Self-Supervised Adversarial Training for Robust Face Forgery Detection

Yueying Gao
gaoyueying@cuc.edu.cn
Weiguo Lin
linwei@cuc.edu.cn
Junfeng Xu†
junfeng@cuc.edu.cn
Wanshan Xu
xws@cuc.edu.cn
Peibin Chen
chenpeibin@cuc.edu.cn

School of Computer and Cyberspace Security
Communication University of China
Beijing, China

Abstract

With the advancement of face forgery technologies, the high-fidelity generation and substitution of human faces have become increasingly prevalent, leading to an emerging research topic of face forgery detection. Despite the outstanding performance of current face forgery detectors in benchmark datasets, their real-life application is fraught with challenges due to complex scenarios. Therefore, we propose a self-supervised adversarial training network to enhance the robustness of face forgery detection, promoting their applicability in real-life scenarios. We generate multiple adversarial examples using a pool of attack strategies and strengthen the sensitivity to perturbations by compelling the model to predict these attack strategies. Additionally, we employ an adversarial training strategy to dynamically generate the most challenging adversarial examples for the current model. A fast training strategy is proposed to reduce the computation cost of adversarial training. Through extensive experiments, we demonstrate that our approach significantly outperforms the baseline and state-of-the-art methods in terms of robustness to perturbations.

1 Introduction

Auto-Encoders [9, 17, 24] and the generative adversarial network (GAN) [11, 26, 30] have emerged as powerful tools for face forgery, enabling high-fidelity generation and substitution of human faces with minimal complexity. Current face forgery detectors [8, 25, 29] exhibit outstanding performance in benchmark datasets. However, their real-life application faces significant challenges due to complicated scenarios, such as low resolution, multiple faces, adversarial perturbations and corruptions caused by image processing operations. Therefore, developing robust face forgery detectors remains a critical area of research.

†Corresponding author

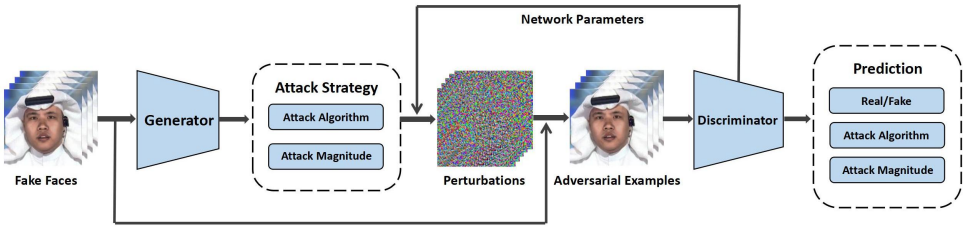


Figure 1: Overview of our approach. The generator takes forgery faces as input and outputs the attack strategies that are further used to generate adversarial examples. The discriminator is designed to not only predict the authenticity of inputs but also the attack strategies employed. Adversarial training is proposed between the generator and the discriminator.

Attempts have been made in recent arts to enhance the robustness. Some studies [21, 29] suggest compressed artifacts exhibit greater stability at high frequency, and focus on mining frequency features. However, these approaches are not generalizable across real-life scenarios, as frequency features can vary significantly between them. Besides, adversarial training has been utilized to improve the robustness of neural networks at the learning mechanism level [12, 37, 42]. But conventional adversarial training [12, 14, 23] is limited by its ability to defend against certain types of adversarial attacks. And the generation of adversarial examples can be computationally expensive, particularly in the context of video-level face forgery detection.

Therefore, we propose a self-supervised adversarial training network. The pipeline of our method is shown in Figure 1. To generate multiple adversarial examples for adversarial training, the generator is introduced to take fake faces as input and output the corresponding attack strategies, which include both the algorithms and the magnitude of perturbations. In addition to predict the authenticity, the discriminator is also required to predict the attack strategy of an input, which improves its sensitivity to perturbations and maximizes the utilization of the adversarial space. We employ an adversarial training framework to dynamically generate the most challenging adversarial examples for the current discriminator. Furthermore, to reduce the computation cost of adversarial training, we propose a fast training strategy by saving network parameters for generating adversarial examples of subsequent frames.

After comparing our approach to the baseline Xception [8] using the ROC curve, it is evident from the results shown in Figure 3 that our approach is highly effective. Extensive experiments demonstrate that our approach significantly outperforms current state-of-the-art methods in terms of robustness to perturbations.

2 Related Work

Face Forgery Detection. Recent studies have made various attempts for face forgey detection. Specifically, classic network structures such as Xception [8] and EfficientNet [34] have been utilized for face forgery detection in the spatial domain. Other studies [7, 29] have focused on high-frequency information that is more stable in artifacts and have achieved promising results for highly compressed videos. Li *et al.* [20] have proposed a novel ap-

proach that does not rely on forged samples, but instead targets the fusion traces for detection. However, current detectors are restricted to constrained datasets, which limits their applicability in real-life scenarios.

Adversarial Training. Adversarial examples are images that have been modified with imperceptible perturbations, causing neural networks to produce inaccurate results [33]. Adversarial training involves training models with adversarial examples as a form of data augmentation. Goodfellow *et al.* [12] first proposed the fast gradient sign method in adversarial training, which improved the robustness of models. Madry *et al.* [23] introduced the PGD attack and clarified the concept of adversarial training as a min-max optimization. However, generating adversarial examples can be computationally expensive, and there is a trade-off between model accuracy and robustness in adversarial training [38].

3 Method

In this section, we introduce our approach in four distinct parts: the generation of adversarial examples, self-supervised tasks, adversarial training and the fast training strategy.

3.1 Generating Adversarial Examples

We select four distinct types of adversarial attacks: gradient-based BIM [18], hyperplane-based Deepfool [27], optimization-based C&W [5], and black-box attack using gradient estimation [15]. These adversarial attacks are labeled as $A_g \in \{0,1,2,3,4\}$, where 4 represents a no-attack choice to use the original fake faces. The distance between adversarial and original examples measured by L_∞ metric is represented by M_g. To ensure that adversarial perturbations are imperceptible, we set the continuous value of $M_g \in [-0.3, 0.3]$ [12].

BIM. Following [5, 28], we define F to be the full neural network, including the softmax function, $Z(x)$ to be the output of all layers except the softmax. We adopt the loss function in line with previous work [5], which is optimized using the iterative gradient sign method [18], while also constraining the magnitude of the perturbation:

$$\begin{aligned} &minimize \;\; loss(x) = max(Z(x)_{fake} - Z(x)_{real}, 0) \\ &such\ that \;\; x_i = x_{i-1} - clip_\varepsilon(\alpha \cdot sign(\nabla loss(x_{i-1}))) \end{aligned} \tag{1}$$

where the function *clip* performs per-pixel clipping, and the result will be in the L_∞ ε-neighbourhood of the input x. In our experiments, we use α = 1/255 to change the value of each pixel. We continue with gradient descent iterations until we achieve attack success or reach the maximum number of iterations.

C&W. The process of generating adversarial examples for fake faces can be defined as finding the shortest distance between the adversarial and original examples. However, since the discriminator network is highly non-linear, we define an objective function f such that $F(x+\delta) = real$ if and only if $f(x+\delta) \leq 0$.

As suggested in [5], we use the L_2 metric to measure the distortion, introduce a new variable h as a box-constraint, and use the best objective function to reformulate the problem:

$$minimize \;\; ||\frac{1}{2}(tanh(h)+1) - x||_2^2 + c \cdot f(\frac{1}{2}(tanh(h)+1)) \tag{2}$$

with f defined as:

$$f(x) = max(Z(x)_{fake} - Z(x)_{real},\ \kappa) \tag{3}$$

Algorithm 1 Deepfool

Input: Image x,classifier F ▷ x is an image, F is a binary classifier
Output: Adversarial example $\hat{x}$ ▷ $\hat{x}$: with the minimum perturbation
Initialize $x_0 \leftarrow x, \quad i \leftarrow 0$ ▷ Keep iterating over the x_i
while $F(x_i) = F(x_0)$ **do** ▷ Stop when x_i changes the sign of F
  $\delta_i \leftarrow -\frac{F(x_i)}{||\nabla F(x_i)||_2^2}\nabla F(x_i)$ ▷ F is linearized around the current point x_i
  $x_{i+1} \leftarrow x_i + S_i$ ▷ x_i adds this distance
  $i \leftarrow i+1$ ▷ Enter the next iteration
end while ▷ End when switches to the other side of the hyperplane
return $\hat{x} = x_i$ ▷ Return the adversarial example of minimum perturbation

Notice that we have added a constant c = 0.001 to the objective function to ensure that it respects our definition. Additionally, we can control the confidence by adjusting the value of κ, which is set to 20 in our experiment.

Deepfool. As a binary classification, the detection of face forgery relies on the classifier hyperplane. Deepfool [27] suggests computing the shortest distance between the original examples and the classification boundary, i.e. $\Delta(x;\ F)^2$. An iterative approach is employed to estimate $\Delta(x;\ F)$, with the adversarial examples being updated until they are misclassified. The Deepfool algorithm is outlined in Algorithm 1.

Black-box Attack. In the case of a black-box attack, it is assumed that the attacker has no access to the network architecture or parameters. Therefore, NES [15] proposes maximizing the class probability $F(x)_{real}$ under a search distribution, which enables gradient estimation in far fewer queries than typical finite-difference methods.

$$maximize\ \ E_{\pi(\phi|x)}[F(\phi)_y] \tag{4}$$

where $\pi(\phi|x)$ denotes a a search distribution of random Gaussian noise around input x. That is, we have $\phi = x + \sigma\zeta$, where $\zeta \sim \mathcal{N}(0, I)$. Once the gradient is estimated, we generate adversarial examples using BIM.

3.2 Self-Supervised Tasks

To strengthen the sensitivity to perturbations, our approach involves introducing auxiliary tasks to predict both the algorithm and the magnitude of perturbations of adversarial examples. We design separate loss functions for each task and combine them using appropriate weighting during optimization.

Main task loss L_{main}. The main task is a binary classification task that involves distinguishing between real and forgery images. To achieve this objective, we have adopted the AM-Softmax Loss function, as recommended by [22]. This loss function has been shown to be effective in reducing intra-class differences and increasing inter-class distances, thereby improving the accuracy of classification.

Attack type estimation loss L_T. In addition to the main task, we have introduced a secondary task estimating the attack algorithm of input images. The label is denoted by A_d,

where $A_d = A_g$ for adversarial examples, $A_d = 4$ for original fake faces, and $A_d = 5$ for real faces. The AM-Softmax Loss function is also used to compute L_T.

Attack magnitude estimation loss L_M. To fully leverage the adversarial domain, we have introduced a third task estimating the attack magnitude of input images. The label is denoted by M_d, which is valid for BIM and black-box attack. And L_1 loss is adopted to compute L_M:

$$L_M = \tau \times ||M_d - M_e||_1 \tag{5}$$

The estimated attack magnitude from the discriminator is denoted by M_e, and we use a binary value τ to determine the relevance of this magnitude estimate. Specifically, τ is set to 1 if $A_d = 1$ or $A_d = 3$, and 0 otherwise. By incorporating this binary value, our approach aims to ensure that the estimated attack magnitude is only used for BIM and black-box attack.

3.3 Adversarial Training

Our approach involves the use of a generator, denoted as $G(\theta)$, and a discriminator, denoted as $D(w)$. The generator's primary objective is to output attack strategies that maximize the training loss, while the discriminator aims to minimize the loss that is maximized by the generator. This optimization process can be formulated as follows:

$$\min_{w} \max_{\theta} \; Loss(\theta, w) = L_{main} + \mu L_T + \lambda L_M \tag{6}$$

To solve this problem, we propose an iterative approach updating the discriminator and the generator in an alternating fashion. Regarding the discriminator $D(w)$, its training process can be formulated as a minimization problem with respect to w, which is updated using gradient descent.

$$w^{t+1} = w^t - \eta \cdot \nabla_w Loss(\theta^t, w^t) \tag{7}$$

In the above equation, η denotes the learning rate used in the gradient descent optimization process. Regarding the generator $G(\theta)$, its training process can be formulated as a maximization problem with respect to θ. The generator plays a zero-sum game in an adversarial framework with the discriminator. Mathematically, this can be formulated as follows:

$$\theta^{t+1} = arg\max_{\theta^t} \; Loss(w^{t+1}, \theta^t) \tag{8}$$

Updating the generator $G(\theta)$ by solving the above formulation is problematic due to the non-differentiable sampling operations that break the gradient flow from $D(w)$ to $G(\theta)$. To address this issue, we apply the REINFORCE algorithm [40] to approximate the gradient calculation.

3.4 Fast Training Strategy

The adversarial training for deepfake detector can be optimized due to several properties of face forgery datasets. In particular, videos have a short duration and show minimal variation in facial features between consecutive frames. Using consecutive frames as input does

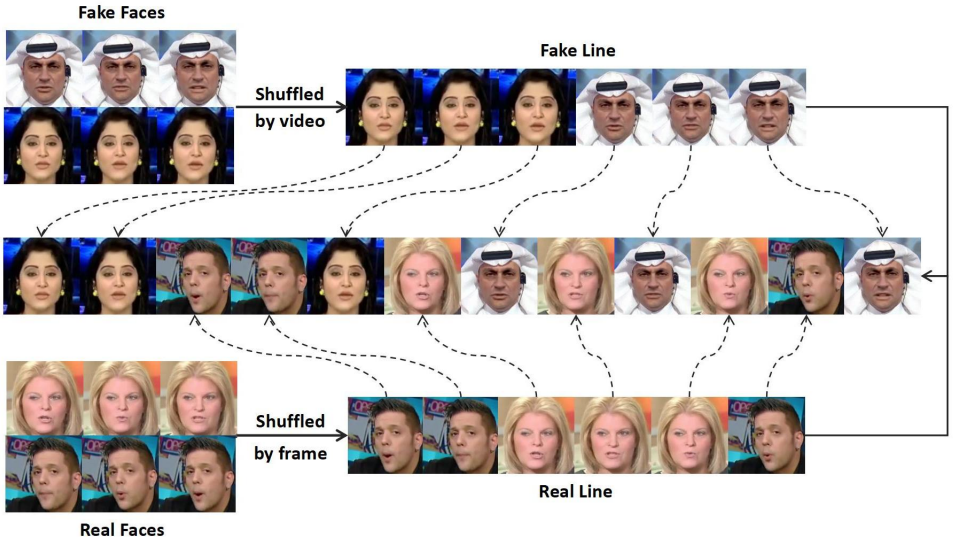


Figure 2: Overview of the data loading process in the fast training strategy. Real images are shuffled by frames and fake images are shuffled by videos. We randomly insert the fake images into the real images in order.

not significantly affect network parameters, such as gradients and logit outputs, and can be considered redundant computations.

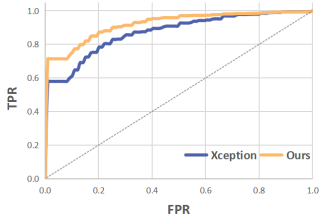
Therefore, we shuffle real images based on frames, while fake images based on videos, and randomly insert the fake images into the real image sequence in order, as shown in Figure 2. Shuffling samples avoids model bias caused by the concentration of positive and negative samples. Meanwhile, we maintain the relative order of frames in fake videos when generating adversarial examples.

For each image in a given batch, no attack is performed if it is real. However, if the image is fake, we compare it with the previous fake image. If both images belong to the same video, we generate the adversarial example using the network parameters of the previous image. On the other hand, if the images do not belong to the same video, we attack the discriminator and save its current parameters. It is important to note that the fast training strategy is not applicable for black-box attacks since it does not have access to network parameters. The impact of the fast training strategy on both training cost and performance can be found in Section 5.3.

4 Experiment

4.1 Settings

Datasets. Following recent face forgery detectors [19, 22, 39], we have trained our model using the Faceforencis++ dataset [31]. This dataset comprises of 1,000 real videos and 4,000 manipulated videos generated by Deepfakes [1], Face2Face [35], FaceSwap [2], and



(a)

Methods	RAW	HQ	LQ
Xception [8]	0.989	0.961	0.895
Face X-ray [20]	0.988	0.866	0.631
SBI [32]	**0.994**	0.950	0.903
Two-branch [25]	0.981	0.958	0.890
Cvit [41]	-	0.937	-
Ours	0.992	**0.988**	**0.934**

(b)

Figure 3: (a) ROC Curve of our approach and baseline Xception [8] on heavily compressed(LQ) videos in FF++ [31]. (b) Robustness comparisons to different compressed levels in term of AUC.

NeuralTextures [36]. To evaluate the robustness of face forgery detection, we employ the DeeperForensics dataset [16], which introduces real-world disruptions by adding block-wise distortions, White Gaussian noise, blurring, and other corruptions to real videos from the Faceforencis++ dataset [31].

Implemenation details. We modify Xception [8] to serve as the backbones for both the generator and discriminator, which are optimized by the Adam. The learning rates for the discriminator and generator networks are set to 0.002 and 0.0005, respectively. The hyper-parameter in loss function are $\mu = 0.05$ and $\lambda = 0.05$. We set the maximum number of iterations to 10 for BIM and black-box attack, and 50 for C&W and Deepfool attacks.

4.2 Robustness to Compressed Images

In this experiment, we train and test separate models for each of the compression levels in FF++ [31]: (1)uncompressed videos(raw), (2)compressed videos at high quality(HQ), and (3) compressed videos at low quality(LQ). Because most of the videos on social media undergo standardised compression operations such as H.264 [29], the experiment setting is necessary for real-life applications.

Figure 3 shows all methods perform well on RAW videos. However, as the compression level increases, some methods experience a significant drop in performance. This outcome is not surprising, given that the blending artifacts detected by Face X-ray [20] and the low-level clues detected by Cvit [41] are largely destroyed when images are highly compressed. Two-branch [25] maintains its performance on compressed videos due to the inclusion of the temporal dimension. In addition to textural features, which are focused by Xception [8] and other CNN-based methods, our approach also emphasizes shape and global features through adversarial training, resulting in a significant performance boost under high-level compression.

4.3 Robustness to Perturbed Images

Given the ubiquity of perturbation in real-life scenarios, face forgery detectors should not be easily subverted by extreme inputs. We train the methods on RAW videos in FF++ [31] and test on perturbed videos in DeeperForensics [16]. Following [13], we employ corruptions, such as White Gaussian noise, block-wise distortion, and Gaussian blur, to simulate real-life perturbations. These corruptions are applied at five different levels, with level 3 being used as the test set. In addition, we incorporate adversarial perturbations into the original videos.

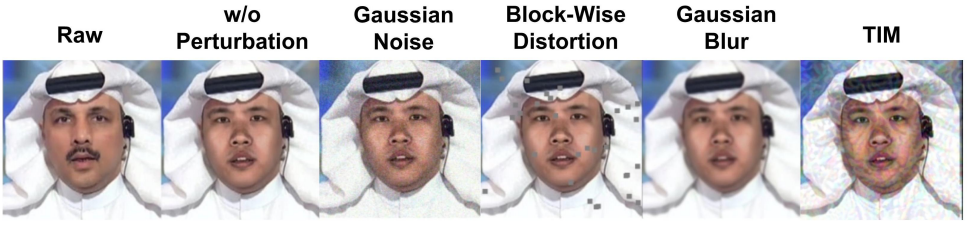


Figure 4: Samples of the perturbed images, including various corruptions at severity level 3 and adversarial example of TIM [10].

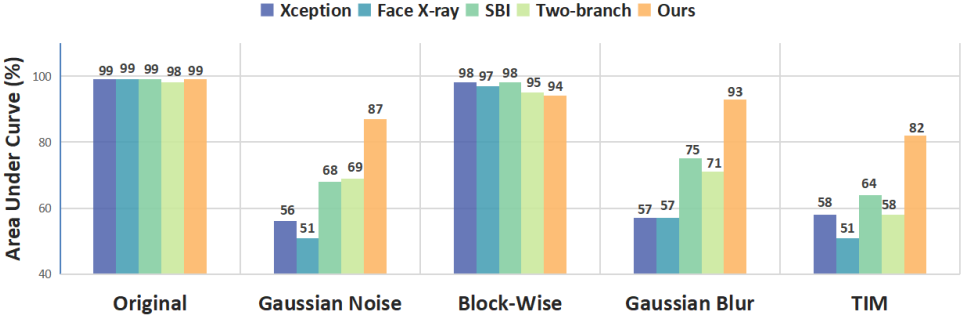


Figure 5: Robustness comparisons to perturbations in term of AUC. Our approach is more robust than others when faced with perturbations.

Given that adversarial training is built upon several seen attacks, we subsequently conduct robustness experiments against unseen attack TIM [10]. Figure 4 presents samples of each perturbation.

Figure 5 illustrates the impact of various perturbations on the performance. Methods are relatively unaffacted by block-wise distortions. However, adding white Gaussian noise and Gaussian blur perturbs the feature extraction process in convolutional neural networks, resulting in a significant deterioration in performance. In contrast, our approach is more robust than others when faced with perturbations. The outstanding performance attributes to the regularization of gradients through adversarial training. This regularization constrains the network parameters to reduce its sensitivity to perturbations.

4.4 Robustness to Different Resolutions

To investigate the impact of different image resolutions on performance, we trained and tested separate models on RAW videos in FF++ [31] resized to three different resolutions, ranging from 256×256 to 64×64.

Based on the experimental results presented in Table 1, our approach outperforms all other methods at lower resolutions. The resizing operation significantly affects the high-frequency content of images, which causes neural networks to fail to extract edge and detail

Methods	256*256	128*128	64*64
Xception [8]	0.989	0.851	0.713
Face X-ray [20]	0.988	0.878	0.755
SBI [32]	**0.994**	0.908	0.760
Two-branch [25]	0.981	0.904	0.778
Ours	0.992	**0.947**	**0.859**

Table 1: Robustness comparisons to different resolutions in term of AUC. Our approach is more robust than other methods at lower resolutions.

L_{main}	L_A	L_M	RAW	HQ	LQ	Mix
✓	-	-	0.989	0.967	0.895	0.865
✓	✓	-	0.990	0.981	**0.947**	0.897
✓	-	✓	0.989	0.979	0.923	0.868
✓	✓	✓	**0.992**	**0.988**	0.934	**0.914**

Table 2: Ablation study of the effectiveness of self-supervised tasks in term of AUC.

information for forgery detection. However, adversarial training can remove the dense mixture and purify hidden weights [4], thereby improving the robustness to changes in resolution.

5 Ablation Study

In this section, we investigate the effectiveness of self-supervised tasks, adversarial training, and the fast training strategy.

5.1 Ablation Study on Self-Supervised Tasks

To demonstrate the benefits of self-supervised tasks in enhancing robustness, we incorporate auxiliary tasks by setting the weight parameters μ and λ in Equation 6 to zero [6]. We conduct separate training and testing for each compression level in FF++ [31]. Additionally, we train the variant on RAW videos in FF++ [31] and evaluate its performance on mixed perturbed videos in DeeperForensics [16].

The results are presented in Table 2, where it can be observed that each of the auxiliary tasks contributes to the improvement of the baseline performance. And our approach, which incorporates both of the auxiliary tasks, achieves the best performance. Moreover, the exclusion of any of the auxiliary tasks results in a lower performance. These demonstrate the effectiveness of self-supervised tasks in fully leveraging the information from adversarial space and achieving a higher level of image understanding.

5.2 Ablation Study on Adversarial Training

To evaluate whether adversarial training can boost the robustness, we conduct the experiment on following variants. (1)Xception [8]: baseline, from which adversarial training and self-supervised tasks are excluded. (2) Xception w/ adv(1): we add adversarial augmentation using BIM to baseline. (3)Xception w/ adv(4): we add adversarial augmentation using

Strategy	RAW	HQ	LQ	Mix
Xception [3]	0.989	0.961	0.895	0.514
Xception w/ adv(1)	0.992	0.966	0.903	0.78
Xception w/ adv(4)	**0.993**	0.970	0.933	0.803
Ours w/o adv	0.989	0.981	0.915	0.877
Ours	0.992	**0.988**	**0.934**	**0.914**

Table 3: Ablation study of the effectiveness of adversarial training in term of AUC.

Strategy	AUC	acc	Hours/epoch
Fast Training	0.897	86.6	5.65
Original Training	0.902	87.2	20.33

Table 4: Ablation study of the effectiveness of fast training strategy.

random four attack algorithms in Section to baseline. (4)Ours w/o adv: we random select attack strategies to generate adversarial examples instead of the generator.

The evaluation results are shown in the Table 3. The Xception w/ adv(1) variant demonstrates that adversarial augmentation enhances the robustness of the baseline. Furthermore, augmentation with multiple adversarial examples achieves a better performance. Additionally, the exclusion of adversarial training from our method results in a significant decline in robustness, proving the crucial role of adversarial training in our approach.

5.3 Ablation Study on Fast Training Strategy

We conducted separate training using both the original and fast training strategy on one-tenth of the LQ videos in FF++ [31]. All experiments were conducted on a single GeForce RTX 3090 using the same environment. The quantitative results are presented in the Table 4. The fast training strategy achieves similar performance to the original training strategy while significantly reducing the training cost by 72.2%.

6 Conclusion

This paper presents a novel approach for face forgery detection, which is specifically designed for real-life scenarios. To generate multiple adversarial examples, we introduce a generator and self-supervised tasks. Furthermore, we propose an adversarial training framework that dynamically generates the most challenging adversarial examples for the current discriminator. Through this training game between the generator and the discriminator, the network improves its sensitivity to perturbations and maximizes the use of the forgery and adversarial space. Our extensive experiments demonstrate the superior robustness of our approach, which represents an important step towards real-life applications for face forgery detection.

References

[1] faceswap. www.github.com/deepfakes/faceswap, 2021. Accessed 2021-04-24.

[2] Faceswap. www.github.com/MarekKowalski/FaceSwap, 2021. Accessed 2021-04-24.

[3] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE international workshop on information forensics and security (WIFS)*, pages 1–7. IEEE, 2018.

[4] Zeyuan Allen-Zhu and Yuanzhi Li. Feature purification: How adversarial training performs robust deep learning. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 977–988. IEEE, 2022.

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[6] Liang Chen, Yong Zhang, Yibing Song, Lingqiao Liu, and Jue Wang. Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18710–18719, 2022.

[7] Shen Chen, Taiping Yao, Yang Chen, Shouhong Ding, Jilin Li, and Rongrong Ji. Local relation learning for face forgery detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1081–1088, 2021.

[8] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[9] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[10] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[13] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Lips don't lie: A generalisable and robust approach to face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5039–5049, 2021.

[14] Yuying Hao, Tuanhui Li, Yong Jiang, Xuanye Cheng, and Li Li. Defending against adversarial examples using defense kernel network. In *BMVC*, page 77, 2019.

[15] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International conference on machine learning*, pages 2137–2146. PMLR, 2018.

[16] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2889–2898, 2020.

[17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[19] Jiaming Li, Hongtao Xie, Jiahong Li, Zhongyuan Wang, and Yongdong Zhang. Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6458–6467, 2021.

[20] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5001–5010, 2020.

[21] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 772–781, 2021.

[22] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16317–16326, 2021.

[23] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[24] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[25] Iacopo Masi, Aditya Killekar, Royston Marian Mascarenhas, Shenoy Pratik Gurudatt, and Wael AbdAlmageed. Two-branch recurrent network for isolating deepfakes in videos. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 667–684. Springer, 2020.

[26] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[27] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[28] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

[29] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII*, pages 86–103. Springer, 2020.

[30] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[31] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1–11, 2019.

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

[33] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[34] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[35] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016.

[36] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *Acm Transactions on Graphics (TOG)*, 38(4): 1–12, 2019.

[37] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[38] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[39] Chengrui Wang and Weihong Deng. Representative forgery mining for fake face detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14923–14932, 2021.

[40] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, pages 5–32, 1992.

[41] Deressa Wodajo and Solomon Atnafu. Deepfake video detection using convolutional vision transformer. *arXiv preprint arXiv:2102.11126*, 2021.

[42] Zheng Zhang, Shuangfei Zhai, Lijun Yin, et al. Identity-based adversarial training of deep cnns for facial action unit recognition. In *BMVC*, page 226. Newcastle, 2018.