Are Watermarks Bugs for Deepfake Detectors? Rethinking Proactive Forensics Supplementary Material

Xiaoshuai Wu , Xin Liao*, Bo Ou , Yuling Liu , Zheng Qin
College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
{shinewu, xinliao, oubo, yuling_liu, zqin}@hnu.edu.cn

In the following, we provide more details on our dataset preparation, publicly available Deepfake detectors, parameter settings for the distortions, additional experiments, etc.

1 Deepfake Generation

This supplementary is for Sec. 5.1 "Dataset Preparation" of the main paper. The dataset contains real images sourced from CelebA-HQ [Karras *et al.*, 2018] and four categories of fake images, i.e., SimSwap [Chen *et al.*, 2020] for face swapping, FOMM [Siarohin *et al.*, 2019] for expression reenactment, StarGAN [Choi *et al.*, 2018] for attribute editing, and StyleGAN [Karras *et al.*, 2019] for entire synthesis. The generation of each fake category is elaborated as follows:

- **SimSwap**[1]**.** The target face used for swapping is randomly selected from the validation set of CelebA [Liu *et al.*, 2015], which includes 19,867 face images.
- **FOMM**[2]**.** The target expression used for reenactment is driven by the randomly selected frame from "2.mp4"[3], a video clip featuring Trump.
- **StarGAN**[4]**.** The attribute we edit is gender, i.e., changing from female to male and vice versa, which alters the original images to a large extent.
- **StyleGAN**[5]**.** Rather than manipulating the real images, we use the fully synthesized images released by NVlabs[6].

2 Deepfake Detectors

This supplementary is for Sec. 5.2 "Implementation Details" of the main paper. As the work is to validate the effectiveness of our helpful adversarial watermarking rather than developing new Deepfake detectors, we utilize nine well-trained detectors, namely Xception [Rossler *et al.*, 2019], EfficientNet [Li *et al.*, 2021], CNND [Wang *et al.*, 2020], FFD [Dang *et al.*, 2020], PatchForensics [Chai *et al.*, 2020], MultiAtt [Zhao *et al.*, 2021], RFM [Wang and Deng, 2021], RECCE [Cao *et al.*, 2022], and SBI [Shiohara and Yamasaki, 2022].

- **Xception**[7]**.** The most classic detector, based on the backbone XceptionNet [Chollet, 2017], was trained on the FaceForensics++ dataset [Rossler *et al.*, 2019].
- **EfficientNet**[8]**.** The detector based on the backbone EfficientNet-B3 [Tan and Le, 2019] was trained on the FFHQ dataset and StyleGAN generated images [Karras *et al.*, 2019].
- **CNND**[9]**.** The detector based on the backbone ResNet-50 [He *et al.*, 2016] was trained on the ProGAN-generated images and real images [Karras *et al.*, 2018]. The image is possibly blurred and JPEG-ed, each with 50% probability.
- **FFD**[10]**.** The detector, based on the backbone XceptionNet and equipped with the manipulation appearance module to generate the attention maps, was trained on the DFFD dataset [Dang *et al.*, 2020].
- **PatchForensics**[11]**.** The patch-based detector which utilizes the truncated Xception Block 5 [Chai *et al.*, 2020], was trained on the ProGAN-generated and real images.
- **MultiAtt**[12]**.** The detector, based on the backbone EfficientNet-B4 with L2 for the feature layer and L5 for the attention layer [Zhao *et al.*, 2021], was trained on the FaceForensics++ dataset.
- **RFM**[13]**.** The detector, based on the backbone XceptionNet and the augmentation strategy of suspicious forgeries erasing [Wang and Deng, 2021], was trained on the DFFD dataset.
- **RECCE**[14]**.** With the reconstruction learning, multi-scale graph reasoning, and reconstruction guided attention module [Cao *et al.*, 2022], the XceptionNet-based detector was trained on the FaceForensics++ dataset.

[*] Corresponding author.

[1] https://github.com/neuralchen/SimSwap

[2] https://github.com/AliaksandrSiarohin/first-order-model

[3] https://github.com/graphemecluster/first-order-model-demo/tree/main/videos

[4] https://github.com/yunjey/stargan

[5] https://github.com/NVlabs/stylegan

[6] https://drive.google.com/drive/folders/14uyb1Du_Vc8woAa8Xj9IEFaPGQyl2ptO

[7] https://github.com/ondyari/FaceForensics

[8] https://github.com/ldz666666/Style-atk

[9] https://github.com/peterwang512/CNNDetection

[10] https://github.com/JStehouwer/FFD_CVPR2020

[11] https://github.com/chail/patch-forensics

[12] https://github.com/yoctta/multiple-attention

[13] https://github.com/crywang/RFM

[14] https://github.com/VISION-SJTU/RECCE

- **SBI**[15]**.** The EfficientNet-B4-based detector was trained on the real images in the FaceForensics++ dataset and the self-blended images [Shiohara and Yamasaki, 2022].

	SimSwap	FOMM	StarGAN	StyleGAN
Xception	8.64	8.92	62.75	2.97
EfficientNet	53.26	17.99	94.76	44.90
CNND	39.52	0.28	51.84	19.83
FFD	53.12	44.62	69.69	71.25
PatchForensics	10.20	19.97	99.86	3.68
MultiAtt	13.17	28.61	27.76	11.90
RFM	99.86	93.48	98.87	37.11
RECCE	29.32	71.81	81.16	59.21
SBI	26.20	17.42	56.52	20.11

Table 1: Accuracy on SimSwap, FOMM, StarGAN, and StyleGAN.

Table 1 shows the accuracy test on four fake subsets. The detection results suggest that it's challenging for most existing detectors to identify out-of-distribution Deepfakes in the wild [Le *et al.*, 2024]. The core of our proposed helpful adversarial watermarking is to make original incorrectly predicted inputs yield correct detection outcomes.

3 Distortion Setup

This supplementary is for Sec. 5.4 "Watermark Extraction" of the main paper. To test the robustness of watermark extraction, we consider various distortions. *Identity* means the noise-free results. *JPEG* indicates the real lossy compression using a default quality factor of 50. *simulated JPEG* refers to the implementation of JPEG-Mask [Jia *et al.*, 2021]. *Resize* reduces the watermarked image to half of the resolution and then zooms back to the original size. *Gaussian Blur* noise blurs the watermarked image with a Gaussian kernel of size 3 and standard deviation 2. *Median Blur* noise blurs the watermarked image with the kernel of size 3. Based on the Kornia library [Riba *et al.*, 2020], we implement *Brightness* with the parameter 0.5, *Contrast* with 0.5, *Saturation* with 0.5, and *Hue* with 0.1. Moreover, *Dropout* means that the pixels with a ratio of 50% are randomly replaced by pixels at the corresponding position of the host image. *Salt Pepper* noise is defined as randomly replacing 10% pixels of the watermarked image with 0 or 255. *Gaussian Noise* means adding Gaussian distributed noise with the deviation 0.1. The distortion effects they bring to the watermarked images can be referred to as the visualization results from SepMark [Wu *et al.*, 2023].

4 Additional Experiments

Fine-tuning Other Watermarking. To verify that the proposed AdvMark can also be seamlessly integrated with INN-based watermarking, we first train the FIN [Fang *et al.*, 2023] from scratch on the 256 × 256 real images. The FIN-watermarked images have a Real/Fake ACC 98.94%/17.07%, showing that FIN is also harmful to the detector Xception. After our adversarial fine-tuning, the ACC of the adversarial images increases to 92.78%/86.51%. Meanwhile, the JPEG BER changes from 0.061% to 0.037%, and the PSNR changes from 46.398 dB to 39.484 dB, where the images remain visually pleasing.

The robustness resists against geometrical attacks may depend on the utilized watermarking backbones. To verify this, we train PIMoG [Fang *et al.*, 2022] from scratch, followed by the adversarial fine-tuning using AdvMark. The PIMoG-watermarked images have a Real/Fake ACC 98.16%/16.22%, and the ACC of the adversarial images increases to 99.68%/93.48%. Meanwhile, the bit error rate, under the screen-shooting noise which involves both valuemetric and geometrical distortions, changes from 2.332% to 0.321%, and the PSNR changes from 37.758 dB to 36.297 dB. The seamless integration with MBRS, SepMark, FIN, and PIMoG verifies the feasibility and effectiveness though the implementation is classical. We also admit that AdvMark aligns with AI for Good, as it addresses important problems in AIGC security and Responsible AI.

	Xception ACC		BER	PSNR	SSIM
	Real	Fake			
FIN	98.94	17.07	0.061	46.40	0.978
FIN + AdvMark	92.78	86.51	0.037	39.48	0.922
PIMoG	98.16	16.22	2.332	37.76	0.930
PIMoG + AdvMark	99.68	93.48	0.321	36.30	0.910

Table 2: Plug-and-play with existing watermarking.

Robustness of Benign Functionality. Taking the white-box detector Xception as an example, the adversarial images have a Real/Fake ACC 99.82%/99.82% for fine-tuned MBRS, and 99.82%/99.68% for fine-tuned SepMark, respectively, before JPEG compression. After JPEG compression, the results drop to 93.70%/57.26% and 96.49%/96.18%, which are still much better than 98.19%/20.82% of the clean images.

We can explicitly adopt the data augmentation strategy in this respect, which simply substitutes x_w with $\widetilde{x_w}$:

$$\mathcal{L}_{\mathcal{D}_N} = \mathcal{F}(\mathcal{D}(\widetilde{x_w}), y) = \mathcal{F}(\mathcal{D}(\mathcal{N}(En_I(\theta; x, w))), y). \quad (1)$$

With $\mathcal{L}_{\mathcal{D}_N}$, the adversarial nature is still preserved at most distortions. Tables 3 shows that when the backbone is MBRS, AdvMark with original fooling loss $\mathcal{L}_{\mathcal{D}}$ obtains an average detection accuracy of 75.48%, while that with $\mathcal{L}_{\mathcal{D}_N}$ improves the accuracy to 91.36% even the watermarked images have been distorted by JPEG compression. Due to the noise layer of MBRS, once the watermarked images have been distorted by Gaussian noise, both $\mathcal{L}_{\mathcal{D}}$ and $\mathcal{L}_{\mathcal{D}_N}$ perform at chance. In contrast, based on SepMark, AdvMark with $\mathcal{L}_{\mathcal{D}_N}$ improves the accuracy from 50.12% to 81.09%, under the distortion of Gaussian noise. These indicate the robustness of benign functionality can also benefit from the noise layer.

		Xception ACC		JPEG	PSNR	SSIM
		JPEG	GN	BER		
MBRS	w/ $\mathcal{L}_{\mathcal{D}}$	75.48	50.00	0.909	38.72	0.909
+ AdvMark	w/ $\mathcal{L}_{\mathcal{D}_N}$	91.36	50.41	0.801	39.02	0.920
SepMark	w/ $\mathcal{L}_{\mathcal{D}}$	96.33	50.12	0.027	37.97	0.924
+ AdvMark	w/ $\mathcal{L}_{\mathcal{D}_N}$	98.64	81.09	0.471	33.80	0.908

Table 3: Robustness of benign functionality.

Pure Real/Fake Attacks. We also try two negative variants with malicious intents, which simply replace y with the

[15] https://github.com/mapooon/SelfBlendedImages

given label:

$$\mathcal{L}_{\mathcal{D}_{real}} = \mathcal{F}(\mathcal{D}(x_w), 0) = \mathcal{F}(\mathcal{D}(En_I(\theta; x, w)), 0), \qquad (2)$$

and

$$\mathcal{L}_{\mathcal{D}_{fake}} = \mathcal{F}(\mathcal{D}(x_w), 1) = \mathcal{F}(\mathcal{D}(En_I(\theta; x, w)), 1). \qquad (3)$$

Taking SBI as an example, as shown in Table 4, the pure real attacks consistently deceive the detector into classifying the watermarked images as genuine. Conversely, the pure fake attacks achieve the opposite, suggesting that the watermarked images are forged. Our proposed AdvMark differs from them but shares a similar spirit with adversarial attacks, opening the door to harmless proactive forensics against Deepfake.

		SBI ACC		JPEG	PSNR	SSIM
		Real	Fake	BER		
MBRS	w/ $\mathcal{L}_{\mathcal{D}_{real}}$	100.00	0.00	0.137	39.83	0.934
+ AdvMark (SBI)	w/ $\mathcal{L}_{\mathcal{D}_{fake}}$	0.00	99.93	1.111	39.77	0.929
SepMark	w/ $\mathcal{L}_{\mathcal{D}_{real}}$	99.65	1.06	0.531	40.21	0.950
+ AdvMark (SBI)	w/ $\mathcal{L}_{\mathcal{D}_{fake}}$	2.58	98.73	0.200	38.99	0.944

Table 4: Pure real and fake attacks.

5 Disscusion

Use Case of AdvMark. We will begin with a concise yet non-trivial case to justify our method. Let $\mathbf{E} = \{E_1, E_2, ..., E_n\}$ and $\mathbf{D} = \{D_1, D_2, ..., D_n\}$ denote a set of watermark encoders and corresponding decoders, respectively. Let $\mathbf{F} = \{F_1, F_2, ..., F_{n'}\}$ represent a set of downstream forensic detectors. Suppose the responsible individual or social network embeds the provenance evidence into the published images using his/her watermark encoder E_i. Unfortunately, the proactive injection will harm the passive detectors $\mathbf{F}$, which are more prevalent and widely deployed than the solely watermark decoder D_i. This is indeed contrary to the belief in AI responsibility and does a disservice. After our adversarial fine-tuning, we obtain fine-tuned $E_i^{'}$ and $D_i^{'}$. Detecting the watermarked images amended by $E_i^{'}$ will help improve detection accuracy without tuning the passive detectors $\mathbf{F}$. Overall, this is the first study that fills the gap regarding the research between proactive and passive forensics, while not aiming to emphasize that we need both watermarking and Deepfake detection.

Comparison With SepMark. Regardless of whether robust tracking watermarking, semi-fragile detecting watermarking, or the multipurpose watermarking SepMark is used, detecting the watermarked images leads to more false-negative results when using most of the detectors $\mathbf{F}$. Moreover, SepMark is designed to protect real images through source tracing and proactive detection; in contrast, we focus on a more general watermarking approach that is applicable to both real and fake images through provenance tracking and detectability enhancement. Therefore, we leverage robust watermarking to fool forensic detectors and have not yet considered semi-fragile proactive detection in our experiments.

Explanation of Transferability. To our knowledge, iterative attacks easily overfit to the seen detector and have inferior transferability compared to fast one-time attacks. Transferability can be boosted by generative model attacks, which mitigate the overfitting by training on the dataset rather than optimizing one specific instance. Since AdvMark performs quite well in the white-box setting, we speculate that it has the potential to overfit to the seen detector, but on some unseen detectors it may have ruined the detection. We tried the relativistic fooling loss [Naseer *et al.*, 2019] to boost transferability, but its white-box performance dropped remarkably. Breaking this dilemma necessitates further exploration.

6 Broader Impact

In the AIGC era, the breakthrough progress in multimedia generation has also accelerated malicious Deepfake, commonly known as “deep learning faked face”, posing a real threat that leads to trust crisis and moral panic. It is evident that, given the current user scale and transmission speed of online social networks, along with the proliferation of highly realistic generated content, proactive traceability has gradually emerged as a complementary solution to passive detection. At present, although proactive watermark injection may not be easily perceived by human senses, passive Deepfake detectors do not share the similar perspective as the naked eyes. We reveal that the proactive injection will harm the passive detectors, which are more prevalent and widely deployed in the wild than the solely watermark decoder.

In view of the issue that current watermarking may unintentionally degrade the detection performance, our work is the first attempt to bridge proactive forensics and passive forensics, the two previously uncorrelated studies. With the belief in “adversarial for good”, our helpful adversarial watermarking exploits the adversarial vulnerability of passive detectors for good. After the injection of our adversarial watermarks, the detectors are intentionally deceived to correctly discriminate the watermarked images, thus achieving harmless provenance tracking and concurrently enhancing forensic detectability of watermarked images.

References

[Cao *et al.*, 2022] Junyi Cao, Chao Ma, Taiping Yao, Shen Chen, Shouhong Ding, and Xiaokang Yang. End-to-end reconstruction-classification learning for face forgery detection. In *CVPR*, 2022.

[Chai *et al.*, 2020] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *ECCV*, 2020.

[Chen *et al.*, 2020] Renwang Chen, Xuanhong Chen, Bingbing Ni, and Yanhao Ge. Simswap: An efficient framework for high fidelity face swapping. In *ACM MM*, 2020.

[Choi *et al.*, 2018] 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.

[Chollet, 2017] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017.

[Dang *et al.*, 2020] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K Jain. On the detection of digital face manipulation. In *CVPR*, 2020.

[Fang *et al.*, 2022] Han Fang, Zhaoyang Jia, Zehua Ma, Ee-Chien Chang, and Weiming Zhang. Pimog: An effective screen-shooting noise-layer simulation for deep-learning-based watermarking network. In *ACM MM*, 2022.

[Fang *et al.*, 2023] Han Fang, Yupeng Qiu, Kejiang Chen, Jiyi Zhang, Weiming Zhang, and Ee-Chien Chang. Flow-based robust watermarking with invertible noise layer for black-box distortions. In *AAAI*, 2023.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[Jia *et al.*, 2021] Zhaoyang Jia, Han Fang, and Weiming Zhang. Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression. In *ACM MM*, 2021.

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

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

[Le *et al.*, 2024] Binh M Le, Jiwon Kim, Shahroz Tariq, Kristen Moore, Alsharif Abuadbba, and Simon S Woo. Sok: Facial deepfake detectors. *arXiv preprint arXiv:2401.04364*, 2024.

[Li *et al.*, 2021] Dongze Li, Wei Wang, Hongxing Fan, and Jing Dong. Exploring adversarial fake images on face manifold. In *CVPR*, 2021.

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

[Naseer *et al.*, 2019] Muzammal Naseer, Salman Khan, Muhammad Haris Khan, Fahad Shahbaz Khan, and Fatih Porikli. Cross-domain transferability of adversarial perturbations. In *NeurIPS*, 2019.

[Riba *et al.*, 2020] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *WACV*, 2020.

[Rossler *et al.*, 2019] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *ICCV*, 2019.

[Shiohara and Yamasaki, 2022] Kaede Shiohara and Toshihiko Yamasaki. Detecting deepfakes with self-blended images. In *CVPR*, 2022.

[Siarohin *et al.*, 2019] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *NeurIPS*, 2019.

[Tan and Le, 2019] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.

[Wang and Deng, 2021] Chengrui Wang and Weihong Deng. Representative forgery mining for fake face detection. In *CVPR*, 2021.

[Wang *et al.*, 2020] 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.

[Wu *et al.*, 2023] Xiaoshuai Wu, Xin Liao, and Bo Ou. Sepmark: Deep separable watermarking for unified source tracing and deepfake detection. In *ACM MM*, 2023.

[Zhao *et al.*, 2021] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *CVPR*, 2021.