Defending Against Model Inversion Attack by Adversarial Examples

Jing Wen†, Siu-Ming Yiu†, Lucas C.K. Hui‡
†Dept. of Computer Science, The University of Hong Kong
‡Hong Kong Applied Science and Technology Research Institute (ASTRI)
Email: {jwen, smyiu}@cs.hku.hk, lucashui@astri.org

***Abstract*—Model inversion (MI) attacks aim to infer and reconstruct the input data from the output of a neural network, which poses a severe threat to the privacy of input data. Inspired by adversarial examples, we propose defending against MI attacks by adding adversarial noise to the output. The critical challenge is finding a noise vector that maximizes the inversion error and introduces negligible utility loss to the target model. We propose an algorithm to craft such noise vectors, which also incorporates utility-loss constraints. Specifically, our algorithm takes advantage of the gradient of an inversion model we train to mimic the adversary and compute a noise vector to turn the output into an adversarial example that can maximize the reconstruction error of the inversion model. Then we apply a label modifier that keeps the label unchanged to achieve zero accuracy loss of the target model. Our defense does not tamper with the training process or need the private training dataset. Thus it can be easily applied to any current neural networks or APIs. We evaluate our method under both standard and adaptive attack settings. Our empirical results show our approach is effective against state-of-the-art MI attacks due to the transferability of adversarial examples and outperforms existing defenses. Furthermore, it causes more reconstruction errors while introducing zero accuracy loss and less distortion than existing defenses.**

I. INTRODUCTION

Deep neural networks (DNNs) have been widely adopted in various applications, including computer vision, speech recognition, and healthcare. Thus the security issues of DNN systems are becoming increasingly critical. On the one hand, researchers discovered DNNs are vulnerable to adversarial examples[1]: an imperceptible noise can be added to an input that alters the prediction. The application of DNNs to domains, on the other hand, involves processing sensitive and proprietary datasets, which raised significant concerns about data privacy. In addition, recent research has shown that DNN models may unintentionally disclose private training data through their outputs or parameters.

In this paper, we are interested in Model Inversion (MI) attacks, which try to reconstruct the input from the model prediction. The first MI attack [2] showed adversaries could infer private genomic information in the training dataset by accessing a linear regression model. Recent studies extended MI attacks to different settings. Yang et al. [3] proposed a black-box MI attack in which the attacker leverages auxiliary information to build an inversion model that can generate the original input sample with high similarity. Zhang et al.[4] propose generative MI attacks that can recover photos of any person from a face recognition model.

Defending against MI attacks is an urgent research problem. However, there are few viable defense methods. The potential reason why MI attacks succeed is that the output of DNN models contains the confidence scores and unexpected redundant knowledge. Our work proposes defending against MI attacks by adding adversarial noise to the output to disrupt the data used to infer the input. Given the output of the target network, our defense seeks the noise vector achieving two goals: 1) maximize the reconstruction error of a potential attacker who wants to use the output to reconstruct the input; 2) introduce bounded distortion to the output without changing the predicted label. Moreover, the distortion can be bounded with a distortion budget provided by the model provider. Achieving the above two objectives can be formulated into an optimization problem. However, solving the optimization problem is computationally challenging due to the large noise space and complex constraints. Inspired by the adversarial examples, we proposed an algorithm to generate the noise vector to approximately solve the optimization problem. Specifically, we first train our inversion model, which mimics MI attacks to minimize the reconstruction error. The noise vector is then computed based on the gradient of the inversion model so that it can maximize the reconstruction error to achieve the defense goal. To achieve zero accuracy loss, we propose to apply a label modifier that retains the original predicted label.

We evaluate our defense method against the state-of-the-art black-box MI attack [3] and compare it to existing defense methods [5], [6] on real-world datasets. Our experimental results show that our strategy can effectively defend against MI attacks with small confidence vector distortion and zero accuracy loss. In particular, our defense can disrupt the reconstruction images, whereas the existing defenses blur the faces for facial datasets. Furthermore, our defense remains successful under an adaptive setting where attackers are aware of our defense and attempt to evade it through adversarial training [7]. In summary, we make the following contribution:

- We present an effective defense against black-box MI attacks by adding a bounded adversarial noise vector to the output, which can prevent attackers from generating accurate input data.
- To the best of our knowledge, no existing defense can achieve zero accuracy loss. Our defense can achieve zero

accuracy loss by applying a label modifier, which is independent of our algorithm and can be used to improve other defenses.
- We evaluate our method under standard and adaptive attack settings. Our empirical results show our approach is practical and outperforms previous defenses as it causes more reconstruction errors while introducing less distortion than existing defenses.

II. Related work and background

A. Deep neural networks

A deep neural network used for classification could be regarded as a general hypothesis function $\boldsymbol{y} = C(\boldsymbol{x})$, which takes an input $\boldsymbol{x}$ from a data distribution $\boldsymbol{X}$ and makes prediction over possible classes. Typically, the output $\boldsymbol{y}$ is a probability distribution vector in a k-dimensional space $\boldsymbol{Y}$, where each dimension represents the possibility that the input falls into each class. In most cases, the softmax function normalizes the aforementioned unbounded confidence score vector, also known as logit, into a vector of real values ranging from 0 to 1 and sum to 1. So they can be interpreted as probabilities. Thus, most neural network can be represented as $\boldsymbol{y} = C(\boldsymbol{x}) = \sigma(\boldsymbol{l})$, where $\boldsymbol{l}$ is logit predicted by the network, and σ is a normalization function. The training process is to minimize the loss between the prediction $\boldsymbol{y}$ and the ground truth classification $\hat{\boldsymbol{y}}$: $\min_f E_{\boldsymbol{x}\sim\boldsymbol{X}}[\mathcal{L}(C(\boldsymbol{x}), \hat{\boldsymbol{y}})]$ (e.g., cross-entropy loss for classification).

B. Model inversion attacks

Model inversion attacks attempt to infer the input data from the corresponding output or other information leaked by the target model. Fredrikson et al. [2] proposed the first algorithm to recover the training data associated with an arbitrary model output given the linear regression model as well as other non-sensitive features of the input. Afterward, they [8] applied MI attacks to more complex models such as decision trees and shallow neural networks in the context of face recognition. The algorithm formulated the MI attack as an optimization problem, intending to find the representative with the highest likelihood or posterior probability for a given class. These optimization-based MI attacks can generate face images with identification accuracy much higher than random guessing. However, the recovered blurry faces do not resemble natural images, especially for complex model architectures.

Unlike developing optimization algorithms to perform inversion tasks, recent MI attacks leverage another neural network to invert the model. Zhang et al.[4] presented a generative model that distills generic knowledge from public datasets and uses it to reconstruct realistic images for deep neural networks. Yang et al.[3] proposed training an additional inversion model that swaps the input and output of the target network. Specifically, their inversion model is as follows: $\bar{\boldsymbol{x}} = A(C(\boldsymbol{x}))$ which takes the logit of a target model as the input and aim to generate the corresponding input. Their experimental results showed significant improvement of the inversion accuracy and recovered image quality over previous works.

C. Existing defense methods

There are very few defenses or countermeasures against current MI attacks. Wang et al. [5] introduced the Mutual Information Regularization based Defense (MID) against privacy attacks. The main idea is to minimize the dependence between inputs and predictions by incorporate their mutual information $\mathcal{I}(\boldsymbol{X}, \boldsymbol{Y})$ as a regularizer into the training objective. Therefore the adversary is less capable of inferring the input data from the model prediction. However, the model needs to be retrained with mutual information loss, which introduces massive amount of overhead for complicated network architectures. Moreover, MID is designed for better protection of privacy rather than preventing MI attacks specifically.

Prediction Purification Framework (PPF) [6] was proposed to purify output with the goal of removing redundant information, which could be used to infer the input by the adversary. Specifically, an autoencoder is trained as a purifier: $\tilde{\boldsymbol{y}} = P(\hat{\boldsymbol{y}})$, which takes the logit of the target model as input and regenerate it. In addition, an inversion model is trained to minimize the reconstruction error as an adversary. The purifier and the adversary are alternatively trained to find the best result, respectively.

D. Adversarial examples

Adversarial attacks [1] transform input into adversarial examples by adding a small amount of deliberately crafted noise that could mislead the DNN models. Various algorithms [9], [10], [11] have been developed to create adversarial examples that deceive both humans and models. Madry et al. [7] further formalized adversarial attacks as a saddle point problem, which can be solved by first-order methods (i.e., the gradient information of the network), thereby motivate projected gradient descent (PGD) as a universal first-order adversarial attack.

III. Problem formulation

A. Threat model

In our scenario, we have three parties: model provider, attacker, and defender. The model provider trains a classification model F on the proprietary training dataset. Then the well-trained model is released as a black box, e.g., as a cloud service, and returns probability distribution vectors $f(\boldsymbol{x})$ to users for their query data $\boldsymbol{x}$. This model is called the target model for convenience.

With access to the target model F, the attacker can query F with any data $\boldsymbol{x}$ to obtain its probability distribution vector $F(\boldsymbol{x})$. Also, the attacker knows some public auxiliary information I such as a similar dataset as the target model used for training. In this paper, we focus on a state-of-the-art black-box MI attack [3] that intends to train another inversion model A on the auxiliary dataset. Then, the inversion model is used to reconstruct the input data from its probability distribution vector.

B. Defender

The defender could be the model provider or a third party who has the same access to the target model and the public auxiliary information as the attacker. The target model predicts the output for any query from users or attackers. Then we add defensive noise to it before releasing it. Formally, we have $\boldsymbol{y'} = \boldsymbol{y} + \boldsymbol{e}$, where $\boldsymbol{e}$ is the noise vector added by the defender that meets the following goals:

1) Defense goal: the noise vector should maximize the reconstruction error $\mathcal{R}(\boldsymbol{x}, A(\boldsymbol{y'_x}))$ thereby prevents attackers from inferring the private input $\boldsymbol{x}$.

2) Utility goal: the modified output should still be a probability distribution vector with the same predicted label. Formally, each value in it should range from 0 to 1 and sum to 1 in all. Since the confidence score vector provides additional information beyond the predicted label to users, the noise itself should introduce minor distortion. It is bounded by a distortion budget ϵ that means the maximum distortion the model provider can tolerate.

C. Mathematic formulation

We formulated the defense against MI attacks as an optimization problem, where $\boldsymbol{y}$ is the output of the target model for input $\boldsymbol{x}$, and the objective is to maximize the reconstruction error of the original input and the reconstructed input as following:

$$\begin{aligned} \max \quad & \mathcal{R}(\boldsymbol{x}, \mathcal{A}(\boldsymbol{y} + \boldsymbol{e})) \\ \textit{subject to}: \quad & \boldsymbol{e} \leq \epsilon \\ & \arg\max(\boldsymbol{y} + \boldsymbol{e}) = \arg\max \boldsymbol{y} \\ & 0 \leq (\boldsymbol{y}_i + \boldsymbol{e}_i) \leq 1,\ \textstyle\sum(\boldsymbol{y}_i + \boldsymbol{e}_i) = 1 \end{aligned} \tag{1}$$

The first constraint is the distortion budget that the model provider can tolerate. The second constraint means the noise added will not change the predicted label of the input. Finally, the vector with noise is still a probability distribution vector, as the last constraint implies.

IV. Our defensive method

Our approach is designed to defend against black-box MI attacks where adversaries exploit the output of the target model to infer the input. Instead of tampering with the training process of the target model (e.g., MID [5]), our defense adds carefully crafted perturbation to the output predicted by the target network with utility-loss guarantees. Thus, our defense could be deployed to an existing network or an API of commercial models without retraining it.

Due to the large noise space, it is computationally challenging to solve this optimization problem directly. Noted that the optimization objective we want to maximize is the same loss function that the inversion model tends to minimize, approximately solving this optimization problem can be viewed as finding an untargeted adversarial example to mislead the inversion model. Specifically, we consider $\boldsymbol{y}$ as the benign example, and $\boldsymbol{y} + \boldsymbol{e}$ is the adversarial example we want to find. However, previous adversarial algorithms are insufficient for our problem because of the unique utility-loss constraints in our defensive setting. Therefore, we present a new algorithm exploiting the concept of adversarial examples to craft the specific noise vector for our problem after eliminating the constraints one by one.

A. Eliminating the constraints

In Equations 1, the last constraint where the vector with noise is still a probability distribution seems complex. It can be eliminated by a change of variables. As the target network is a neural network with a softmax normalization layer, which takes an unbounded confidence score vector $\boldsymbol{c} = F(\boldsymbol{x})$ as input, where F is the part of the target network without the softmax layer, we can add noise to the confidence score vector instead of to the probability distribution vector. Formally, we have $\boldsymbol{y} + \boldsymbol{e} = \sigma(\boldsymbol{c} + \boldsymbol{n})$, where σ is the softmax function, and $\boldsymbol{n}$ is the new variable that we are looking for. For any value of $\boldsymbol{n}$, the noisy output is still a probability distribution. Thereby the constraint is satisfied automatically. Then we obtain our new optimization objective $\max \mathcal{R}(\boldsymbol{x}, \mathcal{A}(\sigma(\boldsymbol{c} + \boldsymbol{n})))$, and the noise vector: $\boldsymbol{e} = \sigma(\boldsymbol{c} + \boldsymbol{n}) - \boldsymbol{y}$. However, there is no need to calculate the noise vector for the probability distribution. To simplify our problem, we directly manipulate the confidence score vector and calculate the noise probability distribution vector based on it.

The second constraint intends to fix the true label of the output after adding the noise vector. Let $l = \arg\max \boldsymbol{y}$ be the lable that the target network predicts. Therefore, we can enforce the the confidence value for entry l to be the maximum. Specifically, we create a label modifier for the entry of predicted label:

$$m_l = ReLU(\max(\boldsymbol{c} + \boldsymbol{n}) - (\boldsymbol{c}_l + \boldsymbol{n}_l)) \tag{2}$$

where $ReLU(x) = \max(0, x)$ is a common-used activation function and $\boldsymbol{c}_l$ and $\boldsymbol{n}_l$ is l_{th} entry of the vector. By adding this modifier, the output can satisfy the second constraint.

B. Generating the noise vector

After eliminating part of the constraints, we can begin to tackle the maximization problem, which can be viewed as finding an adversarial example to mislead the inversion model. Since the reconstruction error we want to maximize is the loss function used to train the inversion model A, we can exploit the gradient of inversion model to calculate the optimal noise vector so that this adversarial gradient signal can maximally deviates from the original gradient. Basically, motivated by simple one step adversarial attack FGSM [1], we develop an algorithm shown in Algorithm 1 to find the noise vector, which can achieve the defense goal as well as satisfy all the constraints. Given an input $\boldsymbol{x}$ and its corresponding confidence score vector $\boldsymbol{c} = F(\boldsymbol{x})$ from the target model, in each step, we calculate the optimal noise vector as follows: $\boldsymbol{n} = \eta \cdot sign[\nabla \mathcal{R}_{c_t}(\boldsymbol{x}, A(\sigma(\boldsymbol{c})))] + m_l$, where η is the step size, and m_{l_t} is the label modifier. If the distortion exceeds the distortion budget, we normalize the distortion under the given norm and multiply it by ϵ to ensure the distortion budget.

Algorithm 1: Calculating the noise vector

Input: original logit $\boldsymbol{c}_0$, current logit $\boldsymbol{c}_t$, corresponding gradient $\boldsymbol{grad}$, step size η, distortion budget ϵ

Output: The adversarial noise vector $\boldsymbol{n}$

Function getNoise$(\boldsymbol{c}_0, \boldsymbol{c}_t, \boldsymbol{grad}, \eta, \epsilon)$ **:**

- $\boldsymbol{c} \leftarrow \boldsymbol{c}_t + \eta * sign(\boldsymbol{grad})$;
- $l \leftarrow \arg\max(\boldsymbol{c}_0)$;
- $m \leftarrow ReLU(\max(\boldsymbol{c}) - \boldsymbol{c}[l])$;
- $\boldsymbol{c}[l] \leftarrow \boldsymbol{c}[l] + m$;
- $\boldsymbol{n} \leftarrow \boldsymbol{c} - \boldsymbol{c}_0$;
- **if** $||\boldsymbol{n}||_p > \epsilon$ **then**
  - $\boldsymbol{n} = \epsilon * \boldsymbol{n}/||\boldsymbol{n}||_p$
- **end**
- **return** $\boldsymbol{n}$

C. Our defense

In our preliminary experiments, we observed that repeatedly perturbing the confidence score vector can improve the defense strength. Therefore, our defense algorithm can be extended into a multi-step version by iteratively finding new noise vectors for current results:

$$\begin{aligned} \boldsymbol{c}_0 &= F(\boldsymbol{x}) \\ \boldsymbol{n}_t &= \eta sign[\nabla \mathcal{R}_{c_t}(\boldsymbol{x}, A(\sigma(\boldsymbol{c}_t)))] + m_{l_t} \\ \boldsymbol{c}_{t+1} &= \boldsymbol{c}_t + \epsilon \cdot \boldsymbol{n}_t/||\boldsymbol{n}_t||_p \end{aligned} \tag{3}$$

Algorithm 2: Our defense algorithm

Input: Input data $\boldsymbol{x}$, Target network F, Inversion model A, iteration$i$, step size η, distortion budget ϵ

Output: The adversarial output $\boldsymbol{y}$

- $\boldsymbol{c}_0 \leftarrow F(\boldsymbol{x})$;
- $\boldsymbol{c} \leftarrow F(\boldsymbol{x})$;
- **for** $i \leftarrow 1$ **to** i **do**
  - $\boldsymbol{recon} \leftarrow A(\sigma(\boldsymbol{c}))$; $\boldsymbol{loss} \leftarrow \mathcal{R}(\boldsymbol{x}, \boldsymbol{recon})$;
  - $\boldsymbol{loss}$.backpropagation();
  - $\boldsymbol{grad} \leftarrow \boldsymbol{c}$.getGradient();
  - $\boldsymbol{c} \leftarrow \boldsymbol{c_0}$ + getNoise$(\boldsymbol{c}_0, \boldsymbol{c}_t, \boldsymbol{grad}, \eta, \epsilon)$;
- **end**
- **return** $\sigma(\boldsymbol{c})$

Algorithm 2 shows our iterative defense based on the noise vector found in algorithm 1. First of all, we keep the original confidence score vector as $\boldsymbol{c}_0$ for controlling the distortion budget. In each iteration, we feed the current input $\boldsymbol{x}$ into the target model and reconstruct the input with the inversion model. Next, we calculate the loss between the current input and the corresponding reconstructed input by the inversion model and obtain the gradient using backpropagation. Then the algorithm 1 returns the optimal noise vector for the current iteration. Finally, by adding the bounded noise vector to the original confidence score vector, we obtain the new result. The process will end when the maximum number of iterations is reached.

V. Evaluation

A. Experimental setup

This section evaluates our defense against MI attacks under different settings and compares it with existing defense methods. We implement our defense, attacks, and existing defenses using PyTorch.

1) Dataset: We use four datasets, which are widely adopted in previous works on model inversion attacks. MNIST [12] contains handwritten digit images in 10 classes. As MNIST is used to train the target network, we use the extended version QMNIST [13] as the public dataset for our defense. Similarly, for the face recognition task, we use FaceScrub530 [14] as the private training set. There are 45,897 downloadable images of 530 individuals. The face was processed according to the official bounding box and resized to 64 × 64. As for public face information, we use CelebA [15] which contains 202,599 images of 10,177 celebrities.

2) Target models: For both tasks, we use the same CNN architecture and training strategy as in [3] to train the target model. The FaceScrub classifier and MNIST classifier achieve 85.7% and 99.6% accuracy on their test set, respectively, which are comparable to the state-of-the-art classification performance.

3) Model inversion attacks: In our experiments, we consider the recent state-of-the-art black-box adversarial model inversion attack [3], where the attacker trains an inversion model to reconstruct the input with high fidelity. We use the same model architecture and attack setting as in [3] to train the inversion model. The inversion model was trained on QMNIST and CelebA, respectively. Noted that the architecture of the testing inversion model is different from the model we used in our defense, our defense remove two hidden layers to see if our defense can transfer from our inversion model to the attacker's inversion model.

4) Existing defenses: We compare the performance of our defense with MID [5] and PPF[6], which are the most effective ones presented in the literature thus far. We implemented MID by retraining the target network with the additional mutual information loss, which was estimated using the information bottleneck [16]. As for PPF, we train an additional autoencoder to purify the confidence score vector according to the paper. Since neither of these two papers provided detailed parameters nor published their implementation, we tried our best to reproduce MID and PPF. For our defense, the step size is 0.1 for MNIST and 0.4 for Facescrub, and the iteration is 10.

B. Evaluation Metrics

The following metrics are used to measure the defense performance, and utility of a defense method.

1) Target model accuracy: we measure the target model accuracy on the testing set before and after the defense method is applied. It reflects whether the defense method will substantially reduce the accuracy of the target model.

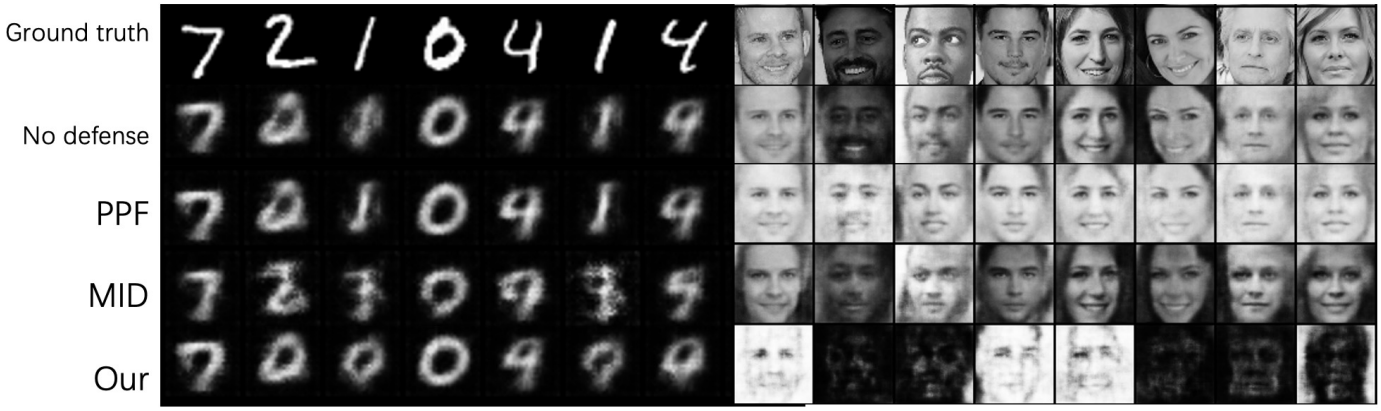

Fig. 1. The reconstruction results by MI attack when different defense methods are deployed.

2) Confidence score distortion: Unlike the accuracy loss, the confidence score distortion may be large while keeping the predicted label unchanged. To measure the confidence score distortion introduced by a defense method, we adopt the commonly used L1-norm of the noise vector, which is the sum of the absolute value of each entry in the vector. Then the loss is averaged for each entry.

3) Reconstruction error: We measure the reconstruction error by computing the mean squared error (MSE) between the original input and the reconstructed input for each pixel. A higher MSE indicates a more effective defense mechanism.

C. Experimental results

Table I demonstrates the comparison of existing defense methods and our defense from the utility for normal users and defense performance against MI attack perspectives. When a normal user queries the target model with defenses, our method achieves zero model accuracy loss for both tasks, while PPF and MID introduce significant utility loss. This is because we add the label modifier that achieves zero model accuracy loss. Technically the label modifier is an independent component of our defense. So PPF and MID can also apply our label modifier to achieve zero model accuracy loss. On the other hand, when an attacker performs MI attacks on the target model, our approach outperforms all the other defense methods with the largest reconstruction error. Especially for facescrub, our defense introduces a minor confidence distortion and significantly increased the reconstruction error by a factor of almost 20 compared to the baseline without defense. This is because our noise vector is elaborately crafted to maximize the reconstruction error. Overall, our approach outperforms existing defense methods against MI attacks in terms of utility and defense performance.

Figure 1 presents the reconstruction results on both datasets with and without defense. For digit images, the performance of MI attacks and defense methods vary significantly for different numbers. For instance, the attacker can generate clear images of 7 and 0 even when defense methods are deployed. However, for numbers 1, 2, and 4, our defense successfully misleads the inversion model, which generates 0 instead. This difference

TABLE I
COMPREHENSIVE RESULTS OF EVALUATED DEFENSE METHODS.

Dataset	Defense	Model Acc.	Conf. Dist.	Recon. Err.
MNIST	None	98.96%	0	0.0096
	PPF	96.67%	**0.4156**	0.0544
	MID	75.83%	1.8206	0.0604
	Ours	**98.96%**	0.4762	**0.0663**
FaceScrub	None	81.52%	0	0.0126
	PPF	69.71%	3.106	0.1256
	MID	55.62%	4.716	0.0446
	Ours	**81.52%**	**2.2352**	**0.2453**

may come from the tiny feature space of digit images. The inversion model learns to generate specific images (e.g., 0 and 7) without using too much information from the confidence vector. As for face image, when no defense is applied, MI attacks can generate a very similar face image with clear facial features. When PPF is applied, the reconstruction images are more like average faces but still keep some prominent facial features that can be used to identify a specific individual. Similarly, with MID deployed, attackers still can generate accurate facial images which only have minimal distortion comparing with the reconstruction results without any defense. This is not a surprise since MID is not explicitly designed for MI attacks. The last row shows the results after applying our defense. The reconstruction of the inversion is successfully disrupted. The results also match the quantified reconstruction error in Table I. The results show that our approach can effectively prevent the model inversion attack with zero utility loss.

D. Adaptive attack with adversarial learning

So far, we have demonstrated the effectiveness of our defense method against MI attacks. We now consider a more realistic scenario where the powerful attackers also know the existence of our defense and seek means to evade our defense. Since we leverage adversarial examples in our defense to mislead the inversion model, adversaries may improve their inversion model to be more robust against adversarial examples in an adaptive setting. Although various defensive

strategies [17], [18], [19] have been explored to defend against adversarial examples, designing such robust classifiers is still considered an open challenge. Nevertheless, we will consider attackers exploit adversarial training to bypass our defense, as adversarial training [7] was considered to be the most empirically robust defense method.

TABLE II
RESULTS OF OUR DEFENSE IN ADAPTIVE SETTING.

	None	Our defense	Adv defense	High dist.
Conf. dist.	0	2.2352	2.738	3.888
Recon. err.	0.0126	0.0271	0.1657	0.0562

Figure 2 and Table II shows the result in an adaptive attack setting. The first row is the ground truth image. The second row demonstrates the reconstructed image of an adversarial training inversion model. Attackers can generate more realistic face images using adversarial training with knowledge of our defense and related parameters. The reconstruction error of our defense is still larger than the baseline. Therefore our defense can prevent attackers from generating accurate face images even when attackers know the mechanism and parameters of our defense.

Fig. 2. The reconstruction results of an adaptive attack with our basic defense (the second row) and improved defense (the last two rows).

Furthermore, our methods can be improved by update our inversion model into an adversarial training version The third row in Figure 2 presents the result of our improved defense, and the average reconstruction error is 0.1657. We could see that the recovered images are more blurry. Moreover, we propose to increase the step size in our primary defense and introduce more distortion to the confidence score vector to counter strong attackers. The last row shows the results of this defense strategy. By increasing the step size to 1, the reconstruction images have low fidelity and noticeable distortion with a slightly higher confidence distortion.

Although in an adaptive attack setting, the performance of our defense drops a little. It is still effective and can be further improved by two different strategies mentioned above.

VI. CONCLUSION

We propose a defense method against black-box MI attacks by turning the output of the target model into an adversarial example that can mislead the attacker. Furthermore, our method is the first defense method that achieves a utility-loss guarantee and zero accuracy loss for the target model. We perform experiments to compare the defense performance and utility-privacy tradeoff on different datasets and models. Our empirical evaluation results show that our defense can achieve extraordinary performance to protect the target model against the state-of-the-art MI attack. We believe it is valuable future work to extend the idea of adversarial examples to defend against other machine learning-based inference attacks such as model stealing attacks [20], and membership inference attacks [21], [22].

REFERENCES

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[2] 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*.

[3] Z. Yang, J. Zhang, E.-C. Chang, and Z. Liang, "Neural network inversion in adversarial setting via background knowledge alignment," in *Proceedings of the 2019 ACM SIGSAC CCS*, 2019.

[4] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in *Proceedings of the 2020 ICCV*.

[5] T. Wang, Y. Zhang, and R. Jia, "Improving robustness to model inversion attacks via mutual information regularization," *arXiv preprint arXiv:2009.05241*, 2020.

[6] Z. Yang, B. Shao, B. Xuan, E.-C. Chang, and F. Zhang, "Defending model inversion and membership inference attacks via prediction purification," *arXiv preprint arXiv:2005.03915*, 2020.

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[8] 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*, 2015, pp. 1322–1333.

[9] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy*.

[10] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*.

[11] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[12] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database."

[13] C. Yadav and L. Bottou, "Cold case: The lost mnist digits," in *Advances in Neural Information Processing Systems 32*, 2019.

[14] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *2014 IEEE International Conference on Image Processing*.

[15] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of ICCV*, 2015.

[16] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," *arXiv preprint arXiv:1612.00410*, 2016.

[17] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*.

[18] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy*. IEEE.

[19] J. Wen, L. C. Hui, S.-M. Yiu, and R. Zhang, "Dcn: Detector-corrector network against evasion attacks on deep neural networks," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*.

[20] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016, pp. 601–618.

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

[22] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 259–274.