IMPROVED TECHNIQUES FOR MODEL INVERSION ATTACKS

Si Chen
Department of Electrical and Computer Engineering
Virginia Tech
chensi@vt.edu

Ruoxi Jia
Department of Electrical and Computer Engineering
Virginia Tech
ruoxijia@vt.edu

Guo-Jun Qi
Futurewei Technologies
guojun.qi@futurewei.com

October 12, 2020

ABSTRACT

Model inversion (MI) attacks in the whitebox setting are aimed at reconstructing training data from model parameters. Such attacks have triggered increasing concerns about privacy, especially given a growing number of online model repositories. However, existing MI attacks against deep neural networks (DNNs) have large room for performance improvement. A natural question is whether the underperformance is because the target model does not memorize much about its training data or it is simply an artifact of imperfect attack algorithm design? This paper shows that it is the latter. We present a variety of new techniques that can significantly boost the performance of MI attacks against DNNs. Recent advances to attack DNNs are largely attributed to the idea of training a general generative adversarial network (GAN) with potential public data and using it to regularize the search space for reconstructed images. We propose to customize the training of a GAN to the inversion task so as to better distill knowledge useful for performing attacks from public data. Moreover, unlike previous work that directly searches for a single data point to represent a target class, we propose to model private data distribution in order to better reconstruct representative data points. Our experiments show that the combination of these techniques can lead to state-of-the-art attack performance on a variety of datasets and models, even when the public data has a large distributional shift from the private data.

1 Introduction

Many attractive applications of machine learning techniques involve training models on sensitive and proprietary datasets. One major concern for these applications is that models could be subject to privacy attacks and reveal inappropriate details of the training data. One type of privacy attacks is MI attacks, aimed at recovering training data form the access to a model. The access could either be black-box or white-box. In the blackbox setting, the attacker can only make prediction queries to the model, while in the whitebox setting, the attacker has complete knowledge of the model. Given a growing number of online platforms where users can download entire models, such as Tensorflow Hub and ModelDepot, whitebox MI attacks have posed an increasingly serious threat to privacy.

Effective MI attacks have been mostly demonstrated on simple models, such as linear models, and low-dimensional feature space [1, 2]. Recent work [3] has proposed the most effective MI attack against DNNs thus far by distilling a generic prior from public data via a GAN and use the GAN to guide the inversion process. However, there still exists a large room to improve the attack performance. For instance, the top-one identification accuracy of face images inverted from the state-of-the-art face recognition classifier is 45% and further decreases when the public data has a large distributional shift from the private data. A natural question is: *Is the underperformance of MI attacks against DNNs because DNNs do not memorize much about private data or it is simply an artifact of imperfect attack algorithm design?*

This paper studies a variety of new techniques to improve the MI attacks against DNNs. Unlike existing work which applies the canonical training procedure of GANs to distill knowledge from public data, we propose to tailor the training objective to the inversion task. Specifically, for the discriminator side, we propose to leverage the target model to label the public dataset and train the discriminator to differentiate not only the real and fake samples but the labels; for the generator side, we propose to maximize the class prediction confidence of inverted samples via an entropy loss. In addition, we propose to explicitly parametrize the private data distribution in order to better reconstruct a representative data point for a given target class. We present a differentiate loss to optimize the parameters of the distribution via backpropogation. We perform experiments on various datasets and network architectures and show that the combination of these techniques could lead to state-of-the-art performance to attack DNNs, even when the public data used for knowledge distillation has a large distributional shift from the private data.

2 Related work

The general goal of privacy attacks against machine learning models is to gain knowledge which is not intended to be shared, such as knowledge about the training data or information about the model. Attacks can be categorized into four types according to the specific goals: model extraction, membership inference, property inference, and model inveresion. Model extraction attacks [5, 6, 7, 8] try to create a substitute model that learns same task as the target model while performing equally good or even better. The other three kinds of attack focus on exposing secrets about training data: membership inference attacks[4] try to determine whether a given datapoint is used as part of the training set; property inference attacks [9, 10, 11] try to extract dataset properties which are not explicitly correlated to the learning task (e.g., extracting the ratio of women and men in a patient dataset where this information is unlabeled). The goal of MI attacks is to recreate training data or sensitive attributes.

The first MI attack algorithm was proposed in [1], which follows the Maximum a Posterior (MAP) principle and constructs the input feature that maximizes the likelihood of observing a given model response and other possible auxiliary information. The authors applied the algorithm to attacking a linear regression model that predicts medical dosage and showed that the algorithm can successfully invert genetic markers which are used as part of the input features. [2] applied the MAP attack idea to more complex models, including decision trees and shallow neural networks. Specifically, for neural networks with high-dimensional input features, the authors proposed to utilize gradient descent to solving the optimization problem underneath the attack. Although the algorithm significantly outperforms random guessing when tested on some shallow networks and single-channel images, the reconstructions are blurry and can hardly reveal private information. Besides, the algorithm completely fails when tested on DNNs and three-channel images.

To improve the attack performance for DNNs with high-dimensional input, [3] proposed a two-pronged attack approach which first trains a GAN on public data (which could have no class intersection with private data and no labels), and then uses the GAN to impose a distributional prior for the search space of the attack optimization. The authors showed that this approach can achieve the state-of-the-art performance for attacking various DNNs. Despite the significant improvement over existing baselines, this approach still has large room to be improved. For instance, the face images reconstructed from the state-of-the-art face recognition model can be identified as the target individual with only a success rate of 46%. Inspired by this work, we also leverage GANs to regularize the search space for the attack optimization problem; however, instead of applying the generic training algorithm, we propose customized training of GAN that can distill private knowledge from the target network. We also propose to model the distribution of private data and learn it from end-to-end. We show that the combination of the techniques can greatly improve the attack performance over [3].

3 Our Approach

3.1 Preliminaries

Attack model. This paper focuses on the whitebox MI attack, in which the attacker has complete access to the target network f. The goal of the attacker is to discover the distribution of input feature X associated with a specific label y. We will use face recognition as a running example for the target network. Face recognition classifiers label an image containing a face with the label corresponding to the identity depicted in the image. The corresponding attack goal is to recover the face images for some specific identity based on the target classifier parameters.

Background on one-to-one MI attack. Existing MI attacks boil down to synthesizing the most likely input for the target network. Specifically, the following optimization problem is solved to synthesize the input for a given label y: $\max_x \log T_y(x)$, where $T_y(x)$ is the probability of label y output by the model T given the input x. When T is a

deep neural network and x is high-dimensional (e.g., images), the corresponding optimization becomes nonconvex and performing gradient descent easily gets stuck in local minima. The local minima might not be semantically meaningful at all. For instance, when the model input is an image, such local minima could be meaningless patterns of pixels. The current state-of-the-art approach to attacking DNNs [3] addressed the challenge by extracting general information about the private data distribution from public data and leveraging the information to regularize the attack optimization problem. Consider the example of attacking face recognition classifiers. There exist ample face images on the internet, which may not contain the target individuals but still provide rich knowledge about how a face image might be structured. Extracting such information is beneficial to synthesize realistic face images. Particularly, the existing work adopts a two-step attack alogrithm: The first step is to train a GAN on public data using the caononical WGAN loss [13]; and the second step is to minimize the following loss which seeks for the synthesized input with high likelihood to produce the target label while remaining realistic: $\max_{x=G(z)} D(x) + \log T_y(x)$, where D is the discriminator which outputs the probability that x comes from the real data and G is the generator.

3.2 One-To-Many Model Inversion Attack Algorithm

Inspired by [3], our proposed attack algorithm consists of two steps. The first step is also to extract the general information related to private data distribution from public data. However, instead of training a generic GAN, we customize the training objective for both generator and discriminator so as to maximally distill the information related to the private domain from public data. In the second step, we make use of the generator learned in the first step and estimate the parameters of the private data distribution. The overall architecture of our method is shown in Figure 1.

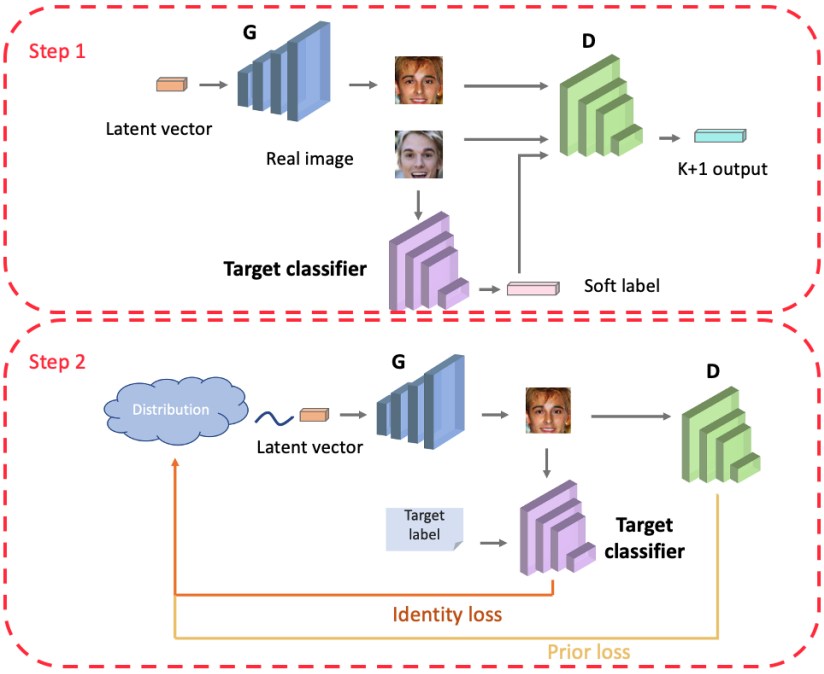

Figure 1: Overall architecture of the proposed attack algorithm. *Step 1.* Build an inversion-specific GAN to distill private information. *Step 2.* Recover the distribution of private domain. Note that both the generator and discriminator are fixed at Step 2.

Step 1: Building an Inversion-Specific GAN. To make the distribution learned from public data more attuned to the private domain, we propose to adopt a discriminator that is able to differentiate not only real data from the fake but the class labels associated with the target network. Suppose that the target network classifies a data point into one of K possible classes. Our discriminator D is a $(K+1)$-classifier, where the first K classes correspond to the labels of the target network and the $(K+1)$-th class represents fake samples. To train such a discriminator, we use the target network T to generate a soft label $T(x)$ for each image from the public dataset.

Formally, the training loss for D has two parts:

$$L_D = L_{\text{supervised}} + L_{\text{unsupervised}} \tag{1}$$

where

$$L_{\text{supervised}} = -\mathbb{E}_{x\sim p_{\text{data}}(x)} \sum_{k=1}^{K} T_k(x) \log p_{\text{disc}}(y = k \mid x) \tag{2}$$

and

$$L_{\text{unsupervised}} = -\{\mathbb{E}_{x\sim p_{\text{data}}(x)} \log D(x) + \mathbb{E}_{z\sim \text{noise}} \log(1 - D(G(z)))\} \tag{3}$$

where p_{data} is the distribution of public data, $p_{\text{disc}}(y|x)$ is the probability that the discriminator predicts x as class y, and $T_k(x)$ is the k-th dimension of the soft label produced by the target network. $D(x)$ outputs the probability of x being a real sample and therefore $D(x) = p_{\text{disc}}(y < K + 1|x)$.

Intuitively, using these data with soft-labels to train the discriminator will encourage the generator to produce image statistics that help infer what class an image represents. Such image statistics are also likely to present in the private training data distribution. Hence, the proposed training process can potentially guide the generator to produce images that share more common characteristics with the private training data.

The proposed training process partially resembles the process of using GAN to perform semi-supervised learning [16], which also leverages a classifier augmented with a new class of fake samples as the discriminator. The proposed training process differs from [16] in that we use soft-labels generated by the target network to train the discriminator, whereas in the semi-supervised learning, there already exists a small set of labeled data instances that can be used to train the discriminator directly.

For training the generator, we introduce an entropy regularizer into the canonical feature matching-based training objective [16]:

$$L_G = \|\mathbb{E}_{x\sim p_{\text{data}}} \mathbf{f}(x) - \mathbb{E}_{z\sim \text{noise}}\, \mathbf{f}(G(z))\|_2^2 + \lambda_h L_{\text{entropy}} \tag{4}$$

where $\mathbf{f}(x)$ denotes activation on an intermediate layer of the discriminator and

$$L_{\text{entropy}} = \mathbb{H}(p_{\text{disc}}(1 \le y \le K|G(z))) = -\sum_{k=1}^{K} p_{\text{disc}}(y = k|G(z)) \log p_{\text{disc}}(y = k|G(z)). \tag{5}$$

The intuition of the entropy regularization term is simple. Because the target network is trained on the private data, the private data should have high confidence when fed into the target network and in turn should get low prediction entropy. In order to encourage the data distribution learned from public data to mimic the private data, we explicitly constrain the entropy in the loss function so that the generated data will have low entropy under the target network.

Step 2: Distributional Recovery. Given the GAN trained above on the public data under the guidance of the target network, the second step of the attack tries to find a model for the private data distribution which achieves maximum likelihood under the target network while containing realistic images. Specifically, we model the private data distribution by $G(z)$, where G is the generator trained in the first step and $z \sim p_{\text{gen}} := \mathcal{N}(\mu, \Sigma)$. We then minimize the following objective function to generate the samples of class k from the private classifier T by estimating μ and Σ:

$$L = L_{\text{prior}} + \lambda_i L_{\text{id}} \tag{6}$$

where

$$L_{\text{prior}} = -\mathbb{E}_{z\sim p_{\text{gen}}} \log D(G(z)) \tag{7}$$

$$L_{\text{id}} = -\mathbb{E}_{z\sim p_{\text{gen}}} T_k(G(z)) \tag{8}$$

The prior loss L_{prior} penalizes unrealistic images and the identity loss L_{id} encourages the estimated private data distribution to have high likelihood of being assigned to a given target label k under the targeted network T.

We adopt the reparameterization trick [14] to make L_{prior} and L_{id} differentiable:

$$z = A\epsilon + b, \epsilon \sim \mathcal{N}(0, I) \tag{9}$$

We can now form Monte Carlo estimates of expectations of L_{prior} and L_{id} as follows and optimize them with respect to A and b:

$$L_{\text{prior}} = -\frac{1}{L} \sum_{l=1}^{L} \log D(G(A\epsilon_l + b)) \tag{10}$$

$$L_{\text{id}} = -\frac{1}{L} \sum_{l=1}^{L} \log T_k(G(A\epsilon_l + b)) \tag{11}$$

where $\epsilon_l \sim \mathcal{N}(0, I)$ for $l = 1, \ldots, L$.

4 Experiment

In this section, we will evaluate our proposed attack in terms of the performance to recover a representative input. The baseline that we will compare against is the generative MI attack (GMI) proposed in [3], which achieved the-state-of-the-art result to attack DNNs.

4.1 Experimental setting

Dataset. We study attacks against models built for different prediction tasks, including face recognition, digit classification, object classification, and disease prediction. For face recognition, we use (1) the CelebFaces Attributes Dataset [44] (CelebA) containing 202,599 face images of 10,177 identities with coarse alignment, (2) Flickr-Faces-HQ (FFHQ) Dataset containing 70,000 high-quality images with considerable variation in terms of age, ethnicity and image background, and (3) FaceScrub consisting of 106,863 face images of male and female 530 celebrities, with about 200 images per person. We use aligned versions of above face datasets, and crop the images at the center and resize them to 64×64 so as to remove most background. For digit classification, we use the MNIST handwritten digit data [36]. For object classification, we adopt the CIFAR-10 dataset [27]. For disease prediction, we use the Chest X-ray Database [42] (ChestX-ray8).

Models. Following the settings in [3], we implement several different target networks with varied complexities. Some of the networks are adapted from existing ones by adjusting the number of outputs of their last fully connected layer to our tasks. For the face recognition task, we use three different network architectures: (1) VGG16 adapted from [20]; (2) ResNet-152 adapted from [21]; (3) face.evoLve adpated from [24]. For digit classification on MNIST, we use a network which consists of 3 convolutional layers and 2 pooling layers. For object classification, we use VGG16. For the disease prediction on ChestX-ray8, we use Resnet-18 adapted from [21].

Attack Implementation. 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. *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. For CelebA, we use 30,027 images of the first 1000 identities as private set and randomly choose 30,000 images of other identities as public set to train GAN. For MNIST and CIFAR10, we use all of the images with label 0, 1, 2, 3, 4 as private set and rest images with label 5, 6, 7, 8, 9 as public set. For ChestX-ray8, we 10,000 images with label "Atelectasis", "Cardiomegaly", "Effusion", "Infiltration", "Mass", "Nodule", "Pneumonia" as private set and 10,000 images belongs to other 7 classes as public set. We train the target networks using SGD optimizer with learning rate 10^{-2}, batch size 64, momentum 0.9 and weight decay 10^{-4}. For training GANs, we use Adam optimizer with learning rate 0.004, batch size 64, $\beta_1 = 0.5$ and $\beta_2 = 0.999$ as [19]. The weight for entropy regularization term is $\lambda_h = 1\text{e}{-4}$. For the step of distribution recovery, we set $\lambda_i = 100$. The distribution is initialized with $\mu = 0, \Sigma = 1$ and optimized for 1500 iterations.

Evaluation Protocol. For our proposed attack, we draw 5 random samples of ϵ and generate corresponding images $G(A\epsilon + b)$. For the baseline attack, we re-start the attack for 5 times with random initialization. To evaluate the reconstruction of a representative input, we compute the average of attack performance on the 5 reconstructed images.

Evaluation Metrics. Evaluating the MI attack performance requires gauging the amount of private information about a target label leaked through the synthesized images. We conduct both qualitative evaluation through visual inspection as well as quantitative evaluation. The quantitative metrics that we use to evaluate the attack performance largely follow the existing literature, including attack accuracy and K-nearest neighbor feature distance. They are generally aimed at measuring the semantic similarity between private data and reconstructions. In addition, we incorporate a metric for image quality, namely, Fréchet Inception Distance (FID) [22], as part of our evaluation. The metrics are expounded as follows.

- 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 label. 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 achieve high performance. The attack accuracy is measured by the prediction accuracy of the evaluation classifier when fed with reconstructed images. For all the face image datasets, we use the model in [24] as our evaluation classifier, which is pretrained on MS-Celeb-1M [23] and fine-tuned on the training set of the target networks. For MNIST, we

train a nevaluation classifier which consists of 5 convolutional layers and 2 pooling layers on all of the 10 digits. For ChestX-ray8, the evaluation classifier is adapted from [20]. For CIFAR10, we use ResNet-18 adapted from [21].

- K-Nearesr Neighbor Distance (KNN Dist). KNN Dist is the shortest feature distance from a reconstructed image to the real private training images for a given class. The feature distance is measured by the l_2 distance between two images when projected onto the feature space, i.e., the output of the penultimate layer of the evaluation classifier.
- FID. FID score measures feature distances between real and fake images, and lower FID values indicate better image quality and diversity. We found that reconstructed images which the evaluation classifier predicts into the target label tend to achieve lower FID scores. Hence, the FID score and attack accuracy are correlated with one another. To make FID a complementary metric to attack accuracy, we only calculate the FID score of those reconstructions which are successfully recognized as the target class by the evaluation classifier. The idea of this FID score is to measure how much detailed information is leaked from a reconstruction that can successfully recover the semantics.

4.2 Result

Comparison with previous state-of-the-art. Table 1 compares the attack performance of our attack and the baseline on various datasets. We can see that our method outperforms the GMI by a large margin. One interesting finding is that, when attacking digit recognition model trained on MNIST, GMI generates images that can be successfully recognized as the target digits by the target classifier but cannot be predicted into the target digits by the evaluation classifier, which leads to 0 average accuracy. A specific example is that GMI tends to generate “7" when attacking digit "1" because it only sees “7” in the public data. In in case, the generated sample can achieve a very low identity loss under the target network because it is trained to only predict 1-5, while having a high identity loss under the evaluation classifier which can predict all ten digits. Our method can overcome this problem to some extent and has better performance. We also compare our attack with the baseline for attacking various models built on the same dataset, namely, CelebA. The models include VGG16, ResNet152, and face.evolve, which have increased complexity. Among these models, face.evolve achieves state-of-the-art face recognition performance. The results for attacking these models are shown in Table 2, showing that our approach significantly improves the baseline on all the target models. The performance improvement achieved by our attack is further corroborated by Figure 2, which exhibits ground truth private images and corresponding reconstructions given by our attack and the baseline. We can see that our reconstructions can mostly better preserve the facial features of a given identity than the baseline.

	CelebA		**MNIST**		**ChestX-ray8**		**CIFAR10**	
	GMI	**Ours**	**GMI**	**Ours**	**GMI**	**Ours**	**GMI**	**Ours**
Attack Acc	.21±.0020	**.72±.0018**	0	**.56±.0208**	.21±.0163	**.47±.0155**	.56±.0264	**.96±.0072**
KNN Dist	2996.91	**2987.05**	126.61	**72.54**	360.32	**220.30**	139.09	**123.07**
FID	52.51	**23.72**	93.06	**88.39**	295.44	**258.81**	319.27	**233.65**

Table 1: Attack performance comparison on various datasets.

	face.evolve		**IR152**		**VGG16**	
	GMI	**Ours**	**GMI**	**Ours**	**GMI**	**Ours**
Acc	.31±.0039	**.81±.0016**	.32±.0027	**.81±.0015**	.21±.0020	**.72±.0018**
Acc5	.53±.0015	**.96±.0004**	.57±.0005	**.96±.0001**	.43±.0014	**.92±.0003**
KNN Dist	2991.75	**2981.49**	3006.37	**2985.51**	2996.91	**2987.09**
FID	33.81	**25.28**	50.11	**26.35**	52.51	**23.72**

Table 2: Attack performance comparison on various models trained on CelebA.

Cross-dataset experiment. We study the effect of distribution shift between public and private data on the attack performance. We train our GAN on Flickr-Faces-HQ Dataset (FFHQ) [25] and FaceScrub [26] to attack the target network VGG16 trained on CelebA. The attack results are presented in Table 3, which shows that both GMI and our attack suffer from a performance drop while ours still outperforms GMI. We notice that the performance drop on FaceSrub is larger than that on FFHQ. One possible reason is that images in FaceScrub has much lower resolution (64×64), and there are a number of images under poor lighting conditions or only showing partial faces.

Figure 2: Qualitative comparison for attacking a face recognition model. The first row shows the ground truth image for a target identity. The second and third rows demonstrate the reconstructions produced by the GMI attack and our attack, respectively.

	FFHQ→CelebA		FaceScrub→CelebA	
	GMI	**Ours**	GMI	**Ours**
Acc	.15±.0015	**.36±.0015**	.03±.0004	**.13±.0008**
Acc5	.35±.0017	**.61±.0012**	.11±.0011	**.30±.0015**
KNN Dist	3014.45	**2994.32**	3003.90	**2997.52**
FID	69.12	**36.02**	112.83	**60.05**

Table 3: Attack performance comparison where there is large distributional shift between public and private data. $A \rightarrow B$ represents the setting when the target network is trained on dataset B and the GAN is trained on dataset A to distill a generic prior for reconstructions.

Ablation study. We proposed a couple of ideas to improve the GMI attack in [3], including (1) soft-label discrimination (SD), which enables the discriminator to differentiate soft-labels produced by the target network, (2) entropy minimization (EM), which minimizes the prediction entropy of images produced by the generator, and (3) distribution recovery (DR), which explicitly models and estimates the private data distribution. We have shown that the combination of these ideas can lead to significant attack performance improvement over the baseline. Here, we conduct an ablation study to investigate the improvement introduced by each individual idea. Table 4 presents the result of ablation study for attacking VGG16 trained on the CelebA dataset. We observe that both the attack accuracy and image quality get improved when we apply the idea of soft-label discrimination. Adding entropy minimization or distributional recovery can further improve the performance. The combination of the three ideas leads to the largest improvement.

	GMI	**SD**	**SD+EM**	**SD+DR**	**SD+EM+DR**
Acc	.21±.0020	.35±0042	.43±.0035	.62±.0028	.72±.0018
Acc5	.43±.0014	.60±.0013	.68±.0017	.87±.0003	.92±.0003
KNN Dist	2996.91	2992.54	2987.12	2994.79	2987.09
FID	52.51	33.75	31.09	23.82	23.72

Table 4: Ablation study of ideas introduced in this paper, including soft-label discrimination (SD), entropy minimization (EM), and distributional recovery (DR).

Runtime. We test the efficiency of our attack and compare it with GMI. Once the GAN is trained, it takes around 180 seconds for GMI to generate one reconstruction for attacking VGG16. Our attack needs 200 seconds to estimate the parameters of the private data distribution yet only needs 0.3 second to generate one reconstruction once the distribution is learned.

5 Conclusion

In this paper, we propose several techniques that can significantly improve whitebox MI attacks against DNNs. Specifically, we propose to customize the training of a GAN to better distill knowledge useful for performing inversion attacks from public data. Additionally, we propose to build an explicit parameteric model for the private data distribution and present methods to estimate its parameters. Our experiments show that the combination of the proposed techniques

can lead to the state-of-the-art atttack performance on various datasets, models, and even when the public data has a large distributional shift from private data.

For future work, we will investigate the potential application of these techniques to improve the MI attack in the blackbox setting.

References

[1] Fredrikson, Matthew, et al. "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing." 23rd USENIX Security Symposium (USENIX Security 14). 2014.

[2] Fredrikson, Matt, Somesh Jha, and Thomas Ristenpart. "Model inversion attacks that exploit confidence information and basic countermeasures." Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. 2015.

[3] Zhang, Yuheng, et al. "The secret revealer: generative model-inversion attacks against deep neural networks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[4] Shokri, Reza, et al. "Membership inference attacks against machine learning models." 2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017.

[5] Merity, Stephen, et al. "Pointer sentinel mixture models." arXiv preprint arXiv:1609.07843 (2016).

[6] Krishna, Kalpesh, et al. "Thieves on sesame street! model extraction of bert-based apis." arXiv preprint arXiv:1910.12366 (2019).

[7] Orekondy, Tribhuvanesh, Bernt Schiele, and Mario Fritz. "Knockoff nets: Stealing functionality of black-box models." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019.

[8] Correia-Silva, Jacson Rodrigues, et al. "Copycat cnn: Stealing knowledge by persuading confession with random non-labeled data." 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018.

[9] Ateniese, Giuseppe, et al. "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers." International Journal of Security and Networks 10.3 (2015): 137-150.

[10] Ganju, Karan, et al. "Property inference attacks on fully connected neural networks using permutation invariant representations." Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018.

[11] Melis, Luca, et al. "Exploiting unintended feature leakage in collaborative learning." 2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019.

[12] Jaynes, Edwin T. "On the rationale of maximum-entropy methods." Proceedings of the IEEE 70.9 (1982): 939-952.

[13] Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein gan." arXiv preprint arXiv:1701.07875 (2017).

[14] Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

[15] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531 (2015).

[16] Salimans, Tim, et al. "Improved techniques for training gans." Advances in neural information processing systems. 2016.

[17] Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009): 7.

[18] Topol, Eric J. "High-performance medicine: the convergence of human and artificial intelligence." Nature medicine 25.1 (2019): 44-56.

[19] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[20] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[21] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[22] Heusel, Martin, et al. "Gans trained by a two time-scale update rule converge to a local nash equilibrium." Advances in neural information processing systems. 2017.

[23] Guo, Yandong, et al. "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition." European conference on computer vision. Springer, Cham, 2016.

[24] Cheng, Yu, et al. "Know you at one glance: A compact vector representation for low-shot learning." Proceedings of the IEEE International Conference on Computer Vision Workshops. 2017.

[25] Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2019.

[26] Ng, Hong-Wei, and Stefan Winkler. "A data-driven approach to cleaning large face datasets." 2014 IEEE international conference on image processing (ICIP). IEEE, 2014.

[27] Krizhevsky, Alex, Vinod Nair, and Geoffrey Hinton. "Cifar-10 (canadian institute for advanced research)." URL http://www. cs. toronto. edu/kriz/cifar. html 5 (2010).

[28] Yeom, Samuel, et al. "Privacy risk in machine learning: Analyzing the connection to overfitting." 2018 IEEE 31st Computer Security Foundations Symposium (CSF). IEEE, 2018.

[29] Iizuka, Satoshi, Edgar Simo-Serra, and Hiroshi Ishikawa. "Globally and locally consistent image completion." ACM Transactions on Graphics (ToG) 36.4 (2017): 1-14.

[30] Yang, Dingdong, et al. "Diversity-sensitive conditional generative adversarial networks." arXiv preprint arXiv:1901.09024 (2019).

[31] Hidano, Seira, et al. "Model inversion attacks for prediction systems: Without knowledge of non-sensitive attributes." 2017 15th Annual Conference on Privacy, Security and Trust (PST). IEEE, 2017.

[32] Wu, Xi, et al. "A methodology for formalizing model-inversion attacks." 2016 IEEE 29th Computer Security Foundations Symposium (CSF). IEEE, 2016.

[33] Yang, Ziqi, Ee-Chien Chang, and Zhenkai Liang. "Adversarial neural network inversion via auxiliary knowledge alignment." arXiv preprint arXiv:1902.08552 (2019).

[34] Dwork, Cynthia, and Aaron Roth. "The algorithmic foundations of differential privacy." Foundations and Trends in Theoretical Computer Science 9.3-4 (2014): 211-407.

[35] Abadi, Martin, et al. "Deep learning with differential privacy." Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016.

[36] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

[37] Yeh, Raymond A., et al. "Semantic image inpainting with deep generative models." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[38] Hore, Alain, and Djemel Ziou. "Image quality metrics: PSNR vs. SSIM." 2010 20th international conference on pattern recognition. IEEE, 2010.

[39] McMahan, H. Brendan, et al. "A general approach to adding differential privacy to iterative training procedures." arXiv preprint arXiv:1812.06210 (2018).

[40] Carrell, David, et al. "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 (2013): 342-348.

[41] Li, Fenghua, et al. "Hideme: Privacy-preserving photo sharing on social networks." IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEE, 2019.

[42] Wang, Xiaosong, et al. "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[43] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

[44] Liu, Ziwei, et al. "Deep learning face attributes in the wild." Proceedings of the IEEE international conference on computer vision. 2015.

[45] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).

[46] Nguyen, Anh, et al. "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks." Advances in neural information processing systems. 2016.

[47] Yosinski, Jason, et al. "Understanding neural networks through deep visualization." arXiv preprint arXiv:1506.06579 (2015).