Mixup Training for Generative Models to Defend Membership Inference Attacks

Zhe Ji[1], Qiansiqi Hu[1], Liyao Xiang[1], Chenghu Zhou[2]
[1]Shanghai Jiao Tong University, [2]Institute of Geographic Sciences and Natural Resources Research, CAS

IEEE INFOCOM 2023 - IEEE Conference on Computer Communications | 979-8-3503-3414-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/INFOCOM53939.2023.10229036

***Abstract*—With the popularity of machine learning, it has been a growing concern on the trained model revealing the private information of the training data. Membership inference attack (MIA) poses one of the threats by inferring whether a given sample participates in the training of the target model. Although MIA has been widely studied for discriminative models, for generative models, neither it nor its defense is extensively investigated. In this work, we propose a mixup training method for generative adversarial networks (GANs) as a defense against MIAs. Specifically, the original training data is replaced with their interpolations so that GANs would never overfit the original data. The intriguing part is an analysis from the hypothesis test perspective to theoretically prove our method could mitigate the AUC of the strongest likelihood ratio attack. Experimental results support that mixup training successfully defends the state-of-the-art MIAs for generative models, yet without model performance degradation or any additional training efforts, showing great promise to be deployed in practice.**

I. Introduction

Recent years have witnessed the rapid development of machine learning, where the data is becoming a key performance factor, and the demand for data is soaring up. However, sensitive data, especially those containing private personal information, raises concerns when they are used as the training data in machine learning. In particular, membership inference attacks (MIAs) pose as a significant threat by inferring whether a particular data sample has been used for training a target model. For example, if an individual's medical record has participated in the training of a disease prediction model, it would violate privacy protocols and expose the participation information. Hence how to protect personal data privacy from being breached while maintaining the usability of the data in the learning task is an urgent issue.

Membership inference attacks, or privacy attacks to the training data on generative models are less discussed compared to the discriminative models, but their threats are raising wide awareness in recent years. Typically, raw data are used to train the generative model, and the trained generative model is released, or the synthetic data generated are published to accomplish downstream learning tasks. However, generative models, such as generative adversarial networks (GANs), usually have the inclination to memorize training samples, making the model vulnerable to attacks that recover, or infer the private training data, from the released models or the published synthetic data. An example is that the powerful natural language model GPT-2, trained on scrapes of the public Internet, is found to be able to extract verbatim text sequences from its training data [1]. Another instance is the membership collision attack [2] against GANs, which allows an adversary to recover partial training data given some synthetic entries.

Depending on the target component, existing membership inference attacks against generative models can be generally classified into two categories: one type is against generators, where the membership criterion is based on the generator reconstruction loss [3], and the other is against discriminators, where discriminator scores of the target data point are used for judgment [4]. However, none of the attacks could claim that they are the strongest attack ever, nor they have stable performance across all circumstances. It is found in [5] that most prior attacks perform poorly with low true positive rate at low false positive rates. And the average-case 'accuracy' metrics in MIA usually fail to characterize whether the attack can confidently identify membership.

Defense against MIAs for generative models in previous works mainly focus on the design of GAN architectures. PrivGAN [6] trains the generator not only against the discriminator, but to defend MIAs, thus preventing memorization of the training set. PAR-GAN [7] partitions the training data into disjoint sets on which multiple discriminators and one generator are trained, reducing the generalization gap by approximating a mixture distribution of the training data. These two works adopt complex network architectures in training, resulting in significant increase in computational overhead. Other methods include differential privacy (DataLens [8]), which inserts controlled noise to limit each sample's impact on the model, but it does not consider the degradation of data utility due to the noise. None of the methods has investigated the defense performance against the strongest MIA, and thus could claim to be the strongest defense .

It is our goal in this work to design a defense method against the strongest MIA for generative models. Different from prior works, we focus on the training data: rather than directly training on the original data, we train the generative models on interpolations of the training data by $mixup$ [9], thus preventing the model from memorizing the original data. We give an in-depth analysis to how $mixup$ is able to defend against the 'strongest' MIA — the likelihood ratio attack, by showing the mixup training indeed reduces the likelihood

Liyao Xiang (xiangliyao08@sjtu.edu.cn) is the corresponding author with John Hopcroft Center, Shanghai Jiao Tong University, China.

This work was partially supported by National Key R&D Program of China under Grant 2021ZD0112801, NSF China (62272306, 62032020, 62136006, 42050105, 62020106005, 62061146002, 61960206002). Authors would like to appreciate the Student Innovation Center of SJTU for providing GPUs.

ratio with a large probability, and results in lowering the upper bound of the MIA AUC. From the training perspective, our method requires no additional architecture, nor any extra training efforts. And the method brings little degradation in data utility due to the effect of $mixup$ in stabilizing the GAN training.

Highlights of our contributions are as follows. *First,* we introduce an effective defense method against MIAs on generative models. The method is as simple as performing mixup training on the original training data, without bringing in additional training efforts or degrading the model performance. *Second,* we give theoretical evidence that our method mitigates the strongest MIA by reducing the likelihood ratio and thereby the attack AUC. *Finally,* supported by experimental results on different datasets, the mixup training successfully defends against the state-of-the-art MIAs, including those specifically designed for generative models and the strongest MIA, and shows superior performance than the prior defense methods in terms of privacy and data utility.

II. Related Work

Our work is mainly related to the following literature.

A. Membership Inference Attacks

In general, the membership inference attack (MIA) refers to a set of attacks that aim to determine whether a targeted sample belongs to the training set of a targeted model. It can be a threat to any machine learning model trained on private training sets, as it reveals the sensitive 'membership' of a data record.

At its early stage, membership inference attack is proposed by Shokri et al. [10] targeting at discriminative models. They assume the attacker has black-box access to the target model, and use shadow models to train a classifier which predicts the membership of a given sample. The shadow model idea is commonly used later on in other MIA methods. MIA against generative models were inspired and proposed in [3], [4], [11]. Among them, Logan [4] considers two scenarios where the attacker's prior knowledge differs: (1) the attacker has access to the discriminator, and (2) the attacker gains black-box accesses to the generator. In the first case, the output of the discriminator fed a target sample is used to infer the membership, based on the fact that the discriminator usually gives a higher score for the training data than the holdout data. For the second case, Logan trains a shadow model on data generated by the target model, and scores the target samples by the discriminator of the shadow model. Monte-Carlo attack [11] infers membership through the degree of aggregation of synthesized samples around the target sample. GAN-leaks [3] takes into account several different access permissions of attackers. The core of GAN-leaks is to determine the membership by the reconstruction loss which measures how well the target generator reconstructs the target sample. It is believed that the training samples enjoy a smaller reconstruction loss than the holdout samples.

Although the above attacks are based on different judging metrics, they are all based on the impact of overfitting the GAN on the training data. Our proposed mixup training approach replaces the training data to prevent the GAN from overfitting the training data at the source. Therefore, mixup training should provide some defense against the above-mentioned attacks.

B. Defense against MIAs

Recent works have been focusing on defense techniques to protect GAN from MIAs, such as privGAN [6], PAR-GAN [7]. PrivGAN constructs multiple generator-discriminator pairs trained on disjoint datasets and a built-in adversary to classify which pair generates the given synthesized sample. The generator is trained to fool both the discriminator and the adversary, and hence the synthetic data generated by the GAN trained on the training samples are indistinguishable from those generated by the GAN trained on other data points from the same distribution. PAR-GAN improves PrivGAN by adversarially training a generator against multiple discriminators where the training data of each discriminator are disjoint. PrivGAN and PAR-GAN are innovative in proposing new GAN architectures but incur significant training overhead as they both involve multiple models in the training frameworks. In contrast, our method mitigates such training overhead by proposing to train the generative models on mixed training data, imposing little changes on the original GAN architecture.

Additionally, Datalens [8] is also regarded as a defense using differential privacy, despite that it is not particularly designed against MIA. Datalens uses PATE framework to insert controlled noise into the aggregation of discriminator's gradients so as to make the participation of any training data record indistinguishable. However, the theoretical differential privacy guarantee induces significant accuracy drop for desired values of the privacy parameter [12].

III. Preliminaries

We introduce background for a better understanding of the paper.

A. The Likelihood Ratio Attack

The likelihood ratio attack (LIRA) [5], posing as the most severe MIA, is based on the hypothesis testing, with null and alternate hypotheses established as follows:

$$H_0: \quad \text{the target example is a member.}$$
$$H_1: \quad \text{the target example is not a member.}$$

We use M to represent the target model, and the hypothesis testing is performed to predict whether M is trained on the target example x. We denote datasets containing the target example as $\mathbb{D}_{in}$, and datasets not containing the target example as $\mathbb{D}_{out}$. For models trained on $\mathbb{D}_{in}$, we use $\mathbb{M}_{in}$ to represent their distribution, and the distribution of those trained on $\mathbb{D}_{out}$ is denoted as $\mathbb{M}_{out}$. The hypothesis test is to predict whether the target model M comes from $\mathbb{M}_{in}$ or $\mathbb{M}_{out}$. According to the Neyman-Pearson lemma [13], the strongest attack given a

fixed false positive rate can be realized based on the likelihood ratio between the two hypotheses, which is defined as:

$$\Lambda(M) := \frac{\Pr(M \mid \mathbb{M}_{in})}{\Pr(M \mid \mathbb{M}_{out})}.$$

However, it is difficult to perform analysis on distributions $\mathbb{M}_{in}$ and $\mathbb{M}_{out}$. Therefore, the likelihood ratio attack replaces distributions of models with distributions of losses, which are denoted by Q_{in} and Q_{out}, and hence defines:

$$\Lambda(l) := \frac{\Pr(l \mid Q_{in})}{\Pr(l \mid Q_{out})}.$$

In the following contents, Λ stands for the abbreviation of $\Lambda(l)$. Attackers depend on the value of Λ to determine whether the target example is a member or not. Specifically, let the rejection region be S_{rej} for Λ. The null hypothesis H_0 is rejected when $\Lambda \in S_{rej}$. We denote $\overline{S}_{rej}$ as the complementary set of S_{rej}, and Λ_{mem}, $\Lambda_{non-mem}$ as the value of Λ for a targeted member, and a non-member, respectively. With the denotations, we define the type I error as $P(\Lambda_{mem} \in S_{rej})$, and the type II error as $P(\Lambda_{non-mem} \in \overline{S}_{rej})$.

Given the likelihood ratio test results, the work of [5] further categorizes samples into inliers and outliers. For an outlier, the loss distributions of a member and of a non-member show great separability, indicating that the two loss distributions are easy to distinguish, and hence leads to a large value of Λ for a targeted member. By contrast, for an inlier, the value of Λ tends to be near 1, and thus the likelihood ratio attack is not good at telling whether it is a member or not. Inspired by these facts, our work focuses on the outliers, and mixup is supposed to be effective in alleviating the impact of outliers on model to enhance privacy protection.

B. Mixup Training

Mixup training [9] regularizes the neural network to favor simple linear behavior in between training examples. It is empirically proved that mixup can improve the generalization capability of neural networks.

In supervised learning, the task is to find a function $f \in \mathcal{F}$ that maps a feature vector x to a target vector y, which follow the joint distribution $P(x, y)$. A loss function is defined to penalize the difference between prediction $f(x)$ and actual target y for $(x, y) \sim P$. The learning process is to minimize the *expected risk*:

$$R(f) = \int l(f(x), y)\, \mathrm{d}P(x, y).$$

Since the distribution $P(x, y)$ is unknown in most practical situations, we usually estimate $P(x, y)$ by a given dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ where $(x_i, y_i) \sim P$:

$$P_\delta(x, y) = \frac{1}{n} \sum_{i=1}^{n} \delta(x = x_i, y = y_i)$$

where $\delta(x = x_i, y = y_i)$ is a Dirac mass centered at (x_i, y_i). Thus

$$R_\delta(f) = \int l(f(x), y) \mathrm{d}P_\delta(x, y) = \frac{1}{n} \sum_{i=1}^{n} l(f(x), y).$$

In contrast, $mixup$ estimates the risk $R(f)$ by a *vicinity distribution*:

$$R_\mu(f) = \frac{1}{m} \sum_{i=1}^{m} l(f(\tilde{x}_i), \tilde{y}_i)$$

where $(\tilde{x}_i, \tilde{y}_i) \sim P_\mu$:

$$P_\mu(\tilde{x}, \tilde{y}) = \frac{1}{n} \sum_{i=1}^{n} \mu(\tilde{x}, \tilde{y} | x_i, y_i),$$

$$\mu(\tilde{x}, \tilde{y} | x_i, y_i) = \frac{1}{n} \sum_{j}^{n} \mathop{\mathbb{E}}_{\lambda} [\delta(\tilde{x} = \lambda \cdot x_i + (1 - \lambda) \cdot x_j, \tilde{y} = \lambda \cdot y_i + (1 - \lambda) \cdot y_j)],$$

where $\lambda \in (0, 1)$ is drawn from random distributions.

IV. Methodology

In this section, we first formally introduce the MIA problem for generative models, and then present our defense against the attack with discussions.

A. Problem Statement

Generative models are often used to generate synthetic data following the same distribution with the input training data, as a complement in cases where the training data is scarce, or as a replacement for the training data with privacy concerns. Done training, the trained generative models or its output generated data are released to perform any downstream tasks. However, it has been found that the generative model would memorize sensitive information of the training data, or the generated output would directly leak the private training data, particularly through membership inference attacks. Hence we aim to mitigate such threats without degrading the quality of the synthetic data. At its core, our design has two objectives, *i.e.,* maintaining the utility of the generated data, and preserving the privacy of the training data.

Utility. In the wide range of downstream tasks, we mainly consider two representative ones: 1) The generated data is used for training supervised machine learning models, and the quality of the generated data is evaluated by the testing accuracy of the model trained. 2) The generated data is directly used, and such data is evaluated by the reconstruction quality, *i.e.,* Frechet Inception Distance (FID) for images to measure the visual distance, or Dimensional Wise Probability (DWP) [14] for tabular data to measure the probability similarity between the original training data and the generated one. The higher the scores, the better the quality of the generated data.

Privacy. As MIA is considered as the major threat, our privacy goal is set to reduce the success rate of MIA on the target model. Specifically, a MIA aims to figure out whether a given sample has been used to train the target model. Let D_{train} be the training set of the target model M, and D_{train} is drawn from an underlying distribution D. D_{adv} represents an adversary-accessible dataset drawn from the same underlying distribution D. With the aid of D_{adv}, the adversary is able to query model M by the query function $Q(M, x)$ on sample

x. The goal of the adversary is to estimate $\Pr(x \in D_{train})$. The adversary wins if it successfully determines whether $x \in D_{train}$ or not. Since MIA outputs binary decisions concerning the given samples, and the ground truth of the samples may be scarce, we use Area Under ROC Curve (AUCROC) as an indicator of attack success, following the convention of [5].

B. Mixup as a MIA Defense

As most of the MIAs are based on the overfitting phenomenon of training examples, many defense methods are proposed to avoid overfitting in training. Different from those approaches, we propose to avoid direct training on the original data to defend MIAs. Instead of training on the raw data, we let the model be trained in a $mixup$ [9] fashion — only linear combination of the training data is seen by the model. Hence the privacy of the original training data is preserved. On the other hand, $mixup$ can largely maintain utility for the downstream tasks.

Specifically, we replace the original training data by their linear combinations where the coefficients are randomly sampled from a beta probability distribution. The probability distribution function (PDF) of the beta distribution, for $0 \leq x \leq 1$ and shape parameters $\alpha, \beta > 0$ is:

$$f(x; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

where B is a normalization function to ensure the total probability is 1. Letting the labeled datasets include the data-label pair (x, y), we randomly sample two pairs (x_i, y_i), (x_j, y_j) from the training set and apply the same sampled coefficient λ to both x and y. Our generative model is a conditional one with labels. Thus the generator is fed a random prior z and a mixed label y_{mix} to produce synthetic examples. At the same time, the discriminator learns to distinguish a mixed example from a fake one. At the end of training, the algorithm outputs a generator model G, or a synthetic dataset $\tilde{\mathcal{D}}$ generated by G. The detail of the algorithm is provided in Alg. 1.

It should be noted that mixup training works well for datasets which do not exhibit approximate collinearity between points of different classes, but it may fail when collinearity occurs [15], causing a significant degradation in data utility. Fortunately, collinearity is rare indeed in real datasets [15].

Algorithm 1: Mixup defense

Input: Training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, $y_i \in \mathcal{Y}$, generator update interval n_g, training iterations n, beta distribution parameter α, learning rate lr_G and lr_D, prior distribution P_z.

Output: Generator G, dataset to release $\tilde{\mathcal{D}}$

$batch\_done = 0$;
randomly initialize the parameters of the generator G and the discriminator D as θ_G, θ_D;
while $batch\_done < n$ **do**
  `/* generate linear combinations */`
  randomly sample $(x_1, y_1), (x_2, y_2)$ from $\mathcal{D}$;
  sample $\lambda \sim \beta(\alpha, \alpha)$;
  $x_{mix} = \lambda x_1 + (1-\lambda) x_2$;
  $y_1 = one\_hot(y_1)$;
  $y_2 = one\_hot(y_2)$;
  $y_{mix} = \lambda y_1 + (1-\lambda) y_2$;
  `/* generate fake samples */`
  sample $z \sim P_z$;
  $fake = G(z, y_{mix})$
  `/* update D */`
  $L_D = -D(x_{mix}, y_{mix}) + D(fake, y_{mix})$;
  $\theta_D = \theta_D - lr_D \cdot \nabla_{\theta_D} L_D$;
  $batch\_done = batch\_done + 1$;
  `/* update G */`
  if $batch\_done \mod n_g == 0$ **then**
    $L_G = -D(fake, y_{mix})$;
    $\theta_G = \theta_G - lr_G \cdot \nabla_{\theta_G} L_G$;
  end
end
$\tilde{\mathcal{D}} \leftarrow \emptyset$;
for $j = 1$ **to** n **do**
  sample $z \sim P_z$;
  sample y from $\mathcal{Y}$;
  $\tilde{\mathcal{D}} = \tilde{\mathcal{D}} \cup \{(G(z, y), y)\}$;
end
return *G*, $\tilde{\mathcal{D}}$

V. Analytical Insights

In this section, we analyze our defense method from the likelihood ratio test perspective, and show how it mitigates the success rate of MIAs. As the likelihood ratio attack (LIRA) is the strongest MIA, we consider a defense method that can successfully defend LIRA is capable of defending other MIAs.

A. A Reduced Ratio for Targeted Members

We first show how the $mixup$ method decreases Λ for targeted members. Denote l as the loss of the target example x. Define Q_{in}, Q_{out} as distributions of losses on x and models trained with and without it. Assume that the distribution of l follows a Gaussian distribution, i.e., $Q_{in} = \mathcal{N}(\mu_{in}, \sigma_{in}^2)$ or $Q_{out} = \mathcal{N}(\mu_{out}, \sigma_{out}^2)$. Hence Λ can be explicitly formulated as:

$$\begin{aligned}\Lambda = \frac{\Pr(l \mid Q_{in})}{\Pr(l \mid Q_{out})} &\propto \exp\left(\frac{(l-\mu_{out})^2}{2\sigma_{out}^2} - \frac{(l-\mu_{in})^2}{2\sigma_{in}^2}\right) \\ &\propto \exp\left[(\sigma_{in}^2 - \sigma_{out}^2)l^2 + 2(\mu_{in}\sigma_{out}^2 - \mu_{out}\sigma_{in}^2)l\right] \\ &\triangleq \exp\left[f(l)\right],\end{aligned} \tag{1}$$

where $f(l)$ is a quadratic function of l. By the first-order condition, we could find the local optimal point at $l_0 = (\mu_{out}\sigma_{in}^2 - \mu_{in}\sigma_{out}^2)/(\sigma_{in}^2 - \sigma_{out}^2)$ when $\sigma_{in}^2 \neq \sigma_{out}^2$. Let l_{orig} be the loss of the target x feeding into a conventionally trained model, and l_{mix} be the loss of the same x feeding into a model trained by the $mixup$ method. Intuitively, as $mixup$ does not optimize directly on the target points, generally we have $l_{mix} > l_{orig}$, as shown in Fig. 1. We compute the reconstruction losses as in [3] of 50 images randomly picked

from CIFAR-10 [16] on 256 primitive GANs and 256 mixup trained GANs. The reconstruction loss is

$$L_r = \min_z f(G(z), x) \tag{2}$$

for GAN (G, D), where f is a distance function. Here we set f as sum of ℓ_2-norm and Learned Perceptual Image Patch Similarity (LPIPS) [17], following [3]. We normalized the reconstruction loss for each image to the range $[0, 1]$ and then put them together to get an average distribution. Fig. 1 is drawn by kernel density estimation implemented by `Scipy` [18].

Similarly, we use kernel density estimation to draw the normalized discriminator losses of 50 records in MIMIC-III [19], shown in Fig. 2. The discriminator loss for our WGAN is defined as

$$L_D = D(G(z)) - D(x) \tag{3}$$

where z is a latent vector and x is a sample in training set. Both z and x are randomly sampled from their corresponding distribution.

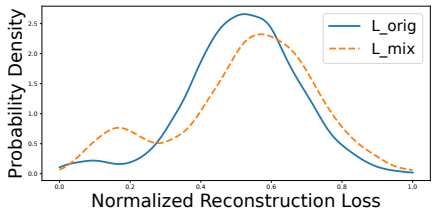


Fig. 1: Reconstruction Loss Distributions on CIFAR-10

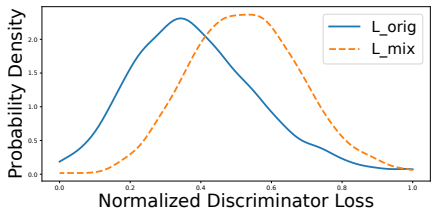


Fig. 2: Discriminator Loss Distributions on MIMIC-III

Proposition 1. *With a probability approximately larger than 0.5, applying mixup to the model training leads to a decrease in Λ for target members.*

Proof. Before the proof, we are aware that for the loss distribution of the reference models, $\mu_{out} > \mu_{in}$ as the loss for the unseen examples are typically larger than that for the seen examples. We will prove this proposition in three cases: (i) $\sigma_{in}^2 - \sigma_{out}^2 > 0$; (ii) $\sigma_{in}^2 - \sigma_{out}^2 < 0$; (iii) $\sigma_{in}^2 - \sigma_{out}^2 = 0$.

(i) In this case, Λ is monotonously decreasing over the region $(-\infty, l_0]$ for the quadratic function of $\Lambda(l)$. Hence we have:

$$l_0 = \frac{\mu_{out}\sigma_{in}^2 - \mu_{in}\sigma_{out}^2}{\sigma_{in}^2 - \sigma_{out}^2} = \mu_{out} + \frac{(\mu_{out} - \mu_{in})\sigma_{out}^2}{\sigma_{in}^2 - \sigma_{out}^2} > \mu_{out}, \tag{4}$$

or equivalently,

$$l_0 = \mu_{in} + (\mu_{out} - \mu_{in})\left(1 + \frac{\sigma_{out}^2}{\sigma_{in}^2 - \sigma_{out}^2}\right) > \mu_{in}. \tag{5}$$

For a target member, l_{orig} follows the distribution $\mathcal{N}(\mu_{in}, \sigma_{in}^2)$. From Eq. (5), we have $\Pr(l_{orig} < l_0) > \Pr(l_{orig} < \mu_{in}) = 0.5$. Given l_{orig} and the symmetric function of $\Lambda(l)$, as long as $l_{mix} \in (l_{orig}, 2l_0 - l_{orig})$, we have $\Lambda(l_{mix}) < \Lambda(l_{orig})$. Hereby, we show the case that $l_{orig} > l_0$ or $2l_0 - l_{orig} < l_{mix}$ happens with small probabilities. For the first case, it is clear that $l_{orig} > l_0 > \mu_{out}$ happens with a small probability since $l_{orig} \sim \mathcal{N}(\mu_{in}, \sigma_{in}^2)$. For the second case, it is less likely for l_{mix} to go beyond μ_{out} which is smaller than $2l_0 - l_{orig}$, since at least a linear combination of the target sample is seen by the model. Hence $2l_0 - l_{orig} < l_{mix}$ takes place with a small probability. Therefore, there is a greater probability (approximately larger than 0.5) of $\Lambda(l_{mix}) < \Lambda(l_{orig})$.

(ii) In this case, $\Lambda(l)$ is monotonously decreasing over the region $[l_0, +\infty)$. Similar to (i), we have:

$$l_0 = \frac{\mu_{out}\sigma_{in}^2 - \mu_{in}\sigma_{out}^2}{\sigma_{in}^2 - \sigma_{out}^2} = \mu_{in} - \frac{(\mu_{out} - \mu_{in})\sigma_{in}^2}{\sigma_{out}^2 - \sigma_{in}^2} < \mu_{in}. \tag{6}$$

Therefore, $\Pr(l_{orig} > l_0) > \Pr(l_{orig} > \mu_{in}) = 0.5$. In the case where $l_{orig} > l_0$, $\Lambda(l_{mix}) < \Lambda(l_{orig})$ as long as $l_{mix} \in (l_{orig}, +\infty)$.

(iii) In this case, $f'(l) = 2(\mu_{in}\sigma_{out}^2 - \mu_{out}\sigma_{in}^2) < 0$, so that $l_{mix} > l_{orig}$ would definitely lead to $\Lambda(l_{mix}) < \Lambda(l_{orig})$. □

It is worth mentioning that without any additional information, $P \triangleq \Pr(\Lambda(l_{mix}) < \Lambda(l_{orig}))$ is related to Q_{in} and Q_{out}. In particular, when the two distributions are close to each other, the value of l_0 is close to μ_{in}, and hence P is close to 0.5. In this case, the target sample x is an inlier, with $\Lambda(l_{orig})$ close to 1. Even if $\Lambda(l_{mix})$ may be larger, its impact is limited. Existing attack mechanisms are good at distinguishing the outliers, not inliers, i.e., $\mu_{out} - \mu_{in}$ is sufficiently large. For those outliers, P will be close to 1. To sum up, *mixup* shrinks the likelihood ratio in the hypothesis test.

B. Lowering Upper Bound of Attack AUC

With analysis from the previous section, we further show how *mixup* lowers the upper bound of AUC of the LIRA which represents the strongest MIA.

Denote TPR as the true positive rate of the likelihood ratio attack and FPR as the false positive rate. Let $\mathcal{P}_m$ be the distribution of Λ for all member examples and $\mathcal{P}_n$ be the distribution of Λ for all non-member examples. According to [20] (see the proof of Remark.A.1 in Appendix A), the upper bound of TPR is given by:

$$\text{TPR} \leq \text{FPR} + \min\{D_{TV}(\mathcal{P}_m, \mathcal{P}_n), 1 - \text{FPR}\}, \tag{7}$$

where $D_{TV}(\mathcal{P}_m, \mathcal{P}_n)$ represents the total variation distance between $\mathcal{P}_m$ and $\mathcal{P}_n$. Since an ROC curve plots TPR vs. FPR and AUC stands for the area under the curve, we can derive the upper bound of likelihood ratio attack AUC as follows:

$$\begin{aligned} \text{AUC} &\leq \frac{1}{2}\left[1 + D_{TV}(\mathcal{P}_m, \mathcal{P}_n)\right]\left[1 - D_{TV}(\mathcal{P}_m, \mathcal{P}_n)\right] + \\ &\quad D_{TV}(\mathcal{P}_m, \mathcal{P}_n) \\ &= -\frac{1}{2}D_{TV}(\mathcal{P}_m, \mathcal{P}_n)^2 + D_{TV}(\mathcal{P}_m, \mathcal{P}_n) + \frac{1}{2}. \end{aligned} \tag{8}$$

Let $\Xi = \log \Lambda$, $\mathcal{Q}_m$ be the distribution of Ξ for all member examples, and $\mathcal{Q}_n$ be the distribution of Ξ for all non-member examples. We theoretically prove for case (iii) in Sec. V-A that, $\mathcal{Q}_m$ and $\mathcal{Q}_n$ follow Gaussian distributions.

Lemma 1. *If $\sigma_{in}^2 = \sigma_{out}^2 = \sigma^2$, $\mathcal{Q}_m$ and $\mathcal{Q}_n$ follow Gaussian distributions.*

Proof. We first prove there is a linear relationship between Ξ and the loss l:

$$\begin{aligned}\Xi = \log \Lambda &= \log\left[\exp\left(\frac{(l-\mu_{out})^2 - (l-\mu_{in})^2}{2\sigma^2}\right)\right] \\ &= \frac{1}{2\sigma^2}\left[2(\mu_{in}-\mu_{out})l + \mu_{out}^2 - \mu_{in}^2\right] = \kappa l + c,\end{aligned} \tag{9}$$

where $\kappa = (\mu_{in} - \mu_{out})/\sigma^2$. Since we have assumed that the distribution of l is Gaussian, the distribution of Ξ is also Gaussian, i.e., $\mathcal{Q}_m$ and $\mathcal{Q}_n$ follow Gaussian distributions. □

Although we cannot prove the same conclusion for case (i) and (ii), we empirically show the conclusion still holds, with the distribution of Ξ depicted in Fig. 3. We perform the likelihood ratio attack to an unprotected GAN trained on 5000 images from CIFAR-10. The query set of the attack is 5000 members and 5000 non-members which is also from CIFAR-10. The distribution resembles a Gaussian one.

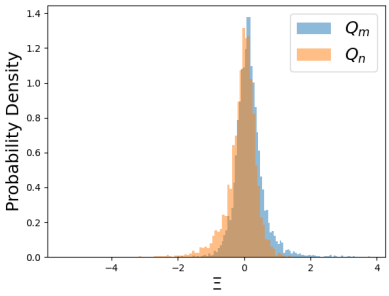


Fig. 3: Distribution of Ξ when $\sigma_{in}^2 - \sigma_{out}^2 \neq 0$.

With $\mathcal{Q}_m$ and $\mathcal{Q}_n$ are Gaussian distributions, we first prove the following lemma:

Lemma 2. *By decreasing Λ for target members, the upper bound of $D_{TV}(\mathcal{Q}_m, \mathcal{Q}_n)$ is reduced.*

Proof. The total variation distance can be upper bounded by Hellinger distance as follows [21]:

$$D_{TV}(\mathcal{Q}_m, \mathcal{Q}_n) \leq \sqrt{2} D_H(\mathcal{Q}_m, \mathcal{Q}_n). \tag{10}$$

We take advantage of the fact that the Hellinger distance between two Gaussian distributions can be written in a closed form. Letting $\mathcal{Q}_m = \mathcal{N}(\mu_m, \sigma_m^2)$, $\mathcal{Q}_n = \mathcal{N}(\mu_n, \sigma_n^2)$, we have:

$$D_H(\mathcal{Q}_m, \mathcal{Q}_n)^2 = 1 - \sqrt{\frac{2\sigma_m\sigma_n}{\sigma_m^2+\sigma_n^2}}\exp\left(-\frac{1}{4}\frac{(\mu_m-\mu_n)^2}{\sigma_m^2+\sigma_n^2}\right). \tag{11}$$

According to Prop. 1, since $mixup$ lowers the value of Λ for target members compared to that of conventional training, the mean of Λ — μ_m decreases correspondingly. It should be noted that μ_m is generally greater than μ_n, which is the foundation of likelihood ratio attacks, as a targeted member usually has a larger conditional probability $\Pr(l \mid Q_{in})$, and thus Λ for member examples tend to be larger than those for non-member examples. Therefore, $(\mu_m - \mu_n)^2$ is reduced, and hence the Hellinger distance between $\mathcal{Q}_m$ and $\mathcal{Q}_n$ is shortened, resulting in a reduced upper bound of $D_{TV}(\mathcal{Q}_m, \mathcal{Q}_n)$. □

With the above lemma, we now can illustrate how $mixup$ reduces the upper bound of the AUC for the LIRA:

Theorem 1. *A decrease in $\mathbb{E}[\mathcal{P}_m]$ reduces the upper bound of AUC for likelihood ratio attacks, where $\mathbb{E}[\cdot]$ denotes the expectation.*

Proof. First, let f_m, f_n, g_m, g_n denote the probability density functions of $\mathcal{Q}_m$, $\mathcal{Q}_n$, $\mathcal{P}_m$ and $\mathcal{P}_n$, respectively. Then we have:

$$g_m(\Lambda) = \frac{1}{\Lambda} f_m(\log\Lambda), \;\; g_n(\Lambda) = \frac{1}{\Lambda} f_n(\log\Lambda). \tag{12}$$

Note that the Hellinger distance and total variation distance are both special forms of F-divergence. The F-divergence between distributions $\mathcal{Q}_m$ and $\mathcal{Q}_m$ can be formulated as:

$$D_f(\mathcal{Q}_m, \mathcal{Q}_n) = \int f_n(\Xi) h\left(\frac{f_m(\Xi)}{f_n(\Xi)}\right) d\Xi, \tag{13}$$

where $h(x)$ is a convex function with $h(1) = 0$. For Hellinger distance, $h(x) = (\sqrt{x} - 1)^2$.

$$\begin{aligned} D_f(\mathcal{P}_m, \mathcal{P}_n) &= \int g_n(\Lambda) h\left(\frac{g_m(\Lambda)}{g_n(\Lambda)}\right) d\Lambda \\ &= \int \frac{1}{\Lambda} f_n(\log\Lambda) h\left(\frac{f_m(\log\Lambda)}{f_n(\log\Lambda)}\right) d\Lambda \\ (d\Lambda = \Lambda d\Xi) \Rightarrow \\ &= \int f_n(\Xi) h\left(\frac{f_m(\Xi)}{f_n(\Xi)}\right) d\Xi \\ &= D_f(\mathcal{Q}_m, \mathcal{Q}_n). \end{aligned} \tag{14}$$

Therefore, a decrease in the Hellinger distance between $\mathcal{Q}_m$ and $\mathcal{Q}_n$ will cause the same decrease in that between $\mathcal{P}_m$ and $\mathcal{P}_n$. From Lemma 2, we can conclude that a decreasing Λ for target members can reduce the upper bound of $D_{TV}(\mathcal{P}_m, \mathcal{P}_n)$. Since the upper bound of LIRA AUC is positively correlated with $D_{TV}(\mathcal{P}_m, \mathcal{P}_n)$, AUC is successfully reduced. As a result, applying $mixup$ to model training can be an effective defense against membership inference attacks. □

VI. Experiment and Evaluation

We aim to answer the following research questions in the experiments:

Q1: Is the likelihood ratio attack the strongest MIA?
Q2: How is the privacy performance of $mixup$ compared to baselines?
Q3: How is the utility performance of $mixup$ compared to baselines?
Q4: Is $mixup$ adaptive-attack-proof?

A. Setup

We implement all experiments using `PyTorch 1.11.0` and run them on NVIDIA GeForce RTX 3090 GPUs.

Datasets. We choose two representative types of data — images and tabular data, for the datasets in generative machine learning tasks. For images, the widely adopted CelebA [22], CIFAR-10 [16] are chosen, and MIMIC-III [19] is selected as the tabular dataset. CelebA contains more than 200,000 celebrity images with 40 binary facial attributes. We use the officially preprocessed CelebA dataset with face alignment, center-crop and resize each image to 64×64 before the GAN training and MIA query. The task on the generated data of CelebA is the attribute classification similar to [8], and the binary attribute gender is chosen. CIFAR-10 consists of 60,000 32×32 colour images in 10 classes. The task on the generated dataset of CIFAR-10 is set to be the 10-category image classification. MIMIC-III is a public Electronic Health Record (EHR) database composed by 46,520 medical records of intensive care unit (ICU) patients. We follow the same procedure as in [14] and [3] to pre-process the data, where each patient is represented by a 1071-dimensional binary feature vector. The task on the generated dataset is a dimensional wise prediction using logistic regression for all attributes.

Metrics and models. Since membership inference is a binary classification task, we use AUC of the MIA attackers' ROC as the privacy metric. We plot both the regular ROC and the ROC in log-log graph as suggested in [5]. For utility, we use the prediction accuracy (for multi-class classification) or AUC (for binary classification) to evaluate the downstream tasks. To directly evaluate the quality of the generated data, we adopt Frechet Inception Distance (FID) [23] between the generated images and the real images, and the dimensional wise probability (DWP) [14] and dimensional-wise prediction (DWpre) [14] for tabular data.

For the target GAN to be trained and attacked, we use WGANGP [24], considering its pleasing performance on data generation. For the downstream classification task, we choose ResNet-18 [25] for image data, and a linear classifier (logistic regression) for tabular data following the setup of [14]. For both the conventional and mixup training, we adopt the following hyperparameters by default: epoch number 200, batch size 100, number of training samples 5000. The Adam optimizer is used for learning.

Attacks. For the evaluation of privacy, we launch the following MIAs to generative models:

GAN-leaks [3]. We adopt the white-box generator version of GAN-leaks, which optimizes the input of generator to find the closest reconstruction of the target sample, and use the distance between the target sample and its reconstruction result to infer the target's membership. The same parameters are reused as with [3].

Logan [4]. We use the accessible discriminator version of Logan, which directly uses the output of the discriminator with the target sample as input to infer its membership.

Likelihood ratio attack [5]. As the original LIRA is designed to attack discriminative models, we adapt the attack to our target GAN model. We implement two versions: the first one takes the generator reconstruction loss distributions in the hypothesis test, and the other uses the discriminator loss distributions in the test. We replace the logits in the LIRA of [5] with those losses. In each attack, 512 reference models are trained to estimate Q_{in} and Q_{out}.

Baselines. The state-of-the-art defense methods to MIAs on generative models include PAR-GAN [7] and relaxLoss [26], which we use as comparison baselines. As PAR-GAN is claimed to be stronger than PrivGAN [6], we omit the latter in our experiments.

RelaxLoss is originally designed for discriminative models, which takes gradient ascent when the loss is lower than a preset threshold to prevent overfitting. We adapt it into the GAN scenario by applying gradient ascent when the discriminator loss $L_D = D(G(z)) - D(x)$ is lower than a threshold $L_D^\star$ which is calculated by reference models. Specifically, we update the discriminator by:

$$\omega_D = \omega_D + \text{lr} \cdot \frac{\partial L_D}{\partial \omega_D} \tag{15}$$

where lr is the learning rate. The reference models are trained on disjoint but identically distributed data. $L_D^\star$ is the average loss of the discriminators of reference models.

PAR-GAN is particularly designed for generative models. PAR-GAN constructs one generator and k discriminators, and divides the whole training set into k parts, each part used as the training set of each discriminator. The generator is adversarially trained against all k discriminators. In our experiments, the number of discriminators is set to be 4 in most cases, referring to the experimental settings of [7].

B. Privacy Performance

We evaluate the attack performance of Logan, GAN-leaks, and LIRA on the target GAN. The results are given in Fig. 4, showing LIRA outperforms Logan and GAN-leaks, in particular at low FPRs (**Q1**). We depict both the regular ROC and the log-log ROC, and find the latter clearer in displaying the low FPR region. Due to space constraints, we only report log-log ROCs below. Meanwhile, we compare defense methods mixup, relaxLoss, PAR-GAN, and the unprotected baseline against the aforementioned attacks. The attack ROCs are displayed in Fig. 5 and 6. Table I, Table II and Table III show the corresponding AUCs. It is clear that $mixup$ is most successful in defending against all MIAs in consideration (**Q2**).

In the experiments, we observe that PAR-GAN performs even worse than unprotected case on CIFAR-10 (Fig. 5(a)). According to [7], PAR-GAN should be able to defend against Logan attack. We compare the experimental settings and find out that the only difference lies in the GAN structure: we use WGANGP [24] with conditions while [7] uses an unconditional one. Since in conditional WGAN, the score of sample x is calculated by score $= D(x, y)$ where y

is the condition. Such a structure might lead to PAR-GAN being prone to overfitting as the size of training dataset of a discriminator in PAR-GAN is $1/k$ of the unprotected case. Hence PAR-GAN may not be suitable for conditional GAN scenarios.

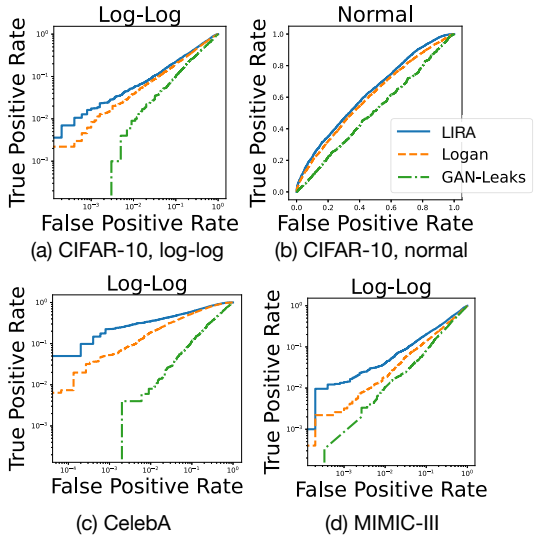


Fig. 4: Attack performance of Logan, GAN-leaks and LIRA against unprotected GAN. Legends are shared.

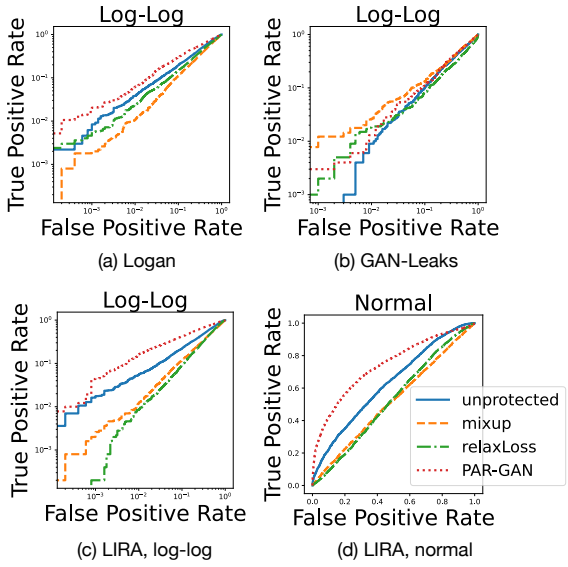


Fig. 5: ROCs of Logan, GAN-Leaks, and LIRA against unprotected, mixup, relaxLoss and PAR-GAN on CIFAR-10. Legends are shared.

TABLE I: Attack AUCROC on CIFAR-10.

	Logan	Ratio	GAN-Leaks
unprotected	0.6435	0.6866	**0.5083**
mixup	**0.5202**	**0.5303**	0.5312
relaxLoss	0.5478	0.5326	0.4197
PAR-GAN	0.6668	0.7398	0.5291

The defense performance of RelaxLoss is not stable: as Fig. 6(b) shows, RelaxLoss sometimes fails to defend Logan. This is because the training procedure of RelaxLoss contains comparison with reference models which are also used in LIRA, and thus RelaxLoss is most likely to defend LIRA but not others.

GAN-Leaks fails to attack the unprotected GAN on CelebA and MIMIC-III (Fig. 4(c)(d)), so we do not apply the attack to any defense approach, and thus omit them from Table II, III. It is reasonable that GAN-Leaks has worse performance than Logan and LIRA, since attacker's prior knowledge is less in GAN-Leaks, which only uses the white-box generator while Logan and LIRA have access to both generators and discriminators. The attack failure on MIMIC-III may also be caused by the lower dimension of the target tabular data compared with image data. GAN-Leaks depends on the assumption that training samples are easier to reconstruct than the holdout data. But if the target data are of lower dimensions, they are easy to reconstruct no matter they are in the training set or not.

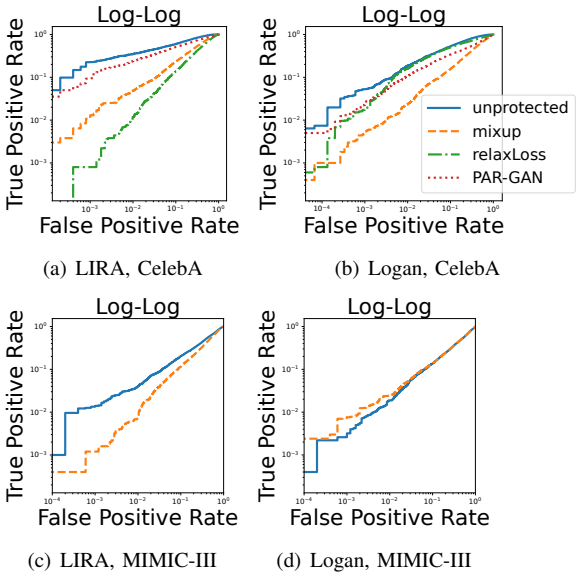


Fig. 6: ROCs of Logan, GAN-Leaks, and LIRA against unprotected, mixup, relaxLoss and PAR-GAN on CelebA and MIMIC-III.

TABLE II: Attack AUCROC on CelebA

	Logan	Ratio	GAN-Leaks
unprotected	0.8346	0.8637	0.5317
mixup	**0.5788**	0.6615	-
relaxLoss	0.7703	**0.5857**	-
PAR-GAN	0.6571	0.7781	-

C. Utility Performance

We test the utility performance of mixup training and baselines, shown in Table IV, to answer **Q3**. On both CIFAR-10 and CelebA, there is no significant decrease in downstream tasks with $mixup$ compared to the unprotected case, while

TABLE III: Attack AUCROC on MIMIC-III

	Logan	Ratio	GAN-Leaks
unprotected	0.5264	0.5913	0.5028
mixup	0.5269	0.5283	-
relaxLoss	0.5296	0.5175	-
PAR-GAN	0.5350	0.5015	-

RelaxLoss incurs a significant loss in classification accuracy. It needs to be explained that due to limited training data (5000) is used to launch MIA, the accuracy of CIFAR-10 is not as high as using the entire training set (0.95 in our experiment, matching the SOTA), which matches the results in [6].

TABLE IV: Downstream classification accuracy and FID on Images datasets.

(a) CIFAR-10

Protection	Acc	FID
unprotected	0.490	150.944
mixup	0.421	159.098
relaxLoss	0.385	102.955
PAR-GAN	0.404	199.053

(b) CelebA-Gender

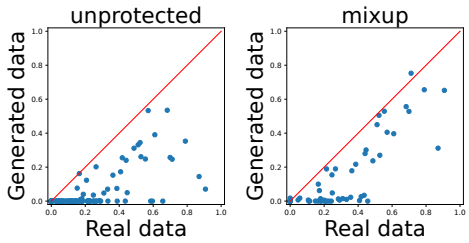
Protection	Acc	FID
unprotected	0.912	111.980
mixup	0.915	104.376
relaxLoss	0.836	97.746
PAR-GAN	0.876	157.724

As we observe from Table IV, RelaxLoss suffer from severe utility loss. This is because the method does not allow the target GAN to reach a lower loss than the average of the reference models. And the original work [26] compensates such utility loss by posterior flattening, which unfortunately cannot be applied to GAN. But surprisingly, the FID of RelaxLoss is significantly lower than other defense methods, which contradicts the downstream accuracy. It indicates that the distribution of the images generated by the RelaxLoss GAN is closer to real images, but the degree of compliance with the input condition is lower than other defenses.

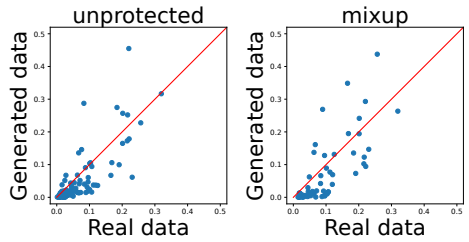
For tabular data MIMIC-III, we use dimensional wise prediction (DWpre) and dimensional wise probability (DWP) to compare the data utility of $mixup$ with the unprotected case. The results are shown in Fig. 7. It can be observed that $mixup$ has a similar utility performance with the unprotected case.

D. Adaptive Attack

We investigate the situation where the attacker is aware of our defense strategy and tries to break it by corresponding attacks (**Q4**). For LIRA, an adaptive attacker could use $mixup$ to train the reference models and calculate the ratio based on these reference models. We also consider a more powerful attacker who is aware of the co-membership [27], *i.e.,* a group of examples are all belong to the training set, or none of them are. In that case, the attacker uses the linear combinations (mixed) of the target samples for MIA query. Therefore, we test the defense performance of $mixup$ under four types of adaptive attacks: the reference models are trained naturally or using $mixup$, and the query examples are a single sample or mixed co-membership samples. The attack AUCs are given in Table V and Table VI. We can see that the strongest adaptive attack is the one using naturally trained reference models and mixed samples for co-membership query. Naturally trained GAN has a clearer distinction between Q_{in} and Q_{out} mentioned in section V-A which gives a better likelihood ratio result. And mixed samples contain the co-membership information which also helps the attacker. Even in that case, $mixup$ still brings a significant privacy gain compared with LIRA against unprotected GAN.



(a) DWpre F1-score of the logistic regression trained on real and generated data



(b) DWP, $\Pr(x_i = 1)$ for each valid i

Fig. 7: DWpre and DWP results on MIMIC-III

TABLE V: Adaptive attack AUCs against $mixup$ on CIFAR-10. The original LIRA against non-protected target GAN has an AUC of 0.6866.

query \ ref. models	mixup trained	naturally trained
mixed query	0.5264	0.6084
single query	0.5426	0.5303

TABLE VI: Adaptive attack AUCs against $mixup$ on CelebA. The original LIRA against non-protected target GAN has an AUC of 0.8637.

query \ ref. models	mixup trained	naturally trained
mixed query	0.5975	0.7283
single query	0.6701	0.6615

VII. Conclusion

The work proposes an effective defense approach, mixup training, against membership inference attack on generative models. By replacing the training data with their interpolations, $mixup$ prevents overfitting to the original training data. Theoretical analysis reveals that $mixup$ indeed mitigates the AUC of the strongest MIA. Experimental results also support that our defense is superior to other baseline defenses, showing great promise to preserve data privacy while maintaining model utility.

REFERENCES

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

[2] A. Hu, R. Xie, Z. Lu, A. Hu, and M. Xue, "Tablegan-mca: Evaluating membership collisions of gan-synthesized tabular data releasing," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 2096–2112.

[3] D. Chen, N. Yu, Y. Zhang, and M. Fritz, "Gan-leaks: A taxonomy of membership inference attacks against generative models," in *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, 2020, pp. 343–362.

[4] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "Logan: Membership inference attacks against generative models," in *Proceedings on Privacy Enhancing Technologies (PoPETs)*, vol. 2019, no. 1. De Gruyter, 2019, pp. 133–152.

[5] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2022, pp. 1519–1519.

[6] S. Mukherjee, Y. Xu, A. Trivedi, and J. L. Ferres, "privgan: Protecting gans from membership inference attacks at low cost to utility," *Proceedings on Privacy Enhancing Technologies*, vol. 2021, no. 3, pp. 142–163, 2021. [Online]. Available: https://doi.org/10.2478/popets-2021-0041

[7] J. Chen, W. H. Wang, H. Gao, and X. Shi, "Par-gan: Improving the generalization of generative adversarial networks against membership inference attacks," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 127–137.

[8] B. Wang, F. Wu, Y. Long, L. Rimanic, C. Zhang, and B. Li, "Datalens: Scalable privacy preserving training via gradient compression and aggregation," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 2146–2168.

[9] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

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

[11] B. Hilprecht, M. Härterich, and D. Bernau, "Monte carlo and reconstruction membership inference attacks against generative models." *Proc. Priv. Enhancing Technol.*, vol. 2019, no. 4, pp. 232–249, 2019.

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

[13] J. Neyman and E. S. 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*, vol. 231, no. 694-706, pp. 289–337, 1933.

[14] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating multi-label discrete patient records using generative adversarial networks," in *Machine learning for healthcare conference*. PMLR, 2017, pp. 286–305.

[15] M. Chidambaram, X. Wang, Y. Hu, C. Wu, and R. Ge, "Towards understanding the data dependency of mixup-style training," in *International Conference on Learning Representations*, 2021.

[16] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[17] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[18] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[19] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[20] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," in *International conference on machine learning*. PMLR, 2015, pp. 1376–1385.

[21] T. Steerneman, "On the total variation and hellinger distance between signed measures; an application to product measures," *Proceedings of the American Mathematical Society*, vol. 88, no. 4, pp. 684–688, 1983.

[22] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[23] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

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

[26] D. Chen, N. Yu, and M. Fritz, "Relaxloss: Defending membership inference attacks without losing utility," in *International Conference on Learning Representations*, 2021.

[27] K. S. Liu, C. Xiao, B. Li, and J. Gao, "Performing co-membership attacks against deep generative models," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 459–467.