Pseudo Label-Guided Model Inversion Attack via Conditional Generative Adversarial Network

Xiaojian Yuan[1], **Kejiang Chen**[*1], **Jie Zhang**[1,2], **Weiming Zhang**[1],
Nenghai Yu[1], **Yang Zhang**[3]
[1]University of Science and Technology of China, [2]University of Waterloo,
[3]CISPA Helmholtz Center for Information Security
xjyuan@mail.ustc.edu.cn, chenkj@ustc.edu.cn, jiezhangsp@gmail.com,
zhangwm@ustc.edu.cn, ynh@ustc.edu.cn, zhang@cispa.de

Abstract

Model inversion (MI) attacks have raised increasing concerns about privacy, which can reconstruct training data from public models. Indeed, MI attacks can be formalized as an optimization problem that seeks private data in a certain space. Recent MI attacks leverage a generative adversarial network (GAN) as an image prior to narrow the search space, and can successfully reconstruct even the high-dimensional data (e.g., face images). However, these generative MI attacks do not fully exploit the potential capabilities of the target model, still leading to a vague and coupled search space, i.e., different classes of images are coupled in the search space. Besides, the widely used cross-entropy loss in these attacks suffers from gradient vanishing. To address these problems, we propose Pseudo Label-Guided MI (PLG-MI) attack via conditional GAN (cGAN). At first, a top-*n* selection strategy is proposed to provide pseudo-labels for public data, and use pseudo-labels to guide the training of the cGAN. In this way, the search space is decoupled for different classes of images. Then a max-margin loss is introduced to improve the search process on the subspace of a target class. Extensive experiments demonstrate that our PLG-MI attack significantly improves the attack success rate and visual quality for various datasets and models, notably, $2 \sim 3\times$ better than state-of-the-art attacks under large distributional shifts. Our code is available at: *https://github.com/LetheSec/PLG-MI-Attack*.

1 Introduction

Deep neural networks (DNNs) have revolutionized a wide variety of tasks, including computer vision, natural language processing, and healthcare. However, many practical applications of DNNs require training on private or sensitive datasets, such as facial recognition (Taigman et al. 2014) and medical diagnosis (Rajpurkar et al. 2017), which may pose some privacy threats. Indeed, the prior study of privacy attacks has demonstrated the possibility of exposing unauthorized information from access to a model (Shokri et al. 2017; Gopinath et al. 2019; Tramèr et al. 2016; Fredrikson, Jha, and Ristenpart 2015). In this paper, we mainly focus on model inversion (MI) attacks, a type of privacy attack that aims to recover the training data given a trained model.

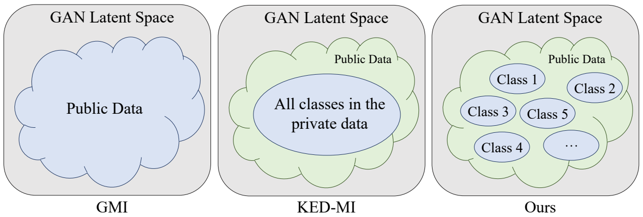


Figure 1: Latent search space for different MI attacks. The blue area represents the latent space that needs to be searched to reconstruct a certain class of images. That is, the adversary should find an optimal latent point in the blue area, so that the generator outputs the private image of a specified class.

Typically, MI attacks can be formalized as an optimization problem with the goal of searching the input space for the sensitive feature value that achieves the highest likelihood under the target model. However, when attacking DNNs trained on more complex data (e.g., RGB images), directly solving the optimization problem via gradient descent tends to stuck in local minima, resulting in reconstructed images lacking clear semantic information. Recent work (Zhang et al. 2020) proposed generative MI attacks (GMI), which used a generative adversarial network (GAN) to learn a generic prior of natural images, avoiding reconstructing private data directly from the unconstrained space. Generally, generative MI attacks can be summarized as the following two search stages:

- **stage-1**: Generator Parameter Space Search. The adversary trains a generative model (i.e., searches for the optimal parameters) on a public dataset that only shares structural similarity with the private dataset.
- **stage-2**: Latent Vector Search. The adversary keeps searching the latent space of the generator trained in stage-1, until the output is close to the images in the private dataset.

Notably, GMI totally ignored the potential capability of the target model for the training process. Inspired by semi-supervised GAN (Salimans et al. 2016), KED-MI (Chen et al. 2021) adopted a classifier as the GAN discriminator and utilized the target model to provide soft labels for public

[*]Corresponding authors.

data during the training process in stage-1, which achieved the state-of-the-art MI attack performance. Even so, the attack performance is still underwhelming, especially when public and private data have a large distributional shift. We infer possible limitations of existing work as follows:

1) **Class-Coupled Latent Space.** The generator obtained by existing work in stage-1 is class-coupled. When the adversary reconstructs a specified class of target in stage-2, it needs to search in the latent space of all classes, which easily causes confusion of feature information between different classes (see Fig. 1).
2) **Indirectly Constrains on the Generator.** The KED-MI attack adopts the semi-supervised GAN framework, which indirectly constrains the generator with class information through the discriminator. However, this implicit constraint relies too much on the discriminator and lacks task specificity for MI.
3) **Gradient Vanishing Problem.** Previous MI attacks have commonly adopted cross-entropy (CE) loss as the optimization goal in stage-2. However, the cross-entropy loss will cause the gradient to decrease and tend to vanish as the number of iterations increases, resulting in the search process to slow down or even stop early.

To address the above limitations, we propose a novel pseudo label-guided MI (PLG-MI) attack. Specifically, we first propose a simple but effective top-*n* selection strategy, which can provide pseudo-labels for public data. Then we introduce the conditional GAN (cGAN) to MI attacks and use the pseudo-labels to guide the training process, enabling it to learn more specific and independent image distributions for each class. In addition, we also impose a task-specific explicit constraint directly on the generator so that class information can be embedded into the latent space. This constraint can force images to be generated towards specific classes in the private data. As shown in Fig. 1, it can be considered as an approximate division of the GAN latent space into separate class subspaces. When reconstructing the private data of an arbitrary class in stage-2, only the corresponding subspace needs to be searched, which avoids confusion between different classes.

Our contributions can be summarized as follows:

- We propose Pseudo Label-Guided MI (PLG-MI) attack, which can make full use of the target model and leverage pseudo-labels to guide the output of the generator during the training process.
- We propose a simple but effective strategy to provide public data with pseudo-labels, which can provide corresponding features according to specific classes in the private dataset.
- We demonstrate the gradient vanishing problem of cross-entropy loss commonly adopted in previous MI attacks and use max-margin loss to mitigate it.
- Extensive experiments demonstrate that the PLG-MI attack greatly boosts the MI attack and achieves state-of-the-art attack performance, especially in the presence of a large distributional shift between public and private data.

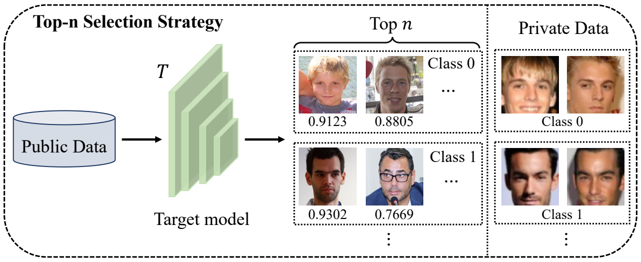


Figure 2: Top-*n* selection strategy. Input the images of public data into the target model and select the top n images with the highest confidence for each class. The right half shows images of the corresponding class in the private dataset.

2 Related Work

Basic Model Inversion Attacks. Fredrikson *et al.* (Fredrikson et al. 2014) first studied MI attacks in the context of genomic privacy and demonstrated that access to linear regression models for personalized medicine can be abused to recover private genomic properties of individuals in the training dataset. Fredrikson *et al.* (Fredrikson, Jha, and Ristenpart 2015) later proposed an optimization algorithm based on gradient descent for MI attacks, which can recover grayscale face images from shallow networks. However, these basic MI attacks that reconstruct private data directly from the pixel space failed when the target models are DNNs.

Generative Model Inversion Attacks. To make it possible to launch MI attacks against DNNs, Zhang *et al.* (Zhang et al. 2020) proposed generative model inversion (GMI) attacks, which trained a GAN on public data as an image prior and then restricted the optimization problem in the latent space of the generator. Wang *et al.* (Wang et al. 2021) proposed viewing MI attacks as a variational inference (VI) problem and provided a framework using deep normalizing flows in the extended latent space of a StyleGAN (Karras et al. 2020). Chen *et al.* (Chen et al. 2021) adopted the semi-supervised GAN framework to improve the training process by including soft labels produced by the target model. Kahla *et al.* (Kahla et al. 2022) extended generative MI attacks to black-box scenarios where only hard labels are available.

3 Method

In this section, we will first discuss the threat model and then present our attack method in detail.

3.1 Threat Model

Adversary's Goal. Given a target model $T : [0,1]^d \rightarrow \mathbb{R}^{|C|}$ and an arbitrary class $c^* \in C$, the adversary aims to reconstruct a representative sample x^* of the training data of the class c^*; d represents the dimension of the model input; C denotes the set of all class labels of the training data and $|C|$ is the size of the label set. It should be emphasized that the reconstructed data need to have good semantic information for human recognition. In this paper, we focus on

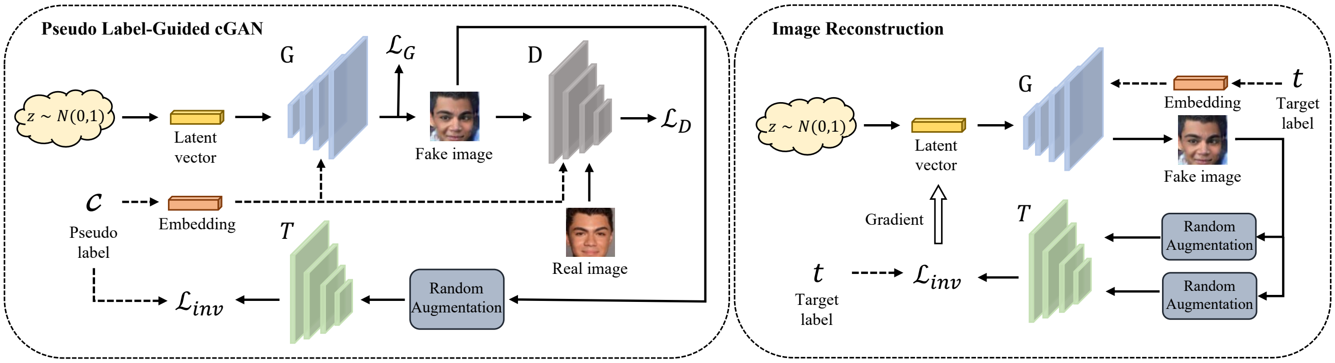


Figure 3: The overall pipeline of the proposed two-stage model inversion attack algorithm. Stage 1: Train a conditional GAN on the public data with guidance provided by pseudo-labels and knowledge of the target model. Stage 2: Leverage the trained generator to reconstruct the specific class of private images using $\mathcal{L}_{inv}$.

the attack against face recognition models, that is, the adversary's goal is to reconstruct its corresponding face image from the target model according to the specified identity.

Adversary's Knowledge. In this paper, we focus on white-box MI attacks, which means that the adversary can have access to all parameters of the target model. In addition, following the settings in previous work (Zhang et al. 2020; Chen et al. 2021; Kahla et al. 2022), the adversary can gain a public dataset of the target task that only shares structural similarity with the private dataset without any intersecting classes. For example, the adversary knows the target model is for face recognition, he can easily leverage an existing open-sourced face dataset or crawl face images from the Internet as the public data.

3.2 Pseudo-Labels Generation

Top-*n* Selection Strategy. In order to make the public data have pseudo-labels to guide the training of the generator, we propose a top-*n* selection strategy, as shown in Fig. 2. This strategy aims to select the best matching n images for each pseudo-label from public data. These pseudo-labels correspond to classes in the private dataset. Specifically, we feed all images in public data into the target model and get the corresponding prediction vectors. Then for a certain class k, we sort all images in descending order of k_{th} value in their prediction vectors and select the top n images to assign the pseudo-label k.

Generally, if the target model has high confidence in the k_{th} class for a certain image, this image can be considered to contain discriminative features of the k_{th} class. We define F^k_{pri} and F^k_{pub} as the distributions of discriminative features contained in the k_{th} class of private data and pseudo-labeled public data. It can be inferred that F^k_{pub} and F^k_{pri} have intersection, which means that the adversary can obtain the required information by sufficiently searching F^k_{pub}, making the private information in F^k_{pri} leaked more easily and accurately.

Narrow the Search Space via Pseudo-Labels. This strategy can narrow the search space of latent vectors in stage-2. Specifically, after reclassifying the public data, we can directly learn the feature distribution of images for each class. When reconstructing images of the k_{th} class in the private dataset, it is only necessary to search for required features in F^k_{pub}, while reducing the interference of irrelevant features from $F^{i\neq k}_{pub}$. Taking face recognition as an example, suppose that the k_{th} class of the private dataset is a young white man with blond hair, then the k_{th} class of the pseudo-labeled public data is also mostly white people with blond hair, as shown in Fig. 2. Consequently, the key features can be preserved and the useless features (e.g., other skin tones or hair colors, etc.) are eliminated, thereby narrowing the search space.

3.3 Pseudo Label-Guided MI Attack

An overview of our attack is illustrated in Fig. 3, which consists of two stages. In stage-1, we train a conditional GAN on public data under the guidance of pseudo-labels. In stage-2, we use the trained generator to reconstruct private images of specified classes.

Problem Formulation. At first, we formulate the MI problem in the context of image classification (i.e., face recognition) with DNNs. We use $\mathcal{D}_s$ to denote the private dataset with sensitive information and $\mathcal{D}_p$ to denote the public dataset available to the adversary. Then using the top-*n* selection strategy to obtain a pseudo-labeled public dataset $\mathcal{D}_r$. We denote a sample image as $x \in \mathcal{D}_s$, and its corresponding label as $y \in \{1, \ldots, K\}$, where K denotes the number of classes. Note that the original $\mathcal{D}_p$ does not have any class intersection with $\mathcal{D}_s$, while the $\mathcal{D}_r$ has the pseudo-labels $\tilde{y} \in \{1, \ldots, K\}$. In the typical case, the target model T will be trained on $\mathcal{D}_s$ to learn the mapping from the input space to the probability vectors.

In generative MI attacks, the adversary uses $\mathcal{D}_p$ to train a GAN and then optimizes the input to the generator, instead of directly optimizing from the pixel space. Denote the trained generator by $G(z)$, where $z \sim \mathcal{N}(0, 1)$ is the latent vector. The optimization problem can be formulated as follows:

$$z^* = \arg\min_{\hat{z}} \mathcal{L}_{inv}(T(G(\hat{z})), c), \tag{1}$$

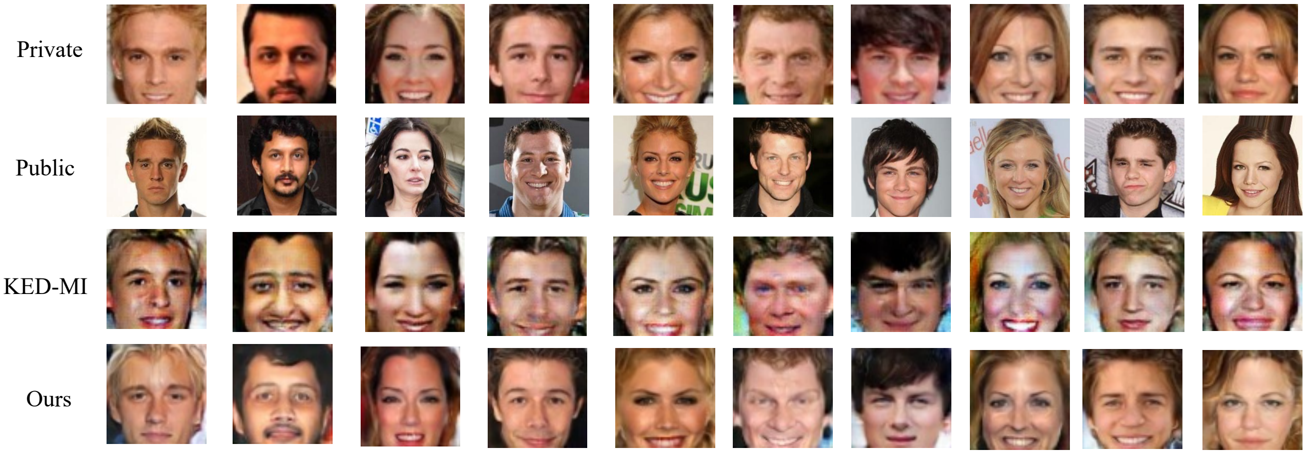


Figure 4: Visual comparison for attacking VGG16 trained on CelebA. The first row shows ground truth images of target identity in the private data. The second row shows the images from the public data with the highest confidence in the target identity. The third and last rows demonstrate the reconstructed images of the target identity using KED-MI and our attack, respectively.

where c is the target class in $\mathcal{D}_s$, and $\mathcal{L}_{inv}$ is a classification loss (e.g., cross-entropy). Then the reconstructed images can be obtained by $x^* = G(z^*)$.

Pseudo Label-Guided cGAN. Although existing generative MI attacks (Zhang et al. 2020; Chen et al. 2021) can learn a prior of natural images, they do not take into account the possible effects of class labels, thus causing all classes to be coupled together in the latent space. This makes it difficult to directly search for private images of the specified class. As mentioned before, in order to narrow the search space and conduct a more independent latent search process. We propose to train a conditional GAN (Miyato and Koyama 2018) to model the feature distribution of each class and use pseudo-labels to guide the direction of the generated images.

Formally, for training the discriminator in the cGAN, we use a hinge version of the standard adversarial loss:

$$\begin{aligned}\mathcal{L}_D =& E_{q(\tilde{y})}\left[E_{q(x|\tilde{y})}[\max(0, 1 - D(x, \tilde{y}))]+\right.\\ & \left.E_{q(\tilde{y})}\left[E_{p(z)}[\max(0, 1 + D(G(z, \tilde{y}), \tilde{y}))\right]\right],\end{aligned} \tag{2}$$

where $q(\tilde{y})$ and $q(x|\tilde{y})$ are the pseudo-label distribution of $\mathcal{D}_r$ and the image distribution in the corresponding class, respectively. $p(z)$ is standard Gaussian distribution and $G(z, \tilde{y})$ is the conditional generator.

To make the generated image more accurate, we use the pseudo-labels $\tilde{y}$ of $\mathcal{D}_r$ to impose an explicit constraint on the generator. The constraint, in principle, guides the generated images to belong to a certain class in the private dataset. Besides, we add a stochastic data augmentation module that performs random transformations on the generated images, including resizing, cropping, horizontal flipping, rotation, and color jittering. This module provides more stable convergence to realistic images while constraining. Then the loss function for the generator can be defined as:

$$\begin{aligned}\mathcal{L}_G =& - E_{q(\tilde{y})}\left[E_{p(z)}\left[D(G(z, \tilde{y}), \tilde{y}))\right]\right] +\\ & \alpha \mathcal{L}_{inv}(T(\mathcal{A}(G(z, \tilde{y}))), \tilde{y}),\end{aligned} \tag{3}$$

where T is the target model being attacked, $\mathcal{A}$ is a set of random augmentations, $\mathcal{L}_{inv}$ is the max-margin loss which we will introduce later, and α is a regularization coefficient.

Image Reconstruction. After getting the GAN trained on the public data, we can use it to reconstruct images of a specified class in the private dataset, as shown in the right half of Fig. 3. Specifically, given a target class c, we aim to search for appropriate latent vectors, so that the generated images constantly approach the images in c. Since we use a conditional generator, only the subspace of the specified class needs to be searched. In order to ensure that reconstructed images are not deceptive (e.g., adversarial example) or just stuck in a local minimum, we transform generated images randomly resulting in multiple correlated views. Intuitively, if the reconstructed image truly reveals key discriminative features of the target class, its class should remain consistent across these views. The objective can be defined as follows:

$$z^* = \arg\min_{\hat{z}} \sum_{i=1}^{m} \mathcal{L}_{inv}(T(\mathcal{A}_i(G(\hat{z}, c))), c), \tag{4}$$

where $\hat{z}$ is the latent vector to be optimized, $\mathcal{L}_{inv}$ is the max-margin loss, $\mathcal{A}_i$ is a set of random data augmentations, and m is the number of augmented views. Then we can obtain the reconstructed images by $x^* = G(z^*, c)$.

3.4 A Better Loss for MI Attacks

Gradient Vanishing Problem. Existing MI attacks have commonly adopted the cross-entropy (CE) loss as $\mathcal{L}_{inv}$. During attack optimization, the CE loss will cause the gradient to decrease and tend to vanish as the number of iterations is increased. For the target class c, the derivative of cross-entropy loss $\mathcal{L}_{\text{CE}}$ with respect to the output logits $\mathbf{o}$ can be derived as (see Appendix for derivation):

$$\frac{\partial \mathcal{L}_{\text{CE}}}{\partial \mathbf{o}} = \mathbf{p} - \mathbf{y}_c. \tag{5}$$

Here, $\mathbf{p}$ is the probability vector of the softmax output, that is $\mathbf{p} = [p_1, p_2, \ldots, p_K]$, $p_c \in [0, 1]$ denotes the predicted

	VGG16			ResNet-152			Face.evoLVe		
	GMI	**KED-MI**	**Ours**	**GMI**	**KED-MI**	**Ours**	**GMI**	**KED-MI**	**Ours**
Attack Acc ↑	.21±.0028	.63±.0018	**.97±.0001**	.31±.0035	.74±.0028	**1.±.0000**	.29±.0030	.74±.0013	**.99±.0001**
Top-5 Attack Acc ↑	.42±.0021	.87±.0015	**1.±.0000**	.55±.0045	.93±.0006	**1.±.0000**	.54±.0040	.94±.0009	**1.±.0000**
KNN Dist ↓	1712.57	1391.52	**1120.61**	1630.25	1323.16	**1026.71**	1638.94	1310.15	**1103.03**
FID ↓	42.86	30.92	**18.63**	42.50	26.23	**23.22**	41.53	27.92	**26.75**

Table 1: Attack performance comparison on various models trained on CelebA. ↑ and ↓ respectively symbolize that higher and lower scores give better attack performance.

		FFHQ → CelebA				FaceScrub → CelebA			
		Attack Acc ↑	**Attack Acc 5 ↑**	**KNN Dist ↓**	**FID↓**	**Attack Acc ↑**	**Attack Acc 5 ↑**	**KNN Dist↓**	**FID↓**
VGG16	**GMI**	.11±.0009	.27±.0048	1771.34	57.05	.02±.0004	.07±.0008	1997.16	150.19
	KED-MI	.34±.0026	.62±.0015	1555.57	49.51	.05±.0008	.14±.0006	1772.85	97.56
	Ours	**.89±.0006**	**.97±.0002**	**1284.16**	**27.32**	**.55±.0020**	**.77±.0012**	**1474.22**	**27.99**
Face.evoLVe	**GMI**	.13±.0009	.31±.0028	1739.88	56.66	.03±.0004	.10±.0012	1918.40	112.96
	KED-MI	.47±.0021	.74±.0013	1489.67	44.48	.09±.0006	.24±.0019	1712.31	99.78
	Ours	**.95±.0004**	**.99±.0001**	**1241.41**	**25.57**	**.57±.0013**	**.78±.0012**	**1502.82**	**34.10**
ResNet-152	**GMI**	.17±.0026	.37±.0030	1687.82	47.11	.04±.0011	.14±.0020	1865.44	109.16
	KED-MI	.74±.0028	.93±.0006	1323.16	26.23	.15±.0011	.36±.0020	1636.81	72.72
	Ours	**1.±.0000**	**1.±.0000**	**1026.71**	**23.22**	**.68±.0020**	**.87±.0011**	**1360.67**	**27.49**

Table 2: Attack performance comparison in the presence of a large distributional shift between public and private data. $A \to B$ represents the GAN and target model trained on datasets A and B, respectively.

probability of class c. $\mathbf{y}_c$ is the one-hot encoded vector of class c, that is, $\mathbf{y}_c = [0_1, \ldots, 1_c, \ldots, 0_K]$. Then, Eq. (5) can be rewritten as:

$$\frac{\partial \mathcal{L}_{\text{CE}}}{\partial \mathbf{o}} = [p_1, \ldots, p_c - 1, \ldots p_K] \,. \tag{6}$$

According to Eq. (6), as the generated image gradually approaches the target class during optimization, p_c will quickly reach 1 while $p_{i \neq c}$ will continue to decrease to 0. Eventually, this changing trend will cause the gradient of $\mathcal{L}_{\text{CE}}$ to vanish, making it difficult to search the latent vector of the generator.

Max-Margin Loss. To address this problem, we propose to replace the CE loss with the max-margin loss, which has been used in adversarial attacks to produce stronger attacks (Carlini and Wagner 2017; Sriramanan et al. 2020). In addition, we eliminate the softmax function and optimize the loss directly on the logits. The max-margin loss $\mathcal{L}_{\text{MM}}$ we use as $\mathcal{L}_{inv}$ is as follows:

$$\mathcal{L}_{\text{MM}} = -l_c(x) + \max_{j \neq c} l_j(x), \tag{7}$$

where l_c denotes the logit with respect to the target class c. For the target class c, the derivative of $\mathcal{L}_{\text{MM}}$ with respect to the logits can be derived as (see Appendix for derivation):

$$\frac{\partial \mathcal{L}_{\text{MM}}}{\partial \mathbf{o}} = \mathbf{y}_j - \mathbf{y}_t, \tag{8}$$

where $\mathbf{y}_j$ and $\mathbf{y}_t$ represent one-hot encoded vectors, so the elements of the gradient consist of constants, thus avoiding gradient vanishing problem. Moreover, max-margin loss encourages the algorithm to find the most representative sample in the target class while also distinguishing it from other classes. Compared with the cross-entropy loss, max-margin loss is more in line with the goal of MI attacks (further comparisons are given in the Appendix).

4 Experiments

In this section, we first provide a detailed introduction of the experimental settings. To demonstrate the effectiveness of our methods, we evaluate the proposed PLG-MI attack from several perspectives. The baselines that we will compare against are GMI proposed in (Zhang et al. 2020) and KED-MI proposed in (Chen et al. 2021), the latter achieved the state-of-the-art result for attacking DNNs.

4.1 Experimental Setting

Datasets. For face recognition, we select three widely used datasets for experiments: CelebA (Liu et al. 2015), FFHQ (Karras, Laine, and Aila 2019) and FaceScrub (Ng and Winkler 2014). CelebA contains 202,599 face images of 10,177 identities with coarse alignment. FFHQ consists of 70,000 high-quality PNG images and contains considerable variation in terms of age, ethnicity and image background. FaceScrub is a dataset of URLs for 100,000 images of 530 individuals. Similar to previous work (Zhang et al. 2020; Chen et al. 2021; Kahla et al. 2022), we crop the images of all datasets at the center and resize them to 64 × 64. More experiments on MNIST (LeCun et al. 1998), CIFAR-10 (Krizhevsky, Hinton et al. 2009) and ChestX-Ray (Wang et al. 2017) can be found in the Appendix.

Models. Following the setting of the state-of-the-art MI attack (Chen et al. 2021), we evaluate our attack on three deep models with various architectures: (1) VGG16 (Simonyan and Zisserman 2014); (2) Face.evoLVe (Cheng et al. 2017); and (3) ResNet-152 (He et al. 2016).

Implementation Details. In the standard setting, previous MI attacks usually split the dataset into two disjoint parts:

FaceScrub → CelebA		VGG16			Face.evoLVe			ResNet-152		
		Attack Acc ↑	KNN Dist ↓	FID ↓	Attack Acc ↑	KNN Dist ↓	FID ↓	Attack Acc ↑	KNN Dist ↓	FID ↓
VGG16	KED-MI	.05±.0008	1772.85	97.56	.10±.0006	1694.13	87.79	.12±.0009	1638.34	87.24
	Ours	**.55±.0020**	**1474.22**	**27.99**	**.76±.0017**	**1356.23**	**25.57**	**.81±.0016**	**1282.36**	**23.74**
Face.evoLVe	KED-MI	.05±.0006	1773.14	103.00	.09±.0006	1712.31	99.78	.13±.0018	1646.85	96.04
	Ours	**.54±.0019**	**1472.86**	**31.16**	**.57±.0013**	**1502.82**	**34.10**	**.71±.0015**	**1390.84**	**27.84**
ResNet-152	KED-MI	.06±.0004	1776.78	107.75	.12±.0012	1697.37	88.28	.15±.0011	1636.81	72.72
	Ours	**.57±.0019**	**1427.91**	**28.35**	**.68±.0026**	**1385.99**	**27.30**	**.68±.0020**	**1360.67**	**27.49**

Table 3: Attack performance comparison when using models with different architectures in the GAN training stage and the image reconstruction stage. The public dataset is FaceScrub which has a larger distributional shift with CelebA.

one part used as the private dataset to train the target model and the other as the public dataset. For training the target model, we use 30,027 images of 1,000 identities from CelebA as the private dataset. The disjoint part of CelebA is used to train the generator. However, this setting is too easy under our stronger PLG-MI attack. Therefore we focus on the scenario where the public dataset has a larger distributional shift with the private dataset, i.e., using two completely different datasets. Specifically, we use FFHQ and FaceScrub as public datasets respectively to train the generator. We set $n = 30$ for the top-*n* selection strategy, that is, each public dataset consists of 30,000 selected images that are reclassified into 1,000 classes by pseudo-labels. In stage-1, the GAN architecture we use is based on (Miyato and Koyama 2018). We apply spectral normalization (Miyato et al. 2018) to the all of the weights of the discriminator to regularize the Lipschitz constant. To train the GAN, we used Adam optimizer with a learning rate of 0.0002, a batch size of 64 and $\beta = (0, 0.9)$. The hyperparameter α in Eq. (3) is set to 0.2. In stage-2, we use the Adam optimizer with a learning rate of 0.1 and $\beta = (0.9, 0.999)$. The input vector z of the generator is drawn from a zero-mean unit-variance Gaussian distribution. We randomly initialize z for 5 times and optimize each round for 600 iterations.

4.2 Evaluation Metrics

The evaluation of the MI attack is based on the similarity of the reconstructed image and the target class image in the human-recognizable features. In line with previous work, we conducted both qualitative evaluation through visual inspection as well as quantitative evaluation. Specifically, the evaluation metrics we used are as follows.

Attack Accuracy (Attack Acc). We first build an evaluation model, which has a different architecture from the target model. We then use the evaluation model to compute the top-1 and top-5 accuracy of the reconstructed image on the target class. Actually, the evaluation model can be viewed as a proxy for a human observer to judge whether a reconstruction captures sensitive information. We use the model in (Cheng et al. 2017) for evaluation, which is pretrained on MS-Celeb1M (Guo et al. 2016) and then fine-tuned on training data of the target model.

K-Nearest Neighbor Distance (KNN Dist). Given a target class, we computed the shortest feature distance from a reconstructed image to the private images. The distance is measured by the ℓ_2 distance between two images in the feature space, i.e., the output of the penultimate layer of the evaluation model. A lower value indicates that the reconstructed image is closer to the private data.

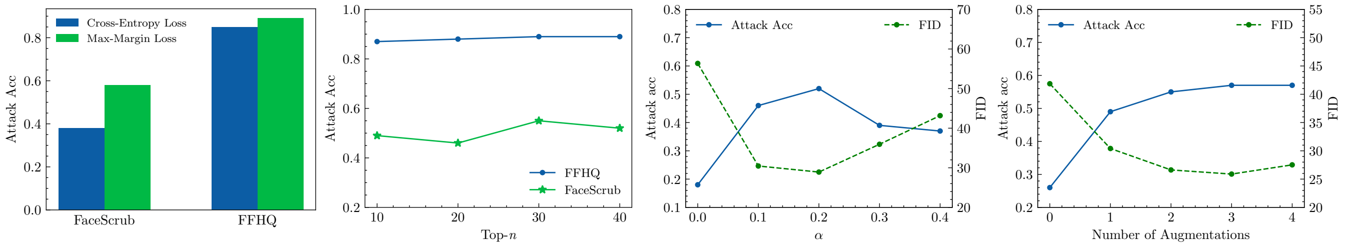
Fréchet Inception Distance (FID). FID (Heusel et al. 2017) is commonly used in the work of GAN to evaluate the generated images. Lower FID values indicate that the reconstructed images have better quality and diversity, making it easier for humans to identify sensitive features in them. Following the baseline setting, we only compute FID values on images where the attack is successful.

4.3 Experimental Results

Standard Setting. We first study the standard setting, i.e.,, dividing the CelebA dataset into a private dataset and a public dataset. As shown in Table 1, our method outperforms the baselines on all three models. The reconstructed images produced by our PLG-MI attack can achieve almost 100% accuracy on the evaluation model (i.e., Attack Acc), which outperforms the state-of-the-art method on average by approximately 30%. Our method is also vastly superior in FID and KNN Dist, significantly improving the visual quality and the similarity of reconstructed images to private datasets.

Fig. 4 shows the visual comparison of different methods. Compared with KED-MI, for different identities in the private dataset, our reconstructed images not only have more similar semantic features to the ground truth images, but are also more realistic. The second row shows the images of the public data with the highest confidence in the corresponding identity of the target model. We use the confidence as the basis for providing pseudo-labels in the top-*n* selection strategy. In most cases, the images selected by the strategy can provide some of the characteristic features needed to reconstruct the identity, such as gender, hair color and skin tone. But in terms of details, it can still be easily distinguished that they are not the same identity. Therefore, the optimization process of stage-2 is also very critical, and these candidate features will be further filtered and reorganized to increasingly resemble the target identity.

Larger Distributional Shifts. To further explore scenarios where the baselines performed poorly, we conducted experiments in the setting with larger distributional shifts. As shown in Table 2, GMI and KED-MI achieve the Attack Acc of 17% and 74% when exploiting FFHQ to attack ResNet-152, respectively. Since they do not fully explore the capa-



(a) (b) (c) (d)

Figure 5: (a)-(b) Attack performance with different $\mathcal{L}_{inv}$ and top-n selection strategy, respectively. (c) Attack Acc and FID with constraints of various strengths in Eq. (3). (d) Attack Acc and FID with various numbers of augmentations in Eq. (4). (c) and (d) use FaceScrub as the public dataset.

bility in the target model, the performance is poor in this more difficult setting. Compared with them, our PLG-MI attack achieves 89% Attack Acc when attacking VGG16 using FFHQ as public data, which is about 2.5 times higher than KED-MI. And the Attack Acc reaches more than 95% when attacking both Face.evoLVe and ResNet-152. In addition, there is a significant improvement in the FID for evaluating visual quality (e.g., a reduction from 49.51 to 27.32).

It should be emphasized that in the case of using FaceScrub as the public dataset, both GMI and KED-MI only obtain an almost failed attack performance. However, our method still maintains a high attack success rate, on average 50% higher than KED-MI on Attack Acc. As for the FID, our method maintains a low value in all cases, with an average reduction of about 3 times relative to KED-MI. The lower KNN Dist also shows that our method can reconstruct more accurate private images.

The results of FFHQ as public data are generally better than those of FaceScrub. The possible reason is that FaceScrub consists of face images crawled from the Internet, which is more different from the distribution of CelebA. Furthermore, we find that the degree of privacy leakage under MI attacks varies between model architectures. In our experiments, ResNet-152 is more likely to leak information in private datasets compared to VGG16 and Face.evoLVe. This phenomenon deserves further study in future work.

Generality of the GAN. The adversary may simultaneously perform MI attacks against multiple available target models to verify the correctness of the reconstructed image for the target identity. However, it will be very time-consuming to train a GAN separately for a specific target model each time. Therefore, we explore a new scenario where the architecture of the classifier used to help train the GAN is different from that of the target model.

Table 3 compares the results of our method with those of the baselines. Our method yields clearly superior results for all three models under various evaluation metrics. When using VGG16 to provide prior information to train a generator in stage-1, and using it to attack ResNet-152 in stage-2, our method achieves 81% Attack Acc, while KED-MI only achieves 12%. In addition, the FID of KED-MI is approximately 3 times higher on average than ours.

The experimental results of this scenario illustrate that regardless of the architecture of the models, our method is able to extract general knowledge to help recover sensitive information. Specifically, the generality of our cGAN can greatly improve the efficiency of attacks when there are multiple target models to be attacked.

Ablation study. We further investigate the effects of the various components and hyperparameters in PLG-MI and conduct attacks against VGG16 trained on CelebA using FaceScrub or FFHQ as the public dataset. As shown in Fig. 5 (a), we compare the impact of different $\mathcal{L}_{inv}$ used in MI attacks. Using the max-margin loss brings a significant improvement on Attack Acc, especially when the distributional shift is larger (i.e., the public dataset is FaceScrub). Fig. 5 (b) presents the results of using different n in the top-n selection strategy, its value has no significant impact on the attack performance, indicating that the strategy is not sensitive to n. Fig. 5 (c) shows the attack performance and visual quality when imposing explicit constraints of various strengths on the generator and $\alpha = 0.2$ is optimal among them. Fig. 5 (d) shows that the data augmentation module in stage-2 has a great influence on improving the Attack Acc and the visual quality of reconstructed images. However, when the number of augmentations is greater than 2, the improvement in attack performance is small, but the attack efficiency will be reduced. Thus we finally force the reconstructed images to maintain identity consistency after 2 random augmentations.

5 Conclusion

In this paper, we propose a novel MI attack method, namely PLG-MI attack. We introduce the conditional GAN and use pseudo-labels provided by the proposed top-n selection strategy to guide the training process. In this way, the search space in the stage of image reconstruction can be limited to the subspace of the target class, avoiding the interference of other irrelevant features. Moreover, we propose to use max-margin loss to overcome the problem of gradient vanishing. Experiments show that our method achieves the state-of-the-art attack performance on different scenarios with various model architectures. For future work, the black-box MI attack is still in its infancy, and the idea of our method can be transferred to the black-box scenario to further improve its performance.

Acknowledgements

This work was supported in part by the Natural Science Foundation of China under Grant U20B2047, 62102386, 62072421, 62002334 and 62121002.

References

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 39–57.

Chen, S.; Kahla, M.; Jia, R.; and Qi, G.-J. 2021. Knowledge-Enriched Distributional Model Inversion Attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16178–16187.

Cheng, Y.; Zhao, J.; Wang, Z.; Xu, Y.; Jayashree, K.; Shen, S.; and Feng, J. 2017. Know you at one glance: A compact vector representation for low-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 1924–1932.

Cohen, J. P.; Morrison, P.; and Dao, L. 2020. COVID-19 image data collection. *arXiv preprint arXiv:2003.11597*.

Dai, Z.; Yang, Z.; Yang, F.; Cohen, W. W.; and Salakhutdinov, R. R. 2017. Good semi-supervised learning that requires a bad gan. *Advances in Neural Information Processing Systems*, 30.

Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 1322–1333.

Fredrikson, M.; Lantz, E.; Jha, S.; Lin, S.; Page, D.; and Ristenpart, T. 2014. Privacy in Pharmacogenetics: An {End-to-End} Case Study of Personalized Warfarin Dosing. In *23rd USENIX Security Symposium*, 17–32.

Gopinath, D.; Converse, H.; Pasareanu, C.; and Taly, A. 2019. Property inference for deep neural networks. In *34th IEEE/ACM International Conference on Automated Software Engineering*, 797–809.

Guo, Y.; Zhang, L.; Hu, Y.; He, X.; and Gao, J. 2016. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, 87–102.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 770–778.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30.

Kahla, M.; Chen, S.; Just, H. A.; and Jia, R. 2022. Label-Only Model Inversion Attacks via Boundary Repulsion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15045–15053.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4401–4410.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8110–8119.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. *Tech Report*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, M.; Deng, C.; Li, T.; Yan, J.; Gao, X.; and Huang, H. 2020. Towards transferable targeted attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 641–649.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3730–3738.

Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

Miyato, T.; and Koyama, M. 2018. cGANs with projection discriminator. *arXiv preprint arXiv:1802.05637*.

Ng, H.-W.; and Winkler, S. 2014. A data-driven approach to cleaning large face datasets. In *IEEE International Conference on Image Processing*, 343–347.

Rajpurkar, P.; Irvin, J.; Zhu, K.; Yang, B.; Mehta, H.; Duan, T.; Ding, D.; Bagul, A.; Langlotz, C.; Shpanskaya, K.; et al. 2017. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. *Advances in Neural Information Processing Systems*, 29.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Srivacy*, 3–18.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sriramanan, G.; Addepalli, S.; Baburaj, A.; et al. 2020. Guided adversarial attack for evaluating and enhancing adversarial defenses. *Advances in Neural Information Processing Systems*, 33: 20297–20308.

Struppek, L.; Hintersdorf, D.; Correia, A. D. A.; Adler, A.; and Kersting, K. 2022. Plug & Play Attacks: Towards Robust and Flexible Model Inversion Attacks. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, 20522–20545. PMLR.

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1701–1708.

Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M. K.; and Ristenpart, T. 2016. Stealing Machine Learning Models via Prediction {APIs}. In *25th USENIX Security Symposium*, 601–618.

Wang, K.-C.; Fu, Y.; Li, K.; Khisti, A.; Zemel, R.; and Makhzani, A. 2021. Variational Model Inversion Attacks. *Advances in Neural Information Processing Systems*, 34: 9706–9719.

Wang, X.; Peng, Y.; Lu, L.; Lu, Z.; Bagheri, M.; and Summers, R. M. 2017. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2097–2106.

Zhang, Y.; Jia, R.; Pei, H.; Wang, W.; Li, B.; and Song, D. 2020. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 253–261.

A Appendix

A.1 The Derivatives of Cross-Entropy Loss and Max-Margin Loss

Cross-Entropy Loss We define the CE loss for the K-classification task as follows:

$$\mathcal{L} = -\sum_{i=1}^{K} y_i \log\left(p_i\right), \tag{9}$$

where $\mathbf{y}_i$ denotes the one-hot encoded vector of class i, in which only the i_{th} element y_i is 1, and the rest are 0. (i.e., $\mathbf{y}_i = [0_1, \dots, 1_i, \dots, 0_K]$). Thus, for the specified target class t, Eq. (9) can be rewritten as:

$$\mathcal{L} = -\log(p_t). \tag{10}$$

Here, we denote the logits of the model output as $\mathbf{o} = [o_1, \dots, o_t, \dots, o_K]$, and denote the softmax output (i.e., probability vector) as $\mathbf{p} = [p_1, \dots, p_t, \dots, p_K]$. Then, the derivative of the CE Loss ($\mathcal{L}$) with respect to the logits ($\mathbf{o}$) is as follow:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{o}} = \frac{\partial \mathcal{L}}{\partial \mathbf{p}} \frac{\partial \mathbf{p}}{\partial \mathbf{o}}. \tag{11}$$

It can be seen from Eq. (10) that $\mathcal{L}$ is only related to p_t, so we can get:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{p}} = \left[0_1, \dots, -\frac{1}{p_t}, \dots, 0_K\right]. \tag{12}$$

And $\frac{\partial \mathbf{p}}{\partial \mathbf{o}}$ of Eq. (11) is a Jacobian matrix, as follows:

$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{o}} = \begin{pmatrix} \frac{\partial p_1}{\partial o_1} & \frac{\partial p_1}{\partial o_2} & \cdots \frac{\partial p_1}{\partial o_K} \\ \vdots & \vdots & \ddots \\ \frac{\partial p_j}{\partial o_1} & \frac{\partial p_j}{\partial o_2} & \cdots \frac{\partial p_j}{\partial o_K} \\ \vdots & \vdots & \ddots \\ \frac{\partial p_K}{\partial o_1} & \frac{\partial p_K}{\partial o_2} & \cdots \frac{\partial p_K}{\partial o_K} \end{pmatrix}. \tag{13}$$

As only the element in row t is not 0 in Eq. (13), we arrive at:

$$\begin{aligned} \frac{\partial \mathcal{L}_{CE}}{\partial \boldsymbol{o}} &= \frac{\partial \mathcal{L}_{CE}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{o}} = -\frac{1}{p_t} \frac{\partial p_t}{\partial \boldsymbol{o}} \\ \text{where } \ p_t &= \frac{e^{o_j}}{\sum_{i=1}^{K} e^{o_i}}. \end{aligned} \tag{14}$$

With Eq. (14), for the case where $i \neq t$, the derivative of CE loss is as follows:

$$\begin{aligned} \frac{\partial L}{\partial o_i} &= \frac{\partial L}{\partial p_t} \frac{\partial p_t}{\partial o_i} \\ &= \frac{\partial L}{\partial p_t} \frac{0 - e^{o_t} e^{o_i}}{\left(\sum_{i=1}^{K} e^{o_i}\right)^2} \\ &= (-\frac{1}{p_t})(-p_t p_i) \\ &= p_i. \end{aligned} \tag{15}$$

And for the case where $i = t$, the derivative of CE loss is as follows:

$$\begin{aligned} \frac{\partial L}{\partial o_i} &= \frac{\partial L}{\partial p_t} \frac{\partial p_t}{\partial o_i} \\ &= \frac{\partial L}{\partial p_t} \frac{e^{o_t} \sum_{i=1}^{K} e^{o_i} - e^{o_t} e^{o_t}}{\left(\sum_{i=1}^{K} e^{o_i}\right)^2} \\ &= (-\frac{1}{p_t}) \left(p_t - p_t^2\right) \\ &= p_t - 1. \end{aligned} \tag{16}$$

From Eq. (15) and Eq. (16), we derive the final derivative of CE loss as:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{o}} &= [p_1, \dots, p_t - 1, \dots p_K] \\ &= [p_1, \dots, p_t, \dots p_K] - [0_1, \dots, 1_t, \dots, 0_K] \\ &= \mathbf{p} - \mathbf{y}_t. \end{aligned} \tag{17}$$

Max-Margin Loss. Without loss of generality, we can define the max-margin loss for a target class t as follows:

$$\mathcal{L} = o_j - o_t, \tag{18}$$

where $j = \arg\max_{i \neq t} o_i$ indicates the highest class except for the target class t. We denote the derivative of $\mathcal{L}$ to the logits $\mathbf{o}$ as $\frac{\partial \mathcal{L}}{\partial \mathbf{o}}$. It can be deduced that $\frac{\partial \mathcal{L}}{\partial o_i} = 0$ when $i \neq j$ and $i \neq t$, and $\frac{\partial \mathcal{L}}{\partial o_i}$ is 1 and -1 when $i = j$ and $i = t$, respectively. Thus, we finally derive at:

$$\frac{\partial L}{\partial \mathbf{o}} = \mathbf{y}_j - \mathbf{y}_t. \tag{19}$$

A.2 Experiments on Other Datasets

We also perform MI attacks on the digit recognition task, the object classification task, and the disease prediction task, using the MNIST (LeCun et al. 1998), CIFAR-10 (Krizhevsky, Hinton et al. 2009), and ChestX-Ray (Wang et al. 2017) datasets for experiments, respectively.

Experimental Details. For MNIST and CIFAR-10, we use the images in the training data with labels 0, 1, 2, 3, and 4 as the private data, containing 30,596 and 25,000 images, respectively. The rest images with labels 5, 6, 7, 8, and 9 are used as the public data, containing 29,404 and 25,000 images, respectively. We adopt ResNet-18 trained on the private data as the target model. As for the evaluation model, we train VGG16 on the original training data instead of the private data to better distinguish deceptive reconstructed images. For ChestX-Ray, we use 14 classes of images with diseases as the private data to train the target model ResNet-18. Then, we randomly select 20,000 images from the class with the label "NoFinding" (i.e., no disease) as the public data to train the evaluation model ResNet-34. Furthermore, we also use a different COVID19 dataset (Cohen, Morrison, and Dao 2020) with 21,165 images as the public data. All images from above datasets are resized to 64 × 64. n for the top-n selection strategy is set to 4,000 for MNIST and CIFAR-10, and 1,000 for ChestX-Ray. The learning rates in stage-2 are set to 0.1 and 0.001, respectively, for 300 iterations. We reconstruct 100 images per class for MNIST and CIFAR-10, and 10 images per class for Chest-X-ray.

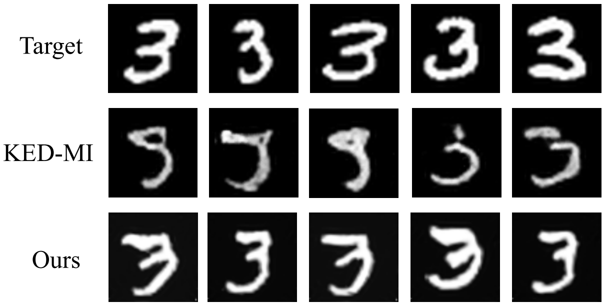


Figure 6: The digit "3" samples of MNIST reconstructed by the baseline and our method.

Experimental Results. As shown in Table 4, our method comprehensively outperforms the baselines by a large margin on MINST and CIFAR-10 datasets. When attacking the digit recognition model trained on MNIST, GMI performs very poorly. Although KED-MI can slightly improve the attack accuracy, it cannot reconstruct the private images of the digit "3" well, as shown in Fig. 6. In contrast, our method reduces interference between different classes, thus allowing private images to be reconstructed with good visual quality on each class. Similar visual comparisons on CIFAR-10 are also shown in Fig. 7. Our method not only significantly outperforms GMI and KED-MI in terms of image realism, but also recovers more accurate semantic information of private classes. It is difficult for GMI to reconstruct accurate private images, while KED-MI tends to reconstruct deceptive images that can be classified by the target model but have few human-recognizable features.

The results of ChestX-Ray are shown in Table 5. Compared to KED-MI, our method improves the attack accuracy by an average of 12% under the two different public datasets. Meanwhile, the FID is even reduced from 109.41 to 47.51 when the public data is COVID19. We also provide the visual comparison of the private data reconstructed by KED-MI and our method, as shown in Fig. 8.

		Attack Acc ↑	KNN Dist ↓	FID ↓
MNIST	**GMI**	.03±.0054	171.18	137.53
	KED-MI	.29±.0228	78.67	116.47
	Ours	**.60±.0465**	**38.33**	**77.68**
CIFAR10	**GMI**	.11±.0173	38.50	198.18
	KED-MI	.50±.0392	15.49	141.18
	Ours	**.76±.0318**	**5.11**	**83.43**

Table 4: Attack performance comparison on MNIST and CIFAR10.

	NoFinding → ChestX-Ray		COVID19 → ChestX-Ray	
	KED-MI	**Ours**	**KED-MI**	**Ours**
Attack Acc ↑	.71±.0034	**.82±.0048**	.68±.0026	**.80±.0020**
KNN Dist ↓	97.26	**82.71**	113.18	**84.56**
FID ↓	97.35	**64.33**	109.41	**47.51**

Table 5: Attack performance comparison with KED-MI on ChestX-Ray.

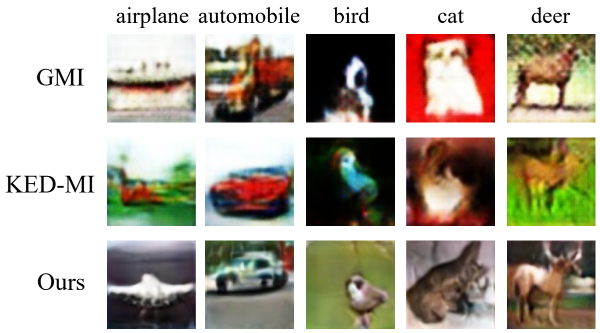


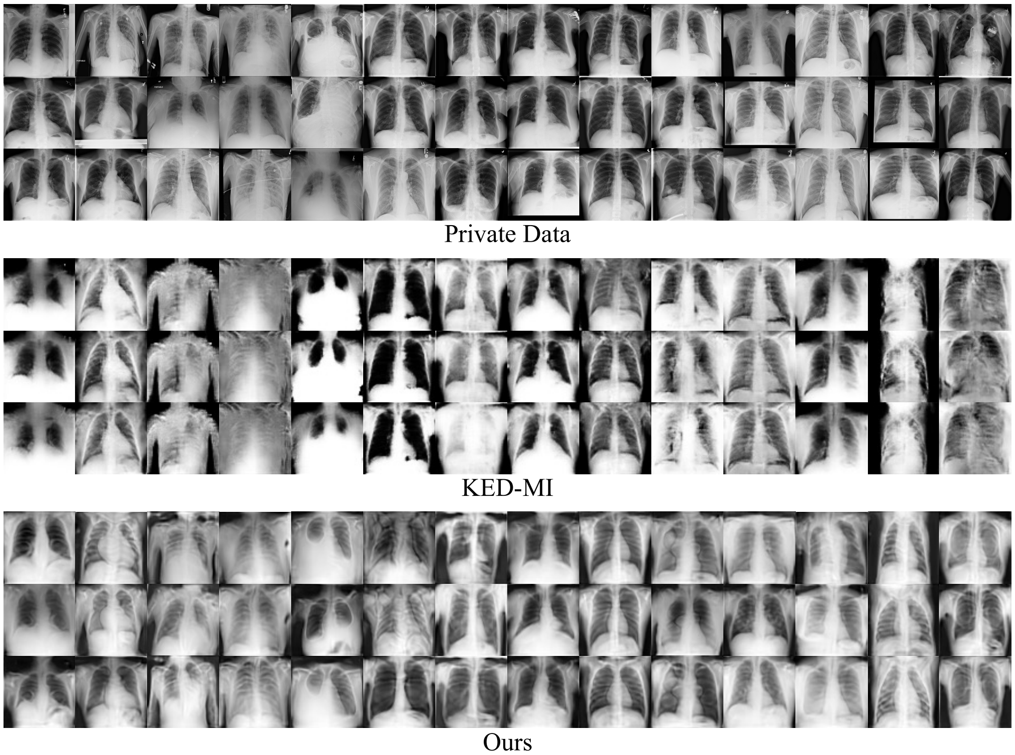
Figure 7: CIFAR-10 samples reconstructed by the baselines and our method.

Analysis on the Failure of KED-MI. As mentioned in the main manuscript, the KED-MI adopts the semi-supervised GAN framework and uses the feature-matching loss to train the generator. However, as demonstrated in (Salimans et al. 2016), the feature-matching loss works better if the goal is to obtain a strong classifier (i.e., the discriminator) using the semi-supervised GAN, otherwise it reduces the visual quality of the generated images. Moreover, Dai *et al.* (Dai et al. 2017) theoretically analyzes that there is a trade-off between the quality of the discriminator and the generator. Obviously, this framework is not suitable for generative MI attacks, since what we need in stage-2 is a good and accurate generator, contrary to the goal of semi-supervised GAN.

Rather than searching the latent vector of each reconstructed image as GMI and our method, KED-MI proposed to search for a distribution $\mathcal{N}(\mu, \sigma^2)$ with two learnable parameters μ and σ^2 for each target class, and the latent vectors can be obtained by randomly sampling from the learned distribution. However, it is very difficult to learn an accurate class distribution by $\mathcal{N}(\mu, \sigma^2)$ when the inner-class distribution of the private dataset varies greatly (e.g., CIFAR-10). As a result, latent vectors sampled from such a distribution are far from the manifold of natural images, although the produced images may be close to private classes in feature space. Whereas we use a more powerful conditional GAN to model the distribution of each private class, making it possible to handle more diverse datasets. In addition, we also perform random transformations on the generated images during the search process to further filter deceptive or adversarial samples.

A.3 Empirical Comparisons of Different $\mathcal{L}_{inv}$

We note that a concurrent work (Struppek et al. 2022) also has a similar observation of the gradient vanishing problem when using cross-entropy loss in MI attacks. Inspired by (Li et al. 2020), they adopt a more complex poincaré loss as



Private Data

KED-MI

Ours

Figure 8: ChestX-Ray samples reconstructed by the baseline (KED-MI) and our method when using COVID19 as the public dataset. Each column represents one category of the 14 diseases.

follows:

$$\begin{aligned}\mathcal{L}_{\text{Poincaré}} &= d(\boldsymbol{u}, \boldsymbol{v}) \\ &= \operatorname{arcosh}\left(1 + \frac{2\|\boldsymbol{u}-\boldsymbol{v}\|_2^2}{(1-\|\boldsymbol{u}\|_2^2)\,(1-\|\boldsymbol{v}\|_2^2)}\right), \quad (20) \\ \boldsymbol{u} &= \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|_1}, \boldsymbol{v} = \max\{\boldsymbol{v}-\xi, 0\},\end{aligned}$$

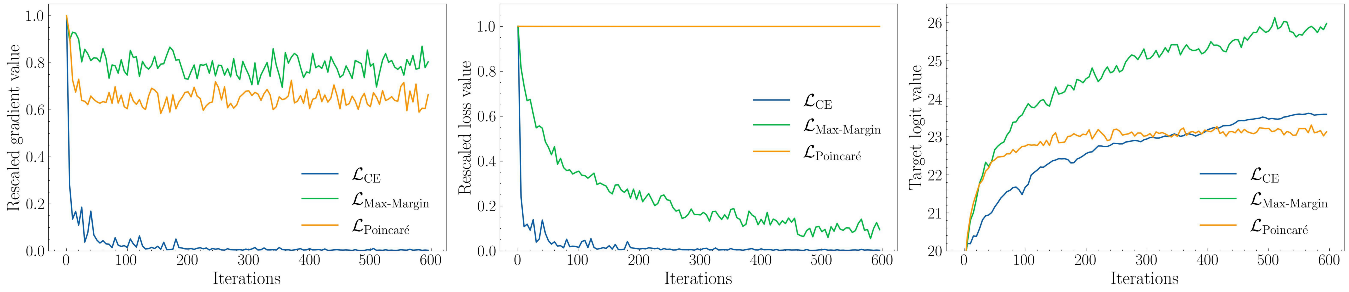
where $\boldsymbol{u}$ is the logits normalized by the ℓ_1 distance, and $\boldsymbol{v}$ is the one-hot encoded target vector with respect to the target class. $\xi = 10^{-5}$ is a small constant to ensure numerical stability.

In the main manuscript, we theoretically compare the cross-entropy loss and the proposed max-margin loss. Here, we further empirically compare them with the additional poincaré loss mentioned in Eq. (20). Specifically, we take these three losses as $\mathcal{L}_{inv}$ respectively in the iterative process of stage-2, and plot the trend curves of gradient values, loss values, and target logit values. Since the gradient values of different losses have a gap, we rescale the gradient values by dividing $\|g_0\|_1$, where g_0 is the gradient of the first iteration. The ℓ_1 norm of the gradient $\|g_i\|_1$ represents its gradient value. Similarly, the loss values are also rescaled by dividing its value of the first iteration. We use FFHQ and CelebA as the public and private dataset, respectively. Then, we use VGG16 as the target model to attack the first 100 classes of CelebA and take their average results for plotting.

From Fig. 9 (a), we can see that the gradient of the cross-entropy loss quickly decreases to 0 as the number of iterations increases, indicating the existence of the gradient vanishing problem. Meanwhile, the max-margin loss maintains a more stable gradient magnitude than the poincaré loss over the iterations, i.e., the degradation is more slight. In Fig. 9 (b), the value of the cross-entropy loss also quickly decreases to 0 due to the gradient vanishing, while the value of poincaré loss keeps almost unchanged, which may make it difficult to judge the optimization situation. In contrast, the max-margin loss can be minimized continuously over the iterations with relatively stable gradients. As shown in Fig. 9 (c), the max-margin loss can make the logit value of the target class steadily increase over the iterations to reach a higher value, while the cross-entropy loss and poincaré loss gradually tend to a relatively constant value in the later iterations. In addition, Fig. 10 shows the attack performance when using these three losses separately in stage-2 of our method, demonstrating that the max-margin loss outperforms the other two losses, especially when there is a larger distributional shift between public and private data.

A.4 More Discussions on the Concurrent Work

Apart from the loss function, the other differences between our method and the concurrent work (Struppek et al. 2022) are three-fold. First, the motivations are different: we propose to design a more powerful general MI attack framework, while they aim to omit the training process of stage-1 to achieve plug-and-play. Second, based on the different goals, the technical contributions are different: we propose a



(a) (b) (c)

Figure 9: (a)-(c) are the trend curves of the rescaled gradient values, the rescaled loss values, and the target logit values for different losses over the iterations, respectively.

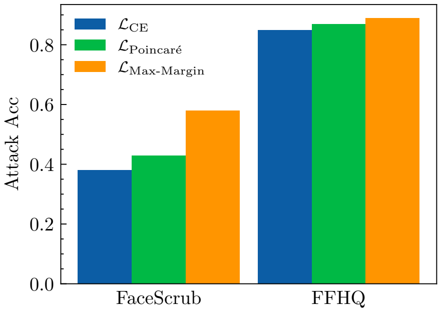


Figure 10: Comparison of attack performance when using different losses in the image reconstruction stage.

top-*n* selection strategy and aim to train a class-independent generator to search private images more accurately, while they focus on extending the reconstruction process in stage-2 to reduce the dependence on the trained generator. Third, the application scenarios are different: what we propose is a general attack method that can be applied to tasks trained on various sensitive data (e.g, disease diagnosis), while their method relies on the task of the publicly available pre-trained GAN and can not applicable for some unique tasks. Overall, our method differs from (Struppek et al. 2022) in terms of motivation, technical contributions, and application scenarios.