Artificial Fingerprinting for Generative Models: Rooting Deepfake Attribution in Training Data

Ning Yu[1,2]* Vladislav Skripniuk[3]* Sahar Abdelnabi[3] Mario Fritz[3]
[1]University of Maryland [2]Max Planck Institute for Informatics
[3]CISPA Helmholtz Center for Information Security
{ningyu,vladislav}@mpi-inf.mpg.de {sahar.abdelnabi,fritz}@cispa.saarland

Abstract

Photorealistic image generation has reached a new level of quality due to the breakthroughs of generative adversarial networks (GANs). Yet, the dark side of such deepfakes, the malicious use of generated media, raises concerns about visual misinformation. While existing research work on deepfake detection demonstrates high accuracy, it is subject to advances in generation techniques and adversarial iterations on detection countermeasure techniques. Thus, we seek a ***proactive*** *and sustainable solution on deepfake detection, that is agnostic to the evolution of generative models, by introducing* ***artificial fingerprints*** *into the models.*

Our approach is simple and effective. We first embed artificial fingerprints into training data, then validate a surprising discovery on the ***transferability*** *of such fingerprints from training data to generative models, which in turn appears in the generated deepfakes. Experiments show that our fingerprinting solution (1) holds for a variety of cutting-edge generative models, (2) leads to a negligible side effect on generation quality, (3) stays robust against image-level and model-level perturbations, (4) stays hard to be detected by adversaries, and (5) converts deepfake detection and attribution into trivial tasks and outperforms the recent state-of-the-art baselines. Our solution closes the responsibility loop between publishing pre-trained generative model inventions and their possible misuses, which makes it independent of the current arms race.*

1. Introduction

In the past years, photorealistic image generation has been rapidly evolving, benefiting from the invention of generative adversarial networks (GANs) [16] and its successive breakthroughs [39, 17, 35, 5, 25, 26, 27, 50, 51]. Given the level of realism and diversity that generative models can achieve today, detecting generated media, well known as *deepfakes*, attributing their sources, and tracing their legal responsibilities become infeasible to human beings.

Moreover, the misuse of deepfakes has been permeating to each corner of social media, ranging from misinformation of political campaigns [24] to fake journalism [44, 40]. This motivates tremendous research efforts on deepfake detection [53] and source attribution [34, 49, 46]. These techniques aim to counter the widespread of malicious applications of deepfakes by automatically identifying generated visual contents and tracking their sources. Most of them rely on low-level visual patterns in GAN-generated images [34, 49, 46, 19, 57] or frequency mismatch [13, 56, 14]. However, these techniques are unable to sustainably and robustly prevent deepfake misuse in the long run; as generative models evolve, they learn to better match the true distribution causing fewer artifacts [53]. Besides, detection countermeasures are also continuously evolving [12, 6, 53].

Motivated by this, we tackle deepfake detection and attribution through a different lens, and propose a ***proactive*** and sustainable solution for detection, which is simple and effective. In specific, we aim to introduce ***artificial fingerprints*** into generative models that enable identification and tracing. Figure 1 depicts our pipeline; we first embed artificial fingerprints into the training data using image steganography [4, 42]. The generative model is then trained with its original protocol without modification. This makes our solution agnostic and plug-and-play for arbitrary models. We then show a surprising discovery on the ***transferability*** of such fingerprints from training data to the model: the same fingerprint information that was encoded in the training data can be decoded from all generated images.

We achieve deepfake detection by classifying images with matched fingerprints in our database as fake and images with random detected fingerprints as real. We also achieve deepfake attribution when we allocate different fingerprints for different generative models. Our solution thus closes the responsibility loop between generative model inventions and their possible misuses. It prevents the misuse of published pre-trained generative models by enabling in-

*Equal contribution.

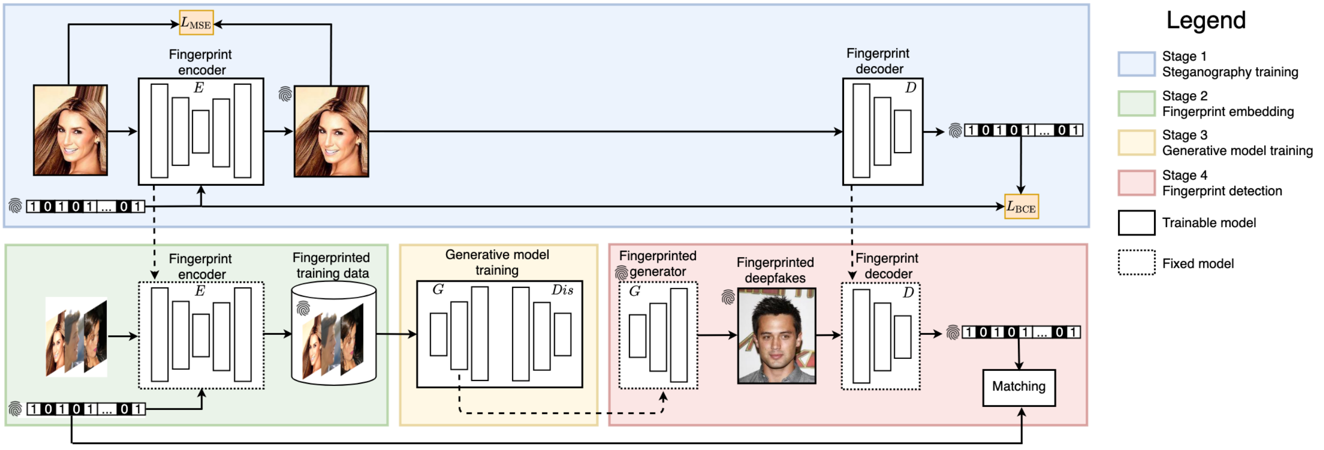


Figure 1: Our solution pipeline consists of four stages. We first train an image steganography encoder and decoder. Then we use the encoder to embed artificial fingerprints into the training data. After that, we train a generative model with its original protocol. Finally, we decode the fingerprints from the generated deepfakes.

ventors to proactively and responsibly embed artificial fingerprints into the models.

We summarize our contributions as follow:

(1) We synergize the two previously uncorrelated domains, image steganography and GANs, and propose the first ***proactive*** and sustainable solution for the third emerging domain, deepfake detection and attribution.

(2) This is the first study to demonstrate the ***transferability*** of artificial fingerprints from training data to generative models and then to all the generated deepfakes. Our discovery is non-trivial: only deep-learning-based fingerprinting techniques [4, 42] are transferable to generative models, while conventional steganography and watermarking techniques [2, 1] are not. See Section 5.2 for comparisons.

(3) We empirically validate several beneficial properties of our solution. ***Universality*** (Section 5.2): it holds for a variety of cutting-edge generative models [25, 26, 27, 5, 36]. ***Fidelity*** (Section 5.3): it has a negligible side effect on generation quality. ***Robustness*** (Section 5.4): it stays robust against many perturbations. ***Secrecy*** (Section 5.5): the artificial fingerprints are hard to be detected by adversaries. ***Anti-deepfake*** (Section 5.6 and 5.7): it converts deepfake detection and attribution into trivial tasks and outperforms the state-of-the-art baselines [49, 46].

2. Related Work

Generative adversarial networks (GANs). GANs [16] was first proposed as a workaround to model the intractable real data distribution. The iterative improvements push the generation realism to brand-new levels [39, 17, 35, 5, 25, 26, 27]. Successes have also spread to many other vision tasks (e.g. [37, 29, 23, 59, 60, 36, 48]). In Section 5, we focus on three categories of cutting-edge generative models: unconditional (ProGAN [25], StyleGAN [26], and StyleGAN2 [27]), class-conditional (BigGAN [5]), and image-conditional (image-to-image translation) (CUT [36]).

Image steganography and watermarking. Image steganography and watermarking hide information into carrier images [15]. Previous techniques rely on Fourier transform [11, 7], JPEG compression [2, 1], or least significant bits modification [38, 21, 22]. Recent works replace hand-crafted hiding procedures with neural network encoding [4, 18, 45, 58, 55, 42, 33]. We leverage recent deep-learning-based steganography methods [4, 42] to embed artificial fingerprints into training data, and validate their transferability to generative models. This is non-trivial because only deep-learning-based fingerprints are transferable to generative models, while conventional ones [2, 1] are not (Section 5.2). Besides, the stealthiness achieved by steganography allows preserving the original generation quality (Section 5.3) and fingerprint secrecy (Section 5.5).

Our fingerprinting is conceptually and functionally orthogonal to all of them. Instead of encoding information into pixels of individual images, our solution encodes information into generator parameters such that all the generated images are entangled with that information. Compared to the pipeline of a generator followed by a watermarking module, our solution introduces zero generation overheads, and obstructs adversarial model surgery that targets to detach watermarking from image generation.

Network watermarking. Different from image watermarking, network watermarking targets to hide information into model parameters without affecting its original performances, similar in spirit to our goal. There are two categories of them: black-box trigger-set-based solutions [3, 54], and white-box feature-based solutions [43, 9, 41]. The former ones embed watermarks through a trigger set of input and decodes watermarks according to the input-output behavior of the model. The latter ones directly embed watermarks in the model parameter space with transformation matrices. It is worth noting that our solution renders conceptual and technical distinctions from network watermarking. In terms of concepts, the previous works target to only discriminative models (e.g., classification), while a solution for generative models is urgently lacking. In terms of techniques, to adapt to generator watermarking, we tune our solution to *indirectly* transfers fingerprints from training data to model parameters. This is because (1) unconditional generative models do not allow deterministic input so that a trigger set is not applicable, and (2) transformations in the parameter space are not agnostic to model configurations so that they are neither scalable nor sustainable along with the evolution of generative models.

Deepfake detection and attribution. Images generated by GAN models bear unique patterns. [34] shows that generative models leave unique noise residuals to generated samples, which allows deepfake detection. [49] moves one step further, using a neural network classifier to attribute different images to their sources. [46] also train a classifier and improve the generalization across different generation techniques. [56, 13, 12] point out that the high-frequency pattern mismatch can be used for deepfake detection, so can the texture feature mismatch [32]. However, these cues are not sustainable due to the advancement of detection countermeasures. For example, spectral regularization [12] is proposed to narrow down the frequency mismatch and results in a significant detection deterioration. Also, detectors [46] are vulnerable to adversarial evasion attacks [6].

In contrast to the previous passive approaches, we propose a novel ***proactive*** solution for model fingerprinting and, thus, for deepfake detection. We differentiate between our term ***artificial fingerprints*** which refers to the information we deliberately and proactively embed into the model, and the term GAN fingerprints [49] which refers to the inherent cues and artifacts of different GAN models. Our work is also distinct from a follow-up proactive technique [52]. They focus on fingerprinting scalability and efficiency while we focus more fundamentally on its transferability and universality.

3. Problem Statement

Model inventors have concerns about releasing their generative models because of potential deepfake misuses. Therefore, we design solution from the model inventors' side (e.g., OpenAI) that introduces traceable artificial fingerprints in generative models. It enables deepfake detection and attribution by decoding the fingerprints from the generated images and matching them to the known fingerprints given to different models. This equips model inventors with a means for a proactive and responsible disclosure when publishing their pre-trained models. This distinguishes our model fingerprinting solution from watermarking the generated images: we aim to defend against the misuse of generative models rather than single deepfakes.

In practice, the training is done by model inventors. The fingerprinting encoder and decoder, and the unique fingerprints given to different models, are privately maintained by the model inventor. Once a deepfake misuse happens, the inventor is able to verify if this is generated by one of their models. If so, they can further attribute to the model user. As a result, they can claim responsible disclosure with a countermeasure against potential misuse when they publish their models.

4. Artificial Fingerprints

The goal of image attribution is to learn a mapping $D_0(\mathbf{x}) \mapsto y$ that traces the source $y \in \mathbb{Y} = \{\text{real}, \text{G}_1, \dots, \text{G}_\text{N}\}$ of an image $\mathbf{x}$. If the domain $\mathbb{Y}$ is limited, predefined, and known to us, this is a closed-world scenario and the attribution can be simply formulated as a multi-label classification problem, each label corresponding to one source, as conducted in [49]. However, $\mathbb{Y}$ can be unlimited, undefined, continuously evolving, and agnostic to us. This open-world scenario is intractable using discriminative learning. To generalize our solution to being agnostic to the selection of generative models, we formulate the attribution as a regression mapping $D(\mathbf{x}) \mapsto \mathbf{w}$, where $\mathbf{w} \in \{0,1\}^n$ is the source identity space and n is the dimension. We propose a pipeline to root the attribution down to the training dataset $\tilde{\mathbf{x}} \in \tilde{\mathbb{X}}$ and close the loop of the regression D. We describe the pipeline (Figure 1) below:

Steganography training. The source identity is represented by the artificial fingerprints $\mathbf{w}$. We use a steganography system [4, 42] to learn an encoder $E(\tilde{\mathbf{x}}, \mathbf{w}) \mapsto \tilde{\mathbf{x}}_\mathbf{w}$ that embeds an arbitrary fingerprint $\mathbf{w}$ (randomly sampled during training) into an arbitrary image $\tilde{\mathbf{x}}$. We couple E with a decoder $D(\tilde{\mathbf{x}}_\mathbf{w}) \mapsto \mathbf{w}$ to detect the fingerprint information from the image. E and D are formulated as convolutional neural networks with the following training losses:

$$\min_{E,D} \mathbb{E}_{\tilde{\mathbf{x}} \sim \tilde{\mathbb{X}}, \mathbf{w} \sim \{0,1\}^n} L_\text{BCE}(\tilde{\mathbf{x}}, \mathbf{w}; E, D) + \lambda L_\text{MSE}(\tilde{\mathbf{x}}, \mathbf{w}; E) \quad (1)$$

$$\text{L}_\text{BCE}(\tilde{\mathbf{x}}, \mathbf{w}; E, D) = \frac{1}{n} \sum_{k=1}^{n} \big(\mathbf{w}_k \log \hat{\mathbf{w}}_k + (1 - \mathbf{w}_k) \log(1 - \hat{\mathbf{w}}_k)\big) \quad (2)$$

$$L_\text{MSE}(\tilde{\mathbf{x}}, \mathbf{w}; E) = ||E(\tilde{\mathbf{x}}, \mathbf{w}) - \tilde{\mathbf{x}}||_2^2 \quad (3)$$

$$\hat{\mathbf{w}} = D\big(E(\tilde{\mathbf{x}}, \mathbf{w})\big) \quad (4)$$

where $\mathbf{w}_k$ and $\hat{\mathbf{w}}_k$ are the k^th bit of the input fingerprint and detected fingerprint separately; and λ is a hyper-parameter to balance the two objective terms. The binary cross-entropy term L_BCE guides the decoder to decode the fingerprint embedded by the encoder. The mean squared error term L_MSE penalizes any deviation of the stego image

$E(\tilde{\mathbf{x}}, \mathbf{w})$ from the original image $\tilde{\mathbf{x}}$. The architectures of E and D are depicted in the supplementary material.

Artificial fingerprint embedding. In this stage, we use the well trained E and D networks. We allocate each training dataset $\tilde{\mathbb{X}}$ a unique fingerprint $\mathbf{w}$. We apply the trained E to each training image $\tilde{\mathbf{x}}$ and collect a fingerprinted training dataset $\tilde{\mathbb{X}}_{\mathbf{w}} = \{E(\tilde{\mathbf{x}}, \mathbf{w}) | \tilde{\mathbf{x}} \in \tilde{\mathbb{X}}\}$.

Generative model training. In order to have a solution that is agnostic to the evolution of generative models, we do not intervene with their training. It makes our solution plug-and-play for arbitrary generation tasks without touching their implementations, and introduces zero overhead to model training. We simply replace $\tilde{\mathbb{X}}$ with $\tilde{\mathbb{X}}_{\mathbf{w}}$ to train the generative model in its original protocol.

Artificial fingerprint decoding. We hypothesize the *transferability* of our artificial fingerprints from training data to generative models: a well-trained generator $G_{\mathbf{w}}(\mathbf{z}) \mapsto \mathbf{x}_{\mathbf{w}}$ contains, in all generated images, the same fingerprint information $\mathbf{w}$ (as embedded in the training data $\tilde{\mathbf{x}}_{\mathbf{w}}$). We justify this hypothesis in Section 5.2. As a result, the artificial fingerprint can be recovered from a generated image $\mathbf{x}_{\mathbf{w}}$ using the decoder D: $D(\mathbf{x}_{\mathbf{w}}) \equiv \mathbf{w}$. Based on this transferability, we can formulate deepfake attribution as fingerprint matching using our decoder D.

Artificial fingerprint matching. To support robustness to post-generation modifications that could be applied to the generated images, we relax the matching of the decoded artificial fingerprints to a soft matching. We perform a null hypothesis test given the number of matching bits k between the decoded fingerprint $\tilde{\mathbf{w}}$ and the fingerprint $\mathbf{w}$ used in generative model training. The null hypothesis H_0 is getting this number of successes (i.e. matching bits) by chance. Under the null hypothesis, the probability of matching bits (random variable X) follows a binomial distribution: the number of trials n is the number of bits in the fingerprint sequence, and k is the number of successes where each bit has a 0.5 probability of success. We can then measure the p-value of the hypothesis test by computing the probability of getting k or higher matching bits under the null hypothesis:

$$Pr(X > k | H_0) = \sum_{i=k}^{n} \binom{n}{i} 0.5^n \tag{5}$$

The fingerprint is verified, $\tilde{\mathbf{w}} \sim \mathbf{w}$, if the null hypothesis results in a very low probability (p-value). Usually, when the p-value is smaller than 0.05, we reject the null hypothesis and regard $1 - p$ as the verification confidence.

5. Experiments

5.1. Setup

Generative models. As a proactive solution, it should be agnostic to genetative models. Without losing representativeness, we focus on three generation applications with their state-of-the-art models. For unconditional generation: ProGAN [25], StyleGAN [26], and StyleGAN2 [27]; for class-conditional generation: BigGAN [5]; for image-conditional generation, i.e., image-to-image translation: CUT [36]. Each model is trained from scratch with the official implementation.

Datasets. Each generation application benchmarks its own datasets. For unconditional generation, we train/test on 150k/50k CelebA [31] at 128×128 resolution, 50k/50k LSUN *Bedroom* [47] at 128×128 resolution, and the most challenging one, 50k/50k LSUN *Cat* [47] at its original 256×256 resolution. For class-conditional generation, we experiment on the entire CIFAR-10 dataset [28] with the original training/testing split at the original 32×32 resolution. For image-conditional generation, we experiment on the entire *Horse*→*Zebra* dataset [59] and *Cat*→*Dog* [10] dataset with the original training/testing split at the original 256×256 resolution. We only need to fingerprint images from the target domains.

5.2. Transferability

The transferability means that the artificial fingerprints that are embedded in the training data also appear consistently in all the generated data. This is a non-trivial hypothesis in Section 4 and needs to be justified by the fingerprint detection accuracy.

Evaluation. Fingerprints are represented as binary vectors $\mathbf{w} \in \{0, 1\}^n$. We use bitwise accuracy to evaluate the detection accuracy. We set $n = 100$ as suggested in [42]. We also report p-value for the confidence of detection.

Baselines. For comparison, we implement an intuitive baseline method. Instead of embedding fingerprints into training data, we enforce fingerprint generation jointly with model training. That is, we train on clean data, and enforce generated images to not only approximate real training images but also contain a specific fingerprint. Mathematically,

$$\begin{aligned} \min_{G,D} \max_{Dis} \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tilde{\mathbf{x}} \sim \tilde{\mathbb{X}}} L_{\text{adv}}(\mathbf{z}, \tilde{\mathbf{x}}; G, Dis) + \\ \eta \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \mathbf{w} \sim \{0,1\}^n} L_{\text{BCE}}(\mathbf{z}, \mathbf{w}; G, D) \end{aligned} \tag{6}$$

where G and Dis are the original generator and discriminator in the GAN framework, L_{adv} is the original GAN objective, and L_{BCE} is adapted from Eq. 2 where we replace $\hat{\mathbf{w}} = D(E(\tilde{\mathbf{x}}, \mathbf{w}))$ with $\hat{\mathbf{w}} = D(G(\mathbf{z}))$. η is set to 1.0 as a hyper-parameter to balance the two objective terms.

We also compare the deep steganography technique used in our solution ([42]) to two well-established, non-deep steganography techniques [2, 1] that alter the frequency coefficients of JPEG compression.

Results. We report the fingerprint detection performance in Table 1 fourth and fifth columns. We observe:

(1) The "Data" row shows the detection accuracy on real testing images for sanity checks: it reaches the 100% satu-

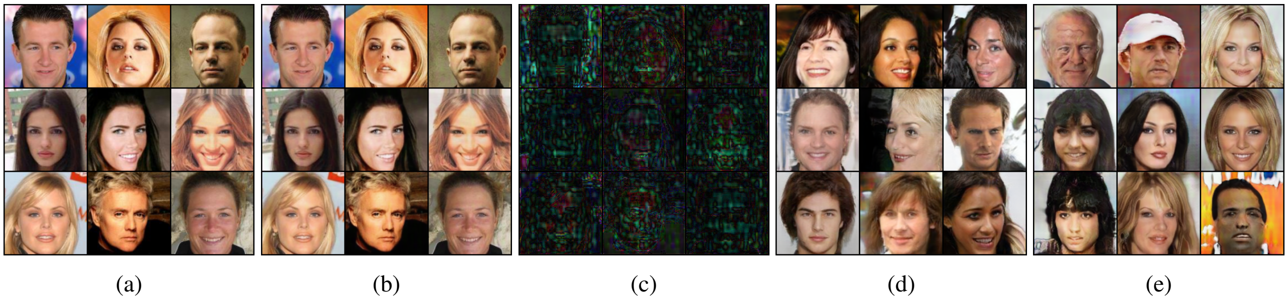


Figure 2: CelebA samples at 128×128 for Table 1 last two columns. (a) Original real training samples. (b) Fingerprinted real training samples. (c) The difference between (a) and (b), 10× magnified for easier visualization. (d) Samples from the non-fingerprinted ProGAN. (e) Samples from the fingerprinted ProGAN. See more samples on the other datasets in the supplementary material.

	Fgpt		Bit		Orig	Fgpt
Dataset	tech	Model	acc ⇑	p-value	FID	FID ⇓
CelebA	Eq. 6	ProGAN	0.93	$< 10^{-19}$	14.09	60.28
	[2]	StyleGAN2	0.51	0.46	6.41	6.93
	[1]	StyleGAN2	0.53	0.31	6.41	6.82
	[42]	Data	1.00	-	-	1.15
	[42]	ProGAN	0.98	$< 10^{-26}$	14.09	14.38
	[42]	StyleGAN	0.99	$< 10^{-28}$	8.98	9.72
	[42]	StyleGAN2	0.99	$< 10^{-28}$	6.41	6.23
LSUN *Bedroom*	[42]	ProGAN	0.93	$< 10^{-19}$	29.16	32.58
	[42]	StyleGAN	0.98	$< 10^{-26}$	24.95	25.71
	[42]	StyleGAN2	0.99	$< 10^{-28}$	13.92	14.71
LSUN *Cat*	[42]	ProGAN	0.98	$< 10^{-26}$	45.22	48.97
	[42]	StyleGAN	0.99	$< 10^{-28}$	33.45	34.01
	[42]	StyleGAN2	0.99	$< 10^{-28}$	31.01	32.60
CIFAR-10	[42]	BigGAN	0.99	$< 10^{-28}$	6.25	6.80
Horse→Zebra	[42]	CUT	0.99	$< 10^{-28}$	22.98	23.43
Cat→Dog	[42]	CUT	0.99	$< 10^{-28}$	55.78	56.09

Table 1: Artificial fingerprint detection in bitwise accuracy (⇑ indicates higher is better) and generation quality in FID (⇓ indicates lower is better). The "Data" row corresponds to real testing images for a sanity check. The "Orig FID" column corresponds to the original (non-fingerprinted) models for references. The first three rows are the baselines.

rated accuracy, indicating the effectiveness of the steganography technique by its nature.

(2) Our artificial fingerprints can be almost perfectly detected from generated images over a variety of applications, generative models, and datasets. The accuracy is ≥ 0.98 except for ProGAN on LSUN *Bedroom*, but its 0.93 accuracy and 10^{-19} p-value are far sufficient to verify the presence of fingerprints. Our hypothesis on the *transferability* from training data to generative models (i.e. generated data) is therefore justified. As a result, artificial fingerprints are qualified for deepfake detection and attribution.

(3) The *universality* of fingerprint transferability over varying tasks and models validates our solution is agnostic to generative model techniques.

(4) The *generalization* of fingerprint autoencoder has been validated across datasets. We have trained only one 128×128 E and D on CelebA, and have cross-validated them on LSUN *Bedroom*. We have trained only one 256×256 E and D on LSUN *Cat*, and have cross-validated them on *Horse→Zebra* and *Cat→Dog*.

(5) The baseline of joint fingerprinting and generation training (first row) is also moderately effective in terms of fingerprint detection, but we show in Section 5.3 it leads to strong deterioration of generation quality.

(6) Conventional steganography methods [2, 1] (second and third rows) do not transfer hidden information into models, indicated by the random guess performance during decoding. We attribute this to the discrepancy between deep generation techniques and shallow steganography techniques. We reason that generative models leverage deep discriminators to approximate common image patterns including low-level fingerprints. Only comparably deep fingerprinting techniques, e.g. [42], are compatible to hide and transfer fingerprints to the models, while hand-crafted image processing is not effective. Therefore, the transferability of our fingerprinting is non-trivial.

5.3. Fidelity

The fidelity of generated images is as critical as the transferability. Fingerprinting should have a negligible side effect on the functionality of generative models. This preserves the original generation quality and avoids the adversary's suspect of the presence of fingerprints. The steganography should enable this, which we validate empirically.

Evaluation. We use Fréchet Inception Distance (FID) [20] to evaluate the generation quality; the lower, the more realistic. We measure FID between a set of 50k generated images and a set of 50k real non-fingerprinted images, in order to evaluate the quality of the generated set.

Results. We compare the generation quality of original and fingerprinted generative models in Table 1 sixth and

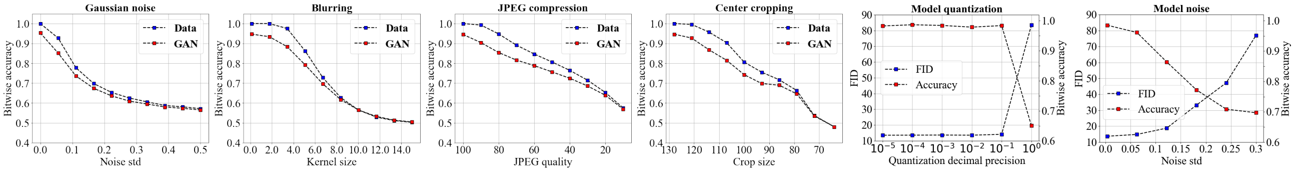
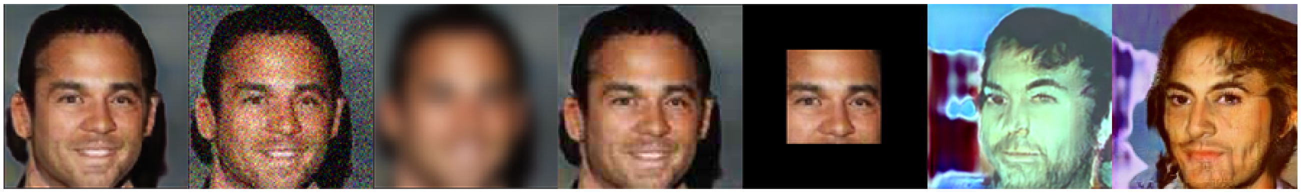


Figure 3: Red plots show the artificial fingerprint detection in bitwise accuracy w.r.t. the amount of perturbations over ProGAN trained on CelebA. In the left four plots (robustness against *image perturbations*), blue dots represent detection accuracy on the fingerprinted real training images, which serve as the upper bound references for the red dots. See the supplementary material for additional results of ProGAN trained on LSUN *Bedroom*. In the right two plots (robustness against *model perturbations*), blue dots represent the FID of generated images from the perturbed models.



(a) Original 0.99 bit acc

(b) Gaussian noise Std 0.1 0.77 bit acc

(c) Blurring Kernel size 5 0.75 bit acc

(d) JPEG Quality 35% 0.75 bit acc

(e) Cropping Crop size 64 0.80 bit acc

(f) Quantize Precision 10^0 0.64 bit acc

(g) Model noise Std 0.16 0.77 bit acc

Figure 4: Perturbed image samples from the fingerprinted ProGAN and the corresponding fingerprint detection accuracy. The detection still performs robustly (bitwise accuracy ≥ 0.75) even when the image quality heavily deteriorates.

seventh columns. We observe:

(1) The "Data" rows are for sanity checks: embedding fingerprints into real images does not substantially deteriorate image quality: FID $\leq$ 1.15 is in an excellent realism range. This validates the secrecy of steganography and lays a valid foundation for high-quality model training.

(2) For a variety of settings, the performance of the fingerprinted generative models tightly sticks to the original limits of their non-fingerprinted baselines. The heaviest deterioration is as small as $+3.75$ FID happening for ProGAN on LSUN *Cat*. In practice, the generated fingerprints are imperceptibly hidden in the generated images and can only be perceived under $10\times$ magnification. See Figure 2 and the supplementary material for demonstrations. Therefore, the fidelity of fingerprinted models is justified and it qualifies our solution for deepfake detection and attribution.

(3) The baseline of joint fingerprinting and generation training (first row) deteriorates generation quality remarkably. This indicates model fingerprinting is a non-trivial task: direct fingerprint reconstruction distracts adversarial training. In contrast, our solution leverages image steganography and fingerprint transferability, sidesteps this issue, and leads to better performance.

5.4. Robustness

Deepfake media and generative models may undergo post-processing or perturbations during broadcasts. We validate the robustness of our fingerprint detection given a variety of image and model perturbations, and investigate the corresponding working ranges.

Perturbations. We evaluate the robustness against four types of *image perturbation*: additive Gaussian noise, blurring with Gaussian kernel, JPEG compression, center cropping. We also evaluate the robustness against two types of *model perturbations*: model weight quantization and adding Gaussian noise to model weights. For quantization, we compress each model weight given a decimal precision. We vary the amount of perturbations, apply each to the generated images or to the model directly, and detect the fingerprint using the pre-trained decoder.

Results. We evaluate the artificial fingerprint detection over 50k images from a fingerprinted ProGAN. We plot the bitwise accuracy w.r.t. the amount of perturbations in Figure 3 (see the supplementary material for additional results of ProGAN trained on LSUN *Bedroom*). We observe:

(1) For all the image perturbations, fingerprint detection accuracy drops monotonously as we increase the amount of perturbation, while for small perturbations accuracy drops rather slowly. We consider accepting accuracy $\geq 75\%$ as a threshold (p-value $= 2.8 \times 10^{-7}$). This results in the working range w.r.t. each perturbation: Gaussian noise standard deviation $\sim [0.0, 0.05]$, Gaussian blur kernel size $\sim [0, 5]$, JPEG compression quality $\sim [50, 100]$, center cropping size $\sim [86, 128]$, quantization decimal precision $\leq 10^{-1}$, and model noise standard deviation $\sim [0.0, 0.18]$, which are reasonably wide ranges in practice.

(2) For image perturbations (the left four subplots) outside the above working ranges, the reference upper bounds drop even faster and the margins to the testing curves shrink quickly, indicating that the detection deterioration is irrelevant to model training but rather relevant to the heavy quality deterioration of training images.

(3) For model perturbations (the right two subplots) outside the above working ranges, image quality deteriorates faster than fingerprint accuracy: even before the accuracy gets lower than 75%, FID has already increased by $> 500\%$.

(4) As a result of (2) and (3), before fingerprint detection degenerates close to random guess ($\sim 50\%$ accuracy), image quality has been heavily deteriorated by strong perturbations (Figure 4), which indicates that our fingerprints are more robust than image functionality itself in the case of these studied perturbations.

5.5. Secrecy

The presence of a fingerprint embedded in a generative model should not be easily detected by a third party, otherwise, it would be potentially manipulated.

Attack. In order to quantify the secrecy, we formulate it as a binary classification problem: the presence of a fingerprint in an image. We follow the attack protocol in [58] to perform the Artificial Training Sets (ATS) attack [30]. We aim to classify between fingerprinted and non-fingerprinted test images, but we have no supervision. The intuition is to train another steganography encoder to expand the test set and establish an artificial setting with known labels which enables supervised training. The original test class space is now a subspace of the artificial training class space and is separable by the training task.

The attack is as follows: We independently train another steganography encoder. We consider the original test images as negative training samples. Then, we apply the encoder twice to the test set to obtain extra images fingerprinted 2 times (corresponding to originally non-fingerprinted images) or 3 times (corresponding to originally fingerprinted images), which are regarded as positive training samples. Then we train an SVM classifier [8] using such positive and negative samples, in order to separate between images fingerprinted 0-1 time (original set), and the ones fingerprinted 2-3 times (artificial training set).

During testing, we first apply the encoder once to the test images so that the originally non-fingerprinted images are now fingerprinted 1 time (belonging to 0-1 class), and the originally fingerprinted images are now fingerprinted 2 times (belonging to 2-3 class). Then we can use the classifier to separate them and propagate the predictions back to the original images. We evaluate the attack on a set of 250/250 fingerprinted/non-fingerprinted deepfake images.

We investigate in three scenarios: (I) the weak attacker cannot access our E and D, so he trains his shadow autoencoder and applies with arbitrary fingerprint codes; (II) the medium attacker can access our well-trained E and D but applies with arbitrary fingerprint codes; (III) the strong attacker can access our E and D and be partially aware of our fingerprint codes, so applies with the fingerprint codes only 10 bits different from ours.

Attacker	Access to E & D	Fgpt	Cls acc $\Downarrow$
Weak	No	Arbitrary	0.503
Medium	Yes	Arbitrary	0.503
Strong	Yes	10-bit diff	0.504

Table 2: Validation on the secrecy of fingerprints. The last column shows the binary classification accuracy of the presence of fingerprints, the smaller the more secret.

Results. In Table 2 all the attackers fail with the binary classification accuracy on the presence of fingerprint close to 0.5 (random guess). It indicates our fingerprinting is secret enough from being detected by adversaries regardless of their access to our encoder and decoder.

5.6. Deepfake Detection

We now discuss how to use our fingerprinting solution for deepfake detection and attribution. Unlike existing methods that detect intrinsic differences between the real and deepfake classes [49, 56, 46, 12], we, *standing for model inventors*, propose a proactive solution by embedding artificial fingerprints into generative models and consequently into the generated images. Then we convert the problem to verifying if one decoded fingerprint is in our fingerprint regulation database or not. Even with a non-perfect detection accuracy, we can still use our solution based on the null hypothesis test in Section 4. We consider deepfake verification given $\geq 75\%$ (p-value $= 2.8 \times 10^{-7}$) bit matching. This is feasible based on two assumptions: (1) The decoded fingerprint of a real image is random; and (2) the fingerprint capacity is large enough such that the random fingerprint from a real image is unlikely to collide with a regulated fingerprint in the database. The second condition is trivial to satisfy, considering we sample fingerprints $\mathbf{w} \in \{0,1\}^n$ and $n = 100$. 2^{100} is a large enough capacity. Then we validate the first assumption by the deepfake detection experiments below.

Baselines. We compare to two recent state-of-the-art CNN-based deepfake detectors [49, 46] as baselines. [49] is trained on 40k real images and 40k generated images equally from four generative models with distinct fingerprints. We consider the *open-world* scenario where disjoint generative models are used in training and testing, to challenge the classifier's generalization. For [46] we use the officially released model because they claim improved generalization across different generation techniques.

Results. We compare our solution to the two baselines on a variety of generation applications, models, and

Dataset	Model	Detector	Detection acc ⇑	Attribution acc ⇑
CelebA	ProGAN	[49]	0.508	0.235
		[46]	0.924	N/A
		Ours	1.000	1.000
	StyleGAN	[49]	0.497	0.168
		[46]	0.906	N/A
		Ours	1.000	1.000
	StyleGAN2	[49]	0.500	0.267
		[46]	0.895	N/A
		Ours	1.000	1.000
LSUN *Bedroom*	ProGAN	[49]	0.493	0.597
		[46]	0.952	N/A
		Ours	1.000	1.000
	StyleGAN	[49]	0.499	0.366
		[46]	0.956	N/A
		Ours	1.000	1.000
	StyleGAN2	[49]	0.491	0.267
		[46]	0.930	N/A
		Ours	1.000	1.000
LSUN *Cat*	ProGAN	[46]	0.951	N/A
		Ours	1.000	1.000
	StyleGAN	[46]	0.923	N/A
		Ours	1.000	1.000
	StyleGAN2	[46]	0.905	N/A
		Ours	1.000	1.000
CIFAR-10	BigGAN	[46]	0.815	N/A
		Ours	1.000	1.000
Horse→Zebra	CUT	[46]	0.836	N/A
		Ours	1.000	1.000
Cat→Dog	CUT	[46]	0.902	N/A
		Ours	1.000	1.000

Table 3: Deepfake detection and attribution accuracy (⇑ indicates higher is better). [46] is not applicable to the multi-source attribution scenarios in the last column.

datasets. We test on 4k real images and 4k generated images equally from four generative models with distinct fingerprints. We report deepfake detection accuracy in Table 3 fourth column. We observe:

(1) Our solution performs perfectly (100% accuracy) for all the cases, turning open-world deepfake detection into a trivial fingerprinting detection and matching problem.

(2) [49] deteriorates to random guess (∼ 50% accuracy) because of the curse of domain gap between training and testing models. In contrast, our solution benefits from being agnostic to generative models. It depends only on the presence of fingerprints rather than the discriminative cues that are overfitted during training.

(3) Our solution outperforms [46] with clear margins. In particular, [46] degenerates when model techniques evolve to be more powerful (from ProGAN to StyleGAN2), or condition on some input guidance. On the contrary, our proactive solution synergizes with this evolution with high fingerprint detection accuracy, and therefore, with perfect deepfake detection accuracy. In general, our work offers higher sustainability in the long run by proactively enforcing a margin between real and generated images.

5.7. Deepfake Attribution

The goal of attribution is to trace the model source that generated a deepfake. It upgrades the binary classification in detection to multi-class classification. Our artificial fingerprint solution can be easily extended for attribution and enable us, standing for model inventors, to attribute responsibility to our users when misuses occur.

Baseline. [46] is not applicable to multi-source attribution. We only compare to [49] in the *open-world* scenario, i.e., the training and testing sets of generative models are not fully overlapping. Given 40k generated images equally from four generative models with distinct fingerprints, we use [49] to train four one-vs-all-the-others binary classifiers. During testing, all four classifiers are applied to an image. We assign the image to the class with the highest confidence if not all the classifiers reject that image. Otherwise, it is assigned to the unknown label.

Results. We compare our solution to [49] on CelebA and LSUN *Bedroom*. We test on 4k/4k generated images equally from four model sources that are in/out of the training set of [49]. We report deepfake attribution accuracy in Table 3 last column. We obtain the same conclusions as those of deepfake detection in Section 5.6. The open-world attribution deteriorates for [49] while our fingerprinting solution maintains the perfect (100%) accuracy.

6. Conclusion

Detecting deepfakes is a complex problem due to the rapid development of generative models and adversarial countermeasure techniques. For the sake of sustainability, we investigate a proactive solution on the model inventors' side to make deepfake detection agnostic to generative models. We root deepfake detection into training data, and demonstrate the transferability of artificial fingerprints from training data to a variety of generative models. Our empirical study shows several beneficial properties of fingerprints, as well as perfect detection and attribution accuracy. Our solution opens up possibilities for inventors' responsibility disclosure by allocating each model a unique fingerprint.

Acknowledgement

This work was partially supported by the DARPA SemaFor (HR001119S0085) program. Ning Yu is partially supported by Twitch Research Fellowship. Vladislav Skripniuk is partially supported by IMPRS scholarship from Max Planck Institute. We thank David Jacobs, Matthias Zwicker, Abhinav Shrivastava, Yaser Yacoob, and Apratim Bhattacharyya for constructive discussion and advice.

References

[1] steghide, http://steghide.sourceforge.net. 2, 4, 5

[2] outguess, http://www.outguess.org/. 2, 4, 5

[3] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *USENIX*, 2018. 2

[4] Shumeet Baluja. Hiding images in plain sight: Deep steganography. In *NeurIPS*, 2017. 1, 2, 3

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2018. 1, 2, 4

[6] Nicholas Carlini and Hany Farid. Evading deepfake-image detectors with white-and black-box attacks. In *CVPR Workshops*, 2020. 1, 3

[7] Francois Cayre, Caroline Fontaine, and Teddy Furon. Watermarking security: theory and practice. In *TSP*, 2005. 2

[8] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. In *TIST*, 2011. 7

[9] Huili Chen, Bita Darvish Rouhani, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models. In *ICMR*, 2019. 2

[10] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, 2020. 4

[11] Ingemar Cox, Matthew Miller, Jeffrey Bloom, and Chris Honsinger. *Digital watermarking*. Springer, 2002. 2

[12] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *CVPR*, 2020. 1, 3, 7

[13] Ricard Durall, Margret Keuper, Franz-Josef Pfreundt, and Janis Keuper. Unmasking deepfakes with simple features. *arXiv*, 2019. 1, 3

[14] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *ICML*, 2020. 1

[15] Jessica Fridrich. *Steganography in digital media: principles, algorithms, and applications*. Cambridge University Press, 2009. 2

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 1, 2

[17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeurIPS*, 2017. 1, 2

[18] Jamie Hayes and George Danezis. Generating steganographic images via adversarial training. In *NeurIPS*, 2017. 2

[19] Yang He, Ning Yu, Margret Keuper, and Mario Fritz. Beyond the spectrum: Detecting deepfakes via re-synthesis. In *IJCAI*, 2021. 1

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 5

[21] Vojtěch Holub and Jessica Fridrich. Designing steganographic distortion using directional filters. In *WIFS*, 2012. 2

[22] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. Universal distortion function for steganography in an arbitrary domain. In *EURASIP JIS*, 2014. 2

[23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2

[24] Charlotte Jee. An indian politician is using deepfake technology to win new voters. 2020. 1

[25] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 1, 2, 4

[26] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1, 2, 4

[27] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 1, 2, 4

[28] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009. 4

[29] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 2

[30] Daniel Lerch-Hostalot and David Megías. Unsupervised steganalysis based on artificial training sets. In *EAAI*, 2016. 7

[31] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 4

[32] Zhengzhe Liu, Xiaojuan Qi, Jiaya Jia, and Philip Torr. Global texture enhancement for fake face detection in the wild. In *CoRR*, 2020. 3

[33] Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar. Distortion agnostic deep watermarking. In *CVPR*, 2020. 2

[34] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do gans leave artificial fingerprints? In *MIPR*, 2019. 1, 3

[35] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 1, 2

[36] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *ECCV*, 2020. 2, 4

[37] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 2

[38] Tomáš Pevnỳ, Tomáš Filler, and Patrick Bas. Using high-dimensional image models to perform highly undetectable steganography. In *IWIH*, 2010. 2

[39] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 1, 2

[40] Dan Robitzski. Someone used deepfake tech to invent a fake journalist. 2020. 1

[41] Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. Deepsigns: an end-to-end watermarking framework for protecting the ownership of deep neural networks. In *ASPLOS*, 2019. 2

[42] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *CVPR*, 2020. 1, 2, 3, 4, 5

[43] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *ICMR*, 2017. 2

[44] James Vincent. An online propaganda campaign used ai-generated headshots to create fake journalists. 2020. 1

[45] Vedran Vukotić, Vivien Chappelier, and Teddy Furon. Are deep neural networks good for blind image watermarking? In *WIFS*, 2018. 2

[46] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *CVPR*, 2020. 1, 2, 3, 7, 8

[47] 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*, 2015. 4

[48] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018. 2

[49] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *ICCV*, 2019. 1, 2, 3, 7, 8

[50] Ning Yu, Ke Li, Peng Zhou, Jitendra Malik, Larry Davis, and Mario Fritz. Inclusive gan: Improving data and minority coverage in generative models. In *ECCV*, 2020. 1

[51] Ning Yu, Guilin Liu, Aysegul Dundar, Andrew Tao, Bryan Catanzaro, Larry Davis, and Mario Fritz. Dual contrastive loss and attention for gans. In *ICCV*, 2021. 1

[52] Ning Yu, Vladislav Skripniuk, Dingfan Chen, Larry Davis, and Mario Fritz. Responsible disclosure of generative models using scalable fingerprinting. *arXiv*, 2020. 3

[53] Baiwu Zhang, Jin Peng Zhou, Ilia Shumailov, and Nicolas Papernot. Not my deepfake: Towards plausible deniability for machine-generated media. *arXiv*, 2020. 1

[54] 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 *CCS Asia*, 2018. 2

[55] Ru Zhang, Shiqi Dong, and Jianyi Liu. Invisible steganography via generative adversarial networks. In *Multimedia Tools and Applications*, 2019. 2

[56] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. In *WIFS*, 2019. 1, 3, 7

[57] Peng Zhou, Ning Yu, Zuxuan Wu, Larry S Davis, Abhinav Shrivastava, and Ser-Nam Lim. Deep video inpainting detection. *arXiv*, 2021. 1

[58] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *ECCV*, 2018. 2, 7

[59] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2, 4

[60] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *NeurIPS*, 2017. 2