1 Additional Ablation Studies

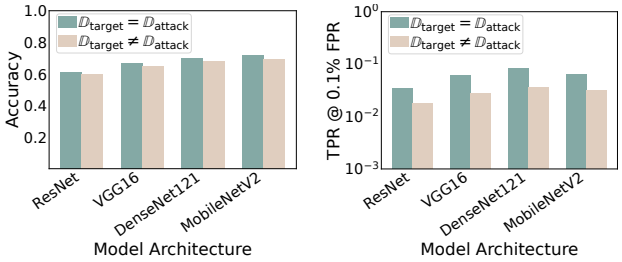
Impact of Query Budget. We evaluate the impact of varying query budgets on CIFAR-10 and CIFAR-100. As shown in Table 19, augmenting random queries (e.g., horizontal flips and shifts by up to ± 4 pixels) improves attack performance. Notably, significant gains occur with just two queries, and performance stabilizes around 18 queries. This observation aligns with findings in prior work [3].

Impact of Mismatched Model Architecture. We evaluate the robustness of IMIA when the adversary is unaware of the exact architecture of the target model. Specifically, we train imitative models using different architectures, and report the results in Figure 5. As expected, the attack performs best when the imitative models match the target model's architecture. However, using a different architecture results in only a slight drop in performance. We attribute this to the imitative training paradigm. By forcing the imitative models to mimic the target's behavior, the training process ensures they develop similar predictive patterns, regardless of their underlying structural differences.



(a) Non-Adaptive setting (b) Adaptive setting

Figure 5: The impact of architecture differences between the target model and the imitative models trained on CIFAR-100.



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

Figure 6: The impact of distribution shift between the target model training dataset and the attacker's dataset.

Attack with Distribution Shift. In realistic attack scenarios, the adversary's data is often not perfectly aligned with the target's data. We follow prior work [3, 19] and conduct the following experiments to evaluate IMIA in such cases:

- $\mathbb{D}_{\text{target}} = \mathbb{D}_{\text{attack}}$. Both the target and imitative 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 CIFAR-10, while the imitative models use the ImageNet [8] portion of the CINIC-10 [7], creating a distribution shift between the target's data and the adversary's data.

The results in Figure 6 show that this distribution shift leads to a noticeable performance decrease, particularly for TPR at 0.1% FPR. This can be attributed to compounded approximation errors: when the adversary's data differs significantly

Table 19: Impact of query budget on IMIA performance (*i.e.,* TPR @ 0%FPR). Querying the model with multiple augmented views enhances attack performance, yielding significant gains with just two queries.

Queries	Non-Adaptive Setting		Adaptive Setting	
	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
1	0.73%	1.07%	1.53%	5.54%
2	1.22%	1.89%	2.06%	7.69%
9	1.43%	2.04%	2.31%	8.42%
18	1.45%	2.10%	2.33%	8.52%
36	1.45%	2.11%	2.33%	8.55%

Table 20: Comparison of attack performance (*i.e.,* TPR @ 0% FPR) between LiRA and IMIA using CMIA [9]. The runtime is measured (in hours) with 9 cascading iterations of both LiRA and IMIA using an A100 GPU.

# Cascading iterations	MNIST		FMNIST		CIFAR-10		CIFAR-100	
	LiRA	IMIA	LiRA	IMIA	LiRA	IMIA	LiRA	IMIA
1	0.80%	1.33%	3.85%	4.62%	1.30%	2.33%	5.41%	8.52%
3	1.05%	1.42%	4.29%	4.86%	2.05%	2.55%	7.49%	10.65%
6	1.46%	1.53%	4.72%	4.92%	2.26%	2.59%	9.30%	11.42%
9	1.53%	1.65%	4.90%	4.98%	2.53%	2.60%	10.62%	11.45%
Runtime (hrs)	128.8	5.8	129.8	5.9	1281.2	60.5	2562.3	120.9

from the target's, the imitative models and proxies become less effective at capturing membership-related behaviors of the target model. Nonetheless, even in this challenging setting, IMIA maintains strong performance on balanced accuracy.

B.2 Incorporating with CMIA

CMIA [9] is a recently proposed attack-agnostic framework that enhances the performance of shadow-based MIAs by iteratively training models on carefully selected datasets. We incorporate this framework with IMIA and compare with LiRA. The result in Table 20 shows that the performance improves with more cascading iterations, particularly during the initial few stages, which is consistent with observations from the original paper. We observe that CMIA provides a much larger boost to LiRA than to IMIA; after 9 iterations, LiRA eventually reaches performance levels comparable to IMIA. We think this is because CMIA trains more reliable shadow models by constructing datasets that resemble the target's training data. This is similar to the goal of imitative training, where we aim to train target-informed imitative models. As a result, CMIA's impact is more noticeable for target-agnostic attacks like LiRA. However, as shown in the table, applying CMIA with LiRA incurs significant computational overhead (over 20× higher than IMIA) and is only feasible in the adaptive setting, which limits its practicality.

Table 21: Attack performance under different overlap ratios between the adversary's dataset and the query set.

# Overlap ratio	**TPR @ 0%FPR**		**TPR @ 0.1%FPR**		**Balanced Accuracy**	
	MNIST	C-10	MNIST	C-10	MNIST	C-10
0	1.01%	1.45%	1.86%	3.42%	54.14%	61.08%
1/10	1.04%	1.44%	1.79%	3.46%	54.12%	61.04%
1/5	1.03%	1.43%	1.84%	3.48%	54.18%	61.10%
1/3	1.08%	1.57%	1.85%	3.52%	54.15%	61.20%

B.3 Impact of Relaxing Assumptions

Equal-size Assumption. While the standard evaluation constructs the query set with an equal number of members and non-members, our attack, like existing MIAs, does not rely on this assumption. First, the training of imitative models is independent of the query set's composition; the adversary randomly selects instances to train imitative *out* models and selects pivot instances to train imitative *in* models regardless of the query set distribution. Second, our primary evaluation metric, TPR at 0% FPR, measures the highest true positive rate while permitting no false positives. This metric remains valid regardless of how the query set is balanced.

Disjoint Assumption. In the non-adaptive setting, prior work [3, 9] assumes that the adversary's dataset and the query set are disjoint, *i.e.,* $D_{\text{adv}}^{\text{non-adapt}} \cap D_{\text{query}} = \emptyset$. This assumption may not hold in realistic scenarios, where high-probability instances can appear in both datasets. IMIA handles this overlap straightforwardly. If a query instance (x, y) appears in the adversary's dataset $D_{\text{adv}}^{\text{non-adapt}}$, we simply discard the imitative *out* models trained with (x, y) and use only those where (x, y) was held out. In expectation, this requires twice as many imitative models to maintain the same number of imitative *out* models per query instance.

We use this approach to evaluate our attack under varying degrees of overlap between $D_{\text{adv}}^{\text{non-adapt}}$ and D_{query}. The results in Table 21 show that attack performance remains stable across different overlap ratios. We also note that this strategy is highly conservative: when substantial overlap exists, one could leverage both the imitative *in* and imitative *out* behaviors for overlapped instances (same as the adaptive version of IMIA) to achieve even stronger performance. Overall, these experiments demonstrate that our attack does not depend on the disjoint assumption and continues to perform reliably when it is relaxed.

B.4 Experiments on Other Models

We extend our evaluation to cover three additional model architectures: VGG16, DenseNet121, and MobileNetV2. For the non-adaptive setting, detailed results are provided in Tables 22 to 24. Due to space constraints, the corresponding ROC curves and the complete results for the adaptive setting are reported in the full version of our paper.

Table 22: Performance comparison of *non-adaptive* attacks on VGG16 across four image datasets.

Method	TPR @ 0% 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%	0.03%	0.14%	52.06%	60.14%	67.38%	88.94%
Entropy	0.00%	0.00%	0.00%	0.02%	0.00%	0.00%	0.02%	0.19%	52.03%	60.08%	67.07%	89.01%
Calibration	0.01%	0.00%	0.91%	1.10%	0.35%	0.10%	2.06%	5.15%	51.20%	54.57%	60.13%	71.17%
Attack-R	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	52.01%	57.64%	66.90%	83.45%
Attack-D	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	52.42%	58.14%	65.64%	86.86%
SeqMIA	0.00%	0.00%	0.14%	0.47%	0.13%	0.06%	1.83%	4.78%	51.39%	57.78%	62.85%	77.02%
LiRA	0.00%	0.00%	1.31%	0.26%	0.06%	0.09%	2.33%	6.77%	50.06%	52.01%	57.59%	74.90%
Canary	0.00%	0.00%	1.35%	0.30%	0.05%	0.08%	2.31%	6.72%	50.01%	52.03%	57.86%	74.98%
GLiRA	0.03%	0.00%	1.10%	0.56%	0.14%	0.11%	2.50%	5.71%	50.05%	58.93%	59.95%	80.09%
RMIA	0.03%	0.02%	1.31%	1.14%	0.38%	0.10%	2.93%	6.00%	52.82%	58.37%	63.16%	78.89%
RAPID	0.00%	0.03%	0.96%	1.24%	0.37%	0.08%	2.04%	5.52%	52.91%	58.25%	64.04%	79.33%
PMIA	0.05%	0.04%	1.44%	2.55%	0.39%	0.12%	5.01%	12.20%	52.07%	60.51%	67.16%	89.03%
IMIA	**0.39%**	**0.34%**	**2.04%**	**5.18%**	**1.40%**	**3.80%**	**6.13%**	**19.05%**	**53.16%**	**60.91%**	**67.89%**	**89.97%**
%Imp.	680.00%	750.00%	41.67%	103.14%	258.97%	3066.67%	22.36%	56.15%	0.47%	0.66%	0.76%	1.06%

Table 23: Performance comparison of *non-adaptive* attacks on DenseNet121 across four image datasets.

Method	TPR @ 0% 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%	0.04%	0.02%	0.03%	0.00%	0.10%	52.27%	60.05%	68.31%	87.90%
Entropy	0.00%	0.00%	0.00%	0.00%	0.03%	0.03%	0.00%	0.09%	52.27%	60.11%	68.04%	88.33%
Calibration	0.05%	0.01%	0.31%	1.74%	0.29%	0.97%	2.31%	3.99%	52.42%	55.27%	60.09%	69.45%
Attack-R	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	52.36%	58.08%	66.74%	85.89%
Attack-D	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	52.57%	58.07%	67.36%	83.91%
SeqMIA	0.01%	0.00%	0.29%	1.26%	0.02%	0.04%	1.04%	2.75%	52.28%	59.68%	63.86%	82.26%
LiRA	0.09%	0.08%	0.81%	3.21%	0.33%	1.02%	2.69%	15.23%	50.49%	52.24%	59.38%	78.81%
Canary	0.08%	0.04%	0.79%	3.36%	0.27%	0.98%	2.50%	14.79%	51.68%	52.74%	59.82%	79.18%
GLiRA	0.02%	0.09%	0.39%	1.53%	0.25%	0.52%	4.45%	9.86%	50.51%	54.05%	62.34%	82.84%
RMIA	0.00%	0.00%	0.45%	1.78%	0.36%	1.23%	2.51%	4.39%	52.29%	58.09%	63.15%	74.58%
RAPID	0.02%	0.03%	0.37%	0.96%	0.29%	0.95%	2.18%	4.28%	51.37%	57.67%	62.19%	72.20%
PMIA	0.23%	0.11%	1.44%	6.03%	0.38%	2.31%	5.37%	18.78%	52.59%	60.14%	68.52%	88.61%
IMIA	**1.01%**	**1.93%**	**4.51%**	**6.59%**	**1.27%**	**7.67%**	**8.42%**	**21.56%**	**52.62%**	**60.46%**	**70.85%**	**88.93%**
%Imp.	339.13%	1654.55%	213.19%	9.29%	234.21%	232.03%	56.79%	14.80%	0.06%	0.53%	3.40%	0.36%

Table 24: Performance comparison of *non-adaptive* attacks on MobileNetV2 across four image datasets.

Method	TPR @ 0% 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%	0.01%	0.01%	0.02%	0.00%	0.10%	55.80%	60.92%	70.01%	89.35%
Entropy	0.00%	0.00%	0.00%	0.02%	0.01%	0.02%	0.00%	0.10%	55.82%	60.50%	70.46%	89.01%
Calibration	0.15%	0.00%	0.44%	1.51%	0.39%	1.01%	2.28%	6.53%	54.03%	55.92%	60.68%	70.28%
Attack-R	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	53.19%	58.41%	67.32%	85.31%
Attack-D	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	53.84%	59.36%	67.12%	86.98%
SeqMIA	0.02%	0.00%	0.08%	0.42%	0.08%	0.12%	0.94%	1.47%	53.12%	55.49%	68.38%	75.40%
LiRA	0.04%	0.01%	0.10%	3.17%	0.14%	0.77%	3.01%	15.34%	50.17%	52.32%	58.91%	80.61%
Canary	0.03%	0.01%	0.12%	3.20%	0.11%	0.72%	3.12%	15.42%	51.02%	52.39%	59.32%	81.95%
GLiRA	0.00%	0.11%	0.45%	0.20%	0.07%	0.13%	3.27%	7.06%	50.08%	54.43%	63.61%	84.47%
RMIA	0.11%	0.00%	0.40%	1.89%	0.40%	1.12%	2.79%	7.83%	55.76%	59.30%	65.11%	74.76%
RAPID	0.06%	0.00%	0.23%	1.73%	0.21%	0.99%	2.05%	6.94%	55.94%	60.41%	70.02%	87.63%
PMIA	0.23%	0.15%	0.46%	4.38%	0.96%	2.67%	4.97%	19.42%	56.31%	61.02%	70.62%	90.02%
IMIA	**1.21%**	**2.80%**	**1.97%**	**6.34%**	**2.41%**	**5.00%**	**6.48%**	**23.57%**	**56.55%**	**61.15%**	**72.27%**	**90.19%**
%Imp.	426.09%	1766.67%	328.46%	44.75%	151.04%	87.27%	30.38%	21.37%	0.43%	0.21%	2.34%	0.19%