Making DeepFakes More Spurious: Evading Deep Face Forgery Detection via Trace Removal Attack

Chi Liu, Huajie Chen, Tianqing Zhu, Jun Zhang, Senior Member, IEEE, and Wanlei Zhou, Senior Member, IEEE

Abstract—DeepFakes are raising significant social concerns. Although various DeepFake detectors have been developed as countermeasures, their vulnerability under attacks remains further explorations. Recently, several attacks, such as adversarial attacks, have successfully fooled DeepFake detectors. However, existing attacks suffer from detector-specific designs, requiring detector-side knowledge, leading to poor transferability. Moreover, they only consider simplified security scenarios; but less is known about the attacking performance in complex scenarios where the capability of detectors or attackers varies. To fill the gap, we propose a novel, detector-agnostic trace removal attack. The attack removes all possible counterfeiting traces arising from the original DeepFake manufacture procedure to make DeepFakes essentially more “realistic” and thus able to defeat arbitrary or unknown detectors. Concretely, we first perform an in-depth DeepFake trace discovery, identifying three intrinsic traces: spatial anomalies, spectral disparities, and noise fingerprints. Then an adversarial learning-based trace removal network (TR-Net) involving one generator and multiple discriminators is proposed. Each discriminator is responsible for one individual trace representation to avoid inner-trace interference. All discriminators are optimized in parallel to enforce the generator to remove various traces simultaneously. We additionally craft heterogeneous security scenarios where the detectors are embedded with different levels of defense and the attackers own varying background data knowledge. The experimental results show that the proposed trace removal attack can significantly compromise the detection accuracy of six state-of-the-art DeepFake detectors while causing only a negligible degradation in visual quality.

Index Terms—Adversarial attack, anti-forensics, deepfake detection, image forgery.

I INTRODUCTION

LONG with the recent progress in automated digital face manipulation techniques based on deep learning, deep face forgeries, also known as “DeepFakes”, are raising serious security concerns [1]. Accordingly, the research community is dedicated to developing forensic countermeasures against DeepFakes, and many DeepFake detectors have been developed to distinguish DeepFake images from real ones [2]. However, the robustness of these detectors against malicious attacks is still in the early stages. To further expose, analyze and understand the vulnerability of DeepFake detectors, researchers have and must continue to engage in developing anti-forensic attacks against DeepFake detectors [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13].

Most existing attacks are based on adversarial attacks that embed imperceptible adversarial perturbations into DeepFake samples to fool machine learning-based detectors [3], [4], [5], [6], [7], [8], [9]. The development of this type of attack requires background knowledge of the detectors, such as the queried outputs and the network parameters. Even in a black-box attack scenario, surrogate detectors are always needed to approximate the knowledge of the target detector. These detector-specific attack designs may lead to poor attack transferability across different detectors and instability against unknown detectors [14], [15]. Another emerging type of attack generally requires reconstructing DeepFake samples to modify the distribution of detectable features the target detector may be interested in [10], [11], [12], [13]. This is also a detector-specific design, which results in attacks being less transferable to detectors focusing on different features. Moreover, these reconstruction attacks mainly pay attention to a single type of feature, and their efficacy may deteriorate significantly against advanced detectors that operate on hybrid features.

In addition, another weakness of existing attacks is that they only got evaluated in oversimplified security scenarios: On the one hand, the attacks are often implemented and evaluated with ideal assumptions, e.g., “the attacker has unlimited access to the target detector” or “the attacker knows the training dataset of the target detector”. On the other hand, the target detectors are often assumed to be as naked as possible, while some common and easy-to-implement defenses are left out of consideration [16].

To address the above weaknesses, in this paper, we propose a novel attack method for DeepFake anti-forensics, called the trace removal attack. Unlike the detector-specific designs, we offer a novel detector-agnostic perspective. As shown in Fig. 1, we pay full attention to the original DeepFake creation pipeline, identifying the manufacturing traces naturally contained in
We performed an in-depth DeepFake trace discovery, identifying three universal traces responsible for DeepFake images’ tractability. We propose a novel attack concept against DeepFake detectors, namely, the trace removal attack. Benefiting from a detector-agnostic design, our attack can defeat arbitrary unknown detectors and detectors equipped with defenses. The attack is implemented via a “one-versus-multiple” adversarial learning network that erases all traces synchronously.

The attack is tested in heterogeneous threat scenarios, where the detector’s defensive capability ranges from weak to strong and the attacker’s data knowledge is limited. Furthermore, performance is evaluated on a wide range of detectors, and a dataset is developed covering all typical DeepFake types to benchmark our evaluation.

II Preliminary and Related Work

A Generative Adversarial Net

The generative adversarial net (GAN) [17] and its variants are a typical type of deep generative models that serve as a key technique for generating DeepFakes [2]. A GAN normally involves a generator $G$ and a discriminator $D$ in an adversarial learning framework where the aim is to train $G$ to generate synthetic images within the target distribution $p_t(x)$. The process can be described as:

$$\min_G \max_D T(G, D) = \mathbb{E}_{p_t(x)}[\log D(x)]$$

$$+ \mathbb{E}_{p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

If an additional control is imposed over the modes of the data to be generated, such as adding an attribute label $y$ as prior guidance, a conditional GAN will be created then.

B DeepFake

DeepFakes are generated in roughly one of three ways: face synthesis, facial attribute editing, or face replacement [2]. Face synthesis means creating an entire non-existent face from random noise with an unconditional GAN, such as ProGAN [18] and StyleGAN [19]. With facial attribute editing, an image’s attributes are altered. Either the appearance attributes (e.g., hair color, makeup, skin color, etc.) or the soft biometric attributes (e.g., identity, gender, age, etc.) can be modified. Conditional GANs, such as StarGAN [20] and STGAN [21], are widely employed for such tasks. Here, the target attribute serves as the extra label $y$ in training a conditional GAN. Face replacement swaps the face of a target image with that of a source image. Factors that need to be considered include the alignment of the face in terms of size, pose, and direction. A deep rendering process then ensures the resulting image looks natural and seamless. In addition, these methods can be combined to produce high-level DeepFakes, like high-fidelity facial reenactments for fake videos.

C DeepFake Forensics

DeepFake forensics is a discipline devoted to detect whether or not an image is a DeepFake image. There are three main categories of detectors: spatial detectors, frequency detectors, and fingerprint detectors.

Spatial detectors learn discriminative features directly from the spatial information of the images, i.e., the pixel inputs. The...
available features range from low-level cues to deep representational features. For example, some researchers classify DeepFakes versus real images based on disparities in their color components [22], [23], [24], global image textures [25], or the consistency of the facial context [26]. Others train more generalized detectors by simulating the shared visual artifacts in DeepFakes led by face manipulation [27] or blending [26]. Chai et al. [28] investigated the semantic properties that make DeepFakes detectable at the image patch level. There are also several studies [16], [29], [30] that investigate the power of deep convolutional neural networks (CNNs), such as Xception [31] and ResNet [32], in DeepFake detection from a data-driven perspective.

**Frequency-based detectors** solve DeepFake detection using frequency cues. These detectors normally mine distinctive features from the image spectra via statistical analysis or machine learning. For instance, Durall et al. [33], Dzanic et al. [34] and Frank et al. [35] determined that there were Fourier spectrum discrepancies between CNN-generated images and real images which could be efficiently captured by a shallow machine learning classifier. Liu et al. [36] highlight that these discrepancies are more significant in phase spectra than in amplitude spectra, which is helpful for DeepFake detection. Zhang et al. [37] identify unknown DeepFakes by detecting and simulating general spectrum artifacts. Qian et al. [38] mine frequency-aware clues from both frequency-aware decomposed image components and local frequency statistics. Some more recent studies [39], [40] have put forward hybrid frameworks that combine both spatial and frequency information to generalize DeepFake detection.

**Fingerprint Detectors.** Similar to cameras leaving their device fingerprints on a photo, researchers have discovered that GANs will leave unique and stable fingerprints in their generated images, which can be exploited to differentiate GAN-generated DeepFakes from real images. For example, Marra et al. [41] estimate GAN fingerprints as average noise residuals for GAN-generated image forensics. Yu et al. [42] use a CNN to learn GAN fingerprint representations as a way of attributing DeepFakes to their source GANs. Yang et al. [43] devised a method to disentangle content-irrelevant fingerprints from GAN-generated images to determine DeepFakes. Yu et al. [44] proposed introducing artificial fingerprints into source models that produce DeepFakes for proactive and sustainable detection.

**D Anti-Forensics for DeepFakes**

**Adversarial Attack.** Since most forgery detectors are machine learning models, adversarial attacks, as a typical type of attack against machine learning classifiers, have become a common anti-forensic choice for DeepFakes. A successful adversarial attack requires embedding imperceptible noise perturbations into the fake sample, which deceives the detector into classifying the image as “real”. Several classic adversarial attack methods, including Fast Gradient Sign Method (FGSM) [45], iterative FGSM [46], Carlini and Wagner l2-norm Attack [47], DeepFool [48] and Projected Gradient Descent (PGD) [49], are explored to expose the vulnerability of DeepFake detectors in both white- and black-box scenarios [3], [4], [5], [6], [7], [14], [15]. Liao et al. [8] improved on the efficiency of these attacks by adding perturbations to key regions of the DeepFakes instead of across the entire image. Wang et al. [9] point out that adding adversarial noise to a transformed color space will ameliorate the perceptual degradation in a perturbed DeepFake image. Huang et al. [10] proposed an adversarial-noise-guided filtering method that retouches DeepFake images to evade detection.

**Reconstruction-Based Attack.** Very recently, some other attacks were proposed based on reconstructing DeepFake samples instead of injecting external noise to the samples for evading forensic detection. Ding et al. [11] proposed an adversarial learning method that re-synthesizes face-swapping images with narrowing down the distribution gap between real and fake faces. Huang et al. [13] proposed FakePolisher, a shallow reconstruction model based on dictionary learning. The model projects DeepFake images onto the manifold subspace learned from real images to reduce spectral discrepancies between the two. Neves et al. [12] proposed GANprintR, a deep convolutional autoencoder that learns the reconstruction-related representation from natural images. Then the potential traceable fingerprints in the DeepFake image can be removed by feeding the image to the learned autoencoder.

**III EMPIRICAL DEEPEFAKE TRACE DISCOVERY**

In this section, we investigate the original DeepFake creation pipeline to seek out the root causes that make DeepFakes detectable, by which the natural DeepFake traces can be identified empirically.

**A The DeepFake Pipeline**

The top-level design of a DeepFake generator may vary, but underneath there is a common pipeline for producing DeepFake images that consists of three core stages: face extraction, fake face creation, and deep rendering, as shown in Fig. 2. In the first stage, the facial region is localized and extracted from the source image, which can be accomplished without a GAN. Next, a fake face is generated by a specific GAN or a model template according to the target face. In most cases, creating a fake face is conditioned upon some knowledge of the target face, such as the identity or attributes. In the last stage, the generated face is aligned to the source face and composed back onto the source image, where GANs or some post-processing operations are employed for semblance blending or rendering.
This pipeline applies to all DeepFake types outlined in Section II.B, partially or entirely. For example, end-to-end face synthesis and facial attribute editing models can be seen as interim productions resulting from the second stage of the pipeline, whereas more sophisticated DeepFake models equipped with a post-rendering process are represented by the entire pipeline.

B Model Traces in Deep Face Creation

Spatial Anomalies. DeepFakes rely on deep generative models such as GANs to synthesize fake faces. Ideally, the generated faces should be visually indistinguishable from real ones. However, due to some practical limitations with the training dataset or the model’s capability, the fake faces may be imperfect which may show spatial abnormalities. Although the latest GANs significantly improve on visual quality over their predecessors, some subtle unnatural traces such as textural inconsistencies and contextual discrepancies can still occur [22], [23], [24], [25], [26]. Spatial anomalies can also be found in faces generated by model templates, due to the manufacturing failures during the template alignment or rendering [27], [50].

Since the subtle spatial anomalies are normally imperceptible to humans but can be captured by machines, we demonstrate their existence and spatial distributions in the RGB color space with the spatial attention map (SAM) of a toy Xception detector. Grad-CAM [51] is used to calculate the SAMs regarding different DeepFake types (details of these DeepFake types and the Xception detector are introduced in Section V). As shown in Fig. 3, there are evident detectable traces in the RGB space, and their distributions exhibit a certain level of common, stable semantic dependency: the ProGAN and STGAN’s anomalies are around the middle-right face region, while DeepfakeTIMIT images expose traces concentrated in the nose area. These results can be seen more clearly in the averaged faces.

Spectral Disparity. Detectable traces can also be revealed in the frequency domain. This is because that CNN-based generative models, typically GANs, will lead to disparities in the spectra of the generated images. Previous studies attributed this phenomenon to the transposed convolution operation, a widely-used upsampling unit in CNN-based generative models for increasing feature dimensionality [33], [35], [36], [37], [52]. We have the following claim:

Claim (1). The transposed convolution operation in upsampling layers leads to quasi-periodic high-frequency artifacts in the resulting feature maps. (The proof is in the Supplement, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TDSC.2023.3241604.)

Fig. 4 illustrates the average spectral disparity between real and different types of DeepFake images. All types of DeepFakes have significant differences from the real images. The disparity patterns in the DeepfakeTIMIT images differs from the other two because only DeepfakeTIMIT involves a post-processing procedure that further change the spectral distribution.

C Model Traces in Deep Rendering

Noise Fingerprint. Two types of manufacturing fingerprints are possibly retained in deep rendering, including GAN fingerprint and post-processing fingerprint, as GAN-based rendering models and some post-processing operations are typically involved in this phase. It has been proved that GANs will maintain unique and stable fingerprints in their generated images [41], [42]. Likewise, post-processing operations will also introduce fingerprints, a form of the characteristic discrepancies in the noise space brought about by image editing. To verify the existence of noise fingerprints, we estimate the fingerprints of a DeepFake model $M$ using the average noise residual, which can be formulated as:

$$F^M = \frac{1}{N} \sum_{i=1}^{N} (I_i^M - W(I_i^M)),$$

where $I_i^M$ is a sample generated by $M$, $W(\cdot)$ is a Wiener denoising filter and $N = 2000$ in our case. For reference, we also used (2) to calculate the average noise residual of an equal
shows the results. For RACE denote the inner product and ·[42]

< F, A > is also trained with different types of

The attacker is re-

>−is strengthened during its training

for applying each individual

This method augments training

R M will be misled into classifying the attack samples

and can use

C We assume the target victim model is an

This method assumed that the

had full knowledge of the attack model and

C} >

The attack is required to

traces distinguishes DeepFake images from real ones. Hence,

DF images, which corroborates the fact that extra noise
discrepancies are introduced into DeepFakes during their pro-
duction. We further demonstrate the distributional differences
between real and fake images by the histogram of normalized

cross-correlation (NCC) scores. The NCC score indicates how
closely the average fingerprint correlates with individual noise
residuals extracted from another 2000 real/fake samples, defined as:

\[ \rho^M_i = \frac{< F^M, R^M_i >}{\| F^M \| \cdot \| R^M_i \|}, \]

where < ·, · > and \( \| \cdot \| \) denote the inner product and \( l_2 \)-norm
respectively; \( R_i = I^M_i - W(I^M_i) \). Fig. 6 shows the results. For all
DeepFake types, the NCC scores between the real images’
noise residuals and DeepFake fingerprints are distributed around
zero, which means that little correlation exists. By contrast, the
NCC scores between the DeepFake images’ noise residuals and
DeepFake fingerprints are remarkably larger than zero, testifying
to a significant correlation with the corresponding fingerprint.

D Discussion

The above model trace analysis identifies three typical model
traces throughout the DeepFake pipeline. The interplay of these
traces distinguishes DeepFake images from real ones. Hence,
a solid and universal attack is expected to eliminate all these
possible traces at once, which is the trace removal attack to be
presented next. In this way, the modified DeepFake images can
become much closer to real images, enabling the evasion of
arbitrary detectors. Since the knowledge of DeepFake traces is
derived from the common DeepFake pipeline, the trace removal
attack is applicable to all types of DeepFake.

IV TR-NET: TRACE REMOVAL ATTACK

A Threat Model

1) Victim Model: We assume the target victim model is an
arbitrary DeepFake detector \( C \), which is a machine learning
classifier that distinguishes trace features between real and
DeepFake images. \( C \) takes an image \( I \) or its hand-crafted features as
input and outputs a decision of \{ Real, Fake \}.

2) Victim Detector’s Capability: The attack is required to
evade an arbitrary DeepFake detector \( C \), which means there are
few restrictions on \( C \)’s capabilities: The developer of \( C \) can use
discretionary model designs and feature engineering, as well as
sufficient training data. \( C \) is also trained with different types of
DeepFakes rather than a single one, so as to have a stronger
generalization ability. In addition, we will consider more robust
detectors that have been embedded with defense. As suggested
in [35], [42], training dataset augmentation with perturbed im-
ages can significantly improve a detector’s robustness against
common attacks. Hence, \( C \) is strengthened during its training
phase with two augmentation strategies that confer weak and
strong defenses, respectively.

Empirical Augmentation. This method augments training
dataset using four empirical perturbation models suggested by
Frank et al. [35], including blurring, cropping, compression
and noising. The four perturbations are combined in a random
order with a probability of 50% for applying each individual
perturbation.

Adversarial Augmentation. This method assumed that the
developer of \( C \) had full knowledge of the attack model and
could collect enough attack samples to augment the training set
directly. This strategy was applied with a probability of 50%
as well.

3) Attacker’s Background Knowledge: The attacker is re-
stricted to having little knowledge of \( C \), e.g., does not know
the network architecture, parameters, or input features of \( C \), nor
does it have any access to the detector or its training dataset.

To train the attack model, the attacker is assumed to have an
auxiliary dataset containing real and fake images. Although this
is a mild assumption as plenty of ready-to-use DeepFake image
datasets and models are publicly available, some additional
restrictions are still imposed on the attacker’s data accessibility

4) Attack Goals: A successful attack means that the target
detector \( C \) will be misled into classifying the attack samples
as “Real”. Meanwhile, the attacker may expect the attack to be
stealthy by preserving the visual quality of the original DeepFake

and The discriminators are used to impel the kernels) and max pool-

}{

to mitigate this problem. First, we ∗:

D is a CNN-based generative model. Thus, it might produce kernels), capturing features at different scales of the images while compacting the spatial information. The decoder path is a symmetric expanding counterpart. In each decoding block, the feature map is upsampled to double size while the number of features is halved. Each decoder block also concatenates the output features with the high-resolution features from the corresponding encoder block, such that the feature and spatial information can be preserved for efficient reconstruction.

An additional challenge is that, as discussed in Section III, G is a CNN-based generative model. Thus, it might produce its own model traces, which may interfere with the trace removal process. The loss functions proposed in the subsequent sections effectively suppress this intrinsic noise brought about by G. In addition, we also made two structural improvements as suggested in [33] and [52] to mitigate this problem. First, we replaced the transposed convolution-based upsampling in the original U-Net with bi-linear interpolation-based upsampling. Second, we added a feature scaling layer before the last convolutional layer of G.

2) Discriminators: The discriminators are used to impel the generator to produce trace-free attack samples via adversarial learning. Therefore, the discriminators should be able to recognize accurate DeepFake trace patterns by learning to classify real and fake images in the trace space. According to our trace discovery, there are three types of traces existing in different domains, each with a unique representation. The inter-domain interference across traces makes a single discriminator learned in one feature domain impractical to represent all traces accurately. To this end, we propose employing a set of parallel discriminators D : {D1, D2, D3} to disentangle different trace representations. As shown in Fig. 9, each discriminator is responsible for one particular input trace representation. All the discriminators have the same network structure built on a five-layer CNN. Note that using a complicated structure for the discriminator is unnecessary since it may lead to extra computational cost and an imbalance between the generator and discriminators during training. A shallow CNN is sufficient to capture these traces accurately in our experiments.

image as high as possible, which means the visual difference between the original DeepFake image and the attack sample should be small enough. Formally, let I ⊕ and I ⊖ be the sets of real images and DeepFake images, respectively. Given a DeepFake image I ⊖ ∈ I ⊖, the attack model learns a mapping \( A : I ⊖ \rightarrow I ^ ∗ \). The attacking sample I^∗ satisfied the following attack goals:

Fraudulence. The attack sample should successfully deceive an arbitrary detector:

\[ \forall C, \quad p(C(I^∗)) = C(I^+) \approx 1. \]

Stealthiness. The attack sample should be perceptually indistiguishable from the original DeepFake image:

\[ \forall I^−, \quad d(I^−, I^∗) \leq \epsilon, \]

where d(·, ·) is a distance function.

A TR-Net

The proposed attack is implemented with a trace removal net-
work (TR-Net) based on adversarial learning. As shown in Fig. 7, TR-Net consists of a generator G and a set of trace discriminators D : {D1, D2, D3}. G takes the original DeepFake images as inputs and evolves to reconstruct them with trace removed to evade being accurately recognized by the trace discriminators. Each discriminator in D is devised for an individual auxiliary trace recognition task. Joint training on D adversarially impels G to remove different traces concurrently. After the adversarial game reaches the equilibrium, the optimal generator G∗ is obtained and adopted as the attack model A, i.e., A = G∗. Then, given a test DeepFake image I^∗ o, the corresponding attack sample is I^∗ o = A(I^∗ o).

1) Generator: The generator G is a deep auto-encoder that learns to generate trace-free samples from the original DeepFake samples with an unchanged image size. The backbone of G is a u-shaped network (U-Net) [53] given its remarkable capacity to reconstruct high-quality images. As shown in Fig. 8, G consists of an encoder path and a decoder path. The encoder involves repeated convolutional layers (with 3 × 3 kernels) and max pool-

ing layers (with 2 × 2 kernels), capturing features at different scales of the images while compacting the spatial information. Therefore, the discriminators should be able to rec-

Fig. 7. Framework overview of the trace removal network.
a) Spatial Discriminator $D_1$: $D_1$ captures potential spatial anomalies in the spatial domain, including distortions, inconsistencies, disharmony, etc. Similar to the original discriminator in a normal GAN, $D_1$ is trained directly with the RGB pixel values, and thus can be seen as an incremental refinement on the raw DeepFake images in terms of visual quality.

b) Spectral Discriminator $D_2$: $D_2$ learns to recognize the spectral disparities between the real and attack samples. Unlike $D_1$, $D_2$ takes the frequency spectrum instead of RGB pixels as its input. The frequency spectrum is transformed from the pixel values by two-dimensional Discrete Fourier Transform (2D-DFT). Given a natural image $I \in \mathbb{R}^{M \times N}$, the 2D-DFT maps each pixel value of the gray-scale component of $I$ to a frequency value $F(u,v) \in \mathbb{R}^{M \times N}$:

$$F(I)(u,v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m,n) \cdot e^{-2\pi i \left( \frac{um}{M} + \frac{vn}{N} \right)}.$$  \hspace{1cm} (4)

As the imaginary part is incompatible with a CNN for calculating gradients, directly applying the 2D-DFT $F(I)$ to $D_2$ is impractical. Instead, we decompose the complex-valued matrix of $F(I)$ into its amplitude response $F_{am}(I)$ and phase response $F_{ph}(I)$. Let the complex form of $F(I)$ be $F(I) = a + bi$, and we have:

$$F_{am}(I) = |F(u,v)| = \sqrt{a^2 + b^2}$$

$$F_{ph}(I) = \angle F(u,v) = \arctan \frac{b}{a}.$$  \hspace{1cm} (5)

Then, the two components are concatenated as a 2-channel real-valued matrix as the input of $D_2$, denoted as $\hat{I} = [F_{am}(I), F_{ph}(I)]$.

c) Fingerprint Discriminator $D_3$: $D_3$ targets the DeepFake’s model fingerprint in the noise space. A reliable fingerprint encoder is required to disentangle accurate fingerprint traces in the input feature space. Existing DeepFake fingerprint encoders include a noise-based method [41] and a learning-based method [42]. We propose to combine the two methods for a more accurate representation. First, a residual noise extraction is performed to represent the noise-level fingerprints. An SRM filter is adopted for this purpose given its effectiveness in estimating local noise distributions for image forensics [54]. The SRM filter has three layers with the following kernels:

$$k_1 = \frac{1}{4} \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & 2 & 4 & 2 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$k_2 = \frac{1}{12} \begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & 2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix},$$

$$k_3 = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The input to $D_3$ is then denoted as $\hat{I} = \text{SRM}(I)$. Then, $D_3$ further learns the fine-grained fingerprints from the noise-level fingerprints in the “real versus fake” classification task. In this way, the representation learning of fingerprints can be integrated into the adversarial training smoothly.

3) Loss Functions: We design an adversarial loss to supervise the training of $G$ and $\mathcal{D}$, which enables trace removal so as to realize the goal of fraudulence. Regarding the goal of stealthiness, a visual similarity loss is imposed on $G$ to ensure that the content and quality of the original DeepFake samples are perfectly preserved in the corresponding attack samples. One additional technical challenge is that an ideal trace removal attack requires simultaneously closing the distribution gap between the attack and real samples at the trace level; and between the attack and original DeepFake samples at the semantic level (see Fig. 10). However, due to the information continuity in an image, the trace features overlap the semantic features to some extent, leading to a potential conflict in feature migration directions during optimization. We mitigate this nontrivial problem specifically in the devised loss function, as shown next.

a) Adversarial Loss: The adversarial learning of TR-Net is performed with the input data pairs in the form of $(I^+, I^-)$.
The discriminators continuously learn to distinguish the generator’s output \(G(I^-)\) from \(I^+\) in different feature spaces, while the generator tries to mislead the discriminators’ prediction of \(G(I^-)\). Conventionally, \(I^+\) and \(I^-\) are randomly sampled from \(I^+\) and \(\mathbb{I}^-\) respectively. However, random sampling is less practical for TR-Net’s optimization considering the conflicts between semantic features and trace features. The visual contents between \(I^+\) and \(I^-\) should be as consistent as possible to enforce the discriminators to focus on purer trace features while reducing their bias to semantic features. Thus, we construct the semantically-closest pairs instead of random sampling for discriminator supervision: If a fake sample is produced by a method where a source real image exists, such as facial attribute editing and face replacement, the source image is applied straightforward as the semantically-closest counterpart. For a face synthesis sample created out of nowhere, its nearest neighbor is retrieved from the real image set \(I^+\) as a counterpart. With the semantically-closest pairs in hand, the adversarial loss for jointly training the “one-versus-multiple” framework is denoted as:

\[
\mathcal{L}_{\text{adv}}(G,D_1,D_2,D_3) = \lambda_1 \mathcal{L}(G,D_1) + \lambda_2 \mathcal{L}(G,D_2) + \lambda_3 \mathcal{L}(G,D_3),
\]

where

\[
\mathcal{L}(G,D_i) = \mathbb{E}_{x^+ \sim x^-}[\log(D_i(x^+)) + \log(1 - D_i(G(x^-)))] ,
\]

The input \(x\) varies for different discriminators, i.e., \(x = I\) for \(D_1\), \(x = \bar{I}\) for \(D_2\) and \(x = \tilde{I}\) for \(D_3\). \(\lambda_1, \lambda_2, \lambda_3\) are weights to balance the contribution of three discriminators, subject to \(\lambda_1 + \lambda_2 + \lambda_3 = 1\).

b) Visual Similarity Loss: To satisfy the stealthiness goal, a visual similarity loss is additionally imposed on the generator \(G\). The commonly-used pixel-wise distance \(\|I^+ - G(I^-)\|_2\) is not applicable to our method as it may cause pixel-level overfitting, which will exacerbate the conflict between the semantic features and the trace features, and thus compromise the trace removal. Moreover, despite having \(D_2\) to encourage the spectra matching from attack samples to real images, we experimentally find that only a \(D_2\) is insufficient to well match high-frequency components. This is because the information in natural images tends to be centralized in lower-frequency components.

Instead, we propose a novel visual similarity loss plus the power spectral density (PSD) regularization to address the above problem. Given an image \(I\), first, a filter is applied to its center-shifted DFT spectrum, decomposing \(I\) into the low-frequency component \(I_l\) and the high-frequency component \(I_h\):

\[
\begin{align*}
I_l &= \mathcal{F}^{-1}(\mathcal{H}(u,v) \cdot \mathcal{F}(u,v)) \\
I_h &= \mathcal{F}^{-1}(1 - \mathcal{H}(u,v) \cdot \mathcal{F}(u,v))
\end{align*}
\]

where \(\mathcal{F}^{-1}\) is the reverse DFT, \(\mathcal{H}(u,v) = e^{\exp(-\frac{u^2+v^2}{2\sigma^2})}\) is a Gaussian filter. Then the visual similarity loss between a source fake image \(I^-\) and its reconstructed version \(G(I^-)\) is computed as the VGG-based perceptual loss \([55]\) on the low frequency components:

\[
\mathcal{L}_{\text{sim}}(G) = \frac{1}{W \times H} \| VGG_k(I_l^-) - VGG_k(G(I^-)_l) \|_2^2 ,
\]

where \(W\) and \(H\) are the dimensions of the respective feature maps within the pretrained VGG network \([56]\) and \(VGG_k\) denotes the features extracted at VGG’s \(k\)-th layer.

A PSD regularization is additionally added to \(\mathcal{L}_{\text{sim}}\) to enforce the mapping of frequency information between the attack and real images. The PSD of an image \(I\) can be represented as a one-dimensional profile of the center-shifted power spectrum resulting from an azimuthal integration over each radial frequency \(\theta\):

\[
\text{PSD} (\omega_k) = \int_0^{2\pi} | \mathcal{F}(I) (\omega_k \cdot \cos(\theta), \omega_k \cdot \sin(\theta)) |^2 d\theta
\]

for \(k = 0, \ldots, M/2 - 1\).

Benefiting from the semantically-closest pair \((I^+, I^-)\) where the in-pair lower frequency components are similar, the PSD regularization can operate on the high-frequency components merely, computed as the euclidean distance between the PSDs of \(I^-_k\) and \(G(I^-)_k\):

\[
\mathcal{L}_{\text{reg}}(G) = \frac{1}{M/2 - 1} \| \text{PSD}(I^-_k) - \text{PSD}(G(I^-)_k) \|_2^2
\]

The final training objective of the TR-Net is:

\[
T = \arg \min_{(G,D_3)} \max_{(D_1,D_2,D_3)} \{ \mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{sim}} + \mathcal{L}_{\text{reg}} \}
\]

C Comparison With Previous Attacks

Regarding the adversarial attack, existing methods have some limitations. First, searching for the optimal adversarial noise to a target detector requires a certain level of detector-side information, such as the parameters, network structure, or outputs. Thus, the attack will have a transferability issue when facing unknown or black-box detectors \([14],[15]\). Second, the attack feasibility against some advanced detectors involving sophisticated network designs will be problematic, because in such cases it becomes difficult, if not impossible, to search for the optimal noise with an acceptable trade-off between attack efficacy and visual quality. Similarly, reconstruction-based attacks are also performed in a “detector-specific” way. The attacker is assumed to know what type of forgery features are of prime interest to the target detector. What is worse, existing methods only focus on an individual type of feature in a single domain, irrespective of the fact that various traces are present in different domains.

Fairly, the trace removal attack is more analogous to the genre of reconstruction-based attacks, but fundamental differences exist. Our method improves anti-forensic attacks by removing multiple forgery traces at the same time. Furthermore, they are removed in a way that is agnostic to the detector, resulting in better transferability to unknown and black-box detectors. Technically, this is more challenging than dealing with a single trace feature considering the interplay between traces and the inter-domain interference, yet the proposed TR-Net is competent to meet the challenge.
V EXPERIMENTAL EVALUATIONS

We evaluate the trace removal attack in heterogeneous security scenarios where the attackers and targets detector have varying knowledge and capabilities. In each scenario, the attack is assessed by verifying whether the two goals, fraudulence and stealthiness, have been satisfied.

A Evaluation Metrics

(1) The fraudulence goal is verified in terms of detection accuracy, calculated as the proportion of correctly classified samples out of all the samples in a single class. Attack samples with higher fraudulence result in lower detection accuracy of the test detector.

(2) The stealthiness goal is verified by assessing the visual similarity and quality consistency between the attack and original DeepFake samples, which are indicated by two common image assessment metrics, structural similarity (SSIM) and peak signal-to-noise ratio (PSNR), respectively. A larger value in PSNR or SSIM indicates better attack stealthiness.

B Datasets

The proposed trace removal attack is applicable to all DeepFake types including face synthesis, facial attribute editing, and face replacement. To the best of our knowledge, the current public DeepFake datasets fail to cover all these methods. Thus, we create a new DeepFake dataset called All-in-One-DF for a thorough evaluation, which consists of 66,000 semantically-closest pairs of real and fake images (i.e., 132,000 images in total) from four sources:

1) CelebA: A large-scale dataset containing more than 200k real face images. The images are cropped and aligned to the size of $128 \times 128 \times 3$ with the face in the centre.

2) Face synthesis: We employ ProGAN, one of the most popular GANs to synthesize non-existing face images. We query 22,000 fake images from the CelebA-pretrained ProGAN instance released by yu et al. [21], and then retrieve their most similar counterparts from the CelebA dataset to construct the semantically-closest pairs.

3) Facial attribute editing: We select STGAN, a state-of-the-art facial attribute editing GAN. We randomly sample 22,000 real images from the remaining CelebA dataset and apply the officially-released STGAN instance [21] to modify facial age or hair colour, resulting in 22,000 fake samples.

4) Face replacement: DeepfakeTIMIT [57] is a GAN-based face-swapping video dataset. There are 320 pairs of source videos and the face-swapped counterparts in DeepfakeTIMIT. We randomly select 22,000 frames from all videos on each side, followed by face-centered cropping to the size of $128 \times 128 \times 3$. Fig. 11 provides some pairwise examples from the All-in-One-DF dataset.

C Selected Victim Detectors

To show the efficacy of the proposed attack model in attacking arbitrary detectors, we select six representative DeepFake detectors recently proposed, which evenly cover the three detector categories outlined in Section II.C.

Spatial-based detectors: Xception [29] is a deep CNN widely adopted as the backbone network in face forgery forensics tasks. It has achieved leading performance in some benchmark datasets by learning directly from RGB pixel inputs. Patch-CNN [28] focuses on the local properties in semantic regions rather than on global semantics. It aggregates the decisions of a set of truncated Xceptions learned from image patches for the final decision.

Frequency-based detectors: DCTA [35] is a shallown CNN classifier learned from the 2D-DCT spectra of images. F^3-Net [38] is one of the state-of-the-arts in DeepFake detection. It involves a two-stream collaborative network that combines frequency-aware decomposition and local frequency statistics to learn frequency-aware clues.

Fingerprint-based detectors: LF [42] is a deep CNN that learns GAN fingerprints in a multi-source identification task. The original multi-classification results are further divided into the real or fake class. NF [41] is a non-trainable method that differentiates GAN images from real ones by the correlations of noise-based fingerprints.

In addition, we consider a stronger detector by ensemble learning the three categories of detectors, denoted as Ensemble. Xception, DCTA and LF are selected as the base detectors, and a random forest classifier is trained using the features from the final pooling layers of base detectors.

D Settings

The All-in-One-DF dataset is randomly partitioned into a training set with 60,000 semantically-closest pairs and an evaluation set with 6,000 pairs. For all detectors, we follow the training settings recommended in the original papers. The detectors are trained on the training set with a 9 : 1 training-validation ratio. Regarding the training of TR-Net, we set the batch size to 150. Both the generator and discriminators are optimized using the RMSprop optimizer [58] with initial learning rates of $1.6e^{-3}$ and $1.6e^{-4}$, respectively, plus a scheduler with a decay rate of 0.5. The scheduler is executed at the end of a training epoch if the loss stopped decreasing. There are 8 training epochs in total. The weight set $\{\lambda_1, \lambda_2, \lambda_3\}$ is set to $\{0.2, 0.6, 0.2\}$. The weight decision process is detailed in the Supplement, available online.
Evaluating Fraudulence: a) Attacking Detectors Without Defense

The bold value indicates the best attack result in each row.

Table I: Performances of Five Attack Methods Against Seven Detectors Without Defense

| Accuracy(%) | Clean | Noise | FGSM | PGD | GANprintR | TR-Net |
|-------------|-------|-------|------|-----|-----------|--------|
| Xception    | 99.88 | 65.44 | 4.43 | 0.01| 58.53     | 17.90  |
| Patch-CNN   | 92.13 | 53.91 | 12.36| 8.91| 57.31     | 13.06  |
| DCTA        | 90.66 | 51.59 | 33.18| 25.37| 70.24    | 20.21  |
| F²-Net      | 99.97 | 85.41 | 49.62| 45.73| 80.73    | 31.10  |
| LF          | 91.85 | 37.12 | 14.00| 15.55| 64.76    | 14.75  |
| NF          | 71.12 | 42.65 | 28.21| 25.70| 31.48    | 22.74  |
| Ensemble    | 99.80 | 81.23 | 50.11| 47.20| 83.33    | 40.21  |
| Average     | 92.16 | 59.62 | 27.70| 24.20| 63.83    | 22.65  |

After training, the checkpoint with the minimal generator loss in the last epoch is nominated as the attack model and applied to the 6,000 fake images in the evaluation set to craft attack samples.

E Attacking With Unlimited Background Knowledge

We first evaluate the attack performance in the scenario where the attacker has no limits on the background knowledge of data, i.e., the whole training set is available for training the attack model. For a comprehensive evaluation, the proposed trace removal attack is compared with several baseline attack methods. Also the detectors with varying abilities are considered, i.e., detectors without defense, with weak defense, and with strong defense.

1) Baseline Attacking Methods: We select four other attack methods to demonstrate baseline performance, including adding random noise (Noise), two classic adversarial attacks FGSM [46] and PGD [49], and a reconstruction-based attack GANprintR [12]. The Noise attack adds Gaussian noise into the images with a Gaussian variance randomly sampled from $U(5.0, 20.0)$. Both the FGSM and PGD attacks are optimized based on the Xception detector and then apply to all detectors so as to assess the white-box and black-box attack capacities simultaneously. The maximum perturbation $\epsilon$ is set as 0.003 for both attacks. For GANprintR, we follow the setting in the original paper.

2) Evaluating Fraudulence: a) Attacking Detectors Without Defense: Table I details the attack results against detectors without defense. The first column shows the detection accuracy on the original clean fake samples. All detectors achieve high accuracy over 90.00%, except for the non-trainable detector NF. After attack, the accuracy of all detectors decreases, indicating that the state-of-the-art detectors are still vulnerable to attacks. The frequency-based detectors, especially F²-Net, are relatively more robust than other individual detectors. The reason may be that all attacks lead to more significant changes in the frequency domain than in the pixel domain. In addition, the ensemble detector targeting all traces, unsurprisingly, outperforms any individual detector targeting a single trace in terms of robustness.

Regarding the attacks, the adversarial attack methods, FGSM and PGD, are particularly destructive to Xception, which is not surprising because they are optimized based on Xception in a white-box manner. They also showed good transferability on Patch-CNN which has similar structural blocks to Xception.

However, as earlier discussed, this detector-specific design leads to poor transferability on other unknown types of detectors. By comparison, TR-Net takes advantage of the detector-agnostic design, achieving competitive or superior results in attacking all six detectors. After the trace removal attack, the classification accuracy of all detectors has decreased markedly, and the average accuracy of the six has dropped from 92.16% to 22.85%. The results indicate the proposed trace removal attack is universal and well transferable across different detectors.

b) Attacking Detectors With Defenses: Next, we test the attacks in the cases that the detectors are embedded with varying defenses as described in Section IV.A.2. Detectors with the weak defense, i.e., the empirical augmentation strategy, are denoted as {model name}(+) and those with the strong adversarial augmentation defense are denoted as {model name}(++).

Weak Defense. Table II shows the classification accuracy of detectors embedded with the empirical data augmentation strategy. The high accuracy values in the first column indicate that all detectors still maintain stable detection capability on clean samples after defense. We can see that after being strengthened with empirical data augmentation, the robustness of all detectors were improved against all attack methods. Regarding the attack methods, the Noise attack is barely misled the detectors, and the attacking specificity of FGSM and PGD on Xception(+) and Patch-CNN(+) was no longer significant. Comparatively, the TR-Net maintained satisfactory, degrading the classification accuracy of almost all detectors to lower than random guess, except for F²-Net(++). Moreover, TR-Net surpassed all baseline attack methods in five out of the six detector groups.

Strong Defense. Table III shows the classification accuracy of the detectors embedded with the adversarial data augmentation strategy. Note that the adversarial augmentation strategy is specific to each attack method, thus, the results on clean (cle) / attacked (att) DeepFake samples are reported individually for each attack method.

From the table, we can see that the adversarial data augmentation strategy substantially improved the robustness of all detectors against the four baseline attack methods. Take PGD, the best baseline attack method in our experiments as an example, the average accuracy of strongly defended detectors only degrades from 92.19% to 73.54%, whereas the corresponding result for the weakly-defended detector is 93.47% down to 48.28%, and...
samples can be released to the public. As mentioned, this goal was achieved through the proposed trace removal attack, which demonstrates the visual quality differences between the naked detector and the attacker. However, the attacker still has access to a limited dataset, we randomly sampled six subsets from the full training set, containing 5% of the examples of three types of DeepFake images and the corresponding attack samples from different attack methods. The bold value indicates the best result in each row.

![Image](311x494 to 554x613)

Fig. 12. The average detection accuracy under five attack methods in three defensive strategy groups.

92.16% down to 24.20% for the naked detector. However, the strong defense only makes a relatively small impact on our attacking method. The average detection accuracy on the TR-Net samples is 59.74%, much lower than the accuracy on other attack samples, at merely a little higher than a random guess.

We also observe that, intriguingly, unlike other augmentation methods, adversarial augmentation with the TR-Net samples can significantly compromise the detectors’ ability to classify clean fake samples. The reason may be that trace-free samples are inherently closer to real samples, which can confuse the detector during training. This reveals the potential to use trace removal attack to poison a DeepFake detection dataset, which remains future investigation.

c) Discussion: Fig. 12 offers an intuitive comparison of five attack methods in three defense strategy groups. The trace removal attack is the most effective in all groups. In the circumstances where detectors are defended with data augmentation strategies, especially the adversarial augmentation strategy, the baseline attack methods generally undergo a considerable loss of efficacy, while TR-Net continues to pose a threat, and the threat is even more serious than the white-box adversarial attacks. In addition, as shown in Tables I, II, and III, The trace removal attack shows superior transferability across different detectors compared to the baselines. Again, we emphasize that, unlike the baseline attacks, our attack was implemented upon all detectors being completely unknown during training. In conclusion, the attacking goal of fraudulence is well satisfied by the proposed trace removal attack.

3) Evaluating Stealthiness: As mentioned, this goal was evaluated based on visual quality. To be considered stealthy, a given attack sample was required to be perceptually indistinguishable from the corresponding DeepFake image.

Table IV demonstrates the visual quality differences between DeepFake samples and attack samples in the evaluation set in terms of the average PSNR and SSIM scores. TR-Net achieved the highest PSNR (35.16 ± 3.32db) and SSIM (0.988 ± 0.004) scores, indicating that attack samples generated by TR-Net contain less noise and have a visual quality closer to the original DeepFake samples. Fig. 13 also provides a qualitative view of the examples of different types of DeepFake images and the corresponding attack samples from different attack methods. As shown in the figure, the methods that add noise, including including Noise, FGSM and PGD, bring perceptual noise or a blurriness to the attack samples that may have been potentially screened out by the forensic investigators. By contrast, the reconstruction-based methods, especially TR-Net, generated high-fidelity attack samples that were perceptually similar to the original ones. Thus, we can conclude that the goal of stealthiness is well satisfied as well.

F Attacking With Limited Background Knowledge

The next evaluation scenario imposes restrictions on the attacker's background knowledge of data. Here, only the Xception and F3-Net detectors are tested considering their generally better detection ability than other detectors.

1) Limited Dataset Size: To simulate that the attacker only has access to a limited dataset, we randomly sampled six subsets from the full training set, containing 1%, 5%, 10%, 25%, 50%, 75% and 100% of data (i.e., for 1% that equates to a total of 660 semantically-closest pairs of real and fake images). Then we trained TR-Net from scratch on each subset individually and

![Image](Original Deepfake, No attack, FGSM, PGD, Xception, F3-Net, TR-Net)

Fig. 13. Examples of three types of DeepFake images and the corresponding attack samples from different attack methods.

![Image](Original Deepfake, No attack, FGSM, PGD, Xception, F3-Net, TR-Net)

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
evaluated our results on the same evaluation dataset as in the previous scenario.

Fig. 14(a) and (b) illustrates the detection accuracy and PSNR and SSIM scores for each subset. From the results, it appears that there is a threshold for the dataset size that is within $10\% - 25\%$, under which both the accuracy and visual quality are affected. This is unsurprising since attack methods based on GAN learning are essentially data-driven. However, when the training set size equates to more than a quarter of the original data set, all metrics increase rapidly and remain relatively stable at a satisfactory level. The results indicate that TR-Net fits well even with a relatively small amount of training samples which are easily collected. This weak data volume-dependency makes TR-Net practically feasible.

2) Out-of-Distribution DeepFake: We also assessed the attack on out-of-distribution DeepFakes to demonstrate its domain independence. In this scenario, the attacker was restricted to train the model with only two types of DeepFake images, while evaluate on all three DeepFake types. For example, "P+S" indicates training with ProGAN and STGAN samples.

Fig. 14(c) and (d) show the results for detection accuracy and visual quality when trained with different training groups. What is shown is that, when implementing the attack on the samples generated by an unknown DeepFake method that is not included in the training set, the resulting attack samples suffer from a decrease in both detection-evasive ability and visual quality. For instance, comparing the ProGAN results in the "P+S" group (where ProGAN is included in the training set) with those in the "S+D" group (where ProGAN is not included in the training set), the TR-Net practically feasible.

The trace features output by the last convolutional layer of the optimal generator $G^*$ are analyzed, with dimensionality reduced by t-SNE [59] so as to obtain an interpretable two-dimensional view of geometrical shifting.

Fig. 15 shows three types of trace encoded by three discriminators individually. We can see that there is a distinct trend that the trace features of attack samples are transferring toward the features of real images, which confirms our conjecture that TR-Net can reduce the distribution gap between the attack and real samples at the trace feature level via adversarial learning. In addition, the spectral traces from $D_1$ and the fingerprint traces from $D_3$ migrate more significantly than the spectral traces from $D_2$, which is a result of the optimization conflict between the semantic features and trace features outlined in Section IV.B.3. Since the frequency components are closely associated with both trace and semantic information where no distinct boundary applies, weakening the trace representations of DeepFake samples while retaining their visual information must lead to a sub-optimum. Even so, the attack efficacy is barely affected as shown in previous experiments.

2) Explanation in the Frequency and Noise Spaces: Next, we explain the trace removal in the frequency and noise space. First, we compare the average PSD distributions of real, DeepFake and attack samples, as shown in Fig. 16. The results show that for all DeepFake types, the distribution gaps between attack and real samples are significantly smaller than those between attack and fake samples. The result again indicates the wide applicability of the trace removal attack to various DeepFake types. The gaps between attack and real samples are smaller in the STGAN and DeepfakeTIMIT groups than in the ProGAN group. We suppose the reason is the semantically-closest pairs: in the STGAN and DeepfakeTIMIT groups, each fake sample in a semantically-closest pair has an exact source real image counterpart. In contrast, the fake samples in the ProGAN group only correspond to nearest-neighbor similar real images where unknown DeepfakeTIMIT samples than that for unknown ProGAN and STGAN samples. The reason is that both the source ProGAN and STGAN models were pre-trained with the CelebA dataset, while the source DeepfakeTIMIT model is developed with another dataset where a domain inconsistency exists. Our findings suggest that fine-tuning TR-Net in a domain to be consistent with the target detector helps to improve the efficacy of the attack.

G A Closer Look into Trace Removal

In this section, we offer a closer look at the trace removal results to understand why and how TR-Net is able to remove all traces.

1) Explanation in the Feature Space: To verify whether different DeepFake trace representations are well disentangled by a set of discriminators, we analyzed the geometrical shifting of trace features encoded by each discriminator in the latent space. Since the generator and discriminators are trained in parallel, the discriminators $D^∗$ resulting from the same checkpoints of the optimal generator $G^*$ are adopted as the trace feature descriptors. The trace features output by the last $512\times 4\times 4$ convolutional layer of $D^*$ are analyzed, with dimensionality reduced by t-SNE [59] so as to obtain an interpretable two-dimensional view of geometrical shifting.

In this section, we offer a closer look at the trace removal results to understand why and how TR-Net is able to remove all traces.
Fig. 15. Trace features in the latent spaces learned by, from left to right, the spatial discriminator $D_1$, the spectral discriminator $D_2$ and the fingerprint discriminator $D_3$. t-SNE is used to project the representations of features from each discriminator’s last convolutional layer onto its two principal components.

Fig. 16. The power spectral density distributions of real, DeepFake and attack samples for three DeepFake types. The zoom in box highlights the main areas of high-frequency spectral distributional gaps.

Fig. 17. The average spectral and noise-level differences between real and original DeepFake samples are much more significant than those between real and attack samples. This result further outlines the fact that successful trace removal will refine the DeepFake images to be closer to the real ones, making them evasive to arbitrary detectors.

3) The Effect of Individual Traces: We also investigate the effect of removing each individual trace instead of all. For this purpose, we evaluate three sub-versions of TR-Net where only an independent discriminator is employed for each, namely TR-Net-$D_1$, TR-Net-$D_2$ and TR-Net-$D_3$, with a comparison to the full version of TR-Net. Table V shows the detection accuracy and visual quality results. We can see that the DeepFake samples with an individual trace being removed are more powerful in evading the corresponding type of detectors than other

| Table V: The Effect of Individual Trace Removal |
|-----------------------------------------------|
|                  | TR-Net-$D_1$ | TR-Net-$D_2$ | TR-Net-$D_3$ | TR-Net |
|------------------|--------------|--------------|--------------|--------|
| Spatial detectors|              |              |              |        |
| Xception         | 20.11        | 40.01        | 24.87        | 17.90  |
| Patch-CNN        | 27.02        | 45.10        | 29.00        | 13.06  |
| Frequency detectors|            |              |              |        |
| DCTA             | 71.82        | 14.31        | 67.52        | 20.21  |
| F³-Net           | 77.94        | 28.99        | 66.66        | 31.10  |
| Fingerprint detectors|        |              |              |        |
| LF               | 45.17        | 40.23        | 48.10        | 14.75  |
| NF               | 35.90        | 38.71        | 40.79        | 22.74  |
| Visual quality   |              |              |              |        |
| SSIM             | 0.991        | 0.954        | 0.993        | 0.988  |
types of detectors. Moreover, the TR-Net-D2 samples show better cross-detector transferability than the TR-Net-D1 and TR-Net-D3 samples, implying that spectral disparity may be the most significant feature differing DeepFakes from real images. However, the good transferability of TR-Net-D3 is achieved at the cost of visual quality, as indicated by the PSNR and SSIM scores, due to the frequency-domain overcorrection will lead to visual distortion. Compared to these sub-versions, TR-Net removing all traces at once results in the best trade-off between transferability and visual quality.

VI CONCLUSION AND FUTURE WORK

In this paper, we focused on an anti-forensics attack against DeepFake detectors. We presented a novel detector-agnostic attack, called a trace removal attack, that is capable of refining DeepFake images by removing all possible DeepFake traces via an one-versus-multiple adversarial learning network. The refined DeepFake images are closer to the real images and can therefore bypass arbitrary and even unknown detectors. We assessed the efficacy of the trace removal attack against a wide range of state-of-the-art detectors in heterogeneous high-level security scenarios where the detectors were embedded with various defensive strategies and the attacker’s knowledge of data was limited. Our findings reveal that, the proposed trace removal attack achieves the highest attack effectiveness while introducing minimal visual quality loss compared with contemporary adversarial and reconstruction-based attacks. In the future, we will focus on developing more robust forensics countermeasures against trace removal attacks.

REFERENCES

[1] M. Westerlund, “The emergence of Deepfake technology: A review,” Technol. Innov. Manage. Rev., vol. 9, no. 11, pp. 39–52, 2019.
[2] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, “DeepFakes and beyond: A survey of face manipulation and fake detection,” Inf. Fusion, vol. 64, pp. 131–148, 2020.
[3] S. Hussain, P. Neekhara, M. Jere, F. Koushanfar, and J. McAuley, “Adversarial DeepFakes: Evaluating vulnerability of DeepFake detectors to adversarial examples,” in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis., 2021, pp. 3348–3357.
[4] P. Neekhara, B. Dohlansky, J. Bitton, and C. C. Ferrer, “Adversarial threats to Deepfake detection: A practical perspective,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 923–932.
[5] N. Carlini and H. Farid, “Evaluating Deepfake-image detectors with white- and black-box attacks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops, 2020, pp. 658–659.
[6] A. Gandhi and S. Jain, “Adversarial perturbations fool DeepFake detectors,” in Proc. Int. Joint Conf. Neural Netw., 2020, pp. 1–8.
[7] L. Fan, W. Li, and X. Cui, “DeepFake-image anti-forensics with adversarial examples attacks,” Future Internet, vol. 13, no. 11, 2021, Art. no. 288.
[8] Q. Liao et al., “Imperceptible adversarial examples for fake image detection,” in Proc. IEEE Int. Conf. Image Process., IEEE, 2021, pp. 3912–3916.
[9] Y. Wang, X. Ding, Y. Yang, L. Ding, R. Ward, and Z. J. Wang, “Perception matters: Exploring imperceptible and transferable anti-forensics for GAN-generated fake face imagery detection,” Pattern Recognit. Lett., vol. 146, pp. 15–22, 2021.
[10] Y. Huang et al., “FakeRetouch: Evading DeepFake detections via the guidance of deliberate noise,” 2020, arXiv:2009.09213.
[11] F. Ding, G. Zhu, Y. Li, X. Zhang, P. K. Atrey, and S. Lyu, “Anti-forensics for face swapping videos via adversarial training,” IEEE Trans. Multimdia, vol. 24, pp. 3429–3441, 2022.
[12] J. C. Neves, R. Tolosana, R. Vera-Rodriguez, V. Lopes, H. Proença, and J. Fierrez, “GANprintR: Improved fakes and evaluation of the state of the art in face manipulation detection,” IEEE J. Sel. Topics Signal Process., vol. 14, no. 5, pp. 1038–1048, Aug. 2020.
[13] Y. Huang et al., “Fakemixer: Making DeepFakes more detection-evasive by shallow reconstruction,” in Proc. 28th ACM Int. Conf. Multimedia, 2020, pp. 1217–1226.
[14] M. Barni, K. Kallas, E. Nowroozi, and B. Tondi, “On the transferability of adversarial examples against CNN-based image forensics,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process. IEEE, 2019, pp. 8286–8290.
[15] X. Zhao and M. C. Stamm, “The effect of class definitions on the transferability of adversarial attacks against forensic CNNs,” Electron. Imag. vol. 2020, no. 4, pp. 119–119-7, 2020.
[16] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, “CNN-generated images are surprisingly easy to spot... for now,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 8695–8704.
[17] I. Goodfellow et al., “Generative adversarial nets,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2014, pp. 2672–2680.
[18] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of GANs for improved quality, stability, and variation,” in Proc. Int. Conf. Learn. Representations, 2018.
[19] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 4401–4410.
[20] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8789–8797.
[21] M. Liu et al., “STGAN: A unified selective transfer network for arbitrary image attribute editing,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 3673–3682.
[22] F. Matern, C. Riess, and M. Stamminger, “Exploiting visual artifacts to expose DeepFakes and face manipulations,” in Proc. IEEE Winter Appl. Comput. Vis. Workshops, IEEE, 2019, pp. 83–92.
[23] H. Li, B. Li, S. Tan, and J. Huang, “Identification of deep network generated images using disparities in color components,” Signal Process., vol. 174, 2020, Art. no. 107616.
[24] B. Chen, X. Liu, Y. Zheng, G. Zhao, and Y.-Q. Shi, “A robust GAN-generated face detection method based on dual-color spaces and an improved exception,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 6, pp. 3527–3538, Jun. 2022.
[25] Z. Liu, X. Qi, and P. H. Torr, “Global texture enhancement for fake face detection in the wild,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 8060–8069.
[26] Y. Nirkin, L. Wolf, Y. Keller, and T. Hassner, “DeepFake detection based on discrepancies between faces and their context,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 10, pp. 6111–6121, Oct. 2022.
[27] Y. Li and S. Lyu, “Exposing DeepFake videos by detecting face warping artifacts,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, 2019.
[28] L. Chai, D. Bau, S.-N. Lim, and P. Isola, “What makes fake images detectable? Understanding properties that generalize,” in Proc. Eur. Conf. Comput. Vis., Springer, 2020, pp. 103–120.
[29] A. Rosseti, D. Cozzolino, L. Yaroslavtsev, C. Rissel, J. Thies, and M. Niellner, “FacForensics++: Learning to detect manipulated facial images,” IEEE Trans. Comput. Vis. Pattern Recognit., 2019, pp. 1–11.
[30] S. Das et al., “Towards solving the DeepFake problem: An analysis on improving DeepFake detection using dynamic face augmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 3776–3785.
[31] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 1251–1258.
[32] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.
[33] R. Durall, M. Keuper, and J. Keuper, “Watch your up-convolution: CNN based generative deep neural networks are failing to reproduce spectral distributions,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 7780–7789.
[34] T. Dzanic, K. Shah, and F. Witherden, “Fourier spectrum discrepancies in deep network generated images,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, pp. 3022–3032.
[35] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, “Leveraging frequency analysis for deep fake image recognition,” in Proc. Int. Conf. Mach. Learn., PMLR, 2020, pp. 3247–3258.
[36] H. Liu et al., “Spatial-phase shallow learning: Rethinking face forgery detection in frequency domain,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 772–781.

[37] X. Zhang, S. Karaman, and S.-F. Chang, “Detecting and simulating artifacts in GAN fake images,” in Proc. IEEE Int. Workshop Inf. Forensics Secur. IEEE, 2019, pp. 1–6.

[38] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, “Thinking in frequency: Face forgery detection by mining frequency-aware clues,” in Proc. Eur. Conf. Comput. Vis., Springer, 2020, pp. 86–103.

[39] S. Chen, T. Yao, Y. Chen, S. Ding, J. Li, and R. Ji, “Local relation learning for face forgery detection,” in Proc. AAAI Conf. Artif. Intell., 2021, pp. 1081–1088.

[40] Y. Luo, Y. Zhang, J. Yan, and W. Liu, “Generalizing face forgery detection with high-frequency features,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 16 317–16 326.

[41] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, “Do GANs leave artificial fingerprints?,” in Proc. IEEE Conf. Multimedia Inf. Process. Retrieval, IEEE, 2019, pp. 506–511.

[42] N. Yu, L. S. Davis, and M. Fritz, “Attributing fake images to GANs: Learning and analyzing GAN fingerprints,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 7556–7566.

[43] T. Yang et al., “Learning to disentangle GAN fingerprint for fake image attribution,” 2021, arXiv:2106.08749.

[44] N. Yu, V. Skripnik, S. Abdelnabi, and M. Fritz, “Artificial fingerprinting for generative models: Rooting DeepFake attribution in training data,” in Proc. IEEE/CVF Conf. Comput. Int. Conf. Vis., 2021, pp. 14 448–14 457.

[45] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2014, arXiv:1412.6572.

[46] A. Kurakin, I. J. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” in Artificial Intelligence Safety and Security, London, U.K./Boca Raton, FL, USA: Chapman and Hall/CRC, 2018, pp. 99–112.

[47] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in Proc. IEEE Symp. Secur. Privacy, IEEE, 2017, pp. 39–57.

[48] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “DeepFool: A simple and accurate method to fool deep neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2574–2582.

[49] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” 2017, arXiv:1706.06083.

[50] L. Li et al., “Face X-ray for more general face forgery detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 5001–5010.

[51] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in Proc. IEEE Conf. Comput. Vis., 2017, pp. 618–626.

[52] K. Chandrasegaran, N.-T. Tran, and N.-M. Cheung, “A closer look at Fourier spectrum discrepancies for CNN-generated images detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 7200–7209.

[53] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention, Springer, 2015, pp. 234–241.

[54] J. Fridrich and J. Kodovsky, “Rich models for steganalysis of digital images,” IEEE Trans. Inf. Forensics Secur., vol. 7, no. 3, pp. 868–882, Jun. 2012.

[55] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in Proc. Eur. Conf. Comput. Vis., Springer, 2016, pp. 694–711.

[56] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

[57] P. Korshunov and S. Marcel, “DeepFakes: A new threat to face recognition? Assessment and detection,” 2018, arXiv:1812.08685.

[58] S. Ruder, “An overview of gradient descent optimization algorithms,” 2016, arXiv:1609.04747.

[59] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, no. 11, pp. 2579–2605, 2008.

Chi Liu received the BSc degree in communication engineering from Wuhan University, China, and the MSc degree in computer science from the China Ship Research and Development Academy, China. He is currently working toward the PhD degree with the School of Computer Science, University of Technology Sydney, Australia. His research interests include multimedia privacy and security, trustworthy and secure machine learning, etc.

Huajie Chen received the BSc degree from Sun Yat-sen University, China, in 2016, and the MSc degree from the Universitä Tübingenm, Germany, in 2020. He is currently working toward the PhD degree with the School of Computer Science, University of Technology Sydney, Australia. His research interests include security and privacy in deep learning, cryp- tography, and bioinformatics.

Tianqing Zhu received the BEEng and MEing degrees from Wuhan University, Wuhan, China, and the PhD degree in computer science from Deakin University, Australia, in 2014. She is currently an associate professor with the School of Computer Science, University of Technology Sydney, Australia. Prior to that, she was a lecturer with the School of Information Technology, Deakin University, from 2014 to 2018. Her research interests include privacy-preserving and AI security.

Jun Zhang (Senior Member, IEEE) received the PhD degree in computer science from the University of Wollongong, Australia. He is the co-founder and director with the Cybersecurity Lab, Swinburne University of Technology, Australia. His research interests include cybersecurity and applied machine learning. He is the chief investigator of several projects in cybersecurity, funded by the Australian Research Council (ARC). He has published more than 100 research papers in many international journals and conferences.

Wanlei Zhou (Senior Member, IEEE) received the PhD degree in computer science and engineering from the Australian National University, Canberra, Australia, in 1991, and the DSc degree (a higher doctorate degree) from Deakin University, in 2002. He is the vice rector (Academic Affairs) and dean with the Faculty of Data Science, City University of Macau, Macao SAR, China. Before joining the City University of Macau, he held various positions including the head of the School of Computer Science, University of Technology Sydney, Australia, the Alfred Deakin professor, chair of Information Technology, associate dean, and head of the School of Information Technology, Deakin University, Australia. His main research interests include security, privacy, and distributed computing. He has published more than 400 papers in refereed international journals and conference proceedings.