THE SECRET REVEALER: GENERATIVE MODEL INVERSION ATTACKS AGAINST DEEP NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper studies *model inversion attacks*, in which the access to a model is abused to infer information about the training data. Since its first introduction by Fredrikson et al. (2014), such attacks have raised serious concerns given that training data usually contain sensitive information. Thus far, successful model inversion attacks have only been demonstrated on simple models, such as linear regression and logistic regression. Previous attempts to invert neural networks, even the ones with simple architectures, have failed to produce convincing results. We present a novel attack method, termed the *generative model inversion attack*, which can invert deep neural networks with high success rates. Rather than reconstructing private training data from scratch, we leverage partial public information, which can be very generic, to learn a distributional prior via generative adversarial networks (GANs) and use it to guide the inversion process. Moreover, we theoretically prove that a model's predictive power and its vulnerability to inversion attacks are indeed two sides of the same coin—highly predictive models are able to establish a strong correlation between features and labels, which coincides exactly with what an adversary exploits to mount the attacks. Our experiments demonstrate that the proposed attack improves identification accuracy over the existing work by about 75% for reconstructing face images from a state-of-the-art face recognition classifier. We also show that differential privacy, in its canonical form, is of little avail to protect against our attacks.

1 INTRODUCTION

Deep neural networks (DNNs) have been adopted in a wide range of applications, including computer vision, speech recognition, healthcare, among others. The fact that many compelling applications of DNNs involve processing sensitive and proprietary datasets raised great concerns about privacy. In particular, when machine learning (ML) algorithms are applied to private training data, the resulting models may unintentionally leak information about training data through their output (i.e., black-box attack) or their parameters (i.e., white-box attack).

A concrete example of privacy attacks is model inversion (MI) attacks, which aim to reconstruct sensitive features of training data by taking advantage of their correlation with the model output. Algorithmically, MI attacks are implemented as an optimization problem seeking for the sensitive feature value that achieves the maximum likelihood under the target model. The first MI attack was proposed in the context of genomic privacy (Fredrikson et al., 2014), where the authors showed that adversarial access to a linear regression model for personalized medicine can be abused to infer private genomic attributes about individuals in the training dataset. Recent work (Fredrikson et al., 2015) extended MI attacks to other settings, e.g., recovering an image of a person from a face recognition model given just their name, and other target models, e.g., logistic regression and decision trees.

Thus far, effective MI attacks have only been demonstrated on the aforementioned simple models. It remains an open question whether it is possible to launch the attacks against a DNN and reconstruct its private training data. The challenges of inverting DNNs arise from the intractability and ill-posedness of the underlying attack optimization problem. For neural networks, even the ones with one hidden

layer, the corresponding attack optimization becomes a non-convex problem; solving it via gradient descent methods may easily stuck in local minima, which leads to poor attack performance. Moreover, in the attack scenarios where the target model is a DNN (e.g., attacking face recognition models), the sensitive features (face images) to be recovered often lie in a high-dimensional, continuous data space. Directly optimizing over the high-dimensional space without any constraints may generate unrealistic features lacking semantic information (See Figure 1).

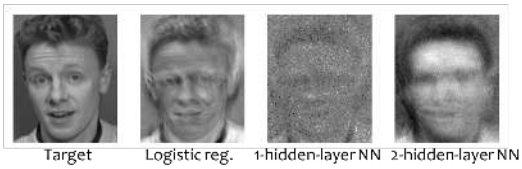

Figure 1: Reconstruction of the individual on the left by attacking three face recognition models (logistic regression, one-hidden-layer and two-hidden-layer neural network) using the existing attack algorithm in (Fredrikson et al., 2015)

In this paper, we focus on image data and propose a simple yet effective attack method, termed the generative model inversion (GMI) attack, which can invert DNNs and synthesize private training data with high fidelity. The key observation supporting our approach is that it is arguably easy to obtain information about the general data distribution, especially for the image case. For example, against a face recognition classifier, the adversary could randomly crawl facial images from the Internet without knowing the private training data. We find these datasets, although may not contain the target individuals, still provide rich knowledge about how a face image might be structured; extraction and proper formulation of such prior knowledge will help regularize the originally ill-posed inversion problem. We also move beyond specific attack algorithms and explore the fundamental reasons for a model's susceptibility to inversion attacks. We show that the vulnerability is unavoidable for highly predictive models, since these models are able to establish a strong correlation between features and labels, which coincides exactly with what an adversary exploits to mount MI attacks.

Our contributions can be summarized as follows: (1) We propose to use generative models to learn an informative prior from public datasets so as to regularize the ill-posed inversion problem. (2) We propose an end-to-end GMI attack algorithm based on GANs, which can reveal private training data of DNNs with high fidelity. (3) We present a theoretical result that uncovers the fundamental connection between a model's predictive power and its susceptibility to general MI attacks and empirically validate it. (4) We conduct extensive experiments to demonstrate the performance of the proposed attack. Experiment code is publicly available at `https://tinyurl.com/yxbnjk4s`.

Related Work Privacy attacks against ML models consist of methods that aim to reveal some aspects of training data. Of particular interest are membership attacks and MI attacks. Membership attacks aim to determine whether a given individual's data is used in training the model (Shokri et al., 2017). MI attacks, on the other hand, aim to reconstruct the features corresponding to specific target labels.

In parallel to the emergence of various privacy attack methods, there is a line work that formalizes the privacy notion and develops defenses with formal and provable privacy guarantees. One dominate definition of privacy is differential privacy (DP), which carefully randomizes an algorithm so that its output does not to depend too much on any individuals' data (Dwork et al., 2014). In the context of ML algorithms, DP guarantees protect against attempts to infer whether a data record is included in the training set from the trained model (Abadi et al., 2016). By definition, DP limits the success rate of membership attacks. However, it does not explicitly protect attribute privacy, which is the target of MI attacks (Fredrikson et al., 2014).

The first MI attack was demonstrated in (Fredrikson et al., 2014), where the authors presented an algorithm to recover genetic markers given the linear regression that uses them as input features, the response of the model, as well as other non-sensitive features of the input. Hidano et al. (2017) proposed a algorithm that allows MI attacks to be carried out without the knowledge of non-sensitive features by poisoning training data properly. Despite the generality of the algorithmic frameworks proposed in the above two papers, the evaluation of the attacks is only limited to linear models. Fredrikson et al. (2015) discussed the application of MI attacks to more complex models including some shallow neural networks in the context of face recognition. Although the attack can reconstruct face images with identification rates much higher than random guessing, the recovered faces are indeed blurry and hardly recognizable. Moreover, the quality of reconstruction tends to degrade for

more complex architectures. Yang et al. (2019b) proposed to train a separate network that swaps the input and output of the target network to perform MI attacks. The inversion model can be trained with black-box accesses to the target model. However, their approach cannot directly be benefited from the white-box setting.

Moreover, several recent papers started to formalize MI attacks and study the factors that affect a model's vulnerability from a theoretical viewpoint. For instance, Wu et al. (2016) characterized model invertibility for Boolean functions using the concept of influence from Boolean analysis; Yeom et al. (2018) formalized the risk that the model poses specifically to individuals in the training data and shows that the risk increases with the degree of overfitting of the model. However, their theory assumed that the adversary has access to the join distribution of private feature and label, which is overly strong for many attack scenarios. Our theory does not rely on this assumption and better supports the experimental findings.

2 GENERATIVE MI ATTACK

An overview of our GMI attack is illustrated in Figure 2. In this section, we will first discuss the threat model and then present our attack method in details.

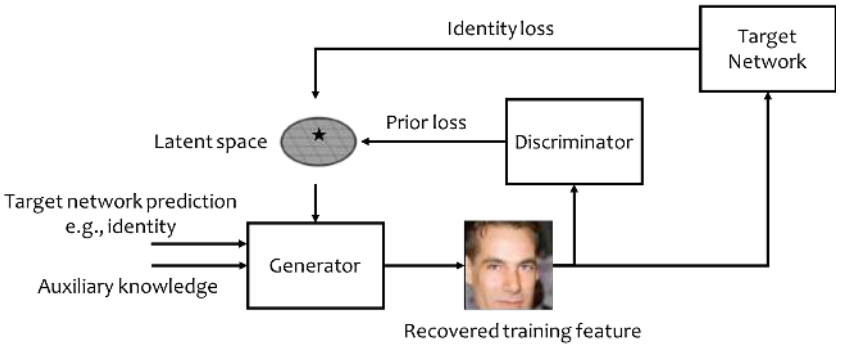

Figure 2: Overview of the proposed GMI attack method.

2.1 THREAT MODEL

In traditional MI attacks, an adversary, given a model trained to predict specific labels, uses it to make predictions of sensitive features used during training. Throughout the paper, we will refer to the model subject to attacks as the *target network*. We will use face recognition classifiers as a running example for the target network. Face recognition classifiers label an image containing a face with an identifier corresponding to the individual depicted in the image. We assume that the adversary employs an inference technique to discover the face image x for some specific identity y output by the classifier f. Following the canonical setup of MI attacks, we assume that the adversary has access to the target network f. In addition to f, the adversary may also have access to some auxiliary knowledge that facilitates his inference.

Possible Auxiliary Knowledge Examples of auxiliary knowledge could be a blurred or corrupted image which only contains nonsenstive information, such as background pixels in a face image. This auxiliary knowledge might be easy to obtain, as blurring and corruption are often applied to protect anonymity of individuals in public datasets (Carrell et al., 2012; Li et al., 2019).

Connection to Image Inpainting The setup of MI attacks on images resembles the widely studied image inpainting tasks in computer vision, which also try to fill missing pixels of an image. The difference is, however, in the goal of the two. MI attacks try to fill the sensitive features associated with a specific identity in the training set. In contrast, image inpainting tasks only aim to synthesize visually realistic and semantically plausible pixels for the missing regions; whether the synthesized pixels are consistent with a specific identity is beyond the scope. Despite the difference, our approach to MI attacks leverages some training strategies from the venerable line of work on image inpainting (Yeh

et al., 2017; Iizuka et al., 2017; Yang et al., 2019a) and significantly improves the recognizability of the reconstructed images over the existing attack methods.

2.2 Inferring Missing Sensitive Features

To realistically reconstruct missing sensitive regions in an image, our approach utilizes the generator G and the discriminator D, all of which are trained with public data. After training, we aim to find the latent vector $\hat{z}$ that achieves highest likelihood under the target network while being constrained to the data manifold learned by G. However, if not properly designed, the generator may not allow the target network to easily distinguish between different latent vectors. For instance, in extreme cases, if the generated images of all latent vectors collapse to the same point in the feature space of the target network, then there is no hope to identify which one is more likely to appear in its private training set of the target network. To address this issue, we present a simple yet effective loss term to promote the diversity of the data manifold learned by G when projected to the target network's feature space.

Specifically, our reconstruction process consists of two stages: (1) *Public knowledge distillation*, in which we train the generator and the discriminators on public datasets in order to encourage the generator to generate realistic-looking images. The public datasets can be unlabeled and have no identity overlapping with the private dataset. (2) *Secret revelation*, in which we make use of the generator obtained from the first stage and solve an optimization problem to recover the missing sensitive regions in an image.

For the first stage, we leverage the canonical Wasserstein-GAN (Arjovsky et al., 2017) training loss. The loss function is adapted to the two discriminators for our case:

$$\min_G \max_D L_{\text{wgan}}(G, D) = \mathbb{E}_x[D(x)] - \mathbb{E}_z[D(G(z))] \tag{1}$$

In addition, inspired by Yang et al. (2019a), we introduce a diversity loss term that promotes the diversity of the images synthesized by G when projected to the target network's feature space. Let F denote the feature extractor of the target network. The diversity loss can thus be expressed as

$$\max_G L_{\text{div}}(G) = E_{\mathbf{z_1}, \mathbf{z_2}} \left[\frac{\|F(G(\mathbf{z_1})) - F(G(\mathbf{z_2}))\|}{\|\mathbf{z_1} - \mathbf{z_2}\|} \right] \tag{2}$$

As discussed above, larger diversity will facilitate the targeted network to discern the generated image that is most likely to appear in its private training set. Our full objective for public knowledge distillation can be written as $\min_G \max_D L_{\text{wgan}}(G, D) - \lambda_d L_{\text{div}}(G)$.

In the secret revelation stage, we solve the following optimization to find the latent vector that generates an image achieving the maximum likelihood under the target network while remaining realistic: $\hat{z} = \arg\min_z L_{\text{prior}}(z) + \lambda_i L_{\text{id}}(z)$, where the prior loss $L_{\text{prior}}(z)$ penalizes unrealistic images and the identity loss $L_{\text{id}}(z)$ encourages the generated images to have likelihood under the targeted network. They are defined, respectively, by

$$L_{\text{prior}}(z) = -E_{\mathbf{z}}[D(G(\mathbf{z}))] \qquad \mathcal{L}_{\text{id}} = -E_{\mathbf{z}} \log[C(G(\mathbf{z}))] \tag{3}$$

where $C(G(\mathbf{z}))$ represent the probability of $G(\mathbf{z})$ output by the target network.

3 Connection Between Model Predictive Power and MI Attacks

For a fixed data point (x, y), we can measure the performance of a model f for predicting the label y of feature x using the log likelihood $\log p_f(y|x)$. It is known that maximizing the log likelihood is equivalent to minimizing the cross entropy loss—one of the most commonly used loss function for training DNNs. Thus, throughout the following analysis, we will focus on the log likelihood as a model performance measure.

Now, suppose that (X, Y) is drawn from an unknown data distribution $p(X, Y)$. Moreover, $X = (X_s, X_{ns})$, where X_s and X_{ns} denote the sensitive and non-sensitive part of the feature, respectively. We can define the predictive power of the sensitive feature X_s under the model f (or equivalently, the predictive power of model f using X_s) as the change of model performance when excluding it from

the input, i.e., $\mathbb{E}_{(X,Y)\sim p(X,Y)}[\log p_f(Y|X_s, X_{ns}) - p_f(Y|X_{ns})]$. Similarly, we define the predictive power of the sensitive feature given a specific class y and nonsensitive feature x_{ns} as

$$U_f(x_{ns}, y) = \mathbb{E}_{X_s\sim p(X_s|y,x_{ns})}[\log p_f(y|X_s, x_{ns}) - \log P_f(y|x_{ns})] \tag{4}$$

We now consider the measure for the MI attack performance. Recall the goal of the adversary is to guess the value of x_s given its corresponding label y, the model f, and some auxiliary knowledge x_{ns}. The best attack outcome is the recovery of the entire posterior distribution of the sensitive feature, i.e., $p(X_s|y, x_{ns})$. However, due to the incompleteness of the information available to the adversary, the best possible attack result that adversary can achieve under the attack model can be captured by $p_f(X_s|y, x_{ns}) \propto p_f(y|X_s, x_{ns})p(X_s|x_{ns})$, assuming that the adversary can have a fairly good estimate of $p(X_s|x_{ns})$. Such estimate can be obtained by, for example, learning from public datasets using the method in Section 2.2. Although MI attack algorithms often output a single feature vector as the attack result, these algorithms can be adapted to output a feature distribution instead of a single point by randomizing the starting guess of the feature. Thus, it is natural to measure the MI attack performance in terms of the similarity between $p(X_s|y, x_{ns})$ and $p_f(X_s|y, x_{ns})$. The next theorem indicates that the vulnerability to MI attacks is unavoidable if the sensitive features are highly predictive under the model. When stating the theorem, we use the negative KL-divergence $S_{\text{KL}}(\cdot||\cdot)$ to measure the similarity between two distributions.

Theorem 1. *Let f_1 and f_2 be two models such that for any fixed label $y \in \mathcal{Y}$, $U_{f_1}(x_{ns}, y) \geq U_{f_2}(x_{ns}, y)$. Then, $S_{KL}(p(X_s|y, x_{ns})||p_{f_1}(X_s|y, x_{ns})) \geq S_{KL}(p(X_s|y, x_{ns})||p_{f_2}(X_s|y, x_{ns}))$.*

We omit the proof of the theorem to the supplementary material. Intuitively, highly predictive models are able to build a strong correlation between features and labels, which coincides exactly with what an adversary exploits to launch MI attacks; hence, more predictive power inevitably leads to higher attack performance.

In Yeom et al. (2018), it is argued that a model is more vulnerable to MI attacks if it overfits data to a greater degree. Their result is seemingly contradictory with ours, because fixing the training performance, more overfitting implies that the model has less predictive power. However, the assumption underlying their result is fundamentally different from ours, which leads to the disparities. The result in Yeom et al. (2018) assumes that the adversary has access to the joint distribution $p(X_s, X_{ns}, Y)$ that the private training data is drawn from and their setup of the goal of the MI attack is to learn the sensitive feature associated with a given label in a specific training dataset. By contrast, our formulation of MI attacks is to learn about private feature distribution $p(X_s|y, x_{ns})$ for a given label y from the model parameters. We do not assume that the adversary has the prior knowledge of $p(X_s, X_{ns}, Y)$, as it is a overly strong assumption for our formulation—the adversary can easily obtain $p(X_s|y, x_{ns})$ for any labels and any values of non-sensitive features when having access to the joint distribution.

4 Experiments

4.1 Experimental Setup

Dataset We evalaute our method using three datasets: (1) the MNIST handwritten digit data (`MNIST`), (2) the Chest X-ray Database (Wang et al., 2017) (`ChestX-ray8`), and (3) the CelebFaces Attributes Dataset (`CelebA`) containing 202,599 face images of 10,177 identities with coarse alignment. We crop the images at the center and resize them to 64×64 so as to remove most background.

Protocol We split each dataset into two disjoint parts: one part used as the private dataset to train the target network and the other as a public dataset for prior knowledge distillation. *The public data, throughout the experiments, do not have class intersection with the private training data of the target network.* Therefore, the public dataset in our experiment only helps the adversary to gain knowledge about features generic to all classes and does not provide information about private, class-specific features for training the target network. This ensures the fairness of the comparison with the existing MI attack (Fredrikson et al., 2015).

Models We implement several different target networks with varied complexities. For all the adapted networks, we modify the FC-layer to fit in our task. For digit classification on `MNIST`, our target network consists of 3 convolutional layers and 2 pooling layers. For the disease prediction on `ChestX-ray8`, we use ResNet-18 adapted from (He et al., 2015) as our target network. For the face recognition tasks on `CelebA`, we use the following networks: (1) `VGG16` adapted from (Simonyan and Zisserman, 2014); (2)`ResNet-152` adapted from (He et al., 2015); (3) `face.eoLVe` adapted from the state-of-the-art face recognition network (Cheng et al., 2017).

Training We split the private dataset defined above into training set (90%) and test set (10%) and use the SGD optimizer with learning rate 10^{-2}, batch size 64, momentum 0.9 and weight decay 10^{-4} to train these networks. To train the GAN in the first stage of our attack pipeline, we set $\lambda_d = 0.5$ and use the Adam optimizer with the learning rate 0.004, batch size 64, $\beta_1 = 0.5$, and $\beta_2 = 0.999$ (Kingma and Ba, 2014). In the second stage, we set $\lambda_i = 100$ and use the SGD optimizer to optimize the latent vector z with the learning rate 0.01, batch size 64 and momentum 0.9. z is drawn from a zero-mean unit-variance Gaussian distribution. We randomly initialize z for 5 times and optimize each round for 1500 iterations. We choose the solution with the lowest identity loss as our final latent vector.

4.2 Evaluation Metrics

Evaluating the success of MI attacks requires to assess whether the recovered image exposes the private information about a target individual. Previous works analyzed the attack performance mainly qualitatively by visual inspection. Herein, we introduce four metrics which allow to quantitatively judge the MI attack efficacy and perform evaluation at a large scale.

Peak Signal-to-Noise Ratio (PSNR) PSNR is the ratio of an image's maximum squared pixel fluctuation over the mean squared error between the target image and the reconstructed image Hore and Ziou (2010). PSNR measures the pixel-wise similarity between two images. The higher the PSNR, the better the quality of the reconstructed image.

However, oftentimes, the reconstructed image may still reveal identity information even though it is not close to the target image pixel-wise. For instance, a recovered face with different translation, scale and rotation from the target image will still incur privacy loss. This necessitates the need for the following metrics that can evaluate the similarity between the reconstructed and the target image at a semantic level.

Attack Accuracy (Attack Acc) We build an *evaluation classifier* that predicts the identity based on the input reconstructed image. If the evaluation classifier achieves high accuracy, the reconstructed image is considered to expose private information about the target individual. The evaluation classifier should be different from the target network because the reconstructed images may incorporate features that overfit the target network while being semantically meaningless. Moreover, the evaluation classifier should be highly performant. For the reasons above, we adopt the state-of-the-art architecture in each task as the evaluation classifier. For `MNIST`, our evaluation network consists of 5 convolutional layers and 2 pooling layers. For `ChestX-ray8`, we adapt VGG-19 from (Simonyan and Zisserman, 2014) as our evaluation network. For `CeleA`, we use the model in (Cheng et al., 2017) for the evaluation classifier. We first pretrain it on the `MS-Celeb-1M` (Guo et al., 2016) and then fine tune on the identities in the training set of the target network. The resulting evaluation classifier can achieve 96% accuracy on these identities.

Feature Distance (Feat Dist) Feat Dist measures the l_2 feature distance between the reconstructed image and the centroid of the target class. The feature space is taken to be the output of the penultimate layer of the evaluation network.

K-Nearest Neighbor Distance (KNN Dist) KNN Dist looks at the shortest distance from the reconstructed image to the target class. We identify the closest data point to the reconstructed image in the training set and output their distance. The distance is measured by the l_2 distance between the two points in the feature space of the evaluation classifier.

Table 1: Comparison of the proposed GMI attack with the existing MI attack in (Fredrikson et al., 2015) (EMI), when the attacker does not have any auxiliary knowledge about the target image.

		KNN Dist	Feat Dist	Attack Acc	Top-5 Attack Acc
VGG16	**EMI**	2397.50	2255.54	0	0
	GMI	**2098.92**	**2012.10**	**28**	**53**
ResNet-152	**EMI**	2422.99	2288.13	0	1
	GMI	**1969.09**	**1886.44**	**44**	**72**
face.evolve	**EMI**	2371.52	2248.81	0	1
	GMI	**1923.72**	**1802.62**	**46**	**76**

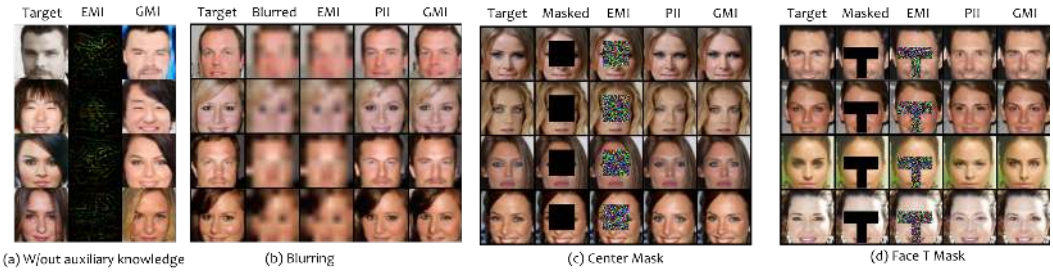

Figure 3: Qualitative comparison of the proposed GMI attack with the existing MI attack (EMI), the pure image inpainting method (PII). The ground truth target image is shown in 1st col.

4.3 Experimental Results

We compare our approach with two baselines: (1) Existing model inversion attack (EMI), which implements the algorithm in (Fredrikson et al., 2015). For this algorithm, the adversary only exploits the identity loss for image reconstruction and return the pixel values that minimize the the identity loss; (2) Pure image inpainting (PII), which minimizes the W-GAN loss and performs image recovery based on the information completely from the public dataset.

4.3.1 Attacking Face Recognition Classifiers

For CelebA, the private set comprises 21,152 images of 1000 identities and samples from the rest are used as a public dataset. We evaluate the attack performance in the three settings: (1) the attacker does not have any auxiliary knowledge about the private image, in which case he will recover the image from scratch; (2) the attacker has access to a blurred version of the private image and his goal is to deblur the image; (3) the attacker has access to a corrupted version of the private image wherein the sensitive, identity-revealing features (e.g., nose, mouth, etc) are blocked.

Table 1 compares the performance of our proposed GMI attack against EMI for different network architectures. We can see that the EMI works poorly on the deep nets and achieve around zero attack accuracy. GMI is much more effective than EMI. Particularly, our method improves the accuracy of the attack against the state-of-the-art face.evoLVe classifier over the existing MI attack by 75% in terms of Top-5 attack accuracy. Also, note that models that are more sophisticated and have more predictive power are more susceptible to attacks. We will examine this phenomenon in more details in Section 4.3.3.

We now discuss the case where the attacker has access to some auxilliary knowledge in terms of blurred or partially blocked images. For the latter, we consider two types of masks—center and face "T", illustrated by the second column of Figure 3 (c) and (d), respectively. The center mask blocks the central part of the face and hides most of the identity-revealing features, such as eyes and nose, while the face T mask is designed to obstruct all private features in a face image.

Table 2 shows that our method consistently outperforms the two baselines discussed above. Since the existing MI attack does not exploit any prior information, the inversion optimization problem is extremely ill-posed and performing gradient descent ends up at some visually meaningless local minimum, as illustrated by Figure 3. Interestingly, despite having the meaningless patterns, these images can all be classified correctly into the target label by the target network. Hence, *the existing*

Table 2: Comparison of the proposed GMI attack with the existing MI attack in (Fredrikson et al., 2015) (EMI), a pure image inpainting (PII) method that recovers the private image based only on the public dataset.

		Blurring			Center Mask			Face T mask		
		EMI	**PII**	**GMI**	**EMI**	**PII**	**GMI**	**EMI**	**PII**	**GMI**
VGG16	**PSNR**	19.66	20.78	**21.97**	18.69	25.49	**27.58**	19.77	24.05	**26.79**
	Feat Dist	2073.56	2042.99	**1904.56**	1651.72	1866.07	**1379.26**	1798.85	1838.31	**1655.35**
	KNN Dist	2164.40	2109.82	**1946.97**	1871.21	1772.74	**1414.37**	1980.68	1916.67	**1742.74**
	Attack Acc	0%	6%	**43%**	14%	34%	**78%**	11%	20%	**58%**
ResNet-152	**PSNR**	19.63	20.78	**22.00**	18.69	25.49	**27.34**	19.89	24.05	**26.64**
	Feat Dist	2006.46	2042.99	**1899.79**	1635.03	1866.07	**1375.36**	1641.31	1838.31	**1594.81**
	KNN Dist	2101.13	2109.82	**1922.14**	1859.78	1772.74	**1403.24**	1847.74	1916.67	**1670.05**
	Attack Acc	1%	6%	**50%**	9%	34%	**80%**	11%	20%	**63%**
face.evoLVe	**PSNR**	19.64	20.78	**22.04**	18.97	25.49	**27.69**	19.86	24.05	**25.77**
	Feat Dist	1997.93	2042.99	**1878.38**	1609.35	1866.07	**1364.42**	1762.57	1838.31	**1624.95**
	KNN Dist	2085.53	2109.82	**1904.47**	1824.10	1772.74	**1403.19**	1962.07	1916.67	**1682.56**
	Attack Acc	1%	6%	**51%**	12%	34%	**82%**	11%	20%	**64%**

Table 3: Evaluation for the impact of public datasets on the attack accuracy.

	CelebA→CelebA				PubFig83→CelebA		EMI
	1:1	**1:4**	**1:6**	**1:10**	**W/o Preproc.**	**W/ Preproc.**	
VGG	78%	77%	75%	72%	48%	67%	14%
LeNet	81%	75%	77%	75%	52%	66%	9%
face.evoLVe	77%	77%	77%	70%	56%	70%	12%

MI attack tends to generate "adversarial examples" that can fool the target network but does not exhibit any recognizable features of the private data. Figure 3 also compares our results with PII, which is completely based on the information from the public dataset to recover the private image. We can see that although PII leads to realistic recoveries, the reconstructed images do not present the same identity features as the target images. This can be further corroborated by the quantitative results in Table 2. Note that the attacks are more effective for the center mask than the face T mask. This is because the face T mask we designed completely hides the identity revealing features on the face while the center mask may still expose the mouth information.

4.3.2 Impact of Public Knowledge

We have seen that distilling prior knowledge and properly incorporating it into the attack algorithm are important to the success of MI attacks. In our proposed method, the prior knowledge is gleaned from public datasets through GAN. We now evaluate the impact of public datasets on the attack performance.

We first consider the case where the public data is from the same distribution as the private data and study how the size of the public data affects the attack performance. We change the size ratio (1:1, 1:4, 1:6, 1:10) of the public over the private data by varying the number of identities in the public dataset (1000, 250, 160, 100). As shown in Table 3, the attack performance varies by less than 7% when shrinking the public data size by 10 times.

Moreover, we study the effect of the distribution shift between the public and private data on the attack performance. We train the GAN on the PubFig83 dataset, which contains 13,600 images with 83 identities, and attack the target network trained on CelebA. There are more faces with sunglasses in PubFig83 than CelebA, which makes it harder to distill generic face information. Without any pre-processing, the attack accuracy drops by more than 20% despite still outperforming the existing MI attack by a large margin. To further improve the reconstruction quality, we detect landmarks in the face images, rotate the images such that the eyes lie on a horizontal line, and crop the faces to remove the background. These pre-processing steps make the public datasets better present the face information, thus improving the attack accuracy significantly.

4.3.3 Attacking Models with Different Predictive Powers

We perform experiments to validate the connection between predictive power and the vulnerability to MI attacks. We measure the predictive power of sensitive feature under a model using the difference of model testing accuracy based on all features and just non-sensitive features. We consider the following

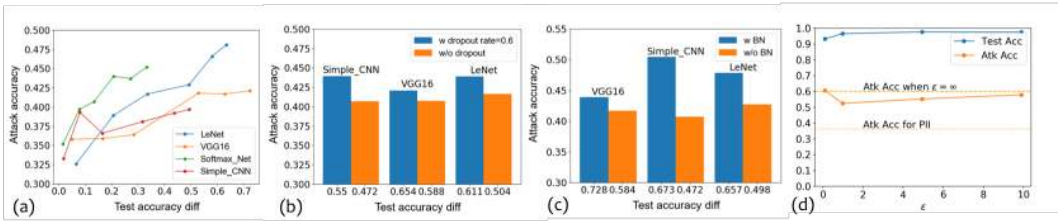

Figure 4: (a)-(c): The performance of the GMI attack against models with different predictive powers by varying training size, dropout, and batch normalization, respectively. (d) Attack accuracy of the GMI attack against models with different DP budgets. Attack accuracy of `PII` is plotted as a baseline.

	KNN Dist	Feat Dist	Attack Acc
EMI	31.60	82.69	40%
GMI	**4.04**	**16.17**	**80%**

Table 4: Comparing the GMI against the EMI attack on MNIST.

different ways to construct models with increasing feature predictive powers, namely, enlarging the training size per class, adding dropout regularization, and performing batch normalization. For the sake of efficiency, we slightly modify the proposed method in Section 2.2 in order to avert re-training GANs for different architectures. Specifically, we exclude the diversity loss from the attack pipeline so that multiple architectures can share the same GAN for prior knowledge distillation. Figure 4 shows that, in general, the attack performance will be better for models with higher feature predictive powers. Moreover, this trend is consistent across different architectures.

4.3.4 Attacking Digit Classifiers

For `MNIST`, we use all 34265 images with labels $5, 6, 7, 8, 9$ as private set, and the rest of 35725 images with labels $0, 1, 2, 3, 4$ as a public dataset. Note that the labels in the private and public data have no overlaps. We augment the public data by training an autoencoder and interpolating in the latent space. Our GMI attack is compared with the baseline in Table 4. We omit the PII baseline because the public and private set defined in this experiment are rather disparate and the PII essentially produce results close to random guesses. We can see from the table that the performance of GMI is significantly better than the EMI. Examples of the recovered images with both attacks are compared in Figure 5.

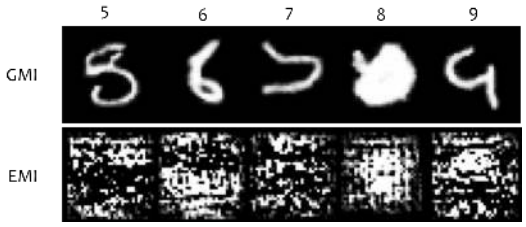

Figure 5: Visualization of the recovered input images by the GMI and the EMI attack.

4.3.5 Attacking Disease Predictors

For `ChestX-ray8`, we use 10000 images of seven classes as the private data and the other 10000 with different labels as public data. The GMI and EMI attack are compared in Table 5. Again, the GMI attack outperforms the EMI attack by a large margin.

	KNN Dist	Feat Dist	Attack Acc
EMI	130.19	155.65	14%
GMI	**63.42**	**93.68**	**71%**

Table 5: Comparing the GMI against the EMI attack on ChestX-ray8.

4.3.6 ATTACKING DIFFERENTIALLY PRIVATE MODELS

We investigate the implications of DP for MI attacks. (ϵ, δ)-DP is ensured by adding Gaussian noise to clipped gradients in each training iteration Abadi et al. (2016). We find it challenging to produce useful face recognition models with DP guarantees due to the complexity of the task. Therefore, we turn to a simpler dataset, `MNIST`, which is commonly used in differential private ML studies. We set $\delta = 10^{-5}$ and vary the noise scale to obtain target networks with different ϵ. The attack performance against these target networks and their utility are illustrated in Figure 4 (d). Since the attack accuracy of the GMI attack on differentially private models is higher than that of `PII` which fills missing regions completely based on the public data, it is clear that the GMI attack can expose private information from differentially private models, even with stringent privacy guarantees, like $\epsilon = 0.1$. Moreover, varying differential privacy budgets helps little to protect against the GMI attack; sometimes, more privacy budgets even improve the attack performance (e.g., changing ϵ from 1 to 0.1). This is because DP, in its canonical form, only hides the presence of a single instance in the training set. Limiting the learning of specific individuals may facilitate the learning of generic features of a class, which, in turn, helps to stage MI attacks.

5 CONCLUSION

In this paper, we present a generative approach to MI attacks, which can achieve the-state-of-the-art success rates for attacking the DNNs with high-dimensional input data. The idea of our approach is to extract generic knowledge from public datasets via GAN and use it to regularize the inversion problem. Our experimental results show that our proposed attack is highly performant even when the public datasets (1) do not include the identities that the adversary aims to recover, (2) are unlabeled, (3) have small sizes, (4) come from a different distribution from the private data. We also provide theoretical analysis showing the fundamental connection between a model's predictive power and its vulnerability to inversion attacks. For future work, we are interested in extending the attack to the black-box setting and studying effective defenses against MI attacks.

REFERENCES

M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

D. Carrell, B. Malin, J. Aberdeen, S. Bayer, C. Clark, B. Wellner, and L. Hirschman. Hiding in plain sight: use of realistic surrogates to reduce exposure of protected health information in clinical text. *Journal of the American Medical Informatics Association*, 20(2):342–348, 2012.

Y. Cheng, J. Zhao, Z. Wang, Y. Xu, K. Jayashree, S. Shen, and J. Feng. Know you at one glance: A compact vector representation for low-shot learning. In *ICCVW*, pages 1924–1932, 2017.

C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 17–32, 2014.

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

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

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

S. Hidano, T. Murakami, S. Katsumata, S. Kiyomoto, and G. Hanaoka. Model inversion attacks for prediction systems: Without knowledge of non-sensitive attributes. In *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, pages 115–11509. IEEE, 2017.

A. Hore and D. Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369. IEEE, 2010.

S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):107, 2017.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278 – 2324, 12 1998. doi: 10.1109/5.726791.

F. Li, Z. Sun, A. Li, B. Niu, H. Li, and G. Cao. Hideme: Privacy-preserving photo sharing on social networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 154–162. IEEE, 2019.

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)*, pages 3–18. IEEE, 2017.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014.

X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers. 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 conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

X. Wu, M. Fredrikson, S. Jha, and J. F. Naughton. A methodology for formalizing model-inversion attacks. In *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, pages 355–370. IEEE, 2016.

D. Yang, S. Hong, Y. Jang, T. Zhao, and H. Lee. Diversity-sensitive conditional generative adversarial networks. *arXiv preprint arXiv:1901.09024*, 2019a.

Z. Yang, E.-C. Chang, and Z. Liang. Adversarial neural network inversion via auxiliary knowledge alignment. *arXiv preprint arXiv:1902.08552*, 2019b.

R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5485–5493, 2017.

S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

A Proof of Theorem 1

Theorem 2. *Let f_1 and f_2 are two models such that for any fixed label $y \in \mathcal{Y}$, $U_{f_1}(x_{ns}, y) \geq U_{f_2}(x_{ns}, y)$. Then, $S_{KL}(p(X_s|y, x_{ns})||p_{f_1}(X_s|y, x_{ns})) \geq S_{KL}(p(X_s|y, x_{ns})||p_{f_2}(X_s|y, x_{ns}))$.*

Proof. We can expand the KL divergence $D_{\text{KL}}(p(X_s|y, x_{ns})||p_{f_1}(X_s|y, x_{ns})$ as follows.

$$D_{\text{KL}}(p(X_s|y, x_{ns})||p_{f_1}(X_s|y, x_{ns})) \tag{5}$$
$$= \mathbb{E}_{X\sim p(X_s|y,x_{ns})}[\log p(X_s|y, x_{ns})] - \mathbb{E}_{X\sim p(X_s|y,x_{ns})}[\log p_{f_1}(X_s|y, x_{ns})] \tag{6}$$

Thus,

$$D_{\text{KL}}(p(X_s|y, x_{ns})||p_{f_1}(X_s|y, x_{ns})) - D_{\text{KL}}(p(X_s|y, x_{ns})||p_{f_2}(X_s|y, x_{ns})) \tag{7}$$

$$= \mathbb{E}_{X\sim p(X_s|y,x_{ns})}[\log p_{f_2}(X_s|y,x_{ns}) - \log p_{f_1}(X_s|y,x_{ns})] \tag{8}$$

$$= \sum_x p(X_s|y,x_{ns})\big(\log \frac{p_{f_2}(y|X_s,x_{ns})p(X_s|x_{ns})}{p_{f_2}(y|x_{ns})} - \log \frac{p_{f_1}(y|X_s,x_{ns})p(X_s|x_{ns})}{p_{f_1}(y|x_{ns})}\big) \tag{9}$$

$$= \sum_x p(X_s|y,x_{ns})\Bigg(\big(\log p_{f_2}(y|X_s,x_{ns}) - \log p_{f_2}(y|x_{ns})\big) - \big(\log p_{f_1}(y|X_s,x_{ns}) - \log p_{f_1}(y|x_{ns})\big)\Bigg) \tag{10}$$

$$= U_{f_2}(x_{ns},y) - U_{f_1}(x_{ns},y) \leq 0 \tag{11}$$

□

B Experimental Details

B.1 Network Architecture

The detailed architectures for the two encoders, the decoder of the generator, the local discriminator, and the global discriminator are presented in Table 6, Table 7, Table 8, Table 9, and Table 10, respectively.

Table 6: The encoder of the generator that takes as input the corrputed RGB image and the binary mask.

Type	Kernel	Dilation	Stride	Outputs
conv.	5x5	1	1x1	32
conv.	3x3	1	2x2	64
conv.	3x3	1	1x1	128
conv.	3x3	1	2x2	128
conv.	3x3	1	1x1	128
conv.	3x3	1	1x1	128
conv.	3x3	2	1x1	128
conv.	3x3	4	1x1	128
conv.	3x3	8	1x1	128
conv.	3x3	16	1x1	128

Table 7: The encoder of the generator that takes as input the latent vector.

Type	Kernel	Stride	Outputs
linear			8192
deconv.	5x5	1/2 x 1/2	256
deconv.	5x5	1/2 x 1/2	128

Table 8: The decoder of the generator.

Type	Kernel	Stride	Outputs
deconv.	5x5	1/2 x 1/2	128
deconv.	5x5	1/2 x 1/2	64
conv.	3x3	1x1	32
conv.	3x3	1x1	3

(1) `LeNet` adapted from (Lecun et al., 1998), which has three convolutional layers, two max pooling layers and one FC layer; (2) `SimpleCNN`, which has five convolutional layers, each followed by a batch normalization layer and a leaky ReLU layer; (3) `SoftmaxNet`, which has only one FC layer.

Table 9: The global Discriminator.

Type	Kernel	Stride	Outputs
conv.	5x5	2x2	64
conv.	5x5	2x2	128
conv.	5x5	2x2	256
conv.	5x5	2x2	512
conv.	1x1	4x4	1

Table 10: The local Discriminator.

Type	Kernel	Stride	Outputs
conv.	5x5	2x2	64
conv.	5x5	2x2	128
conv.	5x5	2x2	256
conv.	1x1	4x4	1

B.2 The Detailed Setting of the Experiments on "Attacking Differentially Private Models"

We split the `MNIST` dataset into the private set used for training target networks with digits $0 \sim 4$ and the public set used for distilling prior knowledge with digits $5 \sim 9$. The target network is implemented as a Multilayer Perceptron with 2 hidden layers, which have 512 and 256 neurons, respectively. The evaluation classifier is a convulutional neural network with three convolution layers, followed by two fully-connected layers. It is trained on the entire MNIST training set and can achieve 99.2% accuracy on the `MNIST` test set.

Differential privacy of target networks is guaranteed by adding Gaussian noise to each stochastic gradient descent step. We use the moment accounting technique to keep track of the privacy budget spent during training (Abadi et al., 2016). During the training of the target networks, we set the batch size to be 256. We fix the number of epochs to be 40 and clip the L2 norm of per-sample gradient to be bounded by 1.5. We set the ratio between the noise scale and the gradient clipping threshold to be $0, 0.694, 0.92, 3, 28$, respectively, to obtain the target networks with $\varepsilon = \infty, 9.89, 4.94, 0.98, 0.10$ when $\delta = 10^{-5}$. For model with $\varepsilon = 0.1$, we use the SGD with a small learning rate 0.01 to ensure stable convergence; otherwise, we set the learning rate to be 0.1.

The architecture of the generator in Section B.1 is tailored to the `MNIST` dataset. We reduce the number of input channels, change the size of kernels, and modify the layers of discriminators to be compatible with the shape of the `MNIST` data. To train the GAN in the first stage of our GMI attack, we set the batch size to be 64 and use the Adam optimizer with the learning rate 0.004, $\beta_1 = 0.5$, and $\beta_2 = 0.999$ (Kingma and Ba, 2014). For the second stage, we set the batch size to be 64 and use the SGD with the Nesterov momentum that has the learning rate 0.01 and momentum 0.9. The optimization is performed for 1500 iterations.

The center mask depicted in the main text is used to block the central part of digits. We report the attack accuracy averaged across 640 randomly sampled images from the private set and 5 random initializations of the latent vector for each sampled image.