Ownership Infringement Detection for Generative Adversarial Networks against Model Stealing

Hailong Hu, Jun Pang

Abstract—**Generative adversarial networks (GANs) have shown remarkable success in image synthesis, making GAN models themselves commercially valuable to legitimate model owners. Therefore, it is critical to technically protect the intellectual property of GANs. Prior works need to tamper with the training set or training process to verify the ownership of a GAN. In this paper, we show that these methods are not robust to emerging model extraction attacks. Then, we propose a new method GAN-Guards which utilizes the common characteristics of a target model and its stolen models for ownership infringement detection. Our method can be directly applicable to all well-trained GANs as it does not require retraining target models. Extensive experimental results show that our new method achieves superior detection performance, compared to watermark-based and fingerprint-based methods. Finally, we demonstrate the effectiveness of our method with respect to the number of generations of model extraction attacks, the number of generated samples, and adaptive attacks.**

Impact Statement—**Generative models are crucial for data synthesis in the era of generative artificial intelligence (AI), with many technology companies leveraging these models to secure a competitive advantage in the global market. However, the deployment of these valuable models exposes them to the risk of illegal theft by adversaries, posing a significant threat to the intellectual property of model owners. To mitigate this risk, our paper introduces a novel ownership infringement detection method that enables model owners to assert their rights over generative adversarial networks (GANs) effectively. Unlike previous methods, which often struggle to verify ownership when models are stolen through advanced model extraction attacks, our method provides robust detection without requiring retraining of the target models. This innovation not only enhances the security of AI deployments but also promotes responsible generative AI use by safeguarding the rights of model owners.**

Index Terms—**Ownership Detection, Generative Adversarial Networks, Watermarks, Fingerprints, Model Confidentiality.**

I. Introduction

Generative artificial intelligence (AI) has already exerted revolutionary influences, ranging from natural language processing [1] to computer vision [2]. Generative adversarial networks (GANs), as one of the most successful generative AI, have also applied to many domains, such as image synthesis [3, 4] and image manipulation [5, 6]. However, building a well-trained state-of-the-art GAN model is not straightforward. It usually requires the complicated and exhausting process of data collection, expert-level knowledge in model architecture design, elaborate hyperparameter tuning, and extensive computing resources. Thus, a high-quality GAN model is very costly and should be regarded as the intellectual property of the model owner.

As GAN models are valuable, this simultaneously incentivizes adversaries to steal these models in various ways. On the one hand, adversaries can physically steal a GAN model via malware infection or insider attacks [7]. An insider attack can directly copy the target model (i.e. victim model) through those who are authorized to access the full model. As a result, the stolen model is totally the same as the target model. On the other hand, adversaries can functionally steal a GAN model via model extraction attacks [8]. This threat exists because an increasing number of technology companies provide Machine Learning as a Service (MLaaS) to their customers, such as Amazon AWS, Google Cloud, and OpenAI API. A model extraction attack enables adversaries to obtain a substitute model via exposed interfaces. As a consequence, the stolen model is functionally similar to the target model. This will lead to financial losses for technology companies, as adversaries who steal their models may either capture their market share or stop using their paid APIs. More importantly, both physical stealing attacks and model extraction attacks seriously jeopardize the intellectual property of legitimate model owners. Therefore, it is paramount to develop protection methods to safeguard the intellectual property of GANs.

Despite confronting these threats, research on ownership protection for GANs is somehow much less explored. Prior works [9, 10, 11] propose to verify ownership of GANs by watermarks. On the one hand, these methods need to retrain or fine-tune target models to embed watermarks, which may compromise the models' generation performance. On the other hand, they rely on specific inputs (i.e. triggers) to extract watermarks [9]. Such dependency might result in their failure to verify models from model extraction attacks. In general, the key challenge is to design methods for GANs that can detect ownership infringement under physical stealing and model extraction attacks while preserving generative performance.

In this paper, we develop a new ownership method for GANs, which can detect ownership infringement on both physical stealing and model extraction attacks. Our method claims the ownership of a GAN by leveraging common characteristics of a target model and its stolen models. The rationale for our method is that stolen models are derived from the target model while honest models are not. Thus, these common

This research was funded by the Luxembourg National Research Fund (FNR), grant reference 13550291.

Hailong Hu is with the National Research Base of Intelligent Manufacturing Service, Chongqing Technology and Business University, Chongqing 400067, China. He was with the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Esch-sur-Alzette 4365, Luxembourg. E-mail: huhailong@ctbu.edu.cn.

Jun Pang is with the Faculty of Science, Technology and Medicine and the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Esch-sur-Alzette 4365, Luxembourg. E-mail: jun.pang@uni.lu.

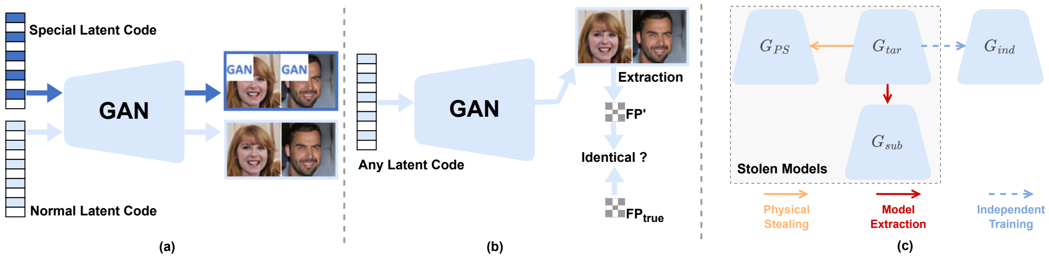


Fig. 1: Different paradigms of ownership infringement detection on GANs. (a) shows a watermark-based method that utilizes watermarked generated images to assert ownership. (b) illustrates a fingerprint-based method that leverages the extracted fingerprints from generated samples to confirm ownership. (c) is our proposed method that utilizes the common characteristics of a target model and its stolen models to claim ownership. The rationale of our method is that stolen models, as shown in the grey area, are derived from the target model and should have common characteristics among them. Here, stolen models include G_{PS} obtained through physical stealing and G_{sub} obtained via model extraction attacks. G_{ind} is an honest model that is trained independently.

characteristics can be leveraged to differentiate stolen models from honest models. More specifically, we utilize generated samples from GANs to build a discriminative classifier to learn these characteristics. This is because the objective of a GAN is to learn the distribution of a training dataset and the learned implicit distribution of a GAN can be represented by these generated samples.

We comprehensively evaluate our new method by comparing it with two state-of-the-art works: watermark-based method (abbreviation as Ong method) [9] and fingerprint-based method (abbreviation as Yu method) [12]. Here, note that fingerprint-based methods are proposed for deepfake detection and attribution [12, 13]. In this work, we are the first to introduce them to the field of ownership infringement detection, considering their common objective: fingerprints can be used to infer whether a suspect sample is from the model. Furthermore, we analyze the ownership detection performance by visualizing the characteristics learned by our method. We also show the stability of our method on the number of generations of model extraction attacks, which is a new emerging threat in the domain of generative AI. Additionally, we perform a comprehensive hyperparameter analysis, such as the effect of the number of generated samples, to assess the sensitivity of our method. Finally, we conduct extensive evaluations under adaptive attacks, where adversaries obtain partial knowledge of our method, to demonstrate its effectiveness.

In summary, we make the following contributions. (1) We show that previous work on watermark-based and fingerprint-based cannot provide ownership infringement detection under model extraction attacks. (2) We propose a brand-new detection method GAN-Guards from a new angle: detecting ownership infringement utilizing the common characteristics of a target model and stolen models. (3) Our method achieves a new state-of-the-art performance on both physical stealing and emerging model extraction attacks.

II. Related Work

Generative adversarial networks (GANs). The seminal GAN introduced in 2014 [3] presents a promising result in image synthesis. Since then, many methods are proposed to further advance the performance of GANs from various aspects, such as network architectures [14, 15], loss functions [16, 17] or normalization [18]. Karras et al. [14] propose a progressive training strategy that enables a GAN to synthesize high-resolution images. In addition to generating high-resolution images, Karras et al. [15] further introduce neural style transfer structures into the architecture of GANs and it empowers GANs to generate a variety of style images. StyleGAN2 [19] continues to enhance the quality of generated images by refining the model architecture and training methods, such as the adoption of progressive training, the application of regularization methods to the generator and the redesign of generator normalization. In the improvement of the loss of GANs, a Wasserstein distance loss is utilized to stabilize the training process of a GAN [16]. Gulrajani et al. [17] add a gradient penalizing term onto the Wasserstein distance loss to further improve the quality of synthetic images. In addition, Takeru et al. [18] introduce spectral normalization to normalize the weights of each layer. As a result, the whole GAN has Lipschitz continuity, which makes the training process stable. Zhang et al. [20] introduce a self-attention mechanism into the GAN architecture to achieve attention-driven image synthesis. In this work, instead of further improving the performance of GANs, we aim at developing an ownership infringement technique to protect the intellectual property of valuable GANs.

Ownership infringement detection. There are numerous works aiming to protect ownership of discriminative models via detecting ownership infringement [21]. These works can generally be classified into three groups: embedding watermarks into model parameters [22], using predefined inputs as triggers [23] and utilizing unique features of models [24, 25]. Uchida et al. [22] propose to encode a message, that is a watermark, into the weights of certain layers of a neural network. Adi et al. [23] introduce to watermark a neural network by backdooring. The watermark, that is the trigger dataset, consists of out-of-distribution images and corresponding labels. Model owners can verify the ownership by labels of watermarking images returned by the suspect model. Nie et al. [26] propose to construct a watermark dataset via image steganography and image compression algorithms. To utilize

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2025.3560921

unique characteristics of models to verify ownership, Maini et al. [24] propose a novel method by verifying whether a suspect model has private knowledge from the target model. Chen et al. [25] present an approach to verify the ownership by a family of multi-level testing metrics which characterize the similarities between victim models and suspect models. Xu et al. [27] propose to use deep Taylor decomposition to extract the intrinsic features of the model which are used for the verification of ownership. Beyond centralized learning scenarios, Nie and Lu [28] study the detection of ownership through watermarking on federated learning scenarios. Furthermore, they also propose to combine boundary samples and attention mechanisms to construct watermarks [29] and explore the embedding of watermarks in multimodal models [30]. However, these methods focus on discriminative models and cannot be applied to generative models since they are different machine learning models.

There are only a few works on ownership infringement detection methods of GANs and they focus on protecting ownership of GANs via watermarks [9, 10, 11]. Ong et al. [9] propose a protection framework for GANs by adding a novel regularization term to the existing loss function, which aims to force the generator to map a trigger input to a specific output. Fei et al. [10] propose to embed watermarks during the fine-tuning phase to reduce time cost. Qiao et al. [11] design a trigger set by combining the watermark label with the verification image, and they claim the ownership of a GAN model if the verification image can be obtained. Additionally, Huang et al. [31] propose a discriminator-based ownership infringement detection method by adding an extra Pearson correlation loss into the training process. However, these works do not consider emerging model extraction attacks. In this work, we will demonstrate that conventional watermark-based methods cannot defend against model extraction attacks while our method exhibits effectiveness in addressing such threats.

III. Background

A. Generative Adversarial Networks

An unconditional GAN generally consists of a generator G and a discriminator D. In the training phase, the generator G aims to generate fake data to fool the discriminator D while the discriminator D attempts to distinguish fake data from the generator from the real data from the training set. Once finishing training, the generator G can be utilized to generate data, given latent codes. Gaussian distribution or uniform distribution is commonly used to obtain latent codes. Mathematically, the generator of a GAN is a function $G : \mathcal{Z} \to \mathcal{X}$ that maps a low dimensional latent space $\mathcal{Z} \subseteq \mathbb{R}^n$ to a high dimension data space $\mathcal{X} \subseteq \mathbb{R}^m$.

B. Paradigms of Ownership Infringement Detection

Current paradigms of ownership infringement detection on GANs can be divided into two classes: watermark-based and fingerprint-based methods. We show them in Figure 1.

Watermark-based methods are initially proposed in the work [9]. The key idea is that model owners embed specific inputs and outputs into a GAN in the training phase. Then, if specific outputs (e.g. watermarked generated samples) can be obtained from a suspect GAN through specific inputs (e.g. triggers), model owners claim ownership of this GAN, as presented in Figure 1(a). To achieve this, Ong et al. propose a detection framework for GANs by adding a novel regularization term to the existing loss function [9]. For simplicity, we refer to this ownership detection method by the first author's name, i.e, Ong [9].

Fingerprint-based methods are initially proposed for deepfake detection and attribution [12, 32]. Here, we present the first exploration of applying these methods to detect GAN ownership infringement. This is because they share a common objective that fingerprints can be used to infer whether a suspect sample is from their model. Specifically, as illustrated in Figure 1(b), the key idea of fingerprint-based methods is that if the fingerprint extracted from generated samples from a GAN is identical to the true fingerprint, model owners can claim ownership of the GAN. To achieve this goal, various methods are proposed, such as adding fingerprints on the training set of a GAN [12], and designing new architectures of a GAN and loss functions [32].

In this paper, considering their excellent performance and the diversity of methods, we choose one watermark-based method—Ong [9] and one fingerprint-based method—Yu [12] to evaluate the performance in ownership protection and make comparisons with our proposed method. The Yu method [12] firstly incorporates fingerprints into the training dataset and a GAN is trained on the fingerprinted dataset. Subsequently, it verifies the presence of these fingerprints in the samples generated by GANs.

Note that although there are some works [13, 33, 34] about fingerprints of GANs, they focus on differentiating different types of GANs, such as PGGAN [14] and SNGAN [18]. Thus, we do not compare these methods because they cannot distinguish different models from the same type of GANs.

IV. A New Ownership Detection Method

Unlike the methods that require forcibly implanting watermarks or fingerprints into target models and retraining target models, our method provides a novel paradigm: the common characteristics of a target model and its stolen models are exploited to claim ownership, motivated by emerging model extraction attacks [8, 35].

A. Threat Model

We assume that defenders, i.e. model providers who deploy an ownership infringement detection method on their target model, only have access to generated samples from a suspect model deployed by the adversaries. Thus, the defenders make an ownership infringement decision only based on these generated samples. This is the most practical and strictest assumption for defenders.

We assume that adversaries can steal a target model by either physical stealing or model extraction attacks. Additionally, we consider more advanced attack capabilities, including: (a) obfuscation, where adversaries employ various obfuscation

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2025.3560921

TABLE I: Main notations.

Notation	Meaning
$\mathfrak{D}_{tar}$	a target dataset
G_{tar}	a target model trained on $\mathfrak{D}_{tar}$
$\mathfrak{D}_{ind}$	an independent dataset
G_{ind}	an independent model trained on $\mathfrak{D}_{ind}$
$\mathfrak{D}_{sub}$	a substitute dataset collected from G_{tar}
G_{sub}	a substitute model trained on $\mathfrak{D}_{sub}$
G_{sus}	a suspect model

techniques to evade detection (see Section VI-C), and (b) adaptive attacks, where adversaries have some knowledge of our method and devise targeted attacks to evade our detection (see Section VIII). By considering different attack capabilities, we aim to comprehensively evaluate our detection method.

B. Key Observations

Our method is based on the following two key observations.

The first key observation is that physical stealing and model extraction attacks are two fundamental but different types of ownership infringement. Physical stealing attacks refer that an adversary physically copies a model G_{sub} from the target model G_{tar}. Therefore, G_{sub} is totally the same as G_{tar}. Model extraction attacks [8] refer that an adversary retrains a substitute model G_{sub} on generated samples from a target model G_{tar}. These samples can be obtained by an adversary when model owners release generated samples or provide a querying interface. Thus, G_{sub} is functionally similar to G_{tar}.

The second key observation is that as stolen models (i.e. constructed by physical stealing or model extraction attacks) are derived from the target model but honest models are not, it is thus natural to assume that stolen models and the target model share common characteristics which do not exist in independently trained honest models. Therefore, we can learn and leverage such characteristics as evidence to differentiate stolen models from honest models. We illustrate and compare different paradigms of ownership infringement detection on GANs in Figure 1.

C. Ownership Infringement Detection Algorithm

Overview of our method. Since stolen models are derived from the target model, they share common characteristics. In order to extract these characteristics from a target model, our method proposes to learn them by training a binary classifier on generated samples. As a result, the learned characteristics are embedded in the classifier and it is utilized for ownership infringement detection. Generated samples from model extraction and physical stealing are labelled as positive while samples from honest models, i.e. independently trained models, are labelled as negative. The reason why we use generated samples is that a GAN model is to learn the distribution of a training set. The learned distribution is implicit, which can be represented through these generated samples [36]. We provide mathematical descriptions in Supplementary A-A. The main notations are summarized in Table I, and the process is illustrated in the deployment phase of Figure 2.

In practice, it is impossible for the defenders to consider all independently trained models and all models constructed by model extraction. Therefore, our method constructs positive and negative GAN models only by the limited knowledge of the defenders: the architectures of the target model and the training set. Specifically, the architectures of all models (i.e. G_{tar}, G_{sub}, G_{ind}) in the deployment phase are the same. The independent set $\mathfrak{D}_{ind}$ and the target set $\mathfrak{D}_{tar}$ are from the same distribution but disjoint. That is, the samples in $\mathfrak{D}_{tar}$ and $\mathfrak{D}_{ind}$ do not overlap. We emphasize that our method is practical as suspect models constructed by the adversaries can be trained on any (unknown for the defenders) GAN architectures and datasets. Our extensive experiments in Section VI demonstrate that our method can well generalize beyond these unknown GANs and correctly recognize them.

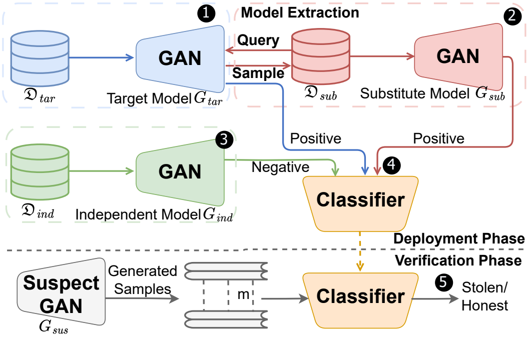


Fig. 2: Overview of our method. ❶ A target model is trained on a dataset $\mathfrak{D}_{tar}$. ❷ A substitute model is constructed by model extraction. ❸ An independent model is trained on a dataset $\mathfrak{D}_{ind}$ that has the same distribution as the dataset $\mathfrak{D}_{tar}$, but it is disjoint with $\mathfrak{D}_{tar}$. ❹ A classifier is trained to discriminate between stolen models and honest models. ❺ The classifier is used for the verification of a suspect model. Here, the target model can also refer to the physically stealing model. The defenders do not have any information about a suspect model G_{sus}, except generated samples, in the verification phase. Generated samples are obtained by feeding latent codes sampled from a Gaussian distribution into the suspect model G_{sus}. In practice, adversaries may employ various obfuscation techniques to release generated samples to evade ownership detection. In Section VI-C, we will evaluate the performance of our method under different obfuscations. During the deployment phase, blocks ❶, ❷, and ❸ represent GAN models, while block ❹ represents the classifier.

Details of the algorithm. As illustrated in the function *build-Protection* of Algorithm 1, specifically, given the generator of a target model G_{tar} and an independent dataset $\mathfrak{D}_{ind}$, we first construct a substitute model G_{sub} by extracting the target model G_{tar}. Next, we train a GAN G_{ind} on the independent dataset $\mathfrak{D}_{ind}$. Samples from G_{sub} and G_{tar} are labeled as positive while samples from G_{ind} are labeled as negative. These samples are used to train a classifier and in this work we choose ResNet50 [37] as the classifier.

After obtaining the trained classifier, we start to perform the verification of ownership. We first collect m generated samples released by a suspect model G_{sus}. These samples are fed into the classifier and m predictions can be obtained. We calculate the percentage of these positive predictions. If it is larger than a predefined threshold, the suspect model is inferred

Algorithm 1: The GAN-Guards Algorithm.

```
Input: a target model: $G_{tar}$; an independent dataset
       $\mathfrak{D}_{ind}$; $m$ samples $X_{sus}$ from a suspect model
       $G_{sus}$.
Output: ownership decision: $OwDecision$
def buildProtection($G_{tar}$, $\mathfrak{D}_{ind}$):
    Sample $\hat{n}$ samples $X_{gen}$ from $G_{tar}$;
    $G_{sub} \leftarrow$ trainGAN($X_{gen}$);
    $G_{ind} \leftarrow$ trainGAN($\mathfrak{D}_{ind}$);
    Sample $n$ samples $X_{gen}$ from $G_{tar}$; ▷ Labelling
      positive for physical stealing.
    Sample $n$ samples $X_{sub}$ from $G_{sub}$; ▷ Labelling
      positive for model extraction.
    Sample $2n$ samples $X_{ind}$ from $G_{ind}$; ▷ Labelling
      negative for the honest model.
    $Classifier \leftarrow$ trainClassifier($X_{gen}, X_{sub}, X_{ind}$);
    return $Classifier$

def performVerification($Classifier$, $X_{sus}$, $\tau$):
    Initialize prediction array $pred$ of length $m$ with 0;
    for $i = 0$ to $m - 1$ do
        $pred[i] \leftarrow Classifier(X_{sus}[i])$; ▷ Prediction: 1
          or 0.
    $ConfiScore = \mathrm{sum}(pred)/m$ ;
    ▷ Making a decision based on multiple samples.
    if $ConfiScore > \tau$ then $OwDecision = 1$;
    else $OwDecision = 0$;
    return $OwDecision$
```

as stealing from the target model. We also analyze how the number of generated samples m affects our performance in Section VII. This process is also illustrated in the function *performVerification* of Algorithm 1. Note that the classifier used for ownership verification should remain confidential. It is exclusively utilized by model owners and does not provide any interface, such as querying, with users or adversaries.

V. Experiments

A. Datasets

We evaluate our method on three benchmark datasets commonly used in image generation: FFHQ [15], Church [38] and CelebA [39]. The FFHQ dataset is designed for human face image synthesis and includes 70,000 images. The Church dataset is from the LSUN dataset, which contains 126,277 outdoor church images. The CelebA dataset is a much larger human face dataset, which includes 202,599 images.

All images are resized to 64 × 64. For each dataset, we randomly split the dataset into three disjoint equal parts and mark each part as the corresponding dataset name plus ‘I’, ‘II’, and ‘III’, respectively, such as FFHQ-I and FFHQ-II. Dataset I, i.e. $\mathfrak{D}_{tar}$, is used to train a target GAN model. Dataset II is used to train a GAN and the model (i.e. Ind-a illustrated in the following Section V-B) is also used as negative to test the performance of detection methods. Dataset III, i.e. $\mathfrak{D}_{ind}$, is used to train a GAN model and the model is used to build a classifier together with the target model. Specifically, we set the size of each part of FFHQ, Church and CelebA as 20,000, 40,000 and 60,000, respectively. Note that experimental results on the Church dataset are presented in Supplementary A-E1.

B. Suspect Models

We consider various suspect models. Positive suspect models are considered ownership infringement and these models are derived from the target models via physical stealing and model extraction, and obfuscation attacks, such as input perturbation, output perturbation, overwriting, and fine-tuning attacks. Negative suspect models are honest models and they are built from independent training.

Specifically, for positive suspect models, models from physical stealing (marked as PS) are the same as target models. We use model extraction attacks proposed in the work [8] to construct models from model extraction (marked as ME). Specifically, given m generated samples from a target model G_{tar}, the adversaries retrain a model (also called the substitute model or attack model) on m generated samples. Attack models can use any architecture, such as SNGAN, PGGAN, or StyleGAN. We will explore the detection performance under model extraction attacks with different GANs as attack models in Section VI-D.

For negative suspect models, similar to the settings used in the work [25], here we also consider two types: *Ind-a* trained on dataset II with the same architectures of target models, and *Ind-b* that is trained on dataset I, with the same architectures of target models but uses different seeds, i.e. different random initializations. Using different seeds means models trained on different training environments and optimization processes. Theoretically, models trained with different seeds should be different, because they do not derive from model extraction and physical stealing, and they are honest models with independent training. Thus, an ownership infringement detection method should be able to differentiate them. Here, setups for negative suspect models are very similar to those for target models because we aim to test whether an ownership infringement detection method hurts honest model providers in the strong assumption setting. This requires that a detection method should be extremely robust.

C. Metrics

We use FID [40] to measure the performance of a GAN. 50K generated samples from a GAN and all training samples are used to compute the FID value.

In terms of detection performance, the Ong method [9] utilizes the SSIM [41] score to measure the similarity between the groundtruth watermark and the watermark extracted from a suspect model. If the SSIM score of an image is higher than a threshold, the image is more likely from the target model. The Yu method [12] calculates a bitwise accuracy between the groundtruth fingerprint and an extracted fingerprint. Claiming ownership of a model based on only one image is not robust enough. Therefore, we make a final decision by computing a confidence score on multiple samples. Specifically, given m samples and each sample gets an output $o \in \{0, 1\}$ from a detection method, the confidence score that recognizes a suspect model as positive is computed by: $ConfidenceScore = \frac{\sum_{i=0}^{m-1} o_i}{m}$. In this work, we set threshold τ

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2025.3560921

TABLE II: Performance of target model SNGAN trained on FFHQ-I on different methods. Lower FID (↓) indicates better performance, with standard deviations shown in parentheses.

Methods	Ong	Yu	Ours
$\text{FID}(\mathfrak{D}, \tilde{G}) \downarrow$	20.13 (0.07)	26.41 (0.03)	20.27 (0.02)

of all methods as 90% for consistency. Thus, a suspect model is predicted as positive (stolen model) if $\tau \geq 90\%$. We fix the number of samples m as 1,000.

D. Experimental Setups

In terms of GANs, we utilize five types of GANs, including SNGAN [18], PGGAN [14], StyleGAN [15], SAGAN[20], and StyleGAN2 [19]. These GANs are widely recognized in the image generation community and are known for their excellent performance in image synthesis. We use the official implementation of each GAN to train GANs. For model extraction attacks, considering the trade-off between attack cost and performance, we set the number of generated samples as 50,000, which is also suggested by the work [8].

For our ownership infringement detection method, we use ResNet50 [37] pretrained on ImageNet [42] dataset for our classifier. The SGD optimizer with a learning rate of 0.003 is used and the number of epochs is fixed as 5. As shown in Algorithm 1, latent codes sampled from a Gaussian distribution are fed into the target model to obtain generated samples, and the number of samples n is set as 100,000. Therefore, 400,000 samples in total are used for training the classifier. For the Ong method [9] and the Yu method [12], we adopt the official implementations with suggested hyperparameters.

VI. Evaluation

In this section, we present our results by comparing our method with two state-of-the-art methods: the Ong method [9] and the Yu method [12]. We evaluate them from various perspectives, including model utility, verification performance, robustness to obfuscations, and robustness to more model extractions. Through these evaluations, we show how prior works fail in model extraction attacks and our work can perform well on both physical stealing attacks and model extraction attacks.

A. Model Utility of Target Models

Table II shows the performance of the target model SNGAN trained on FFHQ-I with different ownership infringement detection methods. The FID is computed by the original training set $\mathfrak{D}$ and the protected GAN $\tilde{G}$. The FID is the average of three runs and the standard deviation is shown in parentheses.

Overall, the watermark-based method Ong and our method achieve similar outstanding performance, while the fingerprint-based method Yu shows worse performance. This is because the Yu method needs to add fingerprints into a training set, which is at the cost of sacrificing model utility. In contrast, the Ong method and our method do not change the training set, which can provide model owners with higher model utility.

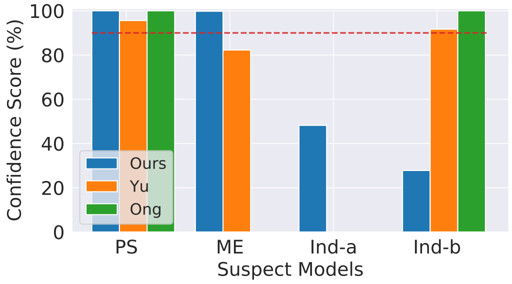


Fig. 3: Detection performance of all methods. The target model SNGAN is trained on FFHQ-I. PS and ME are positive suspect models while Ind-a and Ind-b are negative suspect models. Note that green and orange bars in some cases cannot be observed because their scores are 0%.

B. Detection Performance

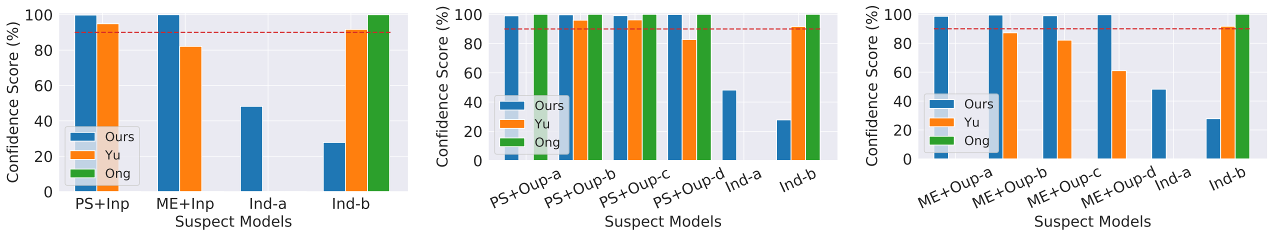
Figure 3 presents the performance of different ownership infringement detection methods. The red dashed line is the threshold τ of the confidence score. A model is predicted as a stolen (positive) model if $\tau \geq 90\%$. PS refers to models from physical stealing while ME refers to models from model extraction. Note that here ME, Ind-a, Ind-b models used in the verification phase are not the same models used in our deployment phase (detailed in Experiments Section V). We choose these suspect models ME, Ind-a and Ind-b with the best performance, and we report the performance of each suspect model in Supplementary A-G.

Overall, our method can correctly differentiate all positive and negative suspect models, achieving 100% accuracy. The Ong and Yu methods cannot defend against model extraction attacks although they can effectively recognize the positive suspect model from PS.

Additionally, the Ong and Yu methods mistakenly recognize the suspect model Ind-b trained with different initializations as a stolen model. This is because embedded watermarks or fingerprints cannot be changed only owing to different initializations of a training process. Thus, their methods lead to false alarms and hurt honest model providers. In contrast, our method can perfectly deal with this case because our method builds on a well-trained model. Theoretically, models trained with different initializations should be different, because they do not derive from model extraction and physical stealing, and an ownership infringement detection method should be able to differentiate them. This also shows that our method can be used for detecting ownership of a well-trained GAN model, rather than ownership of a (training) dataset. Note that the setting of the suspect model Ind-b is also adopted by classification models to detect whether an ownership infringement detection method produces false alarms [25].

C. Robustness to Obfuscations

In order to evade ownership infringement detection, advanced adversaries may utilize obfuscation techniques to obfuscate stolen models. In this work, we consider four types of obfuscation techniques: input perturbation, output perturbation, overwriting and fine-tuning. Input perturbation aims to modify the queries, i.e., latent codes, to evade special queries. Here, we adopt random input perturbation. That



(a) Input perturbation.

(b) PS + output perturbation.

(c) ME + output perturbation.

Fig. 4: Robustness to Obfuscations. Figure 4(a) shows detection performance against input perturbation, while Figure 4(b) and Figure 4(c) show detection performance against output perturbation under the cases of the physical stealing and model extraction, respectively. The target model is SNGAN trained on FFHQ-I. Green and orange bars in some cases cannot be observed because their scores are 0% and cannot defend against these attacks.

is, for any query, a target model resamples latent codes from Gaussian distribution. For brevity, we rename it Inp. Output perturbation refers to perturbing generated samples by various post-processing techniques. We use four different output perturbations: additive Gaussian noise, Gaussian filter, Gaussian blurring, and JPEG compression. We briefly rename them Oup-a, Oup-b, Oup-c, and Oup-d, respectively. The magnitude of these perturbations is set as 0.01, 0.4, 0.5, and 0.85, respectively. Overwriting refers to encoding a different watermark/fingerprint to overwrite the original watermark/fingerprint. Our method does not rely on watermarks and fingerprints, thus intrinsically eliminating the threat of this attack. In this work, we consider wholly fine-tuning where we take the weights of the stolen model as initialization and retrain a GAN model on a different dataset FFHQ-II. Because these obfuscation operations can be added into physical stealing (PS) or model extraction (ME), there are different combinations between obfuscation operations and PS and ME. Here, we mark them as 'PS+' and 'ME+' corresponding obfuscation operations, such as PS+Inp. Implementation details are also illustrated in Supplementary A-B.

Results. Figure 4 shows the robustness of different ownership infringement detection methods under input and output perturbation operations. Overall, our method can still remain 100% accuracy against input perturbation and output perturbation attacks. In contrast, as shown in Figure 4(a), the Ong method totally cannot resist the input perturbation attack. As depicted in Figure 4(b), the Yu method cannot defend against additive Gaussian noise of output perturbation attacks (PS+Oup-a and ME+Oup-a). Again, Figure 4(c) shows that the Ong and Yu methods cannot defend against ME+Output perturbation. We analyze the reasons for the Ong and Yu methods in Supplementary A-C.

We perform the evaluation under the overwriting attack. We do not report results for our method because our method does not rely on watermarks or fingerprints. Overall, the Ong and Yu methods cannot defend against this type of attack and both confidence scores are 0%. It indicates that the overwritten watermarks and fingerprints make their methods unable to extract the expected outputs. We provide the experimental details in Supplementary A-D.

We evaluate the protection performance under the fine-tuning attack. We observe that all methods are not robust to the fine-tuning attack. We analyze that this is because the fine-tuned GAN model has learned a different distribution in a new training set and neural networks suffer from catastrophic forgetting [43, 44]. The former makes that our method recognizes this model as an independent training model while the latter makes that the Ong and Yu methods forget embedded watermarks and fingerprints. This also inspires us to think about the ownership boundary of a GAN and develop more powerful ownership infringement detection methods in future. We summarize the results in Table A.1 in Supplementary.

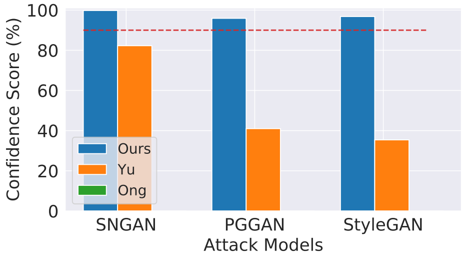


Fig. 5: Detection performance under model extraction attacks with different GANs as attack models. The target model SNGAN is trained on FFHQ-I.

D. *Robustness to More Model Extraction*

When mounting model extraction attacks, adversaries can utilize various architectures of GANs to extract a target model. Figure 5 presents a comprehensive comparison of various ownership infringement detection methods, focusing on the robustness of protection methods to model extraction attacks that utilize different GANs as attack models. Here, the target model is SNGAN.

Our evaluations reveal that the proposed method demonstrates remarkable efficacy in this scenario, consistently identifying and flagging models that have been derived through model extraction attacks. This is in stark contrast to the performance of the Ong and Yu methods, which erroneously categorize these models as legitimate and honest models. The key implication of this observation is that our proposed method exhibits a unique and robust capability to detect models constructed via model extraction attacks, regardless of the specific GAN architectures employed by the adversaries. This underscores the effectiveness of our method in safeguarding against a broad spectrum of model extraction strategies.

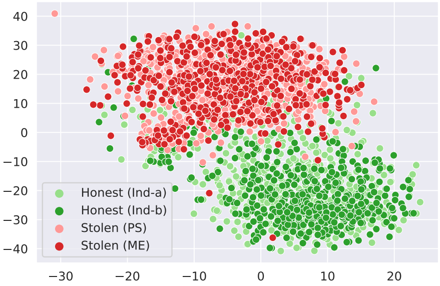


Fig. 6: t-SNE visualization of characteristics learned by our method for stolen models and honest models. The x- and y-axes represent the feature values produced by the t-SNE.

VII. Analysis

In this section, we intensively examine the detection performance of our method in terms of the learned characteristics, the number of generations of model extraction attacks, the number of generated samples, and different thresholds τ. Additionally, we provide a comparative analysis on a new dataset CelebA. Further results, including detection performance on the Church dataset, evaluations with different target models, and performance on transfer learning attacks, are presented in Supplementary A-E, which all show excellent performance.

A. Visualization of Characteristics

Figure 6 shows the t-SNE visualization of characteristics learned by our method. The visualization is generated using the t-SNE implementation from Scikit-learn [45]. The default parameters are used, including a perplexity of 30 and PCA initialization. The t-SNE input is the outputs of the penultimate layer of the classifier, which have a dimensionality of 2,048. The characteristics of the stolen models are represented by pink and red dots. We clearly see that characteristics of stolen models including PS and ME are entangled together and have a clear boundary with that from honest models.

B. Generations of Model Extraction Attacks

Theoretically, model extraction attacks on GANs can continue forever like the process of biological heredity, as shown in Figure 7. Models produced during this process, such as $G^{(i)}$, should be correctly identified by ownership infringement detection methods. This motivates us to investigate whether the detection performance will decrease with the number of generations of model extraction attacks. We emphasize this is our newly identified threat, which is not discussed in the literature. Moreover, this threat will become more common considering the popularity of generative AI.

Here, we fix the number of generated samples as 1,000 and the target model is SNGAN trained on FFHQ-I. We denote the target model SNGAN as SNGAN$^{(0)}$. Subsequent generations of extracted models are denoted as SNGAN$^{(i)}$, where i represents the generation number. Each generation SNGAN$^{(i)}$ is obtained by performing model extraction on SNGAN$^{(i-1)}$ using SNGAN as the attack model architecture. For example, the first generation of model extraction is marked as SNGAN$^{(1)}$, which means an adversary uses an attack model SNGAN to extract the target model SNGAN. We do not show the performance of the Ong method because it cannot defend against model extraction attacks.

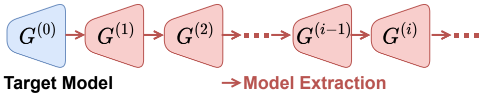


Fig. 7: Generations of model extraction attacks.

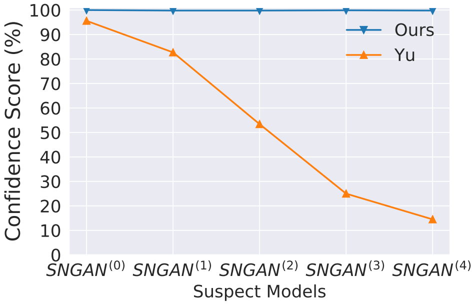


Fig. 8: Detection performance with regard to the number of generations of model extraction attacks.

As shown in Figure 8, we can clearly observe that with the increase in the number of generations of model extraction, the Yu method becomes less and less confident. It also indicates that the fingerprint is not robust, and more and more generated samples cannot extract the corresponding fingerprint. In contrast, our method still remains almost 100% confident to verify ownership of the target model.

C. Number of Generated Samples

Figure 9 presents the detection performance of our method under the different numbers of generated samples, i.e. the number of queries. The target model SNGAN is trained on FFHQ-I. The ground truth of PS and ME is positive while that of Ind-a and Ind-b is negative. We can clearly see that the confidence scores gradually remain stable after 1,000 generated samples on all suspect models. It also shows that our method has advantages with respect to the *efficiency*, i.e. it requires as few as 1,000 samples.

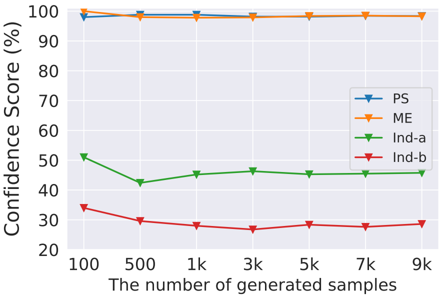


Fig. 9: Detection performance with respect to the numbers of generated samples.

D. Effects of Different Thresholds

Figure 10 shows the detection performance with regard to the threshold τ. The target model SNGAN is trained on FFHQ-I. The positive suspect models are shown in the x-axis from PS to ME-StyleGAN while the negative suspect models are shown in the y-axis from Ind-a to Ind-b. ME and ME+ are

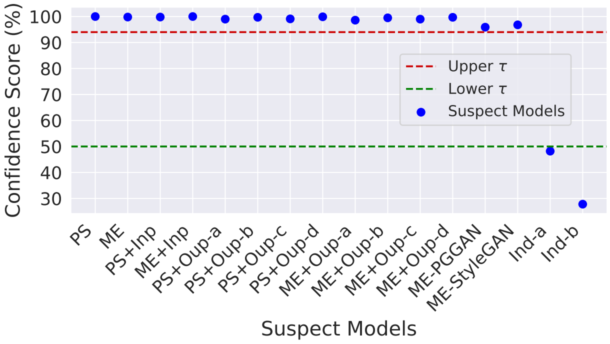


Fig. 10: Detection performance with regard to the threshold τ.

models obtained via model extraction attacks and the attack models are SNGANs. ME-PGGAN and ME-StyleGAN are models obtained via model extraction attacks and the attack models are PGGAN and StyleGAN, respectively.

Theoretically, if an ownership infringement detection method is perfect, the upper bound of τ is the minimum confidence score of the positive (stolen) models, and the lower bound of τ is the maximum confidence score of negative (honest) models. The range of upper and lower bounds can be considered as the tolerance degree of the detection method. In this work, we set τ as 90% to ensure that the detection method identifies positive suspect models with a high degree of confidence.

E. Additional Comparison on the CelebA Dataset

To further demonstrate the generalization of our proposed method, we conduct a comparative analysis with the Huang method [31] on a new dataset CelebA.

The Huang method is a discriminator-based method for detecting ownership infringement in GANs. This method begins by incorporating an additional Pearson correlation loss into the training of the target GAN model. Once the GAN training is complete, it performs fine-tuning on the discriminator using samples generated by the target model. The fine-tuned discriminator is then used to detect ownership infringement of a suspect model. Specifically, the method compares generated samples from both the suspect and target models. If the Area Under Curve (AUC) score for the discriminator on these samples approaches 0.5, the suspect model is classified as a stolen model. In this work, we adapt the evaluation metric to a unified confidence score for consistency. Specifically, the AUC score is transformed by: $ConfidenceScore = 1 - |AUCScore - 0.5| \times 2$. A threshold of 90% was set for the confidence score to determine ownership infringement. For this comparison, we choose the SNGAN trained on the CelebA-I dataset as the target model. We utilize the official implementation of the Huang method and report the best FID values for the target models achieved by the target models across training snapshots.

In terms of model utility, our method significantly outperforms the Huang method, achieving an FID value of 11.62 compared to 44.45 with the Huang method. This result highlights the effectiveness of our method in preserving model utility. The reduced utility in the Huang method can be attributed to its additional loss term during training, which introduces instability into the training process and may compromise the model's performance.

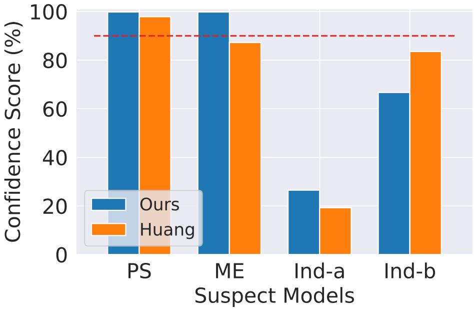


Fig. 11: Detection performance on the CelebA dataset. The target model is SNGAN.

Figure 11 presents the detection performance of our method and the Huang method on the CelebA dataset. Overall, our method consistently outperforms the Huang method. Both methods correctly identify the physical stealing model (PS). However, the Huang method demonstrates lower confidence scores compared to our method when identifying the model extraction model (ME).

Figure 12 illustrates the detection performance of both methods under input and output perturbation attacks. As shown in Figure 12(a), input perturbation attacks do not affect the detection performance of either method. For instance, both methods effectively defend against input perturbations in the PS+Inp scenario. As depicted in Figure 12(b) and Figure 12(c), our method remains robust under output perturbation attacks. In contrast, the Huang method struggles to defend against such attacks in cases like PS+Oup-a, where additive Gaussian noise is applied to generated samples.

Figure 13 evaluates detection performance under additional model extraction attacks using SAGAN and StyleGAN2 as attack models. Our method achieves consistently high confidence scores when identifying model extraction attacks, significantly outperforming the Huang method.

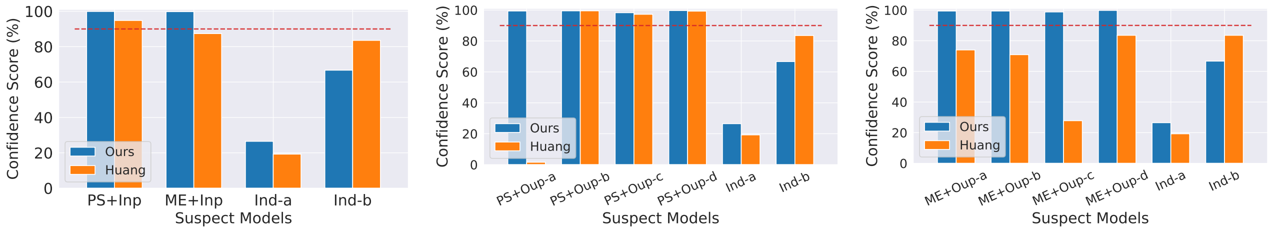
In summary, our method exhibits superior model utility and detection performance compared to the Huang method.

VIII. Adaptive Attacks

In this section, we present the performance of our method under adaptive attacks. That is, we assume that adversaries have some knowledge of our ownership infringement detection method, and design a series of specific attacks to evade it.

We discuss two types of adaptive attack scenarios. The main design principle is that we assume that adversaries perceive our method which is based on the common characteristics of a target model and its stolen models. Therefore, the adversaries attempt to decrease the confidence score of our method by sacrificing model utility (i.e. the quality of generated images). Specifically, in adaptive attack I, adversaries choose an inferior performance GAN from multiple snapshots of a GAN when mounting model extraction attacks. In adaptive attack II, adversaries evade our detection by designing a series of output perturbations by choosing the magnitude of the perturbation.

Figure 14 shows the detection performance under the adaptive attack I. Here, we choose an attack model SNGAN to extract the target model SNGAN trained on FFHQ-I. We



(a) Input perturbation.

(b) PS + output perturbation.

(c) ME + output perturbation.

Fig. 12: Robustness to Obfuscations. Figure 12(a) illustrates detection performance under input perturbation. For output perturbation, detection performance is shown in Figure 12(b) for physical stealing and in Figure 12(c) for model extraction. The target model SNGAN is trained on CelebA-I.

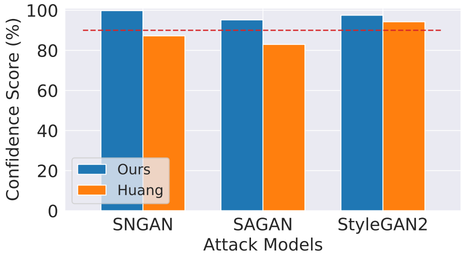


Fig. 13: Detection performance under model extraction attacks with different GANs as attack models. The target model SNGAN is trained on CelebA-I.

choose eight snapshots of SNGAN during model extraction attacks. The performance of the attack model SNGAN, i.e. FID($p_g, p_{\tilde{g}}$), ranges from 22 to 2, as depicted in the red line. p_g and $p_{\tilde{g}}$ are the implicit distribution of the target model and the attack model, respectively. We can observe that confidence scores begin to decrease, then increase and remain at 100%, with the decrease in FID of the attack model SNGAN. In particular, the confidence scores of all snapshots are above 98%, which indicates that our method can correctly recognize all snapshots as stolen models.

Figure 15 shows the detection performance under the adaptive attack II. Considering the model utility, we design three strategies (strategy I, strategy II and strategy III) based on different magnitudes of four types of output perturbation. In Supplementary A-F, we show the magnitudes of output perturbation of each strategy in Table A.2, and we visually present images generated by each strategy in Figure A.7. The quality of generated images becomes much noisier and blurrier from strategy I to strategy III. Overall, we can observe in Figure 15 that our method still can recognize all positive suspect models, although the confidence score of each suspect model decreases from strategy I to strategy III. In addition, although strategy III can lower the confidence of our method, the model almost cannot be used due to the low quality of generated images visually. In practice, the loss in model utility can make the adversaries less competitive in market share, compared to legitimate model owners.

IX. Discussion

Our proposed method utilizes the common characteristics shared between the target model and its stolen models to detect ownership infringement. This is due to the fact that these stolen models are derived from the target model, and thus preserve the common characteristics among them. In contrast, independently trained honest models do not exhibit these characteristics. The robustness of our method is demonstrated across various adversarial scenarios, such as obfuscation techniques and adaptive attacks. In practice, by deploying this detection method, model providers can not only deter adversaries from stealing their models but also assert rightful ownership when disputes arise. Ultimately, it can promote the ethical and responsible use of generative AI.

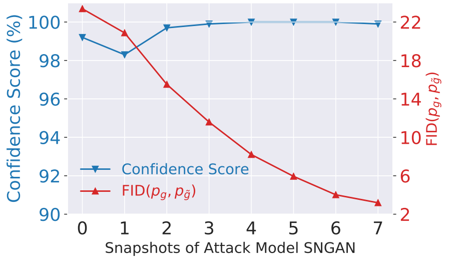


Fig. 14: Detection performance under the adaptive attack I. The target model SNGAN is trained on FFHQ-I. With the increase in the performance of the attack model SNGAN, i.e. FID($p_g, p_{\tilde{g}}$) decreasing from 22 to 2, the confidence score of our method still remains above 98%.

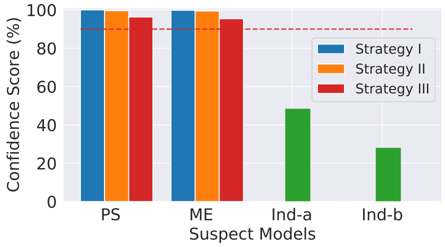


Fig. 15: Detection performance under the adaptive attack II. The target model SNGAN is trained on FFHQ-I.

X. Conclusion

In this paper, we have proposed a novel method to detect GAN ownership infringement by leveraging the common characteristics of a target model and its stolen GANs. Extensive experimental evaluations demonstrate that: (a) In terms of model utility, our method can bring lossless fidelity, compared to models without protection, because it does not modify well-trained target models. (b) In terms of robustness, our method can achieve new state-of-the-art detection performance, compared with watermark-based methods and fingerprint-based methods. Furthermore, we have also shown that our method is still effective under two types of carefully designed adaptive attacks. (c) In terms of undetectability, our method is undetectable for adversaries because it builds on a target model with normal training and does not rely on watermarks or fingerprints. (d) In terms of efficiency, our method requires about 1,000 generated samples to confidently verify the ownership of a GAN. Finally, we also performed a fine-grained analysis of our method from various aspects, such as visualizing learned characteristics, the stability of the performance with regard to the number of generations of model extraction attacks and the number of generated samples.

Fine-tuning attacks remain a challenge for ownership protection on GANs. In future, we plan to design more powerful methods to defend against these types of attacks. In addition, applying our detection method to other domains, such as table data synthesis and text generation, is also an interesting research direction.

References

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Proceedings of Annual Conference on Neural Information Processing Systems (NeurIPS)*, vol. 33. Curran Associates, 2020, pp. 1877–1901.

[2] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 10 684–10 695.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of Annual Conference on Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2014, pp. 2672–2680.

[4] A. Sauer, K. Schwarz, and A. Geiger, "Stylegan-xl: Scaling stylegan to large diverse datasets," in *Proceedings ACM SIGGRAPH Conference (SIGGRAPH '22)*. ACM, 2022, pp. 1–10.

[5] Y. Shen and B. Zhou, "Closed-form factorization of latent semantics in gans," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 1532–1540.

[6] W. Xia, Y. Zhang, Y. Yang, J.-H. Xue, B. Zhou, and M.-H. Yang, "Gan inversion: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3121–3138, 2023.

[7] E. E. Schultz, "A framework for understanding and predicting insider attacks," *Computers & Security*, vol. 21, no. 6, pp. 526–531, 2002.

[8] H. Hu and J. Pang, "Stealing machine learning models: Attacks and countermeasures for generative adversarial networks," in *Annual Computer Security Applications Conference (ACSAC)*. ACM, 2021, pp. 1–16.

[9] D. S. Ong, C. S. Chan, K. W. Ng, L. Fan, and Q. Yang, "Protecting intellectual property of generative adversarial networks from ambiguity attacks," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 3630–3639.

[10] J. Fei, Z. Xia, B. Tondi, and M. Barni, "Supervised gan watermarking for intellectual property protection," in *Proceedings of IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2022, pp. 1–6.

[11] T. Qiao, Y. Ma, N. Zheng, H. Wu, Y. Chen, M. Xu, and X. Luo, "A novel model watermarking for protecting generative adversarial network," *Computers & Security*, vol. 127, p. 103102, 2023.

[12] N. Yu, V. Skripniuk, S. Abdelnabi, and M. Fritz, "Artificial fingerprinting for generative models: Rooting deepfake attribution in training data," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2021, pp. 14 448–14 457.

[13] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, "Do gans leave artificial fingerprints?" in *Proceedings of IEEE conference on multimedia information processing and retrieval (MIPR)*. IEEE, 2019, pp. 506–511.

[14] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for quality, stability, and variation," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[15] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 4401–4410.

[16] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proceedings of Annual Conference on Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2016, pp. 2234–2242.

[17] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Proceedings of Annual Conference on Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2017.

[18] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[19] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 8107–8116.

[20] H. Zhang, I. J. Goodfellow, D. N. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proceedings of International Conference on*

Machine Learning (ICML). PMLR, 2019.

[21] M. Xue, Y. Zhang, J. Wang, and W. Liu, "Intellectual property protection for deep learning models: Taxonomy, methods, attacks, and evaluations," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 6, pp. 908–923, 2021.

[22] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *Proceedings of ACM International Conference on Multimedia Retrieval (ICMR)*. ACM, 2017, pp. 269–277.

[23] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *Proceedings of USENIX Security Symposium (USENIX Security)*. USENIX Association, 2018, pp. 1615–1631.

[24] P. Maini, M. Yaghini, and N. Papernot, "Dataset inference: Ownership resolution in machine learning," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

[25] J. Chen, J. Wang, T. Peng, Y. Sun, P. Cheng, S. Ji, X. Ma, B. Li, and D. Song, "Copy, right? a testing framework for copyright protection of deep learning models," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2022, pp. 1013–1030.

[26] H. Nie, S. Lu, J. Wu, and J. Zhu, "Deep model intellectual property protection with compression-resistant model watermarking," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 7, pp. 3362–3373, 2024.

[27] M. Xue, X. Wang, Y. Wu, S. Ni, L. Y. Zhang, Y. Zhang, and W. Liu, "An explainable intellectual property protection method for deep neural networks based on intrinsic features," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 9, pp. 4649–4659, 2024.

[28] H. Nie and S. Lu, "Fedcrmw: Federated model ownership verification with compression-resistant model watermarking," *Expert Systems with Applications*, vol. 249, p. 123776, 2024.

[29] ——, "Persistverify: Federated model ownership verification with spatial attention and boundary sampling," *Knowledge-Based Systems*, vol. 293, p. 111675, 2024.

[30] ——, "Securing ip in edge ai: neural network watermarking for multimodal models," *Applied Intelligence*, vol. 54, no. 21, pp. 10 455–10 472, 2024.

[31] Z. Huang, B. Li, Y. Cai, R. Wang, S. Guo, L. Fang, J. Chen, and L. Wang, "What can discriminator do? towards box-free ownership verification of generative adversarial networks," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2023, pp. 5009–5019.

[32] N. Yu, V. Skripniuk, D. Chen, L. S. Davis, and M. Fritz, "Responsible disclosure of generative models using scalable fingerprinting," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2022.

[33] N. Yu, L. S. Davis, and M. Fritz, "Attributing fake images to gans: Learning and analyzing gan fingerprints," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 7556–7566.

[34] Y. Ding, N. Thakur, and B. Li, "Does a gan leave distinct model-specific fingerprints," in *Proceedings of the BMVC*, 2021.

[35] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," in *Proceedings of USENIX Security Symposium (USENIX Security)*. USENIX Association, 2016, pp. 601–618.

[36] Y. Bahri, J. Kadmon, J. Pennington, S. S. Schoenholz, J. Sohl-Dickstein, and S. Ganguli, "Statistical mechanics of deep learning," *Annual Review of Condensed Matter Physics*, vol. 11, no. 1, pp. 501–528, 2020.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 770–778.

[38] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[39] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 3730–3738.

[40] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proceedings of Annual Conference on Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2017, pp. 6626–6637.

[41] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[42] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[43] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[44] I. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," *arXiv preprint arXiv:1312.6211*, 2013.

[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.