A Unified Perspective on Adversarial Membership Manipulation in Vision Models

Supplementary Material

A. Related Works

A.1. Privacy Risks in Machine Learning

Machine learning (ML) models, particularly deep neural networks, have become integral to advancements in various high-stakes domains such as healthcare, genomics, and image recognition. These models' exceptional capacity to detect intricate patterns from large datasets has significantly advanced fields like healthcare, genomics, image recognition, network inference, and autonomous decision making [23, 26, 42, 52, 63].However, this proficiency is accompanied by notable privacy concerns, as these models can inadvertently memorize sensitive training data, posing significant privacy risks [12, 44, 51, 55]. Privacy breaches have emerged as a pervasive issue in modern ML, prompting calls for robust user data protection measures [16, 17, 53, 58]. In response, legislative bodies have enacted stringent privacy laws, such as the *General Data Protection Regulation* (GDPR) in the European Union, the *California Consumer Privacy Act* (CCPA) in the United States, and the *Personal Information Protection and Electronic Documents Act* (PIPEDA) in Canada, which legally mandate data privacy safeguards. Within this context, *Membership Inference Attacks* (MIAs) have gained prominence as tools for auditing the degree of information leakage from ML models with respect to their training datasets [7, 53, 64, 65, 67].

A.2. Membership Inference Attacks

Membership Inference Attacks (MIAs) aim to predict whether a specific training example was included in the training set of a given model. Existing MIAs [7, 53, 61, 64, 65, 67] can be naturally framed as hypothesis testing, where the Inferer attempts to distinguish between two hypotheses concerning the presence of a target sample (x, y) in the training data of a target model f. The hypothesis testing framework can be formalized by defining two hypotheses: **(I). Null hypothesis** H_0 **:** The target sample (x, y) was not part of the training data of the model f; **(II). Alternative hypothesis** H_1 **:** The target sample (x, y) was part of the training data of the model f. In such a hypothesis testing problem, the design of test statistics plays a critical role in distinguishing between members and non-members. For example, the *loss attack* [65] uses the sample loss as the test statistic, while the *Likelihood Ratio Attack* (LiRA) [7] uses the likelihood ratio. By setting an appropriate threshold for these statistics, MIAs determine whether a given data point is a member of the training set. Specifically, the decision rule can be written as:

$$I(x, y) = \mathbf{1}[S(x, y) > \tau] \tag{12}$$

where $\mathbf{1}$ is the indicator function, τ is a tunable decision threshold, and S is the test statistic. In this context, the Inferer makes a prediction based on whether the test statistic $S(x, y)$ exceeds the threshold τ. Early methods, such as the loss attack [65], distinguished between members and non-members by setting a loss threshold. The insight here is that data points with lower loss values are more likely to belong to the training set:

$$I(x, y) = \mathbf{1}[-\ell(x, y) > \tau] = \mathbf{1}[\log(p_y) > \tau], \tag{13}$$

where $l(\cdot, \cdot)$ is the cross-entropy loss function and p_y is the softmax probability of the true label y. The shadow model-based attack [53] involves training similar shadow models to simulate the target model's behavior. Building on these early approaches, subsequent works expanded these techniques to more extensive scenarios [10, 12, 24, 32, 33, 45, 50, 55]. Recent advancements have incorporated the concept of sample hardness, with approaches like Watson et al. [61] adjusting the loss based on sample difficulty. Later studies [7, 64, 67] further refined these techniques. Among them, the *Likelihood Ratio Attack* (LiRA) [7] represents the state-of-the-art approach, offering superior performance when sufficient shadow models are available.

Likelihood Ratio Attack In this section, we introduce *Likelihood Ratio Attack* LiRA. LiRA exploits the behavioral differences between two types of shadow models for each sample: those trained with the target sample (IN-Models) and those trained without it (OUT-Models). The Inferer trains multiple shadow models for each target sample, dividing them into two groups: one group is trained with the target sample, and the other is not. The objective of LiRA is to determine whether the target model belongs to the IN-Models group or the OUT-Models group. We consider distributions over models:

$$\mathbb{Q}_{\text{out}}(x, y) = \{f \leftarrow \mathcal{T}(D \setminus \{(x, y)\}) \mid D \leftarrow \mathbb{D}\},$$

$$\mathbb{Q}_{\text{in}}(x, y) = \{f \leftarrow \mathcal{T}(D \cup \{(x, y)\}) \mid D \leftarrow \mathbb{D}\}.$$

where $\mathbb{Q}_{\text{out}}(x, y)$ and $\mathbb{Q}_{\text{in}}(x, y)$ represent models trained excluding or including the target sample (x, y), respectively. The Inferer's task is to determine whether the model f was drawn from $\mathbb{Q}_{\text{out}}(x, y)$ or $\mathbb{Q}_{\text{in}}(x, y)$. In both cases, we can apply the Neyman-Pearson lemma [47] to establish that the most powerful test at a fixed false-positive rate is achieved by

thresholding the *likelihood ratio* between the two hypotheses. Specifically, the likelihood ratio is given by:

$$\Lambda(f; x, y) = \frac{p(\text{observed data} \mid H_1)}{p(\text{observed data} \mid H_0)}. \tag{14}$$

For sample-tailored MIAs, this becomes:

$$\Lambda(f; x, y) = \frac{p(f \mid \mathbb{Q}_{\text{in}}(x, y))}{p(f \mid \mathbb{Q}_{\text{out}}(x, y))}, \tag{15}$$

where $p(f \mid \mathbb{Q}_b(x, y))$ denotes the probability density function (PDF) of the model f under the distribution $\mathbb{Q}_b(x, y)$. However, directly computing these likelihood ratios is generally intractable because the distributions $\mathbb{Q}_{\text{in}}$ and $\mathbb{Q}_{\text{out}}$ are not analytically known. Carlini et al. [7] provided a more tractable one-dimensional statistic: the logit-scaled predicted confidence of the model f on the sample (x, y), denoted as

$$\phi(p_y(x)) = \log\left(\frac{p_y(x)}{1 - p_y(x)}\right). \tag{16}$$

Carlini et al. [7] defined approximate distributions $\tilde{\mathbb{Q}}_{\text{in}}$ and $\tilde{\mathbb{Q}}_{\text{out}}$ as the distributions of $\phi(p_y(x))$ when (x, y) is included or excluded from the training data. The likelihood ratio then simplifies to:

$$\Lambda(f; x, y) = \frac{p(\phi(p_y(x)) \mid \tilde{\mathbb{Q}}_{\text{in}})}{p(\phi(p_y(x)) \mid \tilde{\mathbb{Q}}_{\text{out}})}. \tag{17}$$

This reduction to a one-dimensional statistic allows for efficient computation of the likelihood ratio with f. Specifically, the decision rule in LiRA can be written as:

$$I(x, y) = \mathbf{1}[\Lambda(f; x, y) > \tau] = \mathbf{1}[\frac{p(\phi(p_y(x)) \mid \tilde{\mathbb{Q}}_{\text{in}})}{p(\phi(p_y(x)) \mid \tilde{\mathbb{Q}}_{\text{out}})} > \tau] \tag{18}$$

Assumptions of LiRA. Despite the superior performance, the effectiveness of LiRA relies on several assumptions:

I. Dependence on shadow models. LiRA needs to train a substantial number of shadow models to accurately approximate the distributions of IN-Models and OUT-Models.

II. Knowledge of the training details. LiRA assume that Inferer possesses detailed knowledge of the target model's training environment, including hyperparameters and architectural specifics.

A.3. Overfitting Drives MIA Effectiveness

A key factor influencing the success of MIAs is the overfitting, a phenomenon well-documented in existing literature [9, 32, 51, 53, 65]. Overfitting arises when a model learns to recognize not just general patterns but also noise or overly specific features from its training data, thus enhancing performance on seen data while impairing its ability to generalize to unseen data. This issue is especially prominent in complex models and when training datasets are insufficiently representative of the broader data distribution [3]. Deep learning models are particularly vulnerable to this issue due to their extensive parameterization, which often leads to the memorization of intricate details from the training data [6, 43, 54, 68]. These models, when trained on non-representative datasets, exhibit discrepancies in behavior between training and non-training data, and MIAs exploit these disparities. Theoretical work has shown that overfitting directly contributes to these behavioral differences, which MIAs can detect and use to infer whether a given data point was included in the training set [1].

A.4. Adversarial Attacks.

Recent studies have increasingly highlighted the vulnerability of neural networks to adversarial attacks—slightly altered inputs that are strategically crafted to induce misclassification [4, 30, 60, 70]. These attacks pose a serious threat to security-critical systems, such as autonomous driving and medical diagnostics [8, 18, 37, 48, 57]. A seminal work by Szegedy et al. [57] first pointed out the existence of adversarial examples: given a valid input x with its true label y and a trained classifier f, it is often possible to find another input x' such that $f(x') \neq y$ yet x, x' are close according to some distance metric. This insight laid the foundation for understanding how neural networks can be deceived by small but strategically crafted perturbations. In this paper, we focus on the ℓ_∞ distance metric, and the example x' is referred to as the adversarial example.

Fast Gradient Sign Method (FGSM). Based on the study of Szegedy et al. [57], Goodfellow et al. [20] put forward the Fast Gradient Sign Method (FGSM): given an image x, FGSM sets

$$x' = x + \epsilon \cdot \text{sign}(\nabla \ell(f(x), y)), \tag{19}$$

where ϵ is chosen to be sufficiently small to ensure that the difference is unnoticeable by humans; y is the true label; ℓ is a loss function that measures the performance of the classifier. A common loss function ℓ in Eq. (19) is the *Cross-Entropy* (CE) loss:

$$\text{CE}(f, x, y) = -\log(p_y), \tag{20}$$

where p is the softmax of logits of the model outputs. For each pixel of the image, FGSM uses the gradient of the loss function to determine in which direction the pixel's intensity should be increased or decreased and minimizes the loss function.

Projected Gradient Descent Attack (PGD). Madry et al. [39] introduced a simple refinement of FGSM, which is the projected gradient descent (PGD) attack. Instead of taking a single step of size ϵ in the direction of the gradient sign,

multiple smaller steps are taken in PGD (the result is clipped by the same ϵ). Specifically, we start with setting $x^0 = x$, and then in each iteration:

$$x'_{(t+1)} = \Pi_{\mathcal{B}_\epsilon[x_{(0)}]}(x'_{(t)} + \alpha \operatorname{sign}(\nabla_{x'_{(t)}} \ell(f(x'_{(t)}), y)),$$

$t = 0, 1, 2, \ldots$, where

$$\mathcal{B}_\epsilon[x] = \{x' \mid d_\infty(x, x') \leq \epsilon\},$$

is the closed ball of radius $\epsilon > 0$ centered at x; the $x_{(0)}$ refers to the starting point which corresponds to the natural example x; $x^{(t)}$ is the adversarial example at step t; $\Pi_{\mathcal{B}_\epsilon[x^{(0)}]}(\cdot)$ is the projection function that projects the adversarial variant back to the ϵ-ball centered at $x^{(0)}$ if necessary; the L_∞ distance metric is $d_\infty(x, x') = \|x - x'\|_\infty$; and ℓ is *cross entropy* (CE) loss.

Carlini and Wagner attack (CW). Carlini and Wagner [5] observed the phenomenon of gradient vanishing in the widely used CE loss for potential failure. Motivated by this, [5] replaced the CE loss with several possible choices. Among these choices, the widely used one for the untargeted attack is

$$\mathrm{CW}(x, y) = -z_y(x') + \max_{i \neq y} z_i(x'). \tag{21}$$

where z is the logits of the model outputs.

AutoAttack and Minimum Margin attack (MM). Croce and Hein [13] claimed that the fixed step size and the lack of diversity in attack methods are the main reasons for the limitations of previous studies, and they put forward an ensemble of diverse attacks called AutoAttack. Gao et al. [19] argued that the high computational cost of *AutoAttack* is unnecessary for identifying *the most adversarial example*:

Definition 6 (The most adversarial example)**.** *Given a natural example x with its true label y and, the most adversarial example x^* within $\mathcal{B}_\epsilon[x]$ is defined as:*

$$\forall x' \in \mathcal{B}_\epsilon[x], x^* = \arg\max_{x'} -(z_y(x') - \max_{i \neq y} z_i(x')), \tag{22}$$

where $\mathcal{B}_\epsilon[x] = \{x' \mid d_\infty(x, x') \leq \epsilon\}$ is the closed ball of radius $\epsilon > 0$ centered at x; $z_y(x') = f(x')_y$; $z_i(x') = f(x')_i$.

A.5. Traditional Adversarial Data Detection

In addition to enhancing model robustness through improved adversarial training [11, 60, 62, 69], recent research has focused on the detection of adversarial data. Most of these approaches rely on features extracted from *deep neural networks* (DNNs) and aim to train classifiers that distinguish between natural and adversarial data. Several recent advancements in adversarial data detection include the use of a cascade detector based on the *Principal Component Analysis* (PCA) projection of activations [34], detection subnetworks based on activations [41], and logistic regression detectors that utilize *Kernel Density* (KD) and *Bayesian Uncertainty* (BU) features [21]. Other methods include an augmented neural network detector that employs statistical measures, a learning framework designed to address previously unexplored vulnerabilities in models [49], and a characterization of adversarial data based on *local intrinsic dimensionality* (LID) [36]. Furthermore, generative classifiers based on Mahalanobis distance scores have also been proposed as effective tools for detecting adversarial data [31].

B. Proof of Theorem 1

Assume that the ϵ-ball is small such that the local curvature of $\ell \circ f$ around x can be well approximated by its second-order Taylor approximation, that is, there exists a Hessian matrix $\mathbf{H}$ and a gradient vector $\mathbf{g} = \nabla_x \ell(f(x), y)$ such that $\forall x' \in \mathcal{B}_\epsilon[x]$,

$$\ell(f(x'), y) \approx \ell(f(x), y) + \mathbf{g}^T(x' - x) + \frac{1}{2}(x' - x)^T \mathbf{H}(x' - x).$$

After taking one step of signed gradient descent with respect to the input sample:

$$x' = x - \alpha \cdot \operatorname{sign}(\nabla_x \ell(f(x), y)), \tag{23}$$

We analyze the change of gradient norm in the Taylor approximation.

$$\begin{aligned}\|\nabla_{x'} \ell(f(x'), y)\|^2 &\approx \|\mathbf{H}(x' - x) + \mathbf{g}\|^2 \\ &= \|\mathbf{g}\|^2 + 2\mathbf{g}^T \mathbf{H}(x' - x) + (x' - x)^T \mathbf{H}^2 (x' - x).\end{aligned}$$

Substituting (23), we obtain

$$\begin{aligned}\|\nabla_{x'} \ell(f(x'), y)\|^2 &= \|\nabla_x \ell(f(x), y)\|^2 \\ &\quad - 2\alpha \cdot \mathbf{g}^T \mathbf{H} \operatorname{sign}(\mathbf{g}) + \alpha^2 \operatorname{sign}(\mathbf{g})^T \mathbf{H}^2 \operatorname{sign}(\mathbf{g}).\end{aligned}$$

Thus, by letting

$$\alpha < \frac{2\mathbf{g}^T \mathbf{H} \operatorname{sign}(\mathbf{g})}{\operatorname{sign}(\mathbf{g})^T \mathbf{H}^2 \operatorname{sign}(\mathbf{g})},$$

we have the following inequality (approximately),

$$\|\nabla_{x'} \ell(f(x'), y)\| < \|\nabla_x \ell(f(x), y)\|. \tag{24}$$

C. The Risks of Adversarial Vulnerability of MIAs in Practical Scenarios

MIAs are increasingly used in practical pipelines for privacy auditing [43, 56] and model copyright verification [59]. However, our findings show that their adversarial vulnerability can lead directly to incorrect or even strategically

manipulated conclusions. We highlight several representative scenarios where fabricated membership can create tangible risks.

I. Misleading copyright or dataset-provenance verification. When MIAs are used as third-party tools for detecting copyright infringement [59], their susceptibility to adversarial perturbations creates a vulnerability for malicious exploitation. Under this setting, an adversary could deliberately add imperceptible perturbations to their own private images to fabricate membership. A vulnerable MIA would then falsely conclude that the target vision model was trained on these manipulated inputs. Such fabricated evidence can be used to launch spurious copyright claims or regulatory complaints, even when no data misuse has occurred. This exposes a direct path for legal and financial exploitation if MIAs are trusted as forensic tools.

II. Manipulating MIA-based training data extraction and competitive auditing. Several recent pipelines use MIAs to infer the composition of a competitor's training dataset or to verify compliance with data-deletion requests. In shared-data or competitive industrial settings, adversaries could upload imperceptibly perturbed images into a public or collaborative dataset. These fabricated members would cause MIA-based extraction tools to incorrectly infer that the model trained on these poisoned samples, thereby distorting the reconstructed training distribution. This undermines dataset-governance workflows and enables malicious parties to bias or sabotage competitors' auditing systems.

III. Exploiting MIA vulnerabilities in defensive or adversarial intelligence settings. Because fabricated members follow a distinctive gradient-norm collapse, defenders could intentionally craft such samples to confuse attackers attempting to audit their models using MIAs. For example, an organization concerned about model-stealing or data-reconstruction attacks may release selectively fabricated samples in a public-facing API. Attackers relying on MIA-based analysis would misinterpret these fabricated members as genuine training samples, leading them to incorrectly infer the model's training data or privacy weaknesses. This demonstrates that adversarial membership manipulation can be used not only offensively but also strategically in defensive contexts.

In all scenarios, the underlying risk stems from the same mechanism revealed in our analysis: adversarial perturbations can reliably push non-members into regions that MIAs interpret as strong evidence of membership. Without adversarially robust inference methods, MIAs cannot be reliably used in real-world privacy, copyright, or forensic workflows.

D. Finite-Difference Gradient Estimation

In the main paper, we assume white-box access when constructing the strongest Detector for **MFD**. This assumption allows direct computation of input gradients $\nabla_x \ell(f(x), y)$, which is essential for measuring the gradient-norm collapse phenomenon that distinguishes fabricated from true members. However, in many practical auditing scenarios, the model may only be accessible through black-box APIs that return confidence scores or class probabilities.

Black-box feasibility. Fortunately, the adversarial-robustness literature has established that input gradients can be reliably approximated in a black-box setting using finite-difference or score-based estimators such as NES (Natural Evolution Strategies) and SPSA. These methods exploit the key identity that for sufficiently small perturbation radius δ,

$$\nabla_x f(x) = \mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[\frac{f(x+\delta u) - f(x)}{\delta} u\right],$$

meaning that the directional response of the model to randomized perturbations already encodes information about the gradient. This enables gradient-norm estimation without requiring logits or intermediate activations—confidence scores alone are sufficient.

Finite-difference estimator. We follow the standard formulation of black-box gradient estimation in adversarial attacks and use the symmetric finite-difference estimator:

$$\widehat{\nabla_x f(x)} \;=\; \frac{1}{N}\sum_{i=1}^{N} \frac{f(x+\delta u_i) - f(x-\delta u_i)}{2\delta}\, u_i,$$

where u_i are sampled from $\mathcal{N}(0, I)$ and normalized to unit norm. The estimated gradient norm is then computed as:

$$\widehat{g}(x) \;=\; \left\| \widehat{\nabla_x \ell(f(x)}, y) \right\|.$$

This procedure requires only *model confidence queries* and therefore applies in the strictest black-box setting.

Practical configuration. For CIFAR-10 and ResNet-18, we find the following configuration offers a favorable trade-off between query cost and estimation accuracy:

- number of directions $N = 100$,
- perturbation radius $\delta =$ 1e-3,
- Gaussian u_i followed by ℓ_2 normalization,
- using the model's confidence score $p_y(x)$ in place of logits.

This yields a query-efficient estimate of the gradient norm that is sufficiently stable for MFD.

Effectiveness of black-box MFD. We evaluate the black-box variant of our MFD on CIFAR-10 with ResNet-18 under the same balanced evaluation protocol used in the main paper. While the AUC decreases from **0.9111** (white-box) to approximately **0.82** under finite-difference estimation, the detector remains highly effective and clearly separates fabricated from true members. This result demonstrates that our gradient-based detection principle generalizes beyond

the white-box setting. Overall, this section provides a simple but representative black-box instantiation, showing that **MFD remains effective even without white-box access**, thus broadening the real-world applicability of our detection framework.

E. Confidence-Matched Analysis of Gradient Norm

A natural question is whether the smaller input-gradient norms of fabricated members are simply a byproduct of their higher target-class confidence. To examine this, we perform a confidence-matched comparison between true and fabricated members.

Specifically, we group samples according to their target-class confidence $p_y(x)$ and compare the input-gradient norm $\|\nabla_x \ell(f(x), y)\|$ within each confidence bin. If the gradient-based signal used by **MFD** were merely reflecting confidence, then the difference between true and fabricated members should largely disappear after conditioning on $p_y(x)$.

$p_y(x)$ bin	member	fabricated
$[0.90, 0.95]$	1.09 ± 0.21	$\mathbf{0.73 \pm 0.18}$
$[0.95, 0.98]$	0.93 ± 0.17	$\mathbf{0.58 \pm 0.15}$
$[0.98, 0.99]$	0.82 ± 0.15	$\mathbf{0.49 \pm 0.14}$

Table 2. Confidence-matched comparison of input-gradient norms. Fabricated samples consistently exhibit smaller gradient norms than true members within the same target-class confidence range.

As shown in Table 2, fabricated samples remain consistently lower in gradient norm than true members across all confidence bins. This indicates that the gradient-norm gap is not solely explained by confidence, but also reflects the geometric effect induced by the fabrication process itself. This observation supports the use of $\|\nabla_x \ell(f(x), y)\|$ as a detection statistic in **MFD**.

F. Adaptive MFA

Adaptive fabrication via gradient-penalized optimization. A natural question is whether a Fabricator aware of **MFD** can jointly maximize fabrication success while suppressing the gradient-norm signal used for detection. To examine this, we consider an adaptive variant of MFA that augments the original objective with a penalty on the input-gradient magnitude:

$$\max_{\|\delta\|_\infty \le \epsilon} \Big(p_y(x+\delta) - \lambda_{\text{adv}} \cdot \|\nabla_x \ell(f(x+\delta), y)\| \Big), \quad (25)$$

where $\lambda_{\text{adv}} \in \{0.05, 0.1, 0.5\}$ controls the strength of the penalty. The rest of the settings follow the main experimental configuration: CIFAR-10, ResNet-18, LiRA as the reference MIA, and the same ϵ and optimization budget used in §3.1. This adaptive objective corresponds to a Fabricator attempting to counteract the "gradient-norm collapse" phenomenon formalized in Theorem 1.

Experimental behavior and the trade-off. We first validate the baseline (non-adaptive) MFA on CIFAR-10. Consistent with the results reported in the main paper, applying MFA increases LiRA's *Error Area* from **0.2814** to **0.3523**, and raises the *Equal Error Rate* from **36.70%** to **41.85%**. Meanwhile, our **MFD** detects such fabricated members with a high *AUC* of **0.9111**. These numbers establish a strong starting point for examining whether adaptive fabrication can reduce detectability without sacrificing attack strength.

- We then introduce the gradient penalty. With a mild penalty ($\lambda_{\text{adv}} = 0.05$), we observe a marginal reduction in the MFD signal: the median gradient norm of fabricated samples increases slightly (by roughly 8%), reducing the MFD AUC from **0.9111** to **0.8746**. However, fabrication effectiveness also decreases: LiRA's Error Area only rises to **0.3291** instead of **0.3523**, and the EER drops from **41.85%** to **39.20%**. This reflects a direct manifestation of Theorem 1: suppressing gradient reduction restricts the optimizer's ability to move toward the high-confidence basin where MFA achieves maximal effect.
- Increasing the penalty to $\lambda_{\text{adv}} = 0.1$ produces a clearer shift. The gradient-norm suppression becomes stronger (roughly a 15% increase in median gradient magnitude), and MFD AUC drops further to **0.8327**. Yet fabrication deteriorates significantly: LiRA's Error Area now only reaches **0.3048**, nearly erasing the gains achieved by non-adaptive MFA, and the EER falls to **37.05%**, approaching the original no-attack baseline. This empirically confirms an intrinsic limitation: evading gradient-norm detection forces the optimization to remain in regions of weaker confidence ascent.
- When the penalty is further strengthened to $\lambda_{\text{adv}} = 0.5$, the Fabricator becomes dominated by the constraint. The gradient norm becomes visually indistinguishable from true members in most cases, reducing MFD AUC to **0.5914**. However, fabrication almost entirely collapses: LiRA's Error Area reaches only **0.2877** (barely above the unperturbed value of **0.2814**), and the EER drops to **36.82%**, effectively neutralizing the attack. At this point, the Fabricator is unable to fool the Inferer while simultaneously masking the gradient-norm cue.

Interpretation through gradient-norm collapse. These observations align with the geometry predicted by Theorem 1. The theorem states that, under small ϵ, a signed gradient step inevitably moves the input toward regions of lower input-gradient magnitude. Fabrication *requires* traversing this trajectory into increasingly sharp, high-confidence basins where the loss landscape flattens, naturally shrinking the gradient norm. Adaptive attacks attempting to counter-

act this process must keep gradients artificially large, which fundamentally conflicts with the conditions needed to maximize membership likelihood. Thus, adaptive MFA faces a structural trade-off: *reducing detectability directly undermines the optimization path that produces strong fabrication; strengthening fabrication unavoidably triggers gradient-norm collapse, making detection easier.*

Conclusion of adaptive analysis. Across all settings, we find that while adaptive fabrication can moderately reduce the MFD signal at small λ_{adv}, the cost is a proportional and sometimes severe degradation of fabrication effectiveness. Stronger penalties do suppress gradient signatures but simultaneously collapse the attack. This persistent and quantifiable trade-off confirms that **MFD** remains robust even against adaptive MFA, and that gradient-geometry signals form a structurally unavoidable barrier for adversarial manipulation.

G. Behavior Under MI Defenses

MI defenses are designed to weaken the power of MIAs; from the perspective of adversarial membership manipulation, this actually creates a *more permissive* environment for our framework: once the baseline auditor is weaker, it becomes easier to further degrade its reliability via **MFA**, while the geometry-driven components (**MFD**, **AR-MIAs**) remain effective because they directly exploit gradient behavior rather than raw membership signals. In this section we therefore provide a concise analysis under standard MI defenses on CIFAR-10 with ResNet-18, keeping all other settings identical to the main experiments and focusing on LiRA as a representative strong MIA. Concretely, we consider (i) ℓ_1-regularized training with coefficient $\lambda_{\ell_1} = 10^{-4}$, (ii) knowledge distillation with temperature $T = 2$ and a student trained with a $0.7/0.3$ mix of soft and hard labels, and (iii) DP-SGD with clipping norm $C = 1.0$ and noise multiplier $\sigma = 1.0$.

- **MFA under MI defenses.** For LiRA without MI defenses, our **MFA** with perturbation budget $\|\delta\|_\infty \leq 4/255$ already induces a pronounced degradation, yielding an Error Area of 0.3523 and an Equal Error Rate (EER) of 41.85%. When we retrain the same model under ℓ_1-reg, distillation, or DP-SGD, the underlying LiRA auditor becomes less confident, and **MFA** becomes even more effective: across the three defenses, the Error Area increases into the 0.40–0.45 range and the EER rises to roughly 48%–52% (i.e., about $15\% \uparrow$–$25\% \uparrow$ relative to the undefended case). This confirms the intuition that MI defenses, while reducing raw MIA accuracy, make it easier for a Fabricator to push non-members into regions where the auditor is systematically misled.
- **MFD under MI defenses.** Our detector **MFD** relies on the *gradient-norm collapse* geometry rather than on any particular MIA score. On the undefended model, **MFD** achieves an AUC of 0.9111 against fabricated members from **MFA** ($\|\delta\|_\infty \leq 4/255$). Under ℓ_1-reg, distillation, and DP-SGD, we observe only mild fluctuations, with AUC values remaining in the high 0.88–0.91 range. In other words, although MI defenses significantly reduce the separability between true members and non-members for MIA itself, they do *not* disrupt the gradient-geometry signal that **MFD** exploits; fabricated members still concentrate in low-gradient basins and remain reliably detectable.
- **AR-MIAs under MI defenses.** For adversarially robust LiRA (**AR-MIA** with the weighting in Eq. (10)), we focus on the LiRA series and set $\lambda = 10$. Under standard training, the base LiRA achieves an AUC of 0.6832, while our adversarially robust variant improves this to 0.7937, i.e., an absolute gain of about 0.11 (approximately $15\% \uparrow$ in relative terms). When MI defenses are enabled, LiRA's AUC drops substantially into the 0.55–0.58 band, reflecting the intended regularization effect on memorization; nevertheless, the corresponding adversarially robust LiRA still achieves AUCs around 0.65–0.69, preserving a similar absolute improvement (again on the order of 0.10–0.11). Thus, even in the presence of MI defenses that inevitably weaken membership signals on true members, **AR-MIAs** continue to provide consistent gains over their non-robust counterparts and retain strong discriminative power against fabricated members.

H. Discussion of Limitations on MFA

Member Fabrication Attack, which introduces imperceptible adversarial perturbations to the data, encounters limitations in its application and practicality.

I. Practical Use Case Requirements. The primary limitation of Member Fabrication Attack is its dependency on specific use-case scenarios, such as those where adding noise to data before an inferer's access is feasible. In scenarios where the inferer can directly obtain the original data and conduct membership inference, Member Fabrication Attack cannot be effectively deployed.

II. Limited Application Beyond Image Datasets. The second limitation concerns its restricted applicability beyond image datasets. The specific characteristics of image data allow for the effective implementation of subtle perturbations without compromising data integrity. Extending this method to non-image datasets, such as text or tabular data, presents significant challenges. The concept of 'imperceptible' changes in these datasets demands a distinct approach. Adapting our methodology to accommodate these varied formats is a primary focus of our future work.

I. T-SNE Visualization of Semantic Features

The distribution of fabricated and true members in various semantic feature spaces is visualized using t-SNE [38]. The first row of Figure 8 presents the semantic features from the penultimate layer, with perturbations constrained within the range ($\|\delta\|_\infty \leq 2.0/255$) to ($\|\delta\|_\infty \leq 6.0/255$) from left to right. The second row shows the semantic features at the antepenultimate layer, with the same perturbation range. In these visualizations, red dots represent the true members' semantic features, while blue dots represent the fabricated members. The t-SNE visualization reveals a significant overlap between the distributions of fabricated and true members in the feature space, suggesting that semantic features alone cannot effectively distinguish between the two. This observation underscores the limitations of traditional methods based solely on semantic features for the detection task,

J. The Visualization of Fabricated Members

In Figures 9 to 11, we demonstrate the visual quality of images after adding member fabrication perturbations. The perturbations are imperceptible adversarial changes applied to **ImageNet-100**. For each pair of images, the top image represents the original non-member, while the bottom image shows the corresponding perturbed fabricated member. We used $\epsilon = 2/255$ for $\mathcal{B}_\epsilon[x]$ in these examples. The perturbations are extremely subtle and nearly imperceptible to the human eye, illustrating that the Member Fabrication attack can be effective even with the addition of very small perturbations.

K. Evaluation Metrics

K.1. Evaluation Metrics of MIAs

Several strategies have been proposed for evaluating the effectiveness of MIAs. In this section, we review two widely adopted evaluation metrics used to assess MIA performance in detail.

I. Receiver Operating Characteristic Curve (ROC Curve). To evaluate MIA performance comprehensively, it is essential to consider metrics that reflect an inferer's ability to make accurate predictions while minimizing false positives. The most commonly used approach is to examine the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR). An effective attack should maximize the TPR while minimizing the FPR. This trade-off is captured through the Receiver Operating Characteristic (ROC) curve, which plots TPR against FPR at various decision thresholds. Many prior studies report ROC curves and summarize their performance using the **Area Under the Curve (AUC)** [32, 40, 43, 51, 61, 64]. The **AUC** offers a single score that summarizes performance across all thresholds.

II. True Positive Rate at Low False Positive Rates (TPR @ FPR). While the ROC curve and **AUC** are useful, Carlini et al. [7] argue that these metrics may not fully capture the security risks associated with MIAs. In real-world scenarios, privacy breaches are often more consequential when they occur with a minimal number of false positives. Hence, they suggest that MIAs should be evaluated by focusing on TPR at low FPR values, as this regime better represents practical risk scenarios. This evaluation criterion provides a more robust measure of attack effectiveness, particularly in contexts where privacy violations must be detected with minimal false positives.

For a thorough assessment of performance, we adopt two primary evaluation metrics: *Area Under the ROC Curve (**AUC**)* and *True Positive Rate at Low False Positive Rates (TPR @ FPR)*. To compute TPR values at specific FPR thresholds, we utilized numpy's interpolation methods, as the range of FPR values is often non-continuous in certain experimental settings. Our proposed adversarially robust MIAs also use these evaluation metrics.

K.2. Evaluation Metrics of MFA

We propose to assess the performance of **MFA** by evaluating the predictive performance of Inferer after they have been exposed to fabricated members. In practice, we found the conventional ROC curve, based on the standard False Positive Rate (FPR) and True Positive Rate (TPR), does not clearly distinguish between the different **MFA** methods. To address this limitation, we propose using the TNR-TPR curve, combined with a logarithmic scale, which enhances the clarity of comparisons. Additionally, we employ two primary evaluation metrics: **Error Area** and **Equal Error Rate** (**EER**). The **Error Area** (i.e., 1 - **AUC**) is defined as the complement of the Area Under the Curve (**AUC**) of the TNR-TPR curve, where a higher value indicates better **MFA** performance. The **Equal Error Rate** (**EER**) is the point at which the False Positive Rate (FPR) equals the False Negative Rate (FNR), providing a balanced measure of a method's ability to correctly identify both true and fabricated members. In these evaluations, a better **MFA** should result in Inferer with lower TPR and TNR values, causing the curves to approach the lower-left corner of the TNR-TPR plot. Correspondingly, the **Error Area** should be larger, and the **Equal Error Rate** should also be higher.

K.3. Evaluation Metrics of MFD

We assess the performance of our Member Fabrication Detection using the Area Under the ROC Curve (**AUC**) as the primary evaluation metric. **AUC** provides a comprehensive measure of how well our detection method can distinguish fabricated members from true members across various thresholds. A higher **AUC** value indicates that our detection method is more effective at identifying fabricated members, with the optimal outcome being an **AUC** as close to 1 as

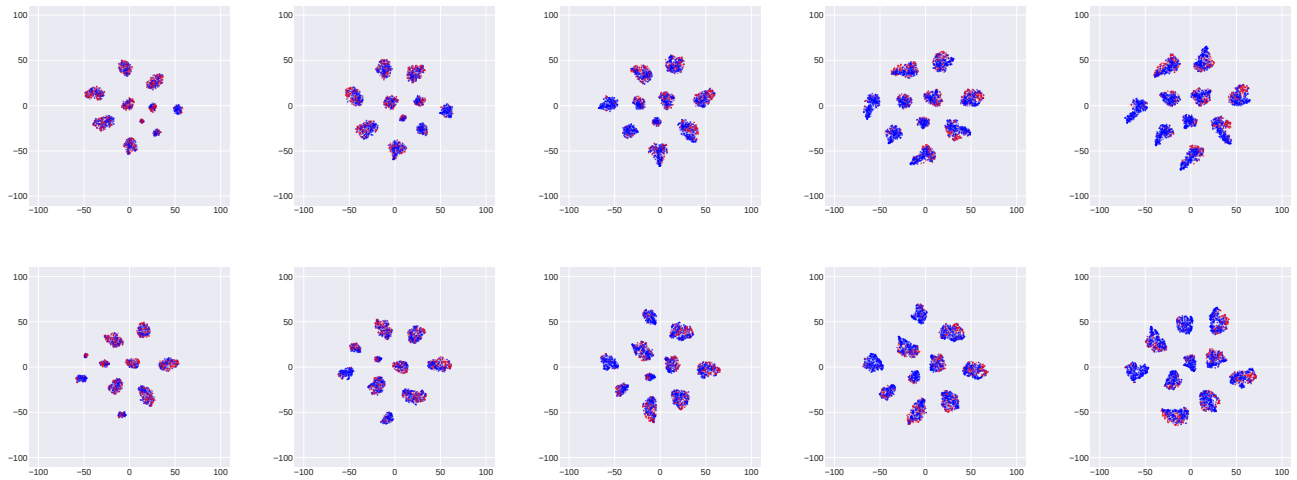


Figure 8. Visualization of the Distribution of Fabricated and True Members in Different Semantic Feature Spaces Using t-SNE [38]. The first row displays the semantic features at the penultimate layer, with perturbation constrained to ($\|\delta\|_\infty \leq 2.0/255$) to ($\|\delta\|_\infty \leq 6.0/255$) from left to right. The second row shows the semantic features at the antepenultimate layer, with the same range of perturbations. Red dots represent true members' semantic features, and blue dots represent fabricated members. We observe a high degree of overlap between the distributions of fabricated and true members, indicating that semantic features alone are insufficient to distinguish between them.

possible.

L. Experimental Details

L.1. Datasets

CIFAR-10 [29] is a widely-used dataset consisting of 60,000 color images, each with a resolution of $32 \times 32 \times 3$. The images are evenly distributed across 10 distinct classes, which represent various objects such as airplanes, automobiles, birds, and more.

CIFAR-100 [29] shares the same structure and resolution as **CIFAR-10**, with 60,000 color images of $32 \times 32 \times 3$. However, it poses a more complex classification challenge, as it is divided into 100 classes, each containing 600 images.

SVHN [46] is a real-world image dataset obtained from house numbers in Google Street View images. It consists of 99,289 color images in the training set and 26,032 color images in the test set, each with a resolution of $32 \times 32 \times 3$. The dataset is divided into 10 classes, representing the digits 0 through 9. **SVHN** is designed for digit recognition tasks in challenging, real-world scenarios with varying backgrounds, lighting conditions, and distortions.

CINIC-10 is an extension of the CIFAR dataset, designed to bridge the gap between **CIFAR-10** and more complex datasets like ImageNet. It contains 270,000 images with a resolution of $32 \times 32 \times 3$, with classes and data distribution similar to **CIFAR-10**. **CINIC-10** combines **CIFAR-10** images with a subset of ImageNet images, providing a larger and more diverse dataset.

ImageNet-100 is a subset of the larger ImageNet dataset, containing 100 classes from the original ImageNet hierarchy. The dataset is significantly more challenging due to its larger and more complex images, typically with a resolution of 224×224. The diversity of classes and image variations adds complexity to classification tasks.

L.2. Implementation details

Datasets, Model Structures and Membership Inference Attacks Used. Our experiments span multiple datasets: **CIFAR-10** [29], **CIFAR-100** [29], **SVHN** [29], **CINIC-10** [14], and **ImageNet-100** [15]. We experiment with the first four datasets using the ResNet-18 architecture [23], and evaluate the larger **ImageNet-100** dataset with the Wide-ResNet-50-2 architecture [66]. We consider the following MIAs in this paper: the basic loss attack [65], and strong existing baselines, including **Attack R** [64], **LiRA** [7], and **RMIA** [67].

Baselines Used for Comparison with MFA. We invert the perturbation direction in traditional adversarial attacks as baselines for comparison with our **MFA**: *(i) Inverted FGSM* (I-FGSM) [20], *(ii) Inverted BIM* (I-BIM) [30], *(iii) Inverted PGD* (I-PGD) [39], *(iv) Inverted Carlini and Wagner Attack* (I-CW) [5], and *(v) Inverted APGD* (I-APGD) [13]. We exclude the full versions of AutoAttack and MM Attack [19], as they rely on adaptive decision-making methods specifically designed for traditional adversarial attacks and cannot generalize to our scenario.

Model Training Details. Consistent random seeds and training settings are maintained across all experiments. For datasets **CIFAR-10** [29], **CIFAR-100** [29], **SVHN** [29] and

CINIC-10 [14], we train the target model or shadow model using 20,000 samples to ensure consistency in data size. For **ImageNet-100** [15], we train the target model using all samples in the original datasets. For testing MIAs, we select a testing set containing 2,000 member samples and 2,000 non-member samples. The models are trained using the Stochastic Gradient Descent (SGD) optimizer with momentum set to 0.9, a weight decay of 10^{-4}, and a batch size of 128. The learning rate is initialized to $\tau = 0.1$ and follows a cosine annealing schedule, gradually decaying to zero over 100 epochs.

Shadow Models. For **Attack R** [64], we train 100 reference models (OUT-Models). For **LiRA** [7] and **RMIA** [67], we train 100 IN-Models and 100 OUT-Models for modeling $\tilde{\mathbb{Q}}_{\text{in/out}}$, and ensure that the same shadow models are used across different methods. Due to the high training cost of shadow models for **ImageNet-100** with the corresponding Wide-ResNet-50-2 architecture, we only conduct basic loss attacks on it. Note that the training samples for shadow models or for modeling $\tilde{\mathbb{Q}}_{\text{m/nm}}$ are disjoint from the testing data.

Data Augmentation. In training models on these datasets, common data augmentation techniques were applied to improve the generalization and robustness of the models. For the first four datasets (**CIFAR-10**, **CIFAR-100**, **SVHN**, and **CINIC-10**), typical augmentation strategies such as random horizontal flipping and random cropping (with padding) were used. For **ImageNet-100**, we employed more advanced augmentation methods, such as random resized cropping and random horizontal flipping, tailored to handle larger image resolutions. These augmentation methods are standard practices for enhancing model performance across various image classification tasks.

Details of Fabricated Member Generation. For generating the fabricated members, we set the L_∞-norm bounded perturbation $\epsilon = [1/255, 8/255]$ for datasets **CIFAR-10** [29], **CIFAR-100** [29], **SVHN** [29] and **CINIC-10** [14], and set the L_∞-norm bounded perturbation $\epsilon = [0.5/255, 2/255]$ for **ImageNet-100** [15]; the maximum number of steps is $K = 100$; initial step size $\alpha = \epsilon/4$, momentum factor $\beta = 0.75$, decay factor $\gamma = 0.9$.

L.3. Required Resources

We implement all methods on Python 3.7.3 (Pytorch 1.13.1+cu117) with four NVIDIA RTX A5000 GPUs and an x86-64 CPU with 32 physical cores.

M. Supplementary Experimental Results

In this section, we present comprehensive supplementary experimental results detailing the performance of our method alongside various baselines. The results thoroughly demonstrate the effectiveness of our approaches.

Experimental Results of MFA. In Figure 12, we show that our **MFA** can effectively deceive different MIAs. Four subfigures depict four representative MIAs: loss attack, Attack R, LiRA, and RMIA. In each subfigure, we present four datasets: **CIFAR-10** [29], **CIFAR-100** [29], **SVHN** [29], and **CINIC-10** [14]. We observe that our Member Fabrication Attack (**MFA**) leads these MIAs to exhibit a low TPR, low TNR, high Error Area, and high Equal Error Rate (EER) in the TNR-TPR curve. The TNR-TPR curve is close to the bottom left. In Figures 13 to 17, we compare our **MFA** against the loss attack, as well as five adapted adversarial attacks across different perturbation levels and datasets: *(i)* Inverted Fast Gradient Sign Method (I-FGSM) [20], *(ii)* Inverted Basic Iterative Method (I-BIM) [30], *(iii)* Inverted Projected Gradient Descent (I-PGD) [39], *(iv)* Inverted Carlini and Wagner Attack (I-CW) [5], and *(v)* Inverted Adaptive PGD (I-APGD) [13]. The results show that our methods perform the best among these techniques in the above metrics. The quantitative comparison results can be found in Tables 3 to 6, where the best results are highlighted in red.

Experimental Results of MFD. In Figures 18 to 22, we show that our **MFD** can effectively distinguish true members from fabricated members across different perturbation levels and datasets. The AUC is greater than 50% and increases as the perturbation level increases.

Experimental Results of AR-MIAs. In Figures 23 to 26, we show that our **AR-MIAs** can be effectively combined with three existing strong MIAs—Attack R, LiRA, and RMIA—and significantly improve the baseline AUC value and TPR @ low FPR across different datasets. The quantitative comparison results can be found in Tables 7 to 9, where the best results are highlighted in red.

Figure 9. Imperceptible Adversarial Perturbations on **ImageNet-100**. For each pair, the top image is the original non-member, and the bottom image is the corresponding perturbed fabricated member, demonstrating that the perturbations are imperceptible to the human eyes. We used $\epsilon = 2/255$ for $\mathcal{B}_\epsilon[x]$ here.

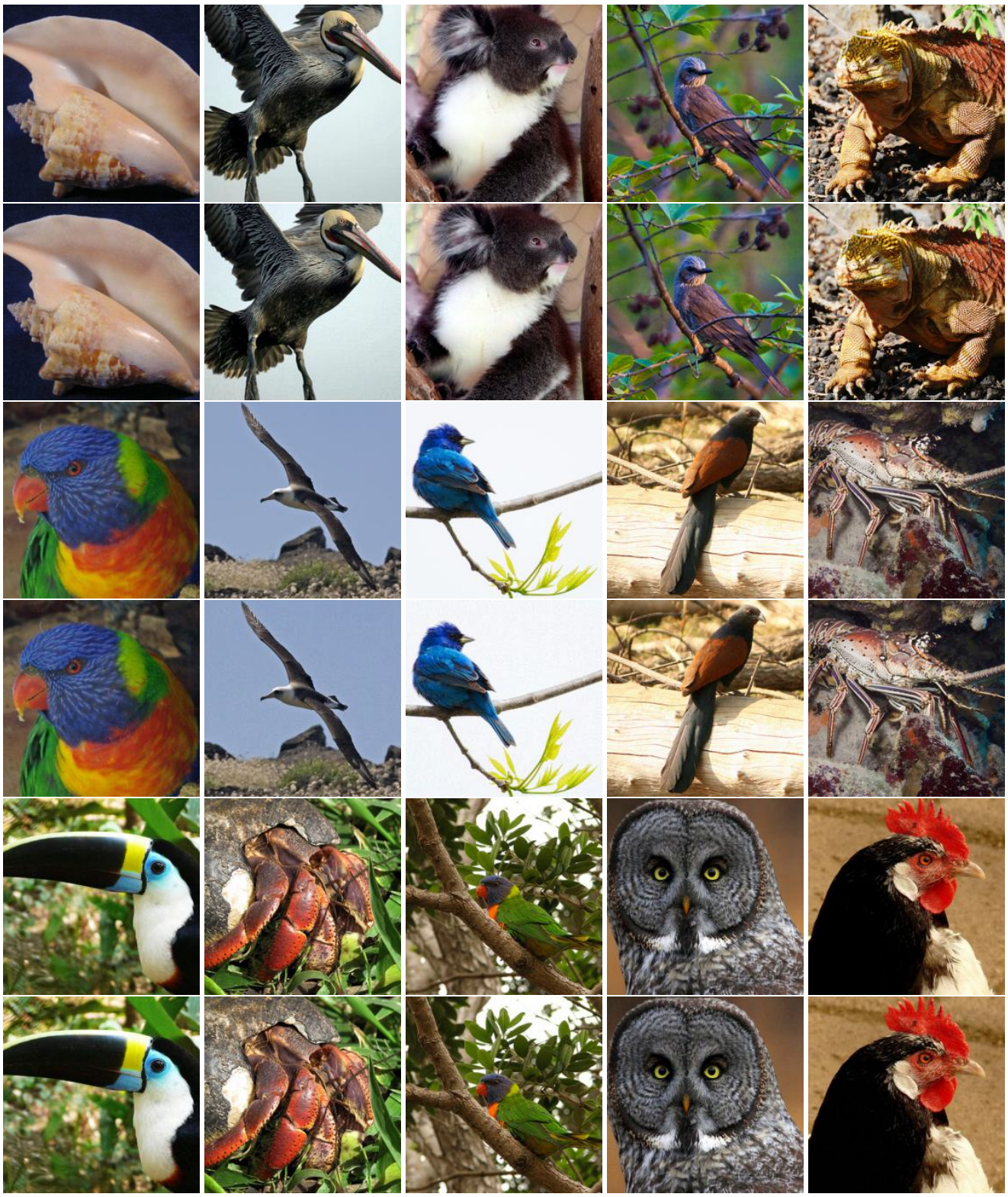
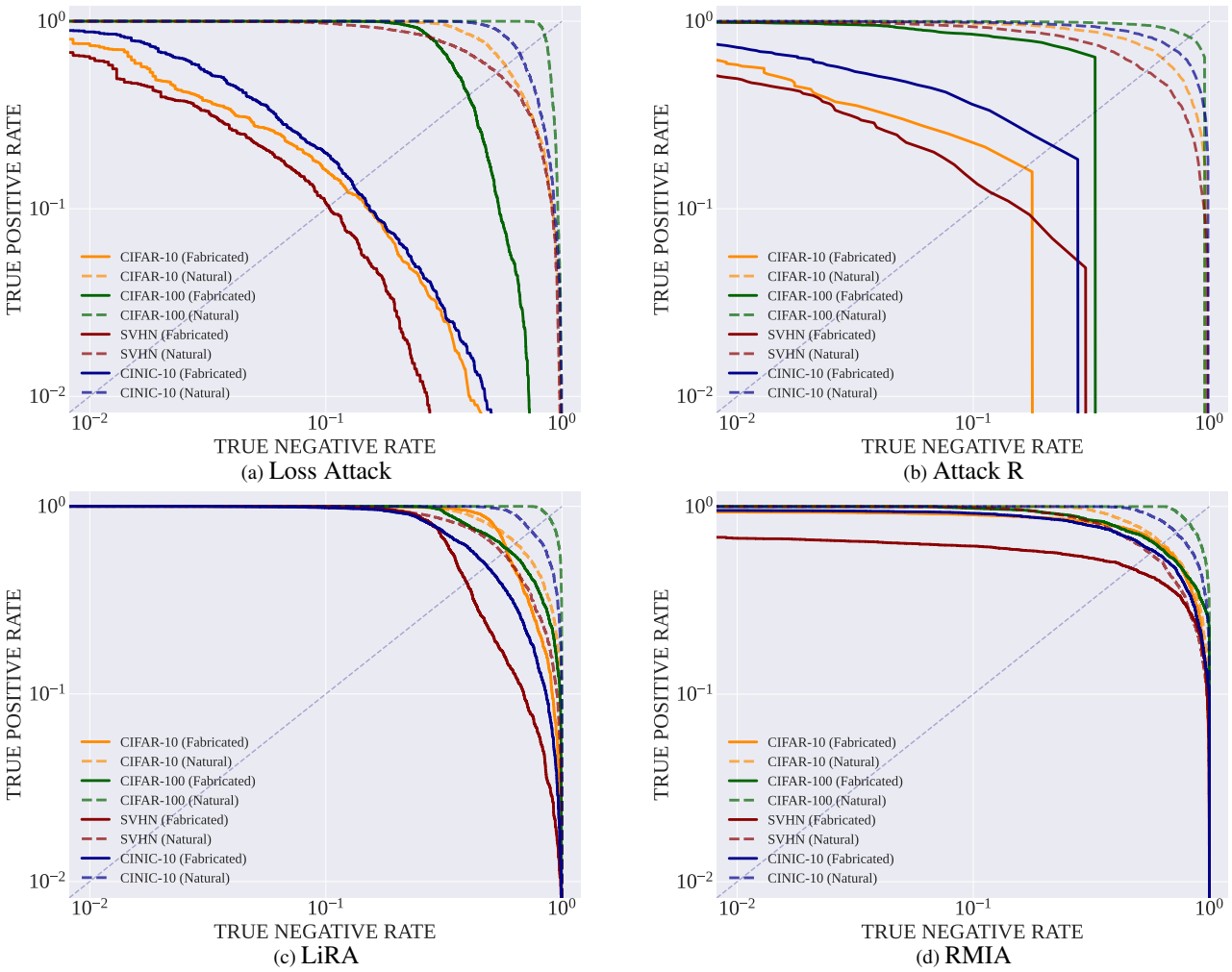


Figure 10. Imperceptible Adversarial Perturbations on **ImageNet-100**. For each pair, the top image is the original non-member, and the bottom image is the corresponding perturbed fabricated member, demonstrating that the perturbations are imperceptible to the human eyes. We used $\epsilon = 2/255$ for $\mathcal{B}_\epsilon[x]$ here.

Figure 11. Imperceptible Adversarial Perturbations on **ImageNet-100**. For each pair, the top image is the original non-member, and the bottom image is the corresponding perturbed fabricated member, demonstrating that the perturbations are imperceptible to the human eyes. We used $\epsilon = 2/255$ for $\mathcal{B}_\epsilon[x]$ here.



(a) Loss Attack (b) Attack R

(c) LiRA (d) RMIA

Figure 12. Comparison of the **Error Area** Between Our Member Fabrication Attack and Baselines Across Diverse MIAs ($\|\delta\|_\infty \leq 4.0/255$).

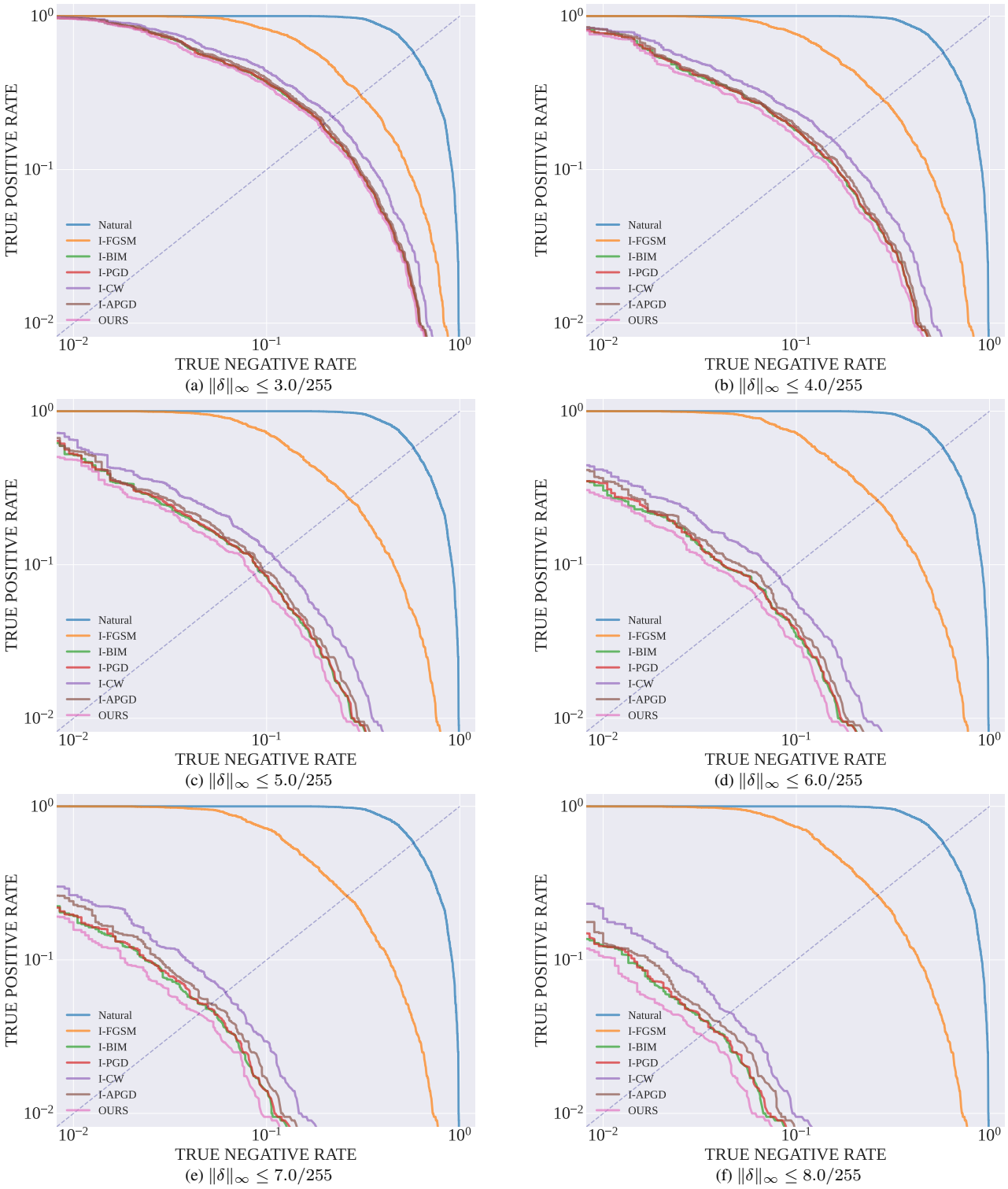


Figure 13. Comparison of the **Error Area** Between Our Member Fabrication Attack and Baselines Across Diverse Perturbation Bounds on **CIFAR-10**.

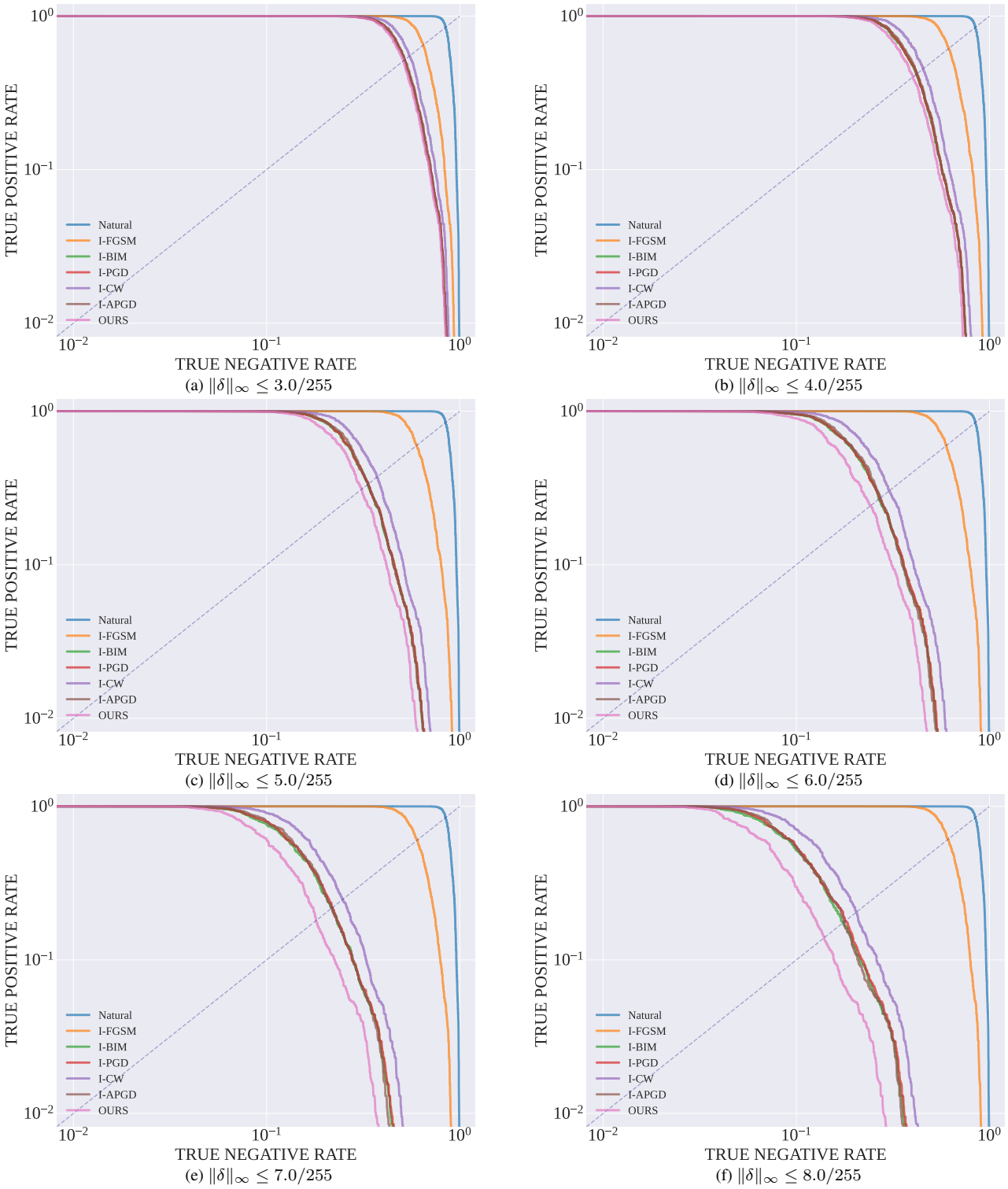


Figure 14. Comparison of the **Error Area** Between Our Member Fabrication Attack and Baselines Across Diverse Perturbation Bounds on **CIFAR-100**.

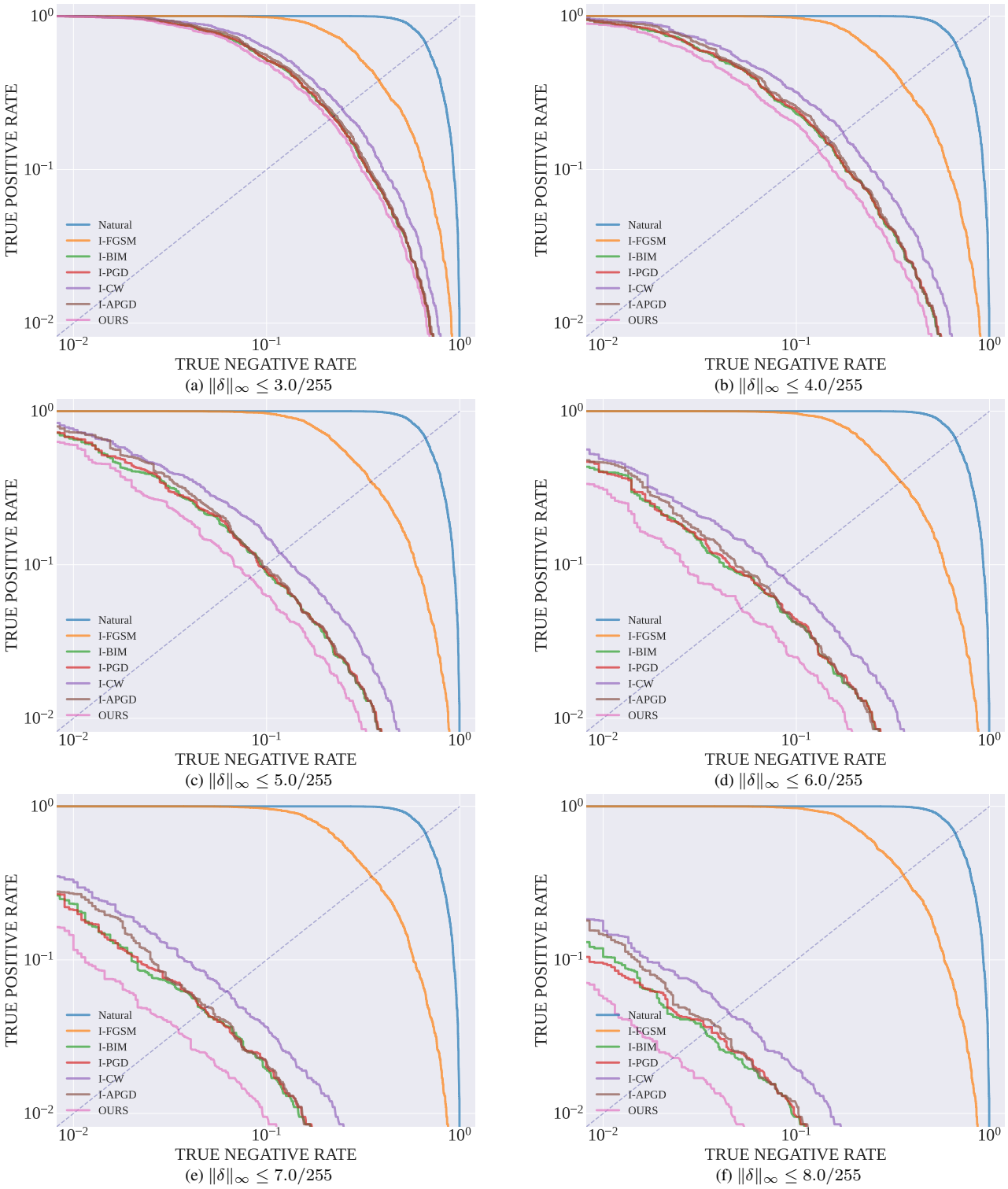


Figure 15. Comparison of the **Error Area** Between Our Member Fabrication Attack and Baselines Across Diverse Perturbation Bounds on **CINIC-10**.

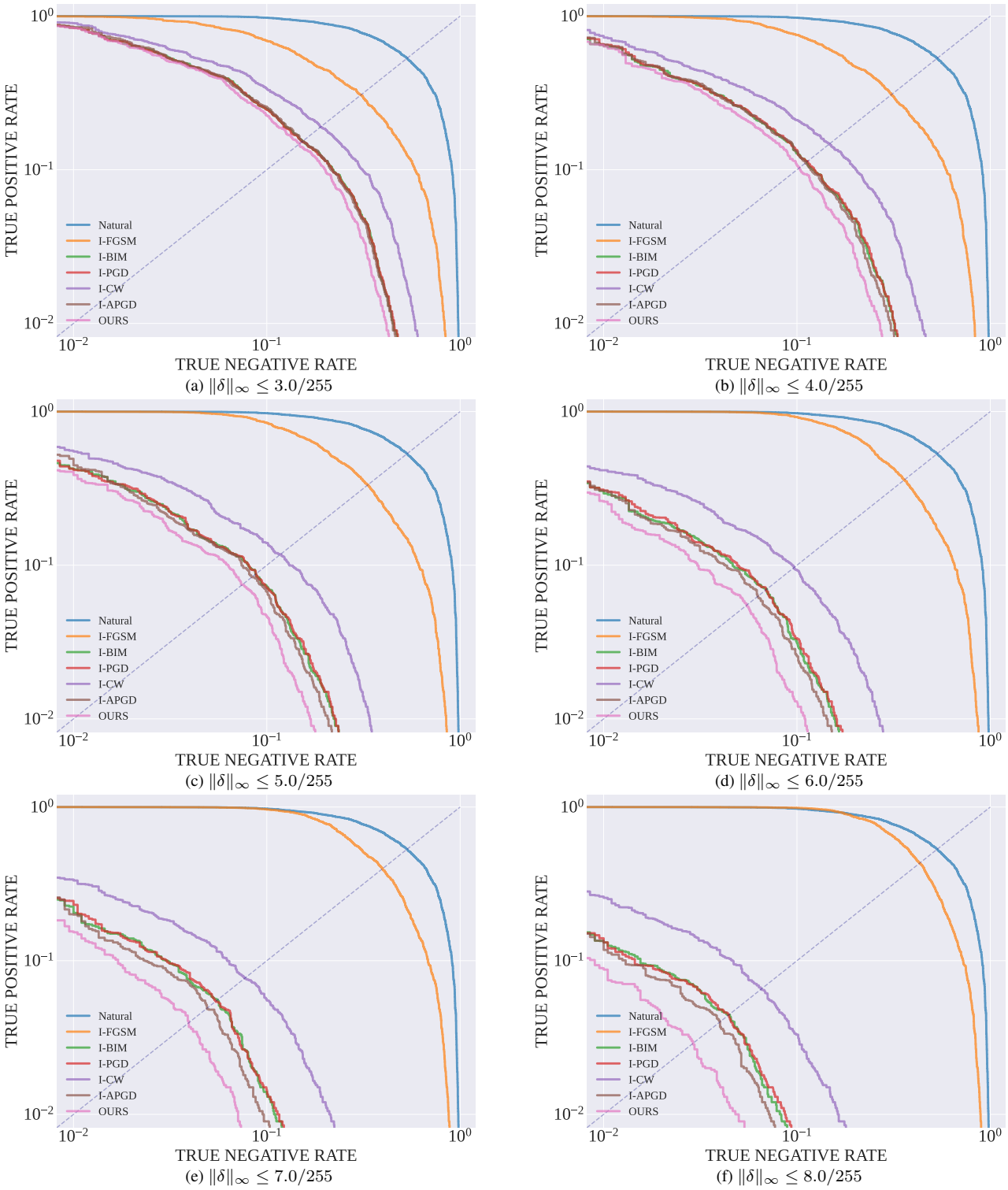


Figure 16. Comparison of the **Error Area** Between Our Member Fabrication Attack and Baselines Across Diverse Perturbation Bounds on **SVHN**.

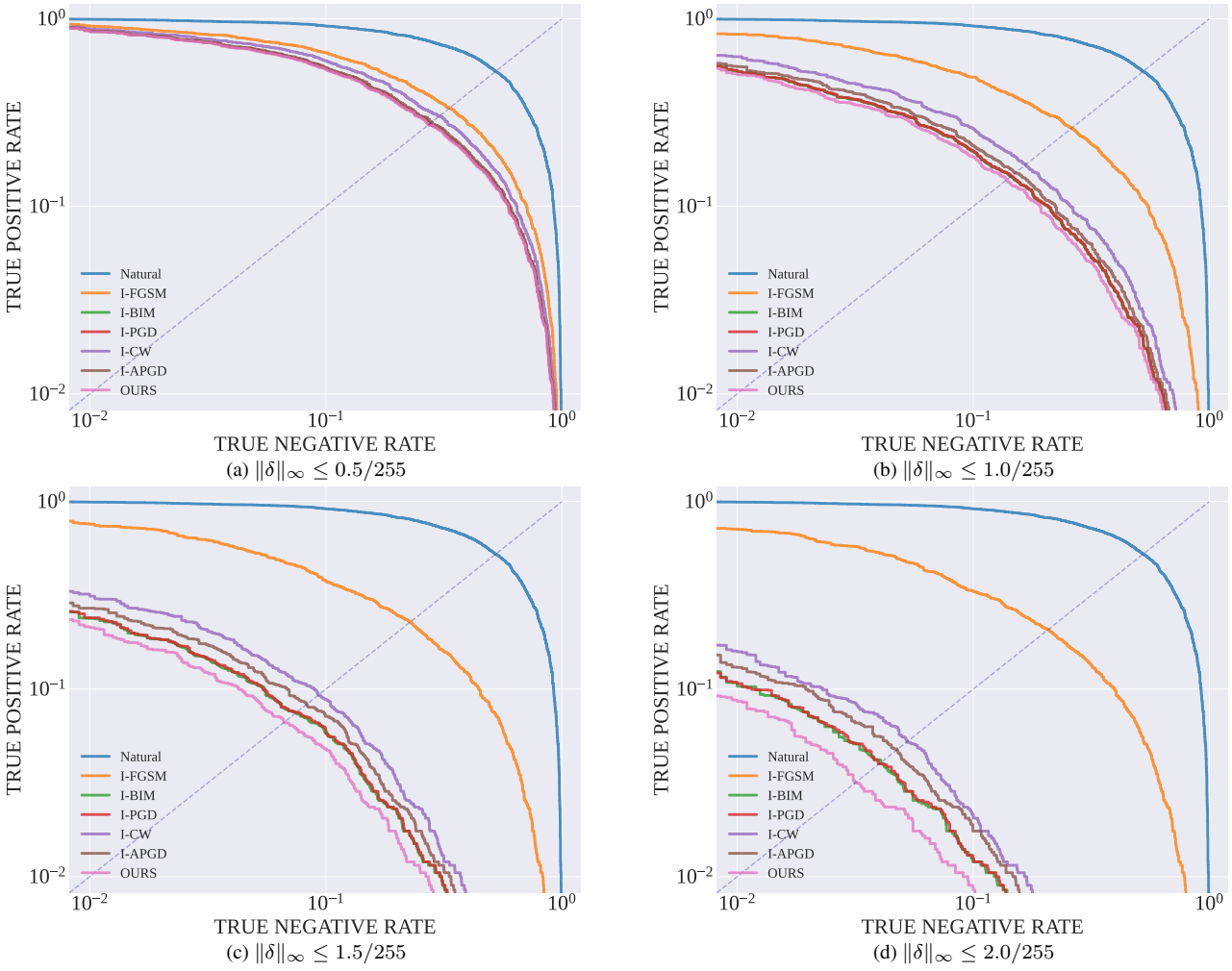
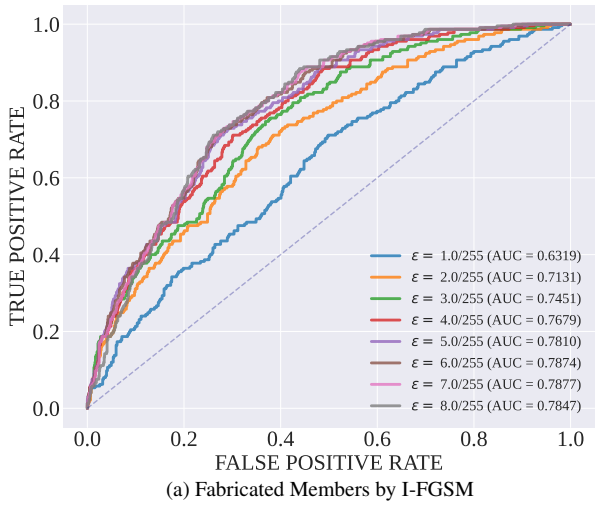
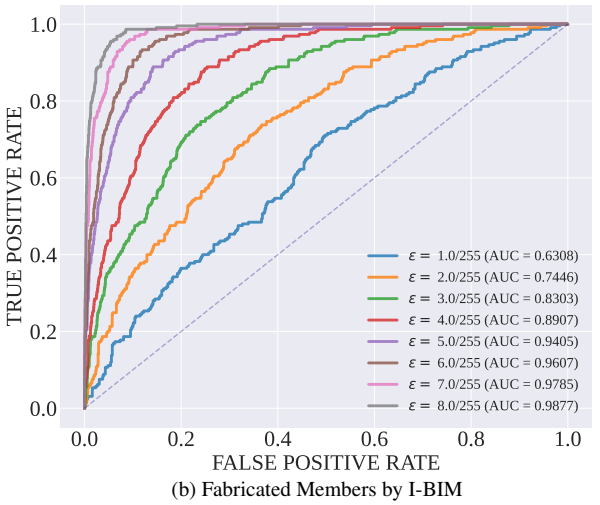


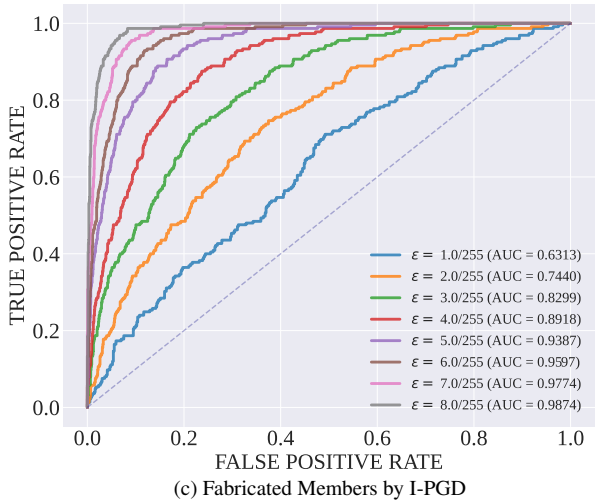
Figure 17. Comparison of the **Error Area** Between Our Member Fabrication Attack and Baselines Across Diverse Perturbation Bounds on **ImageNet-100**.



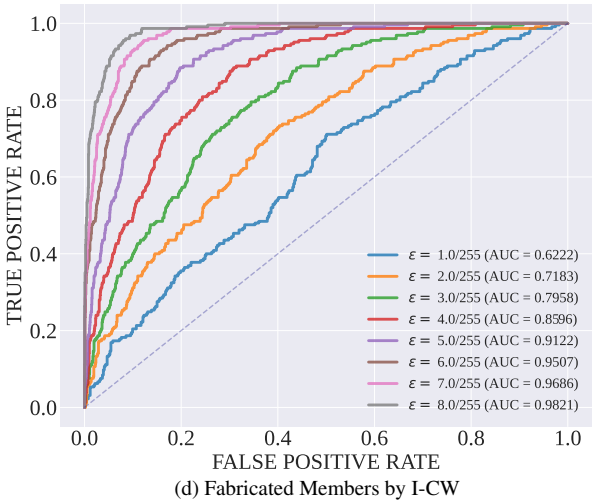
(a) Fabricated Members by I-FGSM



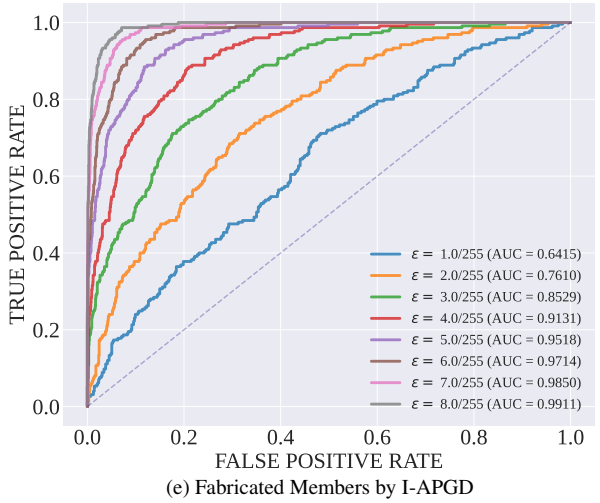
(b) Fabricated Members by I-BIM



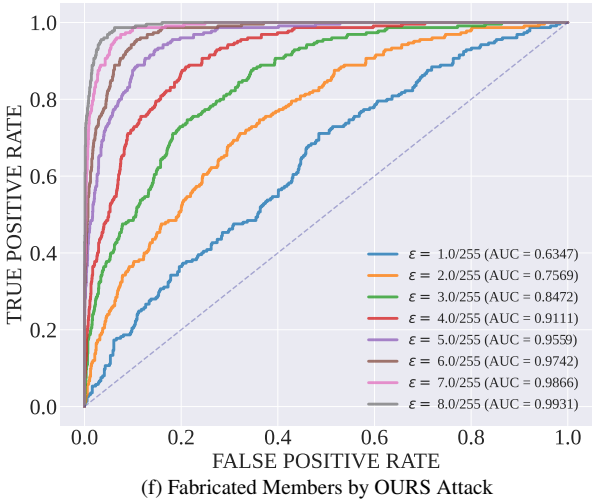
(c) Fabricated Members by I-PGD



(d) Fabricated Members by I-CW

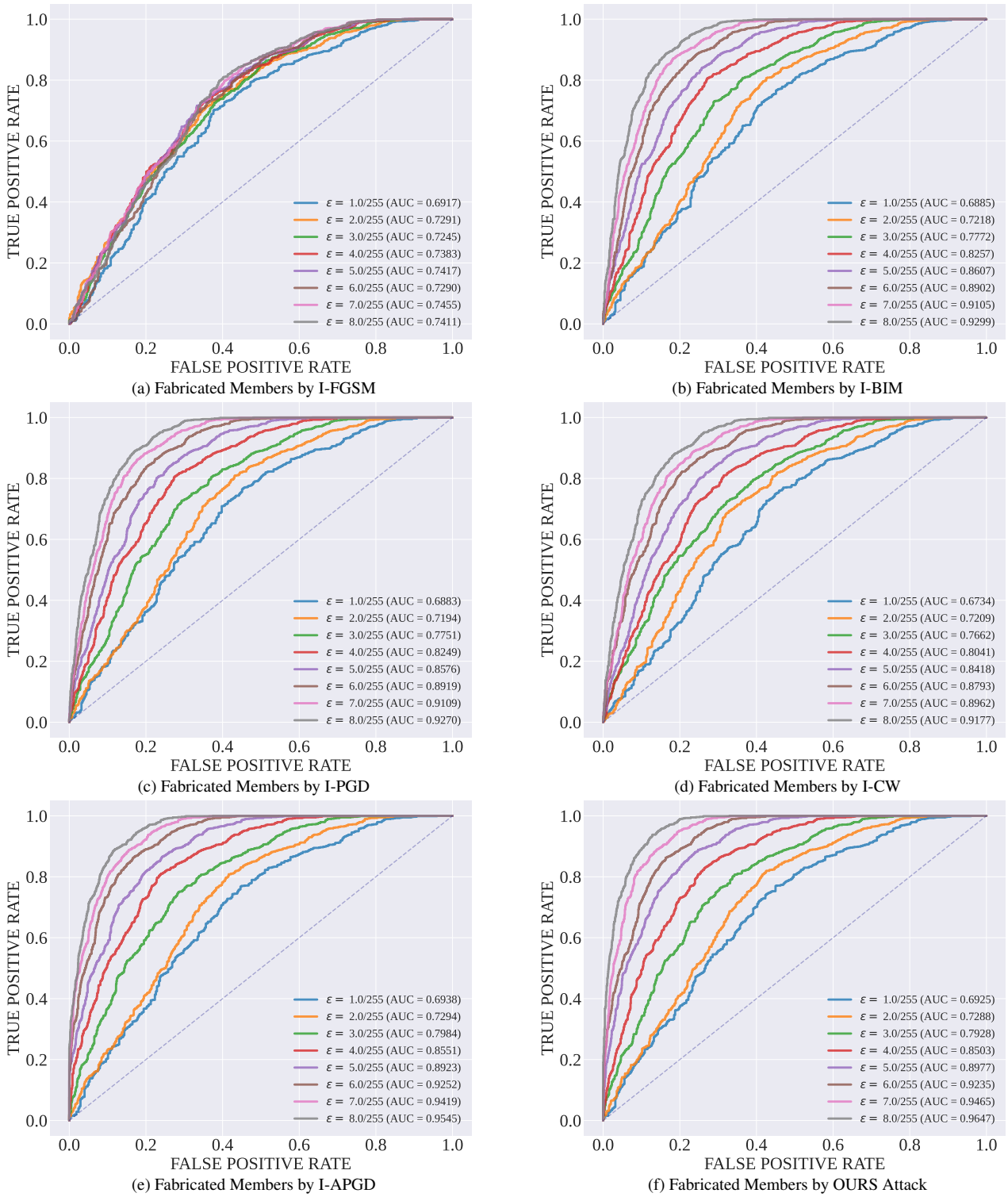


(e) Fabricated Members by I-APGD



(f) Fabricated Members by OURS Attack

Figure 18. Comparison of the ROC Curve for Our Member Fabrication Detection Across Diverse Perturbation Bounds on **CIFAR-10**.



(a) Fabricated Members by I-FGSM

(b) Fabricated Members by I-BIM

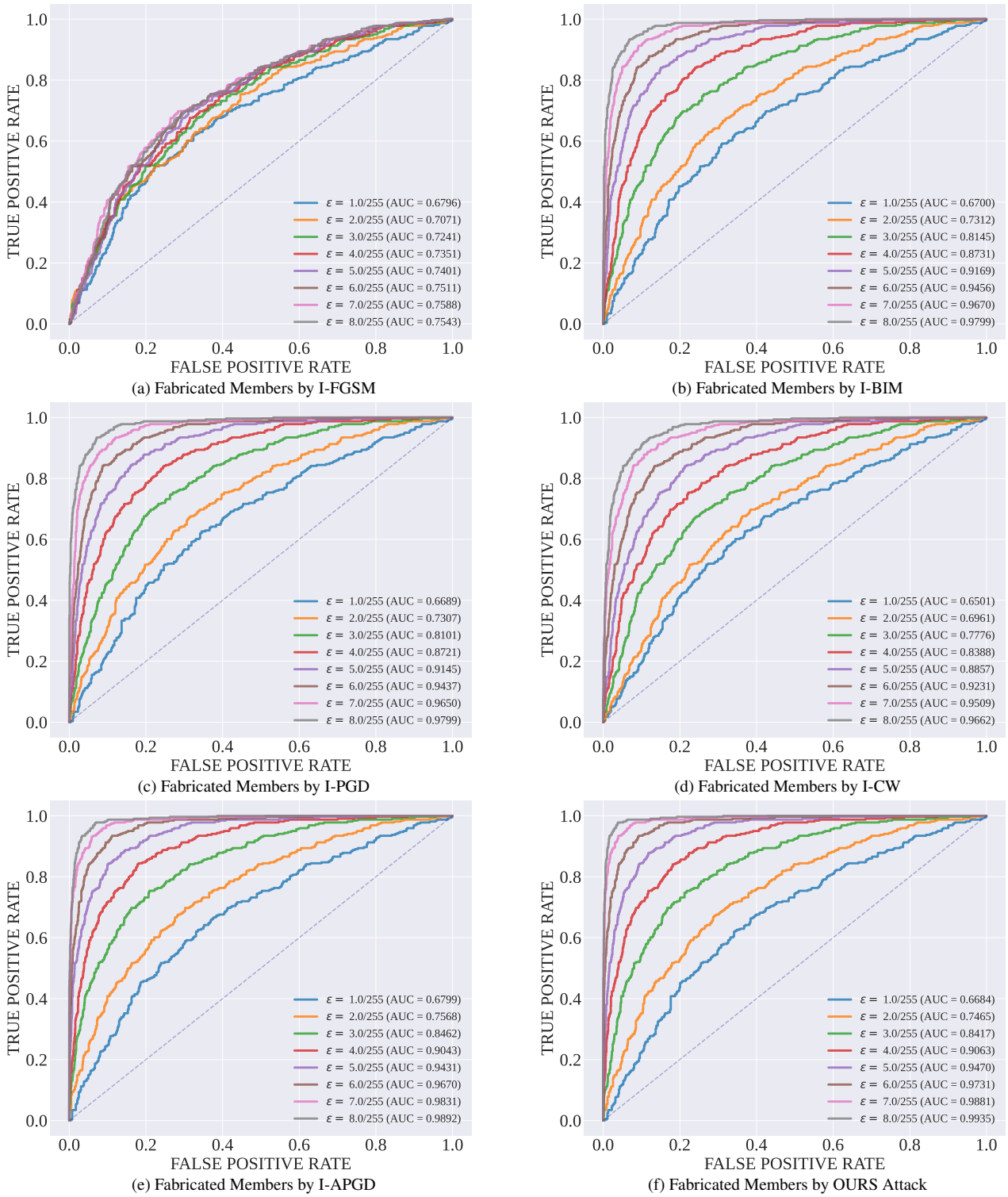
(c) Fabricated Members by I-PGD

(d) Fabricated Members by I-CW

(e) Fabricated Members by I-APGD

(f) Fabricated Members by OURS Attack

Figure 19. Comparison of the ROC Curve for Our Member Fabrication Detection Across Diverse Perturbation Bounds on **CIFAR-100**.



(a) Fabricated Members by I-FGSM

(b) Fabricated Members by I-BIM

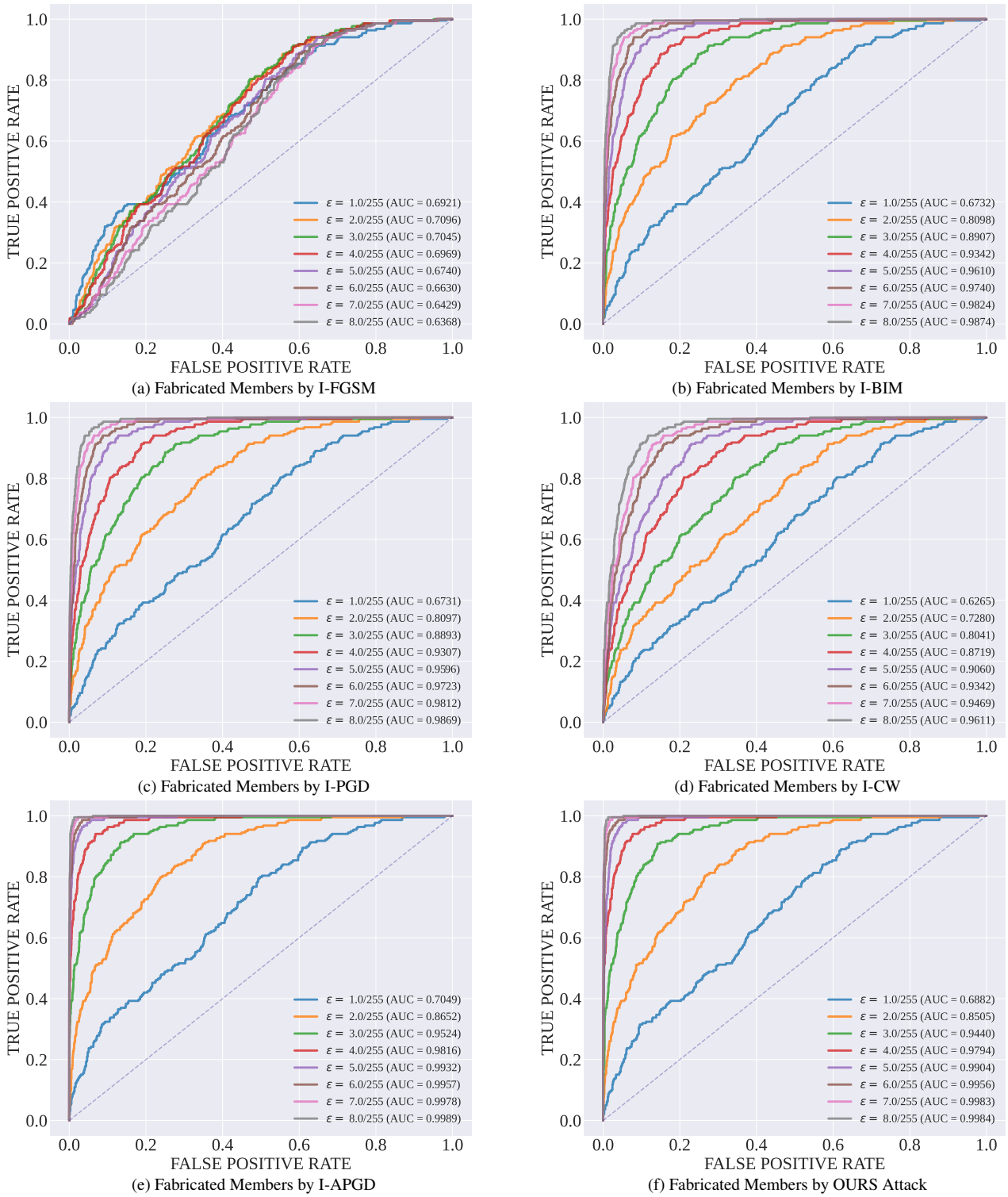
(c) Fabricated Members by I-PGD

(d) Fabricated Members by I-CW

(e) Fabricated Members by I-APGD

(f) Fabricated Members by OURS Attack

Figure 20. Comparison of the ROC Curve for Our Member Fabrication Detection Across Diverse Perturbation Bounds on **CINIC-10**.



(a) Fabricated Members by I-FGSM

(b) Fabricated Members by I-BIM

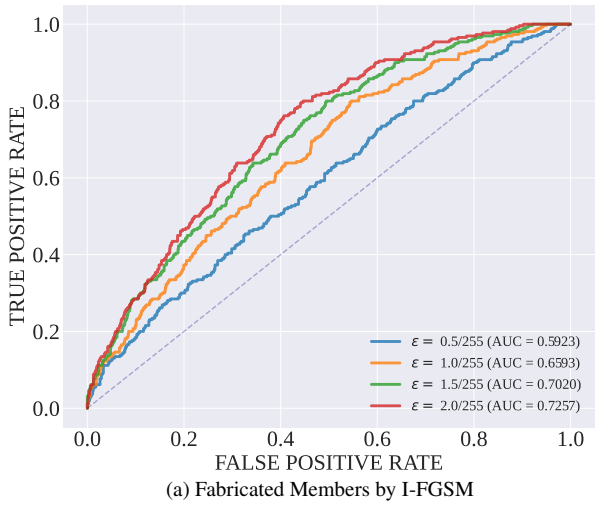
(c) Fabricated Members by I-PGD

(d) Fabricated Members by I-CW

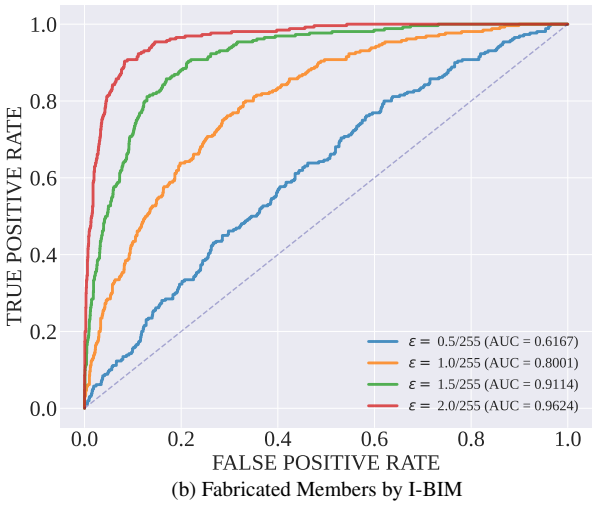
(e) Fabricated Members by I-APGD

(f) Fabricated Members by OURS Attack

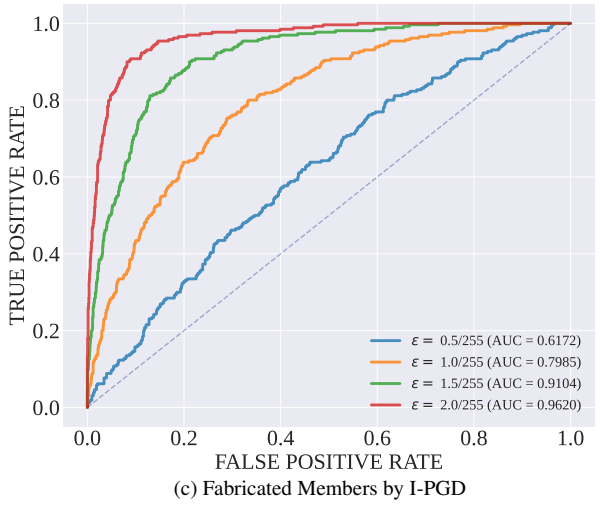
Figure 21. Comparison of the ROC Curve for Our Member Fabrication Detection Across Diverse Perturbation Bounds on **SVHN**.



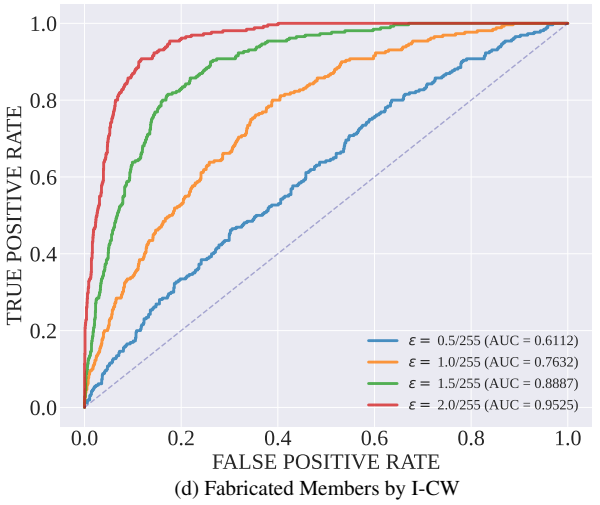
(a) Fabricated Members by I-FGSM



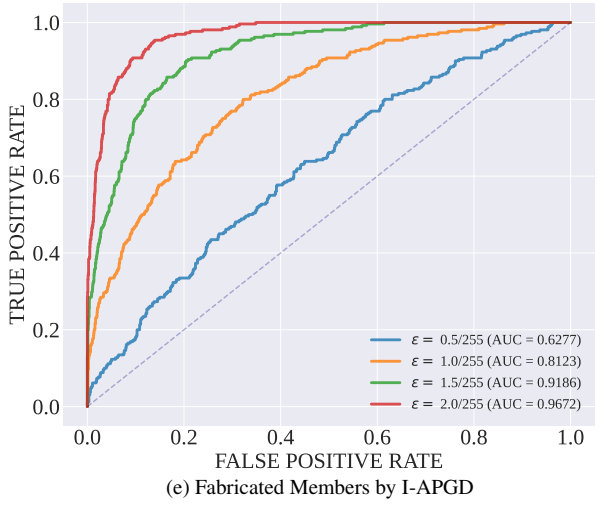
(b) Fabricated Members by I-BIM



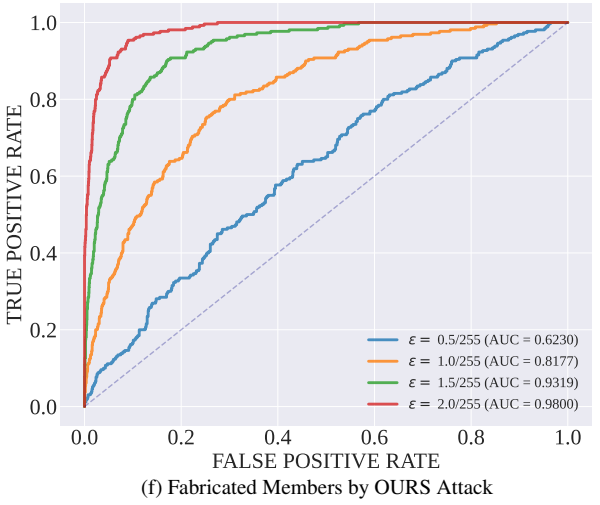
(c) Fabricated Members by I-PGD



(d) Fabricated Members by I-CW

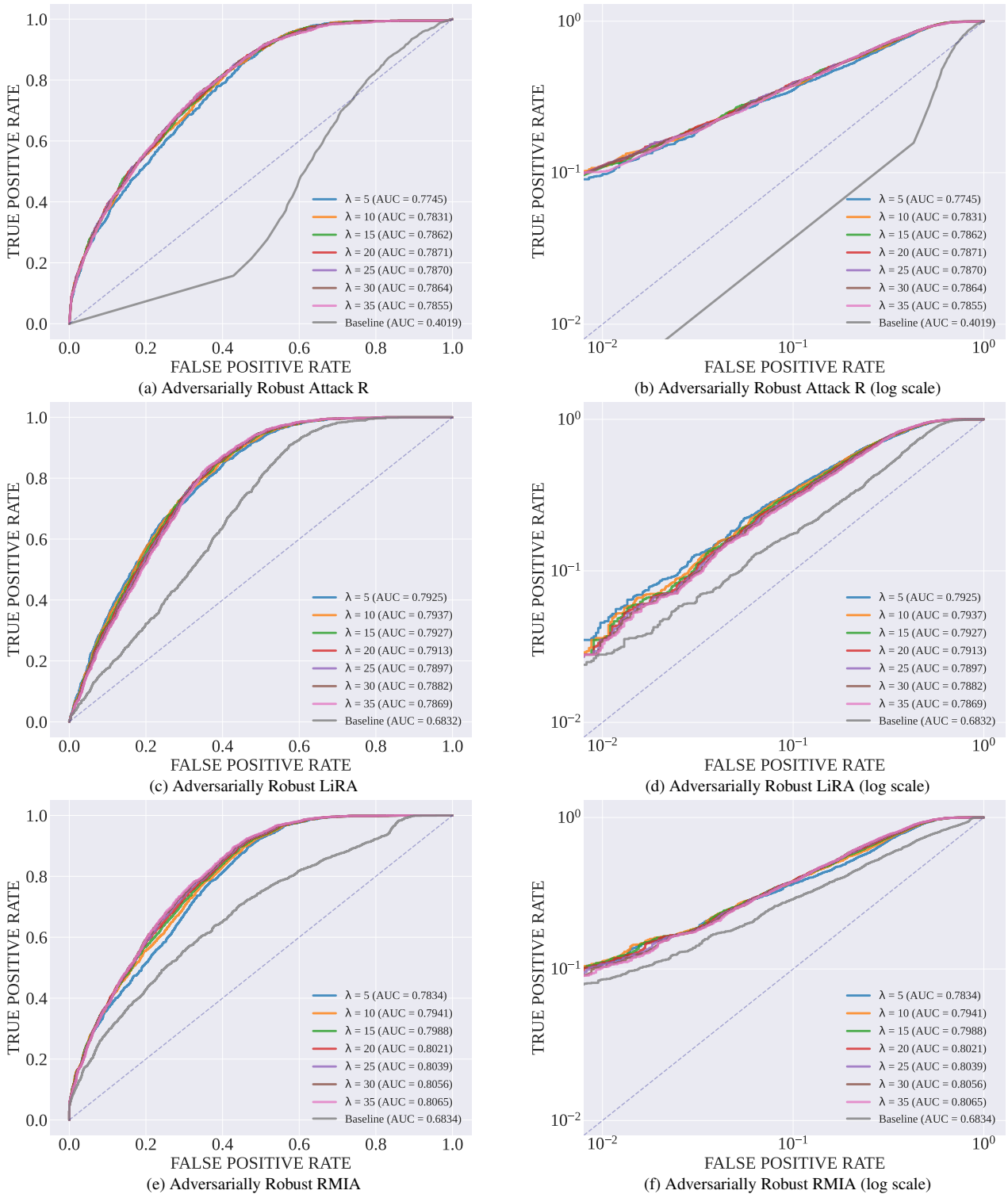


(e) Fabricated Members by I-APGD



(f) Fabricated Members by OURS Attack

Figure 22. Comparison of the ROC Curve for Our Member Fabrication Detection Across Diverse Perturbation Bounds on **ImageNet-100**.



(a) Adversarially Robust Attack R

(b) Adversarially Robust Attack R (log scale)

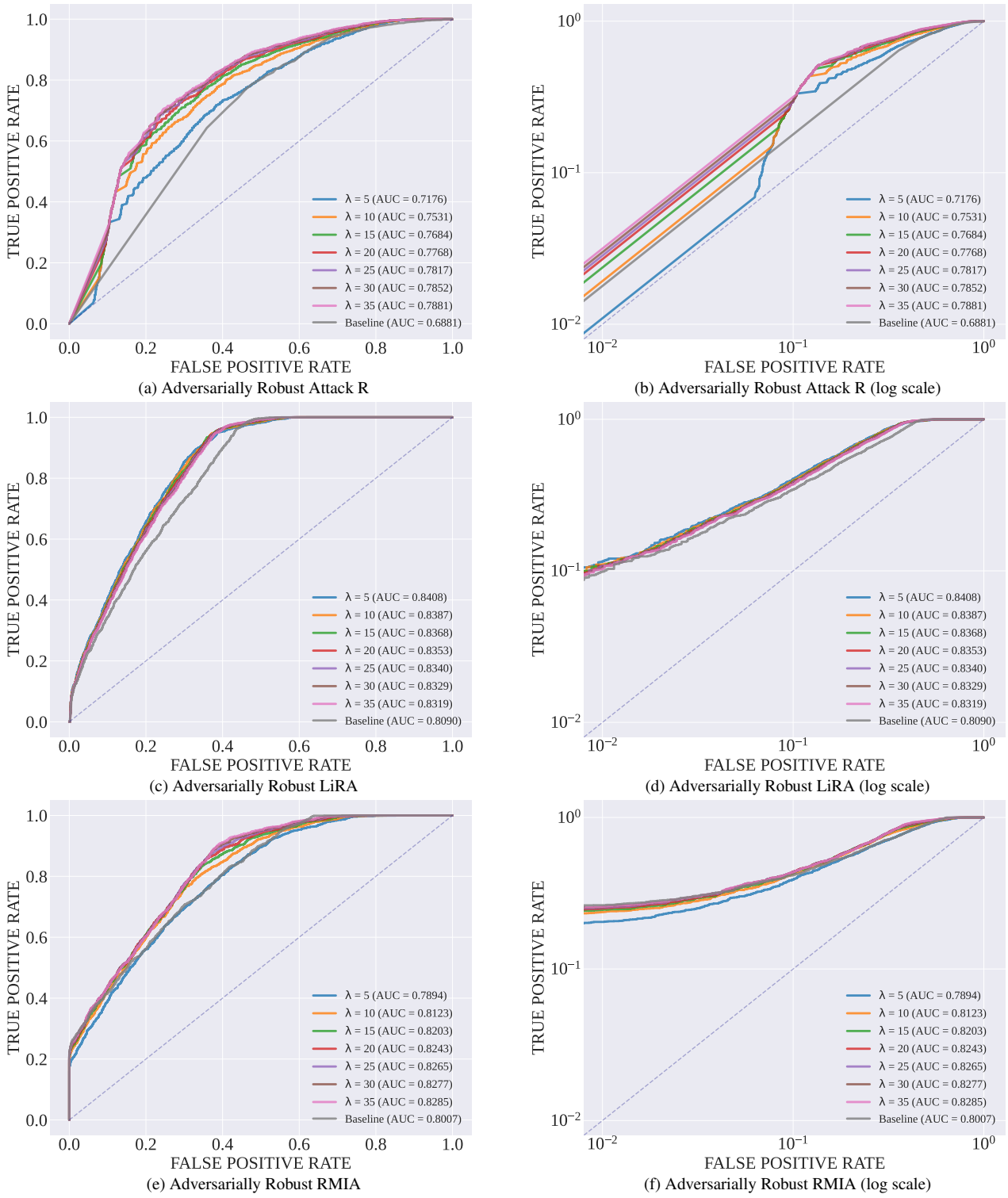
(c) Adversarially Robust LiRA

(d) Adversarially Robust LiRA (log scale)

(e) Adversarially Robust RMIA

(f) Adversarially Robust RMIA (log scale)

Figure 23. Comparison of ROC Curves for Our Adversarially Robust MIAs and Baselines on **CIFAR-10**.



(a) Adversarially Robust Attack R

(b) Adversarially Robust Attack R (log scale)

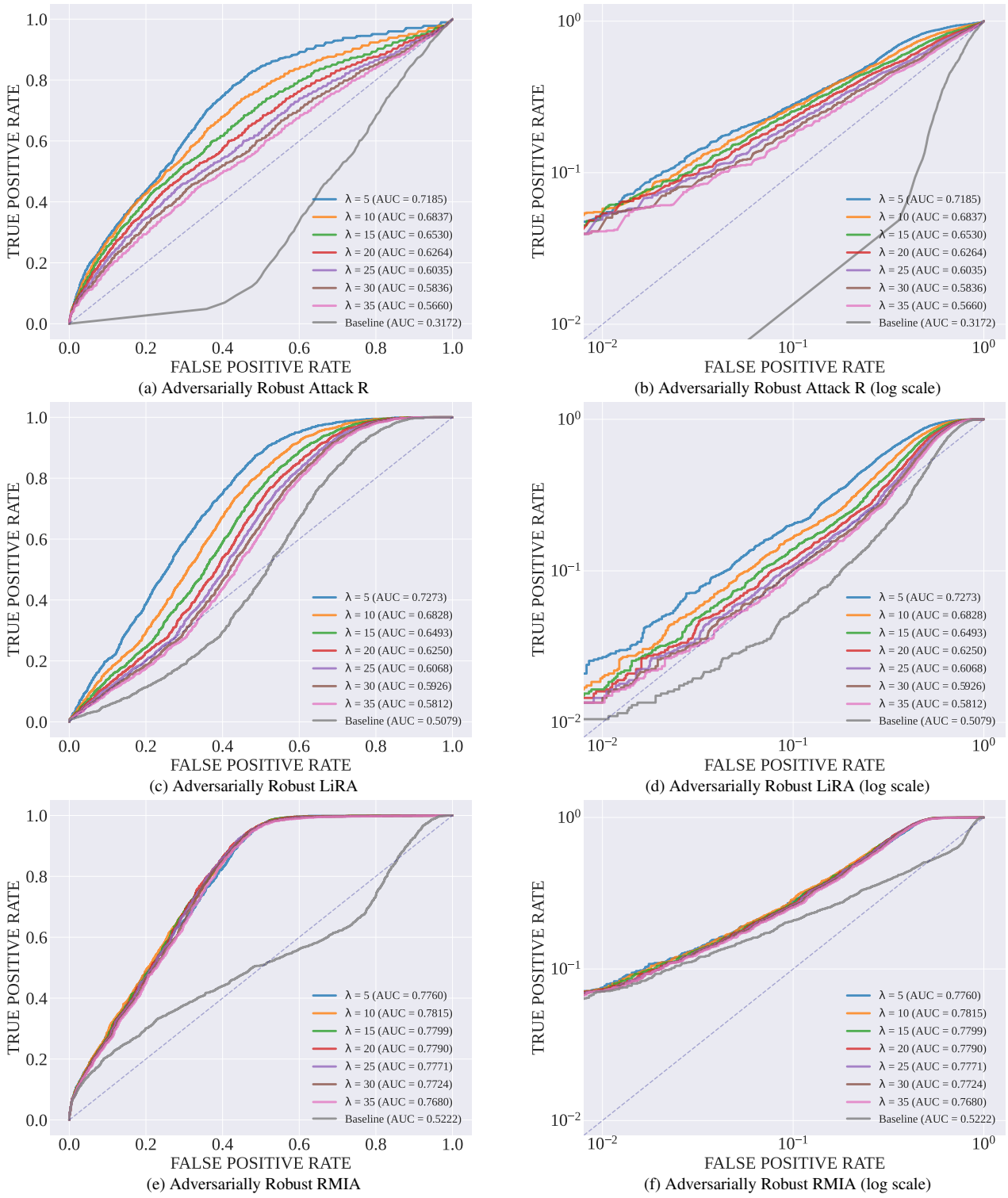
(c) Adversarially Robust LiRA

(d) Adversarially Robust LiRA (log scale)

(e) Adversarially Robust RMIA

(f) Adversarially Robust RMIA (log scale)

Figure 24. Comparison of ROC Curves for Our Adversarially Robust MIAs and Baselines on **CIFAR-100**.



(a) Adversarially Robust Attack R

(b) Adversarially Robust Attack R (log scale)

(c) Adversarially Robust LiRA

(d) Adversarially Robust LiRA (log scale)

(e) Adversarially Robust RMIA

(f) Adversarially Robust RMIA (log scale)

Figure 25. Comparison of ROC Curves for Our Adversarially Robust MIAs and Baselines on **SVHN**.

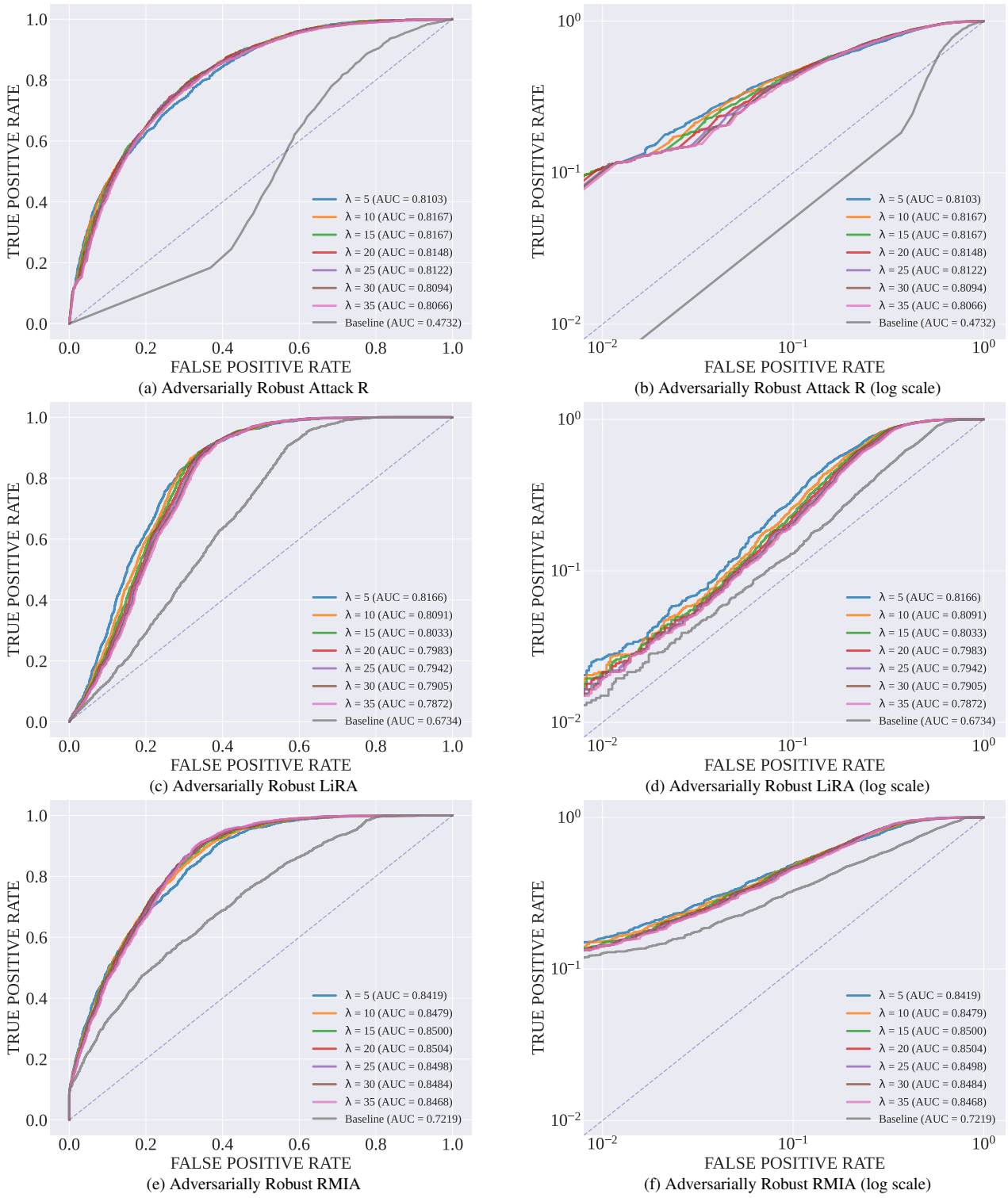


Figure 26. Comparison of ROC Curves for Our Adversarially Robust MIAs and Baselines on **CINIC-10**.

Table 3. The Comparison of **Error Area** between Our Member Fabrication Attack with Baselines across Diverse Datasets.

PERTURBATION	NATURAL	I-FGSM	I-BIM	I-PGD	I-CW	I-APGD	OURS
CIFAR-10							
$\|\delta\|_\infty \leq 1.0/255$	0.3685	0.5649	0.5955	0.5953	0.5818	0.5942	0.5969
$\|\delta\|_\infty \leq 2.0/255$	0.3685	0.6773	0.7723	0.7719	0.7477	0.7679	0.7761
$\|\delta\|_\infty \leq 3.0/255$	0.3685	0.7374	0.8833	0.8827	0.8593	0.8783	0.8887
$\|\delta\|_\infty \leq 4.0/255$	0.3685	0.7706	0.9403	0.9399	0.9228	0.9369	0.9451
$\|\delta\|_\infty \leq 5.0/255$	0.3685	0.7883	0.9684	0.9680	0.9568	0.9657	0.9719
$\|\delta\|_\infty \leq 6.0/255$	0.3685	0.7962	0.9828	0.9826	0.9755	0.9805	0.9852
$\|\delta\|_\infty \leq 7.0/255$	0.3685	0.7973	0.9903	0.9903	0.9854	0.9886	0.9922
$\|\delta\|_\infty \leq 8.0/255$	0.3685	0.7934	0.9946	0.9943	0.9913	0.9931	0.9958
CIFAR-100							
$\|\delta\|_\infty \leq 1.0/255$	0.1039	0.1814	0.1948	0.1947	0.1857	0.1943	0.1955
$\|\delta\|_\infty \leq 2.0/255$	0.1039	0.2547	0.3238	0.3235	0.2971	0.3211	0.3276
$\|\delta\|_\infty \leq 3.0/255$	0.1039	0.3092	0.4664	0.4657	0.4255	0.4616	0.4771
$\|\delta\|_\infty \leq 4.0/255$	0.1039	0.3445	0.5935	0.5917	0.5466	0.5880	0.6136
$\|\delta\|_\infty \leq 5.0/255$	0.1039	0.3626	0.6942	0.6925	0.6489	0.6905	0.7225
$\|\delta\|_\infty \leq 6.0/255$	0.1039	0.3670	0.7730	0.7712	0.7316	0.7695	0.8069
$\|\delta\|_\infty \leq 7.0/255$	0.1039	0.3622	0.8346	0.8317	0.7979	0.8309	0.8682
$\|\delta\|_\infty \leq 8.0/255$	0.1039	0.3512	0.8781	0.8748	0.8472	0.8763	0.9101
SVHN							
$\|\delta\|_\infty \leq 1.0/255$	0.4448	0.6587	0.7042	0.7042	0.6763	0.7023	0.7056
$\|\delta\|_\infty \leq 2.0/255$	0.4448	0.7382	0.8511	0.8512	0.8146	0.8489	0.8555
$\|\delta\|_\infty \leq 3.0/255$	0.4448	0.7539	0.9209	0.9212	0.8898	0.9210	0.9279
$\|\delta\|_\infty \leq 4.0/255$	0.4448	0.7407	0.9554	0.9548	0.9318	0.9560	0.9618
$\|\delta\|_\infty \leq 5.0/255$	0.4448	0.7109	0.9734	0.9733	0.9550	0.9745	0.9794
$\|\delta\|_\infty \leq 6.0/255$	0.4448	0.6709	0.9839	0.9833	0.9699	0.9848	0.9886
$\|\delta\|_\infty \leq 7.0/255$	0.4448	0.6261	0.9895	0.9893	0.9793	0.9906	0.9933
$\|\delta\|_\infty \leq 8.0/255$	0.4448	0.5792	0.9934	0.9934	0.9851	0.9938	0.9963
CINIC-10							
$\|\delta\|_\infty \leq 1.0/255$	0.2688	0.4522	0.4921	0.4919	0.4770	0.4900	0.4936
$\|\delta\|_\infty \leq 2.0/255$	0.2688	0.5698	0.7054	0.7051	0.6749	0.6995	0.7113
$\|\delta\|_\infty \leq 3.0/255$	0.2688	0.6316	0.8450	0.8442	0.8138	0.8383	0.8552
$\|\delta\|_\infty \leq 4.0/255$	0.2688	0.6619	0.9207	0.9198	0.8971	0.9162	0.9324
$\|\delta\|_\infty \leq 5.0/255$	0.2688	0.6750	0.9597	0.9590	0.9438	0.9565	0.9694
$\|\delta\|_\infty \leq 6.0/255$	0.2688	0.6779	0.9788	0.9783	0.9696	0.9768	0.9857
$\|\delta\|_\infty \leq 7.0/255$	0.2688	0.6751	0.9885	0.9880	0.9821	0.9868	0.9931
$\|\delta\|_\infty \leq 8.0/255$	0.2688	0.6679	0.9934	0.9931	0.9896	0.9922	0.9966
IMAGENET-100							
$\|\delta\|_\infty \leq 0.5/255$	0.4664	0.7231	0.7890	0.7888	0.7623	0.7850	0.7920
$\|\delta\|_\infty \leq 1.0/255$	0.4664	0.8151	0.9334	0.9331	0.9135	0.9274	0.9381
$\|\delta\|_\infty \leq 1.5/255$	0.4664	0.8552	0.9799	0.9797	0.9712	0.9763	0.9834
$\|\delta\|_\infty \leq 2.0/255$	0.4664	0.8757	0.9932	0.9931	0.9902	0.9918	0.9954

Table 4. The **Error Area** of Our Member Fabrication Attack across Diverse MIAs and Datasets.

Perturbation	Loss	LiRA	Attack R	RMIA
CIFAR-10				
Natural	0.3685	0.2814	0.3110	0.2830
$\|\delta\|_\infty \leq 4.0/255$	0.9451	0.3523	0.8852	0.3502
CIFAR-100				
Natural	0.1040	0.0423	0.1021	0.0808
$\|\delta\|_\infty \leq 4.0/255$	0.6136	0.3398	0.5217	0.3178
SVHN				
Natural	0.4448	0.3936	0.4284	0.3885
$\|\delta\|_\infty \leq 4.0/255$	0.9618	0.5905	0.9372	0.5671
CINIC-10				
Natural	0.2688	0.1532	0.2164	0.1704
$\|\delta\|_\infty \leq 4.0/255$	0.9324	0.5001	0.8371	0.3859

Table 5. The **Equal Error Rate** of Our Member Fabrication Attack across Diverse MIAs and Datasets.

Perturbation	Loss	LiRA	Attack R	RMIA
CIFAR-10				
Natural	42.30%	36.70%	36.35%	36.40%
$\|\delta\|_\infty \leq 4.0/255$	87.60%	41.85%	82.25%	38.20%
CIFAR-100				
Natural	15.60%	12.10%	17.40%	18.30%
$\|\delta\|_\infty \leq 4.0/255$	59.80%	41.00%	67.20%	39.05%
SVHN				
Natural	46.90%	43.90%	46.40%	43.75%
$\|\delta\|_\infty \leq 4.0/255$	89.70%	60.45%	88.20%	53.25%
CINIC-10				
Natural	33.50%	26.00%	29.70%	27.55%
$\|\delta\|_\infty \leq 4.0/255$	86.85%	51.95%	81.95%	42.00%

Table 6. The Comparison of **Equal Error Rate** between Our Member Fabrication Attack with Baselines across Diverse Datasets.

PERTURBATION	NATURAL	I-FGSM	I-BIM	I-PGD	I-CW	I-APGD	OURS
CIFAR-10							
$\|\delta\|_\infty \leq 1.0/255$	42.30%	56.60%	58.80%	58.75%	57.90%	58.65%	58.90%
$\|\delta\|_\infty \leq 2.0/255$	42.30%	65.00%	72.10%	72.10%	70.00%	71.85%	72.40%
$\|\delta\|_\infty \leq 3.0/255$	42.30%	69.40%	80.55%	80.55%	77.95%	80.10%	81.00%
$\|\delta\|_\infty \leq 4.0/255$	42.30%	71.80%	86.80%	86.75%	84.50%	86.30%	87.60%
$\|\delta\|_\infty \leq 5.0/255$	42.30%	73.20%	90.85%	90.75%	89.05%	90.50%	91.45%
$\|\delta\|_\infty \leq 6.0/255$	42.30%	73.75%	93.20%	93.25%	91.80%	92.85%	93.85%
$\|\delta\|_\infty \leq 7.0/255$	42.30%	73.95%	95.15%	95.05%	93.85%	94.80%	95.55%
$\|\delta\|_\infty \leq 8.0/255$	42.30%	73.60%	96.40%	96.30%	95.45%	96.00%	96.75%
CIFAR-100							
$\|\delta\|_\infty \leq 1.0/255$	15.60%	23.65%	24.80%	24.80%	24.35%	24.80%	24.80%
$\|\delta\|_\infty \leq 2.0/255$	15.60%	30.75%	36.80%	36.75%	34.55%	36.50%	37.10%
$\|\delta\|_\infty \leq 3.0/255$	15.60%	35.30%	48.20%	48.25%	45.00%	48.20%	49.05%
$\|\delta\|_\infty \leq 4.0/255$	15.60%	38.55%	58.05%	58.05%	54.75%	57.85%	59.80%
$\|\delta\|_\infty \leq 5.0/255$	15.60%	40.20%	66.50%	66.45%	62.75%	66.40%	68.95%
$\|\delta\|_\infty \leq 6.0/255$	15.60%	40.20%	73.20%	72.95%	69.75%	72.80%	75.80%
$\|\delta\|_\infty \leq 7.0/255$	15.60%	39.75%	78.25%	78.15%	75.25%	78.25%	82.00%
$\|\delta\|_\infty \leq 8.0/255$	15.60%	39.00%	82.55%	82.05%	79.55%	82.30%	86.05%
SVHN							
$\|\delta\|_\infty \leq 1.0/255$	46.90%	61.75%	65.30%	65.30%	63.20%	65.10%	65.35%
$\|\delta\|_\infty \leq 2.0/255$	46.90%	68.40%	77.60%	77.65%	73.90%	77.40%	77.95%
$\|\delta\|_\infty \leq 3.0/255$	46.90%	69.40%	84.70%	84.75%	81.05%	84.65%	85.25%
$\|\delta\|_\infty \leq 4.0/255$	46.90%	69.20%	88.75%	88.60%	85.30%	88.95%	89.70%
$\|\delta\|_\infty \leq 5.0/255$	46.90%	66.65%	91.25%	91.20%	88.40%	91.60%	92.55%
$\|\delta\|_\infty \leq 6.0/255$	46.90%	64.10%	93.00%	93.10%	90.35%	93.70%	94.50%
$\|\delta\|_\infty \leq 7.0/255$	46.90%	60.35%	94.45%	94.55%	92.30%	95.10%	96.05%
$\|\delta\|_\infty \leq 8.0/255$	46.90%	56.80%	95.50%	95.55%	93.45%	96.00%	97.10%
CINIC-10							
$\|\delta\|_\infty \leq 1.0/255$	33.50%	48.40%	51.15%	51.10%	50.30%	50.90%	51.35%
$\|\delta\|_\infty \leq 2.0/255$	33.50%	57.45%	66.85%	66.75%	64.80%	66.35%	67.20%
$\|\delta\|_\infty \leq 3.0/255$	33.50%	61.90%	77.70%	77.95%	75.40%	77.45%	78.80%
$\|\delta\|_\infty \leq 4.0/255$	33.50%	64.25%	85.35%	85.25%	82.55%	85.00%	86.85%
$\|\delta\|_\infty \leq 5.0/255$	33.50%	65.35%	90.45%	90.45%	88.10%	90.30%	91.80%
$\|\delta\|_\infty \leq 6.0/255$	33.50%	65.35%	93.25%	93.15%	91.50%	92.95%	94.85%
$\|\delta\|_\infty \leq 7.0/255$	33.50%	65.10%	95.25%	95.20%	93.70%	95.05%	96.50%
$\|\delta\|_\infty \leq 8.0/255$	33.50%	64.20%	96.60%	96.35%	95.50%	96.25%	97.70%
IMAGENET-100							
$\|\delta\|_\infty \leq 0.5/255$	47.55%	66.45%	72.55%	72.55%	69.75%	71.80%	72.90%
$\|\delta\|_\infty \leq 1.0/255$	47.55%	73.90%	85.60%	85.45%	83.30%	84.90%	86.15%
$\|\delta\|_\infty \leq 1.5/255$	47.55%	77.30%	92.45%	92.55%	90.70%	91.70%	93.35%
$\|\delta\|_\infty \leq 2.0/255$	47.55%	79.20%	95.95%	96.05%	94.75%	95.25%	96.85%

Table 7. Comparison of Attack R and Our Adversarially Robust Attack R.

METHODS	**AUC** VALUE	EER VALUE	TPR@1%FPR	TPR@5%FPR	TPR@10%FPR	TPR@20%FPR
CIFAR-10 ($\|\delta\|_\infty \leq 4.0/255$)						
ATTACK R BASELINE	0.4019	58.20%	0.37%	1.84%	3.68%	7.35%
[OURS] $\lambda = 5$	0.7745	31.30%	9.80%	25.16%	35.25%	52.25%
[OURS] $\lambda = 10$	0.7831	31.08%	11.05%	26.15%	37.55%	55.55%
[OURS] $\lambda = 15$	0.7862	30.15%	10.85%	26.75%	37.55%	56.08%
[OURS] $\lambda = 20$	0.7871	29.80%	11.13%	25.80%	39.46%	56.10%
[OURS] $\lambda = 25$	0.7870	29.78%	11.05%	25.30%	39.05%	55.70%
[OURS] $\lambda = 30$	0.7864	29.68%	10.95%	24.85%	37.95%	55.82%
[OURS] $\lambda = 35$	0.7855	29.58%	10.20%	24.85%	37.57%	56.43%
CIFAR-100 ($\|\delta\|_\infty \leq 4.0/255$)						
ATTACK R BASELINE	0.6881	35.90%	1.79%	8.96%	17.91%	35.82%
[OURS] $\lambda = 5$	0.7176	33.67%	1.10%	5.48%	29.75%	47.95%
[OURS] $\lambda = 10$	0.7531	31.32%	1.93%	9.64%	29.75%	55.85%
[OURS] $\lambda = 15$	0.7684	29.23%	2.37%	11.85%	29.75%	58.90%
[OURS] $\lambda = 20$	0.7768	28.40%	2.69%	13.45%	29.75%	60.93%
[OURS] $\lambda = 25$	0.7817	28.45%	2.84%	14.21%	29.75%	61.38%
[OURS] $\lambda = 30$	0.7852	27.52%	2.99%	14.96%	29.93%	62.71%
[OURS] $\lambda = 35$	0.7881	27.12%	3.16%	15.80%	31.61%	63.26%
SVHN ($\|\delta\|_\infty \leq 4.0/255$)						
ATTACK R BASELINE	0.3172	62.10%	0.14%	0.68%	1.35%	2.71%
[OURS] $\lambda = 5$	0.7185	33.75%	5.30%	18.97%	28.20%	43.60%
[OURS] $\lambda = 10$	0.6837	36.02%	5.80%	16.70%	27.20%	42.75%
[OURS] $\lambda = 15$	0.6530	39.17%	5.42%	15.40%	25.45%	40.45%
[OURS] $\lambda = 20$	0.6264	41.05%	5.40%	14.25%	23.05%	37.40%
[OURS] $\lambda = 25$	0.6035	43.33%	5.20%	12.55%	20.90%	34.75%
[OURS] $\lambda = 30$	0.5836	45.00%	4.58%	11.42%	19.03%	32.08%
[OURS] $\lambda = 35$	0.5660	45.95%	4.10%	10.95%	17.70%	29.35%
CINIC-10 ($\|\delta\|_\infty \leq 4.0/255$)						
ATTACK R BASELINE	0.4732	53.02%	0.50%	2.49%	4.98%	9.97%
[OURS] $\lambda = 5$	0.8103	27.68%	10.85%	31.23%	46.37%	62.66%
[OURS] $\lambda = 10$	0.8167	26.75%	10.85%	31.15%	46.50%	64.35%
[OURS] $\lambda = 15$	0.8167	26.45%	10.85%	28.45%	45.58%	64.51%
[OURS] $\lambda = 20$	0.8148	26.10%	10.85%	26.85%	45.10%	64.80%
[OURS] $\lambda = 25$	0.8122	26.38%	10.38%	25.37%	44.55%	64.47%
[OURS] $\lambda = 30$	0.8094	26.60%	10.13%	24.40%	42.43%	64.55%
[OURS] $\lambda = 35$	0.8066	26.90%	9.75%	23.60%	41.25%	64.35%

Table 8. Comparison of LiRA and Our Adversarially Robust LiRA.

METHODS	**AUC** VALUE	EER VALUE	TPR@1%FPR	TPR@5%FPR	TPR@10%FPR	TPR@20%FPR
CIFAR-10 ($\|\delta\|_\infty \leq 4.0/255$)						
LIRA BASELINE	0.6832	38.48%	2.80%	9.55%	17.55%	32.25%
[OURS] $\lambda = 5$	0.7925	28.95%	4.55%	18.45%	34.35%	57.05%
[OURS] $\lambda = 10$	0.7937	28.23%	3.60%	17.95%	32.95%	57.00%
[OURS] $\lambda = 15$	0.7927	28.43%	3.55%	17.05%	31.95%	55.50%
[OURS] $\lambda = 20$	0.7913	28.60%	3.50%	16.95%	31.25%	54.25%
[OURS] $\lambda = 25$	0.7897	28.62%	3.50%	16.60%	30.95%	54.05%
[OURS] $\lambda = 30$	0.7882	28.85%	3.50%	16.50%	29.90%	52.40%
[OURS] $\lambda = 35$	0.7869	29.03%	3.30%	16.20%	29.65%	52.00%
CIFAR-100 ($\|\delta\|_\infty \leq 4.0/255$)						
LIRA BASELINE	0.8090	28.78%	9.90%	23.00%	34.40%	56.25%
[OURS] $\lambda = 5$	0.8408	24.73%	11.60%	26.55%	40.65%	65.60%
[OURS] $\lambda = 10$	0.8387	25.45%	11.05%	25.95%	39.50%	64.30%
[OURS] $\lambda = 15$	0.8368	25.50%	10.75%	25.00%	38.80%	63.95%
[OURS] $\lambda = 20$	0.8353	25.65%	10.80%	24.55%	38.50%	63.50%
[OURS] $\lambda = 25$	0.8340	25.77%	10.80%	23.90%	38.20%	62.65%
[OURS] $\lambda = 30$	0.8329	26.05%	10.60%	23.90%	37.60%	61.95%
[OURS] $\lambda = 35$	0.8319	26.35%	10.35%	23.80%	37.25%	61.25%
SVHN ($\|\delta\|_\infty \leq 4.0/255$)						
LIRA BASELINE	0.5079	51.30%	1.05%	2.85%	5.30%	11.35%
[OURS] $\lambda = 5$	0.7273	34.00%	2.70%	11.50%	20.35%	38.65%
[OURS] $\lambda = 10$	0.6828	37.60%	2.00%	8.75%	16.55%	29.50%
[OURS] $\lambda = 15$	0.6493	40.35%	1.65%	7.40%	13.90%	24.50%
[OURS] $\lambda = 20$	0.6250	42.40%	1.60%	6.40%	12.00%	22.70%
[OURS] $\lambda = 25$	0.6068	43.65%	1.45%	5.95%	10.80%	20.20%
[OURS] $\lambda = 30$	0.5926	44.65%	1.45%	5.40%	10.15%	18.60%
[OURS] $\lambda = 35$	0.5812	45.50%	1.35%	5.15%	9.40%	17.50%
CINIC-10 ($\|\delta\|_\infty \leq 4.0/255$)						
LIRA BASELINE	0.6734	38.50%	1.50%	6.90%	13.00%	29.10%
[OURS] $\lambda = 5$	0.8166	24.82%	2.65%	12.05%	29.95%	62.30%
[OURS] $\lambda = 10$	0.8091	25.85%	2.15%	10.90%	26.00%	59.40%
[OURS] $\lambda = 15$	0.8033	26.47%	2.10%	10.35%	23.60%	57.30%
[OURS] $\lambda = 20$	0.7983	27.20%	2.15%	9.85%	22.55%	55.10%
[OURS] $\lambda = 25$	0.7942	27.70%	2.00%	9.45%	21.20%	53.10%
[OURS] $\lambda = 30$	0.7905	28.18%	2.00%	9.30%	20.40%	51.80%
[OURS] $\lambda = 35$	0.7872	28.50%	1.95%	9.15%	20.15%	51.10%

Table 9. Comparison of RMIA and Our Adversarially Robust RMIA.

METHODS	**AUC** VALUE	EER VALUE	TPR@1%FPR	TPR@5%FPR	TPR@10%FPR	TPR@20%FPR
CIFAR-10 ($\|\delta\|_\infty \leq 4.0/255$)						
RMIA BASELINE	0.6834	37.30%	8.55%	18.50%	28.95%	42.80%
[OURS] $\lambda = 5$	0.7834	30.88%	10.90%	25.60%	36.40%	51.65%
[OURS] $\lambda = 10$	0.7941	29.98%	11.30%	25.85%	38.15%	55.65%
[OURS] $\lambda = 15$	0.7988	29.23%	11.10%	25.70%	37.90%	56.90%
[OURS] $\lambda = 20$	0.8021	28.82%	11.00%	25.45%	37.95%	58.25%
[OURS] $\lambda = 25$	0.8039	28.50%	10.75%	25.30%	37.75%	58.50%
[OURS] $\lambda = 30$	0.8056	28.05%	10.35%	25.05%	37.80%	59.20%
[OURS] $\lambda = 35$	0.8065	27.85%	10.30%	25.30%	37.85%	59.45%
CIFAR-100 ($\|\delta\|_\infty \leq 4.0/255$)						
RMIA BASELINE	0.8007	29.80%	26.35%	34.15%	41.55%	56.15%
[OURS] $\lambda = 5$	0.7894	30.18%	20.45%	29.65%	38.85%	56.10%
[OURS] $\lambda = 10$	0.8123	27.73%	23.80%	32.75%	42.55%	59.85%
[OURS] $\lambda = 15$	0.8203	27.57%	24.65%	33.65%	43.00%	60.80%
[OURS] $\lambda = 20$	0.8243	27.50%	25.15%	34.30%	43.95%	60.55%
[OURS] $\lambda = 25$	0.8265	27.50%	25.45%	34.45%	43.70%	60.50%
[OURS] $\lambda = 30$	0.8277	27.52%	25.70%	34.75%	43.40%	60.35%
[OURS] $\lambda = 35$	0.8285	27.62%	25.85%	34.90%	43.55%	60.15%
SVHN ($\|\delta\|_\infty \leq 4.0/255$)						
RMIA BASELINE	0.5222	49.65%	7.10%	14.60%	20.95%	30.05%
[OURS] $\lambda = 5$	0.7760	31.75%	7.55%	17.70%	28.00%	48.45%
[OURS] $\lambda = 10$	0.7815	30.55%	7.35%	17.80%	28.70%	48.90%
[OURS] $\lambda = 15$	0.7799	30.45%	7.35%	17.25%	27.00%	47.80%
[OURS] $\lambda = 20$	0.7790	30.50%	7.35%	17.30%	26.75%	46.75%
[OURS] $\lambda = 25$	0.7771	30.95%	7.15%	17.15%	26.30%	46.75%
[OURS] $\lambda = 30$	0.7724	31.27%	7.15%	16.90%	26.00%	46.55%
[OURS] $\lambda = 35$	0.7680	31.95%	7.15%	16.15%	25.45%	45.00%
CINIC-10 ($\|\delta\|_\infty \leq 4.0/255$)						
RMIA BASELINE	0.7219	35.43%	12.75%	22.35%	32.85%	48.20%
[OURS] $\lambda = 5$	0.8419	25.67%	16.00%	33.60%	48.90%	67.50%
[OURS] $\lambda = 10$	0.8479	24.32%	15.20%	32.00%	47.95%	69.35%
[OURS] $\lambda = 15$	0.8500	23.93%	14.95%	31.95%	47.30%	69.85%
[OURS] $\lambda = 20$	0.8504	23.70%	14.25%	31.30%	46.95%	69.50%
[OURS] $\lambda = 25$	0.8498	23.80%	14.40%	31.20%	46.45%	68.95%
[OURS] $\lambda = 30$	0.8484	24.00%	14.10%	30.70%	46.25%	67.15%
[OURS] $\lambda = 35$	0.8468	24.15%	14.20%	29.30%	45.80%	66.50%

References

[1] Jason W Bentley, Daniel Gibney, Gary Hoppenworth, and Sumit Kumar Jha. Quantifying membership inference vulnerability via generalization gap and other model metrics. *arXiv preprint arXiv:2009.05669*, 2020. 2

[2] Siddhant Bhambri, Sumanyu Muku, Avinash Tulasi, and Arun Balaji Buduru. A survey of black-box adversarial attacks on computer vision models. *arXiv preprint arXiv:1912.01667*, 2019. 4

[3] Christopher M Bishop. Pattern recognition. *Machine learning*, 128(9), 2006. 2

[4] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *ACM Workshop on Artificial Intelligence and Security*, 2017. 2

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *CVPR*, 2017. 3, 8, 9

[6] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security*, 2019. 2

[7] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *IEEE Symposium on Security and Privacy*, 2022. 1, 2, 3, 4, 8, 7, 9

[8] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *ICCV*, 2015. 2

[9] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *ACM SIGSAC Conference on Computer and Communications Security*, 2020. 2

[10] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. When machine unlearning jeopardizes privacy. In *ACM SIGSAC Conference on Computer and Communications Security*, 2021. 1

[11] Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *CVPR*, 2020. 3

[12] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *ICML*, 2021. 1

[13] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020. 3, 4, 8, 9

[14] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018. 8, 9

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 8, 9

[16] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *ACM SIGSAC Conference on Computer and Communications Security*, 2015. 1

[17] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *ACM SIGSAC Conference on Computer and Communications Security*, 2018. 1

[18] Ruize Gao, Feng Liu, Jingfeng Zhang, Bo Han, Tongliang Liu, Gang Niu, and Masashi Sugiyama. Maximum mean discrepancy test is aware of adversarial attacks. In *ICML*, 2021. 2

[19] Ruize Gao, Jiongxiao Wang, Kaiwen Zhou, Feng Liu, Binghui Xie, Gang Niu, Bo Han, and James Cheng. Fast and reliable evaluation of adversarial robustness with minimum-margin attack. In *ICML*, 2022. 3, 8

[20] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 3, 2, 8, 9

[21] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv:1702.06280*, 2017. 3

[22] Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *ICML*, 2019. 4

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 8, 1

[24] Sorami Hisamoto, Matt Post, and Kevin Duh. Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system? *Transactions of the Association for Computational Linguistics*, 8:49–63, 2020. 1

[25] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLOS Genetics*, 4:1–9, 2008. 1, 3

[26] Keke Huang, Ruize Gao, Bogdan Cautis, and Xiaokui Xiao. Scalable continuous-time diffusion framework for network inference and influence estimation. In *WWW*, 2024. 1

[27] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *ICML*, 2018. 4

[28] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019. 1

[29] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 8, 9

[30] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world. In *ICLR*, 2017. 2, 8, 9

[31] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018. 5, 3

[32] Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated white-box membership inference. In *USENIX Security*, 2020. 1, 2, 7

[33] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. Membership inference attacks and defenses in classification models. In *ACM Conference on Data and Application Security and Privacy*, 2021. 1

[34] Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. In *ICCV*, 2017. 3

[35] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 4

[36] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *ICLR*, 2018. 5, 3

[37] Xingjun Ma, Yuhao Niu, Lin Gu, Yisen Wang, Yitian Zhao, James Bailey, and Feng Lu. Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, 2021. 2

[38] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (Nov):2579–2605, 2008. 5, 7, 8

[39] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 3, 4, 8, 2, 9

[40] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *IEEE Symposium on Security and Privacy*, 2019. 7

[41] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *arXiv:1702.04267*, 2017. 3

[42] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19 (6):1236–1246, 2018. 1

[43] Sasi Kumar Murakonda and Reza Shokri. Ml privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning. *arXiv preprint arXiv:2007.09339*, 2020. 2, 3, 7

[44] Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. In *ACM SIGSAC Conference on Computer and Communications Security*, 2018. 1

[45] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE symposium on security and privacy*, 2019. 7, 1

[46] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 8

[47] Jerzy Neyman and Egon Sharpe Pearson. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933. 1

[48] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015. 2

[49] Bita Darvish Rouhani, Mohammad Samragh, Tara Javidi, and Farinaz Koushanfar. Curtail: Characterizing and thwarting adversarial deep learning. *arXiv:1709.02538*, 2017. 3

[50] Alexandre Sablayrolles, Matthijs Douze, Yann Ollivier, Cordelia Schmid, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *ICML*, 2019. 1

[51] Ahmed Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Annual Network and Distributed System Security Symposium*, 2019. 1, 2, 7

[52] Rui Shao, Ruize Gao, Bin Xie, Yixing Li, Kaiwen Zhou, Shuai Wang, Weili Guan, and Gongwei Chen. Hats: Hardness-aware trajectory synthesis for gui agents. In *CVPR*, 2026. 1

[53] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017. 1, 2, 3, 4

[54] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *ACM SIGSAC Conference on Computer and Communications Security*, 2017. 2

[55] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *USENIX Security*, 2021. 1

[56] Shuang Song and David Marn. Introducing a new privacy testing library in tensorflow, 2020. 3

[57] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 3, 2

[58] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *USENIX Security*, 2016. 1

[59] Haonan Wang, Qianli Shen, Yao Tong, Yang Zhang, and Kenji Kawaguchi. The stronger the diffusion model, the easier the backdoor: Data poisoning to induce copyright breaches without adjusting finetuning pipeline. In *ICML*, 2024. 3, 4

[60] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, 2019. 2, 3

[61] Lauren Watson, Chuan Guo, Graham Cormode, and Alex Sablayrolles. On the importance of difficulty calibration in membership inference attacks. *arXiv preprint arXiv:2111.08440*, 2021. 1, 3, 7

[62] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, 2020. 3

[63] Hui Y Xiong, Babak Alipanahi, Leo J Lee, Hannes Bretschneider, Daniele Merico, Ryan KC Yuen, Yimin Hua, Serge Gueroussov, Hamed S Najafabadi, Timothy R Hughes, et al. The human splicing code reveals new insights into the genetic determinants of disease. *Science*, 347(6218), 2015. 1

[64] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *ACM SIGSAC Conference on Computer and Communications Security*, 2022. 1, 2, 3, 4, 8, 7, 9

[65] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE computer security foundations symposium*, 2018. 1, 3, 4, 8, 2

[66] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 8

[67] Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In *ICML*, 2024. 1, 3, 4, 8, 9

[68] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64:107–115, 2021. 2

[69] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *ICLR*, 2021. 3

[70] Yonggang Zhang, Ya Li, Tongliang Liu, and Xinmei Tian. Dual-path distillation: A unified framework to improve black-box attacks. In *ICML*, 2020. 2