ARTIFICIAL GAN FINGERPRINTS: ROOTING DEEP-FAKE ATTRIBUTION IN TRAINING DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Photorealistic image generation is progressing rapidly and has reached a new level of quality, thanks to the invention and breakthroughs of generative adversarial networks (GANs). Yet the dark side of such *deepfakes*, the malicious use of generated media, never stops raising concerns of visual misinformation. Existing research works on deepfake detection demonstrate impressive accuracy, while it is accompanied by adversarial iterations on detection countermeasure techniques. In order to lead this arms race to the end, we investigate a fundamental solution on deepfake detection, agnostic to the evolution of GANs in order to enable a responsible disclosure or regulation of such double-edged techniques. We propose to embed **artificial fingerprints** into GAN training data, and show a surprising discovery on the **transferability** of such fingerprints from training data to GAN models, which in turn enables reliable detection and attribution of deepfakes. Our empirical study shows that our fingerprinting technique (1) holds for different state-of-the-art GAN configurations, (2) turns more effective along with the development of GAN techniques, (3) has a negligible side effect on the generation quality, and (4) stays robust against image-level and model-level perturbations. When we allocate each GAN publisher a unique artificial fingerprint, the margins between real data and deepfakes, and the margins among different deepfake sources are fundamentally guaranteed. As a result, we are able to evidence accurate deepfake detection/attribution using our fingerprint decoder, which makes this solution stand out from the current arms race.

1 INTRODUCTION

In the past years, photorealistic image generation has been evolving rapidly, benefiting from the invention of generative adversarial networks (GANs) (Goodfellow et al., 2014) and its successive breakthroughs (Radford et al., 2016; Gulrajani et al., 2017; Miyato et al., 2018; Brock et al., 2018; Karras et al., 2018; 2019; 2020). Given the level of realism and diversity that GANs can achieve today, detecting generated media, well known as *deepfakes*, attributing their sources, and tracing their legal responsibilities become infeasible to human beings. Along with the advent of deep learning open-source development infrastructures, e.g., PyTorch (Paszke et al., 2016) and Tensorflow (Abadi et al., 2016), and with the prevalence of deep learning computing platforms on the clouds, e.g., Amazon AWS ML, Microsoft Azure ML, deepfake techniques turn increasingly democratized and the misuse of deepfakes have been permeating to each corner of social media, ranging from misinformation of political campaigns (Jee, 2020) to fake journalism (Vincent, 2020; Robitzski, 2020).

This motivates tremendous research efforts on deepfake detection (Zhang et al., 2020) and source attribution (Marra et al., 2019; Yu et al., 2019b; Wang et al., 2020). By automatically identifying and flagging generated visual contents and tracking their sources, these techniques aim to counter the widespread of malicious applications of deepfakes. Most of them rely on low-level visual patterns in GAN-generated images (Marra et al., 2019; Yu et al., 2019b; Wang et al., 2020) or frequency mismatch (Durall et al., 2019; Zhang et al., 2019b; Frank et al., 2020). However, these techniques are unlikely to stop deepfake misuse for good. On the contrary, they equally boost the adversarial iterations on anti-detection techniques higher quality generations and are vulnerable to adversarial evasion attacks (Carlini & Farid, 2020). For example, Durall et al. (2020) propose to conceal high-frequency cues of generated images, resulting in significant performance deterioration of state-of-the-art deepfake detectors. Ascribed to the steady improvement of GANs, this category

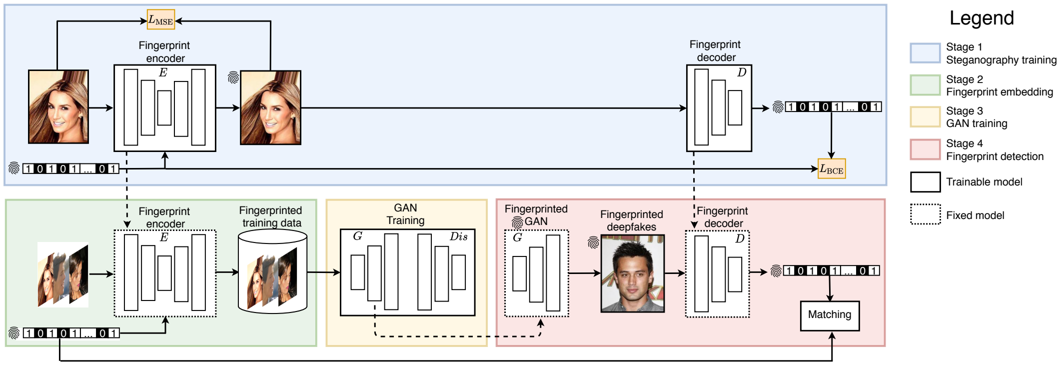


Figure 1: Our solution pipeline consists of four stages as captioned in the legend and described in Section 3. We first train an image steganography encoder and decoder and then use the encoder to embed artificial fingerprints into the GAN training data. We then train a GAN model in the default way. Finally, we decode the fingerprints from the deepfakes; we justify that fingerprints can be transferred from real data to the deepfakes.

of discriminative detectors can barely sustain. Therefore, there is a clear need to seek a fundamental solution that leads to the end of this arms race.

Motivated by this spirit, we tackle deepfake detection and attribution through a different lens, and propose a sustainable solution for detection. In specific, we reason to embed **artificial fingerprints** into the data before any GAN technique is instantiated on it. We present the first study to justify a surprising discovery on the **transferability** of such fingerprints from training data to black-box GAN models, agnostic to GAN configurations. That is, we show the existence of the same fingerprint information across all the GAN-generated images as it was embedded into the GAN training data. We approach this by applying steganography (Baluja, 2017; Tancik et al., 2020) to GAN training data, transferring the stega (artificial fingerprints) to black-box GAN models through training, and finally validating the fingerprints in the generated images using the steganography decoder. Figure 1 depicts our pipeline; we achieve deepfake detection by classifying images with matched fingerprints in our database as fake and images with random detected fingerprints as real (because real images in fact do not contain artificial fingerprints). We also achieve deepfake attribution when we allocate different fingerprints for different GAN training.

We summarize our contributions as follow:

(1) We propose a sustainable solution for deepfake detection and attribution, which is independent of the current arms race between GANs development and deepfake detection.

(2) This is the first study to justify the **transferability** of artificial fingerprints from GAN training data to GAN models, which in turn justifies its feasibility for deepfake detection and attribution.

(3) Our empirical study validates several beneficial properties of such fingerprints. **Generality**: It holds for several state-of-the-art GAN configurations. **Synergy** from GANs: It turns more effective along with the development of GAN techniques. **Fidelity**: It has a negligible side effect on generation quality. **Robustness**: It stays robust against image-level and model-level perturbations.

(4) We demonstrate the advantageous performance of our artificial fingerprints solution on multiple datasets and GAN subjects over a state-of-the-art detector (Yu et al., 2019b). This in turn enables responsible disclosure of GANs by the publishers or even regulation of the GAN disclosure process by allocating each publisher a unique fingerprint.

2 Related Work

Generative adversarial networks (GANs). GANs (Goodfellow et al., 2014) was first proposed as a workaround to model the intractable real data distribution. The iterative improvements push the generation realism to brand-new levels (Radford et al., 2016; Gulrajani et al., 2017; Miyato et al., 2018; Brock et al., 2018; Karras et al., 2018; 2019; 2020). Successes have also been spread to many other vision tasks, including but not limited to texture synthesis (Yu et al., 2019a), semantic image synthesis (Park et al., 2019), super resolution (Ledig et al., 2017), image attribute editing (Choi et al., 2018), image to image translation (Isola et al., 2017; Zhu et al., 2017a;b), inpainting (Yu et al., 2018), etc. In Section 4, we focus on unconditional GANs as the subject of our study and work on the following three recent state-of-the-art GAN techniques: ProGAN (Karras et al., 2018), StyleGAN (Karras et al., 2019), and StyleGAN2 (Karras et al., 2020).

Deepfake detection and attribution. Images generated by GANs bear unique patterns. Marra et al. (2019) show that GANs leave unique noise residuals to generated samples, which facilitate deepfake detection. Yu et al. (2019b) move one step further, using a neural network classifier to attribute different images to their sources. Wang et al. (2020) also train a classifier and improve the generalization across different GAN techniques. Zhang et al. (2019b); Durall et al. (2019; 2020) point out that the high-frequency pattern mismatch can serve as an effective cue for deepfake detection, so can the texture feature mismatch (Liu et al., 2020).

However, these cues are never long-lasting against the steady improvement of GANs because the advancement of deepfake detection is double-edged and can be accompanied by detection countermeasure techniques. For example, spectral regularization Durall et al. (2020) is proposed to narrow down the frequency mismatch and results in a significant detection deterioration. So do adversarial evasion attacks (Carlini & Farid, 2020). Therefore, it is not sustainable to establish deepfake detection based on the known problems of GANs - these problems are also known to malicious individuals and can be sidestepped. That motivates us to propose a novel solution in Section 3 that is independent of this arms race and agnostic to GAN evolution.

Image steganography. Image steganography represents a technique to hide information into carrier images, in the initial purpose of covert communication (Fridrich, 2009). Previous steganography techniques (Cox et al., 2002; Cayre et al., 2005) rely on Fourier transform or least significant bits modification (Pevnỳ et al., 2010; Holub & Fridrich, 2012; Holub et al., 2014). Recent works substitute hand-crafted hiding procedures with neural network embedding (Baluja, 2017; Hayes & Danezis, 2017; Vukotić et al., 2018; Zhang et al., 2019a; Tancik et al., 2020). In this work, we propose to root deepfake detection down to the source of GANs, and therefore leverage steganography to embed artificial fingerprints into training data. This is the first study to train GANs with fingerprinted data, and to justify the transferability of fingerprints from data to GAN models. Thanks to the stealthiness, the original GAN quality is preserved and validated in Section 4.3.

3 Artificial Fingerprints for Deepfake Detection/Attribution

The goal of image attribution is to learn a mapping $D_0(\mathbf{x}) \mapsto y$ that traces the source $y \in \mathbb{Y} = \{\text{real}, \text{GAN}_1, \dots, \text{GAN}_\text{N}\}$ 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. In practice, however, $\mathbb{Y}$ can be unlimited, barely predefined, and agnostic to us. This open-world scenario is intractable using discriminative learning. In order to generalize our solution to being agnostic to the selection of GANs, 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.

In order to further generalize our solution to being agnostic to the evolution of GANs and independent of the detection countermeasure arms race, we propose a pipeline to root the attribution down to the GAN training dataset $\tilde{\mathbf{x}} \in \tilde{\mathbb{X}}$ and close the loop of the regression D. The pipeline consists of four stages depicted in Figure 1 and described below.

Steganography training. We introduce the concept of **artificial fingerprints** representing the source identity $\mathbf{w}$. We use steganography techniques (Baluja, 2017; Tancik et al., 2020) to learn an encoder $E(\tilde{\mathbf{x}}, \mathbf{w}) \mapsto \tilde{\mathbf{x}}_\mathbf{w}$ that embeds an arbitrary fingerprint $\mathbf{w}$ into an arbitrary image $\tilde{\mathbf{x}}$. In the meanwhile, 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 loss:

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

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

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

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

where $\mathbf{w}_k$ and $\hat{\mathbf{w}}_k$ are the kth 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 whatever 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 architecture of E and D are depicted in the Figure 5 and 6 in Appendix.

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

GAN training. Our solution is agnostic to GAN techniques and therefore tackles GAN training as a black box. We simply replace $\tilde{\mathbb{X}}$ with $\tilde{\mathbb{X}}_{\mathbf{w}}$ to train GAN in the original manner.

Attribution via fingerprint detection. We *hypothesize* the **transferability** of our artificial fingerprints from training data to GAN models: A well-trained generator $G_{\mathbf{w}}(\mathbf{z}) \mapsto \mathbf{x}_{\mathbf{w}}$ contains the same fingerprint information as well, i.e., $D(\mathbf{x}_{\mathbf{w}}) \equiv \mathbf{w}$, each $\mathbf{x}_{\mathbf{w}}$ is also fingerprinted with the same fingerprint $\mathbf{w}$ as embedded in $\tilde{\mathbf{x}}_{\mathbf{w}}$. We empirically justify our hypothesis, the transferability, in Section 4.2. Based on the transferability, we can formulate the attribution regression mapping using our well-trained steganography decoder D.

4 EXPERIMENTS

We describe the experimental setup in Section 4.1. We justify the transferability of our artificial fingerprints, its generality and synergy in Section 4.2. We justify its fidelity in Section 4.3. The transferability in turn enables accurate deepfake detection and attribution, which is evaluated and compared in Section 4.4 and 4.5 respectively. In Section 4.6, we validate its robustness and working ranges. In addition, we articulate our network designs and training details in Section A.1 in Appendix, as well as validate the secrecy of fingerprints in Sec A.2 in Appendix.

4.1 SETUP

Datasets. We conduct experiments on CelebA human face dataset (Liu et al., 2015) with image size $128 \times 128 \times 3$, and LSUN bedroom scene dataset (Yu et al., 2015) with image size $128 \times 128 \times 3$. We train/evaluate on 150k/50k CelebA, and 50k/50k LSUN.

GAN models. Our solution is agnostic to GAN configurations. Without losing representativeness, we focus on three recent state-of-the-art GAN architectures: ProGAN (Karras et al., 2018), StyleGAN (Karras et al., 2019), and StyleGAN2 (Karras et al., 2020). Each model is trained from scratch with the official implementations.

4.2 TRANSFERABILITY OF FINGERPRINTS

The transferability indicates the artificial fingerprints that are embedded in the GAN training data also appear consistently in the GAN-generated data. This is a non-trivial hypothesis in Section 3 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 fingerprint detection accuracy. We set $n = 100$ as suggested in (Tancik et al., 2020).

Baseline. For comparison, we implement a straightforward baseline method. Instead of embedding fingerprints into GAN training data, we enforce fingerprint generation jointly with GAN training. That is, we train on clean data, and enforce each generated image to not only look realistic approximating the real training data, but also contain a specific fingerprint. Mathematically,

$$\min_{G,D} \max_{Dis} \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(\mathbf{0},\mathbf{I}), \tilde{\mathbf{x}} \sim \tilde{\mathbb{X}}} L_{\text{adv}}(\mathbf{z}, \tilde{\mathbf{x}}; G, Dis) + \eta \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(\mathbf{0},\mathbf{I}), \mathbf{w} \sim \{0,1\}^n} L_{\text{BCE}}(\mathbf{z}, \mathbf{w}; G, D) \tag{5}$$

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.

Table 1: Fingerprint detection in bitwise accuracy ($\Uparrow$ indicating a higher value is more desirable) and generation quality in FID ($\Downarrow$ indicating a lower value is more desirable). The "Data" rows correspond to real testing images for sanity check. The "Original FID" column corresponds to the generation quality of original (non-fingerprinted) GANs for references.

Dataset	Model	Bitwise accuracy $\Uparrow$	Original FID $\Downarrow$	Fingerprinted FID $\Downarrow$
CelebA	Data	1.00	-	1.15
	ProGAN (baseline)	0.93	14.09	60.28
	ProGAN	0.98	14.09	14.38
	StyleGAN	0.99	8.98	9.72
	StyleGAN2	0.99	6.41	6.23
LSUN	Data	1.00	-	1.02
	ProGAN (baseline)	0.87	29.16	183.63
	ProGAN	0.93	29.16	32.58
	StyleGAN	0.98	24.95	25.71
	StyleGAN2	0.99	13.92	14.71

Results. We report fingerprint detection accuracy in Table 1. We observe:

(1) The "Data" rows are for sanity checks: They reach the 100% saturated accuracy, indicating the effectiveness of the steganography technique on real data.

(2) Our artificial fingerprints can be almost perfectly detected over varying datasets and GAN configurations, with accuracy ≥ 0.98 on CelebA and LSUN, except for ProGAN on LSUN which is a challenging case as ProGAN is known not good at generating LSUN (original FID only 29.16). Our hypothesis on the **transferability** from training data to GAN models (generated data) is justified. As a result, artificial fingerprints are qualified for deepfake detection in Section 4.4 where only generated images contain fingerprints, and qualified for deepfake attribution in Section 4.5 where different GANs are trained with different fingerprints.

(3) The **generality** of fingerprint transferability over varying GAN configurations justifies our solution is agnostic to GAN techniques. Furthermore, along with the evolution of GAN techniques (from ProGAN to StyleGAN and then to StyleGAN2), the fingerprint accuracy also improves, indicating the **synergy** of our solution from GANs: If a GAN technique can approximate real data distribution more accurately, it also transfers the artificial fingerprints more accurately. That makes our solution independent of the detection countermeasure arms race.

(4) The baseline method fails with fingerprint detection accuracy moderately worse than ours and FID far worse than ours and the original ones. This indicates GAN fingerprinting is a non-trivial task, and direct fingerprint reconstruction is incompatible with the adversarial training. In contrast, our solution of leveraging image steganography and fingerprint transferability sidesteps this issue and leads to advantageous performance.

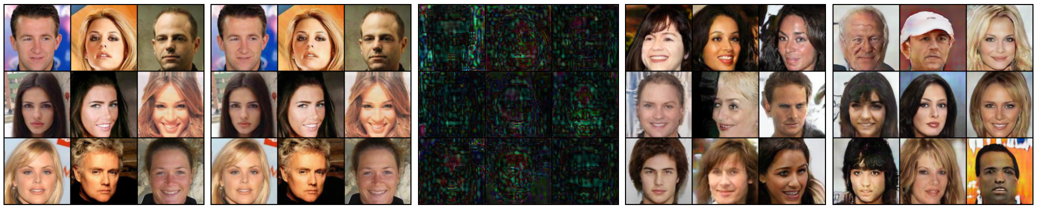
4.3 Fidelity of Fingerprints

The fidelity of fingerprints is as critical as its transferability. It requires a negligible side effect of our fingerprints on the original functionality of GANs. On one hand, it preserves the original generation quality. On the other hand, it avoids the adversary's suspect of the presence of fingerprints. In principle, the steganography technique we used should enable this, and we validate it empirically.

Evaluation. We use Fréchet Inception Distance (FID) (Heusel et al., 2017) to evaluate generation quality, the lower the more realistic. We measure FID between a set of generated images and a set of real non-fingerprinted images, in order to evaluate the quality of the former set. When calculating FID for different generations, the latter set is unchanged.

Results. We compared generation quality between original and fingerprinted GANs in Table 1. We observe:

(1) The "Data" rows are for sanity checks: Embedding fingerprints on the real images does not substantially deteriorate image quality: FID ≤ 1.15 is in an excellent realism range. That validates the secrecy of the steganographic technique and lays a valid foundation for high-quality GAN training.



(a) Original real training samples. (b) Fingerprinted real training samples. (c) Difference between 7a and 7b (10× magnified). (d) Samples from the non-fingerprinted GAN. (e) Samples from the fingerprinted GAN.

Figure 2: Samples for Table 1 on CelebA. See more samples on LSUN in Figure 7 in Appendix.

(2) The performance of the fingerprinted GANs tightly sticks to the performance limit of the non-fingerprinted baselines with the FID variance within a range of $\pm 8.2\%$ on CelebA and $\pm 11.7\%$ on LSUN. In practice, the generated fingerprints are imperceptibly hidden in the generated images and can only be perceived with $10\times$ magnification. See Figure 2 and Figure 7 in Appendix for demonstrations. Thus, the fidelity of our fingerprints is justified and it qualifies our solution for deepfake detection and attribution in Section 4.4 and 4.5.

4.4 Deepfake Detection

Unlike existing methods that detect intrinsic differences between the real and deepfake classes (Yu et al., 2019b; Zhang et al., 2019b; Durall et al., 2020), we root the classification performance down to the origin by embedding artificial fingerprints into GAN models and their generated images. In particular, we enable GAN publishers with responsible disclosure to publicize fake images only from fingerprinted GAN models. Then we convert the problem to verifying if one decoded fingerprint is in our fingerprint regulation database or not. Considering our non-perfect fingerprint detection accuracy, we allow a 1-7 bits margin depending on the selection of GANs. 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 unlikely collides 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.

Baseline. Without losing representativeness, we compare to a recent state-of-the-art CNN-based deepfake detector (Yu et al., 2019b) as a baseline method. It is trained on 50k real images and 50k generated images equally from four fingerprinted GANs. We consider two scenarios, a **closed world** and an **open world**, depending on whether or not the set of GAN models used for classifier training covers that used for testing. The open-world scenario challenges the generalization of detection.

Results. We report deepfake detection accuracy in Table 2. We observe:

(1) Agnostic to datasets and GAN techniques, deepfake detection based on our fingerprints performs equally perfectly ($\sim 100\%$ accuracy) to that based on the CNN classifier in the closed world.

(2) More advantageously, our solution performs equally well in the open-world scenario while the CNN classifier deteriorates to random guess ($\sim 50\%$ accuracy). This is because the CNN classifier is troubled by the domain gap between training and testing GAN models. In contrast, our solution enjoys the advantage of being agnostic to GAN models. It depends only on the presence of fingerprints rather than the discriminative information overfit to a closed world.

(3) As a conclusion, this suggests an administrative practice for media broadcasts. We urge the media administrators to regulate responsible disclosure of media publications: Publicizing GAN models or deepfake media requires fingerprinting in advance.

4.5 Deepfake Attribution

The goal of the attribution is to trace the GAN source that generated a deepfake media. It plays an important role in tracing the responsible of a deepfake publisher. Our artificial fingerprint solution is straightforward to extend for attribution.

Table 2: Closed/open-world deepfake detection accuracy and attribution accuracy (⇑ indicating a higher value is more desirable).

Dataset	Model	Method	Deepfake detection accuracy ⇑		Deepfake attribution accuracy ⇑	
			Closed world	Open world	Closed world	Open world
CelebA	ProGAN	Yu et al. (2019b)	0.997	0.508	0.998	0.235
		Ours	1.000	1.000	1.000	1.000
	StyleGAN	Yu et al. (2019b)	0.994	0.497	0.999	0.168
		Ours	1.000	1.000	1.000	1.000
	StyleGAN2	Yu et al. (2019b)	0.995	0.500	1.000	0.267
		Ours	1.000	1.000	1.000	1.000
LSUN	ProGAN	Yu et al. (2019b)	1.000	0.493	0.986	0.597
		Ours	1.000	1.000	1.000	1.000
	StyleGAN	Yu et al. (2019b)	0.994	0.499	0.995	0.366
		Ours	1.000	1.000	1.000	1.000
	StyleGAN2	Yu et al. (2019b)	0.988	0.491	1.000	0.267
		Ours	1.000	1.000	1.000	1.000

Closed-world scenario. In the closed-world scenario, the model space is finite and known in advance. Without losing generalization, we train four GAN models using four different fingerprints. The task is to attribute a mixture of 50k images evenly generated by these models. We apply our decoder to decode the fingerprint from an image, and assign that image to the GAN with the closest GAN fingerprint.

Open-world scenario. We further consider the open-world scenario to validate if an attribution approach can accurately reject images from unknown GANs. We introduce another four GANs trained on unknown fingerprints and require to attribute another 12.5k images evenly generated by these four GANs, meaning to label them as not belonging to any of the four known GANs. Our fingerprint solution classifies an image as unknown if and only if the number of matching bits between the detected fingerprint and the closest known fingerprint is less than 75%.

Baseline. Yu et al. (2019b) use a CNN classifier to solve deepfake attribution as a multi-class classification problem, which is limited to the closed world. We followed their protocol in the closed world scenario: training over 50k images generated evenly by each of the four GANs. We also extend their method to the open world via training 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, we assign the image to the unknown label.

Results. We report deepfake attribution accuracy in Table 2. We obtain the same discoveries and conclusions as those of deepfake detection in Section 4.4.

4.6 Robustness of Fingerprints

Deepfake media and GAN 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 fingerprint detection in bitwise accuracy over 50k images from a fingerprinted ProGAN. We plot the bitwise accuracy w.r.t. the amount of perturbations in Figure 3 and 8 (Appendix). We observe:

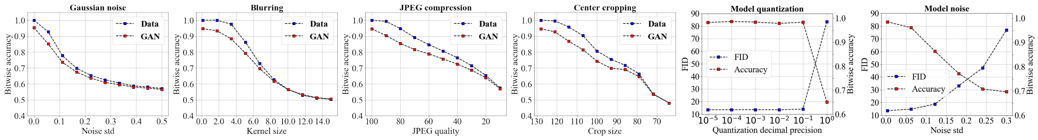


Figure 3: Zoom-in needed. Red plots show the fingerprint detection in bitwise accuracy w.r.t. the amount of perturbations over ProGAN trained on CelebA. In the left four plots (robustness against image perturbations), blue dots represent detection accuracy on the fingerprinted real training images, which serve as the upper bound references for the red dots. See Figure 8 in Appendix for additional results over ProGAN trained on LSUN. In the right two plots (robustness against model perturbations), blue dots represent FID of generated images from the perturbed models.

(a) Original 0.99 bit acc	(b) Im 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) GAN 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 w.r.t. each perturbation.

(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 $\geq 75\%$ and result 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) out of the above working ranges, the reference upper bounds drop even faster and the margins to the testing curves shrink quickly, indicating the detection deterioration is irrelevant to GAN training but rather relevant to the heavy quality deterioration of images.

(3) For model perturbations (the right two subplots) out of the above working ranges, image quality deteriorates faster than fingerprint accuracy, such that before accuracy turns lower than 75%, FID has increased by $> 500\%$.

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

5 CONCLUSION

The adversarial iterations between deepfakes and detection form an arms race. In order to lead it to the end, we investigate a fundamental and sustainable solution on the detection side, agnostic to the evolution of GANs. We present the first study to embed artificial fingerprints into GAN models. We root deepfake detection/attribution into GAN training data, and justify the transferability of artificial fingerprints from training data to GAN models. Our empirical study justifies several beneficial properties of fingerprints, including generality, synergy to GAN development, fidelity, and robustness. Based on these, we demonstrate our advantageous detection/attribution performance on multiple datasets and GAN subjects over a state-of-the-art detector (Yu et al., 2019b). This in turn enables responsible disclosure by GAN publishers or even regulation on the GAN disclosure process by allocating each publisher a unique fingerprint.

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, and Matthieu Devin. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. In *arXiv*, 2016.

Amazon AWS ML. URL https://aws.amazon.com/machine-learning.

Shumeet Baluja. Hiding images in plain sight: Deep steganography. In *NeurIPS*, 2017.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2018.

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

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

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

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 *CVPR*, 2018.

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

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

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.

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

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

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.

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

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

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.

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

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

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

Charlotte Jee. An indian politician is using deepfake technology to win new voters. 2020. URL https://www.technologyreview.com/2020/02/19/868173/an-indian-politician-is-using-deepfakes-to-try-and-win-voters.

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

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

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

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.

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

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

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

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

Microsoft Azure ML. URL `https://azure.microsoft.com/en-us/services/machine-learning`.

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

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

Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. Pytorch. 2016. URL `https://github.com/pytorch/pytorch`.

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

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

Dan Robitzski. Someone used deepfake tech to invent a fake journalist. 2020. URL `https://futurism.com/the-byte/deepfake-fake-journalist`.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *CVPR*, 2020.

James Vincent. An online propaganda campaign used ai-generated headshots to create fake journalists. 2020. URL `https://www.theverge.com/2020/7/7/21315861/ai-generated-headshots-profile-pictures-fake-journalists-daily-beast-investigation`.

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

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.

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.

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

Ning Yu, Connelly Barnes, Eli Shechtman, Sohrab Amirghodsi, and Michal Lukac. Texture mixer: A network for controllable synthesis and interpolation of texture. In *CVPR*, 2019a.

Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *ICCV*, 2019b.

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

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

Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. In *WIFS*, 2019b.

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

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

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*, 2017b.

A Appendix

A.1 Implementation Details

Steganography encoder. The encoder is trained to embed a fingerprint into an image while minimizing the pixel difference between the input and stego images. We follow the technical details in (Tancik et al., 2020). The binary fingerprint vector is first passed through a fully-connected layer and then reshaped as a tensor with one channel dimension and with the same spatial dimension of the cover image. We then concatenate this fingerprint tensor and the image along the channel dimension as the input to a U-Net architecture (Ronneberger et al., 2015). The output of the encoder, the stego image, has the same size as that of the input image. Note that passing the fingerprint through a fully-connected layer allows for every bit of the binary sequence to be encoded over the entire spatial dimensions of the input image and flexible to the image size. In our experiments, the image size is set to $128 \times 128 \times 3$ without losing representativeness. The fingerprint length is set to 100 as suggested in (Tancik et al., 2020). The length of 100 bits leads to a large enough space for fingerprint allocation while not having a side effect on the fidelity performance. We visualize the encoder architecture in Figure 5.

Steganography decoder. The decoder is trained to detect the hidden fingerprint from the stego image. We follow the technical details in (Tancik et al., 2020). It consists of a series of convolutional layers with kernel size 3x3 and strides ≥ 1, dense layers, and a sigmoid output activation to produce a final output with the same length as the binary fingerprint vector. We visualize the encoder architecture in Figure 6.

Steganography training. The encoder and decoder are jointly trained end-to-end w.r.t. the objective in Eq. 1 and with randomly sampled fingerprints. The encoder is trained to balance fingerprint detection and image reconstruction. At the beginning of training, we set $\lambda = 0$ to focus on fingerprint detection, otherwise, fingerprints cannot be accurately embedded into images. After the fingerprint detection accuracy achieves 95% (that takes 3-5 epochs), we increase λ linearly up to 10 within 3k iterations to shift our focus more on image reconstruction. We train the encoder and decoder for 30 epochs in total. Given the batch size of 64, it takes 3 hours using 1 NVIDIA Tesla V100 GPU with 16GB memory.

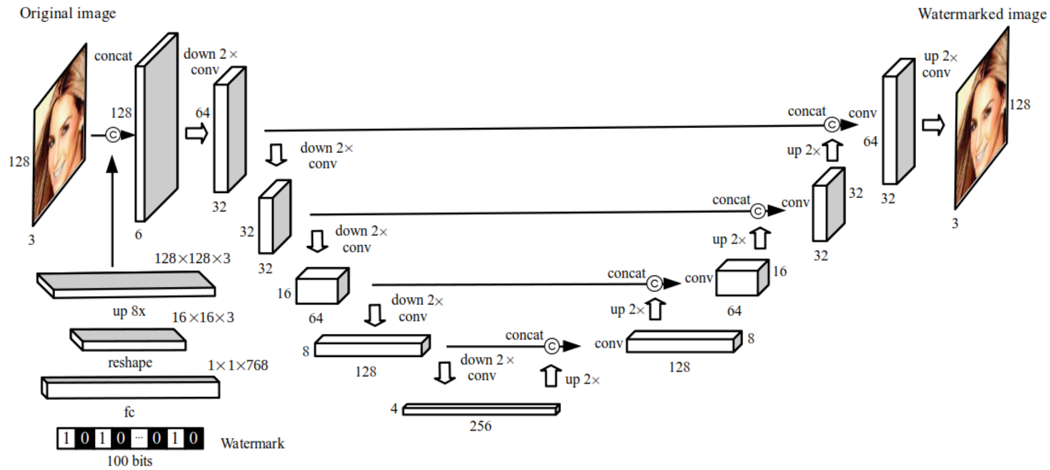


Figure 5: Steganography encoder architecture.

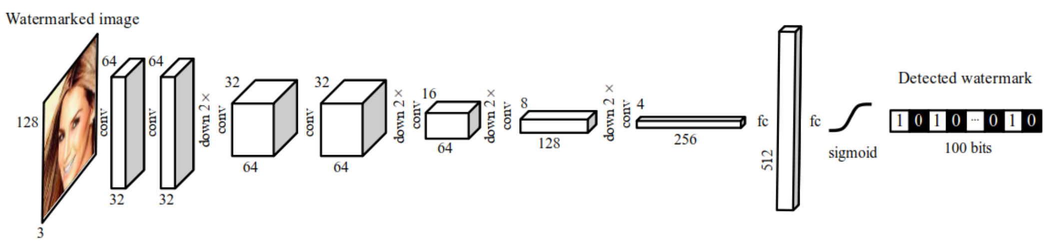


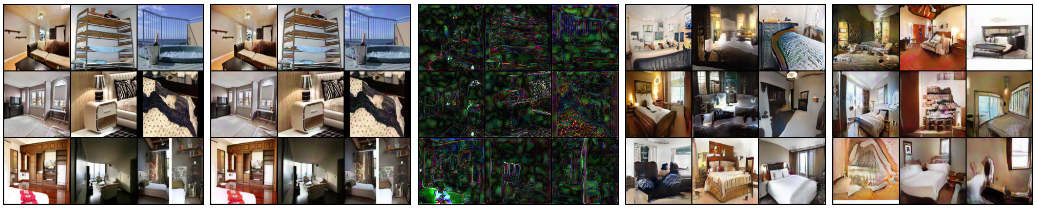
Figure 6: Steganography decoder architecture.

A.2 Secrecy of Fingerprints

The presence of a fingerprint embedded in a GAN model should not be easily detected by the third party, otherwise, it would be potentially manipulated and restart the deepfake arms race. This property is more demanding than fidelity because high fidelity just avoids intuitive detection while high **secrecy** requires technical counter-detection against steganalysis.

Attack. In order to design a quantitative evaluation on secrecy, we consider from the outsider side a binary classification problem: the presence of fingerprint in an image. We follow the attack protocol in (Zhu et al., 2018) to perform the Artificial Training Sets (ATS) attack (Lerch-Hostalot & Megías, 2016). We target to separate testing images fingerprinted 0 or 1 time but we have no supervision. The intuition is to expand the testing set and establish an artificial setting with known labels that enable supervised training, such that the original testing class space is a subspace of the artificial training class space and is separable by the training task. The attack is as follows: We independently trained another steganography encoder. We regard the original testing images as negative training samples. Then, we apply the encoder twice to the testing 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 (Chang & Lin, 2011) 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 testing images so that the originally non-fingerprinted images now are 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 our SVM to separate them and propagate the predictions back to the original images. We evaluate the attack on a set of 250 fingerprinted deepfake images and 250 non-fingerprinted deepfake images.

Results. The binary classification accuracy on the existence of fingerprint is $\mathbf{0.502}$ according to the ATS attack, which is close to random guess ($\sim 50\%$ accuracy). We reason that the third-party



(a) Original real training samples. (b) Fingerprinted real training samples. (c) Difference between 7a and 7b (10× magnified). (d) Samples from the non-fingerprinted GAN. (e) Samples from the fingerprinted GAN.

Figure 7: samples for Table 1 on LSUN, supplementary to Figure 2 in the main text.

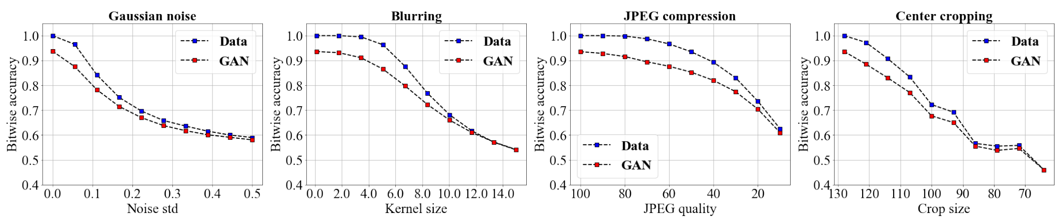


Figure 8: Red plots show the fingerprint detection in bitwise accuracy w.r.t. the amount of perturbations over ProGAN trained on LSUN. Blue dots represent detection on the fingerprinted real training images, which serve as the upper bound references for the red dots. This is supplementary to Figure 3 in the main text.

steganography encoder trained from different initialization uses different patterns to hide the fingerprint, and therefore does not couple well with the victim encoder. In conclusion, as long as we keep our encoder private, the existence of fingerprint in a GAN model is validated secret from the ATS attack.