IPRemover: A Generative Model Inversion Attack against Deep Neural Network Fingerprinting and Watermarking

Wei Zong[1], Yang-Wai Chow[1], Willy Susilo[1], Joonsang Baek[1], Jongkil Kim[2], Seyit Camtepe[3]
[1]Institute of Cybersecurity and Cryptology (iC[2]), University of Wollongong, Australia
[2]Ewha Womans University, South Korea
[3]CSIRO Data61, Australia
wzong@uow.edu.au, caseyc@uow.edu.au, wsusilo@uow.edu.au, baek@uow.edu.au, Jongkil@ewha.ac.kr, Seyit.Camtepe@data61.csiro.au

Abstract

Training Deep Neural Networks (DNNs) can be expensive when data is difficult to obtain or labeling them requires significant domain expertise. Hence, it is crucial that the Intellectual Property (IP) of DNNs trained on valuable data be protected against IP infringement. DNN fingerprinting and watermarking are two lines of work in DNN IP protection. Recently proposed DNN fingerprinting techniques are able to detect IP infringement while preserving model performance by relying on the key assumption that the decision boundaries of independently trained models are intrinsically different from one another. In contrast, DNN watermarking embeds a watermark in a model and verifies IP infringement if an identical or similar watermark is extracted from a suspect model. The techniques deployed in fingerprinting and watermarking vary significantly because their underlying mechanisms are different. From an adversary's perspective, a successful IP removal attack should defeat both fingerprinting and watermarking. However, to the best of our knowledge, there is no work on such attacks in the literature yet. In this paper, we fill this gap by presenting an IP removal attack that can defeat both fingerprinting and watermarking. We consider the challenging data-free scenario whereby all data is inverted from the victim model. Under this setting, a stolen model only depends on the victim model. Experimental results demonstrate the success of our attack in defeating state-of-the-art DNN fingerprinting and watermarking techniques. This work reveals a novel attack surface that exploits generative model inversion attacks to bypass DNN IP defenses. This threat must be addressed by future defenses for reliable IP protection.

Introduction

Deep Neural Networks (DNNs) have achieved great success in many domains, such as computer vision (Carion et al. 2020; Krizhevsky, Sutskever, and Hinton 2017), Automatic Speech Recognition (ASR) (Hannun et al. 2014; Amodei et al. 2016), Natural Language Process (NLP) (Chowdhary 2020), and so on. Successful training of DNNs requires large volumes of labeled data in the specific domain. Acquiring such data can be costly when they are difficult to collect or significant domain expertise is required for labeling them. Therefore, it is crucial for model owners to be able to protect the Intellectual Property (IP) of DNN models trained using valuable data.

Researchers have recently proposed DNN IP protection techniques that identify the unique characteristics, i.e., the fingerprints, of a model (Lukas, Zhang, and Kerschbaum 2021; Peng et al. 2022; Chen et al. 2022). This direction of work is attracting more and more attention in the research community because fingerprinting techniques do not alter a model's weights, so the performance of the model is preserved. The key to fingerprinting is to find unique properties that only exist in a victim model because if these properties appear in another model, that model is determined to be a stolen model. As an example, Adversarial Examples (AEs) have been used for this purpose. AEs generated by a victim model can be used to uniquely represent a model's decision boundaries (Chen et al. 2022). A stolen model will also be fooled by the same AEs because its decision boundaries will resemble the decision boundaries of the victim model.

However, the robustness of existing fingerprinting techniques has only been evaluated using general attacks, such as fine-tuning all layers of a DNN, adversarial training (Shafahi et al. 2019), and model extraction attacks (Orekondy, Schiele, and Fritz 2019). Although fingerprinting techniques were shown to be robust against these attacks, such attacks were not specifically designed to defeat fingerprint detection. In other words, the robustness of state-of-the-art fingerprinting techniques has not been thoroughly verified.

In addition to DNN fingerprinting, researchers have also explored the embedding of watermarks in DNNs for IP protection (Jia et al. 2021). Ownership can be claimed if a similar watermark can be extracted from a suspect model. For instance, model owners can use a backdoor technique to make their model output a predefined label whenever a special trigger is stamped on its input (Adi et al. 2018).

Although researchers have systematically studied the vulnerabilities of existing watermarking schemes (Lukas et al. 2022), most of the watermark removal attacks, such as transfer learning and adversarial training, have been shown to be ineffective against fingerprinting (Chen et al. 2022). In practice, an effective attack must bypass both DNN fingerprinting and watermarking without prior knowledge of their defenses. This is challenging because DNN watermarking and fingerprinting techniques vary significantly because they are

based on different mechanisms. Such an attack will pose a real-world threat to model owners because the IP of their models cannot be protected reliably.

In this paper, we propose an IP removal attack, called IPRemover, that can evade detection by both state-of-the-art DNN fingerprinting and watermarking techniques. We focus on the challenging data-free scenario where an adversary has no access to any existing data. We assume that a victim model can be accessed in a white-box manner, which is straightforward when an adversary has a local copy of the victim model.

A vital component of our method is a model inversion attack that inverts training data from a victim model. Nonetheless, our goal is to remove IP protection while preserving satisfactory model performance. This goal is different from typical model inversion attacks that aim to compromise privacy by reconstructing representative training data (Nguyen et al. 2023). Moreover, state-of-the-art model inversion attacks assume access to a large volume of data with structural similarity to the original training data (Zhang et al. 2020; Wang et al. 2021; Chen et al. 2021). In contrast, in the context of DNN IP protection removal, an adversary has limited access to useful data. Otherwise, there is no motivation for the adversary to steal the model because the adversary can train a satisfactory model independently via supervised or semi-supervised learning (Zheng et al. 2022).

Our work is related to recent emerging research on data-free Knowledge Distillation (KD) (Yin et al. 2020; Fang et al. 2021; Yu et al. 2023). Data-free KD generates training data from a teacher model, then applies KD to train a student model. The generated data may differ significantly from the original training data because it only needs to be effective for KD. While DNN fingerprinting techniques can detect attacks using KD, to date, there is no method that can detect IP infringement from generated data. Hence, if a stolen model can be trained from scratch on generated data without KD, model owners cannot claim IP infringement on the stolen model because it was trained independently on "legal" data. In comparison with the state-of-the-art in data-free KD, Contrastive Model Inversion (CMI) (Fang et al. 2021), the data generated using our method results in over 10% higher accuracy if models are trained without KD.

Our contributions are summarized as follows:

- To the best of our knowledge, we are the first in the literature to propose a data-free attack, called IPRemover, that can evade detection by both DNN fingerprinting and watermarking techniques.
- We empirically demonstrate that IPRemover can universally defeat a diverse range of recent state-of-the-art DNN fingerprinting and watermarking techniques, namely, MetaFinger (Yang, Wang, and Wang 2022), IP-Guard (Cao, Jia, and Gong 2021), DeepJudge (Chen et al. 2022), Jia (Jia et al. 2021), CosWM (Charette et al. 2022), and Adversarial Frontier Stitching (FS) (Le Merrer, Perez, and Trédan 2020).
- Our work reveals a novel attack surface based on generative model inversion attacks. Future defenses must address this line of attack for reliable DNN IP protection.

Related Work

DNN Fingerprinting

Most existing DNN fingerprinting techniques are based on AEs (Cao, Jia, and Gong 2021; Chen et al. 2022; Peng et al. 2022). The underlying assumption of AE-based fingerprinting is that the decision boundaries of independently trained models differ significantly from one another, such that AEs can be used to uniquely identify their decision boundaries. If the decision boundaries of a suspect model resemble the decision boundaries of a victim model, this indicates that the suspect model is a copy of the victim model.

Other than AEs, researchers recently discovered that meta-learning can be exploited for DNN fingerprinting (Yang, Wang, and Wang 2022). Instead of generating adversarial perturbations to fool DNNs, meta-learning aims to produce noisy input that can only be correctly classified by the model in question.

The robustness of state-of-the-art fingerprinting techniques has not been thoroughly verified because they were only evaluated against attacks that were not specifically designed for defeating fingerprint detection. A recent study by (Wang et al. 2023) proposed to apply preprocessing to mitigate DNN fingerprinting. However, their attack can be easily identified and made ineffective by removing the preprocessing module.

DNN Watermarking

A watermark is a unique signature that can be extracted from a suspect model for verification. A model owner can claim ownership if the watermark extracted from a suspect model resembles the watermark in the owner's model. Jia et al. (Jia et al. 2021) recently proposed to entangle representations of watermarks and benign input to defend against extraction attacks. The assumption is that representations in a stolen model will also contain watermark information, which can be used for IP infringement detection.

Charette et al. (Charette et al. 2022) proposed a watermarking technique, called CosWM, that is robust to model extraction attacks. Their defense was specifically designed for detecting whether an adversary applied KD (Hinton, Vinyals, and Dean 2015) to steal knowledge from a victim model. CosWM empirically showed that an embedded cosine signal can be extracted from the output of a stolen model. Lukas et al. (Lukas et al. 2022) conducted a systematic study on the robustness of existing watermarking techniques using watermark removal attacks. However, most of the studied watermark removal attacks have been shown to be ineffective against fingerprinting (Chen et al. 2022). A practical attack must bypass both DNN fingerprinting and watermarking without prior knowledge of the defense.

The Proposed Method

Threat Model

In this study, we develop a data-free IP removal attack that is able to defeat state-of-the-art DNN IP protection mechanisms. We consider the following threat model:

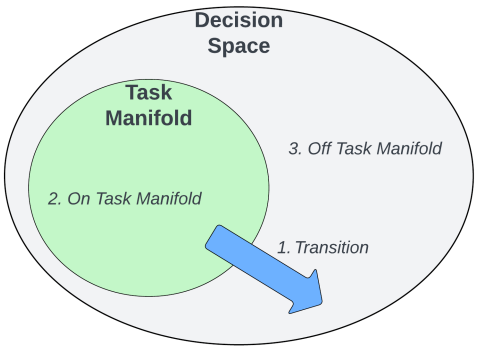


Figure 1: The three directions of watermarking and fingerprinting techniques.

- The adversary's goal is to obtain a stolen model, with decent performance, from a victim model. The adversary has white-box access to the victim model, which is straightforward if the adversary has a local copy of the model.
- The adversary does not require additional data because training data can be inverted from the victim model. This makes the stolen model depend solely on the victim model.
- The adversary has no knowledge of the DNN fingerprinting or watermarking techniques used to protect the victim model. As such, the stolen model needs to universally defeat DNN watermarking and fingerprinting techniques so that IP infringement cannot be proven.

A Unifying View of DNN Fingerprinting and Watermarking

Before detailing IPRemover, we present a unifying view of fingerprinting and watermarking techniques. This is important for proposing an attack that can defeat both watermarking and fingerprinting. Otherwise, different attack strategies will likely be required to defeat different defenses.

While the underlying techniques behind watermarking and fingerprinting vary significantly on the surface, they all exploit the unique characteristics of a model's decision space. Specifically, a model's decision space can be divided into space that is on the task manifold and space that is off the task manifold. As depicted in Fig. 1, existing fingerprinting and watermarking techniques are based on one of three directions.

The first direction focuses on unique transitions from space on the task manifold to space off the task manifold. These unique transitions only exist in the model under protection and cannot be found in other independently trained models. Most existing DNN IP protection techniques are based on this. For instance, AEs can be seen as data points that are off the task manifold (Gilmer et al. 2018; Khoury and Hadfield-Menell 2018). This means that AE based fingerprinting techniques use adversarial perturbations as unique transitions that move data points off the task manifold (Chen et al. 2022; Cao, Jia, and Gong 2021). The assumption is that a stolen model will inherit the unique transitions from the victim model. Hence, when these unique transitions are applied to specific data points, a stolen model will behave the same way as the victim model.

The second direction focuses on unique characteristics of space on the task manifold. As an example, LeMerrer et al. (Le Merrer, Perez, and Trédan 2020) proposed adversarial frontier stitching, which basically extends the task manifold by fine-tuning a model on AEs. In other work, DAWN (Szyller et al. 2021) deliberately introduces incorrect predictions for specific input so that given the same input a stolen model will also make the same errors. Such incorrectly classified input resides in space on the task manifold, where they uniquely characterize the behavior of the model under protection.

The third direction focuses on the unique characteristics of space off the task manifold. A recently proposed method known as MetaFinger (Yang, Wang, and Wang 2022), utilized meta-learning to generate fingerprints that are off the task manifold.

Fingerprinting and watermarking both exploit the unique characteristics of a model's decision space based on one of three directions. Fingerprinting finds unique characteristics that already exist, whereas watermarking actively introduces unique characteristics by modifying the model. Hence, to universally defeat both watermarking and fingerprinting, a successful attack must significantly change the model's decision space while maintaining satisfactory model performance.

IPRemover

Overview To universally evade IP infringement detection, an adversary must significantly change the decision space. However, as fingerprints and watermarks are secret information, an adversary will not know which part of the decision space to modify. Therefore, while a straightforward approach is to alter the entire decision space, doing this will significantly deteriorate the stolen model's performance.

We adopt an approach that looks at the problem from a different perspective. An overview of our method is depicted in Fig. 2. IPRemover consists of 3 stages. In the first, training data is inverted from a victim model. In the second, a stolen model is trained from scratch on the generated data. Finally, a specially designed variant of KD, called Virtual Ensemble Knowledge Distillation (VEKD), is applied to distill knowledge from the victim model while evading IP infringement detection. Algorithm 1 provides the workflow of our attack. Details of generating training data and VEKD are presented in the following subsections.

Model Inversion Our model inversion technique trains a generative model from scratch when recovering a batch of training data from a victim model. This strategy was inspired by the state-of-the-art data-free KD, CMI, proposed by Fang et al. (Fang et al. 2021).

The architecture of the generative model simply stacks convolutional layers and upsampling layers. The method for

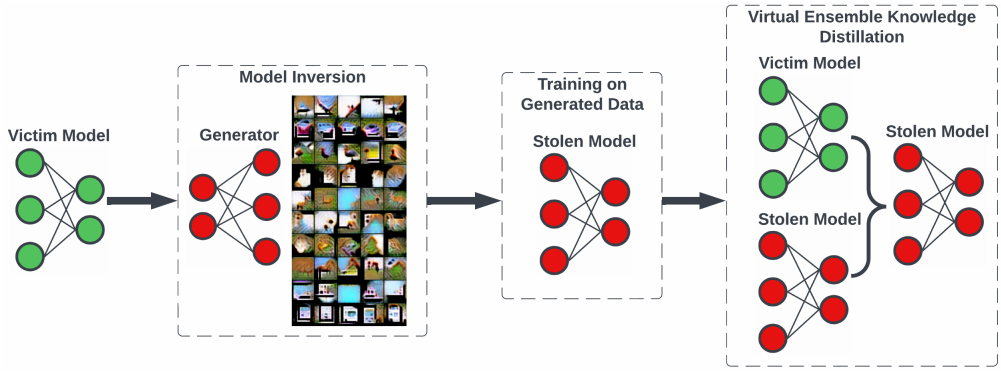


Figure 2: An overview of the IPRemover stages.

Algorithm 1 IPRemover.

Input: a victim model v; the size of generated data C
Output: a stolen model s

generated set: $\mathcal{D} \leftarrow \emptyset$
while $|\mathcal{D}| < C$ **do**
  $z \leftarrow \mathcal{N}(0, 1)$
  initialize θ_g
  randomly sample labels y
  solve Equation 1
  $\mathcal{D} \leftarrow \mathcal{D} \cup$ generated data
end while

create and initialize s
train s on $\mathcal{D}$
apply VEKD to transfer knowledge from v to s using $\mathcal{D}$
return s

training the generative model is as follows:

$$\min_{\theta_g} \mathcal{H}(v \circ g(z; \theta_g), y) + \sum_{l}^{L_v} \alpha_l \ell_{bn}^l (v \circ g(z; \theta_g)) \quad (1)$$

where θ_g represents parameters of the generative model to optimize. v represents the victim model. $\mathcal{H}$ denotes the cross-entropy loss. z is sampled from standard distribution $\mathcal{N}(0, 1)$ and y are randomly selected labels. It should be noted that z is not optimized in our method while CMI optimizes z and θ_g together. We observed that optimizing z makes the generated data for the same label look almost the same, which destroys diversity in the generated data and is thus detrimental to our purpose. L_v denotes the number of batch normalization layers in the victim model. ℓ_{bn} is the widely used batch normalization regularization originally proposed in (Yin et al. 2020). ℓ_{bn} aims to constrain the features of each batch normalization layer to be consistent with the running-mean and running-variance values so that generated data will visually resemble the original training data. ℓ_{bn}^l denotes the loss calculated for the l^{th} layer with α_l for weighting. Each α_l is independently initialized from a uniform distribution, which differs from common practice that uses a constant for weighting all ℓ_{bn}^l. The benefit of random sampling is that it improves diversity in the generated data because different batch normalization layers correspond to different weights.

Random initialization of θ_g and α_l will result in diversity in the generated data. A small batch size is used so that a large number of generative models will be trained from scratch. After completing the training, only correctly classified data is kept. In practice, to be efficient, multiple batches of data can be generated in parallel.

Virtual Ensemble Knowledge Distillation (VEKD) After a stolen model is trained on the generated data from scratch, VEKD is applied to transfer knowledge from the victim model to the stolen model while evading IP infringement detection. Unlike naive KD, which only uses the victim model as a single teacher, VEKD includes the stolen model itself as an additional teacher to create an ensemble. When the stolen model serves as a teacher, we changed its output probabilities into a form where the predicted label corresponds to a high probability while the remaining probability mass is evenly distributed among the other labels. This approach is similar to the teacher-free KD proposed by Yuan et al. (Yuan et al. 2020). The purpose of including the stolen model itself as an additional "virtual" teacher is to reduce the resemblance between the stolen model and the victim model during knowledge transfer, which is beneficial for bypassing IP defenses.

The loss function for VEKD is defined as follows:

$$\ell = \mathcal{H}(p, y) + \beta K D_\tau(p, \epsilon v + (1 - \epsilon) q) \quad (2)$$

where p and y are probabilities output by the stolen model and the label of generated data, respectively. v denotes probabilities output by the victim model. KD_τ is the KD loss defined in (Hinton, Vinyals, and Dean 2015) with temperature τ. q is the output probability when the stolen model serves as an additional teacher:

$$q_i = \begin{cases} Q, & \text{if } i = \arg\max(p) \\ (1 - Q)/(K - 1), & \text{otherwise} \end{cases} \quad (3)$$

where Q is a predefined value representing high probability and K is the number of classes. Finally, β balances different loss values and ϵ balances output probabilities. If $\epsilon = 1$,

Category	Method
Transition	Jia (Jia et al. 2021) DeepJudge (Chen et al. 2022) IPGuard (Cao, Jia, and Gong 2021)
On Manifold	CosWM (Charette et al. 2022) FS (Le Merrer, Perez, and Trédan 2020)
Off Manifold	MetaFinger (Yang, Wang, and Wang 2022)

Table 1: The different DNN watermarking and fingerprinting techniques used to evaluate IPRemover.

VEKD is the same as the naive KD. Whereas if $\epsilon = 0$, VEKD is the same as the teacher-free KD.

Experimental Results

Setup

All experiments were conducted on an Ubuntu 22.04 server with 64G RAM and a Nvidia A100 GPU. We report average values and the standard deviation of results in the form $a \pm b$, where appropriate[1].

We evaluated our method against the different state-of-the-art IP protection techniques shown in Table 1. These techniques cover all three directions previously discussed. Specifically, for watermarking, we considered Jia (Jia et al. 2021), FS (Le Merrer, Perez, and Trédan 2020) and CosWM (Charette et al. 2022). We focused on watermarking techniques that aimed to be robust against model extraction attacks because IPRemover is essentially a model extraction attack. Lukas et al. (Lukas et al. 2022) demonstrated that Jia (Jia et al. 2021) and FS (Le Merrer, Perez, and Trédan 2020) were robust against model extraction attacks. On the other hand, watermarking techniques that are not robust against extraction attacks, such as DeepMarks (Lukas et al. 2022), were not considered as these techniques are already susceptible to such attacks. In addition, the recently proposed CosWM (Charette et al. 2022) was specifically designed for combating ensemble KD. It should be noted that we did not evaluate our method against DAWN (Szyller et al. 2021), even though it is supposed to be robust against model extraction. The reason is that DAWN is an additional component added to a victim model. So under a white-box assumption, an adversary can easily remove such a component.

For fingerprinting, we evaluated our method against IPGuard (Cao, Jia, and Gong 2021), DeepJudge (Chen et al. 2022) and MetaFinger (Yang, Wang, and Wang 2022). These are recently proposed techniques that present practical solutions and demonstrated state-of-the-art results. The details of each defense method are discussed in Technical Appendix.

The datasets used were those widely adopted in deep learning security research, i.e., CIFAR10 (Krizhevsky, Hinton et al. 2009) and the German Traffic Sign Recognition Benchmark (GTSRB) (Stallkamp et al. 2011). CIFAR10 consists of 32×32 color images in 10 classes. There are 50,000 training images and 10,000 test images. We trained wide ResNet (Zagoruyko and Komodakis 2016) on CIFAR10. GTSRB consists of 26,640 images for training and 12,630 images for testing in 43 classes. We resize images in GTSRB to 48×48 and trained VGG11 (Sengupta et al. 2019) on it. More details of the datasets and trained models are provided in Technical Appendix.

For each dataset, we independently trained 3 models from scratch. When calculating detection thresholds for fingerprinting, we considered one model as a victim model while the other independently trained models were treated as benign models. For each victim model, we ran IPRemover 3 times. To ensure fairness in our evaluation, whenever the authors of a particular defense published their pre-trained models, we used the pre-trained models rather than training our own.

Scenario	Accuracy (%)	Query (%)	Detected*
GTSRB	85.32 ± 0.11	85.33 ± 0.94	3/3
CIFAR100	89.37 ± 0.13	98.67 ± 1.25	3/3
Data-free	82.98 ± 0.18	44.33 ± 3.77	0/3
1% data	84.59 ± 0.29	39.67 ± 2.49	0/3
5% data	87.40 ± 0.19	33.00 ± 4.32	0/3

*: IP infringement is detected if the accuracy on the query set exceeds a threshold of 62%. The victim model achieved 90.92% accuracy on the test set and 100% accuracy on the query set.

Table 2: Experimental results for the MetaFinger case study.

A Case Study on MetaFinger

We present a detailed case study to evaluate our method against a recently proposed DNN fingerprinting technique called MetaFinger (Yang, Wang, and Wang 2022). MetaFinger is effective in detecting IP infringement based on KD because its fingerprints are generated based on the Kullback–Leibler (KL) divergence, which is also the key component in KD. The purpose of this case study is to demonstrate the effectiveness of our method through comparative experiments. As the authors of MetaFinger published their trained models[2], our experiments used their wide ResNet pre-trained on CIFAR10. Using the open-source code, we generated a query set of 100 samples in which the victim model achieved 100% accuracy on this query set.

We considered several model stealing scenarios described below:

1. *Scenario "GTSRB":* KD is applied using GTSRB as out-of-distribution (OOD) data to transfer knowledge from the victim model to a stolen model.
2. *Scenario "CIFAR100":* KD is applied using CIFAR100 (Krizhevsky, Hinton et al. 2009) as OOD data to transfer knowledge from the victim model to a stolen model.
3. *Scenario "Data-free":* Using VEKD with only generated data.
4. *Scenario "1% data":* Using VEKD with a mixture of generated data and 1% labeled training data.
5. *Scenario "5% data":* Using VEKD with a mixture of generated data and 5% labeled training data.

[1]Our code is on https://github.com/WeiZong01/IPRemover

[2]https://github.com/kangyangWHU/MetaFinger/

Defense	Victim Acc (%)	Stolen Acc (%)	Threshold†	Detect+	Stolen Metric
IPGuard	93.25	83.71 ± 0.10	0.0	↑	0.0 ± 0.0
DeepJudge*	93.25	83.71 ± 0.10	RobD:0.326 JSD:0.217	↓	0.426 +- 0.006 0.282 +- 0.005
Jia	92.16	83.65 ± 0.26	10% (99%)	↑	$5.22 \pm 1.10\%$
FS	93.30	85.24 ± 0.23	87% (100%)	↑	$84.00 \pm 0.82\%$
CosWM	82.40	75.23 ± 0.34	8.0 (39.83)	↑	1.85 ± 1.36

*: DeepJudge proposed two metrics: "RobD" and "JSD".
+: ↑ (↓) means IP infringement is detected if the measured metric is higher (lower) compared to the metric of benign models.
†: for fingerprinting, the detection threshold was set to the worst value calculated based on benign models; for watermarking, the watermark accuracy of the victim model or a provided detectable stolen model is shown in parentheses.

Table 3: Experimental results on CIFAR10.

Defense	Victim Acc (%)	Stolen Acc (%)	Threshold†	Detect+	Stolen Metric
IPGuard	97.24	91.66 ± 0.29	0.03	↑	0.060 +- 0.008
DeepJudge*	97.24	91.66 ± 0.29	RobD:0.133 JSD:0.079	↓	0.271 +- 0.077 0.156 +- 0.048
Jia	96.67	90.77 ± 0.21	2.33% (81.33%)	↑	$0.89 \pm 0.83\%$
FS	96.95	90.48 ± 0.44	77% (96%)	↑	$68.33 \pm 0.47\%$

*: DeepJudge proposed two metrics: "RobD" and "JSD".
+: ↑ (↓) means IP infringement is detected if the measured metric is higher (lower) compared to the metric of benign models.
†: for fingerprinting, the detection threshold was set to the worst value calculated based on benign models; for watermarking, the watermark accuracy of the victim model or a provided detectable stolen model is shown in parentheses.

Table 4: Experimental results on GTSRB.

It should be noted that while we focused on the data-free attack, the purpose of including Scenarios 4 and 5, which used a small amount of training data, in the case study was to demonstrate the potential benefits of an adversary having access to extra data. For each class, we generated 5000 images to make the size of our generated data match the size of the original training data. In Scenarios 4 and 5 with "1% data" and "5% data", respectively, labeled training data was upsampled by 100 times and 20 times in order for their size to be equal to the size of generated data.

As MetaFinger did not define a method for calculating detection thresholds, we used the maximum accuracy on the query set achieved by fine-tuning a ResNet as the detection threshold. The ResNet was pre-trained on ImageNet and fine-tuning it on CIFAR10 did not infringe on the IP of the victim model. We ran the experiments 3 times and achieved a threshold of 62%.

The experimental results are shown in Table 2. As expected, IP infringement can be easily detected when using KD, even when data for a different task, i.e., GTSRB, was used. In contrast, our IPRemover managed to evade detection. For the data-free case, the stolen model achieved an 82.98% accuracy, which was 7.94% lower than the 90.92% accuracy achieved by the victim model. If our generated data was mixed with 5% labeled training data, the accuracy of the stolen model increased to 87.40%, which was only 3.52% lower than the accuracy of the victim model.

An interesting observation was that if more labeled training data were accessible, the accuracy on the query set decreased. Specifically, the accuracy on the query set was 44.33% for the "Data-free" scenario and this value decreased to 33.00% when 5% labeled training data were included. This implies that stolen models would less resemble the victim model when more labeled training data were accessible.

Defeating Other Defenses

The data-free IPRemover was evaluated against other defenses. For CIFAR10, we generated 5000 images for each class. For GTSRB, we generated at most 600 images for each class. We observed that the success rate for generating some classes of GTSRB was low. This may be due to the uneven distribution of the original training set. We stopped generating images for GTSRB when most classes consisted of 600 images and the success rate of generating the remaining images was low. Details can be found in Technical Appendix.

The experimental results on CIFAR10 and GTSRB are shown in Tables 3 and 4, respectively. We used the same set of hyperparameters and generator architecture for all the defenses on both datasets. This means the accuracy of our stolen models in the experiments represents the lower bound. If an adversary has knowledge about the defense, the hyperparameters can be adaptively adjusted to improve the accuracy of the stolen models. In addition, if additional training data were used, the performance gap between stolen models and victim models would decrease further.

There was only one exception for IPGuard on GTSRB where the metrics of our stolen models slightly exceeded the worst metric of benign models. Nonetheless, the met-

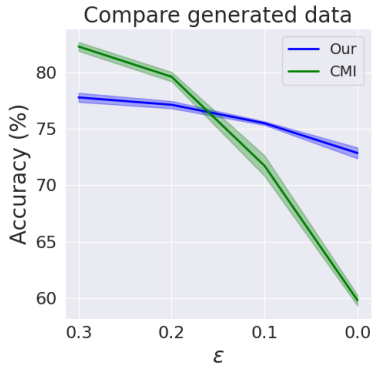


Figure 3: Mean accuracy and standard deviation obtained by running VEKD with different ϵ.

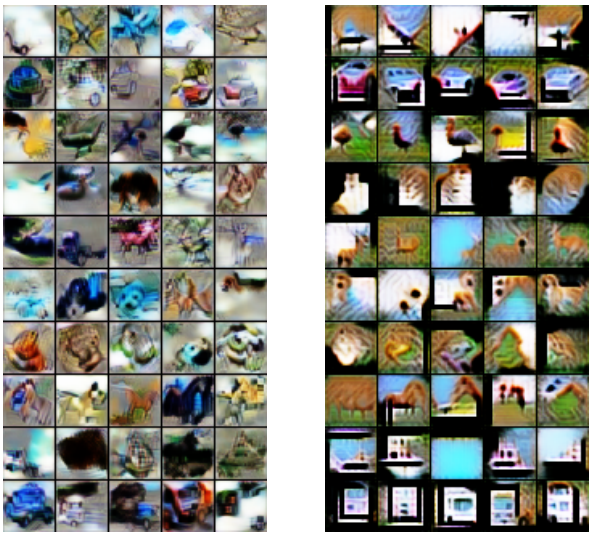
rics of our stolen models were still close to 0, which will force model owners to use a low threshold for IPGuard, e.g., 0.04. Such a low threshold will render IPGuard to be unreliable in practice. Moreover, we only used 2 benign models in the experiments. The threshold for IPGuard is expected to increase if more benign models are involved. Hence, we conclude that our IPRemover bypassed all the defenses on CIFAR10 and GTSRB.

For DeepJudge, an interesting observation was that untargeted AEs were highly transferable to other independently trained models. For example, running two iterations of PGD with a perturbation bound of 0.1 and 0.001 made the RobD and JSD of independently trained models less than 0.1 on CIFAR10. These low values made DeepJudge unreliable since the thresholds would be close to 0. Empirically, this high transferability of untargeted AEs is strongly related to the number of iterations of PGD. It can be seen in Technical Appendix that using different values for the perturbation bounds slightly affects the overall metrics. In contrast, running 3 or more iterations significantly lowers the metrics making them close to 0. A potential reason for this is that we applied standard normalization to the input. This differs from the open-source implementation of DeepJudge. However, standard normalization is common practice for many machine learning tasks to make input features on a similar scale as it stabilizes the training process and improves the performance of trained models. Our experimental results imply that standard normalization may also facilitate the transferability of untargeted AEs between independently trained models.

Comparison with CMI

To date, there is no method that can detect IP infringement in generated data. Hence, it is advantageous for a stolen model to be trained from scratch without KD. In this case, model owners cannot claim IP on the stolen model because it was trained independently on "legal" data.

We compared our method with CMI by varying the dependence on KD. The authors of CMI published their inverted data for a "wrn-40-2" model trained on CIFAR10 [3]. Hence, we also applied our method to this model to generate training data. We used the same set of hyperparameters and generator architecture for generating data as in the previous experiments on CIFAR10, although this victim model was much smaller.



(a) CMI (b) Our method

Figure 4: Visual comparison of the generated images.

Fig. 3 shows the results of running VEKD 3 times with different ϵ. Recall that a smaller ϵ corresponds to less dependence on KD. The results show that for both CMI and our method, the accuracy of the models decreased with less dependence on KD. However, models trained on our generated data achieved higher accuracy when $\epsilon \leq 0.1$. When no KD was applied, $\epsilon = 0$, models trained on our generated data achieved 10% higher accuracy. Fig. 4 shows randomly selected images for a visual comparison. Compared to CMI, the colors of our generated data are visually clearer and more natural.

Conclusion and Future Work

In this work, we proposed a generative model inversion attack that can defeat both DNN fingerprinting and watermarking techniques. We considered the challenging data-free scenario where data is inverted from a victim model. After a stolen model is trained on generated data, VEKD is applied to transfer knowledge from the victim model to the stolen model while evading IP infringement detection. Our work reveals a novel attack surface that exploits model inversion attacks to bypass DNN IP protection.

In future work, we will explore methods of detecting IP infringement from generated data, which is an untouched research direction. Another interesting direction is to extend our work to defeat IP protection in areas other than image recognition.

[3] https://github.com/zju-vipa/CMI

Acknowledgments

This work is partially supported by the Data61 CRP project. W. Susilo is supported by the Australian Research Council Australian Laureate Fellowship FL230100033. J. Kim is partially supported by IITP grant funded by the Korea government (MSIT) (No.RS-2022-00155966, Artificial Intelligence Convergence Innovation Human Resources Development (Ewha Womans University)).

References

Adi, Y.; Baum, C.; Cisse, M.; Pinkas, B.; and Keshet, J. 2018. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX Security Symposium (USENIX Security 18)*, 1615–1631.

Amodei, D.; Ananthanarayanan, S.; Anubhai, R.; Bai, J.; Battenberg, E.; Case, C.; Casper, J.; Catanzaro, B.; Cheng, Q.; Chen, G.; et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, 173–182. PMLR.

Cao, X.; Jia, J.; and Gong, N. Z. 2021. IPGuard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, 14–25.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-End Object Detection with Transformers. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J., eds., *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, 213–229. Springer.

Charette, L.; Chu, L.; Chen, Y.; Pei, J.; Wang, L.; and Zhang, Y. 2022. Cosine Model Watermarking against Ensemble Distillation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, 9512–9520. AAAI Press.

Chen, J.; Wang, J.; Peng, T.; Sun, Y.; Cheng, P.; Ji, S.; Ma, X.; Li, B.; and Song, D. 2022. Copy, right? A testing framework for copyright protection of deep learning models. In *2022 IEEE Symposium on Security and Privacy (SP)*, 824–841. IEEE.

Chen, S.; Kahla, M.; Jia, R.; and Qi, G.-J. 2021. Knowledge-enriched distributional model inversion attacks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 16178–16187.

Chowdhary, K. 2020. Natural language processing. *Fundamentals of artificial intelligence*, 603–649.

Fang, G.; Song, J.; Wang, X.; Shen, C.; Wang, X.; and Song, M. 2021. Contrastive Model Inversion for Data-Free Knowledge Distillation. *CoRR*, abs/2105.08584.

Gilmer, J.; Metz, L.; Faghri, F.; Schoenholz, S. S.; Raghu, M.; Wattenberg, M.; and Goodfellow, I. 2018. Adversarial spheres. *arXiv preprint arXiv:1801.02774*.

Hannun, A.; Case, C.; Casper, J.; Catanzaro, B.; Diamos, G.; Elsen, E.; Prenger, R.; Satheesh, S.; Sengupta, S.; Coates, A.; et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Jia, H.; Choquette-Choo, C. A.; Chandrasekaran, V.; and Papernot, N. 2021. Entangled watermarks as a defense against model extraction. In *30th USENIX Security Symposium (USENIX Security 21)*, 1937–1954.

Khoury, M.; and Hadfield-Menell, D. 2018. On the geometry of adversarial examples. *arXiv preprint arXiv:1811.00525*.

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

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90.

Le Merrer, E.; Perez, P.; and Trédan, G. 2020. Adversarial frontier stitching for remote neural network watermarking. *Neural Computing and Applications*, 32(13): 9233–9244.

Lukas, N.; Jiang, E.; Li, X.; and Kerschbaum, F. 2022. Sok: How robust is image classification deep neural network watermarking? In *2022 IEEE Symposium on Security and Privacy (SP)*, 787–804. IEEE.

Lukas, N.; Zhang, Y.; and Kerschbaum, F. 2021. Deep Neural Network Fingerprinting by Conferrable Adversarial Examples. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Nguyen, N.-B.; Chandrasegaran, K.; Abdollahzadeh, M.; and Cheung, N.-M. 2023. Re-thinking Model Inversion Attacks Against Deep Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16384–16393.

Orekondy, T.; Schiele, B.; and Fritz, M. 2019. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4954–4963.

Peng, Z.; Li, S.; Chen, G.; Zhang, C.; Zhu, H.; and Xue, M. 2022. Fingerprinting Deep Neural Networks Globally via Universal Adversarial Perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13430–13439.

Sengupta, A.; Ye, Y.; Wang, R.; Liu, C.; and Roy, K. 2019. Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in neuroscience*, 13: 95.

Shafahi, A.; Najibi, M.; Ghiasi, M. A.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L. S.; Taylor, G.; and Goldstein, T. 2019. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32.

Stallkamp, J.; Schlipsing, M.; Salmen, J.; and Igel, C. 2011. The German traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, 1453–1460. IEEE.

Szyller, S.; Atli, B. G.; Marchal, S.; and Asokan, N. 2021. DAWN: Dynamic Adversarial Watermarking of Neural Networks. In Shen, H. T.; Zhuang, Y.; Smith, J. R.; Yang, Y.; César, P.; Metze, F.; and Prabhakaran, B., eds., *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021*, 4417–4425. ACM.

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

Wang, M.; Qiu, H.; Zhang, T.; Qiu, M.; and Thuraisingham, B. 2023. Mitigating Query-based Neural Network Fingerprinting via Data Augmentation. *ACM Transactions on Sensor Networks*.

Yang, K.; Wang, R.; and Wang, L. 2022. MetaFinger: Fingerprinting the Deep Neural Networks with Meta-training. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, 776–782. ijcai.org.

Yin, H.; Molchanov, P.; Alvarez, J. M.; Li, Z.; Mallya, A.; Hoiem, D.; Jha, N. K.; and Kautz, J. 2020. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8715–8724.

Yu, S.; Chen, J.; Han, H.; and Jiang, S. 2023. Data-Free Knowledge Distillation via Feature Exchange and Activation Region Constraint. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24266–24275.

Yuan, L.; Tay, F. E.; Li, G.; Wang, T.; and Feng, J. 2020. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3903–3911.

Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. In Wilson, R. C.; Hancock, E. R.; and Smith, W. A. P., eds., *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*. BMVA Press.

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

Zheng, M.; You, S.; Huang, L.; Wang, F.; Qian, C.; and Xu, C. 2022. Simmatch: Semi-supervised learning with similarity matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14471–14481.