Are Watermarks Bugs for Deepfake Detectors? Rethinking Proactive Forensics

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

Abstract
AI-generated content has accelerated the topic of media synthesis, particularly Deepfake, which can manipulate our portraits for positive or malicious purposes. Before releasing these threatening face images, one promising solutions is to find the injection of robust watermarks to track their own provenance. However, we argue that current watermarking models, originally devised for genuine images, may harm the deployed Deepfake detectors when directly applied to forged images, since the watermarks are prone to overlap with the forgery signals used for detection. To bridge this gap, we thus propose AdvMark, on behalf of proactive forensics, to exploit the adversarial vulnerability of passive detectors for good. Specifically, AdvMark serves as a plug-and-play procedure for fine-tuning any robust watermarking into adversarial watermarking, to enhance the forensic detectability of watermarked images; meanwhile, the watermarks can still be extracted for provenance tracking. Extensive experiments demonstrate the effectiveness of the proposed AdvMark, leveraging robust watermarking to fool Deepfake detectors, which can help improve the accuracy of downstream Deepfake detection without tuning the in-the-wild detectors. We believe this work will shed some light on the harmless proactive forensics against Deepfake.

1 Introduction
Large generative models like ChatGPT and Stable Diffusion are transforming the way we communicate, illustrate, and create [Hacker et al., 2023]. Despite their positive potential in applications such as entertainment, education, art, etc., we cannot neglect the negative impacts. Particularly, the photorealistic Deepfake [Hu et al., 2022], interpreted as “fake face” generated through “deep learning”, has taken false messages and misleading content to a new level, thereby damaging social trust and public interest [Juefei-Xu et al., 2022]. In response to this dilemma, emerging standards such as the Coalition for Content Provenance and Authenticity [C2PA, 2021] call for embedding provenance information within the released images online [Bui et al., 2023]. To build Responsible AI, major tech companies, including Google and Microsoft, are also developing tools to trace the origin and identify synthetically altered or created images [Zhao et al., 2023a]. It has never been more important to understand the provenance and authenticity of the images we see, helping us make informed decisions and protect ourselves from deception.

Robust watermarking is the prevalent measure for the task of provenance tracking [Zhu et al., 2018; Tancik et al., 2020]. In such a watermarking prototype, subtle watermark signals that identify the provenance are imperceptibly embedded into the host image by a watermark encoder, rendering them almost invisible to the human naked eyes. After that, even if the watermarked image has been subjected to severe distortions, the watermark can still be extracted correctly by a watermark decoder in a blind manner, i.e., without knowledge of the original host image. It is also acknowledged that the watermark embedding can be accomplished during the content generation [Xiong et al., 2023], similar to the encoder implanted into a specific generative model. It should be noted that in this paper, we focus on more general watermarking.
methods, which modify the host images and are applicable to arbitrary generative models as well as non-generated images. In previous studies, proactive watermark injection and passive Deepfake detection were completely independent, without any consideration of their correlation. However, we reveal that in practice, the presence of watermarks leads to more false-negative results when using most passive detectors (see Table 1), which is indeed contrary to the belief in AI responsibility and does a disservice. This can be explained by the fact that current watermarking models were originally devised for genuine images, where the watermarks are prone to overlap with the forgery signals used for detection. One underlying assumption for the watermark effects is that well-trained detectors are asymmetrically tuned to detect patterns that make an image fake [Ojha et al., 2023]. Consequently, anything else without these patterns would be classified as genuine.

To gain more insights into this intriguing phenomenon, we remark that invisible watermarks are analogous to natural adversarial perturbations [Hou et al., 2023]. Both of them are crafted by subtle signals that are not easily perceptible by human eyes; adversarial perturbations are intended to cause misjudgments of the detectors, while watermarks are designed to enable provenance tracking but unintentionally reduce the detection performance. More fundamentally, from the perspective “adversarial examples are features”, they can be attributed to non-robust features that are highly predictive for the detectors [Ilyas et al., 2019]. Therefore, in another vein, it is possible for the embedded watermarks to both recoverable and adversarial at the same time.

To this end, we propose AdvMark, which exploits the adversarial vulnerability of passive detectors for good, making the watermarks adversarial to positively improve the performance of Deepfake detectors. The distinctions between the proposed AdvMark and current watermarking models are observable in Figure 1. Unlike previous harmful adversarial examples and watermarking, our proposed helpful adversarial watermarking deceives the detector into correctly classifying the input image following its ground-truth label. Specifically, in contrast to the original images that were incorrectly predicted, the watermarked images force the detector to report a correct prediction. Meanwhile, the watermarks should not change the detection results of the original correctly predicted images. It is also worth noting that the meaningful watermarks are capable of tracking the image’s provenance once extracted, while the meaningless perturbations are not.

Our contributions can be summarized into three-fold:

- **We present a harmless proactive forensics solution, AdvMark**, which leverages robust watermarking to fool Deepfake detectors for the first time, where the embedded watermarks behave both recoverable and adversarial, thereby achieving the purposes of provenance tracking and detectability enhancement simultaneously.

- To the best of our knowledge, this is the first attempt to formulate the concept of helpful adversarial watermarking, which amends the watermarked images to improve the accuracy of downstream Deepfake detection without tuning the in-the-wild detectors.

- Extensive experiments conducted on well-trained detectors demonstrate the effectiveness of the proposed AdvMark under white- and black-box attacks, on various types of Deepfake, such as face swapping, expression reenactment, attribute editing, and entire synthesis.

2 Related Work

2.1 Deepfake Forensics

Up to now, Deepfake has covered massive types and advanced rapidly, leading to more realistic and convincing face forgeries. To control the dissemination of these, countermeasures are continually developed and can be roughly classified into three categories: passive forensics, proactive defense, and proactive forensics [Wu et al., 2023]. Among them, proactive defense performs adversarial attacks to destroy the creation of Deepfake [Huang et al., 2022]; however, it leaves few forensic clues, as the visually perceptible artifacts are easily perceived and can alert the adversary [Chen et al., 2021].

**Passive Forensics.** Deep learning-based detectors exploit powerful backbone networks, such as Xception [Rossler et al., 2019] and EfficientNet [Li et al., 2021], and their automatic feature extraction empirically outperforms hand-craft features. CNN-D [Wang et al., 2020] incorporates appropriate data augmentations during the network training. FFD [Dang et al., 2020] utilizes informative attention maps that indicate suspicious regions to detect the forgery. PatchForensics [Chai et al., 2020] develops the patch-based detector with limited receptive fields to exaggerate local artifacts in patch regions. MultiAtt [Zhao et al., 2021] captures local discriminative features from multiple attentive regions to identify subtle and local differences between genuine and forged images. RFM [Wang and Deng, 2021] erases several sensitive facial regions to guide the detector to allocate more attention to the representative regions. RECCCE [Cao et al., 2022] mines common compact representations of genuine faces from the reconstruction perspective. SBI [Shiohara and Yamasaki, 2022] follows the self-supervised paradigm where the more general forgeries can be produced on-the-fly.

**Proactive Forensics.** To combat Deepfake proactively, FakeTagger [Wang et al., 2021] is derived from robust watermarking, where the embedded watermarks can survive both before and after Deepfake, enabling the tracking of the provenance of watermarked images. FaceSigns [Neekhara et al., 2022] introduces a semi-fragile watermarking framework, and its proactive detection is realized through verifying the extracted watermarks. SepMark [Wu et al., 2023] enables the embedded watermark to be extracted at different levels of robustness, achieving the purposes of provenance tracking and proactive detection simultaneously. Likewise, BiFPro [Liu et al., 2023] extends watermarking into diverse forensic scenarios where the watermark manifests either fragility or robustness. The embedded watermarks can be identity-related messages [Zhao et al., 2023b; Wang et al., 2023; Guan et al., 2023]. Other than deep watermarking, Source-ID-Tracker [Lin et al., 2022] employs steganography to hide the original images within their forged versions. PIMD [Asnani et al., 2022] and MaLP [Asnani et al., 2023] add the learned templates to conduct proactive detection and location, respectively. Moreover, AFP [Yu et al., 2021] and Faketracer
[Sun et al., 2022a] poison the training data, resulting in watermarked generative models.

2.2 Adversarial Attacks

Seminal work [Szegedy et al., 2014] discloses the vulnerable nature of deep neural networks to adversarial examples. Transferability is an intriguing property of adversarial examples; deliberately perturbed inputs generated for the surrogate model can mislead other victim models’ predictions. Adversarial attacks can be categorized as white-box and black-box, depending on whether the target model can be accessed (e.g., architecture and parameters). Moreover, untargeted attacks aim to cause the model to misclassify inputs regardless of the specific category. Accordingly, targeted attacks aim to cause the model to misclassify inputs into another specific category. Notably, the generative attacks [Poursaeed et al., 2018; Xiao et al., 2018] leverage a trainable generator to directly generate desired perturbations, which are ingeniously similar to current watermarking models and require only a single forward pass in the inference stage, as opposed to instance-specific iterative optimization attacks. For more details on this avenue, the review [Zhao et al., 2023c] is recommended.

Table 1: Accuracy of well-trained Deepfake detectors tested on non-watermarked/JPEG-compressed/Gaussian noisy/watermarked images in detecting separate real and fake subsets. We have the following observations: i) the detectors predict almost all the genuine images with high accuracy, regardless of whether they are watermarked or not, but struggle with diverse types of forged images; ii) the watermarks distort the images in a counter-intuitive fashion, which is actually different from JPEG and Gaussian noise; and iii) the watermarked forged images are more likely to be predicted as real by the detectors, compared to the non-watermarked forged counterparts. The experimental setup here is consistent with Section 5.

4.1 Problem Formulation

Adversarial Perturbations Harm Detectors. Let $x$ denote the clean host image, and $y$ denote its ground-truth label. We use $\hat{x}$ to represent the adversarial image, i.e., $\hat{x} = x + \eta$, where $\eta$ is the imperceptible perturbation with the $L_p$-norm constraint $\|\eta\|_p \leq \epsilon$, $p \in \{0, 2, \infty\}$. Suppose there is a perturbation generator $\mathcal{P}(\cdot)$ generating the perturbations directly, i.e., $\hat{x} = x + \mathcal{P}(x)$, and a Deepfake detector $D(\cdot)$ predicting the label of the input. We recall the vanilla adversarial attack, where the objective is to mislead the detector into making an incorrect prediction by

$$\max_p \mathbb{E}_{(x,y)} \mathcal{F}(D(x + \mathcal{P}(x)), y),$$ (1)

where $\mathcal{F}$ denotes the commonly used cross-entropy loss or binary cross-entropy loss for the task of Deepfake detection. It should be noted that the perturbations generated by $\mathcal{P}(\cdot)$ are typically constrained by a small $L_p$-norm budget, to maintain the lower bound of visual quality but may not align well with human perception, with respect to perturbed images.
Adversarial Watermarks Help Detectors. Given a watermark encoder $En(\cdot)$ with the input of the host image $x$ and watermark $w$, we can obtain the watermarked image $x_w$, i.e., $x_w = En(x, w)$. A corresponding watermark decoder $De(\cdot)$ should be able to extract the watermark $w$ exactly from the image $x_w$. Nevertheless, considering that the watermarked image is sometimes delivered and distributed through the dirty channel, it is proposed to introduce a noise layer $N(\cdot)$ to distort the image $x_w$ with diverse data augmentations, where the distorted image $\tilde{x}_w = N(x_w)$. In this way, the robust decoder can extract an approximate watermark from the image $\tilde{x}_w$, i.e., $\hat{w} = De(\tilde{x}_w)$, meaning $\hat{w} \approx w$. Formally, the objective of our adversarial watermarking, which helps the detector classify watermarked images, can be formulated as

$$\min_{En, De} \mathbb{E}_{x,y,w} \left[ F(D(En(x, w)), y) + G(En(x, w), x) + H(De(N(En(x, w))), w) \right],$$

where $G$ denotes the loss function used for image similarity measures (e.g., mean-squared error loss and LPIPS loss [Zhang et al., 2018]), and $H$ represents the mean-squared error loss or binary cross-entropy loss, which usually characterizes the watermark extraction as either a regression or a classification problem, respectively.

4.2 AdvMark Framework

As illustrated in Figure 3, the pipeline of our proposed AdvMark consists of five components: Watermark Encoder, Noise Layer, Watermark Decoder, Adversary Discriminator and downstream Deepfake Detector. Specifically, we showcase the functionalities of each component as follows:

1) The encoder $En$ receives a batch of the host image $x \in \mathbb{R}^{B \times 3 \times H \times W}$ (which belongs to either genuine or forged), labeled by $y \in \{0, 1\}^B$, and random watermark bits $w \in \{0, 1\}^{B \times L}$ as input. It produces the watermarked image $x_w$, which has intriguing adversarial properties for helping the downstream Deepfake detectors.

2) The noise layer $N$ augments the watermarked image $x_w$ with various simulated or real distortions, resulting in the distorted image $\tilde{x}_w$. It is parameterless but differentiable, which connects the encoder $En$ and decoder $De$ to facilitate end-to-end robust training.

3) The decoder $De$ extracts the watermarked $\tilde{w}$ from the distorted image $\tilde{x}_w$, where $\tilde{w} \approx w$, to reliably track the provenance of watermarked image $x_w$. Note that it also participates in updating the encoder $En$ to generate the image that is more conducive to watermark extraction.

4) The adversary discriminator $Ad$ attempts to distinguish between the host image $x$ and the watermarked image $x_w$. Conversely, the encoder $En$ should mislead the discriminator so that $Ad$ cannot distinguish $x_w$ from $x$, to acquire better visual quality of the watermarked image.

5) The surrogate detector $D$ classifies the watermarked image $x_w$, as genuine or forged. In the case of $D(x) \neq y$, the watermarked image generated by the encoder $En$ makes sense when $D(x_w) = y$, reflecting the enhanced forensic detectability. This watermarked image is even easier to be detected by other downstream detectors.

Below we will provide a detailed description of the pre-training stage of robust watermarking, the fine-tuning stage of adversarial watermarking, and the inference stage of final watermark encoder and decoder in turn.

Encoder-Decoder Robust Training. For the watermark encoder $En$, its watermarked image $x_w$ should be visually similar to the original host image $x$:

$$\mathcal{L}_{En} = L_2(x, x_w) = L_2(x, En(\theta; x, w)), \quad (3)$$

where $\theta$ denotes the trainable parameters of the encoder.

For the watermark decoder $De$, its extracted watermarked $\tilde{w}$ should be identical to the original embedded watermark $w$, from the distorted image $\tilde{x}_w$ augmented by noise layer $N$:

$$\mathcal{L}_{De} = L_2(w, \tilde{w}) = L_2(w, De(\phi; x_w)), \quad (4)$$

where

$$\tilde{x}_w = N(x_w) = N(En(\theta; x, w)). \quad (5)$$
and $\phi$ represents the trainable parameters of the decoder.

For the adversary discriminator $Ad$, there are several loss variants for stabilizing the GAN training [Jolicoeur-Martineau, 2019], and for ease of presentation, here we describe the standard GAN loss [Goodfellow et al., 2014]:

$$L_{Ad} = \log(1 - Ad(\gamma; En(x, w))) + \log(Ad(\gamma; x)),$$

where $\gamma$ indicates the trainable parameters of the discriminator. Note that the discriminator is trained alternately with the encoder-decoder, where the encoder $En$ is also updated by

$$L_{En} = \log(Ad(x_w)) = \log(Ad(En(\theta; x, w))).$$

To sum up, the total loss for Stage I can be formulated by

$$L_I = \lambda_1 L_{En} + \lambda_2 L_{De} + \lambda_3 L_{Ad},$$

where $\lambda_1$, $\lambda_2$, $\lambda_3$ are the weights for respective loss terms. After training, we get the pre-trained encoder $En_{11}$ and decoder $De_{11}$. We can also freely adopt the released models of the pre-trained encoder and decoder for this stage. More implementation details can be found in Section 5.2.

**Adversarial Fine-tuning Procedure.** The advent of highly realistic generated content can easily deceive human eyes. Manually examining a large volume of images through observation is extremely expensive, even when the defects are obvious. Therefore, it is preferable to develop automatic Deepfake detectors on the machine end. However, with the proliferation of generated images, merely verifying their authenticity seems insufficient without understanding their provenance, since it’s difficult to differentiate between images created for malicious intents and those created for neutral or positive purposes. Building upon the watermarking at Stage I, a responsible individual or social network can embed the provenance evidence into the published images using his/her watermark encoder $En_{11}$. Unfortunately, the proactive injection will harm the passive detectors, which are more prevalent and widely deployed in the wild than the solely watermark decoder $De_{11}$. This is indeed contrary to the belief in AI responsibility and does a disservice. In light of the above, we propose a plug-and-play targeted attack procedure that fine-tunes robust watermarking into adversarial watermarking, aiming to help improve the accuracy of downstream Deepfake detection by deliberately fooling the passive detectors (i.e., crossing the decision boundary; see Figure 2).

To achieve the fooling objective, we freeze a surrogate detector $D$ and force the detector to predict the correct label $y$ with respect to the watermarked image $x_w$ by updating $En_{11}$:

$$L_D = \mathcal{F}(D(x_w), y) = \mathcal{F}(D(En_{11}(\theta; x, w)), y),$$

where $\mathcal{F}$ denotes the loss function (e.g., binary cross-entropy loss) used to train the original detector $D$.

To summarize, the total loss for Stage II is formulated by

$$L_{II} = \lambda_1 L_{En} + \lambda_2 L_{De} + \lambda_3 L_{Ad} + \lambda_4 L_D,$$

where $\lambda_4$ is the weight for the fooling loss term. After fine-tuning, we get the final encoder $En_{11}$ and decoder $De_{11}$. Detecting the watermarked images emended by $En_{11}$ will help improve detection accuracy without tuning the detectors.

[1] https://github.com/fzyustc/MBRS

**Model Inference.** In the inference process, only the well-trained watermark encoder $En_{11}$ and decoder $De_{11}$ are adopted. Consequently, depending on our adversarial watermarks, we can categorize the harmless proactive forensics into the following cases.

Case 1: If $\tilde{w} \neq w$, the suspicious image is either not watermarked by the encoder $En_{11}$ at all or has been extremely altered by significant distortions, reflecting that it is impossible to track its provenance by the decoder $De_{11}$.

Case 2: If $\tilde{w} \approx w$, the image has been watermarked by the encoder $En_{11}$, and we can extract the watermark $\tilde{w}$ to serve as evidence of the image’s provenance, and effortlessly determine whether it is genuine or forged by the detector $D$, which we refer to as enhanced forensic detectability.

Case 3: Even in a more challenging setting where the detector $D'$ was unseen during the training process, the watermarked image $x_w$ is also more easily distinguished by $D'$, which is known as adversarial transferability.

5 Experiments
5.1 Dataset Preparation

We collect the real faces sourced from Celeba-HQ [Karras et al., 2018] and resize them to the resolution of 256 $\times$ 256. These real faces are then manipulated by recent Deepfake generative models: SimSwap [Chen et al., 2020] for face swapping, FOMM [Siarohin et al., 2019] for expression reenactment, and StarGAN [Choi et al., 2018] for attribute editing. We further utilize the entire synthesized faces provided by StyleGAN [Karras et al., 2019], which are also resized to 256 $\times$ 256. These four fake categories are evenly sampled to balance the numbers of real and fake faces. More specifically, there are equal numbers of faces in real and fake subsets, and we divide them into training, validation, and testing, respectively, referencing the official split 24183/2993/2824.

5.2 Implementation Details.

As the proposed work is not focusing on developing a new robust watermarking model, we directly adopt MBRS and the robust branch of SepMark as the pre-trained encoder and decoder in Stage I, and also as our baselines. To show that the performance gain is due to our adversarial fine-tuning procedure rather than training for more iterations, we establish additional baselines where the two watermarking models undergo a vanilla fine-tuning procedure (without fooling loss) on the dataset. Moreover, we utilize nine well-trained Deepfake detectors, as described in Section 2.1, to validate the capability of forensic detectability and adversarial transferability. For the hyper-parameter settings of MBRS and SepMark, we strictly follow their original implementations. For example, besides the difference in the length of the watermark bits (256-bit for MBRS and 128-bit for SepMark), the noise layers they use are also inconsistent. To be specific, MBRS includes Identity, JPEG, and simulated JPEG, while SepMark contains all the noises listed in Table 3. Therefore, we will compare our AdvMark with the respective baselines for each to mitigate the model bias introduced by different backbones. Lastly, the whole fine-tuning lasted for 10 epochs with a batch size of 8, and we set the weight of the fooling loss $\lambda_4$ to 0.1.
The Real/Fake ACC on the separate real and fake subsets. The is also the case when several detectors, e.g., Xception, Multi-

The benign functionality of adversarial watermarking is con-

5.3 Detector Attack

The benign functionality of adversarial watermarking is confirmed in both white- and black-box scenarios. The evaluation metric here is the accuracy (ACC) of the detector $D$: 

$$ACC(D) = \frac{1}{B} \sum_{i=1}^{B} D(x^{i} \times 3 \times H \times W) - y^{i} \times 1 \times 100\%,$$  

where 

$$D(x) = \begin{cases} 
0 & \text{if } x \text{ is predicted as real,} \\
1 & \text{if } x \text{ is predicted as fake,} 
\end{cases}$$

and $y \in \{0, 1\}^{B \times 1}$ means the ground-truth label, $\cdot \cdot \cdot$ denotes the absolute value. For a comprehensive comparison, we report the Real/Fake ACC on the separate real and fake subsets. The higher the better, the better the detection performance.

White-box Attack. In the white-box scenario, we evaluate the "training accuracy" with the detector seen during training. When the watermark encoder-decoder, Deepfake detector, and the watermark are colluding, the surrogate detector used can be consistent with the downstream detector. By observing Table 2, we note that the white-box detector without any tuning can get significant performance gain in detecting the images watermarked by AdvMark. The baselines MBRS and SepMark are similar to the image processing operations, e.g., JPEG compression and Gaussian noise, which are harmful to the detector, where they distort the host image in different manners. The fine-tuned baselines also have inferior detection performance compared to the clean host images without watermarking. By contrast, our helpful AdvMark turns the watermark from an enemy into a friend, ensuring nearly 100% accuracy for the detector in the white-box setting. This is also the case when several detectors, e.g., Xception, MultiAtt, and RFM, are included for the adversarial fine-tuning.

Black-box Attack. We further evaluate the "testing accuracy" with the detector unseen during training, in the black-box scenario. Table 2 shows that compared to the clean host images, the watermarked images using AdvMark are more easily distinguished by other unseen detectors, thanks to the transferability of adversarial watermarks. For example, using MultiAtt or SBI as the surrogate detector, the resulting detection performance with AdvMark is higher than that without watermarking, on average. We notice that AdvMark is inferior to the baselines on a few unseen detectors, which may be due to the different decision logic shared by the detectors. E.g., most detectors mainly identify distinct fake patterns in the images, while the exception RECCe seeks to discriminate what is real. 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. Suggested by ensemble attacks [Pour-saeed et al., 2018], we can attack an ensemble of detectors, which is more conducive in the real world. By simultaneously attacking Xception, MultiAtt, and RFM, in the black-box setting we obtain an overall improved detection performance, which is indeed better than attacking only one detector.

5.4 Watermark Extraction & Visual Quality

Since our AdvMark focuses on improving the forensic detectability of watermarked images without compromising provenance tracking, we expect as few performance degradations in Bit Error Rate (BER) as possible. Suppose that the embedded watermark $w \in \{0, 1\}^{B \times L}$ and the extracted watermark $\hat{w} \in \{0, 1\}^{B \times L}$ formally,

$$BER(w, \hat{w}) = \frac{1}{B \times L} \sum_{i=1}^{B} \sum_{j=1}^{L} |w(i) \times j - \hat{w}(i) \times j| \times 100\%.$$  

Table 2: Detection accuracy (Real/Fake ACC) $\uparrow$ under white & black-box attacks. "x" indicates seen detector during training.
Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)

Table 3: Extraction error rate (BER \((\%\)) & PSNR \((\text{dB})\) and SSIM \((\%\)) of watermarked images. “0.000” indicates error-free extraction. Gaussian Blur, Median Blur, Salt Pepper, and Gaussian Noise are abbreviated as GB, MB, SP, and GN, respectively.

<table>
<thead>
<tr>
<th>MBRs</th>
<th>Identity</th>
<th>JPEG</th>
<th>Resize</th>
<th>GB</th>
<th>MB</th>
<th>Brightness</th>
<th>Contrast</th>
<th>Saturation</th>
<th>Hue</th>
<th>Dropout</th>
<th>SP</th>
<th>GN</th>
<th>Average</th>
<th>PSNR (dB)</th>
<th>SSIM (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdvMark (Ensemble)</td>
<td>0.000</td>
<td>0.002</td>
<td>1.001</td>
<td>3.000</td>
<td>2.500</td>
<td>1.500</td>
<td>1.000</td>
<td>1.500</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>41.31</td>
<td>0.972</td>
</tr>
<tr>
<td>AdvMark (CNND)</td>
<td>0.000</td>
<td>0.001</td>
<td>1.001</td>
<td>2.000</td>
<td>1.500</td>
<td>1.000</td>
<td>0.500</td>
<td>0.500</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>40.00</td>
<td>0.960</td>
</tr>
<tr>
<td>AdvMark (RecCE)</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.500</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>39.50</td>
<td>0.957</td>
</tr>
<tr>
<td>AdvMark (PatchFor.)</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>39.00</td>
<td>0.956</td>
</tr>
<tr>
<td>AdvMark (MultiAtt)</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>38.50</td>
<td>0.955</td>
</tr>
<tr>
<td>Vanilla Fine-tuning</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>38.00</td>
<td>0.954</td>
</tr>
</tbody>
</table>

6 Conclusion

In this work, we make the first attempt to bridge proactive forensics and passive forensics, the two previously uncorrelated studies, and propose the concept of helpful adversarial watermarking, reflecting the belief in “adversarial for good”. Without training the watermarking model from scratch, AdvMark can serve as a plug-and-play procedure, seamlessly integrating with existing watermarking. By fine-tuning robust watermarking into adversarial watermarking, we do not harm the performance of passive detectors while fulfilling proactive forensics, achieving harmless provenance tracking and concurrently enhancing forensic detectability. We also investigate the transferability of adversarial watermarks, allowing AdvMark to generalize the watermarking to unseen Deepfake detectors, further validating its effectiveness in the wild. In our future work, we plan to conduct more analyses of the watermark effects on other downstream tasks, such as tamper localization [Zhang et al., 2024a].
Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant Nos. U22A2030, U20A20174), National Key R&D Program of China (Grant No. 2022YFB3103500), Hunan Provincial Funds for Distinguished Young Scholars (Grant No. 2024JJ2025). Supplementary material, including more details and experiments of our work, can be found in the supplement at https://github.com/shInewu/AdvMark.

References


[Ilyas et al., 2019] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In NeurIPS, 2019.


