What can Discriminator do? Towards Box-free Ownership Verification of Generative Adversarial Networks

Ziheng Huang[1†], Boheng Li[1†], Yan Cai[1], Run Wang[1*], Shangwei Guo[2],
Liming Fang[3], Jing Chen[1], Lina Wang[1]
[1] Key Laboratory of Aerospace Information Security and Trusted Computing,
Ministry of Education, School of Cyber Science and Engineering, Wuhan University, China
[2] College of Computer Science, Chongqing University, China
[3] College of Computer Science and Technology, Nanjing University of
Aeronautics and Astronautics, China
[†] Equal contribution [*] Corresponding author. E-mail: wangrun@whu.edu.cn

Abstract

*In recent decades, Generative Adversarial Network (GAN) and its variants have achieved unprecedented success in image synthesis. However, well-trained GANs are under the threat of illegal steal or leakage. The prior studies on remote ownership verification assume a black-box setting where the defender can query the suspicious model with specific inputs, which we identify is not enough for generation tasks. To this end, in this paper, we propose a novel IP protection scheme for GANs where ownership verification can be done by checking outputs only, without choosing the inputs (*i.e.*, box-free setting). Specifically, we make use of the unexploited potential of the discriminator to learn a hypersphere that captures the unique distribution learned by the paired generator. Extensive evaluations on two popular GAN tasks and more than 10 GAN architectures demonstrate our proposed scheme to effectively verify the ownership. Our proposed scheme shown to be immune to popular input-based removal attacks and robust against other existing attacks. The source code and models are available at https://github.com/AbstractTeen/gan_ownership_verification.*

1. Introduction

With the rapid development of GANs, we have witnessed fruitful applications of GAN in many fields, such as realistic facial images synthesis [45], fine-grained attribute editing [55], *etc*. Unlike the classification model with specified label prediction, the GANs learn a data distribution and output the synthesized data sample within a certain distribution. In GANs, the discriminator and generator are two essential components, where the discriminator works as a judger to discriminate whether the sample is produced by the generator, the generator learns to generate more realistic samples to confuse the discriminator [16, 7]. Usually, the discriminator is discarded after training since the generator is the core asset for synthesizing high-quality images.

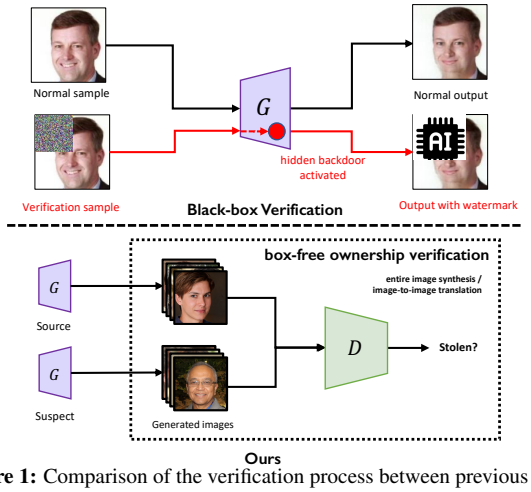


Figure 1: Comparison of the verification process between previous black-box watermark-based verification paradigm [37] and our box-free method. In the black-box setting, carefully-crafted verification samples should be fed to the suspicious model to activate a hidden backdoor in the model (the red circle) and generate watermarked outputs. However, in box-free setting, querying the model with deterministic inputs is not allowed. Ownership verification should be done with only output images.

Training a decent GAN requires a huge investment of resources, such as computing resources, labeled/unlabeled training dataset, time, and human labors [47, 31]. However, well-trained generators are under the threat of unintentional leakage and theft. The adversary may deploy the stolen model on the Internet for profit and the owner (also the *defender*) is only able to verify the ownership remotely by querying the suspicious model [35, 37, 23, 14].

Most existing works on IP protection of DNN models as-

sume the owner can query the suspicious model with specific inputs (*i.e.*, black-box setting). Based on this fundamental assumption, two schools of solutions have been proposed: model watermark [64, 37, 29] and model fingerprint [39, 34]. They either proactively embed or passively extract a hidden functionality in the model, where outlier outputs can be activated by a specific query set (known as the *verification set*). For verification, the defender queries the model with this set and observes whether the outputs match the source model. Since it is improbable for any other model to perform the same abnormal behavior, the owner can judge whether the suspicious model is a stolen copy.

However, the black-box assumption is challenged in generation tasks. For example, the whole-image synthesis task takes a randomized latent representation as input. In reality, this representation is usually sampled from a pre-defined distribution, such as normal distribution. Therefore, the adversaries can prohibit the verification by sampling the latent representation themselves. The black-box methods are also shown to be vulnerable to input transformation-based removal attacks [19, 53]. Moreover, recent literature has shown the outlier verification samples can be detected, inspected, or reverse-engineered [18, 56]. Note that the verification set plays a similar role to private keys in cryptography [22, 30]. Once it is disclosed, the adversaries are capable to launch ambiguity attacks, or invalidate the watermark/fingerprint via methods such as adversarial training. These limitations inspire us to raise an important question: *can ownership verification be done via checking outputs only, without choosing the inputs (*i.e.*, box-free setting)?* This setting is more challenging because queries made by the defender are totally equivalent to those of normal users. There is no chance to activate a hidden functionality.

In this paper, we make the first attempt to box-free ownership verification of GANs. Based on the fact that GAN suffers from unstable training, we reveal the unexploited potential of the discriminator to capture the model-specific distribution learned by the paired generator. We utilize the discriminator's representations to learn a network featuring a hypersphere that encloses the distribution learned by the generator. Our proposed scheme does not require specifying any input nor training additional detection networks, the ownership verification can be done effectively via feeding a batch of suspicious images to the learned network. However, due to the gradual degradation issue, it is challenging for the discriminator to extract meaningful feature representations without sacrificing the performance of the generator. To tackle this problem, we leverage the pearson correlation coefficient [38] to quantify the implicit reconstruction ability of the discriminator, and prevent the degradation via adding the term into the loss function of the discriminator.

To comprehensively evaluate the effectiveness and robustness of our proposed method, we conduct experiments on two popular GAN tasks (*i.e.*, entire image synthesis, image-to-image translation) and 10 state-of-the-art GAN architectures to demonstrate the effectiveness of our scheme in verifying the ownership of generator. We also show qualitatively and quantitatively that our scheme is immune to popular and powerful removal attacks (*e.g.*, input transformation-based and reverse engineering-based attacks) and robust to other existing attacks.

Our main contributions are summarized as follows:

- We identify a fundamental limitation of black-box setting-based ownership verification schemes on generation tasks, *i.e.*, choosing deterministic inputs is not allowed for applications like unconditioned image synthesis.
- We reveal the unexploited potential of the well-trained discriminator for capturing the unique distribution learned by the paired generator. Based on this finding, we make the first attempt towards box-free verification scheme of GANs, which does not require specifying the input and does not rely on additional models.
- Extensive evaluations on two popular GAN applications and more than 10 GAN architectures demonstrate our proposed scheme to effectively verify the ownership. Through qualitative and quantitative analysis, we show that our proposed scheme is immune to popular removal attacks and robust to other existing attacks.

2. Related Work

2.1. Ownership Verification

Black-box Ownership Verification. The black-box ownership verification scheme assumes the defender can verify the ownership via querying the suspicious model with specific inputs [1, 64, 28, 54]. Towards this end, the owner either predetermines and trains the model to learn a set of abnormal input-output behaviours (model watermarks), or extract a set of boundary samples (*e.g.*, adversarial examples) that can identify the model (model fingerprint). Then, this hidden behavior can be extracted remotely by querying the model with these special inputs. Recently, Ong et al. [37] proposed the first work towards ownership verification of GANs. In their work, the IP information was embedded through backdooring the generator, after which trigger inputs (*i.e.*, carefully-crafted inputs with trigger noises, as shown in Fig. 1 top panel) will result in a visible watermark like a company's logo on generated outputs. However, for some GAN applications (*e.g.*, unconditional image synthesis), black-box is not enough since deterministic inputs are not allowed [62]. Moreover, the recent removal attacks have shown great threats to the survival of backdoor or adversarial input-based black-box verification paradigms.

Removal Attacks. The black-box methods rely on hidden functionalities in the model. Therefore, current attacks

Table 1: Comparison of our approach with prior works. In the "Img. Syn./Trans." column, ✘ denotes not applicable or not evaluated in the original paper.

Method	Year	Technique	Target model	Purpose	Box-free?	Img. Syn./ Trans.?	Ambiguity attack?	No external model?
Uchida et al. [52]	2017	Model Watermarking	DNN	Ownership Verification	✘	—	✘	✔
Adi et al. [1]	2018	Model Watermarking	Classifiers	Ownership Verification	✘	—	✘	✔
Zhang et al. [64]	2018	Model Watermarking	Classifiers	Ownership Verification	✘	—	✘	✔
Rouhani et al. [43]	2019	Model Watermarking	DNN	Ownership Verification	✘	—	✘	✔
Lukas et al. [34]	2019	Model Fingerprinting	Classifiers	Ownership Verification	✘	—	✘	✔
Le Merrer et al. [28]	2020	Model Watermarking	Classifier	Ownership Verification	✘	—	✘	✔
Zhao et al. [68]	2020	Model Fingerprinting	Classifiers	Ownership Verification	✘	—	✘	✔
Jia et al. [22]	2021	Model Watermarking	Classifier	Ownership Verification	✘	—	✔	✔
Cao et al. [8]	2021	Model Fingerprinting	Classifiers	Ownership Verification	✘	—	✘	✔
Bansal et al. [4]	2022	Model Watermarking	DNN	Ownership Verification	✘	—	✔	✔
Peng et al. [39]	2022	Model Fingerprinting	Classifiers	Ownership Verification	✘	—	✘	✔
Yang et al. [58]	2022	Model Fingerprinting	Classifiers	Ownership Verification	✘	—	✘	✘
Yu et al. [61]	2019	Fingerprint Extraction	GAN	Model Attribution	✔	✔ / ✘	✘	✘
Yu et al. [62]	2021	GAN Fingerprinting	GAN	Model Attribution	✔	✔ / ✔	✔	✘
Girish et al. [15]	2021	Fingerprint Extraction	GAN	Model Attribution	✔	✔ / ✔	✘	✘
Asnani et al. [3]	2021	Fingerprint Extraction	GAN	Model Attribution	✔	✔ / ✔	✘	✘
Yu et al. [63]	2022	GAN Fingerprinting	GAN	Model Attribution	✔	✔ / ✘	✔	✘
Guarnera et al. [17]	2022	Fingerprint Extraction	GAN	Model Attribution	✔	✔ / ✘	✘	✘
Ong et al. [37]	2021	GAN Watermarking	GAN	Ownership Verification	✘	✔ / ✔	✔	✔
Ours	2023	Distribution Capturing	GAN	Ownership Verification	✔	✔ / ✔	✔	✔

aim to remove or avoid activating this hidden functionality. A straightforward attack is to eliminate the hidden functionality through model modifications like pruning [35]. A more targeted attack is first to reverse-engineer the verification samples and then invalidate the watermarks/fingerprints through adversarial training [56, 48]. Observing the verification samples are less robust than normal inputs, another attack is to evade verification via preprocessing the input [19, 56]. This attack has become the recent trend since it is model architecture-careless and not limited to a specific watermarking technique. It is also intractable as the input transformations can be diverse and usually hard to consider in advance. In contrast, our proposed box-free ownership verification scheme is free from choosing verification samples thus totally immune to the intractable input transformation-based attacks and reverse engineering-based attacks.

Ambiguity Attack. Recent works [12, 13, 37] revealed the concept of ambiguity attack, where it is proved that unless an irreversible verification scheme is adopted, the adversary can forge his/her own vouch using exactly the same technique the owner adopted. Subsequently, when one claims ownership of the model, the adversary can also claim the ownership due to the existence of his/her own vouch. Finally, the ownership is in doubt. To ensure the owner is free from this concern, a feasible technique used for verifying ownership should necessarily be non-reproducible, even if the adversary has full control of the stolen model and acquires knowledge of the adopted ownership verification paradigm. A practical and robust ownership verification scheme should well survive these two threats.

2.2. Model Attribution

The model attribution was initially developed to combat DeepFakes, where researchers focus on attributing certain fake images to the specific types of GAN that generated them [61, 62, 63, 2, 17]. Generally, the attribution techniques aim to analyse the unique fingerprints carried by the GAN-generated images, or proactively watermark the output images through methods like steganography [57, 65, 66]. Due to the merit that these techniques can usually work in the box-free setting where only generated images are available, it has shown potential or even applications in ownership verification of GANs. However, the existing studies on GAN model attribution are not ready for this challenging task, since i) most works are limited to classifying the model architecture and/or datasets [6, 15, 59] rather than a specific GAN model; ii) nearly all attribution techniques require training a powerful external classifier, which is both time and resource consuming; and iii) most works' external classifiers can be trained with only real/fake images [61, 3, 11], and the steganography-based techniques could be easily reproduced by the attacker [57, 65], which denotes that the adversary can easily forge the verification vouch and perform ambiguity attacks or overwriting attacks. A comparison between our approach with prior studies is illustrated in Tab. 1.

3. The Proposed Verification Scheme

Motivated by the unstable training phenomenon of GAN training, the insight behind our proposal is to capture the model-specific distribution by exploiting the potential of the paired discriminator. Before stepping into the details of our proposed scheme, we first introduce the organization of this section. First, we formalize our threat model. Then, we describe the details of the essential loss terms used in our proposed scheme. Finally, we introduce the pipeline and training flow of our proposed approach.

3.1. Threat Model

In our threat model, two opposing parties are considered. A *model owner* (also the *defender*) who trains the source

model, and an *adversary* who has stolen the source model through illegal means and deploys the piracy model on Internet APIs publicly accessible for profit.

Defender's Goals and Capabilities. The goal of the defender is to identify the stolen models that are remotely deployed by the attacker. The defender (1) has white-box access to the source model, (2) can query the suspicious model but, (3) can not specify the query input (*i.e.*, box-free setting). This setting is more challenging than the black-box setting widely adopted by prior works [1, 37, 54], as specifying carefully-crafted inputs is not allowed.

Adversary's Goals and Capabilities. The adversary's goal is not to be verified as pirated while keeping the piracy a similar performance as the source model. For this purpose, the adversary may modify the model (model modification) or manipulate the inputs/outputs (sample transformation). Observe that our scheme does not suffer from the intractable input transformation-based attacks since our inputs are totally equivalent to those of normal users. Adopting such attacks would only impair the performance yet does not bring any benefits. The assumptions we make are standard and are also widely adopted by prior works [1, 64, 22, 37].

3.2. Compactness Loss

The discriminator witnessed the gradual development of the generator and has comparable parameters to the latter. It learns a hyperplane in its embedding space to distinguish between real and generated images. Intuitively, it potentially learned how to extract special representations in images generated by the paired generator. However, in real-world scenarios, suspicious images are usually generated by unknown and unseen GANs. Since embedding spaces of different generators are not aligned, it is difficult to harness the discriminator and find maximum margin hyperplanes to distinguish between different GAN instances.

Inspired by previous works on data description [49, 50], we propose to separate the data via optimizing a hypersphere instead of a hyperplane. Specifically, let $\mathcal{X} \subseteq \mathbb{R}^{c\times h\times w}$ be the data space, and $\phi : \mathbb{R}^{c\times h\times w} \to \mathbb{R}^d$ be the "encoder" part of the discriminator (explained later), which maps the data to a d-dimensional feature space. Our objective is to find the smallest hypersphere specified by a center $\boldsymbol{c} \in \mathbb{R}^d$ and radius $R > 0$ that encloses the majority of the data distribution of the paired generator in feature space. Therefore, our main objective is to minimize the "compactness" of representations [40, 44, 42]. Given the data $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ from the paired generator, our objective is defined as:

$$\min_{\mathcal{W},R} \quad R^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \max\{0, \|\phi(\boldsymbol{x}_i;\mathcal{W}) - \boldsymbol{c}\|^2 - R^2\} \quad (1)$$

where $\mathcal{W}$ indicates the weights of ϕ. This objective contains two terms. The first term aims to minimize the radius thus volume of the hypersphere, while the second term penalizes the points outside the hypersphere. $\nu \in (0, 1]$ is a hyperparameter that balances the importance of the two terms. We empirically set ν to 0.35 in the following experiments.

The benefits of our approach are as follows. First, our training is unsupervised, *i.e.*, only requires data generated by the paired GAN, analogous to the training of GAN itself and does not introduce additional annotation overhead. Second, prior works utilizing the compactness loss as their objective are shown vulnerable to "hypersphere collapse" where the hypersphere radius collapses to zero and the network converges to trivial solutions [51]. Fixing $\boldsymbol{c}$ as the mean of the network representations is shown to be helpful to avoid overfitting and hypersphere collapse [44, 10]. However, it is still difficult to define a proper hypersphere center $\boldsymbol{c}$ with an initial network whose parameters are random. In contrast, our well-trained discriminator network potentially provides us with a robust network representation thus a robust $\boldsymbol{c}$. Empirical results show that this strategy makes the convergence faster and more robust, the hypersphere collapse hardly exist.

3.3. Pearson Correlation Loss

Recall that we expect our discriminator to provide a robust initial center for training. This is feasible since we assume the discriminator extracts data representations well, thus network representations are useful. However, in reality, we observe that the discriminator usually converges to a constant and no helpful network representation is preserved. This is not surprising since the more the GAN is trained, the less possible to distinguish between samples of real and generated data. The optimal discriminator converges to $1/2$ and thus no extraction ability is preserved, as many existing literature has pointed out [16, 67].

To tackle this challenge, we propose to preserve the useful network representations of the discriminator via encouraging it to implicitly reconstruct the ground truth latent representation $\boldsymbol{z}$. Our key insight is that this additional task encourages the discriminator to fit the generator's latent distribution and prohibits its trivial convergences. However, we empirically found that the straightforward MSE loss makes the training extremely unstable. In our approach, we leverage the pearson correlation loss, which is inspired by the pearson Correlation Coefficient (PCC) [38], to measure the quality of the reconstructed latent representation. We define our term as:

$$\rho(\boldsymbol{z}, \hat{\boldsymbol{z}}) = \frac{\sum_{i=1}^{n_z}(\boldsymbol{z}_i - \mu_z)(\hat{\boldsymbol{z}}_i - \mu_{\hat{z}})/n_z}{\sigma_z \sigma_{\hat{z}}} \quad (2)$$

where $\hat{\boldsymbol{z}}$ is the reconstructed latent representation and $\boldsymbol{z}$ is the ground truth. n_z is the dimension of z. μ and σ indicates the mean value and standard deviation, respectively. $\rho(\cdot,\cdot) \in [-1, 1]$ measures the linear correlation between the variables. The higher ρ indicates better reconstruction performance.

PCC is "milder" than MSE [46]. The pearson correlation

loss ensures that the discriminator can implicitly reconstruct the latent representation from its corresponding generated image while avoiding making the training unstable. For training, obviously, this additional task can be easily cooperated with the BCE loss of the native GAN training. There is no annotation cost and no notable training overhead. We show in the supplementary materials that this additional loss does not bring any degradation to the original generation task.

Note that this loss is optional for GANs that are trained in supervised setting (*e.g.*, StarGAN). This is because these tasks usually require the discriminator to do an additional classification task, therefore the aforementioned convergence problem does not necessarily exist.

3.4. Training Pipeline

The whole training pipeline of our scheme is presented in Algorithm 1. We utilize the pearson correlation loss to preserve the network representations of the well-trained discriminator and harness these representations to train a robust hypersphere in the embedding space which captures the unique data distribution of the paired generator. We show the whole training pipeline as follows.

Step 1. Redefine the Training Objective: For unsupervised GANs, we introduce an additional task (*i.e.*, reconstructing the latent representation) to ensure the network representations of the well-trained discriminator are useful. The native GAN is composed of a generator mapping $\mathbb{R}^d \rightarrow \mathbb{R}^{c\times h\times w}$ and a discriminator mapping $\mathbb{R}^{c\times h\times w} \rightarrow \mathbb{R}$, which can be divided into an "encoder" $\phi : \mathbb{R}^{c\times h\times w} \rightarrow \mathbb{R}^d$ and a "classifier" that maps $\mathbb{R}^d \rightarrow \mathbb{R}$. As explained in Sec. 3.3, we wish the reconstructed latent representation $\hat{\boldsymbol{z}} = \phi(G(\boldsymbol{z}); \mathcal{W}) \in \mathbb{R}^d$ to be close to the ground truth representation $\boldsymbol{z} \in \mathbb{R}^d$. This is done via adding the pearson correlation loss to the adversarial objective of both G and D. The training objective of the generator is:

$$\min_G \mathcal{L}_G = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}))) - \lambda\rho(\boldsymbol{z}, \hat{\boldsymbol{z}})] \quad (3)$$

The training objective of the discriminator is:

$$\max_D \quad \mathcal{L}_D = \mathbb{E}_{\boldsymbol{x}\sim p_{\mathcal{D}}(\boldsymbol{x})}[\log D(\boldsymbol{x}) - \lambda\rho(\boldsymbol{z}, \hat{\boldsymbol{z}})] \quad (4)$$

where λ is a hyper-parameter that adjusts the pearson correlation strength. We empirically set $\lambda = 0.5$ in experiments.

Step 2. Train the GAN Models: We then initialize and train the GAN models G and D with some training data $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subseteq \mathcal{X}$ until we reach the maximum epochs. Except for the additional pearson correlation loss, the training of the GANs follow the settings in their original papers. After this, the generator is capable to tackle the task distribution and can be deployed.

Step 3. Empower the Discriminator: The final step is to utilize the discriminator's well-trained network representations and optimizes the network which forms a hypersphere to enclose the learned distribution. We first harness the generator to generate a few data points sampled from the distribution, forming a set of training data $\mathcal{D}' = \{\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_n\} \subseteq \mathcal{X}$, where $\boldsymbol{x}'_i = G(\boldsymbol{z})$, $\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})$. As explained in Sec. 3.2, we fix $\boldsymbol{c}$ as the mean of the network representations of the discriminator and train the network with data $\mathcal{D}'$ and the objective described in Eq. (1).

Algorithm 1: Training Pipeline

Input : Training data $\mathcal{D}$, Iteration K, Interval k, Learning rates τ and τ'.
Output : Generator G, Network ϕ with parameters $\mathcal{W}$, center $\boldsymbol{c}$, and radius R.

Initialize the generator G and discriminator D.
for $i \in \{1...K\}$ **do**
  Get a batch of $\mathcal{D}$
  Calculate adversarial loss of D
  Update $D \leftarrow D - \tau \cdot \nabla_D \mathcal{L}_D$
  Calculate adversarial loss of G
  Update $G \leftarrow G - \tau \cdot \nabla_G \mathcal{L}_G$
end
Form a dataset $\mathcal{D}'$ that consists of n samples generated by G
$\mathcal{D}' = \cup_{i=1}^n G(\boldsymbol{z})$
Initialize center $\boldsymbol{c}$, and radius R
step = 0
while *loss not converge* **do**
  step += 1
  Get a batch of $\mathcal{D}'$
  Calculate compactness loss:

$$\mathcal{L}_c = R^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \max\{0, \|\phi(\boldsymbol{x}_i; \mathcal{W}) - \boldsymbol{c}\|^2 - R^2\}$$

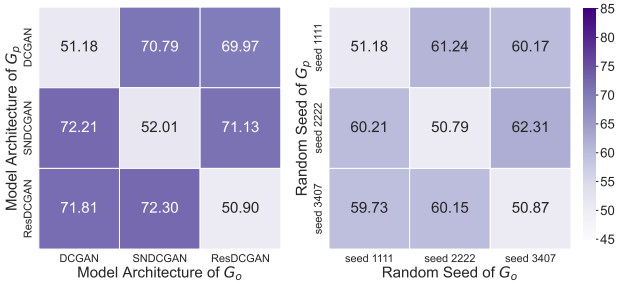
  Update $\mathcal{W} \leftarrow \mathcal{W} - \tau' \cdot \nabla_\phi \mathcal{L}_c$
  if *step* % k == *0* **then**
    Update R via line search
  end
end

We use stochastic gradient descent (SGD) to optimize the parameters $\mathcal{W}$ of the neural network with backpropagation. Noticeably, using one common SGD learning rate may be inefficient to optimize $\mathcal{W}$ and R simultaneously since they usually have different scales, as Ruff et al. [44] pointed out. Therefore, we optimize $\mathcal{W}$ and R alternatively as suggested. In detail, we first fix the radius R and train the network parameters $\mathcal{W}$ for every interval $k \in \mathbb{N}$ epochs. Then, after every k epoch, we solve for radius R via line search with the current network parameters. We train the network parameters $\mathcal{W}$ and radius R until convergence.

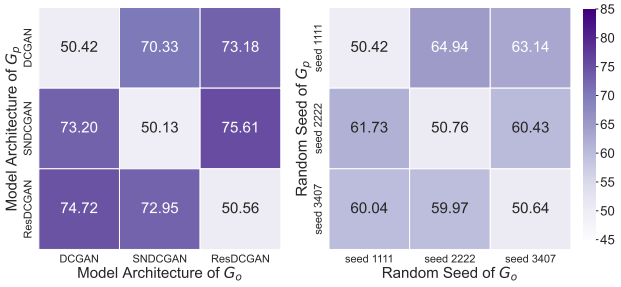
3.5. Ownership Verification

In predicting a given input, we can calculate a score which measures the representation proximity to the captured unique distribution using the network parameters. Given an input $\boldsymbol{x}$, the representation proximity score is calculated by the distance of the point to the center of the hypersphere:

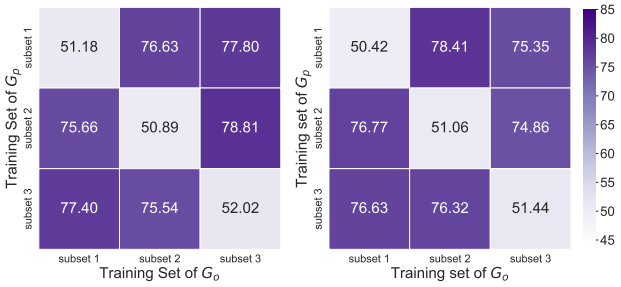
$$s(\boldsymbol{x}) = \|\phi(\boldsymbol{x}; \mathcal{W}) - \boldsymbol{c}\|^2 - R^2 \quad (5)$$



(a) AUC scores on LSUN dataset.



(b) AUC scores on CelebA dataset.



(c) AUC scores on different LSUN (left) and CelebA (right) subsets.

Figure 2: Effectiveness evaluation on image synthesis, in terms of three critical elements. The top panel indicates the evaluation on LSUN and the bottom panel denotes the evaluation on CelebA. We use D_o to distinguish source model G_o and suspect model G_p. The results on the diagonal represent that the two models are identical (i.e. G_p is copied from G_o).

The prediction is time and memory-efficient, since $s(\boldsymbol{x})$ is totally characterized by the network parameters $\mathcal{W}$, radius R, and the representation center $\boldsymbol{c}$. We do not require storing any other data for prediction and the prediction is done within a single forward pass.

Note that $s(\boldsymbol{x})$ has different scales in different cases. Therefore, we feed a batch of images produced by the suspicious GAN and use Area Under Curve (AUC) score to measure the performance. This avoids selecting a deterministic proximity score threshold. Through extreme results on AUC scores (see Sec. 4.2), we set the suspicious AUC score to 60%. That is, if a batch of suspicious images (batch size is empirically set to 500 in the following experiments) has an AUC $< 60\%$, we claim ownership of the suspicious model.

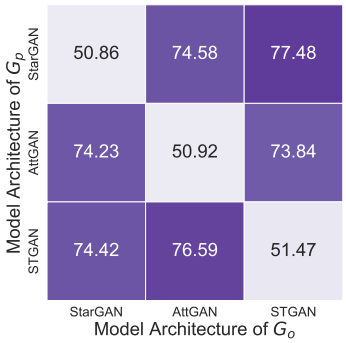


Figure 3: Effectiveness evaluation for image-to-image translation.

4. Experiments

In this section, we mainly explore the effectiveness, scalability and robustness of our proposed approach. Some additional experiments and ablation studies of our method refer to supplementary materials. For the below experiments, we report the average result of ten independent replicates.

4.1. Setup

Models. For image synthesis, we use DCGAN [41] and its two variants: SNDCGAN [36] and DCGAN established with residual block [20]. We also conducted experiments on SOTA architectures including ProGAN [24], StyleGAN [25], StyleGAN2 [26] and StyleGAN3 [27]. For image-to-image translation, we consider three popular GANs that edit face attributes: StarGAN [9], AttGAN [21], and STGAN [32].

Datasets. We evaluate our method on two popular datasets, LSUN [60] and CelebA [33]. We use LSUN bedroom for image synthesis and CelebA for both image synthesis and image-to-image translation.

Evaluation Metrics. For evaluating the effectiveness, we use the AUC score as mentioned earlier. We use structural similarity (SSIM) and Frechet Inception Distance (FID) [5] to measure image quality and similarity.

4.2. Effectiveness

We first show preliminarily that our method can effectively find out the piracy and would not mistakenly recognize homogeneous models (*i.e.*, the model independently trained in similar settings) as piracy as well, even only small factors (*e.g.*, initial seeds) are different. Fig. 2 and Fig. 3 present the experimental results on verifying the two popular GAN tasks (*i.e.*, entire images synthesis, image-to-image translation) measured with AUC score.

The closer the AUC score is to 100%, the greater the difference between two batches of synthesized outputs exposed. On the contrary, the closer the AUC score is to 50%, the more likely the two batches of images are from the same model because ϕ judges that there is no notable difference in the data distribution represented by the two batches.

Model Architectures. We consider the case where GANs with different architectures are trained on the same dataset and initial seeds. Experimental results in Fig. 2 and 3 illustrated that our method can identify the differences between GANs with different model architectures with an AUC score larger than 70% and determine a piracy GAN with the same elements with an AUC score close to 50%.

Training Datasets. We also investigate the effect of training a GAN on different datasets. We respectively split CelebA and LSUN into 3 disjoint subsets with 50k images each. This ensures the training data are from the similar distribution. Fig. 2 (c) shows that the AUC scores of two GANs trained on different datasets are all above 75%. The AUC score is higher than the other two elements, which indicates that the GAN training is sensitive to the training datasets.

Initialization seeds. We finally investigate the effects of initial random seeds in ownership verification. Fig. 2 shows that our method maintains AUC around 60%. Meanwhile, all the AUC scores of two identical models are less than 53%. We note that the initialization random seed is the smallest variable in training a homogeneous GAN. This extreme situation (*i.e.*, the source model and the piracy only differ in random seeds) is almost impossible to happen in reality. However, our scheme still achieves an AUC score of $\sim 60\%$, notably margins from the AUC score of two same models ($\sim 50\%$). This is why we set the suspicious AUC score to 60%.

	StyleGAN2	StyleGAN3	StyleGAN	ProGAN
StyleGAN2	**53.39**	82.57	84.93	85.66
StyleGAN3	83.12	**53.28**	83.52	86.34

Table 2: Evaluation on unknown and SOTA GANs. The AUC means the classifier is trained with {row}'s discriminator, measuring the images generated by the {column}'s G. We mark the results from the paired G and D in **bold font** and unpaired in normal font.

4.3. Scalability to SOTA Architectures

Table 2 shows the experimental results on the SOTA GAN Architectures. The results show that on SOTA architectures the performances are even better (AUC $> 80\%$). This is because the more complex the task is (*e.g.*, larger datasets, higher resolution, more sophisticated network architecture), the discriminator has larger parameters and the distributions learned by different GAN instances are more complex thus more different from each other. This shows that our method generalizes well on SOTA and potential future architectures.

4.4. Robustness

In this section, we mainly evaluate the robustness of our method in tackling the three existing attacks, *i.e.*, model pruning, output transformations, and ambiguity attack. The experiments here are done on DCGAN trained on CelebA with size 128×128. Specifically, we note that the discriminator plays a similar role to private keys and should be kept secret by the owner. Therefore, according to our threat model, attacks that require the presence of the discriminator (*e.g.*, fine-tuning) are not considered.

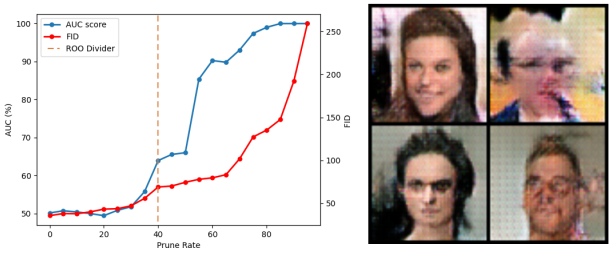


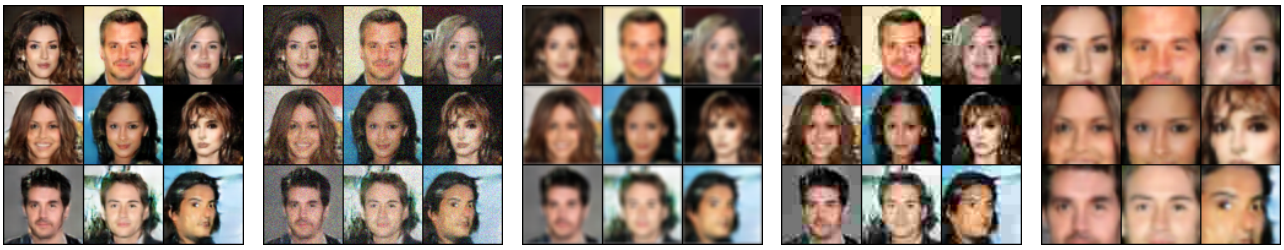
Figure 4: *left*: Performance in evaluating the robustness against model pruning measured by AUC score. The dotted line is the ROO Divider, the operation region where acceptable image quality holds. *right*: Visualization of synthesized images when the pruning rate is 40%.

Pruning. The model pruning aims to reset the the unimportant weights to 0 without introducing any performance degradation to the main task, like image synthesis in GANs. In experiments, we randomly reset the weights in GANs to explore whether our proposed method is sensitive to the model pruning. Experimental results in Fig. 4 (left) shows the AUC score for measuring the similarity of two models is less than 60% when the pruning rate reaches almost 40%. This indicates that our proposed method could verify two models in high confidence in this pruning rate settings. Here, the pruning rate less than 40% is a *region of operation* (ROO) as the quality of synthesized image is almost acceptable, where the FID score is larger than 69. Fig. 4 (right) visualizes the synthesized images when the pruning rate is 40%, which exhibits obvious damage. Thus, our method could survive the model pruning well.

Image transformation (magnitude)	CelebA		LSUN	
	AUC	SSIM	AUC	SSIM
noise (ϵ=0.05)	56.41	0.81	55.93	0.84
blur (ks=3, σ=2)	57.16	0.78	57.24	0.80
JPEG (factor=60)	54.90	0.73	56.77	0.69
crop (15%)	54.15	0.25	56.71	0.27

Table 3: Evaluation for the image transformation attacks measured by AUC scores and the similarity of images is measured by SSIM. The value of magnitude indicates the ROO, where the image has acceptable quality.

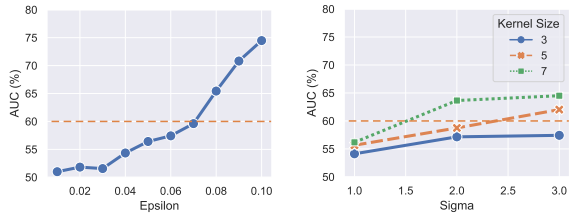
Image Transformations. Though input transformations are vain attempts, the adversary may conduct various output transformations to evade the verification. In experiments, we explore whether the sample transformation brings any degradation to our verification. Fig. 6 shows the performance of our method measured by AUC score in identifying suspicious models under four types of image transformations. The dotted line in Fig. 6 indicates a ROO where the synthesized images have no serious damage to human eyes. Fig. 5 visualizes the images under the ROO setting and Tab. 3 shows the corresponding magnitudes for the four types of image transformation. Experimental results illustrated



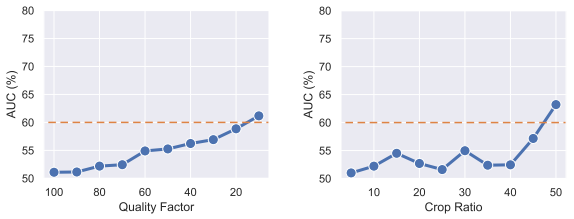
(a) Original (b) Noise (c) Blur (d) JPEG (e) Crop

Figure 5: Visualization of synthesized images where the magnitude of image transformation is in the ROO (the value refer to Tab. 3).

that our method could identify the ownership effectively in the ROO of image transformations. We attribute this to the distribution-capturing property of our proposed method, which potentially learned some high-level features that are robust against these low-level transformations.



(a) Noise (b) Blur



(c) JPEG (d) Crop

Figure 6: Four image transformation attacks under different intensities.

Ambiguity Attacks. In a real scenario, the adversary launches an ambiguity attack by obtaining a classifier that performs as well as the owner's one after replicating the model illegally. We simulate this attack by training a classifier using exactly the same technique described in Sec. 3 but without the help of the discriminator. Tab. 4 shows that random initialization will significantly degrade the performance of the learned hypersphere. The reason, as we have mentioned earlier, is that the discriminator provides a strong and unique center $\boldsymbol{c}$, compared to naive random initialization. Thus, the attacker failed to obtain a well-performed classifier even if he obtains the philosophy of our proposed method.

5. Conclusion and Discussion

In this paper, we propose a novel ownership verification scheme for GANs, which working in box-free manner, universal to popular GAN tasks, and resisting the powerful ambiguity attack well. Inspired by the power of discriminator in witnessing the development of generators in synthesizing images gradually, we empower the discriminator to capture the unique GAN training which is important for ownership verification. Evaluation experiment results show that our method is highly effective, general and robust.

Training strategy	AUC (same) ↓	AUC (different) ↑
w/o The Discriminator (piracy)	51.91	57.21
w/ The Discriminator (Ours)	**50.18**	**75.63**

Table 4: Performance in resisting ambiguity attacks. The column *same* indicates the verification of two same GAN models, while the column *different* denotes the verification for different GAN models.

Limitations and Discussions. Though there seem to be no trivial adaptive attacks, our method relies on the empowered discriminator to capture the unique distribution learned by the generator. Therefore, if the discriminator is disclosed, the adversary may train a same classifier to confuse the verification, or leverage the discriminator to perform adversarial attacks. However, the discriminator is conventionally considered useless and will not be used after training. In the popular MLaaS scenario, there is also no reason for the model owner to open it to a third party. The adversary could not even access the discriminator through APIs. In our proposed scheme, the discriminator plays a similar role to private keys and should be kept secret by the owner.

There are many spaces worth discovering for future works. For example, one may extend our insight of distribution capturing to other generative models like diffusion models. It is also interesting to explore effective methods to evade our box-free verification, which could be our future work.

6. Acknowledgment

This research was supported in part by the National Key Research and Development Program of China under No.2021YFB3100700, the National Natural Science Foundation of China (NSFC) under Grants No. 62202340, the Open Foundation of Henan Key Laboratory of Cyberspace Situation Awareness under No. HNTS2022004, Wuhan Knowledge Innovation Program under No. 2022010801020127, the Fundamental Research Funds for the Central Universities under No. 2042023kf0121, the Natural Science Foundation of Hubei Province under No. 2021CFB089.

References

[1] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1615–1631, 2018. 2, 3, 4

[2] Michael Albright and Scott McCloskey. Source generator attribution via inversion. In *CVPR Workshops*, volume 8, 2019. 3

[3] Vishal Asnani, Xi Yin, Tal Hassner, and Xiaoming Liu. Reverse engineering of generative models: Inferring model hyperparameters from generated images. *arXiv preprint arXiv:2106.07873*, 2021. 3

[4] Arpit Bansal, Ping-yeh Chiang, Michael J Curry, Rajiv Jain, Curtis Wigington, Varun Manjunatha, John P Dickerson, and Tom Goldstein. Certified neural network watermarks with randomized smoothing. In *International Conference on Machine Learning*, pages 1450–1465. PMLR, 2022. 3

[5] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019. 6

[6] Tu Bui, Ning Yu, and John Collomosse. Repmix: Representation mixing for robust attribution of synthesized images. In *European Conference on Computer Vision*, pages 146–163. Springer, 2022. 3

[7] Zhipeng Cai, Zuobin Xiong, Honghui Xu, Peng Wang, Wei Li, and Yi Pan. Generative adversarial networks: A survey toward private and secure applications. *ACM Computing Surveys (CSUR)*, 54(6):1–38, 2021. 1

[8] Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 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*, pages 14–25, 2021. 3

[9] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018. 6

[10] Penny Chong, Lukas Ruff, Marius Kloft, and Alexander Binder. Simple and effective prevention of mode collapse in deep one-class classification. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2020. 4

[11] Yuzhen Ding, Nupur Thakur, and Baoxin Li. Does a GAN leave distinct model-specific fingerprints? In *BMVC*, page 22. BMVA Press, 2021. 3

[12] Lixin Fan, Kam Woh Ng, and Chee Seng Chan. Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks. *Advances in neural information processing systems*, 32, 2019. 3

[13] Lixin Fan, Kam Woh Ng, Chee Seng Chan, and Qiang Yang. Deepipr: Deep neural network ownership verification with passports. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 44(10):6122–6139, 2022. 3

[14] Guanhao Gan, Yiming Li, Dongxian Wu, and Shu-Tao Xia. Towards robust model watermark via reducing parametric vulnerability. In *ICCV*, 2023. 1

[15] Sharath Girish, Saksham Suri, Sai Saketh Rambhatla, and Abhinav Shrivastava. Towards discovery and attribution of open-world gan generated images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14094–14103, October 2021. 3

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1, 4

[17] Luca Guarnera, Oliver Giudice, Matthias Nießner, and Sebastiano Battiato. On the exploitation of deepfake model recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 61–70, 2022. 3

[18] Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. In *ICLR*, 2023. 2

[19] Shangwei Guo, Tianwei Zhang, Han Qiu, Tao Xiang, and Yang Liu. Fine-tuning is not enough: A simple yet effective watermark removal attack for dnn models. pages 3635–3641, 08 2021. doi: 10.24963/ijcai.2021/500. 2, 3

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[21] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE transactions on image processing*, 28(11):5464–5478, 2019. 6

[22] Hengrui Jia, Christopher A Choquette-Choo, Varun Chandrasekaran, and Nicolas Papernot. Entangled watermarks as a defense against model extraction. In *USENIX Security Symposium*, pages 1937–1954, 2021. 2, 3, 4

[23] Felix Juefei-Xu, Run Wang, Yihao Huang, Qing Guo, Lei Ma, and Yang Liu. Countering malicious deepfakes: Survey, battleground, and horizon. *International Journal of Computer Vision*, pages 1–57, 2022. 1

[24] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 6

[25] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 6

[26] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 6

[27] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 6

[28] Erwan Le Merrer, Patrick Perez, and Gilles Trédan. Adversarial frontier stitching for remote neural network watermarking. *Neural Computing and Applications*, 32(13):9233–9244, 2020. 2, 3

[29] Yiming Li, Yang Bai, Yong Jiang, Yong Yang, Shu-Tao Xia, and Bo Li. Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection. In *NeurIPS*, 2022. 2

[30] Yiming Li, Mingyan Zhu, Xue Yang, Yong Jiang, Tao Wei, and Shu-Tao Xia. Black-box dataset ownership verification via backdoor watermarking. *IEEE Transactions on Information Forensics and Security*, 2023. 2

[31] Ji Lin, Richard Zhang, Frieder Ganz, Song Han, and Jun-Yan Zhu. Anycost gans for interactive image synthesis and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14986–14996, 2021. 1

[32] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. Stgan: A unified selective transfer network for arbitrary image attribute editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3673–3682, 2019. 6

[33] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 6

[34] Nils Lukas, Yuxuan Zhang, and Florian Kerschbaum. Deep neural network fingerprinting by conferrable adversarial examples. *arXiv preprint arXiv:1912.00888*, 2019. 2, 3

[35] Nils Lukas, Edward Jiang, Xinda Li, and Florian Kerschbaum. Sok: How robust is image classification deep neural network watermarking? In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 787–804. IEEE, 2022. 1, 3

[36] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 6

[37] Ding Sheng Ong, Chee Seng Chan, Kam Woh Ng, Lixin Fan, and Qiang Yang. Protecting intellectual property of generative adversarial networks from ambiguity attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3630–3639, 2021. 1, 2, 3, 4

[38] Karl Pearson. Vii. mathematical contributions to the theory of evolution.—iii. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, (187):253–318, 1896. 2, 4

[39] Zirui Peng, Shaofeng Li, Guoxing Chen, Cheng Zhang, Haojin Zhu, and Minhui Xue. Fingerprinting deep neural networks globally via universal adversarial perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13430–13439, 2022. 2, 3

[40] Pramuditha Perera and Vishal M Patel. Learning deep features for one-class classification. *IEEE Transactions on Image Processing*, 28(11):5450–5463, 2019. 4

[41] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 6

[42] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. Panda: Adapting pretrained features for anomaly detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2806–2814, 2021. 4

[43] Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. Deepsigns: an end-to-end watermarking framework for protecting the ownership of deep neural networks. In *ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2019. 3

[44] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018. 4, 5

[45] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3626–3636, 2022. 1

[46] Jianlin Su. O-gan: extremely concise approach for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1903.01931*, 2019. 4

[47] Yuchen Sun, Tianpeng Liu, Panhe Hu, Qing Liao, Shouling Ji, Nenghai Yu, Deke Guo, and Li Liu. Deep intellectual property: A survey. *arXiv preprint arXiv:2304.14613*, 2023. 1

[48] Guanhong Tao, Guangyu Shen, Yingqi Liu, Shengwei An, Qiuling Xu, Shiqing Ma, Pan Li, and Xiangyu Zhang. Better trigger inversion optimization in backdoor scanning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13368–13378, 2022. 3

[49] David MJ Tax and Robert PW Duin. Support vector domain description. *Pattern recognition letters*, 20(11-13):1191–1199, 1999. 4

[50] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54:45–66, 2004. 4

[51] Jialin Tian, Xing Xu, Fumin Shen, Yang Yang, and Heng Tao Shen. Tvt: Three-way vision transformer through multi-modal hypersphere learning for zero-shot sketch-based image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2370–2378, 2022. 4

[52] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on international conference on multimedia retrieval*, pages 269–277, 2017. 3

[53] Run Wang, Haoxuan Li, Lingzhou Mu, Jixing Ren, Shangwei Guo, Li Liu, Liming Fang, Jing Chen, and Lina Wang. Rethinking the vulnerability of dnn watermarking: Are watermarks robust against naturalness-aware perturbations? In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1808–1818, 2022. 2

[54] Run Wang, Jixing Ren, Boheng Li, Tianyi She, Wenhui Zhang, Liming Fang, Jing Chen, and Lina Wang. Free fine-tuning: A plug-and-play watermarking scheme for deep neural networks. In *Proceedings of the 31th ACM International Conference on Multimedia*, 2023. 2, 4

[55] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11379–11388, 2022. 1

[56] Zhenting Wang, Kai Mei, Hailun Ding, Juan Zhai, and Shiqing Ma. Rethinking the reverse-engineering of trojan triggers. *Advances in Neural Information Processing Systems*, 35:9738–9753, 2022. 2, 3

[57] Hanzhou Wu, Gen Liu, Yuwei Yao, and Xinpeng Zhang. Watermarking neural networks with watermarked images. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(7):2591–2601, 2020. 3

[58] Kang Yang, Run Wang, and Lina Wang. Metafinger: Fingerprinting the deep neural networks with metatraining. 3

[59] Tianyun Yang, Ziyao Huang, Juan Cao, Lei Li, and Xirong Li. Deepfake network architecture attribution. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI 2022)*, 2022. 3

[60] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 6

[61] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7556–7566, 2019. 3

[62] Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14448–14457, 2021. 2, 3

[63] Ning Yu, Vladislav Skripniuk, Dingfan Chen, Larry Davis, and Mario Fritz. Responsible disclosure of generative models using scalable fingerprinting. In *International Conference on Learning Representations (ICLR)*, 2022. 3

[64] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 159–172, 2018. 2, 3, 4

[65] Jie Zhang, Dongdong Chen, Jing Liao, Han Fang, Weiming Zhang, Wenbo Zhou, Hao Cui, and Nenghai Yu. Model watermarking for image processing networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12805–12812, 2020. 3

[66] Jie Zhang, Dongdong Chen, Jing Liao, Weiming Zhang, Huamin Feng, Gang Hua, and Nenghai Yu. Deep model intellectual property protection via deep watermarking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4005–4020, 2021. 3

[67] Zhaoyu Zhang, Mengyan Li, and Jun Yu. On the convergence and mode collapse of gan. In *SIGGRAPH Asia 2018 Technical Briefs*, pages 1–4. 2018. 4

[68] Jingjing Zhao, Qingyue Hu, Gaoyang Liu, Xiaoqiang Ma, Fei Chen, and Mohammad Mehedi Hassan. Afa: Adversarial fingerprinting authentication for deep neural networks. *Computer Communications*, 150:488–497, 2020. 3