Supplementary Material for *TraceEvader*: Making DeepFakes More Untraceable via Evading the Forgery Model Attribution

Mengjie Wu[1], Jingui Ma[1], Run Wang[1*], Sidan Zhang[1], Ziyou Liang[1], Boheng Li[1], Chenhao Lin[2], Liming Fang[3], Lina Wang[1,4]

[1] Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, China

[2]Xi'an Jiaotong University, China [3] Nanjing University of Aeronautics and Astronautics, China

[4] Zhengzhou Xinda Institute of Advanced Technology

Abstract

In this supplementary material, we present a deep understanding of our proposed method and details of employed networks and baselines. Moreover, we also show extensive experiments to explore the effectiveness of our TraceEvader in evading the popular AttNet in the specific model instance attribution and SOTA DeepFake detector.

- We present the related work and details of the experiments setup.
- We show more visualization of samples by adding our generated adversarial perturbations, especially, compare with the competitive baseline MI-FGSM.
- We investigate extensive experiments for exploring the effectiveness of evading AttNet for model-instance attribution, and explore the potential to evade popular DeepFake detectors.

1 Related Work

1.1 Model Attribution

Model-instance attribution refers to tracing DeepFake images to a particular model instance, which is uniquely determined by its model architecture and training settings (, training data, initial seeds). Marra et al. (2019) introduce the *device* fingerprint (Luka, Fridrich, and Goljan 2006), known as the PRNU pattern, into GAN-generated images and proposes the concept of the GAN fingerprint for the first time. This GAN fingerprint is handcrafted from averaged noise residuals of images generated by GAN instances. Yu, Davis, and Fritz (2019) find that with a fixed GAN architecture, any change in model parameters leads to a distinct fingerprint. In order to extract this subtle weight-specific fingerprint, they adopt a learning-based method instead of the traditionally handcrafted fingerprint formulation. A neural network classifier AttNet and a counterpart for visualization are proposed to learn a model fingerprint per GAN instance and a content-related fingerprint per image.

Model-architecture attribution seeks to attribute generated images to the source architecture rather than the specific model instance. Frank et al. (2020) investigate traces left by the classic upsampling structure in the frequency domain. The GAN fingerprint is represented by the mean of the Discrete Cosine Transform (DCT) spectrum and used as a feature for the architecture attribution classifier. Yang et al. (2022) observes that architecture-related traces exhibit global consistency, whereas the traces left by model weights vary across different regions. Therefore, they propose the DNA-Det framework that employs pre-training on the image transformation classification task and patch-wise contrastive learning to extract architecture fingerprints. Asnani et al. (2021) introduces a reverse engineering approach to conduct model attribution. This work designs four novel loss functions to estimate the model fingerprint explicitly, from which the model architecture (, the number of layers and the type of loss function) is predicted.

As model attribution techniques are unknown to us, in this paper, we hope that our proposed evasion attack could defend the above two model attribution methods effectively and be applicable to the existing popular generative models, including GANs and DMs.

1.2 Adversarial Attacks

A black-box adversarial attack indicates that the adversary crafts adversarial examples without obtaining any knowledge of the target model and fools the victim model to return erroneous outputs. However, the white-box adversarial attack requires full knowledge of the victim model (, architecture, parameters) which is not practical in real scenarios. Here, we mainly focus on popular black-box adversarial attacks.

Traditional black-box adversarial attack can be categorized into query-based and transfer-based adversarial attacks. The query-based adversarial attack sends a large number of queries to the victim models to craft adversarial examples via gradient estimator (Ilyas, Engstrom, and Madry 2018) or optimization techniques (Chen et al. 2017; Binh et al. 2022). However, computational costs and the requirement to access the output of the model limit its practicality. Transfer-based adversarial attack trains a surrogate model and conducts a white-box attack (, BIM (Kurakin, Goodfellow, and Bengio 2016) and MI-FGSM (Dong et al. 2018)) on this model to generate adversarial perturbations, which is assumed that victim models have a similar decision

*Corresponding author. E-mail: wangrun@whu.edu.cn

Model Type	Model& DataSet	DNA-Det ↑	Reverse ↑	DCT ↑	AttNet ↑
GAN	ProGAN (Karras et al. 2017)	0.987	0.8487	0.469	0.81
	MMDGAN (Wang, Sun, and Halgamuge 2018)	1.0	1.0	0.996	1.0
	SNGAN (Miyato et al. 2018)	0.94	0.997	0.994	0.99
	CramerGAN (Bellemare et al. 2017)	1.0	0.995	0.985	1.0
	CycleGAN (Zhu et al. 2017)	1.0	0.423	0.924	0.575
	StyleGAN2 (Karras et al. 2020)	0.998	0.779	0.811	0.97
DMs	PNDM (Liu et al. 2022)	1.0	0.574	0.953	0.302
	LDM (Rombach et al. 2021)	1.0	0.248	0.633	0.978

Table 1: The performance of four model attribution methods in attributing the six GANs and two DMs.

boundary as the substitute model. Prior studies (Carlini and Farid 2020; Hussain et al. 2021) reveal that DeepFake detectors are vulnerable to adversarial attacks in both white- and black-box scenarios. In real-world applications, like model attribution, the implementation details of employed model attribution methods are mostly unavailable. Thus seeking a proper surrogate model is not an easy task.

Non-box adversarial attack attempts to attack the victim model without involving any model query or relying on a substitute model. Since no image-specific output information can be exploited, non-box apply universal perturbations or universal processes to generate adversarial examples, similar to universal adversarial perturbations (UAPs) proposed by Moosavi-Dezfooli et al. (2017). FakePolisher (Huang et al. 2020) utilizes a dictionary-based shallow reconstruction network and projects fake images onto the subspace learned from natural images. Recently, Wesselkamp et al. (2022) develop various attack variants including removing peaks from the frequency differences of fake and natural images to produce UAPs, which are shown to be effective against DeepFake detectors. However, this method is based on DeepFake common traces for differentiating real and fake. In this paper, we craft UAPs in non-box settings from the perspective of classification-dependent traces of two types of model attribution.

2 Experiments Setup

Model Attribution Techniques. In experiments, we consider the two types of model attribution, including model-architecture attribution and model-instance attribution. Specifically, we evaluate the effectiveness of our TraceEvader against the four different model attribution techniques which achieve SOTA performance in model attribution. For the model-architecture attribution, we evaluate DNA-Det (Yang et al. 2022), Reverse (Asnani et al. 2021) and DCT (Frank et al. 2020). For the model-instance attribution, we evaluate AttNet (Yu, Davis, and Fritz 2019). In order to evaluate the performance of our TraceEvader correctly, we finetune all the model attribution techniques on highly diverse and extensive data from 8 forgery models to ensure they have the best performance in model attribution. Table 1 shows the performance of four model attribution techniques evaluated in 8 GMs.

Evaluation Metrics. We employ the attack success rate (ASR) and two image quality metrics (, SSIM (Wang et al. 2004), PSNR) to evaluate the effectiveness of our TraceEvader. SSIM and PSNR are widely applied for measuring the quality of images after adding adversarial perturbations, where a higher value indicates higher image quality.

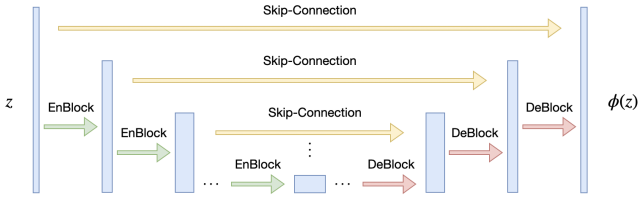


Figure 1: The architecture of the network employed for learning UITs.

Baselines. We compare TraceEvader with two commonly used adversarial attacks, the one is a transfer-based attack (, BIM (Kurakin, Goodfellow, and Bengio 2016), MI-FGSM (Dong et al. 2018)). The other one is a non-box attack that served for evading the DeepFake detectors (,peak attack in (Wesselkamp et al. 2022) and FakePolisher(Huang et al. 2020)). Specifically, the peak attack removes unusual periodic peaks detected in the frequency domain by a frequency coefficients threshold, while FakePolisher reconstructs the image to remove traces.

Implementation Details. For baselines, we specify the optimal thresholds to implement the peak attack in Table 4, which are determined following the paper (Wesselkamp et al. 2022) to reproduce the best performance. To implement BIM and MI-FGSM, we set $\epsilon = 8/255$ and the max iterations $T = 40$ and adopt the $\mu = 1.0$ in MI-FGSM. We directly use the model provided by Fakepolisher author.

In UIT generation, we employ a network ϕ based on the U-Net architecture illustrated in Figure 1. Architecture details are demonstrated in Table 2. The Encoder consists of four convolutional blocks *EnBlock* and the Decoder contains four convolutional modules *DeBlock* and a final layer. All the blocks are detailed in Table 3. The Kernel column represents the kernel size of convolutional layers, which is specified in the format *[filter height, filter width, stride]*. c_i and c_o stand for the input and output channels, where the '+' indicates that the input includes skip connections. The generator's input is a latent code z with 16 channels. The Output column describes the output shape. Conv represents the convolutional layer. BN stands for batch normalization. LReLU indicates Leaky ReLU. SC means Skip-Connection. h and w are the input shape. f_{HP} is a pre-trained DnCNN from the prior study (Zhang et al. 2017) where the weight is frozen during the generation of imitated traces. The predefined margin m in Eq.5 sets to 0.01. The learning rate is set to $5e^{-4}$, and the Adam optimizer is chosen for optimization. In our experiment, the UIT is learned from a training dataset with merely 1,000 fake images and 1,000 real images. In adding adversarial perturbations, weight factor λ is set to 0.01 for injecting the UIT and $\sigma = 8$ for the Gaussian mean shift. We choose this setting in order to ensure the PSNR value of the adversarial examples above 35dB, which is considered invisible after visual inspection.

Architecture	Layer	c_i	c_o
Encoder	z	16	16
	EnBlock	16	32
	EnBlock	32	64
	EnBlock	64	128
	EnBlock	128	256
Decoder	DeBlock	256+256	128
	DeBlock	128+128	64
	DeBlock	64+64	32
	DeBlock	32+32	32
	Final_Layer	32	3

Table 2: Network architecture of the encoder and decoder.

Architecture	Layer	Kernel	Output
EnBlock	Conv	[3,3,1]	$c_o \times h \times w$
	BN,LReLU	-	$c_o \times h \times w$
	Conv	[3,3,1]	$c_o \times h \times w$
	BN,LReLU	-	$c_o \times h \times w$
	Maxpool	[2,2,2]	$c_o \times h/2 \times w/2$
DeBlock	SC	-	$2c_i \times h \times w$
	Conv	[3,3,1]	$c_o \times h \times w$
	BN,LReLU	-	$c_o \times h \times w$
	Conv	[3,3,1]	$c_o \times h \times w$
	BN,LReLU	-	$c_o \times h \times w$
	DeConv	[2,2,2]	$c_o \times 2h \times 2w$
	BN	-	$c_o \times 2h \times 2w$
Final_Layer	Conv	[1,1,1]	$c_o \times h \times w$
	TanH	-	$c_o \times h \times w$

Table 3: EnBlock, DeBlock and Final_Layer architecture.

3 Extensive Experimental Results

3.1 Ablation Study

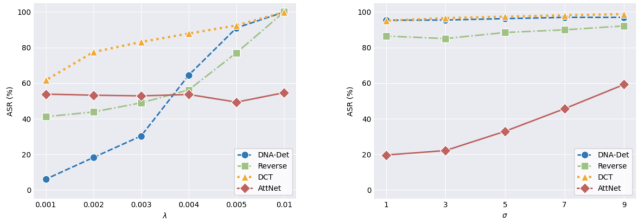
The effect of sole adversarial perturbations. Here, we explore the impact of each attack pattern (, UIT for HFC, adversarial blur for LFC) on the attack effectiveness. As experimental results shown in Table 5, when attacking DNA-Det, only applying UIT achieves the equally high attack success rate as the proposed method in most cases. When attacking AttNet, only applying adversarial blur produces a similar attack performance to the proposed attack. It can be concluded that the sole UIT is effective in attacking discriminators that use high-frequency components for attribution but have limited attack performance to evade discriminators that use fingerprint traces in low-frequency components for attribution, while the sole adversarial blur exposes the opposite results. Hence, they both yield limited effectiveness. Our hybrid adversarial attack has successfully achieved prominent attack performance, which indicates that our method combines the benefits from each design and integrates them compatibly.

GMs	ProGAN	MMDGAN	SNGAN	CramerGAN	CycleGAN	StyleGAN2	PNDM	LDM
Threshold	0.65	0.25	0.4	0.2	0.7	0.5	0.95	0.6

Table 4: the optimal threshold in the peak attack for each GM.

GMs	Attack patterns	DNA-Det	Reverse	DCT	AttNet
ProGAN	only UIT	96.9	86	100	4.2
	only blur	40.7	12.5	90.0	59.0
	UIT & blur	96.9	92.3	99.6	59.0
MMDGAN	only UIT	98.5	81.5	88.2	8.0
	only blur	0	0	25.8	88.2
	UIT & blur	98.7	78.8	96.4	82.7
SNGAN	only UIT	100	97.3	56.2	3.9
	only blur	36.2	2	49.6	72
	UIT & blur	100.0	98.5	65.8	71.0
CramerGAN	only UIT	95.9	99.8	100	4.9
	only blur	0	0	0	51.0
	UIT & blur	97.5	100.0	93.4	48.2

Table 5: Performance of sole UIT and adversarial blur on four model attribution techniques with four GANs.



(a) λ value (b) σ value

Figure 2: The ASR (%) of different weight factors λ and σ against four attribution techniques.

The effect of weight factor λ for HFC adversarial perturbations. We discuss the impact of the intensity of adversarial UIT which is controlled by weight factor λ in determining the final model attribution evading. Specifically, we set λ from 0.001 to 0.01 and investigate the attack success rate on a forgery model ProGAN. As shown in Figure 2(a), the attack success rate of AttNet remains stable, while the attack success rate of attacks on the other three methods increases as the degree of perturbation increases. We attribute the phenomenon to the fact that AttNet utilizes weight traces mostly distributed in the LFC of the image, and its performance is marginally influenced by perturbations in the high-frequency components. In contrast, the attack success rate on other methods relying on traces in the high frequency will increase with an increase in the value of λ.

The effect of σ for LFC adversarial perturbations. In this part, we examine how the degree of blurring affects the success rate of attacks. We tune the Gaussian kernel width σ from 1 to 9. As shown in Figure 2(b), the degree of blur operation has a significant impact on instance-level attribution AttNet. However, the attack success rates are marginally enhanced for the architecture-level attribution methods.

3.2 Evading Model-instance

To further explore the vulnerability of model instance attribution, we train the AttNet to attribute 10 ProGAN instances

Figure 3: Confusion matrice on attributing cross-seed ProGAN instances.

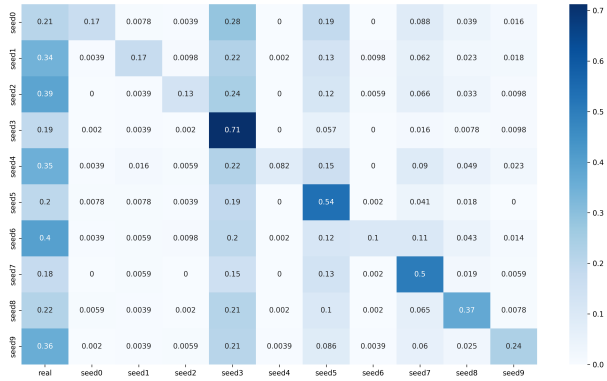


Figure 4: Confusion matrice on attributing the samples with our added adversarial perturbations.

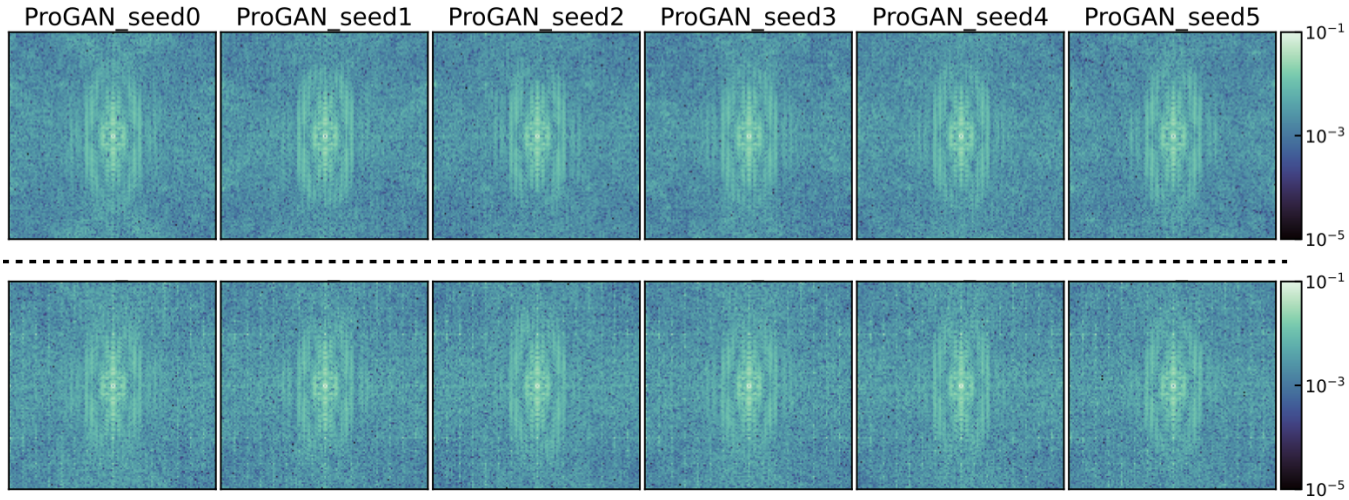


Figure 5: Spectrum analysis for different ProGAN instances, including 6 ProGANs with distinct initialization seeds. We visualize the mean DFT spectra of high-pass filtered (top row) generated raw fake images and (bottom row) images added with our generated adversarial perturbations.

with different initialization seeds, denoted as *seed#i*. Figure 3 and 4 illustrate the difference in model-instance attribution before and after applying the TraceEvader attack. Obviously, our TraceEvader could evade the AttNet for model-instance attribution effectively. The detection success rate dropped by an average of **61.5%** and **90.8%** in the worst case. Figure 5 visualizes the effect of the adversarial perturbation on the frequency domain of images. Experimental results provide additional evidence that our attack can exploit the vulnerability of the model instance attribution method.

3.3 Evading DeepFake Detectors

Here, we explore whether our proposed TraceEvader shows the potentials to evade the popular DeepFake detectors. Table 6 shows the performance of our proposed method in evading the spatial-based, frequency-based and fingerprint-based DeepFake detection methods, which are DCT, CNNDetection(Wang et al. 2020) and AttNet. The experimental results show that our method gives an average attack success rate of more than 43.3% against the three DeepFake detection methods over four different forgery methods. It achieves comparable performance to other non-box methods specifically designed for DeepFake detection and even outperforms these baselines in some cases.

GM&Detector	ProGAN			MMDGAN		
Methods	DCT	CNNDetection	AttNet	DCT	CNNDetection	AttNet
Vera22-peak	82.2	44.9	0	0.6	24.5	0.2
FakePolisher	92.4	47.5	25.7	22.0	30.1	**38.2**
TraceEvader	**97.0**	**49.0**	48.8	**41.2**	**33.0**	20.9
GM&Detector	SNGAN			CramerGAN		
Methods	DCT	CNNDetection	AttNet	DCT	CNNDetection	AttNet
Vera22-peak	**72.8**	**62.4**	0.2	0.2	20.7	0
FakePolisher	55.4	60.5	**46.2**	26.4	30.3	**43.4**
TraceEvader	56.1	61.9	30.4	**82.3**	**55.4**	4.5

Table 6: The performance of four adversarial attack in evading Deepfake detection.

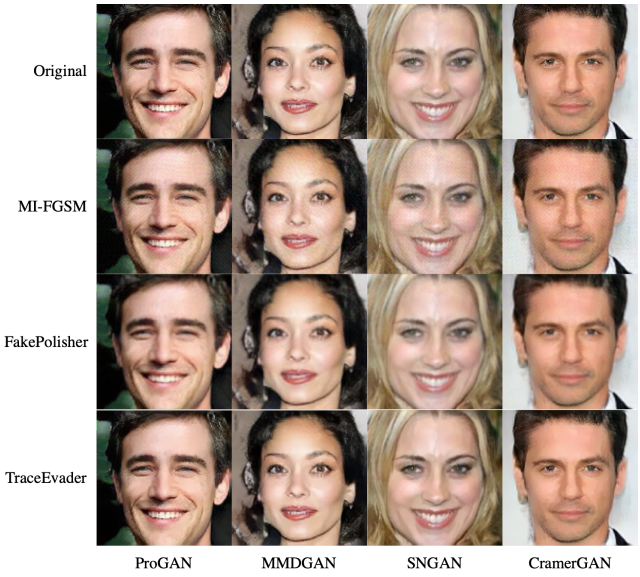


Figure 6: Visualization of samples added by our generated adversarial perturbations. The row *Original* indicates the clean images without any perturbations. The row *MI-FGSM* represents the samples added adversarial perturbations generated by MI-FGSM. The row *FakePolisher* displays the images reconstructed by FakePolisher. The row *TraceEvader* denotes the images added by our TraceEvader.

3.4 Visualization of Crafted Samples

Due to the limited page in the main manuscript, we present more visualization of our crafted samples with our generated adversarial perturbations and the competitive baseline MI-FGSM. Specifically, we show the clean image, and perturbated images with the strongest baselines MI-FGSM and TraceEvader. Figure 6 shows that samples with perturbations generated by MI-FGSM expose more noticeable distortions and images reconstructed by FakePolisher exhibit a noticeable blurriness. In contrast, samples with our TraceEvader are less perceptible without any artifacts.

4 Discussion

Limitations. Studies have shown that the parameters of machine learning models could be inferred by leveraging the available intermediate hardware information, such as the information leaks of memory access and side-channel information like timing and electromagnetic emanations (Hua, Zhang, and Suh 2018; Batina et al. 2019). Our proposed TraceEvader is limited to the disruption of traces left in DeepFake creation and fails in tackling the aforementioned model reverse engineering techniques. However, the hardware information is mostly unavailable to the defender and the performance in inferring our generative models is unclear. How to evade the aforementioned model attribution technique could be our future work.

Social impacts. Our TraceEvader injects adversarial perturbations to evade the popular model attribution techniques without compromising image naturalness, which poses a real threat to current DeepFake attribution techniques. We hope that our work can inspire more general solutions to robust DeepFake attribution schemes toward addressing this common trace tampering. This stealthy and transferable attack can contribute to improving the robustness of model attribution techniques through adversarial training. Besides, the generated imitated traces in our attack can be used for data augmentation in the fingerprint domain to improve the generalizability of DeepFake detectors, as well as the robustness of attribution techniques through adversarial training.

References

Asnani, V.; Yin, X.; Hassner, T.; and Liu, X. 2021. Reverse Engineering of Generative Models: Inferring Model Hyperparameters from Generated Images. arXiv:2106.07873.

Batina, L.; Bhasin, S.; Jap, D.; and Picek, S. 2019. CSI NN: Reverse engineering of neural network architectures through electromagnetic side channel.

Bellemare, M. G.; Danihelka, I.; Dabney, W.; Mohamed, S.; Lakshminarayanan, B.; Hoyer, S.; and Munos, R. 2017. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*.

Binh, N. T. M.; Hoang Long, D.; Ngoc, N.; Thanh Binh, H. T.; and Phuong, N. K. 2022. Investigate Evolutionary Strategies for Black-Box Attacks to Deepfake Forensic Systems. In *Proceedings of the 11th International Symposium on Information and Communication Technology*, SoICT '22, 126–133. New York, NY, USA: Association for Computing Machinery. ISBN 9781450397254.

Carlini, N.; and Farid, H. 2020. Evading Deepfake-Image Detectors With White- and Black-Box Attacks. In *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition (CVPR) Workshops*.

Chen, P.-Y.; Zhang, H.; Sharma, Y.; Yi, J.; and Hsieh, C.-J. 2017. ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks without Training Substitute Models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting Adversarial Attacks with Momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Frank, J.; Eisenhofer, T.; Schönherr, L.; Fischer, A.; Kolossa, D.; and Holz, T. 2020. Leveraging Frequency Analysis for Deep Fake Image Recognition. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Hua, W.; Zhang, Z.; and Suh, G. E. 2018. Reverse engineering convolutional neural networks through side-channel information leaks. In *Proceedings of the 55th Annual Design Automation Conference*, 1–6.

Huang, Y.; Juefei-Xu, F.; Wang, R.; Guo, Q.; Ma, L.; Xie, X.; Li, J.; Miao, W.; Liu, Y.; and Pu, G. 2020. FakePolisher: Making DeepFakes More Detection-Evasive by Shallow Reconstruction. In *Proceedings of the 28th ACM International Multimedia*.

Hussain, S.; Neekhara, P.; Jere, M.; Koushanfar, F.; and McAuley, J. 2021. Adversarial Deepfakes: Evaluating Vulnerability of Deepfake Detectors to Adversarial Examples. In *Proceedings of the IEEE/CVF Winter Applications of Computer Vision (WACV)*, 3348–3357.

Ilyas, A.; Engstrom, L.; and Madry, A. 2018. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and Improving the Image Quality of StyleGAN. In *Proc. CVPR*.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*.

Liu, L.; Ren, Y.; Lin, Z.; and Zhao, Z. 2022. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*.

Luka, J.; Fridrich, J.; and Goljan, M. 2006. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2): 205–214.

Marra, F.; Gragnaniello, D.; Verdoliva, L.; and Poggi, G. 2019. Do GANs leave artificial fingerprints. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*.

Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1765–1773.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2021. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752.

Wang, S.-Y.; Wang, O.; Zhang, R.; Owens, A.; and Efros, A. A. 2020. CNN-generated images are surprisingly easy to spot...for now. In *CVPR*.

Wang, W.; Sun, Y.; and Halgamuge, S. 2018. Improving MMD-GAN training with repulsive loss function. *arXiv preprint arXiv:1812.09916*.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Wesselkamp, V.; Rieck, K.; Arp, D.; and Quiring, E. 2022. Misleading Deep-Fake Detection with GAN Fingerprints. In *2022 IEEE Security and Privacy Workshops (SPW)*, 59–65.

Yang, T.; Huang, Z.; Cao, J.; Li, L.; and Li, X. 2022. Deepfake Network Architecture Attribution. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI 2022)*.

Yu, N.; Davis, L.; and Fritz, M. 2019. Attributing Fake Images to GANs: Learning and Analyzing GAN Fingerprints. In *IEEE International Conference on Computer Vision (ICCV)*.

Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; and Zhang, L. 2017. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7): 3142–3155.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on computer vision*.