Improving Generalization of Deepfake Detectors by Imposing Gradient Regularization

Weinan Guan, *Student Member, IEEE*, Wei Wang, *Member, IEEE*, Jing Dong, *Senior Member, IEEE*, and Bo Peng, *Member, IEEE*

***Abstract*— The rapid development of face forgery technology has posed a significant threat to information security. While deepfake detection has proven to be an effective countermeasure, it often struggles to detect fake images generated by unknown forgery methods. Thus, the generalization ability of deepfake detectors to unseen forgery data is a critical concern. Despite many efforts aimed at discovering new forgery artifacts, they often fail to generalize to new manipulation technologies. In this paper, we tackle this challenge by focusing on the difference in texture patterns between training forgeries and unseen forgeries, which can lead to a degradation of generalization. Based on this principle, we propose a new conjecture that encourages deepfake detectors to reduce their sensitivity to forgery texture patterns, thereby improving the detection performance. To this end, we introduce an additional gradient regularization term to the original empirical loss during training. However, computing the Hessian matrix in the gradient calculation process of the regularization term poses a computational complexity. In order to overcome this issue, we optimize the formulation of the gradient regularization term using a first-order approximation method based on Taylor expansion and design a Perturbation Injection Module (PIM) to simplify the implementation process. Additionally, we provide a theoretical analysis from an optimization perspective and explore an interesting aspect of our method. Extensive experiments demonstrate the effectiveness of our approach in improving the generalization ability of deepfake detectors. Importantly, our method is orthogonal to recent advancements in powerful backbones and training data augmentation techniques. When combined with other effective techniques, our method achieves state-of-the-art experimental results.**

***Index Terms*— Deepfake detection, forgery texture patterns.**

Manuscript received 1 September 2023; revised 2 February 2024 and 22 April 2024; accepted 24 April 2024. Date of publication 2 May 2024; date of current version 13 May 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFC3320103 and in part by the National Natural Science Foundation of China (NSFC) under Grant 62372452 and Grant 62272460. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Tiziano Bianchi. *(Corresponding author: Wei Wang.)*

Weinan Guan is with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS), Center for Research on Intelligent Perception and Computing (CRIPAC), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China (e-mail: weinan.guan@cripac.ia.ac.cn).

Wei Wang, Jing Dong, and Bo Peng are with the State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS), Center for Research on Intelligent Perception and Computing (CRIPAC), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China (e-mail: wwang@nlpr.ia.ac.cn; jdong@nlpr.ia.ac.cn; bo.peng@nlpr.ia.ac.cn).

Digital Object Identifier 10.1109/TIFS.2024.3396064

I. Introduction

RECENTLY, the computer vision community has witnessed the remarkable progress of face forgery generation methods [1], [2], [3], [4], [5], [6] due to the success of deep learning, especially the rapid development of generative adversarial networks (GAN) [7]. This raises a security issue that an attacker can easily utilize these face forgery tools to achieve malicious purposes, e.g., spreading fake news, defaming celebrities, and falsifying evidence. To mitigate the abuse of the manipulated faces (called deepfakes), the community is keen on developing deepfake detection methods.

Most existing deepfake detection methods [8], [9], [10], [11], [12] exhibit excellent performance on in-domain datasets. This is because CNN models are strongly biased to textures [13]. For a trained deepfake detector, it can easily capture method-specific fake textures [14]. Therefore, if testing data is generated by the same forgery methods as training data (in-domain testing scenario), the learned textures help detectors distinguish forgery samples from pristine ones well. However, different forgery algorithms always have unique network architectures and processing streams, which results in different fake textures [15]. For the fake samples manipulated by unseen forgery methods, the performance of trained deepfake detectors always drops significantly, as some previous works [16], [17], [18], [19], [20] mentioned.

To address this limitation in generalization, many solutions have been proposed from the perspective of effective forgery artifacts detection [20], [21], [22], [23], [24], [25] or improving feature representations [9], [10], [26], [27]. To ensure that detectors can learn method-agnostic forgery features, some previous studies [21], [28], [29], [30], [31] focus on utilizing semantic information to explore some universal inconsistency or anomaly in forgery images. These works first extract semantic information from various methods by some pretrained models, and then explore their inconsistency or anomaly as deepfake detection clues to improve generalization performance. This is because that the semantic clue extraction is barely affected by the forgery textures. Some methods train the detectors with synthetic data [8], [11], [12], [32], [33], [34]. These methods utilize pristine data to synthesize training samples without any generative models, which causes the training process not impacted by any fake textures. The synthetic data is applied to imitate the blending process in the deepfake generation pipeline. The blending

artifact is a generic clue in deepfake samples. Besides, some works [9], [11], [26], [27], [35], [36] also employ the attention mechanism and frequency-aware methods to improve feature representations for better performance. However, the previous methods always require network architecture designs or data synthesizing strategies to improve the generalization ability of deepfake detection. The former easily suffer from extra computational overhead in the inference phase, while the latter may be invalid with the advancement of generative methods. Therefore, we would like to explore a method to further develop the capabilities of deepfake detectors without additional architecture modifications and training data.

In this paper, we improve the generalization performance of deepfake detectors from a novel perspective. Different from the previous methods, we focus on further exploiting the capabilities of detectors themselves. Our method can help the detectors further enhance their generalization ability, without introducing additional architecture modifications or training data. Specifically, we propose a novel loss function, which is comprised of the original empirical loss and a proposed gradient regularization item, to train deepfake detectors. We aim to further exploit the capabilities of detectors and achieve better generalization performance. To this end, we first analyze the reasons of poor generalization ability in the previous deepfake detectors. According to previous works [15], the poor detection performance on cross-domain forgeries is ascribed to the difference of image texture patterns due to different manipulation algorithms. Inspired by AdaIN [37] and MixStyle [38], we use feature statistics from shallow layers to represent image textures, and demonstrate that fluctuations in these statistics can impact the performance of deepfake detection. Building on this observation, we propose a new conjecture: enhancing detector robustness to perturbations in shallow feature statistics can improve detection performance, particularly with respect to generalization ability. To this end, we devise a gradient regularization item to impose on the original empirical loss. This improved loss function is utilized to further develop the capabilities of deepfake detectors for better generalization performance.

In the training phase, the gradient calculation of the proposed regularization item involves the complex Hessian matrix computation. Inspired by [39] and [40], we find an approximation solution of the regularization item by leveraging Taylor expansion, and optimize the total loss function. Besides, we propose a plug-and-play module to implement our method in the training phase. The module can be ignored in the inference process, such that no extra computation overhead is introduced.

We then comprehensively evaluate the effectiveness of our method on improving the generalization ability of deepfake detectors. When equipped with the powerful backbone (e.g. ConvNeXT-base) and advanced self-blended synthetic training data, the proposed detection model can generalize better on unseen manipulations compared to previous works.

Furthermore, we analyze our method from the perspective of optimization process. Different from the previous optimization methods, our method can be regarded as searching for a flat minimum along the variation of image texture patterns instead of network parameters. Based on this, we explore a straightforward improvement by combining our method with a previous optimization method [39], [40]. The experimental results show the superiority of our method in the deepfake detection task.

Our contributions can be summarized into four-folds:

- We propose to impose a novel regularization item on the original empirical loss for further developing the capabilities of deepfake detectors and achieving better generalization performance. It is an orthogonal improvement to existing techniques.
- We empirically demonstrate that our method can reduce the model sensitivity to forgery texture patterns. And we also devise a simple approximation solution for avoiding the computation of Hessian matrix.
- Equipped with other effective techniques for improving generalization, the detectors trained by our method achieve state-of-the-art cross-domain evaluation results. The extensive experiments validate the effectiveness of our proposed method.
- We further explore our method from the perspective of optimization process, and provide a straightforward improvement based on it. The experiments verify the superiority of our method in boosting generalization ability of deepfake detectors.

II. Related Work

A. Face Manipulation

Recent face manipulation methods can be roughly divided into three categories, i.e. face swapping, face reenactment and face attribute editing. Face swapping is replacing the face of one person in a video or image with the face of another person. The methods have gradually developed from specific face-swap objects [1], [41] to arbitrary objects [2], [3]. In the early stage, face swapping is achieved by a shared encoder and two individual-specific decoders. When swapping faces, a face is decoded by another decoder after encoding to the latent space. However, these face-swapping methods require to train a decoder for each person, which is not sufficiently flexible. FSGAN [2] first proposes a subject-agnostic face-swapping method by adjusting both pose and expression variations of source faces, which can simultaneously achieve face swapping and face reenactment. Different from FSGAN, FaceShifter [3] achieves face-swapping by thorough and adaptive integration of target attributions. After that, more identity preservation based methods [4], [5], [6] are emerged and achieve better performance. These methods first decouple the identities and attributes of source and target faces. After this, the source identity and target attributes are integrated to regenerated the swapped faces. Face reenactment, also named facial expression swap, is to modify the facial expression of the person. The most popular techniques are Face2Face [42] and NeuralTextures [43]. The former aims to animate the facial expressions of the target video based on 3D facial information, while the latter optimizes a neural texture in conjunction with a rendering network to compute the reenactment result [44]. Besides, talking face generation [45], [46], [47], [48], [49] is

also an interesting topic, which aims to synthesize a sequence of face images that correspond to a clip of speech. And face attribute editing [50], [51] can edit the attributes in a fine-grained manner, like changing the color of hair and wearing eyeglasses.

With the advances of face manipulation technology, deepfakes are more realistic than before. Some flaws in the previous deepfakes have been fixed, which makes some deepfake detection methods lose effectiveness.

B. Face Forgery Detection

Recent studies of deepfake detection achieve remarkable success. Although in-domain deepfake detection has been solved well, the generalization ability on the cross-domain scenarios is still a challenging problem. Some efforts [8], [11], [12], [32], [33], [34], [52] focus on designing more efficient network architectures for enhancing the learning ability of the networks. In these efforts, some researchers [9], [10], [53], [54], [55], [56] focus more on deeply mining the forgery artifacts by utilizing spatial and temporal attention mechanism. In addition, for improving the generalization, an effective solution is to train models with synthetic data, which encourages models to learn generic representations for deepfake detection, such as DSP-FWA [21], Face X-ray [22], PCL+I2G [24], ICT [25], SBI [20] and so on. Utilizing these synthetic data can remove the impacts of the artifacts of specific manipulation methods, which makes the learned features more generalizable. Specifically, these methods aim to learn some semantically related clues, which are caused by the blending process in the deepfake generation pipeline. DSP-FWA focuses on the resolution differences between the swapped face regions and other regions. Face X-ray aims to make detectors learn the blending boundaries, while SBI concentrates on the mismatch between the blended regions and the original regions, including resolution, color, boundary and landmarks. PCL+I2G utilize the synthetic data to explore the local source features in a suspected image and measure their self-consistency to detect forgeries. Different from them, ICT focuses on the identity inconsistency between the inner and outer face to detect face-swapping images. And then, there are still some methods to address this problem from other perspectives, such as frequency domain [27], [57], [58], meta-learning methods [59] and reinforcement learning [60]. Some researchers [61] also direct their attention towards ensuring the fairness of deepfake detectors across demographic variables, aiming for broad applicability in practical scenarios. Another interesting solution of deepfake detection is proactive defense, such as [62]. The authors [62] propose to embed a unique watermark at an assigned location of fake images to spot the deepfakes in general.

Most previous methods for improving generalization aim to find common forgery artifacts and learn generic representations among various deepfakes. Different from them, we are interested in searching for a set of network weights which is relatively stable to the variation of forgery texture patterns. Theoretically, it can be combined with any of the above methods for further improving their generalization.

TABLE I
THE GENERALIZATION PERFORMANCE OF CONVNEXT-BASE [63] DETECTOR ON UNSEEN FORGERIES

Training Set	Testing Set (AUC)			
	DF	FS	F2F	NT
DF	**1.0000**	0.4329	0.6672	0.6740
FS	0.6419	**0.9997**	0.6603	0.3737
F2F	0.8910	0.6251	**1.0000**	0.4861
NT	0.9015	0.3828	0.6883	**0.9784**

III. METHOD

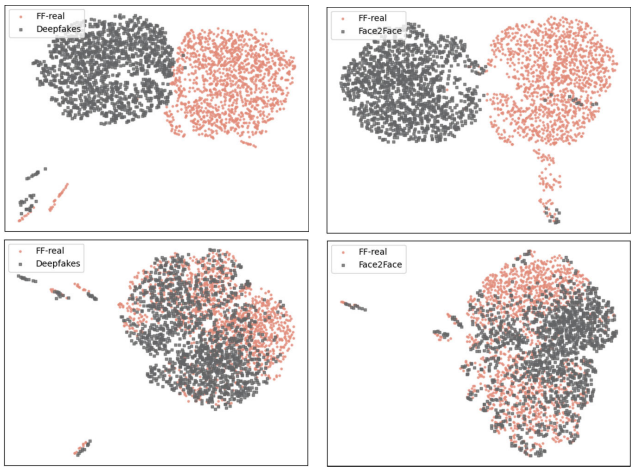
In this section, we begin by outlining the motivation behind our work and provide supporting evidence through a series of experiments (Sec III-A). Building upon these findings, we introduce a regularization term to enhance the original empirical loss function (Sec III-B). To streamline the implementation process, we also present a theoretical approximation of the loss function formulation (Sec III-C). Subsequently, we elaborate on the implementation details of our method across various neural networks and provide a comprehensive algorithm (Sec III-D). Finally, we validate the effectiveness of our proposed approach in mitigating the degradation of deepfake detection performance caused by variations in shallow feature statistics (Sec III-E).

A. Motivation

It is well known that deepfake detection easily suffers from the distribution mismatch problem, which means we can obtain a very promising performance when evaluating the model on the in-dataset scenario (seen forgeries in the training phase) but poor performance on the cross-dataset scenario (unseen forgeries) [20]. The reason is that the deep CNN models learn to capture the method-specific color textures for forgery detection [14], [15]. Once a CNN model has been biased to one kind of fake textures, it is hard to generalize to another one. As shown in Table I, we respectively train ConvNeXT-base [63] detectors on four forgeries (i.e. Deepfakes (DF), FaceSwap (FS), Face2Face (F2F) and NeuralTextures (NT)) of FaceForensics++ (FF++) dataset [44]. When evaluated on unseen forgeries, the performance of detectors drops significantly.

Inspired by AdaIN [37] and MixStyle [38], image textures can be represented by instance-level channel-wise means and variances of shallow features. Therefore, an intriguing question arises: are the first-order (mean) and second-order (variance) statistics of shallow features correlated with deepfake detection performance? To investigate this hypothesis, we employ ConvNeXT-base detectors trained on the DF and F2F datasets as our example. Figure 1 demonstrates Umap [64] visualizations of feature vectors from the last layers of the models. In the first row of Figure 1, it is evident that the detectors can effectively differentiate between forgery and genuine data.

To assess the impact of shallow feature statistics on deepfake detection performance, we introduce perturbations to the statistics (scaling the mean and variance to 1.2 and 0.65 times their original values to limit variation). In our experiments, we consider all three blocks of the first stage in the ConvNeXT-base model as the shallow layers, while the last features prior to the fully connected layers serve



(a) DF (b) F2F

Fig. 1. Feature space visualizations of the trained deepfake detectors on Deepfakes and Face2Face datasets of FF++. The classification feature distribution before (the first row) and after (the second row) adding perturbations on the shallow feature statistics are respectively shown here.

TABLE II
THE IMPAIRED DETECTION PERFORMANCE AFTER IMPOSING PERTURBATIONS ON THE SHALLOW FEATURE STATISTICS

Dataset	Perturbation	Testing Metric	
		Acc (↑)	Logloss (↓)
DF	before	0.9964	0.0193
	after	0.8107(−0.1857)	0.5190(+0.4997)
F2F	before	1.0000	0.0147
	after	0.9071(−0.0929)	0.2107(+0.1960)

as classification features. The second row of Figure 1 also showcases the distribution of classification features after perturbing the shallow feature statistics, revealing significant changes where the separability between real and forgery image classification features diminishes. Additionally, we present deepfake detection results in Table II, using Accuracy (Acc) and Logloss as evaluation metrics. The imposed perturbations notably degrade the detection performance, indicating that fluctuations in shallow feature statistics can indeed impact deepfake detection performance.

The aforementioned experiment inspires us to explore methods for improving the deepfake detection performance, particularly in terms of generalization ability, by enhancing the robustness of detectors to perturbations in shallow feature statistics. However, in theory, detectors should be robust to perturbations in any direction. Implementing non-unique fluctuation directions in detail can be challenging. To address this issue, we propose an alternative strategy to reduce model sensitivity to shallow feature statistics. Specifically, we suggest adding an additional regularization term to the empirical loss function. Our goal is to minimize the variation in model performance when there are changes in the first- and second-order statistics of shallow features. This regularization item aims to enhance the stability and robustness of the detector against variations in shallow feature statistics.

B. Loss Function and Regularization

For deepfake detection tasks, when given a training dataset $D_{train} = \{(x_i, y_i)\}_{i=0}^{N-1}$, a neural network model f_θ is trained to adapt the training data distribution, where $\theta = \{\theta_s, \theta_d\}$ includes the network parameters in shallow (θ_s) and deep (θ_d) layers. In the training process, f_θ is generally optimized by the objective function $\min_\theta L(x, y, \theta)$, where

$$L(x, y, \theta) = \frac{1}{N} \sum_{i=0}^{N-1} CE(f_{\theta_d}(f_{\theta_s}(x_i)), y_i), \tag{1}$$

is the empirical loss. Here, $CE(\cdot, \cdot)$ denotes the cross entropy loss function, and N is the number of training samples. It is noticed that in the subsequent derivations, we may utilize $L(f_{\theta_s}(x), y, \theta_d)$ to represent $L(x, y, \theta)$, particularly when it relates to the formula for $f_{\theta_s}(x)$.

As mentioned above, our objective is to reduce the model sensitivity to the variation of the shallow features $f_{\theta_s}(x)$. Since we have no knowledge about the unseen forgeries, the statistics of $f_{\theta_s}(x)$ are unpredictable. We utilize directional derivatives to represent this model sensitivity to $f_{\theta_s}(x)$. It can be formulated as follows:

$$||\frac{\partial L(f_{\theta_s}(x), y, \theta_d)}{\partial f_{\theta_s}(x)}||_2. \tag{2}$$

For simplification, we rewrite Eq (2) as

$$||\frac{\partial L(f_{\theta_s}(x), y, \theta_d)}{\partial l}||_2, \tag{3}$$

where l represents any variation direction of (μ_s, σ_s). μ_s and σ_s are respectively the mean and variance statistics of the shallow features ($f_{\theta_s}(x)$). By adding this regularization to the empirical loss function, its fluctuations with respect to the shallow feature will be suppressed, by which the model sensitivity is reduced.

The objective function can be further formulated as

$$\min_\theta \max_l L(x, y, \theta) + \lambda||\frac{\partial L(f_{\theta_s}(x), y, \theta_d)}{\partial l}||_2, \tag{4}$$

by only limiting the upper bound of Eq (3), which shows a min-max gaming. Here, λ is a weight parameter for balancing the regularization and empirical loss items.

C. Optimization

Based on Eq (4), we find two problems in the implementation. First, it is difficult to find the upper bound of the regularization item by directly optimizing the direction of l. Second, due to the computation involving the Hessian matrix, the gradient of the regularization item is complicated to solve.

To address these problems, we adopt an approach similar to [39] and [40]. Our derivations are also inspired by them. For the first question, when employing min-max game-based training for our models as depicted in Eq (4), the computation of derivatives for the regularization term during the optimization process incurs a substantial computation overhead. Therefore, we simplify the process by utilizing gradients to represent the upper bound of the regularization term, referring to the property of the directional derivatives. Eq (4) can be further streamlined as follows:

$$\min_\theta L(x, y, \theta) + \lambda||\nabla_{\mu_s, \sigma_s} L(f_{\theta_s}(x), y, \theta_d)||_2. \tag{5}$$

However, Eq (5) still involves the second problem. To tackle it, we aim to search for an approximate solution of the regularization term.

It is noticed that we define

$$f_{\theta_s}(x)_{norm} = \frac{f_{\theta_s}(x) - \mu_s}{\sigma_s}. \tag{6}$$

And $f'_{\theta_s}(x)$ is obtained by injecting perturbations to the shallow feature statistics of $f_{\theta_s}(x)$, which is defined as follows:

$$f'_{\theta_s}(x) = f_{\theta_s}(x)_{norm}(\sigma_s + \Delta\sigma_s) + (\mu_s + \Delta\mu_s). \tag{7}$$

where $\Delta\mu_s$ and $\Delta\sigma_s$ are the increment of μ_s and σ_s. Based on Taylor expansion, we have:

$$\begin{aligned} L(f'_{\theta_s}(x), y, \theta_d) = {} & L(f_{\theta_s}(x), y, \theta_d) \\ & + [\nabla_{\mu_s,\sigma_s} L(f_{\theta_s}(x), y, \theta_d)]^T \Delta l \\ & + o(||\Delta l||_2^2), \end{aligned} \tag{8}$$

where $\Delta l = (\Delta\mu_s, \Delta\sigma_s)$. In our problem, our primary focus is solely on the gradient of $L(f_{\theta_s}(x), y, \theta_d)$ as depicted in Eq (5). This inspires us to consider Taylor expansion only in the gradient direction. We set

$$\Delta l = r \frac{\nabla_{\mu_s,\sigma_s} L(f_{\theta_s}(x), y, \theta_d)}{||\nabla_{\mu_s,\sigma_s} L(f_{\theta_s}(x), y, \theta_d)||_2}, \tag{9}$$

where the value of r should be sufficiently small to ignore the high-order infinitesimal term ($o(||\Delta l||_2^2)$). According to the calculation formula of L_2 norm (i.e. $||x||_2^2 = x^T \cdot x$), the Eq (8) can be approximated to

$$\begin{aligned} L(f'_{\theta_s}(x), y, \theta_d) \approx {} & L(f_{\theta_s}(x), y, \theta_d) \\ & + r||\nabla_{\mu_s,\sigma_s} L(f_{\theta_s}(x), y, \theta_d)||_2. \end{aligned} \tag{10}$$

Then

$$\begin{aligned} & ||\nabla_{\mu_s,\sigma_s} L(f_{\theta_s}(x), y, \theta_d)||_2 \\ & = \frac{1}{r}[L(f'_{\theta_s}(x), y, \theta_d) - L(f_{\theta_s}(x), y, \theta_d)]. \end{aligned} \tag{11}$$

Hence, the objective function can be further reformulated as:

$$\min_{\theta}(1-\alpha)L(x, y, \theta) + \alpha L(f'_{\theta_s}(x), y, \theta_d), \tag{12}$$

where $\alpha = \lambda / r$.

The simplified objective function can be directly employed as our loss function in the training phase. Compared to the initial objective function (Eq (4)), we leverage the property of derivatives to streamline the min-max game process, avoiding the substantial computation overhead in the maximization process. And then we further provide an approximate solution for the regularization term, eliminating the need for complex Hessian matrix computations. These simplifications ensure that the detectors can be trained by our method without heavy computational costs.

D. Implementation on Neural Network

To implement our regularization term in deepfake detection models, we design a **P**erturbation **I**njection **M**odule (which we called **PIM**) as shown in Figure 2. Based on Eq (6), we first normalize the features of every channel in every shallow layers as $Normf_s$, and respectively calculate their mean (μ) and variance (σ) values. For the regularization item, we impose some perturbations ($\Delta\mu$ and $\Delta\sigma$) on the shallow feature statistics along their gradient directions. And then the features are denormalized by the new mean ($\mu' = \mu + \Delta\mu$) and variance ($\sigma' = \sigma + \Delta\sigma$) values. It is noticed that $\Delta\mu$ and $\Delta\sigma$ are shared among a batch in our implementation.

In the training phase, when a batch of data is given, we first set $\Delta\mu$ and $\Delta\sigma$ as zero vectors, and obtain $f'_{\theta_s}(x)$ based on Eq (7). It is noticed that $f'_{\theta_s}(x) = f_{\theta_s}(x)$ at this step. After the first back propagation, we can respectively calculate the gradients of the first item in the loss function (Eq (12)) with respect to the network parameters (θ) and shallow feature statistics ($\Delta\mu$ and $\Delta\sigma$). Following Eq (9), we only update $\Delta\mu$ and $\Delta\sigma$, and recalculate $f'_{\theta_s}(x)$. Based on this, the second item in Eq (12) is obtained to solve the gradient of θ. We accumulate both of the gradients and update the whole network parameters. The whole process of our method is shown in Algorithm 1. It is noticed that our method is essentially an improved loss function with an extra regularization item. The perturbation injection modules are not involved in the inference phase. Therefore, it has no extra computational overhead in the inference process.

Algorithm 1 Implementation of Our Method

Input: Training set $D_{train} = \{(x_i, y_i)\}_{i=0}^{N-1}$; network f with parameters θ (including θ_s and θ_d) which is initialized as θ^0; empirical loss function $L(x, y, \theta)$; learning rate η; total step T; weight parameter α; approximation scalar r.

Output: Optimized parameters $\hat{\theta}$

$\theta^0 = \theta$
for step $t = 0$ **to** T **do**
  Get batch data pairs $\{(x_i, y_i)\}_{i=0}^{B-1}$ sampled from D_{train}.
  Calculate the gradient $g_1 = \nabla_{\theta^t} L(f_{\theta_s^t}(x), y, \theta_d^t))$
  $\{\Delta\mu_s^t, \Delta\sigma_s^t\} = r \frac{\nabla_{\mu_s^t,\sigma_s^t} L(f_{\theta_s^t}(x), y, \theta_d^t))}{||\nabla_{\mu_s^t,\sigma_s^t} L(f_{\theta_s^t}(x), y, \theta_d^t))||_2}$
  Calculate $f'_{\theta_s^t}(x) = f_{\theta_s^t}(x)_{norm}(\sigma_s^t + \Delta\sigma_s^t) + (\mu_s^t + \Delta\mu_s^t)$ according to Eq (6) and Eq (7).
  $g_2 = \nabla_{\theta^t} L(f'_{\theta_s^t}(x), y, \theta_d^t))$
  $g = (1-\alpha)g_1 + \alpha g_2$.
  Based on (SGD) optimizer, update the parameters with the final gradient, $\theta^{t+1} = \theta^t - \eta g$
end for
return Finally optimized parameters $\hat{\theta}$

E. Robustness to Variation of Shallow Feature Statistics

In order to verify the effectiveness of our method in enhancing model robustness to variations in shallow feature statistics, we conduct experiments following the same settings as described in Section III-A. We train the ConvNeXT-base detectors using our method. As depicted in Figure 3, we observe a clear improvement in the separability between real and fake images when introducing perturbations to the shallow feature statistics. This observation is consistent with the results presented in Table III. With the assistance of our

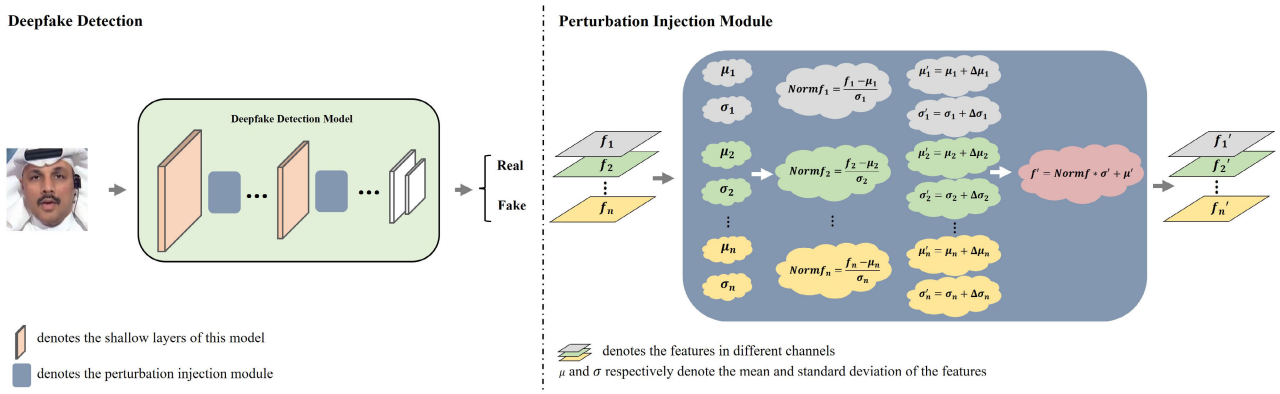


Fig. 2. The overview of our method. The left part shows the deepfake detection pipeline of our method. The right part details the proposed perturbation injection module.

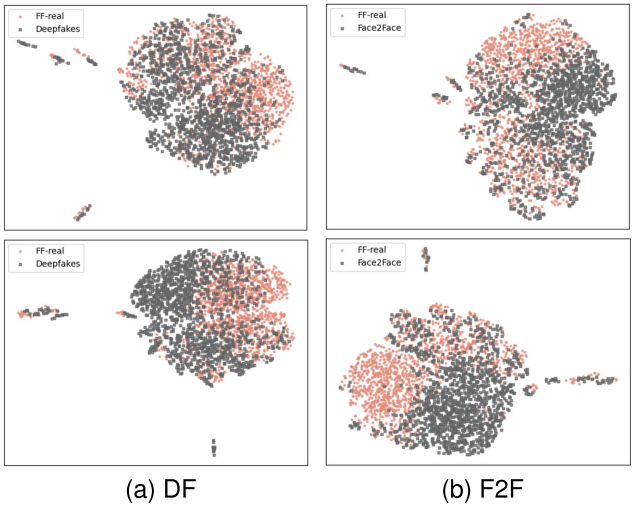


Fig. 3. Feature space visualizations of the trained deepfake detectors on Deepfakes and Face2Face datasets of FF++ when adding perturbations on the shallow feature statistics. The classification feature distribution trained without (the first row) and with (the second row) our method are respectively shown here.

method, the detection performance is significantly enhanced. Thus, we can conclude that our method effectively enhances the model's robustness to variations in shallow feature statistics, demonstrating its ability to address the aforementioned hypothesis.

IV. Explanation From the Optimization Perspective

In this section, we provide another analysis of our method from the perspective of optimization. In the optimization process, data (including images and labels) and the trained network are required to calculate the loss values. Previous works (e,g. SAM [39] and GNP [40]) have proven that the well-designed optimizers can help models improve generalization by finding flat minima [65]. However, as mentioned above, besides the model parameters, the optimizer loss function is also affected by the training data. Revisiting our method from this perspective, we find that our method can be regarded as searching for a flat minimum along the variation of image texture patterns. It means that we aim to find a set of network parameters such that when the image texture pattern changes, the fluctuation of the loss function is as flat as possible.

TABLE III
Detection Performance of Detectors Trained With/Without Our Method Under Perturbations on Shallow Feature Statistics

Dataset	Our Loss	Testing Metric	
		Acc (↑)	Logloss (↓)
DF	w/o	0.8107	0.5190
	w	0.9179(+0.1072)	0.2065(−0.3125)
F2F	w/o	0.9071	0.2107
	w	0.9536(+0.0465)	0.1453(−0.0654)

It is noticed that the flatness mentioned in the previous work is different from that in our method. The former focuses on the flatness along the network parameters, while our method concentrates on the flatness along the data distribution variation (i.e. image texture patterns in our task). A straightforward improvement is combing the solution procedure of SAM [39] or GNP [40] and our method to simultaneously search for a flat minimum along both directions.

As an interesting extension of our method (which we call DualFlat), we first reformulate the loss function (Eq (1)) as follows,

$$\begin{aligned} L'(x, y; \theta) = & L(x, y, \theta) \\ & + \lambda ||\nabla_{\mu_s, \sigma_s} L(f_{\theta_s}(x), y, \theta_d)|| \\ & + \gamma ||\nabla_{\theta} L(x, y, \theta)||, \end{aligned} \tag{13}$$

where L' is the new loss version, and λ and γ are weight parameters for balancing the regularization and empirical loss items. Following the above simplification procedure, we have

$$\begin{aligned} L'(x, y; \theta) = & (1 - \alpha - \beta) L(x, y, \theta) \\ & + \alpha L(f'_{\theta_s}(x), y, \theta_d) \\ & + \beta L(x, y, \theta + r' \frac{\nabla_{\theta} L(x, y, \theta)}{||\nabla_{\theta} L(x, y, \theta)||}). \end{aligned} \tag{14}$$

Here, r' is also the approximation scalar, which should be sufficiently small such that the high-order infinitesimal term in Taylor expansion can be ignored in the approximation. And $\beta = \frac{\gamma}{r'}$ which is also a balance coefficient.

V. Experiments

A. Experimental Settings

1) Datasets: We adopt the widely used benchmark **FaceForensics++** [44] (FF++) as our training dataset.

FF++ contains 1,000 original videos collected from YouTube and 4,000 fake videos forged by four forgery algorithms, i.e., Deepfakes [1] (DF), FaceSwap [66] (FS), Face2Face [42] (F2F), and NeuralTextures [43] (NT). DF and FS are utilized to swap faces between two different individuals, while F2F and NT are utilized to achieve face reenactment. For evaluating the generalization performance, we use four extra deepfake datasets, including **Celeb-DF-v2** dataset [67] (CDF), **Deepfake Detection Challenge Preview** dataset [68] (DFDCP), **Deepfake Detection Challenge** public test dataset [69] (DFDC) and **WildDeepfake** dataset [70] (WDF). CDF is comprised of 590 real videos and 5,639 deepfake videos, in which the former are chosen from publicly available YouTube videos, corresponding to interviews of 59 celebrities. The fake parts are generated by an improved deepfake algorithm. DFDCP and DFDC are created for Deepfake Detection Challenge competition, in which all videos are created by entering into agreements from paid actors. DFDCP contains around 5,000 videos, while DFDC has 4,000 videos. WDF is collected from varies video-sharing websites, containing 7,314 face sequences (Real $\rightarrow$ 3,805 and Fake $\rightarrow$ 3,509) from 707 videos. In our experiments, we only utilize the test sets of CDF, DFDCP and WDF to evaluate our method, which are officially provided.

2) Evaluation Metrics: In our experiments, we report video-level AUC (area under the receiver operating characteristic curve) as the evaluation metric. We select 32 frames from each video at equal intervals, and these frame-level scores are averaged as the video-level prediction result. The experimental results of other deepfake detection methods which we use for comparison are directly cited.

3) Implementation Details: Due to the similar objective function with [39] and [40], we respectively set the hyper-parameters r in Eq (10) and α in Eq (12) as $r = 0.1$ and $\alpha = 1$ in our experiments. All models we train are initialized with the pretrained weights on ImageNet [71] and trained for a maximum of 100 epochs. In the training phase, we only sample 8 frames per video and resize the crop faces with specific size based on different trained models (e.g. the face is resized to 224×224 for ConvNeXt-base [63] and 380×380 for EfficientNet [72]). We levearage the recent ConvNeXT-base as our backbone model, which shows excellent performance on image classification tasks. The batch size is 64, and we do not use any data augmentation in our experiments.

B. Effects on Various Backbones

To verify our assumption, we adopt several commonly used deepfake detection backbones to explore the effects of our methods on them. As shown in Table IV, we respectively utilize ConvNeXT [63] (ConvNeXT-base), XceptionNet [73], EfficientNet [72] (efficientnet-b4) and ResNet (ResNet-50) as our backbones. We respectively plug PIMs into their shallow layers to train with our loss function, and compare their generalization performance on four cross-domain datasets with the corresponding models trained by the empirical loss. To provide a comprehensive comparison, we also evaluate their generalization performances when employing commonly used data augmentations in the training phase, including HorizontalFlip, RGBShift, HueSaturationValue, RandomBrightnessContrast, ImageCompression, and GaussNoise.

TABLE IV
THE EFFECTIVENESS OF OUR METHOD ON THE GENERALIZATION PERFORMANCE OF VARIOUS BACKBONES. THE BEST RESULTS ARE IN BOLD

Data Augmentation	Backbones	Testing Set (AUC %)				
		CDF	DFDCP	DFDC	WDF	Average
Without	ResNet-50	75.00	69.18	61.06	**73.30**	69.64
	+PIM	**76.37**	**70.72**	**62.07**	71.40	**70.14**
	XceptionNet	**74.63**	68.98	59.08	67.11	67.45
	+PIM	74.22	**71.39**	**60.14**	**70.58**	**69.08**
	EfficientNet-b4	75.67	80.94	**65.15**	74.15	73.98
	+PIM	**77.53**	**81.15**	65.09	**76.16**	**74.98**
	ConvNeXT-base	86.70	77.15	67.57	**72.75**	76.04
	+PIM	**87.34**	**80.69**	**68.04**	72.28	**77.09**
With	ResNet-50	78.92	69.24	62.43	73.60	71.05
	+PIM	**81.42**	**72.21**	**63.70**	**75.56**	**73.22**
	XceptionNet	82.99	77.59	64.23	**75.03**	74.96
	+PIM	**85.92**	**78.25**	**64.84**	**75.03**	**76.01**
	EfficientNet-b4	83.06	81.68	67.13	74.21	76.52
	+PIM	**84.93**	**82.31**	**67.22**	**74.93**	**77.35**
	ConvNeXT-base	84.41	79.14	70.45	71.57	76.39
	+PIM	**86.87**	**80.32**	**71.73**	**74.33**	**78.31**

From Table IV, it is obvious that the regularization item in our proposed loss can help these commonly used deepfake detection models improve the generalization ability, which also demonstrates that our method is a model-agnostic generalization ability improvement strategy. For the training phase without data augementaions, the early classification model (ResNet) who is trained by our loss function, ResNet achieves better AUC on CDF (76.37 VS 75.00), DFDCP (70.72 VS 69.18) and DFDC (62.07 VS 61.06) datasets. For the commonly recommended deepfake detectors (XceptionNet and EfficientNet), XceptionNet+PIMs and EfficientNet+PIMs respectively achieve better average generalization performance on the four cross-domain datasets. Especially, the former respectively achieves 2.41% and 3.47% better AUC on DFDCP and WDF, and the latter respectively achieves 1.86% and 2.01% better AUC on CDF and WDF. And for the recent powerful ConvNeXT, the usage of PIMs boosts higher performance (+1.05% average AUC) by limiting the model sensitivity to image texture patterns. When the data augmentations are applied to the training data, we observe significant improvements in the generalization ability of various detectors. Our method further enhances their generalization performances on various unseen forgery datasets, which is consistent with our previous experimental results. The generalization performances of various detectors are universally enhanced with the assistance of our method. It is noticed that we introduce no additional architecture modifications or training data. This demonstrates that our method indeed helps the detectors further exploit their capabilities. This also provides a potential application for our method to further improve the performance, that is, replacing the empirical loss with our proposed loss and applying PIM as a plug-and-play module to other SOTA methods.

C. Comparison With State-of-the-Art Methods

In this part, we focus on investigating how to fully take advantage of our method for better generalization. In our method, we improve the deepfake detection generalization ability by alleviating the model sensitivity to image color

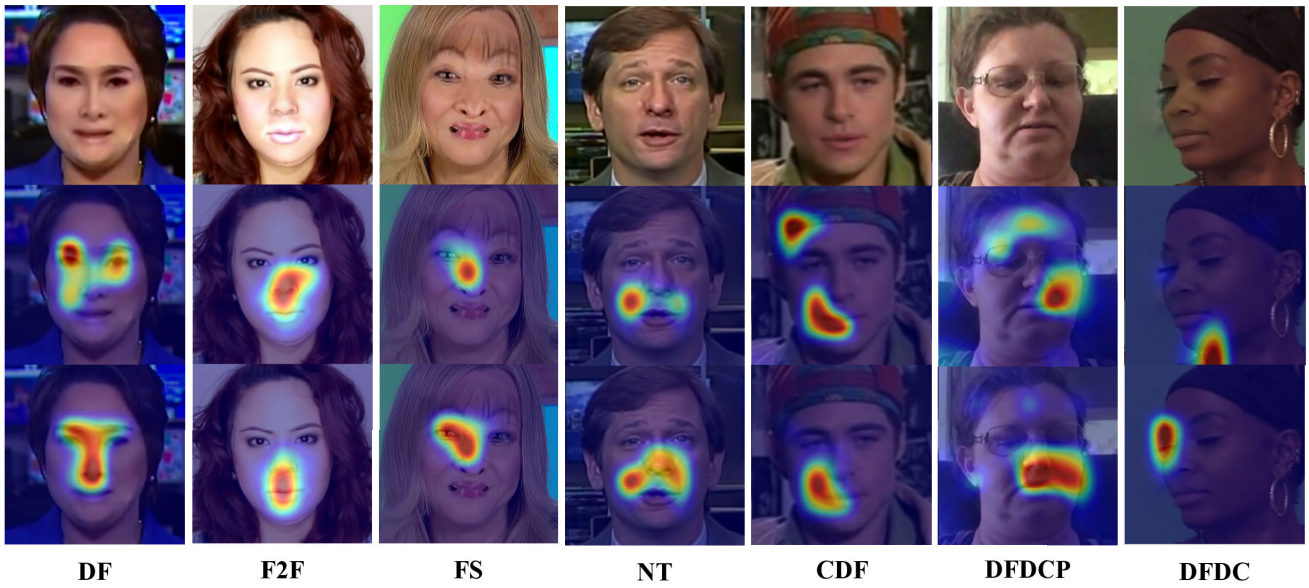


Fig. 4. Saliency map visualization of convnext-base detectors (trained with/without our method) on different datasets. The first row is the fake faces in different datasets. The middle and last rows are separately the saliency maps of the detectors trained without and with our method.

textures. This is because that, inspired by the previous work [15], CNN-based detectors are biased to capture the specific forgery texture patterns. This leads to poor generalization performance when they encounter unseen forgery methods. In other words, our method weakens the effect of image texture patterns on deepfake detection. However, the generalization of deepfake detection is affected by multiple factors. An interesting question is whether we can further improve the generalization performance of deepfake detection by combining our method with other recent improvements.

In our implementation, we integrate the designed PIMs into shallow layers of the detection backbone for improving its generalization ability with our proposed loss function. From the perspective of network architecture, we demonstrate the effectiveness of our method by combining with multiple detection backbones. To comprehensively evaluate our method, we leverage SBI [20] data to train the networks equipped with our PIMs and loss function to further improve the generalization performance from the perspective of combining with this SOTA data augmentation method.

For demonstrating the superiority of our method, we also show the performance of other recent methods, as shown in Table V. The state-of-the-art detectors we utilize for comparison includes Face X-ray [22], LRL [74], FRDM [15], PCL+I2G [24], RFM [10], MultiAtt [11], LipForensics [29], RECCE [12], FTCN [75], SLADD [76], Wavelet [77], DisGRL [78], STC [79], MRL [55] and SBI [20]. The training data of their experiments are all based on FF++ dataset. It is noticed that we retrain a ConvNeXT-base model on SBI data, following the official code, denoted as SBI* in Table V. SBI* has similar performance to the official reports. Due to the same training environment and settings as ours, we use it as the baseline of our results for fair comparison. It is shown that equipped with our method, the ConvNeXT-base model trained on SBI data beats its competitors (SBI*) on CDF and DFDC by large margins (1.25% and 2.59%, respectively), although they have identical architectures and training data. This demonstrates the effectiveness of our method in enhancing the capability of the state-of-the-art detector. Unlike previous approaches focused on discovering new forgery artifacts, our method improves generalization by enhancing the inherent capabilities of deepfake detectors themselves. Hence, it introduces no additional computational overhead or requires new training data. Compared to other state-of-the-art methods, our method provides additional improvements in generalization performances.

D. Analysis of Saliency Map Visualization

In this section, we employ Grad-CAM [80] to visualize the regions that deepfake detectors focus on when analyzing deepfake faces. Figure 4 illustrates the results. For in-domain deepfake faces (DF, F2F, FS, and NT), our method-trained detector tends to make decisions based on more forged regions. This suggests that our method encourages the detector to identify more forgery clues, such as blending artifacts, as observed in the DF example. On the other hand, for cross-domain deepfake faces (CDF, DFDCP, and DFDC), our method helps the detector reduce its attention to irrelevant regions when confronted with unseen forgeries. Furthermore, the detector, with the assistance of our method, focuses more on forgery artifacts, even if they are subtle (as demonstrated in the DFDC example). This observation may explain why our method improves the detection performance, particularly in terms of generalization ability.

E. Ablation Study

In this section, we focus on analyzing the effects of the approximation scalar r in Eq (9) and balance coefficient α in Eq (12) on model generalization performance.

1) Approximation Scalar: The approximation scalar r is utilized to ignore the high-order infinitesimal term $o(||\Delta l||_2^2)$ in Eq (8). Based on this, we can obtain an approximation of the second item in Eq (5) for evading the complex Hessian

TABLE V

COMPARISON OF GENERALIZATION ABILITY WITH STATE-OF-THE-ART METHODS USING AUC. THE BEST RESULTS ARE IN BOLD. AND THE SECOND-BEST VALUES ARE UNDERLINED. THE RESULTS OF OUR METHOD ARE THE CONVNEXT-BASE MODEL TRAINED ON SBI DATA, WHICH INTRODUCES **PIM** INTO THE ORIGINAL ARCHITECTURE AND IS TRAINED WITH OUR LOSS FUNCTION. * DENOTES THE RESULTS ARE EVALUATED BY OURSELVES FOLLOWING THE OFFICIAL CODES. SBI* DENOTES THAT WE RETRAIN THE CONVNEXT-BASE MODEL ON SBI DATA, WITH THE OFFICIAL CODE. † DENOTES THE RESULTS ARE CITED FROM [12], [77], [78]. THE OTHER RESULTS ARE DIRECTLY CITED FROM THE OFFICIAL PAPERS OR [20]

Methods	Pub./Year	Training Data	Testing Data (AUC %)				
			FF++	CDF	DFDCP	DFDC	WDF
Face X-ray [22]	CVPR'20	FF++-Real & BI	98.52	–	71.15	–	–
Face X-ray [22]	CVPR'20	FF++ & BI	–	–	80.92	–	–
FTCN [75]	ICCV'21	FF++	–	86.90	74.00	71.00	65.60*
PCL+I2G [24]	ICCV'21	FF++-Real & I2G	99.11	90.03	74.37	67.52	–
LRL [74]	AAAI'21	FF++	99.46	78.26	76.53	–	68.76
FRDM [15]	CVPR'21	FF++	98.36†	75.31†	71.76†	58.47*	73.00†
RFM [10]	CVPR'21	FF++	98.79†	65.63†	74.66*	61.57*	57.75†
MultiAtt [11]	CVPR'21	FF++	99.27†	76.65†	67.34†	68.01†	70.15†
RECCE [12]	CVPR'22	FF++	99.32	68.71	–	69.06†	64.31
Wavelet [77]	MM'22	FF++	**99.55**	84.75	78.51	–	73.80
DisGRL [78]	IJCAI'23	FF++	<u>99.48</u>	70.03	–	70.89	66.73
STC [79]	MM'23	FF++-Real & STG	99.41	83.37	**86.82**	<u>73.39</u>	–
MRL [55]	TIFS'23	FF++	98.27	83.58	71.53	–	–
SBI* [20]	CVPR'22	FF++-Real & SBI	99.39	<u>93.89</u>	85.48	72.06	<u>74.00</u>
Ours	-	FF++	99.17	87.34	80.69	68.04	72.28
Ours	-	FF++-Real & SBI	99.17	**95.14**	<u>85.75</u>	**74.65**	**74.40**

matrix solution in the optimization process. The approximation scalar r should be set carefully, since it would directly affect the approximation precision. The scalar should be sufficiently small for ignoring the high-order infinitesimal term. However, if it is excessively small, the shallow feature statistics variation (Δl) will be too weak, which makes Eq (10) equal to the empirical loss $L(f_{\theta_s}(x), y, \theta_d)$.

To investigate its effect on the model generalization performance, we fix the balance coefficient ($\alpha = 1$) and choose different approximation scalars ($r = \{0, 0.05, 0.1, 0.2\}$) in our proposed method to train the deepfake detectors. According to Eq (12), when r equals 0, the loss function is actually the empirical loss. As depicted in Table VI, when our method is introduced (i.e. r>0), the generalization performance exhibits improvements, demonstrating the effectiveness of our method. However, as previously mentioned, the scalar r should be sufficiently small to disregard the high-order infinitesimal term. It is noticed that when r equals 0.2, the improvement of generalization performance is significantly impaired, aligning with our previous analysis. This demonstrates the importance of selecting an appropriate scalar value. If the approximation scalar is not sufficiently small, the high-order infinitesimal term in Taylor expansion may not be ignored. This directly affects the approximation precision of Eq (10), resulting in the limited improvement of the generalization performance. Conversely, when r equals 0.05 and 0.1, the model achieves similarly good generalization performance, significantly outperforming the baseline. This highlights that the appropriate scalar values enable our method to more effectively enhance the generalization ability of deepfake detectors.

2) Balance Coefficient: The balance coefficient α is related to the balance between the empirical loss item and the regularization item in the loss function. As shown in Eq (12), after simplifying the latter formulation to avoid complex calculations on Hessian matrix, α is transformed into a balance

TABLE VI

THE IMPACT OF THE APPROXIMATION SCALAR (r) ON GENERALIZATION ABILITY IN THE PROPOSED METHOD. THE METRIC IS AUC. AVG-AUC DENOTES THE AVERAGE AUC VALUES ON THE FOUR CROSS-DOMAIN DATASETS

Ours		Testing Data (AUC %)				Avg-AUC %
α	r	CDF	DFDCP	DFDC	WDF	
1	0	86.70	77.15	67.57	72.75	76.04
1	0.05	86.92	80.61	68.34	72.56	77.10
1	0.1	87.34	80.69	68.04	72.28	77.09
1	0.2	87.99	77.51	67.95	71.76	76.30

between the original empirical loss and the empirical loss with the shallow feature statistics perturbation. It is noticed that we only explore its impact when α does not exceed the range of [0, 1]. This is because that when $\alpha < 0$, λ in Eq (5) is negative, which encourages the model more sensitivity to the shallow feature statistics. This breaks the assumption of our method. And if $\alpha > 1$, this may encourage the model to increase the empirical loss to achieve a lower total loss value. This may cause the model to focus more on the model sensitivity while ignoring the deepfake detection performance. Therefore, we limit the variation range of α in the following experiments.

To investigate the impact of balance coefficient α on model generalization, we conduct ablation studies on our optimization strategy. As shown in Table VII, we fix the approximation scalar ($r = 0.1$) and vary the balance coefficients ($\alpha = \{1, 0.75, 0.50, 0.25, 0\}$) to optimize our models. It is evident that as α increases, the generalization performance shows an increasing trend. When $\alpha = 1$, the trained model shows the best generalization performance (Average AUC) on the cross-domain datasets. Conversely, when α is too small (especially for $\alpha = 0.50, 0.25$), there is a noticeable degradation in average generalization performance. Through the comparison in Table VII, it can be concluded that the value of α does affect generalization performance to some extent because it is related to the punishment of the model sensitivity to image texture

TABLE VII

THE IMPACT OF THE BALANCE COEFFICIENT (α) ON GENERALIZATION ABILITY IN THE PROPOSED METHOD. THE METRIC IS AUC. AVG-AUC DENOTES THE AVERAGE AUC VALUES ON THE FOUR CROSS-DOMAIN DATASETS

Ours		Testing Data (AUC %)				Avg-AUC %
α	r	CDF	DFDCP	DFDC	WDF	
1	0.1	87.34	80.69	68.04	72.28	77.09
0.75	0.1	87.01	78.54	68.85	72.88	76.82
0.50	0.1	87.40	77.11	68.04	72.07	76.16
0.25	0.1	86.57	77.47	66.75	72.65	75.86
0	0.1	86.70	77.15	67.57	72.75	76.04

TABLE VIII

THE IMPACT OF THE BALANCE COEFFICIENTS (α AND β) ON GENERALIZATION ABILITY IN THE PROPOSED DUALFLAT OPTIMIZATION STRATEGY. THE METRIC IS AUC. THE BEST RESULTS ARE IN BOLD. AVG-AUC DENOTES THE AVERAGE AUC VALUES ON THE FOUR CROSS-DOMAIN DATASETS

DualFlat		Testing Data (AUC %)				Avg-AUC %
$\alpha+\beta$	$\alpha:\beta$	CDF	DFDCP	DFDC	WDF	
1.0	1.0 : 0.0	**87.34**	**80.69**	68.04	72.28	**77.09**
	0.0 : 1.0	83.42	77.71	66.15	69.30	74.15
1.0	0.8 : 0.2	84.24	77.55	69.42	74.66	76.47
	0.5 : 0.5	83.80	77.16	**69.64**	74.52	76.28
	0.2 : 0.8	84.56	78.59	68.60	71.52	75.82
0.8	0.8 : 0.2	86.14	78.59	68.87	73.89	76.87
	0.5 : 0.5	84.33	76.99	68.93	73.78	76.01
	0.2 : 0.8	84.77	78.25	68.56	73.40	76.25
0.5	0.8 : 0.2	86.67	77.46	68.74	75.12	77.00
	0.5 : 0.5	85.47	77.95	68.68	73.68	76.45
	0.2 : 0.8	85.48	77.09	68.63	74.99	76.55
0.2	0.8 : 0.2	86.66	76.52	68.72	73.56	76.37
	0.5 : 0.5	86.64	76.74	69.18	**75.75**	77.08
	0.2 : 0.8	86.87	76.88	68.68	74.83	76.82

patterns. This observation further validates the rationality of our method.

F. Comparison With Flat Minima Optimization

To explore the effectiveness of the flat minima along both directions, we conduct some experiments on different combinations of two optimization directions, i.e. different combinations of α and β. We fix $r = 0.1$ and $r' = 0.05$ according to our approximation scalar ablation study and SAM [39]. The trained model is ConvNeXT-base and the rest of the experiment settings are the same as before. As shown in Table VIII, we explore different combinations of balance coefficients (i.e. $\alpha:\beta = \{0.8:0.2, 0.5:0.5, 0.2:0.8\}$) based on various values of $\alpha+\beta$ ($\{1, 0.8, 0.5, 0.2\}$). These combinations allow us to consider both data distribution flatness (our method) and network parameter flatness (SAM) simultaneously during the optimization process. In addition, according to Eq (14), it is evident that SAM and our method represent two special implementations of DualFlat, when we respectively set $\{\alpha = 0, \beta = 1\}$ and $\{\alpha = 1, \beta = 0\}$. We also provide their results for comprehensive comparisons.

From Table VIII, it can be concluded that DualFlat has the best generalization performance when it is degraded to our method (i.e. $\{\alpha = 1, \beta = 0\}$). It is undeniable that simultaneously searching the flat minimum in both data distribution and network parameter directions has remarkable improvement in generalization performance compared to SAM. However, when we only search flat minimum along the data distribution variation, the trained model can obtain the best generalization ability. It can be seen that DualFlat may be not a better optimization method in our tasks. The flatness along the network parameters may even drag down our generalization performance. Therefore, it demonstrates that our optimization method has the superiority of improving the generalization in deepfake detection.

TABLE IX

THE IMPACT OF THE BALANCE COEFFICIENTS (α AND β) ON GENERALIZATION ABILITY IN THE PROPOSED DUALFLAT OPTIMIZATION STRATEGY. THE TRAINING DATA IS FF++-REAL AND SBI. THE METRIC IS AUC. THE BEST RESULTS ARE IN BOLD. AVG-AUC DENOTES THE AVERAGE AUC VALUES ON THE FOUR CROSS-DOMAIN DATASETS

SBI-DualFlat		Testing Data (AUC %)				Avg-AUC %
$\alpha+\beta$	$\alpha:\beta$	CDF	DFDCP	DFDC	WDF	
1.0	1.0 : 0.0	95.14	**85.75**	**74.65**	**74.40**	**82.49**
	0.0 : 1.0	93.89	85.48	72.06	74.00	81.36
1.0	0.8 : 0.2	94.85	85.13	73.47	74.03	81.87
	0.5 : 0.5	94.78	85.57	72.53	73.44	81.58
	0.2 : 0.8	94.01	85.49	72.21	73.63	81.34
0.8	0.8 : 0.2	94.93	85.71	73.74	73.62	82.00
	0.5 : 0.5	94.98	85.59	72.81	73.49	81.72
	0.2 : 0.8	94.69	85.51	72.22	73.15	81.39
0.5	0.8 : 0.2	95.10	85.06	73.64	73.18	81.75
	0.5 : 0.5	94.98	85.02	72.95	73.25	81.55
	0.2 : 0.8	94.81	85.33	72.61	73.13	81.47
0.2	0.8 : 0.2	**95.69**	85.39	73.35	73.28	81.93
	0.5 : 0.5	95.05	84.94	73.08	73.05	81.53
	0.2 : 0.8	95.08	85.10	72.93	72.76	81.47

In addition, to more comprehensively investigate the generalization ability of DualFlat optimizer, we further demonstrate the generalization results of combining DualFlat with state-of-the-art method. Specifically, we repeat the above experiments with SBI settings. In the experiments, the combination strategies of α and β in DualFlat are the same as above. And the training data are substituted by SBI data and the real part of FF++ dataset.

As shown in Table IX, DualFlat also has the best generalization ability when it is degraded into our method (i.e., $\{\alpha = 1, \beta = 0\}$). It is clear that with the balance coefficient of our method (α) increasing, the generalization performance of DualFlat gradually improves. This indicates that in DualFlat mechanism, our method plays a vital role to improve the generalization ability of deepfake detection models, which is consistent with the previous experimental results. Therefore, compared to SAM and DualFlat, our method shows the best generalization ability in deepfake detection tasks.

VI. CONCLUSION

In this work, we investigate the generalization ability of face forgery detection from the perspective of the forgery texture pattern variation. Based on this, we design a novel regularization item to improve the original empirical loss. The improved loss function aims to reduce the model sensitivity to forgery texture patterns. To simplify the implementation process, we theoretically approximate the formulation of the regularization item. In addition, we also investigate an analysis from the perspective of optimization process, and provide an interesting generalization of our method. Extensive experiments demonstrate that our method consistently helps deepfake detectors achieve better generalization performance. Besides, we also verify that our method can generalize better to unseen manipulations when equipped with the powerful backbone and self-blended synthetic training data.

REFERENCES

[1] *Deepfakes*. Accessed: Feb. 17, 2023. [Online]. Available: https://github.com/deepfakes/faceswap

[2] Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject agnostic face swapping and reenactment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7184–7193.

[3] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, "Advancing high fidelity identity swapping for forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5074–5083.

[4] G. Gao, H. Huang, C. Fu, Z. Li, and R. He, "Information bottleneck disentanglement for identity swapping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3404–3413.

[5] R. Chen, X. Chen, B. Ni, and Y. Ge, "SimSwap: An efficient framework for high fidelity face swapping," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 2003–2011.

[6] Y. Zhu, Q. Li, J. Wang, C.-Z. Xu, and Z. Sun, "One shot face swapping on megapixels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 4834–4844.

[7] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds. New York, NY, USA: Curran Associates, 2014, pp. 1–9.

[8] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2018, pp. 1–7.

[9] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, "On the detection of digital face manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5781–5790.

[10] C. Wang and W. Deng, "Representative forgery mining for fake face detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14923–14932.

[11] H. Zhao, T. Wei, W. Zhou, W. Zhang, D. Chen, and N. Yu, "Multi-attentional deepfake detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2185–2194.

[12] J. Cao, C. Ma, T. Yao, S. Chen, S. Ding, and X. Yang, "End-to-end reconstruction-classification learning for face forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4113–4122.

[13] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; Increasing shape bias improves accuracy and robustness," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–22.

[14] Z. Liu, X. Qi, and P. H. S. Torr, "Global texture enhancement for fake face detection in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8057–8066.

[15] 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.*, Jun. 2021, pp. 16317–16326.

[16] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva, "ForensicTransfer: Weakly-supervised domain adaptation for forgery detection," 2018, *arXiv:1812.02510*.

[17] M. Du, S. Pentyala, Y. Li, and X. Hu, "Towards generalizable deepfake detection with locality-aware AutoEncoder," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manag.*, Oct. 2020, pp. 325–334.

[18] A. Khodabakhsh, R. Ramachandra, K. Raja, P. Wasnik, and C. Busch, "Fake face detection methods: Can they be generalized?" in *Proc. Int. Conf. Biometrics Special Interest Group (BIOSIG)*, Sep. 2018, pp. 1–6.

[19] X. Xuan, B. Peng, W. Wang, and J. Dong, "On the generalization of GAN image forensics," in *Biometric Recognition*, Zhuzhou, China. Cham, Switzerland: Springer, 2019, pp. 134–141.

[20] K. Shiohara and T. Yamasaki, "Detecting deepfakes with self-blended images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18699–18708.

[21] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 8261–8265.

[22] L. Li et al., "Face X-ray for more general face forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5001–5010.

[23] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1–7.

[24] T. Zhao, X. Xu, M. Xu, H. Ding, Y. Xiong, and W. Xia, "Learning self-consistency for deepfake detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 15023–15033.

[25] X. Dong et al., "Protecting celebrities from DeepFake with identity consistency transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9458–9468.

[26] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, "Thinking in frequency: Face forgery detection by mining frequency-aware clues," in *Computer Vision—ECCV*, Glasgow, U.K. Cham, Switzerland: Springer, 2020, pp. 86–103.

[27] H. Liu et al., "Spatial-phase shallow learning: Rethinking face forgery detection in frequency domain," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 772–781.

[28] S. Agarwal, H. Farid, T. El-Gaaly, and S. Lim, "Detecting deep-fake videos from appearance and behavior," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2020, pp. 1–6.

[29] A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic, "Lips don't lie: A generalisable and robust approach to face forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5039–5049.

[30] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha, "Emotions don't lie: An audio-visual deepfake detection method using affective cues," in *Proc. 28th ACM Int. Conf. Multimedia*, New York, NY, USA, Oct. 2020, pp. 2823–2832.

[31] X. Zhu, H. Wang, H. Fei, Z. Lei, and S. Z. Li, "Face forgery detection by 3D decomposition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2928–2938.

[32] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," in *Proc. IEEE 10th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Sep. 2019, pp. 1–8.

[33] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Use of a capsule network to detect fake images and videos," 2019, *arXiv:1910.12467*.

[34] Z. Gu et al., "Spatiotemporal inconsistency learning for deepfake video detection," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 3473–3481.

[35] W. Lu et al., "Detection of deepfake videos using long-distance attention," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 6, 2023, doi: 10.1109/TNNLS.2022.3233063.

[36] S. Woo et al., "ADD: Frequency attention and multi-view based knowledge distillation to detect low-quality compressed deepfake images," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 1, pp. 122–130.

[37] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1501–1510.

[38] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Adversarial domain generalization with MixStyle," in *Proc. ICLR*, 2021, pp. 379–385.

[39] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–20.

[40] Y. Zhao, H. Zhang, and X. Hu, "Penalizing gradient norm for efficiently improving generalization in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 26982–26992.

[41] I. Korshunova, W. Shi, J. Dambre, and L. Theis, "Fast face-swap using convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 3677–3685.

[42] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-time face capture and reenactment of RGB videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2387–2395.

[43] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, Aug. 2019.

[44] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1–11.

[45] H. Zhou, Y. Sun, W. Wu, C. C. Loy, X. Wang, and Z. Liu, "Pose-controllable talking face generation by implicitly modularized audio-visual representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4174–4184.

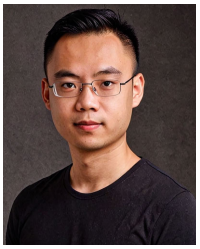
[46] K. Cheng et al., "VideoReTalking: Audio-based lip synchronization for talking head video editing in the wild," in *Proc. SIGGRAPH Asia Conf. Papers*, Nov. 2022, pp. 1–9.

[47] Y. Ma et al., "StyleTalk: One-shot talking head generation with controllable speaking styles," 2023, *arXiv:2301.01081*.

[48] S. Yang, W. Wang, J. Ling, B. Peng, X. Tan, and J. Dong, "Context-aware talking-head video editing," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 7718–7727.

[49] S. Yang et al., "Learning dense correspondence for nerf-based face reenactment," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 6522–6530.
[50] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8789–8797.
[51] M. Liu et al., "STGAN: A unified selective transfer network for arbitrary image attribute editing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3673–3682.
[52] Q. Yin, W. Lu, B. Li, and J. Huang, "Dynamic difference learning with spatio-temporal correlation for DeepFake video detection," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 4046–4058, 2023.
[53] C. Zhao, C. Wang, G. Hu, H. Chen, C. Liu, and J. Tang, "ISTVT: Interpretable spatial–temporal video transformer for deepfake detection," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1335–1348, 2023.
[54] P. Yu, J. Fei, Z. Xia, Z. Zhou, and J. Weng, "Improving generalization by commonality learning in face forgery detection," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 547–558, 2022.
[55] Z. Yang, J. Liang, Y. Xu, X.-Y. Zhang, and R. He, "Masked relation learning for deepfake detection," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1696–1708, 2023.
[56] J. Yang, A. Li, S. Xiao, W. Lu, and X. Gao, "MTD-Net: Learning to detect deepfakes images by multi-scale texture difference," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4234–4245, 2021.
[57] 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.*, 2020, pp. 3247–3258.
[58] J. Li, H. Xie, J. Li, Z. Wang, and Y. Zhang, "Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6458–6467.
[59] K. Sun et al., "Domain general face forgery detection by learning to weight," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 2638–2646.
[60] A. V. Nadimpalli and A. Rattani, "On improving cross-dataset generalization of deepfake detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 91–99.
[61] A. V. Nadimpalli and A. Rattani, "GBDF: Gender balanced deepfake dataset towards fair deepfake detection," in *Proc. ICPR Workshops*, 2022, pp. 320–337.
[62] A. V. Nadimpalli and A. Rattani, "ProActive DeepFake detection using GAN-based visible watermarking," *ACM Trans. Multimedia Comput., Commun., Appl.*, Sep. 2023, doi: 10.1145/3625547.
[63] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 11976–11986.
[64] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*.
[65] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural Comput.*, vol. 9, no. 1, pp. 1–42, 1997.
[66] *Faceswap*. Accessed: Feb. 17, 2023. [Online]. Available: https://github.com/MarekKowalski/FaceSwap/
[67] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A large-scale challenging dataset for DeepFake forensics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3207–3216.
[68] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (DFDC) preview dataset," 2019, *arXiv:1910.08854*.
[69] B. Dolhansky et al., "The DeepFake detection challenge (DFDC) dataset," 2020, *arXiv:2006.07397*.
[70] B. Zi, M. Chang, J. Chen, X. Ma, and Y.-G. Jiang, "WildDeepfake: A challenging real-world dataset for deepfake detection," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 2382–2390.
[71] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[72] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
[73] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.
[74] S. Chen, T. Yao, and Y. Chen, "Local relation learning for face forgery detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 2, pp. 1081–1088.
[75] Y. Zheng, J. Bao, D. Chen, M. Zeng, and F. Wen, "Exploring temporal coherence for more general video face forgery detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15044–15054.
[76] L. Chen, Y. Zhang, Y. Song, L. Liu, and J. Wang, "Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 18710–18719.
[77] J. Li, H. Xie, L. Yu, and Y. Zhang, "Wavelet-enhanced weakly supervised local feature learning for face forgery detection," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 1299–1308.
[78] Z. Shi, H. Chen, L. Chen, and D. Zhang, "Discrepancy-guided reconstruction learning for image forgery detection," 2023, *arXiv:2304.13349*.
[79] M. Li et al., "Spatio-temporal catcher: A self-supervised transformer for deepfake video detection," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 8707–8718.
[80] 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 Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

Weinan Guan (Student Member, IEEE) received the B.Eng. degree from Northeastern University, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Artificial Intelligence, University of Chinese Academy of Sciences; and the State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS), Institute of Automation, Chinese Academy of Sciences, China. His current research focuses on deepfake detection and computer vision.



Wei Wang (Member, IEEE) received the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences (CASIA), in 2012. He is currently an Associate Professor with the State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS), CASIA. His research interests include artificial intelligence safety and multimedia forensics.

Jing Dong (Senior Member, IEEE) received the Ph.D. degree in pattern recognition from the Institute of Automation, Chinese Academy of Sciences, China, in 2010. She joined the Institute of Automation, Chinese Academy of Sciences, where she is currently a Professor. Her research interests include pattern recognition, image processing, and digital image forensics, including digital watermarking, steganalysis, and tampering detection. She served as the Deputy General of the Chinese Association for Artificial Intelligence.

Bo Peng (Member, IEEE) received the B.Eng. degree from Beihang University in 2013 and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2018. In 2018, he joined the Institute of Automation, Chinese Academy of Sciences, where he is currently an Associate Professor. His current research focuses on computer vision and image forensics.