Proactive Deepfake Defence via Identity Watermarking

Yuan Zhao[1], Bo Liu[1], Ming Ding[2], Baoping Liu[1], Tianqing Zhu[1], Xin Yu[1]

[1]University of Technology Sydney, [2]Data61, CSIRO

{yuan.Zhao,baopoing.liu}@student.uts.edu.au,{bo.liu,tianqing.zhu,xin.yu}@uts.edu.au,
ming.ding@data61.csiro.au

Abstract

The explosive progress of Deepfake techniques poses unprecedented privacy and security risks to our society by creating real-looking but fake visual content. The current Deepfake detection studies are still in their infancy because they mainly rely on capturing artifacts left by a Deepfake synthesis process as detection clues, which can be easily removed by various distortions (e.g. blurring) or advanced Deepfake techniques. In this paper, we propose a novel method that does not depend on identifying the artifacts but resorts to the mechanism of anti-counterfeit labels to protect face images from malicious Deepfake tampering. Specifically, we design a neural network with an encoder-decoder structure to embed watermarks as anti-Deepfake labels into the facial identity features. The injected label is entangled with the facial identity feature, so it will be sensitive to face swap translations (i.e., Deepfake) and robust to conventional image modifications (e.g., resize and compress). Therefore, we can identify whether watermarked images have been tampered with by Deepfake methods according to the label's existence. Experimental results demonstrate that our method can achieve average detection accuracy of more than 80%, which validates the proposed method's effectiveness in implementing Deepfake detection.

1. Introduction

The advancement of deep generative approaches has led to various powerful Deepfake methods, which can synthesize visually authentic images/videos. However, abusing Deepfake techniques poses a pressing threat to the integrity of multimedia information and personal privacy, such as fake news or rumours. To counterbalance the aggressiveness of Deepfake, a new research branch known as Deepfake Detection arises, which aims to utilize traditional media forensics methods or deep learning technology to differentiate the fake images/videos from the real ones.

Existing Deepfake detection approaches mainly focus on passively capturing the artifacts introduced during the Deepfake synthesis as clues to identify the fake images/videos, which suffer from two fundamental issues: **(1) Generalization: artifact-based detection methods are difficult to generalize to unknown scenarios.** These methods depend highly on the artifacts learned during the training process, so they exhibit poor performance in dealing with unknown and strange artifacts [29]. Besides, Deepfake techniques are developed with an alarming speed, leaving fewer detectable artifacts in their synthesized results [7,22]. These methods are thus struggling to keep up with the development of the Deepfake techniques. **(2) Robustness: artifact-based detection methods are not robust against real-world distortions.** Conventional image manipulations (e.g. cropping, compression) might destroy the artifacts in Deepfake results. These effects would further make the artifact-based detection methods less reliable in such scenarios [17,38]. Besides, the carefully crafted imperceptible adversarial noise in Deepfake images/videos can also significantly reduce the effectiveness of the artifact-based detection [4,12].

To overcome the above problems, we propose a novel framework to proactively watermarks the identity feature of face images and then determine whether these images are Deepfake or not according to the existence of the watermark. The mechanism of our method is similar to anti-counterfeit. Before sharing personal images online, the user can use our method to embed his/her watermark into these images. The watermark acts as the anti-Deepfake label to protect the user's authenticity of these images. Once images with similar identities to the watermarked images appear online, the owner of watermarked images can verify these suspect images' authenticity according to the existence of his/her watermark. More details about the proposed method's real-world application scenario are in Appendix B.

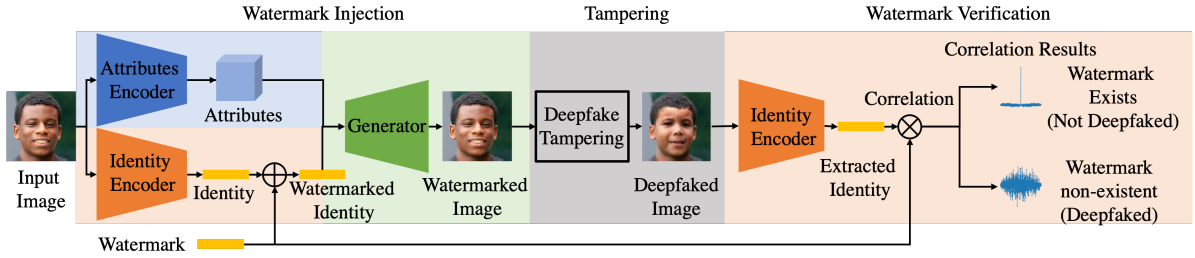

Figure 1. Our method's overall framework consists of watermark injection and watermark verification. The watermark injection step aims to generate watermarked images that are perceptually similar to the original and contain the anti-Deepfake watermark. The watermark verification step aims to verify the existence of the watermark in the image's identity representation to determine whether it is counterfeited or not. The tampering part in the middle represents potential Deepfake manipulations on our watermarked image, which is not our framework's component but is the objective we aim to detect.

As shown in Fig. 1, our proposed framework consists of two major steps: watermark injection and watermark verification. In the watermark injection step, the input face image is first disentangled via two dedicated networks into an identity representation and multi-level attributes representation. Then we embed a pseudo-random sequence into the identity representation to generate the watermarked identity. The embedded sequence has no dependence on the face image, so it can be randomly selected but will be preserved and used for the watermark verification step. Another generative network integrates the watermarked identity with the original attributes to synthesize the watermarked image. This watermarked image is perceptually similar to the original one, excluding the negative impact on the image's normal use from the watermark.

The watermark verification step aims to verify the existence of the watermark in the image to determine whether Deepfake has manipulated it. We employ the same network used in watermark injection to extract the image's identity representation and then calculate its correlation with the preserved watermark to inspect whether the watermark exists. According to the feature of the pseudo-random sequence, if a peak appears in the correlation result, it indicates the watermark is still in the corresponding image, so it has not tampered with Deepfake. Otherwise, it will be determined as a fake one. More details about the proposed method will be explained in the following sections.

In summary, our main contributions are summarized as follows: (1) We propose a novel proactive Deepfake detection method by embedding an anti-counterfeiting watermark into images' identity vectors. (2) We design a simple but effective encoder-decoder network to implement invisible anti-Deepfake watermarking, which requires neither pre-annotation nor pre-detection information. (3) We conduct extensive experiments to evaluate the performance of our method in terms of effectiveness, robustness, utility and security.

2. Related Work

The imperfection of Deepfake techniques inevitably introduces various artifacts to their results, which are the primary clues for existing studies to identify Deepfakes.

McCloskey [32] first utilizes colour distortions to detect fake images. Nataraj *et al.* [33] then propose to identify Deepfake by analyzing the combination of pixel co-occurrence matrices. Chai [5] differentiates fake faces by the redundant spatial artifacts in the image's local patches. The study [25] spots the discrepancies across the blending boundary to distinguish the modified faces. To improve the detection performance, Dang [9] first adopt attention mechanisms on CNN models to detect Deepfake artifacts. Zhao *et al.* [54] reformulate the Deepfake detection as a fine-grained classification task and propose a new multi-attentional architecture to capture local discriminative features from multiple faces attentive regions. Yu *et al.* [50] propose a commonality learning strategy to learn the universal Deepfake features from different databases to better generalize in unknown forgery methods.

Except for the spatial artifacts, the biological signal artifacts are another obvious clue for the forge. Lyu [26] first proposes to spot Deepfake videos by observing the lack of eye blinking in the synthesized face. In [46], inconsistent head poses are employed to reveal forged videos. FakeCatcher [8] combine six different biological signals to distinguish the natural or fake videos. Haliassos [14] targets the inconsistencies in mouth movements learned via lipreading to detect forged videos. Yang *et al.* [45] employs the multi-task learning scheme to extract more comprehensive and accurate lip features to gain more powerful fake discriminability.

Some researchers investigate the frequency artifacts in Deepfake results, which are believed to originate from the architecture of GANs. AutoGAN [53] first observes that the up-sampling design in GAN would introduce artifacts in the synthesized images which can be used to detect GAN-generated images. Other studies [30, 48] then intro-

duce the GAN fingerprints for classifying the authentic or GANs-faked images. [11] further utilize the GAN fingerprints for Deepfake attribution. Recently, to improve the detector's generalization, Luo [28] combine the image's high-frequency features and colour textures to trace the forgery.

Meanwhile, some proactive measures [1, 42, 47] are fighting malicious Deepfake by embedding an invisible tag into the original image, which can remain retrievable after the Deepfake generation process. Then, the user can retrieve the tag and block the dissemination. Yu *et al.* [49] embed artificial fingerprints into the generative model and then to its generated Deepfakes so that they can achieve detection according to the extracted fingerprints. Compared with these works, our method focuses more on semantic level protection, i.e., preventing manipulating face images' identity features.

3. Methodology

Our proposed framework includes two steps: watermark injection and watermark verification. We will introduce the details in this section.

3.1. Watermark Injection

The watermark injection step aims to insert a sequence into a face image to entangle with its identity feature while keeping the watermarked image perceptually similar to the original. The rationale behind this step is that slightly disturbing the identity feature while preserving residual attributes will not significantly distort the face image. Moreover, the conventional image modifications, e.g., cropping, resizing and compression, usually do not impact the identity feature of the facial images. Hence, the embedded watermark can avoid being modified and remain robust against conventional image manipulations. In contrast, Deepfake methods, whose objectives are editing or swapping the image's identity, will inevitably alter the inserted watermark. Therefore, we can utilize this mechanism to detect whether Deepfake modifies a watermarked image or not.

To this end, the watermark injection step consists of three processes: (1) Feature disentanglement, which disentangles the face image as two independent representations, namely identity and attributes; (2) Identity watermarking, which embeds a watermark into the extracted identity representation (vector); (3) Image reconstruction, which integrates the watermarked identity and original attribute to synthesize the corresponding watermarked image. The overview of the watermark injection is illustrated on the left part of Fig. 1, and architecture details are in Appendix C.

Feature Disentanglement: Given an input face image, we employ two dedicated networks, namely identity encoder and attributes encoder, to respectively extract the independent representations, $z_{id}(X)$ and $z_{att}(X)$, from the image.

Identity Encoder: The identity representation is the high-level human biometric feature for characterizing a specific person with lesser intra-personal variations and larger inter-personal differences. Similar to most research for disentangling representations of identity and attributes [35, 44], the identity encoder in our work employs the pre-trained face recognition network [10] as the backbone to extract the input image's last feature vector generated before the final fully-connected layer as identity representation. Specifically, the identity representation is a 512-dimension vector, which is formulated as $z_{id}(X) = Arc(X)$, where X denotes the input image and $Arc(\cdot)$ represents the face recognition network.

Attributes Encoder: The attributes representation of face image is defined as spatial features such as pose, expression, background etc. According to the details of these features, attributes can be divided into different levels, from coarse (e.g., overall spatial outline), to fine (e.g., exact shape). Therefore, we adopt multi-level feature maps to preserve such details to represent the attributes. Specifically, we feed the input image into a U-Net style network and then use the feature maps generated from the U-Net decoder as attribute representations. The formal attributes representation is denoted as:

$$z_{att}(X) = \left\{ z^1_{att}(X), z^2_{att}(X), ..., z^n_{att}(X) \right\}_2, \tag{1}$$

where $z^n_{att}(X)$ represents the n-th level feature map from the U-Net decoder, and n is the number of feature levels. The attributes encoder in this work does not require extra annotations as it extracts the attributes using self-supervised training, which is trained to keep the original image X and generated watermarked image $\hat{X}$ have the representation of the same attribute.

Identity Watermarking: After feature disentanglement, we add a bit-wise binary sequence z_{seq} to the identity representation $z_{id}(X)$ to generate the corresponding watermarked identity. The binary sequence can be user-defined or random-generated, which will serve as a signature for future verification, so the user of our method should preserve his/her embedded sequence and keep it secret from adversaries. Besides, to reduce the watermark's perturbation on the identity representation, we regulate it with a constant weight α. Unless otherwise stated, α will be set to 0.1 in our experiments. Therefore, the final watermark sequence values are minimal compared with the original identity sequence. The identity watermarking is formulated as:

$$z^w_{id}(X) = z_{id}(X) + \alpha z_{seq}, \tag{2}$$

where $z^w_{id}(X)$ represents the watermarked identity vector.

Image Reconstruction: The subsequent process is to integrate watermarked identity and the original attributes to synthesize the watermarked image. Previous studies [2, 34]

revealed that simply concatenating identity and attributes to synthesize images will incur severe visual quality degradation and distortion. To avoid this problem and generate the high-fidelity watermarked image, we employ a novel *Adaptively Attentional Denormalization* (AAD) [24] mechanism to accomplish feature integration.

The image reconstruction network adopts multiple cascaded AAD Residual Blocks (ResBlk) to integrate the identity and attributes. Each AAD ResBlk consists of multiple AAD layers, which employ an attention mechanism with denormalization to adaptively adjust the participation of identity representation and attribute representation for synthesizing different regions. For instance, the identity will provide more importance on generating the facial area, which is most discriminative for distinguishing identities, while the attributes will focus more on the regions related to spatial features, such as skin colour and background.

We formally define the reconstruction procedure as:

$$\hat{X} = Gen(z_{id}^{w}(X), z_{att}(X)) = z_{id}^{w}(X) \oplus z_{att}(X), \quad (3)$$

where the $\oplus$ denote the ADD ResBlk's integration and $Gen(\cdot)$ denote the reconstruction network.

3.2. Watermark Verification

Different from aiming to accurately recover the inserted watermark like the traditional watermark techniques [23,37, 55], the objective of our watermark verification is to detect whether the watermark still exists in the watermarked image's identity feature and, in turn, determine whether Deepfake modifies this image or not. Since the difficulty of watermark detection is much easier than watermark recovery, our method can thus provide more reliable verification results.

In more detail, our watermark verification step consists of two processes: (1) Extraction, which extracts the input image's identity representation; (2) Verification, which calculates the correlation between extracted identity and predefined watermark to verify the existence of watermark in the input image.

Extraction: We re-use the identity encoder adopted in feature disentanglement to extract the identity representation from the watermarked image, which is formulated as $z_{id}(\hat{X}) = Arc(\hat{X})$. The rationale behind this process is that the watermarked identity integrated by our reconstruction process is believed to preserve in the watermarked image's identity feature, so we can use the same identity encoder to extract the corresponding watermarked identity representation. For the watermarked image not modified by Deepfake, the extraction process is defined as:

$$\begin{aligned} z_{id}(\hat{X}) &= Arc(\hat{X}) \\ &= Arc(z_{id}^{w}(X) \oplus z_{att}(X)) \\ &= Arc((z_{id}(X) + \alpha z_{seq}) \oplus z_{att}(X)) \\ &\approx z_{id}(X) + \alpha z_{seq}. \end{aligned} \quad (4)$$

The attributes $z_{att}(X)$ is omitted because $Arc(\cdot)$ only extract identity features.

Verification: After obtaining the identity representation $z_{id}(\hat{X})$, we calculate its correlation with the watermark sequence z_{seq} to verify whether the watermark is present in the input image. Since the $z_{id}(\hat{X})$ and z_{seq} can be regarded as 1-dimensional real discrete sequences, the function computes the correlation of them is defined as:

$$Corr[l] = \sum_{n=0}^{N-1} z_{id}(\hat{X})[n] * z_{seq}[n - l + N - 1], \quad (5)$$

where $l = 0, 1, ..., 2N - 2$ is the index for correlation result,n denotes the index for discrete sequences, N represents the their length, and $z_{seq}[m]$ is 0 when m is outside of the range of z_{seq}.

As demonstrated in Eq. 3, for the real watermarked image, the correlation function in Eq. 4 equals to:

$$\begin{aligned} Corr[l] &= \sum_{n=0}^{N-1} z_{id}(\hat{X})_{rec}[n] * z_{seq}[n - l + N - 1] \\ &\approx \sum_{n=0}^{N-1} (z_{id}(X)[n] + \alpha z_{seq}[n]) * z_{seq}[k] \\ &= \sum_{n=0}^{N-1} z_{id}(X)[n] * z_{seq}[k] + \alpha z_{seq}[n] * z_{seq}[k], \end{aligned} \quad (6)$$

where we set $k = n-l+N-1$ for simplicity. Therefore, the correlation between extracted identity and the pre-defined watermark can be assumed as the sum of two independent calculations: cross-correlation of original identity representation with watermark sequence, and auto-correlation of watermark sequence itself. In contrast, if the watermarked image is modified by Deepfake, its identity representation and entangled watermark sequence will be distorted, so the correlation between its extracted identity and the predefined watermark cannot factorize like Eq. 6 but can only be assumed as two different sequences' cross-correlation like Eq. 4.

According to the auto-correlation's property, the maximum correlation value will appear at the index of $(N-1)$*th*. While for the cross-correlation, there is no such property. Hence, we can detect if there is a distinct peak value at the $(N-1)$*th* index to determine whether the watermarked image's identity feature is tampered with by Deepfake methods.

3.3. Training Procedure

No extra annotations are required in our training procedure, and except for the identity encoder, all other networks are trainable.

Adversarial Loss: To make the reconstructed image more realistic, we employ a multi-scale discriminator $Dis(\cdot)$ from [19] with hinge loss functions to train our model in an adversarial way:

$$\mathcal{L}_{Adv} = \log Dis(X_m) + \log(1 - Dis(\hat{X}_m)), \tag{7}$$

where X_m and $\hat{X}_m$ indicate the low-resolution original and corresponding reconstructed image after m-th downsampling.

Attributes Preservation Loss: We also calculate the attributes representations' $\mathcal{L}_2$ distance between original and reconstructed image to enforce attributes preservation:

$$\mathcal{L}_{Att} = \frac{1}{2}\sum_{k=1}^{n}\left\| z_{att}^{k}(X) - z_{att}^{k}(\hat{X})\right\|_2^2, \tag{8}$$

where the n denotes the level of attributes.

Reconstruction Loss: In addition, to keep the reconstructed image resemble the original and mitigate the conflict with watermark injection at pixel-level, we define a perceptual similarities loss LPIPS [52] between the original and reconstructed image rather than the common pixel-level reconstruction loss:

$$\mathcal{L}_R = \left\| L(X) - L(\hat{X})\right\|_2, \tag{9}$$

where $L(\cdot)$ represents the perceptual features extractor.

Watermark Preservation Loss: To minimise the distortion of embedded watermark sequence in the reconstructed image, a watermark preservation loss function is used to measure the cosine similarity between the watermarked identity vector and extracted identity vector:

$$\mathcal{L}_W = 1 - Cos(\hat{z}_{id}(X) - Arc(\hat{X})), \tag{10}$$

where $Cos(\cdot)$ denotes the operation of cosine similarity.

Our framework is finally trained with a weighted sum of the above losses, which is defined as:

$$\mathcal{L}(X) = \lambda_R\mathcal{L}_R + \lambda_{Adv}\mathcal{L}_{Adv} + \lambda_{Att}\mathcal{L}_{Att} + \lambda_W\mathcal{L}_W, \tag{11}$$

where $\lambda_R, \lambda_{Adv}, \lambda_{Att}, \lambda_W$ are tunable constant weighting corresponded loss. Unless stated otherwise, the λ values are set as $\lambda_R = 10, \lambda_{Adv} = 0.1, \lambda_{Att} = 10$ and $\lambda_W = 1$.

4. Experiment

We conduct extensive experiments to evaluate our method from the following aspects: i) the different pseudo-random sequences' impacts on model performance, ii) effectiveness in Deepfake detection, iii) visual quality of watermarked images, and iv) security in potential attack scenarios. The experiment results demonstrate that our method can achieve the best performance regarding various qualitative and quantitative evaluation metrics. Note that the security analysis results are in Appendix F.

4.1. Experiment Setup

Datasets: We train our method on the **Flickr-Faces-HQ (FFHQ) [21]** dataset, and conduct experiments on **CelebA-HQ [20]** and **CelebA [27]** datasets to reveal its generalizability. Unless stated otherwise, all images in the experiment have been aligned and cropped to the size of 256×256.

Baselines: We select the work whose authors released the source codes and pre-trained models in our comparison experiment for results reproducible.

Deepfake Detection: Passive methods [3, 5, 15, 41] and proactive method [49] are selected because they represent the latest reproducible Deepfake detection methods.

Digital Watermarking: We chose StegaStamp [40] and UDH [51] as the baseline because they achieve the SOTA performance in embedding information and exhibit appealing visual quality results.

We use the official codes and pre-trained models for all the above-mentioned methods.

Evaluation Metrics: We evaluate the performance using three different categories of metrics: (1) For both Deepfake detection, to measure the miss detection rate and false alarm rate, we compute image-level **Accuracy(ACC)** and **F1-Score**. **AUC** and related **ROC** curve, which are decision-threshold-free metrics, are also reported to select optimal models; (2) Regarding robustness evaluation, the proportion of correctly detected watermarked images after various post-processing is calculated and denoted as **Detection Ratio (DR)**; (3) **Peak-Signal-to-Noise Ratio (PSNR)** and **Structural Similarity Index Measure (SSIM)** are used to calculate the similarity between the watermarked images and original images to show the visual quality of watermarked images.

The above metrics are higher when the associated methods show better performance, except stated otherwise.

4.2. Impacts of the Types of Sequences

According to Eq. 4, the correlation property of the embedded sequence plays a significant role in the watermark verification. Thus, we first analyse the impact of the different watermark sequences on our method. Two representative Pseudorandom Noise (PN) sequences: Maximum-Length Sequence (MLS) and Gold Code (Gold), and two most common random sequences: Gaussian noise and Laplace noise, are selected for comparison. All embedded sequences are set to a length of 512, the same as the identity representation. For a fair comparison, the network is trained FFHQ images by randomly selecting the above watermarks in each iteration. Then, we apply four different sequences to the 10k randomly chosen Celeba-HQ images to generate

Table 1. Different sequences' correlation results and corresponding watermarked images' visual quality.

Sequences types	Correlation results			Visual Quality	
	Peak	Average	PAR ↑	SSIM ↑	PSNR ↑
Original	0.77	0.54	1.43	1.0	48.0
Gaussian	0.96	0.54	1.79	0.95	34.65
Gold	5.61	0.52	**10.83**	0.94	33.32
Laplace	1.82	0.74	2.45	0.95	34.84
MLS	4.82	0.53	9.17	0.95	33.5

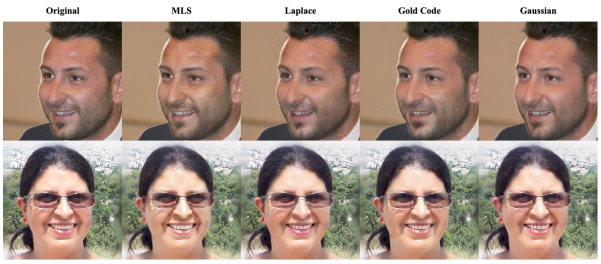

Figure 2. Qualitative comparison of different sequences' watermarked images. Despite injecting different sequences, all watermarked images are perceptually identical to the original images.

watermarked testing images, resulting in 40k testing images and calculate the defined metrics over these images to evaluate the impact.

Correlation Results. In Table 1, we present the averaged correlation results, where ***Peak*** denotes the correlation value appear at the zero-lag, ***Average*** represents the mean of the residual correlation results, and ***Peak-to-Average Ratio (PAR)*** outputs a ratio of ***Peak*** over ***Average***, which indicates how significant the ***Peak*** stands out in the correlation results. Except for the results from watermarked images, we also report the correlation of original images in Table 1's first row, which acts as a reference. As shown in Table 1, the watermarked images' Peak and PAR are significantly higher than the reference, where the Gold sequence achieves the highest values. The apparent difference between watermarked and non-watermarked images' correlation results demonstrates that our method can effectively embed and extract the watermark in images.

Visual Quality. Afterwards, we evaluate the visual quality of images after watermarking different sequences. The best SSIM and PSNR values are also reported in Table 1's Original row for reference. According to the quantitative and qualitative results exhibited in Table 1 and Fig. 2, no matter what types of sequence are embedded, the watermarked images can maintain high visual quality and look perceptually identical to the original, which demonstrates that watermarking images via our method would not affect its utility.

Effectiveness. In this section, we explore the effectiveness of our method in identifying the watermarked or non-watermarked images. Our method discriminates 10k watermarked images and 10k randomly chosen Celeba-HQ non-

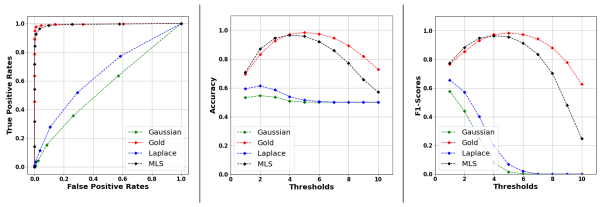

Figure 3. **ROC** ↑, **Accuracy** ↑ and **F1 Score** ↑ of different sequences under different PAR thresholds. Gold and MLS sequences' performance to discriminate between watermarked or non-watermarked images are superior to Gaussian and Laplace.

watermarked images for different types of watermark sequences. Besides, according to the correlation results summarized in Table 1, different types of watermark sequences have different PARs. Here, we consider adopting different PAR values ranging from 1 to 10 with a step size of 1 as the threshold to decide whether a watermark exists in the corresponding image. More specifically, an image with PAR higher than the threshold in its correlation results will be regarded as watermarked. We calculate related ACC and F1 Scores further to analyze our method's discriminability under different PAR thresholds, and also plot the ROC curve and compute corresponding AUC to provide more convincing results.

Fig. 3 displays the experiment results. According to the trend of ACC and F1 Score curves, the optimal thresholds for different sequences are 2 for Gaussian and Laplace, 5 for Gold and 4 for MLS. We will adopt these thresholds in the subsequent robustness evaluations. Besides, their AUC values are 0.5552, 0.9952, 0.6466 and 0.9917 for Gaussian, Gold, Laplace and MLS, respectively. The ROC curve of Gold and MLS are much closer to the top left than Gaussian and Laplace. These results indicate that adopting Gold and MLS as watermark sequences would perform our method better than Gaussian and Laplace.

Robustness. We test the impact of different sequences on our method's robustness. Five common post-processing operations are adopted in the experiment, i.e., Gaussian blurring, colour adjustment, JPEG, horizontal flipping, resizing and cropping. For Gaussian blurring, we consider kernel standard deviation ranging from 0.5 to 1.0 with a step size of 0.1. For JPEG, we consider quality factors ranging from 50 to 100 with a step size of 10. For resizing and cropping, we consider first cropping the image's peripheral sizes ranging from 50% to 100% with a step size of 10% and then resizing it to 256×256. For Horizontal flipping and colour adjustment, we employ the PyTorch torchvision.transformations' functions RandomHorizontalFlip with the probability of the image being flipped set as 1.0 and ColorJitter with the default setting to achieve all image's horizontal flipping and randomly brightness, contrast, saturation and hue change. Examples of the modification are visualized in Fig. 4.

Table 2. **Detection ratio** ↑ of different sequences against color adjustment and horizontal flipping.

Image Manipulations	Sequences types			
	Gaussian	Gold	Laplace	MLS
ColorJitter	63.9%	**94.8%**	45.4%	91.2%
Flip	61.9%	**97.2%**	52.4%	96.7%

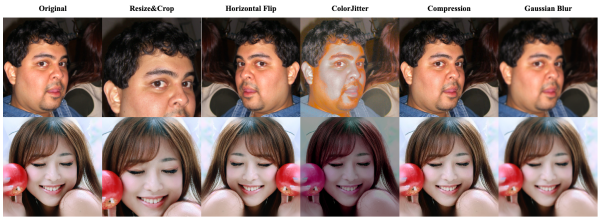

Figure 4. Samples of each post-processing operation results adopted in our experiment.

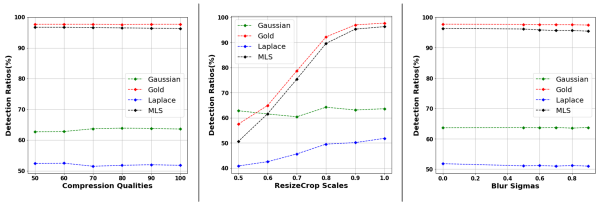

Figure 5. **Detection ratio** ↑ of different sequences against JPEG, Resize-Crop and Gaussian Blur.

For each sequence's watermarked images, we apply the above operations to generate corresponding images and then employ our method to detect watermarks from these processed images and compute the detection ratio **DR**. Table 2 and Fig. 5 present each sequence's robustness performance. Our method shows a minor performance degradation when dealing with compression and blur but is susceptible to resizing and cropping. As illustrated in Fig. 4 row1 column 2, the main reason is that a crop size smaller than 80% would cut off partial facial regions, damaging the corresponding identity feature. However, this problem is not severe because a cropped face image is unlikely to be used in practice.

The results of Gold and MLS watermark sequences reflect our method's robustness against these image post-processing, where the Gold sequence achieves the best performance, slightly superior to MLS but much better than Gaussian and Laplace. Therefore, we will adopt the Gold sequence as the watermark to compare our method with other works in Deepfake detection.

4.3. Deepfake Detection

We compare our method with other Deepfake detection approaches in image-level real or fake classification. Two attributes manipulation methods, i.e., AttGAN [16] and StarGAN2 [7], two identity swap methods, i.e., InfoSwap [13] and SimSwap [6], and two face anonymization approaches, namely CIAGAN [31] and DeepPrivacy [18] are

Figure 6. Samples of non-watermarked and our watermarked images' Deepfake results. The watermarked image's forgery result is perceptually identical to the non-watermark image's.

employed in this experiment. We adopt the official codes and pre-trained models of these works, so our experiment results are reliable and reproducible, which thus can refer for future comparison.

Celeba and CelebaHQ images are employed in this experiment to represent low- and high-resolution Deepfake cases. We apply Deepfake methods to watermarked and non-watermarked images to generate corresponding fake outputs. The watermarked real and fake images are utilised to evaluate our method's discriminability, while the non-watermarked real and fake images are adopted to evaluate other detection methods' performance. According to the analysis of different sequences' performance, our method adopts the Gold sequence as the embedded watermark in the comparison experiment and sets the PAR threshold to 5.

We first illustrated Fig. 6 to have a qualitative comparison of Deepfaked non-watermarked and watermarked outputs where we can see that the non-watermarked and our watermarked images' Deepfake results are perceptually identical to each other, demonstrating that our method well maintains the utility of the image. Moreover, it also makes it hard for the Deepfake adversaries to distinguish the protected and non-protected images, increasing our method's secrecy.

Table 3 summarizes the comparison results between ours with passive methods. Our method achieves more than 0.8 ACC and F1 Scores on detecting all Deepfake methods' outputs, revealing its superior effectiveness and generalization. Except PF perform better than ours on StarGAN2, our method outperforms all other baselines with a clear margin. Our method performs poorly on StarGAN2 (still achieves second-rank performance) because StarGAN2 does not modify face identity-related features. To verify this, we employ AttGAN to manipulate identity-related attributes, e.g., gender and skin colour. The result in Table 3 shows that our method can accurately detect these manipulated images. On the contrary, other passive detection methods only perform well in detecting limited Deepfake methods, deteriorating to random guesses (50% accuracy) in detecting other Deepfake methods.

Then, we compare our method with the latest proactive detect method, i.e., AGF [49], to detect FaceShifter's Deepfake outputs on Celeba-HQ images and present the

Table 3. **Accuracy** ↑ and **F1 Scores** ↑ of different methods' DeepFake detection results.

Detection methods	Low Resolutions(Celeba)						High Resolutions(Celeba-HQ)					
	AttGAN	CIAGAN	DeepPrivacy	InfoSwap	SimSwap	StarGAN2	AttGAN	CIAGAN	DeepPrivacy	InfoSwap	SimSwap	StarGAN2
BTS [15]	0.86/0.87	0.51/0.66	0.5/0.66	0.49/0.66	0.51/0.67	0.53/0.67	0.86/0.87	0.5/0.66	0.5/0.66	0.49/0.65	0.5/0.66	0.55/0.68
CD [43]	0.88/0.86	0.51/0.03	0.51/0.01	0.54/0.17	0.51/0.01	0.78/0.71	0.81/0.77	0.51/0.04	0.52/0.07	0.52/0.07	0.52/0.06	0.84/0.81
ICPR [3]	0.59/0.69	0.62/0.62	0.58/0.68	0.49/0.64	0.6/0.71	0.46/0.63	0.53/0.68	0.65/0.62	0.56/0.69	0.51/0.66	0.55/0.69	0.49/0.65
PF [5]	0.76/0.79	0.51/0.66	0.52/0.65	0.57/0.68	0.54/0.67	**0.99/0.98**	0.75/0.79	0.51/0.66	0.55/0.68	0.56/0.69	0.54/0.68	**0.98/0.97**
RFM [41]	0.5/0.67	0.51/0.67	0.51/0.67	0.5/0.67	0.51/0.67	0.5/0.67	0.5/0.67	0.5/0.67	0.51/0.67	0.5/0.67	0.5/0.67	0.5/0.67
SBI [39]	0.79/0.82	0.77/0.8	0.78/0.82	0.77/0.81	0.78/0.81	0.72/0.75	0.8/0.8	0.72/0.78	0.78/0.8	0.76/0.77	0.83/0.84	0.69/0.7
Ours	**0.94/0.94**	**0.87/0.86**	**0.98/0.98**	**0.98/0.98**	**0.97/0.98**	0.82/0.84	**0.94/0.94**	**0.85/0.82**	**0.99/0.98**	**0.99/0.99**	**0.98/0.98**	0.85/0.87

Table 4. DeepFake detection performance of proactive methods.

	Acc ↑	F1 Scores ↑	Precision ↑	Recall ↑
AGF [49]	0.7179	0.9148	0.8444	0.9980
Ours	0.9955	0.9955	0.9980	0.9930

Table 5. **AUC** ↑ of our method on different Deepfake and datasets.

Datasets	DeepFake Methods					
	AttGAN	CIAGAN	DeepPrivacy	InfoSwap	SimSwap	StarGAN22
Celeba	0.98	0.95	0.99	0.99	0.98	0.98
CelebaHQ	0.98	0.95	0.99	0.99	0.98	0.98

detection results in Table 4. The details about the experiment setting are provided in Appendix G. Although AGF achieves impressive detection performance, with 0.99 Recall and 0.91 F1 Score, our method still beats it on almost all metrics (only 0.005 lower Recall which is negligible). In particular, our method has a significant advantage in detection accuracy due to AGF producing more false alarms on authentic images (only 0.84 Precision).

We also report the AUC of our method on different Deepfakes and datasets in Table 5 and plot the ROC, accuracy and F1 Scores curves when adopting different thresholds in Appendix H . These results reveal our method's exceptional image level real or fake classification capability when facing different Deepfake methods. In general, the experiment results demonstrate that our method has superior Deepfake detection performance to existing methods.

4.4. Digital Watermarking

We compare our method with the AGF and SOTA digital watermarking techniques, namely Stegastamp and UDH, in the visual quality of watermarked images to show that our method does not sacrifice the normal utility of images. Here, we adopt the widely used metrics PSNR and SSIM to quantitatively reflect comparison results in Table 6. The results demonstrate that the outputs of our method have much better visual quality than others. Fig. 7 also illustrates perceptual comparison, where our watermarked images are more visual-realistic which accurately preserves the hue and light of the original images. In contrast, UDH introduces apparent artifacts in its watermarked images. StegaStamp's outputs have noticeable colour distortion in the facial area. Therefore, qualitative and quantitative results indicate that our method can generate high-quality images with a robust watermark.

Table 6. Quality of different watermarking methods' outputs.

Quality metrics	Proactive Detection		Deep Hiding	
	Ours	AGF	UDH	StegaStamp
SSIM ↑	**0.94**	0.91	0.69	0.89
PSNR ↑	**33.32**	30.69	20.39	29.77

Figure 7. Qualitative comparison between our method, AGF and SOTA watermarking techniques StegaStamp and UDH.

4.5. Limitations

First, our method requires pre-processing, which will introduce computational overhead (in Appendix E) and cannot perform detection of already synthesized images. Therefore, we think the best application case of our method is to protect critical images and employ our method before spreading them over the social network. Second, current watermarking is embedded in the identity features of face images, as we believe it presents the most severe threat if a person's identity is faked. Our method needs further improvement to adapt to other types of Deepfake content.

5. Conclusion

This work poses a proactive method to protect face images from malicious Deepfake. By embedding an invisible watermark into the face image's identity, our method provides users with a reliable approach to verifying their image's authenticity, reducing the negative impact of Deepfake forgery. The experiment results have demonstrated our method's superior performance in identifying Deepfake, preserving reconstructed images' visual quality, retaining watermarked sequence robustness, and resilience to potential malicious attacks.

Acknowledge. This research is funded in part by ARC-Linkage grant (LP180101150 to TZ and BL), ARC-Discovery grant (DP220100800 to XY) and ARC-DECRA grant (DE230100477 to XY). We thank all anonymous reviewers and ACs for their constructive suggestions.

References

[1] Vishal Asnani, Xi Yin, Tal Hassner, Sijia Liu, and Xiaoming Liu. Proactive image manipulation detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15386–15395, 2022.

[2] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Towards open-set identity preserving face synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6713–6722, 2018.

[3] Nicolo Bonettini, Edoardo Daniele Cannas, Sara Mandelli, Luca Bondi, Paolo Bestagini, and Stefano Tubaro. Video face manipulation detection through ensemble of cnns. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5012–5019, 2021.

[4] Nicholas Carlini and Hany Farid. Evading deepfake-image detectors with white-and black-box attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 658–659, 2020.

[5] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *European Conference on Computer Vision*, pages 103–120. Springer, 2020.

[6] Renwang Chen, Xuanhong Chen, Bingbing Ni, and Yanhao Ge. Simswap: An efficient framework for high fidelity face swapping. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2003–2011, 2020.

[7] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020.

[8] Umur Aybars Ciftci, Ilke Demir, and Lijun Yin. Fakecatcher: Detection of synthetic portrait videos using biological signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[9] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K Jain. On the detection of digital face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, pages 5781–5790, 2020.

[10] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[11] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7890–7899, 2020.

[12] Apurva Gandhi and Shomik Jain. Adversarial perturbations fool deepfake detectors. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[13] Gege Gao, Huaibo Huang, Chaoyou Fu, Zhaoyang Li, and Ran He. Information bottleneck disentanglement for identity swapping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3404–3413, 2021.

[14] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Lips don't lie: A generalisable and robust approach to face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5039–5049, 2021.

[15] Yang He, Ning Yu, Margret Keuper, and Mario Fritz. Beyond the spectrum: Detecting deepfakes via re-synthesis. *arXiv preprint arXiv:2105.14376*, 2021.

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

[17] Yihao Huang, Felix Juefei-Xu, Run Wang, Qing Guo, Lei Ma, Xiaofei Xie, Jianwen Li, Weikai Miao, Yang Liu, and Geguang Pu. Fakepolisher: Making deepfakes more detection-evasive by shallow reconstruction. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1217–1226, 2020.

[18] Håkon Hukkelås, Rudolf Mester, and Frank Lindseth. Deepprivacy: A generative adversarial network for face anonymization. In *International Symposium on Visual Computing*, pages 565–578, 2019.

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

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

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[23] S Katzenbeisser and FAP Petitcolas. Digital watermarking. *Artech House, London*, 2, 2000.

[24] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Faceshifter: Towards high fidelity and occlusion aware face swapping. *arXiv preprint arXiv:1912.13457*, 2019.

[25] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5001–5010, 2020.

[26] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7. IEEE, 2018.

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

[28] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16317–16326, 2021.

[29] Siwei Lyu. Deepfake detection: Current challenges and next steps. In *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2020.

[30] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do gans leave artificial fingerprints? In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 506–511. IEEE, 2019.

[31] Maxim Maximov, Ismail Elezi, and Laura Leal-Taixe. Ciagan: Conditional identity anonymization generative adversarial networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5447–5456, 2020.

[32] Scott McCloskey and Michael Albright. Detecting gan-generated imagery using color cues. *arXiv preprint arXiv:1812.08247*, 2018.

[33] Lakshmanan Nataraj, Tajuddin Manhar Mohammed, BS Manjunath, Shivkumar Chandrasekaran, Arjuna Flenner, Jawadul H Bappy, and Amit K Roy-Chowdhury. Detecting gan generated fake images using co-occurrence matrices. *Electronic Imaging*, 2019(5):532–1, 2019.

[34] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. Rsgan: face swapping and editing using face and hair representation in latent spaces. *arXiv preprint arXiv:1804.03447*, 2018.

[35] Yotam Nitzan, Amit Bermano, Yangyan Li, and Daniel Cohen-Or. Face identity disentanglement via latent space mapping. *arXiv preprint arXiv:2005.07728*, 2020.

[36] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019.

[37] Christine I Podilchuk and Edward J Delp. Digital watermarking: algorithms and applications. *IEEE signal processing Magazine*, 18(4):33–46, 2001.

[38] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European Conference on Computer Vision*, pages 86–103. Springer, 2020.

[39] Kaede Shiohara and Toshihiko Yamasaki. Detecting deepfakes with self-blended images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18720–18729, 2022.

[40] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2126, 2020.

[41] Chengrui Wang and Weihong Deng. Representative forgery mining for fake face detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14923–14932, 2021.

[42] Run Wang, Felix Juefei-Xu, Meng Luo, Yang Liu, and Lina Wang. Faketagger: Robust safeguards against deepfake dissemination via provenance tracking. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3546–3555, 2021.

[43] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8695–8704, 2020.

[44] Yunqian Wen, Li Song, Bo Liu, Ming Ding, and Rong Xie. Identitydp: Differential private identification protection for face images. *arXiv preprint arXiv:2103.01745*, 2021.

[45] Chen-Zhao Yang, Jun Ma, Shilin Wang, and Alan Wee-Chung Liew. Preventing deepfake attacks on speaker authentication by dynamic lip movement analysis. *IEEE Transactions on Information Forensics and Security*, 16:1841–1854, 2020.

[46] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265. IEEE, 2019.

[47] Yuankun Yang, Chenyue Liang, Hongyu He, Xiaoyu Cao, and Neil Zhenqiang Gong. Faceguard: Proactive deepfake detection. *arXiv preprint arXiv:2109.05673*, 2021.

[48] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7556–7566, 2019.

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

[50] Peipeng Yu, Jianwei Fei, Zhihua Xia, Zhili Zhou, and Jian Weng. Improving generalization by commonality learning in face forgery detection. *IEEE Transactions on Information Forensics and Security*, 2022.

[51] Chaoning Zhang, Philipp Benz, Adil Karjauv, Geng Sun, and In So Kweon. Udh: Universal deep hiding for steganography, watermarking, and light field messaging. In *34th Conference on Neural Information Processing Systems, NeurIPS 2020*, volume 33, pages 10223–10234, 2020.

[52] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[53] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. In *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2019.

[54] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2185–2194, 2021.

[55] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 657–672, 2018.