Membership Inference Attacks as Privacy Tools: Reliability, Disparity and Ensemble

Zhiqi Wang
Rensselaer Polytechnic Institute
Troy, New York, USA
wangz56@rpi.edu

Chengyu Zhang
Rensselaer Polytechnic Institute
Troy, New York, USA
zhangc26@rpi.edu

Yuetian Chen
Purdue University
West Lafayette, Indiana, USA
yuetian@purdue.edu

Nathalie Baracaldo
IBM Research
USA
baracald@us.ibm.com

Swanand Kadhe
IBM Research
USA
Swanand.Kadhe@us.ibm.com

Lei Yu
Rensselaer Polytechnic Institute
Troy, New York, USA
yul9@rpi.edu

Abstract

Membership inference attacks (MIAs) pose a significant threat to the privacy of machine learning models and are widely used as tools for privacy assessment, auditing, and machine unlearning. While prior MIA research has primarily focused on performance metrics such as AUC, accuracy, and TPR@low FPR—either by developing new methods to enhance these metrics or using them to evaluate privacy solutions—we found that it overlooks the disparities among different attacks. These disparities, both between distinct attack methods and between multiple instantiations of the same method, have crucial implications for the reliability and completeness of MIAs as privacy evaluation tools. In this paper, we systematically investigate these disparities through a novel framework based on coverage and stability analysis. Extensive experiments reveal significant disparities in MIAs, their potential causes, and their broader implications for privacy evaluation. To address these challenges, we propose an ensemble framework with three distinct strategies to harness the strengths of state-of-the-art MIAs while accounting for their disparities. This framework not only enables the construction of more powerful attacks but also provides a more robust and comprehensive methodology for privacy evaluation.

CCS Concepts

• **Computing methodologies → Machine learning**.

Keywords

Membership Inference Attack; Data Privacy; Machine Learning; Privacy Assessment; Evaluation

ACM Reference Format:
Zhiqi Wang, Chengyu Zhang, Yuetian Chen, Nathalie Baracaldo, Swanand Kadhe, and Lei Yu. 2025. Membership Inference Attacks as Privacy Tools: Reliability, Disparity and Ensemble. In . ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, July 2017, Washington, DC, USA
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

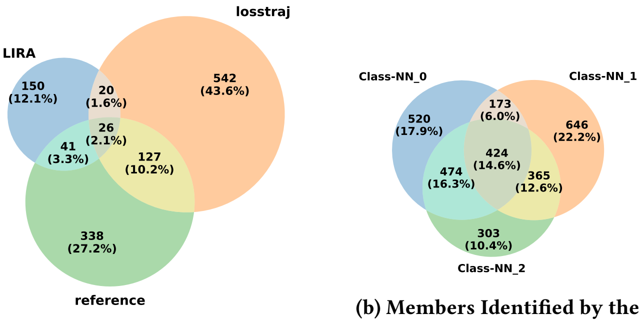
1 Introduction

With the burgeoning development of machine learning (ML) applications, there is an increasing use of sensitive data, including financial transactions, medical records, and personal digital footprints, for training purposes. Numerous studies [39, 42, 53] have highlighted serious privacy risks associated with ML models, such as data extraction [14], membership inference [18], and property inference [15] attacks, primarily due to their capacity to memorize training datasets.

Membership inference attacks (MIAs) on ML models aim to determine if a specific data sample was used to train a target model or not. These attacks have received significant attention and are widely studied in ML privacy research. Beyond highlighting membership inference as a critical privacy threat, they are also frequently employed as evaluation tools across a broad range of privacy-related tasks and research efforts. These include:

- Privacy Risk Assessment: MIAs have been increasingly utilized to examine privacy risks in various machine learning contexts, such as generative adversarial networks [6], explainable ML [29], diffusion models [32], federated learning [36], large language models [4, 33], and multi-modal models [10]. MIAs are also applied across diverse applications such as social media networks [28], recommendation systems [57], and clinical models [19].
- Privacy Auditing: MIAs are often used as empirical tools for privacy auditing to quantify privacy leakage [21, 35]. With their underlying privacy notion closely tied to differential privacy (DP), MIAs have been used to validate the bounds of DP algorithms [37] and debug their implementations [48].
- Machine Unlearning Verification: Machine unlearning [2] involves removing the influence of a data item from a model to ensure privacy and compliance. MIAs are often used to assess whether a sample has been unlearned or not [23].
- Benchmarking performance of privacy-enhancing methods: Because of the effectiveness of MIAs in the above tasks, they are widely used to evaluate and benchmark the effectiveness of various privacy-preserving solutions [41, 49], DP algorithms [11], and unlearning methods [13, 38].

Due to the critical nature of these tasks, extensive research efforts are being made to develop more effective and powerful MIAs [18]. These advances are important to ensure a more accurate and comprehensive assessment of privacy risks, auditing, and unlearning verification. While *balanced accuracy* and *AUC* are commonly used



(a) Members Identified by Different Attacks

(b) Members Identified by the Same Attack's Different Instances

Figure 1: Venn diagram of member sets detected by (a) different attacks at a low FPR (0.1), (b) different instances of the Class-NN attack (with the same auxiliary dataset) at a low FPR (0.1). All Attacks are done on CIFAR-10.

to measure the performance of MIAs, Carlini et al. [3] argue that these aggregate metrics often do not correlate with success rates at low false positive rates (FPRs), which are crucial for a practically meaningful evaluation of MIA effectiveness. Therefore, the true positive rate at low FPR (TPR@low FPR) has become the standard metric for evaluating the "practical effectiveness" of MIAs. In recent works on MIAs [3, 30, 50, 55], both aggregate metrics and TPR@low FPR are used to evaluate and demonstrate the superiority of their proposed methods over prior attacks.

In this paper, however, we argue that the evaluation, even with all these metrics, may still not capture a complete picture of MIA performance. To elaborate on this, consider a target model F_T trained on dataset $\mathcal{D}$ and two attack instances $\mathcal{A}_a$ and $\mathcal{A}_b$ having the same FPR. Suppose $\mathcal{D}_a$ and $\mathcal{D}_b$ represent the member subsets that can be detected by $\mathcal{A}_a$ and $\mathcal{A}_b$, respectively. Even if $\mathcal{A}_a$ performs better than $\mathcal{A}_b$ in both aggregate metrics and TPR@low FPR, relying only on $\mathcal{A}_a$ may not reliably assess privacy risks and verify unlearning outcomes associated with $\mathcal{D}_b \setminus \mathcal{D}_a$. For illustration, Figure 1a shows a Venn diagram of member subsets detected by three different attacks LiRA [3], Loss Trajectory [30], and Reference Attack [51], with the same FPR in our MIA experiment on CIFAR-10. The minimal overlap among them indicates that different attacks may implicitly target different subsets of members. This observation highlights a potential limitation in the common practice of favoring one attack over another in privacy-related tasks based solely on performance metrics. Better metrics do not necessarily imply a greater overall capability of an MIA, as a sample undetected by a "stronger" attack may still be exposed by another. It raises two important questions relevant to ongoing MIA privacy research and practice:

- **Q1**: Should the effectiveness of an MIA be judged solely based on those traditional metrics [10, 18]? More broadly, should research on developing new MIAs primarily focus on improving performance metrics while overlooking the member detection disparities between different methods?
- **Q2**: Is it sufficient for privacy evaluations to rely on a single "top-performing" MIA based on performance metrics without accounting for the disparities between different MIAs?

In this paper, we argue that the significant disparities in member detection at the sample level across different MIAs should not be overlooked when evaluating their effectiveness and employing them as tools for privacy assessment.

In addition, *reliability* and *consistency* are essential attributes that MIAs must possess to function effectively as privacy evaluation tools. Most existing works [10, 46, 47] that utilize MIA for privacy assessment and machine unlearning verification employ a single instance of MIA in their experiments. However, the construction of these MIAs involves inherent randomness, associated with data shuffling/sampling and training shadow/attack models, where randomness stems from factors such as optimization, weight initialization, and data batching. It has been shown that the training of randomly initialized neural networks explores different modes in the function space [12]. Therefore, factors involving randomness inevitably lead to different decision boundaries for membership detection, often resulting in significant variance in attack outcomes among different instances of the same attack with the same auxiliary knowledge. Figure 1b shows that for the same attack, Class-NN attack [44], three different instances that are trained on the same shadow dataset with different random seeds have large non-overlapping member sets. This indicates that the attack outcome can be highly sensitive to the randomness of attack construction. This raises another common issue in current research that uses MIAs for privacy assessment and performance evaluation:

- **Q3**: Is it sufficient to evaluate and report results based solely on a single instance of an MIA—as is common in existing works—while disregarding the disparities among instances that naturally arise from randomness in attack construction?

In this paper, we argue that using MIAs in their current form for evaluation, without accounting for these disparities, may lead to incomplete or potentially unreliable results.

To address these concerns, this paper first systematically investigates the disparities among different MIA methods and their instances. We propose a novel framework that introduces *coverage* and *stability* analysis to evaluate and quantify the disparities of MIA methods through multiple attack instances constructed with different random seeds. Our extensive experiments highlight significant issues of instability and disparity inherent in MIAs. To better understand these disparities, we analyze the signals and features used by different MIAs to determine membership and the influence of randomness in their constructions. Our analysis reveals that different attacks may focus on samples with distinct characteristics, resulting in divergent member detection outcomes.

Furthermore, we propose an ensemble framework with three different strategies to address disparity issues in MIAs. It integrates different MIA methods from distinct perspectives: coverage-oriented, stability-oriented, and majority-oriented. These strategies combine multiple random instances of each MIA and further integrate different MIA methods to account for detection disparities. This framework not only enables the construction of more powerful attacks by leveraging the diverse strengths of existing MIAs and incorporating future advancements, but also provides an evaluation protocol to enhance the comprehensiveness of privacy evaluation. Our extensive experiments demonstrate that these ensemble strategies achieve higher performance in traditional metrics. For example, compared to the top-performing MIA, our ensemble improves the ROC AUC and balanced accuracy by 36% and 24%, respectively, and increases the TPR at 0.1% FPR by a factor of five on CIFAR-10. In addition, we discuss and evaluate practical strategies to reduce the computational cost of the ensemble.

Beyond the metrics, our ensemble strategies and their pronounced increase in attack performance serve as *constructive proof* of the issues raised in Q1, Q2, and Q3. Specifically, a "less powerful" but high-disparity MIA remains valuable for uncovering privacy risks that are overlooked by other attacks and can further improve overall effectiveness through the ensemble (Q1). Relying on a single attack or instance, even one considered state-of-the-art, may underestimate true membership privacy risks, as members undetected by one attack may still be exposed by another (Q2, Q3). This has concrete implications for privacy practitioners and researchers applying MIAs in machine unlearning, privacy auditing, and defense evaluation: current evaluation practices that rely on a single attack instance may be unreliable, since they fail to capture the full spectrum of vulnerabilities posed by inherent disparities in MIAs. We conclude this paper with a discussion of these implications and actionable directions for future MIA research, advocating for holistic consideration of disparities and applying ensemble strategies as an evaluation protocol to enable more reliable and comprehensive privacy assessments. The source code is accessible at https://github.com/RPI-DSPlab/mia-disparity.

2 Background and Related Work

Membership Inference Attacks (MIAs) aim to identify whether or not a specific sample was used as training data for a target model. This paper focuses on black-box attacks in which attackers can only query the target model to obtain a prediction for a data point and use it to infer membership. In addition, attackers are able to leverage an auxiliary dataset that comes from a similar distribution as the training set of the target model. Formally, given a target sample x, a target model F_T trained on the dataset $\mathcal{D}_T$, and an auxiliary dataset $\mathcal{D}_A$, membership inference attack $\mathcal{A}$ can be defined as:

$$\mathcal{A}(F_T, \mathcal{D}_A, x, \phi) \rightarrow \{0, 1\} \tag{1}$$

Here ϕ represents a feature extraction function applied to samples, and $\mathcal{A}$ uses $\phi(x)$ as a signal to determine the membership of x, where 1 indicates x is a member, i.e., $x \in \mathcal{D}_T$, and 0 indicates otherwise. For simplicity, x represents a sample and its ground truth class as a pair (x, y). Current MIAs utilize various feature extraction functions ϕ, such as loss [8, 52], full confidence vector output [44] of F_T, or the loss trajectory [30]. As a typical intermediate step, $\mathcal{A}$ assigns a membership score $\text{Score}_{\mathcal{A}}(x)$ to every sample x, and compares it with a threshold to decide the membership.

2.1 Representative MIAs

MIA has been developed widely for different applications, language models, etc. In this paper, we focus on a number of representative MIAs that have been widely used for privacy evaluation and assessment, unlearning, or as the comparing target for developing better MIAs against them.

LOSS Attack (Yeom et al. [52]) This method considers an instance x a member of the training set if the loss of F_T on x is less than a global threshold set as the average loss across the training set. Formally, let $\ell(x, F_T)$ be the loss of F_T on instance x. The LOSS attack predicts $x \in \mathcal{D}_T$ if:

$$\ell(x, F_T) < \frac{1}{|\mathcal{D}_T|} \sum_{x' \in \mathcal{D}_T} \ell(x', F_T) \tag{2}$$

The right-hand side of (2) serves as the threshold for membership prediction. For each sample x, its MIA score is computed as 1 minus its normalized loss on F_T.

Class-NN (Shokri et al. [44]) This attack involves training class-specific neural networks as membership classifiers for each class using data from shadow models. The adversary divides $\mathcal{D}_A$ into subsets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$, then further split each subset to $\mathcal{D}_k^{in}$ and $\mathcal{D}_k^{out}$. Subsequently, k shadow models $f_1, f_2, \ldots, f_k$ are trained on $\mathcal{D}_1^{in}, \mathcal{D}_2^{in}, \ldots, \mathcal{D}_k^{in}$. An attack dataset can be constructed as

$$\{(f_i(x), y, \text{in}) | x \in D_i^{in}, \forall i \in k\} \cup \{(f_i(x), y, \text{out}) | x \in D_i^{out}, \forall i \in k\} \tag{3}$$

To train the attack classifier C_j for class j, it finds entries in the attack dataset where $(f_i(x), y, \text{in/out}), y = j$. Then it uses those entries to train C_j for each class j. To determine if a sample x belongs to the training set of F_T, the adversary queries $C_{j=y}$ with $F_T(x)$, where y is the label of x. The MIA score of this attack is the logit of the attack classifier C_j in sample x_j being a member.

Augmentation Attack (Choquette-Choo et al. [8]) This label-only attack uses data augmentation techniques to generate translated versions of data points, querying a shadow model trained on $\mathcal{D}_A$ to gather predictions which train an attack classifier C. To infer membership for a data point x, it generates translated versions $\{\hat{x}_1, \ldots, \hat{x}_n\}$, queries them on the target model F_T to obtain predictions $\{F_T(\hat{x}_1), \ldots, F_T(\hat{x}_n)\}$, and use these predictions to make inferences using C. The MIA score is the logit of the attack classifier's prediction $C(x)$ being a member.

Difficulty Calibration Loss Attack (Watson et al. [50]) This attack improves traditional loss-based attacks by calibrating membership scores using losses from both the target and shadow models, accounting for difficulty. It queries both the target model F_T and shadow model (trained on $\mathcal{D}_A$) f_s on all $x \in \mathcal{D}_T$, producing two sets of predictions: $\hat{y}^T$ and $\hat{y}^s$. The losses for each prediction, ℓ^T and ℓ^s, are computed using cross-entropy loss. The uncalibrated membership scores ℓ^T are adjusted by computing $s^{cal} = \ell^T - \ell^s$. The threshold τ for determining membership is done by optimizing the prediction accuracy by splitting the $\mathcal{D}_A$ to members (trainset for shadow model) and non-members. This time, the target model becomes the model we use to calibrate. And τ is selected by optimizing the accuracy of losses of $\mathcal{D}_A$ on the shadow model f_s calibrated by the target model F_T. Similar to the Loss Attack, the MIA scores are the $1 - \vec{\ell}$ where $\vec{\ell}$ are normalized calibrated losses of $\mathcal{D}_T$.

LiRA (Carlini et al. [3]) This Likelihood Ratio-based Instance-specific attack computes the likelihood ratio of losses for models trained with and without a particular instance, determining membership based on a threshold that optimizes attack effectiveness. For each instance x, let $\mathcal{D}_{A,x}$ and $\mathcal{D}_{A,\bar{x}}$ be the subsets of $\mathcal{D}_A$ with and without x, respectively. The adversary trains shadow models $\{f_{x,1}, f_{x,2}, \ldots, f_{x,m}\}$ on random subsets of $\mathcal{D}_{A,x}$, and $\{f_{\bar{x},1}, f_{\bar{x},2}, \ldots, f_{\bar{x},n}\}$ on random subsets of $\mathcal{D}_{A,\bar{x}}$. The likelihood ratio for x is then computed as:

$$LR(x) = \frac{\prod_{i=1}^{m} p(\ell(x, f_{x,i}) \mid x \in \mathcal{D}_T)}{\prod_{i=1}^{n} p(\ell(x, f_{\bar{x},i}) \mid x \notin \mathcal{D}_T)} \tag{4}$$

where $p(\cdot \mid x \in \mathcal{D}_T)$ and $p(\cdot \mid x \notin \mathcal{D}_T)$ are the probability density functions of the losses conditioned on x being a member or non-member of $\mathcal{D}_T$, respectively. The adversary then chooses a threshold τ for the likelihood ratio that optimizes the effectiveness of the attack, especially aiming for a low false-positive rate. The MIA score of LiRA is the likelihood of x being a member.

Reference Attack (Ye et al. [51]) This attack (Attack R) uses a similar approach to LiRA by Carlini et al. [3]. It prepares m shadow models $\{f_{x,1}, f_{x,2}, \ldots, f_{x,m}\}$ on $\mathcal{D}_\mathrm{A}$ with different train-test partition. It calculates the membership score as:

$$\mathrm{Pr}_{\theta'}\left(\frac{\mathrm{Pr}(x|\theta)}{\mathrm{Pr}(x|\theta')} \geq 1\right) \tag{5}$$

where $\mathrm{Pr}(x|\theta')$ is the likelihood (confidence) of sample x evaluated on all shadow models $\theta' \in \{f_{x,1}, f_{x,2}, \ldots, f_{x,m}\}$, and θ is the target model F_T. Similar to LiRA, the MIA score is the likelihood of x being a member.

Loss Trajectory Attack (Liu et al. [30]) This attack monitors the change in the loss of each sample over multiple epochs, using knowledge distillation and cross-entropy loss to track and compare loss trajectories for membership inference. It involves training a shadow model f_s on $\mathcal{D}_A$ and applying knowledge distillation [17] on both f_s and F_T with saving the checkpoints f^I at each epoch I over n training epochs, for capturing the loss trajectory for each sample. For each sample $x \in \mathcal{D}_A$, its loss trajectory $\ell(x, f_s)$ can be obtained using each distillation checkpoint of the shadow model. We collect all loss trajectories to construct an attack training set similar to (3) to train an attack classifier C. For a target sample x, the loss trajectory $\ell(x, F_T)$ is obtained using the distillation checkpoints of the target model F_T. The classifier C is then queried with $\ell(x, F_T)$ to determine membership. The MIA score is C's output logit on the loss trajectory of x for predicting it as a member.

2.2 MIA Performance Metrics.

The following metrics are commonly used to evaluate the performance of MIAs [3, 8, 30, 50].

- **Balanced Accuracy** measures the accuracy of membership predictions on a test set with balanced priors (equal numbers of members and non-members).
- **ROC** (Receiver Operating Characteristic) curve plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold levels, providing a comprehensive view of the trade-offs between MIA's TPR and FPR.
- **AUC** (Area Under the ROC Curve) provides a single scalar value summarizing the overall performance of an attack. A higher AUC indicates that the attack is better at distinguishing members from non-members across all thresholds.
- **TPR@Low FPR** focuses on the practical effectiveness of MIAs. A low false positive rate imposes a constraint on membership predictions, requiring the model to be more "cautious" when predicting members to minimize false alarms.

3 Disparity Evaluation Methodology

In this section, we present the metrics and methodology to evaluate disparities of MIAs at both the instance level and the method level.

3.1 Instance Level Disparity Over Randomness

Most MIAs [3, 30, 44, 51] for deep learning rely on the shadow training technique, which trains multiple shadow models on an auxiliary dataset to replicate the behavior of the target model. This process inherently involves randomness from several sources, including the partitioning of the auxiliary dataset into member and non-member sets, weight initialization in the training algorithm, and data shuffling and batching. These factors introduce variability in the outcomes of both shadow models and attack classifiers, ultimately affecting the detection outcomes of MIAs. In our study, we abstract this randomness using a single random seed, representing a random MIA instance that an attacker might create using the same algorithm but under different randomness sources in real-world scenarios.

Disparity Metric: To evaluate an MIA's instance-level disparity in member detection, we introduce *consistency* score, which quantifies the similarity of membership predictions between attack instances using pairwise *Jaccard Index*. The Jaccard Index (or Jaccard Similarity) measures the similarity between finite sample sets and is defined as the size of the intersection divided by the size of the union of the sample sets.

Given a set of random seeds S, we create $|S|$ number of instances for an MIA $\mathcal{A}$, its consistency on target dataset $\mathcal{D}$ is defined as the average of Jaccard Index between every pair of attack instances $\mathcal{A}^i$ and $\mathcal{A}^j$ on their detected member sets $\mathbb{M}_\mathcal{D}(\mathcal{A}^i)$ and $\mathbb{M}_\mathcal{D}(\mathcal{A}^j)$, i.e.,

$$\mathrm{Consistency}_S^\mathcal{D}(\mathcal{A}) = \frac{1}{\binom{|S|}{2}} \sum\nolimits_{i<j}^{i,j\in S} J(\mathbb{M}_\mathcal{D}(\mathcal{A}^i), \mathbb{M}_\mathcal{D}(\mathcal{A}^j)) \tag{6}$$

where the Jaccard Index $J(\mathbb{M}(\mathcal{A}^i), \mathbb{M}(\mathcal{A}^j)) = \frac{|\mathbb{M}(\mathcal{A}^i)\cap\mathbb{M}(\mathcal{A}^j)|}{|\mathbb{M}(\mathcal{A}^i)\cup\mathbb{M}(\mathcal{A}^j)|}$.

A lower consistency score indicates greater discrepancy in the detected member sets across different instances of the same attack, even when provided with identical auxiliary information and target model. We use this metric to measure the variance in an MIA method's membership detection outcomes due to randomness in its construction, indicating that evaluations based on a single instance may not reliably capture an MIA method's true privacy risks. For the LOSS attack, which uses a fixed global threshold and does not involve randomness in its construction, its consistency score is 1.

Coverage and Stability: Due to the discrepancy in member detection across random instances of an MIA, evaluations based on a single random instance—as is common in most existing works—cannot fully capture the true privacy risk posed by an MIA method at the method level under the same auxiliary knowledge, as opposed to the leakage revealed by a specific instantiation. Consequently, such evaluations may provide an incomplete picture of the effectiveness of a privacy solution. To address this limitation, we introduce the evaluation measures *coverage* and *stability*.

The union of true positive attack results from multiple instances of an MIA constructed with different random seeds, referred to as the **coverage** of an attack. Formally, given an attack $\mathcal{A}$ and a set of possible seeds S, for each random seed $s \in S$, we can construct an instance of $\mathcal{A}$ with randomness generated from random seed s, denoted by $\mathcal{A}^s$. The membership prediction for the data point x is $\mathcal{A}^s(x)$. The coverage of attack $\mathcal{A}$ is represented as

$$\text{Coverage}_{\mathcal{S}}(\mathcal{A}) = \left\{ x \in \mathcal{D}_T : \bigcup_{s \in \mathcal{S}} \mathcal{A}^s(x) = 1 \right\} \tag{7}$$

Similarly, we define **stability** as the intersection of true positives across all instances, reflecting how consistently an attack identifies members despite randomness. This excludes members whose status is inconsistently predicted across runs:

$$\text{Stability}_{\mathcal{S}}(\mathcal{A}) = \left\{ x \in \mathcal{D}_T : \bigcap_{s \in \mathcal{S}} \mathcal{A}^s(x) = 1 \right\} \tag{8}$$

Coverage reflects the extent of potential privacy leakage, while stability captures the consistency of privacy vulnerability under an MIA method. Because the privacy risks they reveal are independent of any specific instance, given their convergence observed in Section 4.3, we compute the Jaccard similarity of coverage and stability to characterize the method-level disparities across different MIAs, i.e., the differences in the subsets of the training data targeted by different MIA methods (regardless of any specific instance).

Illustrative Example: Figure 2 shows the union (coverage) and intersection (stability) of three instances for each attack method. As we can see, all attacks that involve randomness from shadow model training, except the LOSS attack [52], exhibit significant variations in member detection, with their coverage and stability changing considerably from one instance to three instances. Therefore, it is evident that single-instance-based MIA evaluations or assessments may be unreliable.

3.2 Multi-instance Attack Analysis

To analyze the instance-level disparity from the lens of coverage and stability, we introduce a multi-instance analysis framework. As demonstrated in Figure 3, we first prepare n instances of an attack $\mathcal{A}$ using the same auxiliary dataset $\mathcal{D}_A$ but different random seeds. To attack a target model F_T, each MIA instance performs inferences on the target dataset $\mathcal{D}_T$ with access to F_T. The MIA scores obtained are converted to binary membership predictions using Algorithm 1 (**AdjustFPR**) to obtain predictions at a specific FPR level β. It is crucial because we need to maintain a consistent level of FPR to ensure a fair comparison between different MIA instances on their coverage and stability derived from true positive detections. With the predictions of multiple instances of $\mathcal{A}$, we can compute the coverage and stability of $\mathcal{A}$ over n instances.

Algorithm 1 FPR-Based Thresholding (*AdjustFPR*) This algorithm predicts membership by determining a MIA score threshold τ that achieves a specified target FPR level β.

```
Require: Membership ground truth array $gt$, MIA score array $scores$,
    target false positive rate $\beta$
FPRs, TPRs, thres ← roc_curve($gt, scores$)    ▷ function roc_curve
    from scikit-learn[40].
$idx \leftarrow \text{argmin}(|\text{FPRs} - \beta|)$
$\tau \leftarrow \text{thres}[idx]$
$pred \leftarrow \mathbb{1}[scores \geq \tau]$    ▷ membership indicator function.
return $pred$
```

Algorithm 2 Multi-instance Attack Analysis Framework This procedure handles the training, preparation, and execution of attacks, and computes aggregated results to assess stability and coverage across different attack configurations.

```
Require: Attacks $\mathbb{A}$, untrained model $f$, Dataset $\mathcal{D}$, seeds $\mathcal{S}$, stability or
    coverage analysis $analyze$, desired false positive rate $\beta$
 1: $\mathcal{D}_T, \mathcal{D}_A \leftarrow$ partition($\mathcal{D}$)
 2: $\mathcal{D}_{\text{target\_train}}, \mathcal{D}_{\text{target\_test}} \leftarrow$ partition($\mathcal{D}_T$)
 3: $gt = \vec{1}_{\texttt{len}(\mathcal{D}_{\text{target\_train}})} \oplus \vec{0}_{\texttt{len}(\mathcal{D}_{\text{target\_test}})}$    ▷ ground truth array
 4: $F_T$ = train_model($f, \mathcal{D}_{\text{target\_train}}$)
 5: $f_{\text{access}}$ = blackbox_access($f_{\text{target}}$)
 6: for each attack $\mathcal{A} \in \mathbb{A}$ do
 7:     for each seed $s$ in $\mathcal{S}$ do
 8:         set_seeds($s$)
 9:         $\mathcal{A}$.prepare($f_{\text{access}}, \mathcal{D}_A$)    ▷ shadow model training
10:         $scores = \mathcal{A}$.infer($\mathcal{D}_{\text{target}}$)
11:         $preds_s \leftarrow$ AdjustFPR($gt, scores, \beta$)
12:     end for
13:     $analyze(\{preds_s : s \in \mathcal{S}\}, gt)$
14: end for
```

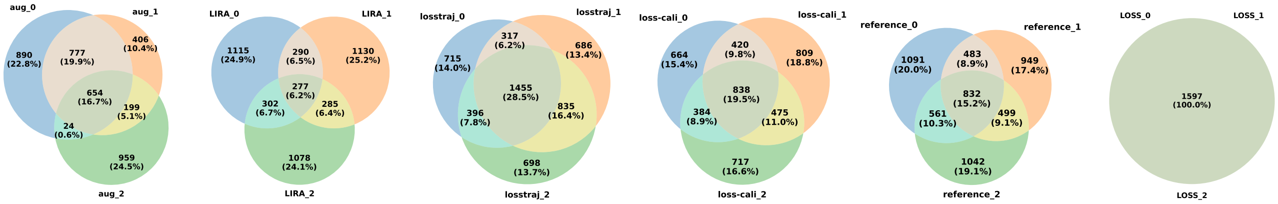
Given multi-instance membership inference predictions, coverage helps capture all possible risks, i.e., members that are vulnerable to any MIA instance at a given FPR level β, while stability focuses on vulnerable members that are consistently detected by an MIA across different random instantiations. Our complete evaluation framework follows Algorithm 2 to compare different attack instances under the same conditions. It splits the dataset $\mathcal{D}$ into two non-overlapping datasets, the auxiliary dataset $\mathcal{D}_A$ and the target dataset $\mathcal{D}_T$. $\mathcal{D}_T$ is further divided into two equal-size, non-overlapping subsets in line 2, one for training the target model (constituting the members) and one for testing (comprising the non-members). This division ensures that both subsets are of equal size ($|\mathcal{D}_{\text{target\_train}}| = |\mathcal{D}_{\text{target\_test}}|$), ensuring a balanced prior of memberships. Lines 6 to 12 apply the pipeline in Figure 3 to each attack method. Line 13 calculates the stability and coverage of each attack over $|S|$ instances.

3.3 MIA Method Level Disparity

As discussed in Section 3.1, we use coverage and stability to evaluate disparities between different MIAs. These measures enable us to analyze how existing attacks differ at the method level in terms of both the extent of vulnerable members they expose (i.e., coverage) and the consistency of vulnerable samples across instances (i.e., stability), despite the presence of instance-level variance.

To assess the method-level disparity, we compute the Jaccard index between the coverage/stability sets of different MIA methods. For each MIA, both coverage and stability are evaluated based on the predictions from the same number of instances at an identical FPR level to ensure fairness, as described in Section 3.2. In addition, we conducted a preliminary empirical analysis of the following two aspects to explore potential factors that contribute to method-level disparities.

- **Stability Difference in Model Output Space**: We define $\mathcal{A}$-*unique samples* as the set of members correctly identified by the stability of MIA $\mathcal{A}$, but not by the stability of any other MIA methods. Formally, the set of $\mathcal{A}$-unique samples, denoted



(a) Augmentation Attack

(b) LIRA

(c) Loss Trajectory Attack

(d) Diff-Calibration Loss Attack

(e) Reference Attack

(f) LOSS Attack

Figure 2: Venn Diagram of three MIA instances at FPR = 0.1 for different attack methods. Each set represents the true positive samples from one instance. The Venn diagram of the Class-NN attack is shown in Figure 1b.

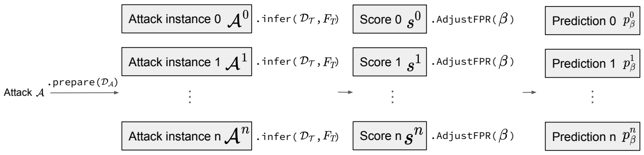

Attack instance 0 $\mathcal{A}^0$
Attack instance 1 $\mathcal{A}^1$
Attack instance n $\mathcal{A}^n$
.infer($\mathcal{D}_T$, F_T)
Score 0 s^0
Score 1 s^1
Score n s^n
.AdjustFPR(β)
Prediction 0 p^0_β
Prediction 1 p^1_β
Prediction n p^n_β
Attack $\mathcal{A}$
.prepare($\mathcal{D}_\mathcal{A}$)

Figure 3: MIA Multi-Instance Analysis Pipeline. The process includes preparing attack instances, inferring membership, and adjusting predictions based on a given FPR.

as $S^{\text{unique}}_{\mathcal{A}}$, is expressed as:

$$S^{\text{unique}}_{\mathcal{A}} = \{x \mid x \in \text{Stability}(\mathcal{A}) \land x \notin \bigcup_{B \neq \mathcal{A}} \text{Stability}(B)\} \tag{9}$$

In a black-box attack setting, logits encapsulate the maximum information returned by a query. Given the model's logits output on those samples uniquely "consistently captured", we look into the difference in their distributions to understand if an MIA may target or is more sensitive to distinct output distributions of members, which may help explain the disparities among MIAs.

- **Attack Signal Difference**: Different MIA methods use different feature extraction method ϕ, resulting in different signals for MIA. To understand the impact of signals while isolating the effect of randomness and MIA methodology difference, we focused on the Class-NN MIA method and $\mathcal{A}$-*covered samples* that are the members identified by MIA $\mathcal{A}$'s coverage,
$$S^{\text{covered}}_{\mathcal{A}} = \{x \mid x \in \text{Coverage}(\mathcal{A})\} \tag{10}$$
Class-NN uses logits as attack signals, so we can easily manipulate the signal by restricting it to only the top-x logits while masking out the rest, referred to as "x-top" Class-NN. This modification allows us to observe how variations in the signals received by the same MIA influence the resulting detected member sets.

4 Evaluation

In this section, we evaluate the disparities between the seven widely used MIAs described in Section 2, using the methodology introduced earlier to assess both the instance-level and the method-level disparities, and investigate their potential causes.

4.1 Experiment Setup

To make sure our empirical analysis is comprehensive, our experiment uses five datasets and four neural network architectures, listed below. A more detailed set-up including the hyperparameter choices of MIAs is provided in Appendix Section A.

Datasets. We use five datasets commonly adopted in MIA research: CIFAR-10, CIFAR-100, CINIC-10, Purchase100, and Texas100. CIFAR-10 and CIFAR-100 consist of 60,000 32x32 color images divided into 10 and 100 classes, respectively. CINIC-10, an extension of CIFAR-10, includes 270,000 images derived from CIFAR-10 and ImageNet. Purchase100 and Texas100 are structured datasets representing consumer purchase behaviors and hospital discharge records, respectively. Detailed dataset descriptions are provided in Appendix Section A.1. Unless otherwise specified, we present experimental results based on CIFAR-10.

Models. We employed ResNet-56 [16], MobileNetV2 [43], VGG-16 [45], and WideResNet-32 [54] as our primary model architectures for image datasets, with ResNet-56 being the main model used for reporting experimental results. All models are optimized with SGD and a cosine learning rate scheduler [31]. We choose MLP for tabular datasets Purchase100 and Texas100. Training and evaluation configurations, including dataset partitions and training epochs, are detailed in Appendix Section A.2.

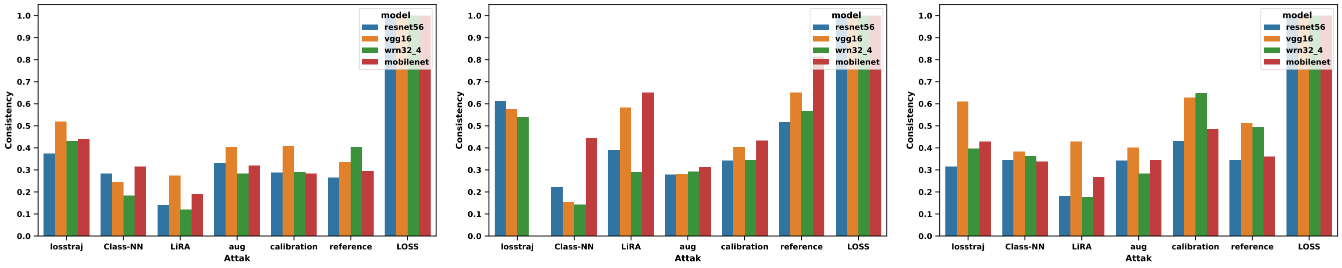
MIA setup. For most MIAs examined in this paper, we adhered to the standard settings used to produce the main results in their respective papers, except LiRA. Our experiment uses LiRA's online version in its paper. A detailed discussion of the setup for these MIAs and LiRA is provided in Appendix A.3, and the consistency result of offline LiRA is discussed in Appendix Section D. For disparity evaluation, we utilize six instances to compute coverage, stability, and consistency scores, as these metrics generally start to converge in most cases at this number of instances, as shown in Section 4.3. Additionally, we focus on presenting results at FPR=0.1, with results for other FPR settings available in the Appendix.

Additionally, we examine the impact of outliers and auxiliary-target dataset distribution gap, with relevant results presented in Appendix Section E and Section C.4.

4.2 MIA Instance Level Disparity

4.2.1 Inherit Low Consistency of MIA. Following the methodology introduced in Section 3.1, we evaluate the consistency scores of different MIAs, each using six instances under the standard setting. Figure 4 shows the consistency scores for each MIA. Except for the LOSS attack, all other attacks exhibit low consistency across datasets and model architectures, with an average consistency score below 0.4, highlighting the inherent instance-level inconsistency.

The instance-level MIA consistency appears to be influenced by the dataset. Attacks on CIFAR-100 demonstrate the highest overall consistency, likely due to increased over-fitting, as indicated

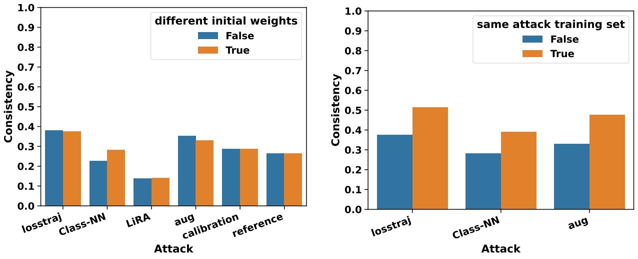


(a) CIFAR-10

(b) CIFAR-100

(c) CINIC-10

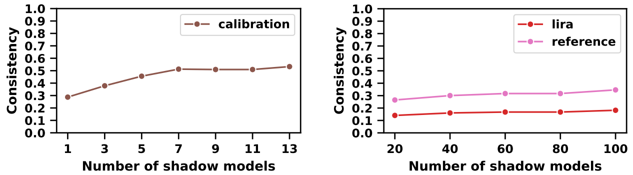
Figure 4: Consistency score shows inherent disparities among pairs of instances of MIAs (except LOSS attack) across datasets and models (ResNet-56, VGG-16, WideResNet-32, MobileNetV2). Consistency evaluated at FPR=0.1.



(a) Shadow Model Training

(b) Attack Model Training

Figure 5: Shadow model training and attack model training both contribute to the disparity. Consistency is measured with six instances at FPR=0.1 on CIFAR-10.



(a) Diff-Calibration

(b) LiRA and Reference Attack

Figure 6: Number of shadow models' relation with Consistency. the model architecture is ResNet-56, and the dataset is CIFAR-10. Consistency is measured with six instances at FPR=0.1.

by the larger generalization gap between the training and testing performance of target models (Appendix Table 2). Greater overfitting results in a larger set of common members that are easier for MIAs to infer, thereby leading to higher consistency. Additionally, certain attacks exhibit higher consistency on specific datasets and model architectures. For example, LiRA and Reference Attack achieve relatively high consistency on CIFAR-100 with VGG-16 and MobileNetV2. In contrast, Class-NN consistently shows low consistency across all datasets. This inconsistency arises from its non-overlapping shadow training sets, which lead to less aligned shadow models.

4.2.2 Disparity Factors. The randomness in MIA construction involves random partitioning of the auxiliary dataset into member and non-member sets, shadow model training, and attack model training. We investigate how these factors contribute to instance-level disparity in MIAs, particularly at low false positive rates (FPR).

Shadow Model Training: For MIAs that rely on shadow models, shadow model training involves data shuffling and partitioning, weight initialization, and other randomness factors that are specific to an MIA, such as model distillation in Loss Trajectory Attack.

To analyze the effects of shadow model training, we compute the consistency score for instances created with different initial weights and compare it to the score for instances created with the same initial weights. When all instances start with the same set of initial weights for shadow models, data shuffling and partitioning become the primary sources of randomness for shadow model training. Figure 5a shows the consistency scores of each MIA under these two conditions. The "False" condition represents the same set of initial model weights. Comparing the two, we observe that the effect of varying initial weights on disparity is minimal (i.e., changes in the consistency score are no more than 0.05), indicating that data random shuffling and partitioning are the primary contributors to the disparity.

Attack Model Training: For attacks that include attack classification models, the training of these models can also contribute to instance-level disparity. To evaluate this effect, we fix the shadow models across all instances, ensuring that their attack models are trained on the same attack training set (corresponding to the "True" case in Figure 5b). Comparing this setup with the normal scenario where shadow models and attack training sets vary, we find that the consistency score increases by 8% to 12%. This indicates that attack model training also contributes to MIA disparity, though its impact is less pronounced than that of shadow model training.

These findings highlight that both shadow model training and attack model training are critical factors driving high instance-level disparity in MIAs.

Number of Shadow Models: Additionally, we examine the impact of the number of shadow models used in an MIA on disparity. The Class-NN Attack is excluded from this analysis, as it trains shadow models on disjoint datasets, meaning that increasing the number of shadow models reduces the training data size and thus the quality of shadow models. Loss Trajectory and Augmentation attacks are also excluded, as their methodologies do not specify how they operate with multiple shadow models.

As shown in Figure 6, increasing the number of shadow models in an MIA slightly increases the consistency at the instance level. For the Difficulty Calibration Loss Attack, the consistency increases as the calibration term (computed from shadow-model losses) becomes a more stable empirical estimate of difficulty. This reduces the variance of the calibration loss and enhances consistency. For LiRA and the Reference Attack, the increase in consistency is relatively smaller. Overall, despite the increase in the number of shadow models, instance-level disparity remains significant across MIAs.

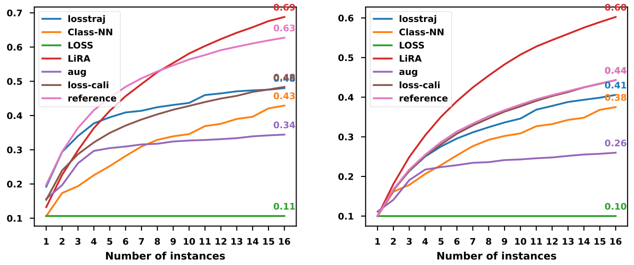
4.3 Coverage and Stability Over Randomness

4.3.1 Coverage Over Randomness. To evaluate the coverage of each attack, we compute the union of positive membership predictions across varying numbers of instances and present the results in Figure 7. Figure 7a shows that as we increase the number of instances, TPR (i.e., coverage) increases accordingly. However, as shown in Figure 7b, the FPR also rises with additional instances, indicating that more nonmembers are incorrectly classified as members. Figure 9a further illustrates the corresponding decrease in precision. Notably, the drop in precision is less pronounced, as the growth in true positives partially offsets the increase in false positives. As the true positive set stabilizes, precision also converges.

We repeat the same experiment by configuring each MIA instance with different FPR thresholds: 0.001, 0.01, and 0.2. The results are provided in Figure 18 in Appendix Section B. We observe that Loss Trajectory, Reference, and Calibrated Loss attacks consistently achieve the highest number of true positive samples when multiple instances are used. At FPR 0.01, the coverage of these attacks with multiple instances captures approximately five times more members compared to a single instance. In contrast, the coverage of the Loss attack remains unchanged across all metrics, as it is not affected by randomness. We also observe convergence at the tail end of each TPR curve, indicating that an MIA instance under a fixed FPR can only identify a subset of members within a bounded group, even under different randomization conditions.

4.3.2 Stability Over Randomness. Following the same setting as the coverage evaluation, we compute the intersection of positive membership predictions to assess stability. The results are presented in Figure 8. The figures show that both TPR and FPR decrease with the number of instances, indicating substantial variance in the detection results between different MIA instances, except for the LOSS attack, which is not affected by randomness.

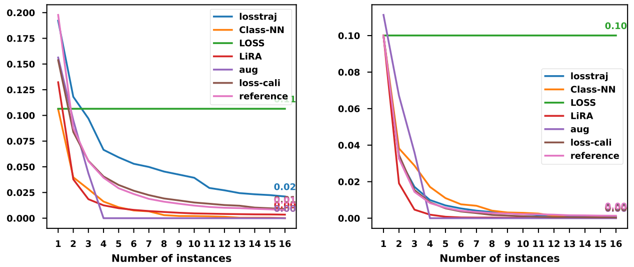
We also conduct the same experiment with MIA instances at different FPR values of 0.001, 0.01, and 0.2. The results are provided in Figure 19 in Appendix Section B. We observed that at low FPRs (e.g., 0.001, 0.01), the stability of most attacks converge to fewer than 10 and 50 true positive members, respectively. This finding highlights that only a small subset of data points is consistently vulnerable to a given MIA method, regardless of randomization effects. As the number of consistently identified members decreases, precision (Figure 9b) increases significantly. Specifically, the Loss Trajectory, Calibrated Loss, and LiRA achieve precision values that exceed 95%. In contrast, the Augmentation attack and Class-NN attack fail to show similar improvements, reflecting their limited capability to consistently predict vulnerable members. Importantly, achieving high precision does not require all 16 instances; most precision gains are realized within the first six instances, while the FPR of stability drops to near zero.



(a) True positive rate

(b) False Positive Rate

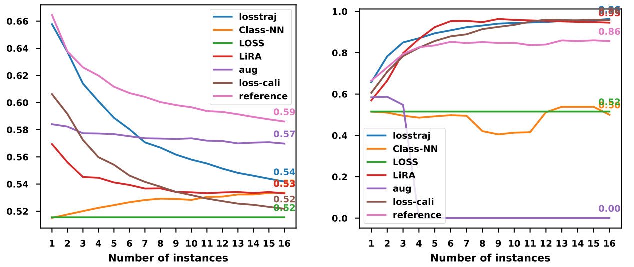
Figure 7: Trends in TPR and FPR for coverage under different numbers of instances with $FPR = 0.1$. For all attacks, each instance is created using the same auxiliary dataset of 30,000 samples, and they predict membership on a disjoint target dataset of 30,000 samples.



(a) True positive rate

(b) False Positive Rate

Figure 8: Trends in TPR and FPR for stability, following the same setup as Figure 7.



(a) Coverage Precision

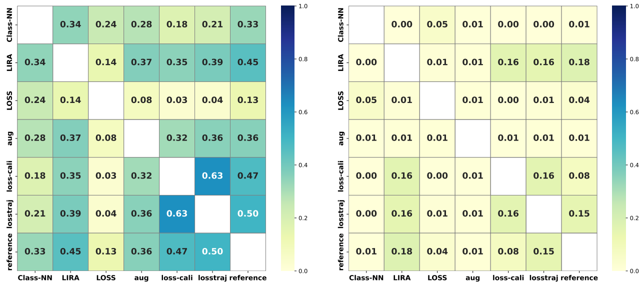
(b) Stability Precision

Figure 9: Precision of coverage and stability corresponding to Figure 7 and 8.

4.4 MIA Method Level Disparity

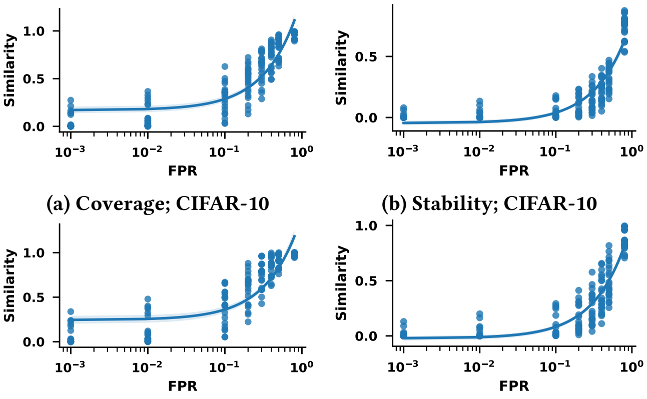
Following the methodology in Section 3.3, we compute the Jaccard similarity of coverage/stability between every pair of MIA methods. Figure 10 presents the results for coverage and stability derived from six instances constructed for each MIA method with FPR=0.1. Both coverage and stability show low similarity between most attack pairs, with Jaccard similarity generally below 0.4 for coverage and 0.1 for stability. The notably lower Jaccard similarity in stability compared to coverage underscores the significant disparities in consistently detected vulnerabilities across different MIAs.

Certain attack pairs exhibit similar trends in both coverage and stability. For example, LiRA and Reference Attack show higher similarity in both measurements, likely due to their shared approach



(a) Coverage (b) Stability

Figure 10: MIA Method Disparity. The values represent the average Jaccard similarity of 4 experiment runs. Attacks' coverage and stability are calculated with six instances at FPR=0.1 on CIFAR-10.



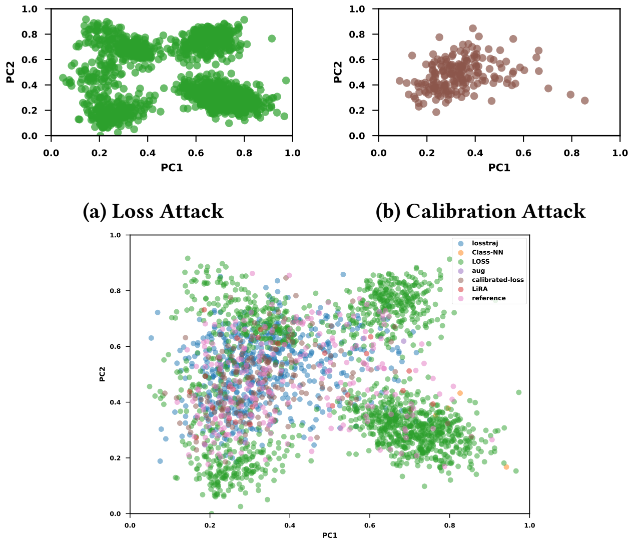
(a) Coverage; CIFAR-10 (b) Stability; CIFAR-10

(c) Coverage; CIFAR-100 (d) Stability; CIFAR-100

Figure 11: Correlation of Disparity and FPR. The line is a linear regression on all Jaccard similarity scores for different FPR values of instances.

to shadow model training. Similarly, Loss Calibration, Loss Trajectory, and Reference Attacks show mutual similarity, likely because they are all based on loss signals. Conversely, some attack pairs show significant disagreement; for example, the Loss Attack consistently demonstrates low Jaccard similarity with all other attacks. We also observe that attacks that perform well at low FPR (e.g., Loss Trajectory, Reference attack, Loss Calibration attack, and LiRA) tend to be more mutually similar compared to attacks designed for average-case performance (e.g., Class-NN, Loss, Augmentation attack). Overall, while certain attack pairs produce membership predictions with moderately higher similarity than others, the Jaccard similarity remains low across the board when predictions are made at FPR=0.1.

We also evaluate the pairwise similarity between different MIA methods in terms of coverage and stability under varying instance-level FPR values (from 0.001 to 0.2). The results are presented in Appendix Section B. Across different FPR settings, the overall similarity trend remains consistent, and we observe a positive correlation between FPR and similarity. In Figure 11, each point represents the Jaccard Similarity value between a pair of attacks at a given FPR level. As shown, both coverage and stability exhibit low similarity at low FPRs. As the FPR increases, the similarity also increases, with coverage similarity values approaching 1. The relatively low similarity at low FPR suggests that different attacks may have distinct insights into predicting members, especially at low FPRs. The member predictions in which these attacks are most confident tend to be nearly disjoint across methods, implying that traditional metrics—particularly those evaluated at low FPR—do not fully capture the diverse behaviors and strengths of different attacks.



(a) Loss Attack (b) Calibration Attack

(c) All Attacks

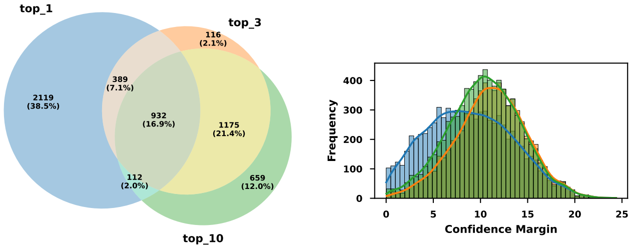
Figure 12: PCA of $\mathcal{A}$-unique Members. We conduct PCA on all the target model F_T's logits of all samples in target dataset $\mathcal{D}_T$, and plot $\mathcal{A}$-unique members (defined in Section 3.3) for each attack. It is obtained with six instances at FPR@0.1. Figure 12a and Figure 12b are picked to demonstrate the distribution difference between two MIAs. The explained variance ratio is 48.56%, and 15.95% for the PC1 (first component) and PC2 (second component), respectively.

4.5 Disparity Empirical Analysis

In this section, we conduct a preliminary empirical analysis to analyze the potential causes of disparities between MIAs with the methodology presented in Section 3.3. This analysis offers insights into the underlying factors that lead different MIA methods to implicitly target different subsets of training data.

4.5.1 Output Distribution of $\mathcal{A}$-Unique Samples. To understand the difference in data points that are vulnerable to different attacks, we analyze the output space of the target model F_T. We apply Principal Component Analysis (PCA) to extract feature scores from the logits predicted by F_T for each member. Focusing on $\mathcal{A}$-unique members (as defined in Section 3.3) allows us to understand the disparities among MIAs through their uniquely identified members.

Figure 12 shows that the $\mathcal{A}$*-unique* samples identified by different attacks can exhibit different distributions in the model output space. This suggests that each MIA may implicitly favor certain groups or distributions of members, although PCA may only be able to partially capture the underlying characteristics. In particular, as shown in Figures 12a and 12b, the samples uniquely identified by



(a) Top-x Class-NN Attack's Coverage

(b) Distribution Shift of Confidence Margin

Figure 13: Top-x Class-NN Attack-Covered Samples with Different Signals. The coverage is calculated using 6 instances at FPR = 0.1.

the Loss attack and those identified by the Augmentation attack form visibly different clusters.

4.5.2 Attack Signals of $\mathcal{A}$-Covered Samples. In addition to difference in the distribution of detected samples in the model output space, disparities among MIAs also stems from their distinct methodologies and signals they exploit. Quantifying these methodological differences is challenging, as each attack employs different processes that are difficult to formalize within a unified framework. Therefore, we focus on examining how the signals obtained from the target and shadow models contribute to variation in member detection. Following the methodology in Section 3.3, we use "Top-x" Class-NN attack with varying signals.

Figure 13a shows a Venn diagram illustrating the coverage of three variations: Top-1, Top-3, and Top-10 Class-NN attacks. We observe a greater overlap between Top-3 and Top-10 Class-NN attacks compared to their overlap with Top-1. This is because Top-3 and Top-10 attacks leverage more similar signals, as logits beyond the top 3 positions are typically close to zero and contribute little additional information.

To better understand the distribution difference of detected members under different top-x signals, we measure the **Confidence Margin** of the target model's predictions for each member. The confidence margin is defined as the difference between the highest confidence score (representing the most likely class) and the second highest confidence score (representing the next most likely class) in the output of the target model. It represents how much more confident the model is in its top prediction relative to the closest alternative. Figure 13b presents the kernel density estimate (KDE) of confidence margin values for correctly predicted members, showing that less similar signals lead to greater disparity in the distribution of detected members. Specifically, the confidence margin distributions for Top-3 and Top-10 signals are similar, while the distribution for Top-1 is more left-skewed and exhibits higher variance. This difference arises because the Class-NN attack with access to Top-3 or Top-10 logits can leverage additional information beyond the single highest logit used by Top-1. As a result, Top-3 and Top-10 attacks tend to identify more common members with larger confidence margins. These observations on confidence margin distributions align with the overlap patterns observed in the Venn diagram in Figure 13a.

The impact of attack signals extends beyond the Top-x Class-NN model. Existing MIAs rely on various signals including loss, confidence score vector, and loss trajectories. While the loss, computed from the confidence score vector, provides a scalar summary, it may omit finer details present in the vector itself which can be beneficial for membership inference. The loss trajectory offers insights into the training-time patterns that may reveal more nuanced information. These differences suggest that each type of signal carries unique information that may contribute to disparities in member detection across different MIAs.

4.6 Practical Implications of MIA Disparities

Given the significance of instance-level and method-level disparities in MIAs, caution must be exercised when using them for privacy assessment and performance evaluation. Evaluating each MIA separately or relying on a single instance may fail to capture the full spectrum of privacy leakage risks, leading to incomplete assessments. Below, we examine several privacy tasks where MIAs are commonly used and discuss the implications of MIA disparities.

Privacy Auditing and Risk Assessment: Several open-source toolkits have been developed for privacy assessment (e.g., [22, 27]), which often incorporate multiple MIA methods to assess privacy leakage in trained models. However, these tools typically run one instance per method and report risk based on population-level metrics derived from single-instance results. This overlooks both instance-level and method-level disparities, which may result in undetected vulnerable member samples and lead to overly optimistic privacy estimates.

Machine Unlearning: In many unlearning works (e.g., [7, 24]), MIAs are used to evaluate the effectiveness of unlearning by checking how many samples from a forgetting set (a subset of the training set) remain identifiable after unlearning. A common practice is to instantiate an MIA with an auxiliary dataset and report results from that single instance. However, due to instance-level variability, a sample undetected by the reported instance may still be identified by others, potentially leading to overestimation of unlearning effectiveness. Similarly, relying on a single MIA method can overlook member samples that would be detected by other methods due to method-level disparities.

Privacy Defense Evaluation: Many defense mechanisms (e.g., [20, 26]) are evaluated against a single MIA instance. The reported metric reflects exposure risk only for the members detected by that specific instance. However, due to randomness in shadow model training and method-specific biases, it is unclear whether a defense appears stronger simply because it performs better on the particular subset exposed by that instance. It remains uncertain whether the defense would perform similarly on samples exposed by other instances or MIA methods.

Thus, explicitly addressing MIA disparities is essential for fair and reliable evaluation of privacy risks and defenses.

5 MIA Ensemble

In this section, we propose an ensemble framework that employs various strategies to account for MIA disparities. This framework not only enables the construction of more powerful attacks but also provides an evaluation protocol for more comprehensive and reliable privacy assessments.

5.1 Ensemble Strategies

5.1.1 Attack Stability Ensemble. Section 4.3 shows that MIA stability converges as more instances are aggregated, and this aggregation reduces false positives, resulting in lower FPR and improved precision. These findings suggest that stability captures members consistently identified across instances regardless of randomness, and can be leveraged within each attack to achieve higher precision. Furthermore, Section 4.4 shows that the stability of different attacks has very little overlap, particularly at low FPR (Figure 11). Accordingly, we propose a 2-step ensemble approach:

1) **Multi-instance Stability**: This step uses multiple instances of an attack $\mathcal{A}$ and uses the "logical and" (i.e., conjunction) to determine membership to improve precision. That is, a sample x is regarded as a member only if all of $\mathcal{A}$'s instances determine x is a member.
2) **Multi-attack Union**: This step capitalizes on the high precision achieved by multi-instance and the complementary nature of different attacks, which tend to identify distinct sets of members. By taking the "logical or" (i.e., disjunction) of prediction from multiple attacks from the multi-instance step, the ensemble can detect members across all MIAs' stable predictions while maintaining high precision.

Formally, let $p_i^{\mathcal{A}}(x)$ represent the membership prediction of i-th instance of attack $\mathcal{A}$ on sample x. Note that $p_i^{\mathcal{A}}(x)$ is a prediction that's thresholded to either 1 or 0. The multi-instance prediction $P_n^{\mathcal{A}}$ is the conjunction of $\{p_1^{\mathcal{A}}, \ldots, p_n^{\mathcal{A}}\}$ for attack $\mathcal{A}$. The multi-attack prediction $P_n^{\{\mathcal{A}_1,\ldots,\mathcal{A}_m\}}$ is the disjunction of multi-instance predictions $\{P_n^{\mathcal{A}_1}, \ldots, P_n^{\mathcal{A}_m}\}$.

$$P_n^{\mathcal{A}_j}(x) = \bigwedge_{i=1}^{n} p_i^{\mathcal{A}}(x) \tag{11}$$

$$P_n^{\{\mathcal{A}_1,\ldots,\mathcal{A}_m\}}(x) = \bigvee_{j=1}^{m} P_n^{\mathcal{A}_j}(x) \tag{12}$$

5.1.2 Attack Coverage Ensemble. The previous attack ensemble strategy applies multi-instance intersection to improve reliability and precision, however, at the cost of coverage. In contrast, the attack coverage strategy here applies the multi-instance union to improve the coverage, followed by the same multi-attack union step. Similarly, we can describe this approach as follows:

1) Multi-instance Coverage:
$$P_n^{\mathcal{A}_j}(x) = \bigvee_{i=1}^{n} p_i^{\mathcal{A}}(x) \tag{13}$$

2) Multi-attack Union:
$$P_n^{\{\mathcal{A}_1,\ldots,\mathcal{A}_m\}}(x) = \bigvee_{j=1}^{m} P_n^{\mathcal{A}_j}(x) \tag{14}$$

5.1.3 Attack Majority Ensemble. While coverage and stability represent two extremes—capturing all potential risks and the most consistently vulnerable samples, respectively—the majority-voting ensemble offers a balanced alternative. This strategy captures samples that are identified as members by the majority of the running instances of a given MIA method. Formally, we have:

1) Multi-instance Majority Voting:
$$P_n^{\mathcal{A}_j}(x) = \left(\sum_{i=1}^{n} p_i^{\mathcal{A}}(x)\right) > \frac{n}{2} \tag{15}$$

2) Multi-attack Union:
$$P_n^{\{\mathcal{A}_1,\ldots,\mathcal{A}_m\}} = \bigvee_{j=1}^{m} P_n^{\mathcal{A}_j} \tag{16}$$

5.2 Evaluation

For the ensemble, we consider four attacks: Difficulty Calibration Loss Attack, Reference Attack, LiRA, and Loss Trajectory Attack, because out of our seven implemented attacks, only these four improve the precision with stability over multiple instances at a low FPR, as shown in Figure 9b. As in our previous setup, we utilize six instances of each MIA for the ensemble. Our proposed ensemble operates on membership predictions rather than membership scores. Therefore, to measure its performance in the TPR-FPR plane, we vary the FPR of base instances with 100 different FPR values, ranging from 10^{-6} to 1, evenly spaced on a logarithmic scale. Under each instance FPR, we compute the predictions by ensemble and derive the corresponding TPR and FPR values for the ensemble.

In Figure 15, we observe that the TPR of all three ensembles consistently outperforms single-instance and *multiple-instance* methods in the TPR-FPR plane. Here, the *multi-instance* method refers to the ensemble approach without the multi-attack union step, i.e., only using (11), (13), or (15). Interestingly, we find that the multi-instance method alone often outperforms its single-instance counterpart, particularly when using stability or majority-voting strategies. This further demonstrates that relying on a single MIA instance for evaluation underestimates the true privacy risks, as, in real-world scenarios, multiple MIA instances could be generated by the same or different attackers, and inherent instance-level disparities in membership inference persist. Additionally, we evaluate all possible combinations of the four attacks and compare their ROC curves. As shown in Appendix Figure 24, the full ensembles leveraging all four attacks consistently achieve higher TPR across all FPR values compared to ensembles using fewer or different combinations of attacks.

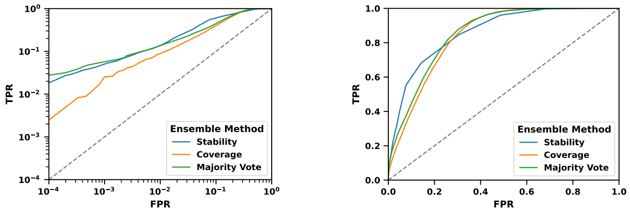
Table 1 further compares our full ensembles and multi-instance-only ensembles against each single MIA instance in terms of AUC, accuracy, and TPR at 0.1% FPR (see Appendix Table 5 for results on Texas100 and Purchase100). We choose FPR = 0.1% to showcase the ensemble's capabilities under low FPR conditions, aligning with evaluation metrics used in recent works [3, 30]. The results are based on the ResNet-56 architecture, and comparisons for other model architectures can be found in Appendix Table 7. Across all settings, the final three rows in Table 1 show that all three full ensemble strategies consistently outperform individual instances under three traditional performance metrics. Compared to single-instance attacks, the multi-instance-only ensemble (denoted as 'Multi-inst' in the table) shows improved performance under both stability and majority-voting strategies. However, it underperforms under the coverage-based ensemble, where only the multi-instance

		CIFAR-10			CIFAR-100			CINIC-10		
Ens. Lvl	**Attack**	**AUC**	**ACC**	**TPR**	**AUC**	**ACC**	**TPR**	**AUC**	**ACC**	**TPR**
Single-inst.	losstraj	0.635	0.588	0.002	0.852	0.771	0.042	0.673	0.619	0.008
	reference	0.608	0.596	0.010	0.813	0.774	0.034	0.615	0.605	0.005
	lira	0.585	0.570	0.005	0.802	0.729	0.034	0.601	0.578	0.003
	calibration	0.603	0.572	0.005	0.721	0.676	0.008	0.623	0.593	0.002
Multi-inst. Coverage	losstraj	0.595	0.557	0.010	0.788	0.695	0.032	0.643	0.592	0.006
	reference	0.638	0.600	0.012	0.863	0.783	0.045	0.652	0.619	0.005
	lira	0.570	0.561	0.005	0.801	0.729	0.036	0.590	0.576	0.003
	calibration	0.551	0.532	0.007	0.652	0.607	0.010	0.617	0.589	0.003
Multi-inst. Stability	losstraj	0.659	0.610	0.022	0.898	0.817	0.108	0.685	0.630	0.013
	reference	0.582	0.582	0.008	0.765	0.772	0.054	0.585	0.582	0.005
	lira	0.603	0.588	0.011	0.839	0.764	0.087	0.623	0.593	0.009
	calibration	0.650	0.620	0.013	0.855	0.763	0.031	0.643	0.616	0.003
Multi-inst. Maj-vote	losstraj	0.668	0.602	0.018	0.877	0.786	0.061	0.712	0.647	0.008
	reference	0.636	0.615	0.011	0.860	0.806	0.070	0.637	0.626	0.006
	lira	0.596	0.579	0.012	0.840	0.766	0.068	0.621	0.595	0.008
	calibration	0.629	0.586	0.013	0.741	0.672	0.018	0.634	0.603	0.003
Multi-attack	Coverage	0.841	0.781	0.025	0.913	0.835	0.077	**0.898**	**0.834**	**0.014**
	Stability	**0.863**	0.770	0.050	**0.978**	**0.935**	**0.280**	0.807	0.740	0.009
	Maj-vote	0.858	**0.789**	**0.056**	0.961	0.897	0.208	0.865	0.800	**0.014**

Table 1: Performance of ensembles with four attacks vs. single instance attacks. TPR is measured at 0.1% FPR.

Reference attack shows a slight performance improvement. By comparing the full ensembles with the multi-instance-only ensembles, we observe that the benefit gained from multi-attack union often exceeds that achieved through multi-instance aggregation alone.

While all three full ensembles achieve improved performance, each exhibits unique strengths across different FPR ranges. Figure 16 shows their ROC curves side by side. From linear-scale ROC in Figure 16b, we can see that the Stability Ensemble outperforms the Coverage Ensemble in the lower FPR region (FPR < 0.3), while the Coverage Ensemble achieves a higher TPR in the higher FPR region (FPR > 0.3). On log-scale ROC, we can see that the Majority Voting performs comparably to the Stability Ensemble at low FPR (also demonstrated in Table 1), and also exceeds the Stability’s performance in the high FPR region. These trends align with the design of each ensemble method. Coverage tends to cover more potential risks at the cost of increased FPR, stability focuses on consistently identifying vulnerabilities with high precision, while Majority Voting balances their strengths.



(a) Log Scale

(b) Linear Scale

Figure 16: ROC Curves of Different Ensemble Strategies.

5.3 Ensemble in Practice

5.3.1 Optimization Strategies for Ensemble. The ensemble framework leverages both multi-instance and multi-attack approaches, achieving comprehensive coverage of privacy risks at the expense of increased computational cost. Below, we discuss practical strategies to mitigate this computational overhead.

Low-Cost Attack as an Add-on. Among the four attacks we examined, the Difficulty Calibration Loss Attack requires much less time to prepare than the others, requiring only a single shadow model. This makes it an ideal add-on attack.

Attacks Sharing the Same Process. Many membership inference attacks share similar, if not identical, preparation processes. For example, LIRA and the Reference Attack both rely on the same shadow model training process (as detailed in Appendix Section A.3). In our experiments, LIRA and the Reference Attack utilized the same 20 shadow models, making their ensemble nearly as cost-effective as preparing just one of them. This ensemble identified approximately twice as many members as either individual attack. Similarly, the Difficulty Calibration Loss Attack can serve as a “free” add-on if another attack already involves training a shadow model, as it only requires one shadow model to calibrate the MIA score [50].

5.3.2 Cost Analysis. We measure the computation cost of ensembles in GPU hours for each MIA instance, considering different numbers of instances per ensemble. When both LiRA and the Reference Attack are included in an ensemble combination, we apply the above optimization strategy to combine and deduct their shadow model training time. The Majority Voting Ensemble is evaluated with odd numbers of instances to avoid ties in voting.

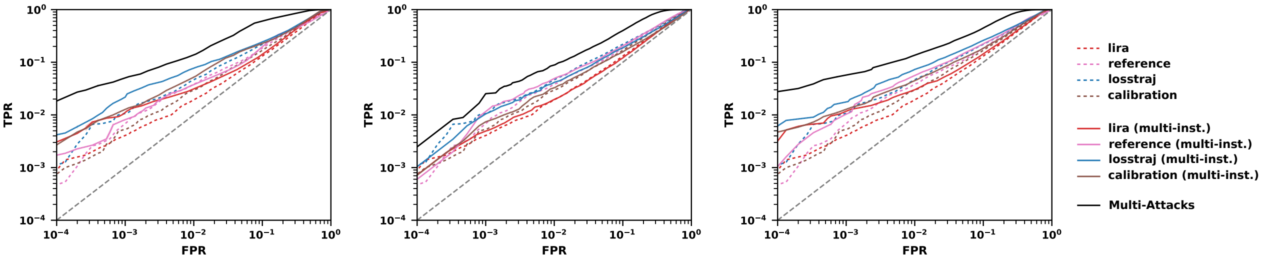
Figure 17 presents cost (in GPU time) v.s. performance (in TPR @0.1%FPR) given different numbers of instances and different combinations of attacks. A more detailed description and study of the cost is provided in the Appendix Section C.1. Overall, we observe a positive correlation between computation cost and performance. Notably, ensembles involving all four attacks consistently achieve the best performance, underscoring the importance of combining multiple attack methods in an ensemble. From additional experiments across different datasets, we conclude that this trend holds true for Stability and Majority Voting Ensembles but does not always apply to Coverage Ensembles.

Additionally, when comparing configurations with similar performance, we observe that cost-effective options often exist, achieving target TPRs with minimal GPU time (indicated by the leftmost points on a given TPR line). For example, in Figure 17a, with target TPR=0.05, an ensemble of four attacks with three instances achieves the same performance as using six instances, effectively reducing the training time by half, from around 3500 mins to 1700 mins. This significant reduction in computation cost demonstrates that a careful selection of attack combinations and instance counts can achieve similar levels of effectiveness without incurring unnecessary overhead. Practitioners may find their desired ensemble configuration to achieve robust privacy evaluations given a resource budget. We leave efficiently identifying optimal configurations for future work.

6 Discussion

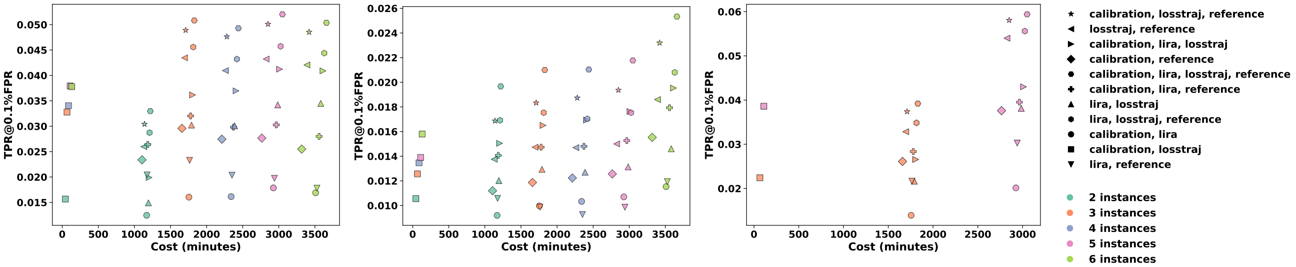
The instance-level and method-level disparities among MIAs, along with the performance gains achieved through ensemble strategies, highlight the practical relevance of MIA disparities and the risk of underestimating privacy vulnerabilities in the tasks discussed in Section 4.6. In this section, we discuss actionable directions for addressing these issues in future MIA research.

MIA performance evaluation and development. We advocate for incorporating disparity analysis into the development and evaluation of MIAs, using our proposed coverage and stability measures to examine and quantify how an MIA differs from others in member



(a) Stability Ensemble (b) Coverage Ensemble (c) Majority Voting Ensemble

Figure 15: ROC Curve for Ensemble. Dashed lines show single-instance ROC, solid lines show multi-instance ROC and the black line represents the complete four-attack ensemble.



(a) Stability Ensemble (b) Coverage Ensemble (c) Majority Voting Ensemble

Figure 17: Performance vs. Cost Analysis for CIFAR-10 using different ensembles.

detection. These measures offer a complementary perspective to traditional population-level metrics such as AUC and TPR@Low FPR by providing additional insight into the extent of privacy risks an attack can expose (via coverage) and the consistency with which it reveals those risks across different runs (via stability).

An MIA that achieves similar or even lower population-level metrics, such as AUC, may still hold significant value if it detects a substantially different subset of members—indicating high disparity—which can be revealed through our disparity analysis based on coverage and stability. This diversity enhances our understanding of privacy vulnerabilities by uncovering risks that other attacks may miss. Our ensemble results further support this insight, demonstrating that combining multiple attacks—including those traditionally considered "weaker"—often leads to improved overall performance. Therefore, MIAs with high disparity contribute to a more complete and robust assessment of privacy risks, especially when leveraged through our proposed ensemble strategies.

Complete and reliable privacy evaluation. As discussed in Section 4.6, the common practice of single-instance-based evaluation is insufficient and may lead to unreliable conclusions. Our ensemble framework addresses this by capturing the full spectrum of privacy risks posed by different MIA methods through multi-instance and multi-attack ensembles based on coverage, stability, and majority voting. Incorporating these ensemble strategies into privacy evaluation ensures a more accurate and robust assessment of privacy defenses, unlearning mechanisms, and overall privacy risk. Given this, it may also be necessary to revisit prior evaluations of privacy defenses and unlearning techniques that relied solely on a single random MIA instance.

Additional Caveats in Using MIA for Privacy Evaluation. Recently, there has been significant interest in MIAs against large language models (LLMs). However, several works [9, 34, 56] have raised concerns about the construction of evaluation datasets, particularly regarding the distribution shift between member and non-member samples. In many LLM MIA evaluations, non-member data were collected from web content published after the model's training cutoff date, such that member samples originate from the training data distribution, while non-member samples come from a different and later distribution (e.g., different time periods). Such distribution shifts can artificially inflate MIA performance, as attacks may exploit these distribution differences rather than truly detecting membership status. Consequently, evaluations based on such datasets may overstate privacy risks.

For privacy defense evaluation, Aerni et al. [1] argues that prior evaluations using MIAs, which report attack performance averaged across all training samples, can be misleading because they may fail to reflect a defense's effectiveness against the most vulnerable examples. In addition, some evaluations have relied on relatively weak, non-adaptive attacks, potentially overstating the robustness of the proposed defenses.

Our study is orthogonal to these concerns by addressing a different overlooked issue: the instance-level and method-level disparities among MIAs, which are often neglected in current evaluation practices. Addressing these disparities requires a more holistic evaluation protocol, such as our proposed ensemble framework, to enable more complete and reliable privacy assessments.

7 Conclusion

In this paper, we have provided critical insights into the disparities of Membership Inference Attacks (MIAs). Our findings challenge conventional evaluation methods of MIAs and highlight the impact of randomness and disparity in these MIAs. Additionally, our proposed ensemble framework not only enables the construction of more powerful attacks but also offers a more comprehensive evaluation methodology. Moving forward, our goal is to develop more sophisticated strategies for ensemble, further improving the efficiency and effectiveness of MIA.

References

[1] Michael Aerni, Jie Zhang, and Florian Tramèr. 2024. Evaluations of Machine Learning Privacy Defenses are Misleading. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security* (Salt Lake City, UT, USA) *(CCS '24)*. Association for Computing Machinery, New York, NY, USA, 1271–1284. https://doi.org/10.1145/3658644.3690194

[2] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine Unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*. 141–159. https://doi.org/10.1109/SP40001.2021.00019

[3] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. 2022. Membership Inference Attacks From First Principles. In *2022 IEEE Symposium on Security and Privacy (SP)*. 1897–1914. https://doi.org/10.1109/SP46214.2022.9833649

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

[5] Dingfan Chen, Ning Yu, and Mario Fritz. 2022. RelaxLoss: Defending Membership Inference Attacks without Losing Utility. arXiv:2207.05801 [cs.LG] https://arxiv.org/abs/2207.05801

[6] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. 2020. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*. 343–362.

[7] Dasol Choi and Dongbin Na. 2023. Towards Machine Unlearning Benchmarks: Forgetting the Personal Identities in Facial Recognition Systems. arXiv:2311.02240 [cs.CV] https://arxiv.org/abs/2311.02240

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

[9] Debeshee Das, Jie Zhang, and Florian Tramèr. 2025. Blind Baselines Beat Membership Inference Attacks for Foundation Models. arXiv:2406.16201 [cs.CR] https://arxiv.org/abs/2406.16201

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

[11] Christophe Dupuy, Radhika Arava, Rahul Gupta, and Anna Rumshisky. 2022. An efficient dp-sgd mechanism for large scale nlu models. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4118–4122.

[12] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. 2019. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757* (2019).

[13] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. 2024. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 12043–12051.

[14] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (Denver, Colorado, USA) *(CCS '15)*. Association for Computing Machinery, New York, NY, USA, 1322–1333. https://doi.org/10.1145/2810103.2813677

[15] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. 2018. 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* (Toronto, Canada) *(CCS '18)*. Association for Computing Machinery, New York, NY, USA, 619–633. https://doi.org/10.1145/3243734.3243834

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs.CV]

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531

[18] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. 2022. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)* 54, 11s (2022), 1–37.

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

[20] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples. arXiv:1909.10594 [cs.CR]

[21] Mishaal Kazmi, Hadrien Lautraite, Alireza Akbari, Mauricio Soroco, Qiaoyue Tang, Tao Wang, Sébastien Gambs, and Mathias Lécuyer. 2024. PANORAMIA: Privacy Auditing of Machine Learning Models without Retraining. *arXiv preprint arXiv:2402.09477* (2024).

[22] Sasi Kumar and Reza Shokri. 2020. ML Privacy Meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning. In *Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs)*.

[23] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. 2023. Towards Unbounded Machine Unlearning. arXiv:2302.09880 [cs.LG] https://arxiv.org/abs/2302.09880

[24] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. 2023. Towards Unbounded Machine Unlearning. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 1957–1987. https://proceedings.neurips.cc/paper_files/paper/2023/file/062d711fb777322e2152435459e6e9d9-Paper-Conference.pdf

[25] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. 2021. Membership Inference Attacks and Defenses in Classification Models. In *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy (CODASPY '21)*. ACM. https://doi.org/10.1145/3422337.3447836

[26] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. 2024. MIST: defending against membership inference attacks through membership-invariant subspace training. In *Proceedings of the 33rd USENIX Conference on Security Symposium* (Philadelphia, PA, USA) *(SEC '24)*. USENIX Association, USA, Article 134, 18 pages.

[27] Qinbin Li, Junyuan Hong, Chulin Xie, Jeffrey Tan, Rachel Xin, Junyi Hou, Xavier Yin, Zhun Wang, Dan Hendrycks, Zhangyang Wang, Bo Li, Bingsheng He, and Dawn Song. 2024. LLM-PBE: Assessing Data Privacy in Large Language Models. *Proc. VLDB Endow.* 17, 11 (July 2024), 3201–3214. https://doi.org/10.14778/3681954.3681994

[28] Gaoyang Liu, Chen Wang, Kai Peng, Haojun Huang, Yutong Li, and Wenqing Cheng. 2019. SocInf: Membership inference attacks on social media health data with machine learning. *IEEE Transactions on Computational Social Systems* 6, 5 (2019), 907–921.

[29] Han Liu, Yuhao Wu, Zhiyuan Yu, and Ning Zhang. 2024. Please tell me more: Privacy impact of explainability through the lens of membership inference attack. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 120–120.

[30] Yiyong Liu, Zhengyu Zhao, Michael Backes, and Yang Zhang. 2022. Membership Inference Attacks by Exploiting Loss Trajectory. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (Los Angeles, CA, USA) *(CCS '22)*. Association for Computing Machinery, New York, NY, USA, 2085–2098. https://doi.org/10.1145/3548606.3560684

[31] Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv:1608.03983 [cs.LG]

[32] Tomoya Matsumoto, Takayuki Miura, and Naoto Yanai. 2023. Membership inference attacks against diffusion models. In *2023 IEEE Security and Privacy Workshops (SPW)*. IEEE, 77–83.

[33] Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison. *arXiv preprint arXiv:2305.18462* (2023).

[34] Matthieu Meeus, Igor Shilov, Shubham Jain, Manuel Faysse, Marek Rei, and Yves-Alexandre de Montjoye. 2025. SoK: Membership Inference Attacks on LLMs are Rushing Nowhere (and How to Fix It). arXiv:2406.17975 [cs.CL] https://arxiv.org/abs/2406.17975

[35] Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. 2022. Quantifying privacy risks of masked language models using membership inference attacks. *arXiv preprint arXiv:2203.03929* (2022).

[36] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. 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)*. IEEE, 739–753.

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

[38] Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. 2022. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299* (2022).

[39] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P Wellman. 2018. Sok: Security and privacy in machine learning. In *2018 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 399–414.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[41] Shahbaz Rezaei, Zubair Shafiq, and Xin Liu. 2023. Accuracy-privacy trade-off in deep ensemble: A membership inference perspective. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 364–381.
[42] Maria Rigaki and Sebastian Garcia. 2023. A survey of privacy attacks in machine learning. *Comput. Surveys* 56, 4 (2023), 1–34.
[43] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2019. MobileNetV2: Inverted Residuals and Linear Bottlenecks. arXiv:1801.04381 [cs.CV]
[44] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.
[45] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.
[46] Liwei Song and Prateek Mittal. 2021. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2615–2632.
[47] Nexhi Sula, Abhinav Kumar, Jie Hou, Han Wang, and Reza Tourani. 2024. Silver Linings in the Shadows: Harnessing Membership Inference for Machine Unlearning. arXiv:2407.00866 [cs.LG] https://arxiv.org/abs/2407.00866
[48] Florian Tramer, Andreas Terzis, Thomas Steinke, Shuang Song, Matthew Jagielski, and Nicholas Carlini. 2022. Debugging differential privacy: A case study for privacy auditing. *arXiv preprint arXiv:2202.12219* (2022).
[49] Taiyu Wang, Qinglin Yang, Kaiming Zhu, Junbo Wang, Chunhua Su, and Kento Sato. 2023. Lds-fl: Loss differential strategy based federated learning for privacy preserving. *IEEE Transactions on Information Forensics and Security* (2023).
[50] Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. 2022. On the Importance of Difficulty Calibration in Membership Inference Attacks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=3eIrli0TwQ
[51] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. 2022. Enhanced Membership Inference Attacks against Machine Learning Models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (Los Angeles, CA, USA) *(CCS '22)*. Association for Computing Machinery, New York, NY, USA, 3093–3106. https://doi.org/10.1145/3548606.3560675
[52] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. 268–282. https://doi.org/10.1109/CSF.2018.00027
[53] Lei Yu, Meng Han, Yiming Li, Changting Lin, Yao Zhang, Mingyang Zhang, Yan Liu, Haiqin Weng, Yuseok Jeon, Ka-Ho Chow, and Stacy Patterson. 2024. A Survey of Privacy Threats and Defense in Vertical Federated Learning: From Model Life Cycle Perspective. *arXiv preprint arXiv: 2402.03688* (2024).
[54] Sergey Zagoruyko and Nikos Komodakis. 2017. Wide Residual Networks. arXiv:1605.07146 [cs.CV]
[55] Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. 2024. Low-Cost High-Power Membership Inference Attacks. arXiv:2312.03262 [stat.ML] https://arxiv.org/abs/2312.03262
[56] Jie Zhang, Debeshee Das, Gautam Kamath, and Florian Tramèr. 2025. Membership Inference Attacks Cannot Prove that a Model Was Trained On Your Data. arXiv:2409.19798 [cs.LG] https://arxiv.org/abs/2409.19798
[57] Minxing Zhang, Zhaochun Ren, Zihan Wang, Pengjie Ren, Zhunmin Chen, Pengfei Hu, and Yang Zhang. 2021. Membership inference attacks against recommender systems. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 864–879.

A Experiment Setup Details

A.1 Datasets

We use five datasets in our experiments: CIFAR-10, CIFAR-100, CINIC-10, Purchase100, and Texas100.

- **CIFAR-10**: Consists of 60,000 32x32 color images in 10 classes, with 50,000 training and 10,000 testing samples.
- **CIFAR-100**: Similar to CIFAR-10, but contains 100 classes with 600 images per class.
- **CINIC-10**: Extends CIFAR-10 to include 270,000 images from both CIFAR-10 and ImageNet. For experiments, we use a balanced subset of 30,000 CIFAR-10 images and 30,000 ImageNet images. Throughout the paper, we shuffle these two subsets.
- **Purchase100**: A structured dataset derived from Kaggle's "Acquire Valued Shoppers" challenge, representing 197,324 shopping records. We use a subset of 60,000 samples, each with 600 binary features.
- **Texas100**: Contains hospital discharge records from the Texas Department of State Health Services. We use 60,000 samples from the dataset, predicting the 100 most frequent procedures.

A.2 Models

We utilize ResNet-56 [16], MobileNetV2 [43], VGG-16 [45], and WideResNet-32 [54] as model architectures. ResNet-56 is the primary architecture for reporting results due to its small generalization gap, making it a harder case for MIAs [5, 25]. For the tabular datasets Purchase100 and Texas100, we use a 4-layer MLP with layer units=[512, 256, 128, 64]. The target models' performances on these datasets are described in 2.

Optimization and Training. All models are trained using SGD with a momentum of 0.9 and an initial learning rate of 0.1, with a cosine learning rate scheduler [31]. Target models and shadow models are trained for 60 epochs on CIFAR-10 and CINIC-10, 100 epochs on CIFAR-100, and 30 epochs on Purchase100 Texas100. Data augmentation techniques such as random cropping and horizontal flipping are applied to reduce over-fitting.

A.3 Setup for MIAs.

We implemented seven MIAs using the setup described below, which we adopt as the **standard setting**. Unless explicitly stated otherwise, this setup is used consistently across all experiments presented in the paper.

- **LOSS [52].** This attack queries the target model F_T with all samples from $\mathcal{D}_A$, and then we use the average loss as the global threshold to make membership predictions.
- **Augmentation Attack [8].** We implement the Augmentation attack with 5 rotations as the augmentation technique. It uses a 2-layer MLP as the attack model.
- **Loss Trajectory Attack [30].** We distillate both the target model and the shadow model for 100 epochs. We perform model distillation on a distillation set $\mathcal{D}_A^d$ that's half of the auxiliary dataset $\mathcal{D}_A$. For the rest of the dataset $\mathcal{D}_A^s$ that's disjoint from $\mathcal{D}_A^d$, we partition it again into halves as the train (members) and test (non-members) set for the shadow model.
- **LiRA [3].** Our implementation trains 20 shadow models. Each shadow model is trained using half of the auxiliary dataset. We follow the same procedure as Carlini et al. [3] to split the auxiliary dataset $\mathcal{D}_A$ such that for each datapoint in $\mathcal{D}_A$, it's partitioned into a shadow training set 10 times and a shadow testing set 10 times. In LiRA's paper, they use 256 models for most experiments. In Appendix B.B, they introduce the estimation of global variance to improve the performance of LiRA with a smaller number of models, and demonstrate that 16 shadow models can perform on par with their best attack. We have also justified that the inconsistency of MIA is invariant to the number of shadow models used in Section 4.2. The version of LiRA considered in the paper

Target Model	CIFAR-10			CIFAR-100			CINIC-10		
	Train acc	Test acc	Gen Gap	Train acc	Test acc	Gen Gap	Train acc	Test acc	Gen Gap
ResNet-56	89.3%±6.1%	80.9%±2.3%	8.4%	87.0%±16.3%	49.6%±4.8%	37.4%	80.4%±0.3%	60.5%±0.4%	19.9%
VGG-16	96.7%±5.3%	84.4%±1.7%	12.4%	90.8%±15.8%	53.9%±5.0%	36.9%	98.8%±0.2%	65.5%±0.4%	33.3%
MobileNetV2	90.9%±7.6%	73.3%±2.7%	17.6%	88.1%±20.6%	39.7%±7.4%	48.4%	88.4%±0.5%	53.7%±0.5%	34.7%
WideResNet-32	83.3%±2.7%	75.9%±2.9%	7.4%	64.3%±10.0%	41.6%±3.1%	22.7%	66.9%±2.0%	56.6%±0.13%	10.3%

(a) Training statistics for existing models on CIFAR-10, CIFAR-100, and CINIC-10 datasets.

Target Model	Texas100			Purchase100		
	Train acc	Test acc	Gen Gap	Train acc	Test acc	Gen Gap
MLP	99.9%±0.0%	54.5%±0.3%	45.4%	100.0%±0.0%	78.6%±0.4%	21.4%

(b) Training statistics for the MLP model on Purchase-100 and Texas-100 datasets.

Table 2: Target model's training statistics across various datasets and architectures. All accuracies are averaged from 4 different sets of experiments, each with a distinct partition of the target and auxiliary datasets (available to the attacker). All models are trained and tested on 15,000 disjoint samples. Gen Gap represents the generalization gap between Top-1 training accuracy and testing accuracy.

Dataset	$\mathcal{D}_T^{train}$	$\mathcal{D}_T^{test}$	$\mathcal{D}_A$
CIFAR-10	15,000	15,000	30,000
CIFAR-100	15,000	15,000	30,000
CINIC-10	15,000	15,000	30,000
Purchase-100	15,000	15,000	30,000
Texas-100	15,000	15,000	30,000

Table 3: Dataset Partitioning Sizes. Each subset is disjoint from the others after partitioning.

is the "online" version. For a discussion of the "offline" version of LiRA, see Appendix Section D.

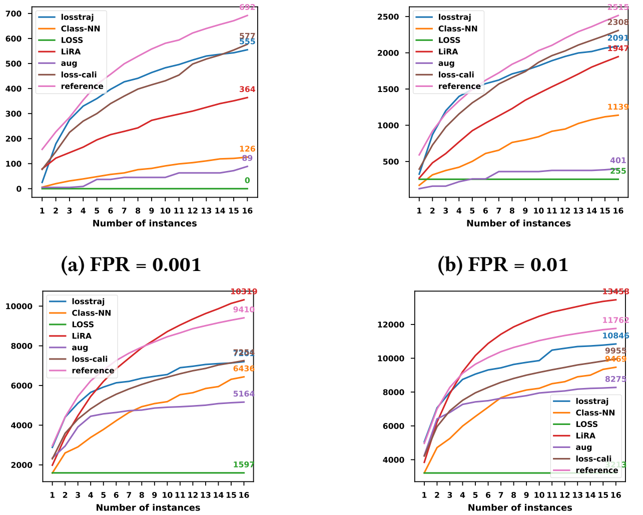
- **Class-NN [44].** This attack uses 10 shadow models and 3-layer MLP attack models. The number of attack models is equal to the number of labels, making it class-specific.
- **Difficulty Calibration Loss Attack [50].** This attack trains a single shadow model f_s on $\mathcal{D}_A$. The threshold τ for making membership predictions is determined by sampling 1,000 thresholds and selecting the one that maximizes accuracy on $\mathcal{D}_A$.
- **Reference Attack [51].** To save computational resources, Reference Attack shares the 20 shadow models with LiRA. Reference Attack's source code[1] re-used the source code[2] for LiRA. Therefore, we reuse the same shadow model set trained for LiRA for the reference attack.

B MIA Method Disparity at Different FPR

In Appendix Figure 11, the trend of Jaccard Similarity shows a positive correlation with FPR, indicating that disparity between attacks is highest at low FPRs and decreases as FPR increases. To further analyze this trend, we present the similarity between pairs of attacks across various FPR levels in Appendix Figure 20.

The similarity between attacks, measured in terms of both coverage and stability, remains quite low overall. This is particularly evident for stability at FPR = 0.001 (Appendix Figure 20e), where half of the attack pairs exhibit similarity values below 0.004, and the highest observed similarity is just 0.08. As FPR increases, the patterns of high similarity between certain pairs of attacks persist. For example, LiRA and the Reference Attack, LiRA and the Calibration Loss Attack, and the Calibration Loss Attack and the Loss Trajectory Attack consistently show relatively high correlation across different FPR levels. Conversely, pairs with high disparity, such as the LOSS Attack and the Calibration Loss Attack or the LOSS Attack and the Loss Trajectory Attack, consistently demonstrate low similarity across all analyzed FPR levels (e.g., Appendix Figure 20d).

We conclude that the inherent characteristics of each attack drive their disparity, with some attacks naturally aligning due to shared



(a) FPR = 0.001

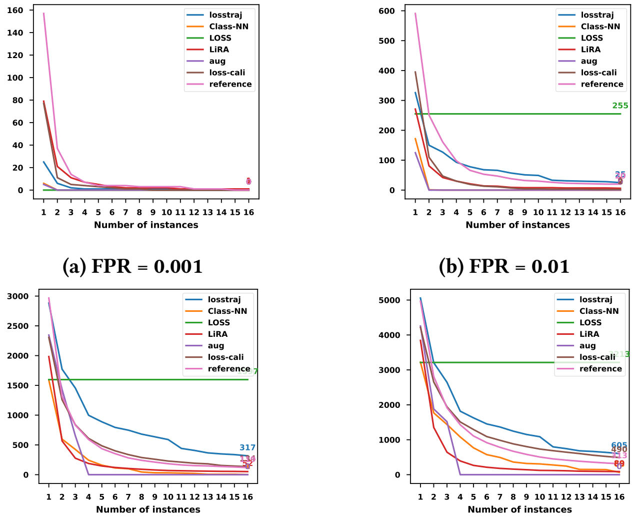
(b) FPR = 0.01

(c) FPR = 0.1

(d) FPR = 0.2

Figure 18: Trend of Coverage with Varying Numbers of Instances for Each MIA. Each subplot represents the coverage of instances with different FPRs.).

[1] https://github.com/privacytrustlab/ml_privacy_meter/tree/295e7e37e889e12df4083b812f71ed2e2ddd8b4a/research/2022_enhanced_mia
[2] https://github.com/tensorflow/privacy/tree/master/research/mi_lira_2021



(a) FPR = 0.001

(b) FPR = 0.01

(c) FPR = 0.1

(d) FPR = 0.2

Figure 19: Trend of Stability. Each line shows the stability of the respective attack across varying numbers of instances, following the setup in Figure 18.

methodologies or signals, while others diverge significantly. This underscores the importance of leveraging a diverse set of attacks in privacy auditing to capture a comprehensive view of membership vulnerabilities.

C Ensemble

C.1 Cost Analysis

As detailed in the main body of the paper, we model the computational cost of each attack in terms of GPU minutes. Our experiments were conducted on Nvidia L40S GPUs, with both target and shadow models being ResNet-56. Below, we outline the parameters for each attack that influence compute time:

- **LiRA (580 minutes)**: Shadow model training is performed with a batch size of 512, while queries to the target and shadow models use a batch size of 256. Each query includes 18 augmentations. A total of 20 shadow models are prepared for each instance.
- **Loss Trajectory Attack (17 minutes)**: Includes shadow model training and distillation for both the target and shadow models, with a batch size of 512.
- **Calibration Loss Attack (5 minutes)**: Shadow model training is conducted with a batch size of 512, and losses are queried sequentially with a batch size of 1.
- **Reference Attack (540 minutes)**: Shadow model training uses a batch size of 512, and queries to the target and shadow models use a batch size of 256. Similar to LiRA, 20 shadow models are prepared for each instance.

As discussed in Section 5.3.1, the Reference Attack and LiRA share the same shadow models. To avoid double-counting in our cost analysis, the training cost for shadow models (540 minutes) is deducted when both LiRA and the Reference Attack are included in an ensemble combination.

We extend our analysis of the cost-performance relationship (Section 5.3.2) in two key dimensions:

C.1.1 Attack Combination. In Appendix Figure 22, each performance sample is represented by a distinct marker shape, corresponding to the attack combinations detailed in the legend. When comparing ensembles with the same number of instances (indicated by the same color), the ensemble combining all four attacks consistently dominates other combinations. This finding highlights that all three ensemble strategies can effectively utilize the computational cost associated with employing diverse attacks.

C.1.2 Number of Instances. The general trend indicates that the best-performing ensemble does not always correspond to the one using the most instances. Notably, for the Coverage ensemble, the relationship between cost and performance is sometimes inverse. This occurs because the multi-instance coverage step (Equation 13) trades an increase in the false positive rate for a higher true positive rate. As a result, the ensemble's TPR at low FPR is reduced, impacting performance. For the other two ensembles, more instances are usually associated with better performance.

From our analysis, we know our ensemble would always benefit from utilizing more attacks (in the multi-attack step), but not always from more instances (in the multi-instance step). We encourage researchers who wish to adapt our ensemble to prepare attacks with multi-instance to their compute budget, and then perform MIA on a held-out set to find their desired ensemble performance.

C.2 Ensemble on Other Model Architectures

Our ensemble methods demonstrate consistent improvements across three additional model architectures: WideResNet-32, MobileNetV2, and VGG-16, as shown in Table 7. This table extends Table 1 by evaluating AUC, Accuracy, and TPR@0.1% FPR metrics across CIFAR-10, CIFAR-100, and CINIC-10 datasets.

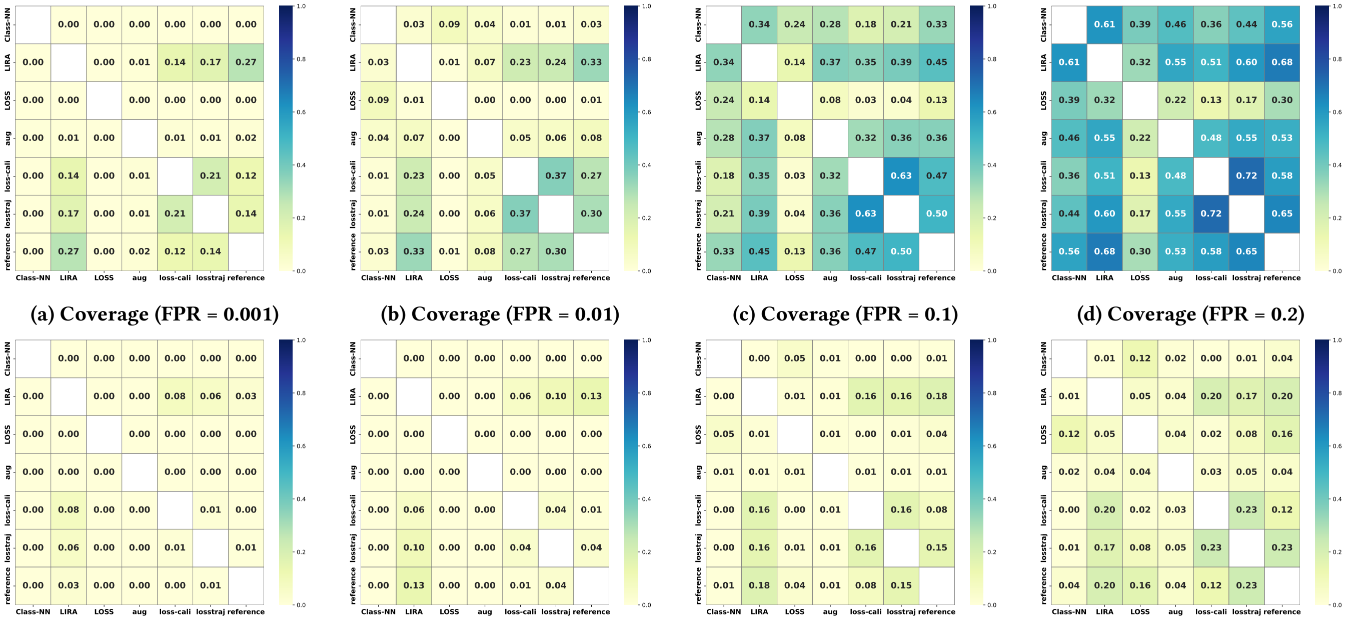
The Stability Ensemble and Majority Voting Ensemble consistently outperform single-instance attacks across all metrics and datasets. The Majority Voting Ensemble achieves performance comparable to the Stability Ensemble, particularly at low FPR, while maintaining competitive TPR at higher FPR levels. This highlights its balanced ability to draw strengths from both Stability and Coverage ensembles.

In contrast, the Coverage Ensemble demonstrates relatively lower performance on most metrics, particularly at 0.1% FPR, where its TPR does not consistently surpass that of single-instance attacks. This behavior reflects its inherent trade-off, prioritizing broader member coverage at the expense of precision.

Overall, the Stability Ensemble delivers the highest precision and robustness across all architectures, making it the most reliable option when low FPR and high precision are critical. The Majority Voting Ensemble provides a strong alternative, particularly for scenarios requiring balanced performance across multiple metrics.

C.3 Ensemble on Additional Datasets

In addition to Table 1, we have also evaluated the performance of our ensemble on two additional datasets mentioned in Section 4.1. Appendix Table 5. It should be that our ensemble consistently outperforms single-instance attacks on these two additional datasets.



(a) Coverage (FPR = 0.001) **(b) Coverage (FPR = 0.01)** **(c) Coverage (FPR = 0.1)** **(d) Coverage (FPR = 0.2)**

(e) Stability (FPR = 0.001) **(f) Stability (FPR = 0.01)** **(g) Stability (FPR = 0.1)** **(h) Stability (FPR = 0.2)**

Figure 20: Disparity of Membership Inference Attacks Across Different FPRs. The values represent the average Jaccard similarity across 4 experimental runs. Coverage and stability for each attack are calculated with 6 instances at varying FPR values ($0.001, 0.01, 0.1, 0.2$**), illustrating how similarity changes across thresholds.**

Attack	Shadow Models	Attack Models	Other Factors Involving Randomness	Inference Classifier
Augmentation attack [8]	1	1	Data augmentation	Attack model
Loss trajectory attack [30]	1	1	Model distillation	Attack model
LiRA [3]	20	0	None	Likelihood-ratio test
Reference attack [51]	20	0	None	Hypothesis test
Class-NN [44]	10	= # of classes	None	Attack model
LOSS [52]	0	0	None	Global threshold
Difficulty calibration loss attack [50]	1	0	None	Global threshold

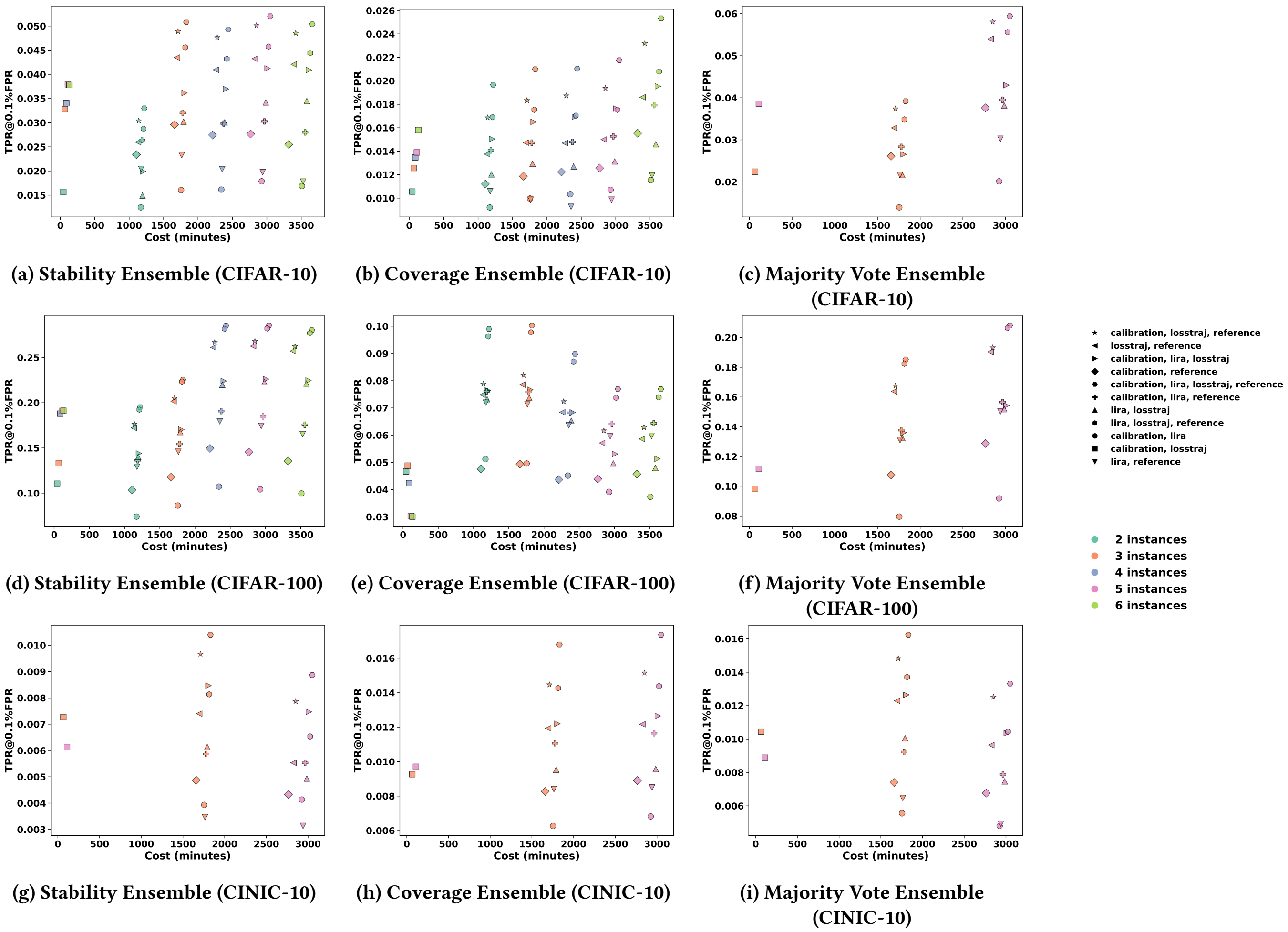
Table 4: A Summary of Membership Inference Attacks. The set-up follows the standard setting in Appendix Section A.3.

C.4 Performance under Auxiliary-Target Distribution Mismatch

For CINIC-10, we additionally consider a more challenging setting where the shadow model is trained on a different data distribution from the target model. Specifically, the auxiliary dataset $\mathcal{D}_A$ consists only of CIFAR-10 subsets from CINIC-10, while the target dataset $\mathcal{D}_T$ consists of ImageNet subsets, with each subset containing 30,000 samples. The results are presented in Appendix Table 6. Under this distribution mismatch, all base attacks perform significantly worse compared to their performance on CINIC-10 without such a mismatch, while our ensemble still significantly outperforms the individual attacks (see Appendix Table 1).

D LiRA Online v.s. Offline

In Carlini et al. [3], both online and offline versions of LiRA were proposed. For a given sample (x, y), the online version trains half of the shadow models with (x, y) included in the training set (IN models), and the other half with (x, y) excluded (OUT models). The confidence scores from both sets are then used to perform a likelihood ratio test. This setup requires retraining IN models for each new inference sample, making the online version computationally expensive. To improve efficiency, Carlini et al. [3] introduced an offline version that only trains OUT models and performs a one-sided hypothesis test. While this approach avoids the additional randomness introduced by IN models, it still exhibits similarly low consistency, as demonstrated in Appendix Figure 25. The target model is CIFAR-10, and the offline LiRA shares the same 20 shadow models used for online LiRA, but it only obtains confidence scores in OUT models.



(a) Stability Ensemble (CIFAR-10)

(b) Coverage Ensemble (CIFAR-10)

(c) Majority Vote Ensemble (CIFAR-10)

(d) Stability Ensemble (CIFAR-100)

(e) Coverage Ensemble (CIFAR-100)

(f) Majority Vote Ensemble (CIFAR-100)

(g) Stability Ensemble (CINIC-10)

(h) Coverage Ensemble (CINIC-10)

(i) Majority Vote Ensemble (CINIC-10)

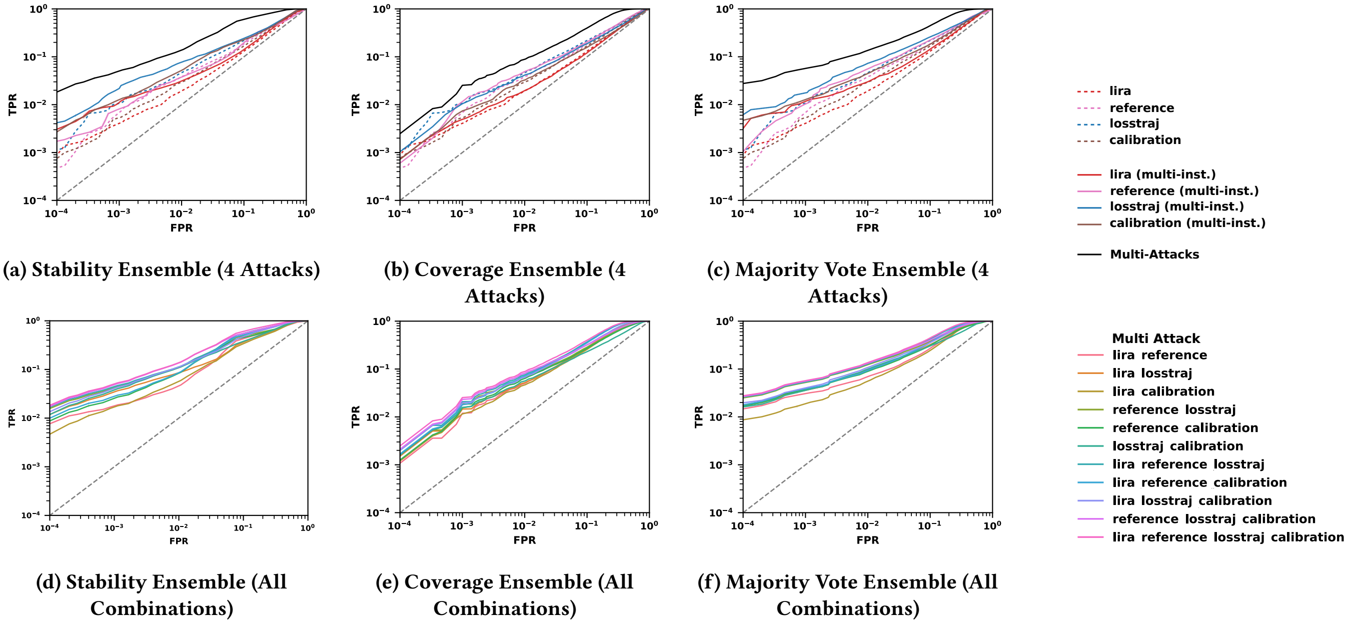
Figure 22: Comparison of ensemble strategies (Stability Ensemble, Majority Vote Ensemble, Coverage Ensemble) across different datasets. Each instance makes membership inferences at FPR=0.01. The target model is ResNet-56. Each row corresponds to a dataset, and each column represents an ensemble strategy. The marker legend explains the markers used for different attack combinations, and the color legend represents the number of instances. For the Majority Vote, we have skipped even numbers of instances since the majority vote with an even number could lead to a tie.

E MIA on Out of Distribution Sample

In this section, we extend our analysis to out-of-distribution (OOD) samples, focusing on the MIAs' disparities on extreme OOD members. Following prior work [1, 3], we create a set of mislabeled data points, referred to as *canary samples*, to simulate such cases. Specifically, we sample a subset $\mathcal{D}_C$ from the target dataset $\mathcal{D}_T$ and relabel each point $(x, y) \in \mathcal{D}_C$ as (x, y') such that $y' \neq y$. These mislabeled points are inserted back into $\mathcal{D}_T$ and used as members in the training set. Since the target model must overfit to these samples to classify them correctly, they represent highly vulnerable members.

We expect attacks to have higher consistency on canary members compared to regular members, as they should be easier to identify and most instances should agree on their membership. We also expect attacks to have higher similarity in predicting canary samples for the same reason.

In our experiment, we inject 300 canary samples into CIFAR-10. We exclude the LOSS attack, as its predictions are deterministic across different instances. In Appendix Figure 26, the regular members refer to samples in the target training set that are not relabeled. Some attacks show higher consistency on canary members than on regular members. For example, the Calibration Loss Attack achieves consistency greater than 0.9. This is because the loss of mislabeled (canary) data is significantly higher on the shadow model, while the target model has memorized the new label (y'). Therefore, this task becomes trivial for the Calibration Loss Attack. In general, identifying an extreme outlier member is trivial for attacks that compare the loss on the target model to the loss on the shadow model—this is the case for both LiRA and the Calibration Loss Attack. It explains the large gap in consistency between regular and canary members for these two attacks. For the Loss Trajectory



(a) Stability Ensemble (4 Attacks)

(b) Coverage Ensemble (4 Attacks)

(c) Majority Vote Ensemble (4 Attacks)

(d) Stability Ensemble (All Combinations)

(e) Coverage Ensemble (All Combinations)

(f) Majority Vote Ensemble (All Combinations)

Figure 24: Comparison of ensemble ROC curves for different attack combinations on the CIFAR-10 dataset. Each row compares different ensemble strategies (Stability Ensemble, Coverage Ensemble, Majority Vote Ensemble) for a given set of attacks. The row on the bottom shows different combinations of attacks used in the ensemble. All Ensemble are performed with 6 instances.

		TEXAS-100			PURCHASE-100		
Ensemble Level	Attack	AUC	ACC	TPR	AUC	ACC	TPR
Single-instance	losstraj	0.789	0.722	0.014	0.667	0.629	0.005
	reference	0.841	0.785	0.066	0.729	0.690	0.014
	lira	0.836	0.750	0.078	0.726	0.664	0.010
	calibration	0.757	0.694	0.002	0.661	0.639	0.002
Multi-inst. Coverage	losstraj	0.732	0.667	0.018	0.624	0.599	0.003
	reference	0.873	0.789	0.068	0.757	0.684	0.017
	lira	0.826	0.561	0.005	0.711	0.651	0.010
	calibration	0.759	0.698	0.006	0.602	0.592	0.003
Multi-inst. Coverage	losstraj	0.789	0.715	0.027	0.678	0.635	0.004
	reference	0.854	0.789	0.083	0.744	0.691	0.020
	lira	0.848	0.759	0.072	0.743	0.674	0.014
	calibration	0.759	0.698	0.005	0.662	0.637	0.004
Multi-inst. Stability	losstraj	0.849	0.766	0.046	0.725	0.657	0.006
	reference	0.808	0.780	0.118	0.731	0.688	0.018
	lira	0.853	0.761	0.079	0.747	0.677	0.011
	calibration	0.758	0.696	0.008	0.720	0.668	0.002
Multi-attack	Coverage	0.952	0.909	0.094	0.850	0.789	0.026
	Stability	0.902	0.816	0.165	0.880	0.812	0.046
	Maj-vote	0.740	0.718	0.003	0.694	0.647	0.008

Table 5: Performance Comparison for additional datasets in addition to Table 1. The model architecture is MLP.

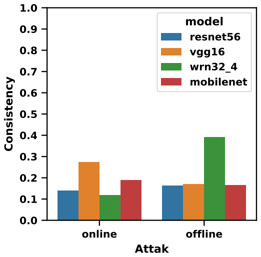


Figure 25: Consistency of LiRA Online vs LiRA Offline.

		CINIC-10		
Ens. Lvl	Attack	AUC	ACC	TPR
Single-inst.	losstraj	0.501	0.503	0.001
	reference	0.506	0.507	0.001
	lira	0.505	0.505	0.001
	calibration	0.499	0.503	0.001
Multi-attack	Coverage	0.781	0.726	0.002
	Stability	0.796	0.730	0.011
	Maj-vote	0.775	0.708	0.008

Table 6: Performance on CINIC-10 dataset with distribution shift. TPR is measured at 0.1% FPR.

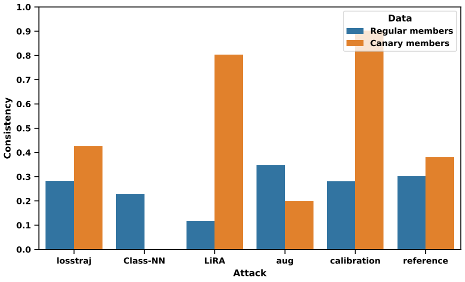


Figure 26: Consistency on Outliers VS Regular Samples.

and Reference Attacks, this task is less trivial but still easier, as canary samples are more memorized. However, the Class-NN Attack and Augmentation Attack exhibit decreased performance, since these attacks don't have those extreme OOD samples presented in their attack training sets. In Appendix Figure 28, we observe that the Class-NN and the Augmentation Attack perform worse

		CIFAR-10			CIFAR-100			CINIC-10		
Ens. Lvl	**Attack**	**AUC**	**ACC**	**TPR**	**AUC**	**ACC**	**TPR**	**AUC**	**ACC**	**TPR**
Single-inst.	losstraj	0.589	0.560	0.005	0.716	0.650	0.007	0.585	0.565	0.002
	reference	0.536	0.542	0.003	0.651	0.654	0.017	0.542	0.543	0.003
	lira	0.527	0.525	0.001	0.661	0.624	0.008	0.534	0.527	0.001
	calibration	0.572	0.551	0.004	0.671	0.631	0.006	0.567	0.552	0.002
Multi-inst. Coverage	losstraj	0.556	0.535	0.003	0.678	0.615	0.009	0.572	0.546	0.002
	reference	0.559	0.557	0.002	0.701	0.676	0.010	0.557	0.552	0.003
	lira	0.516	0.524	0.001	0.654	0.623	0.009	0.525	0.525	0.001
	calibration	0.545	0.528	0.003	0.637	0.595	0.006	0.563	0.548	0.002
Multi-inst. Maj-vote	losstraj	0.594	0.561	0.006	0.726	0.650	0.010	0.597	0.570	0.003
	reference	0.543	0.546	0.003	0.664	0.664	0.023	0.546	0.548	0.003
	lira	0.531	0.531	0.002	0.671	0.632	0.015	0.539	0.531	0.002
	calibration	0.587	0.563	0.005	0.690	0.637	0.010	0.565	0.550	0.002
Multi-inst. Stability	losstraj	0.605	0.571	0.005	0.767	0.701	0.012	0.593	0.567	0.003
	reference	0.523	0.528	0.003	0.509	0.619	0.016	0.530	0.533	0.003
	lira	0.542	0.537	0.003	0.671	0.629	0.020	0.544	0.533	0.003
	calibration	0.587	0.565	0.005	0.734	0.674	0.013	0.568	0.551	0.002
Multi-attack	Coverage	0.751	0.705	0.002	0.764	0.695	0.016	0.892	0.825	0.013
	Stability	0.839	0.741	0.034	0.935	0.868	0.125	0.754	0.698	0.003
	Maj-vote	0.831	0.758	0.028	0.925	0.856	0.076	0.832	0.770	0.006

(a) WideResNet-32

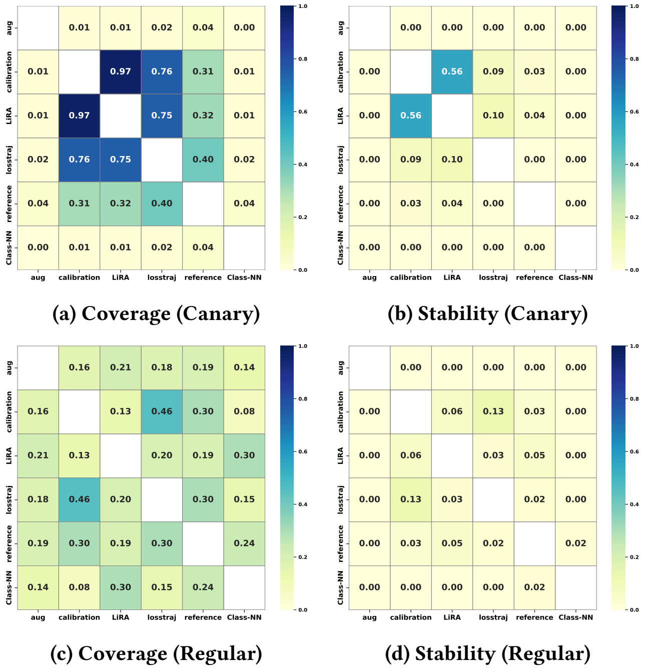
		CIFAR-10			CIFAR-100			CINIC-10		
Ens. Lvl	**Attack**	**AUC**	**ACC**	**TPR**	**AUC**	**ACC**	**TPR**	**AUC**	**ACC**	**TPR**
Single-inst.	losstraj	0.703	0.640	0.018	0.497	0.506	0.001	0.757	0.690	0.022
	reference	0.652	0.632	0.006	0.926	0.883	0.220	0.710	0.681	0.006
	lira	0.622	0.595	0.009	0.935	0.857	0.260	0.689	0.638	0.017
	calibration	0.633	0.603	0.005	0.772	0.730	0.012	0.715	0.659	0.003
Multi-inst. Coverage	losstraj	0.640	0.584	0.014	0.897	0.830	0.072	0.721	0.646	0.020
	reference	0.681	0.639	0.009	0.935	0.868	0.222	0.761	0.692	0.010
	lira	0.615	0.585	0.015	0.935	0.856	0.239	0.697	0.639	0.019
	calibration	0.561	0.542	0.005	0.713	0.660	0.014	0.672	0.625	0.002
Multi-inst. Stability	losstraj	0.706	0.662	0.031	0.706	0.756	0.054	0.713	0.685	0.057
	reference	0.621	0.614	0.021	0.901	0.891	0.371	0.660	0.654	0.021
	lira	0.647	0.611	0.020	0.952	0.882	0.331	0.719	0.665	0.020
	calibration	0.722	0.667	0.014	0.894	0.863	0.050	0.770	0.708	0.008
Multi-inst. Maj-vote	losstraj	0.746	0.664	0.029	0.961	0.894	0.100	0.823	0.735	0.042
	reference	0.703	0.666	0.016	0.945	0.897	0.252	0.771	0.722	0.016
	lira	0.652	0.614	0.017	0.956	0.887	0.310	0.738	0.676	0.022
	calibration	0.647	0.610	0.012	0.778	0.690	0.045	0.740	0.677	0.005
Multi-attack	Coverage	0.848	0.781	0.028	0.956	0.887	0.285	0.916	0.849	0.053
	Stability	0.909	0.834	0.091	0.992	0.957	0.545	0.892	0.829	0.032
	Maj-vote	0.887	0.817	0.085	0.986	0.949	0.460	0.914	0.849	0.048

(b) MobileNetV2

		CIFAR-10			CIFAR-100			CINIC-10		
Ens. Lvl	**Attack**	**AUC**	**ACC**	**TPR**	**AUC**	**ACC**	**TPR**	**AUC**	**ACC**	**TPR**
Single-inst.	losstraj	0.668	0.605	0.021	0.826	0.738	0.031	0.779	0.698	0.027
	reference	0.663	0.632	0.016	0.874	0.821	0.005	0.775	0.717	0.028
	lira	0.651	0.605	0.021	0.888	0.793	0.140	0.771	0.693	0.023
	calibration	0.636	0.599	0.003	0.728	0.702	0.008	0.706	0.660	0.004
Multi-inst. Coverage	losstraj	0.609	0.563	0.013	0.743	0.664	0.025	0.711	0.638	0.012
	reference	0.657	0.619	0.016	0.853	0.777	0.005	0.763	0.687	0.045
	lira	0.643	0.593	0.028	0.891	0.787	0.149	0.766	0.678	0.025
	calibration	0.565	0.548	0.003	0.641	0.598	0.006	0.652	0.619	0.003
Multi-inst. Stability	losstraj	0.722	0.646	0.048	0.838	0.792	0.079	0.830	0.741	0.070
	reference	0.673	0.649	0.037	0.893	0.855	0.190	0.794	0.752	0.075
	lira	0.670	0.623	0.039	0.903	0.810	0.194	0.799	0.720	0.043
	calibration	0.703	0.647	0.011	0.875	0.821	0.026	0.769	0.718	0.007
Multi-inst. Maj-vote	losstraj	0.672	0.604	0.031	0.857	0.757	0.047	0.796	0.703	0.037
	reference	0.686	0.647	0.035	0.894	0.828	0.123	0.811	0.735	0.064
	lira	0.680	0.626	0.032	0.919	0.826	0.180	0.810	0.722	0.037
	calibration	0.646	0.606	0.006	0.729	0.687	0.018	0.704	0.653	0.004
Multi-attack	Coverage	0.740	0.718	0.003	0.926	0.843	0.132	0.924	0.853	0.085
	Stability	0.871	0.790	0.083	0.976	0.934	0.322	0.893	0.825	0.053
	Maj-vote	0.852	0.790	0.070	0.958	0.896	0.273	0.906	0.841	0.066

(c) VGG-16

Table 7: Performance of Ensemble for Different Architectures. TPR stands for True Positive Rate measured at 0.1% FPR. This table is an extension of Table 1.



(a) Coverage (Canary)

(b) Stability (Canary)

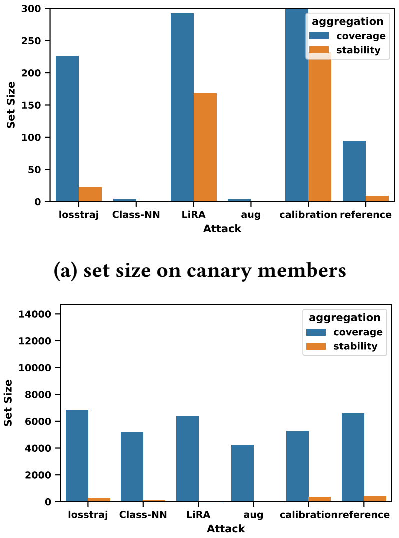
(c) Coverage (Regular)

(d) Stability (Regular)

Figure 27: Disparity of Membership Inference Attacks on Outliers (canary samples) VS regular samples. Instances are prediting at FPR=0.1.

on canary samples than on regular samples, further demonstrating their inability to handle extreme OOD samples.

In Appendix Figure 27, we observe that for canary samples, the Jaccard similarity between the coverage and stability of some attack pairs is significantly higher compared to regular samples. For example, LiRA and the Calibration Loss Attack exhibit a coverage similarity of 0.97, indicating that they are almost predicting the same set of members. In Appendix Figure 28a, we confirm that the coverage of LiRA and the Calibration Loss Attack nearly includes all canary members, which explains this high similarity.



(a) set size on canary members

(b) set size on regular members

Figure 28: Set size of coverage and stability on canary and regular members. The target dataset is CIFAR-10 with 300 canary members and 14,700 regular members.