Transcending Forgery Specificity with Latent Space Augmentation for Generalizable Deepfake Detection

Zhiyuan Yan[1] Yuhao Luo[1] Siwei Lyu[2] Qingshan Liu[3] Baoyuan Wu[1,†]
[1]The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), China
[2]University at Buffalo, State University of New York, USA
[3]Nanjing University of Information Science and Technology, China
yanzhiyuan1114@gmail.com, luo7502@gmail.com
siweilyu@buffalo.edu, qsliu@nuist.edu.cn, wubaoyuan@cuhk.edu.cn

Abstract

Deepfake detection faces a critical generalization hurdle, with performance deteriorating when there is a mismatch between the distributions of training and testing data. A broadly received explanation is the tendency of these detectors to be overfitted to forgery-specific artifacts, rather than learning features that are widely applicable across various forgeries. To address this issue, we propose a simple yet effective detector called LSDA (Latent Space Data Augmentation), which is based on a heuristic idea: representations with a wider variety of forgeries should be able to learn a more generalizable decision boundary, thereby mitigating the overfitting of method-specific features (see Fig. 1). Following this idea, we propose to enlarge the forgery space by constructing and simulating variations within and across forgery features in the latent space. This approach encompasses the acquisition of enriched, domain-specific features and the facilitation of smoother transitions between different forgery types, effectively bridging domain gaps. Our approach culminates in refining a binary classifier that leverages the distilled knowledge from the enhanced features, striving for a generalizable deepfake detector. Comprehensive experiments show that our proposed method is surprisingly effective and transcends state-of-the-art detectors across several widely used benchmarks.

1. Introduction

Deepfake technology has rapidly gained prominence due to its capacity to produce strikingly realistic visual content. Unfortunately, this technology can also be used for malicious purposes, *e.g.,* infringing upon personal privacy, spreading misinformation, and eroding trust in digital media. Given these implications, there is an exigent need to devise a reliable deepfake detection system.

The majority of previous deepfake detectors [29, 37, 39, 40, 57, 59, 63] exhibit effectiveness on the within-dataset scenario, but they often struggle on the cross-dataset scenario where there is a disparity between the distribution of the training and testing data. In real-world situations characterized by unpredictability and complexity, one of the most critical measures for a reliable and efficient detector is the generalization ability. However, given that each forgery method typically possesses its specific characteristics, the

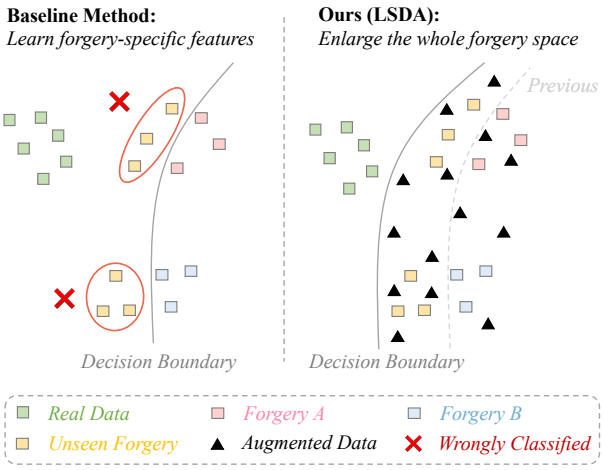

Figure 1. Toy examples for intuitively illustrating our proposed latent space augmentation strategy. The baseline can be overfitted to forgery-specific features and thus cannot generalize well for unseen forgeries. In contrast, our proposed method avoids overfitting to specific forgery features by enlarging the forgery space through latent space augmentation. This approach aims to equip our method with the capability to effectively adjust and adapt to new and previously unseen forgeries.

†Corresponding Author

overfitting to a particular type of forgery may impede the model's ability to generalize effectively to other types (also indicated in previous works [34, 42, 54]).

In this paper, we address the generalization problem of deepfake detection from a heuristic idea: *enlarging the forgery space through interpolating samples encourages models to learn a more robust decision boundary and helps alleviate the forgery-specific overfitting*. We visually demonstrate our idea in Fig. 1, providing an intuitive understanding. Specifically, to learn a comprehensive representation of the forgery, we design several tailored augmentation methods both within and across domains in the latent space. For the within-domain augmentation, our approach involves diversifying each forgery type by interpolating challenging examples[1]. The rationale behind this approach is that challenging examples expand the space within each forgery domain. For the cross-domain augmentation, we utilize the effective Mixup augmentation technique [58] to facilitate smooth transitions between different types of forgeries by interpolating latent vectors with distinct forgery features.

Moreover, inspired by previous work [20], we leverage the pre-trained face recognition model ArcFace [10] to help the detection model learn a more robust and comprehensive representation for the real. It is reasonable to believe that the pre-trained face recognition model has already captured comprehensive features for real-world faces. Therefore, we can employ these learned features to finetune our classifier to learn features of the real. Our approach culminates in refining a binary classification model that leverages the distilled knowledge from the comprehensive forgery and the real features. In this manner, we aim to strive for a more generalizable deepfake detector.

Our proposed latent space method offers the following potential advantages compared to other RGB-based augmentations [4, 28, 29, 60]. **Robustness**: these RGB-based methods typically synthesize new face forgeries (pseudo fake) through pixel-level blending to reproduce simulated artifacts, *e.g.,* blending artifacts [28, 60]. However, these artifacts could be susceptible to alterations caused by post-processing steps, such as compression and blurring (as verified in Fig. 3). In contrast, since our proposed augmentation only operates in the latent space, it does not directly produce and rely on pixel-level artifacts for detection. **Extensibility**: these RGB-based methods typically rely on some specific artifacts (*e.g.*, blending artifacts), which may have limitations in detecting entire face synthesis [27] (as verified in Tab. 3). This limitation stems from the fact that these methods typically define a "fake image" as one in which the face-swapping operation (blending artifact) is present. In contrast, our method aims to perform augmentations in the latent space that do not explicitly depend on these specific pixel-level artifacts for detection.

Our experimental studies confirm the effectiveness of our proposed method. We surprisingly observe a substantial improvement over the baseline methods within the deepfake benchmark [55]. Moreover, our method demonstrates enhanced generalization and robustness in the context of cross-dataset generalization, favorably outperforming recent state-of-the-art detectors.

[1]Challenging examples are that farthest from the center. Within each mini-batch, they are determined by measuring the Euclidean distance between the mean of the samples and other samples.

2. Related Work

Deepfake Generation Methods Deepfake generation typically involves face-replacement [9, 16, 31], face-reenactment [47, 48], and entire image synthesis [26, 27]. Face-replacement generally involves the ID swapping utilizing the auto-encoder-based [9, 31] or graphics-based swapping methods [16], whereas face-reenactment utilizes the reenactment technology to swap the expressions of a source video to a target video while maintaining the identity of the target person. In addition to the face-swapping forgeries above, entire image synthesis utilizes generative models such as GAN [26, 27] and Diffusion models [22, 38, 43] to generate whole synthesis facial images directly without face-swapping operations such as blending. Our work specifically focuses on detecting face-swapping but also shows the potential to detect entire image synthesis.

Deepfake Detectors toward Generalization The task of deepfake detection grapples profoundly with the issue of generalization. Recent endeavors can be classified into the detection of image forgery and video forgery. The field of detecting image forgery have developed novel solutions from different directions: data augmentation [4, 28, 29, 42, 60], frequency clues [17, 33, 34, 37, 52], ID information [13, 23], disentanglement learning [32, 54, 56], designed networks [7, 59], reconstruction learning [3, 50], and 3D decomposition [64]. More recently, several works [24, 45] attempt to generalize deepfakes with the designed training-free pipelines. On the other hand, recent works of detecting video forgery focus on the temporal inconsistency [19, 53, 61], eye blinking [30], landmark geometric features [44], neuron behaviors [51], optical flow [2].

Deepfake Detectors Based on Data Augmentation One effective approach in deepfake detection is the utilization of data augmentation, which involves training models using synthetic data. For instance, in the early stages, FWA [29] employs a self-blending strategy by applying image transformations (*e.g.,* down-sampling) to the facial region and then warping it back into the original image. This process is designed to learn the wrapping artifacts during the deepfake generation process. Another noteworthy contribution is Face X-ray [28], which explicitly encourages detectors to learn the blending boundaries of fake images.

Similarly, I2G [60] uses a similar method of Face X-ray to generate synthetic data and then employs a pair-wise self-consistency learning technique to detect inconsistencies within fake images. Furthermore, SLADD [4] introduces an adversarial method to dynamically generate the most challenging blending choices for synthesizing data. Rather than swapping faces between two different identities, a recent art, SBI [42], proposes to swap with the same person's identity to reach a high-realistic face-swapping.

3. Method

3.1. Architecture Summary

Our framework follows a novel **distillation-based learning architecture** beyond previous methods that train all data in a unique architecture. Our architecture consists of the teacher and student modules. **Teacher module** involves: (1) Assigning a dedicated teacher encoder to learn domain-specific features for each forgery type; (2) Applying within- and cross-domain augmentations to augment the forgery types; (3) Employing a fusion layer to combine and fuse the features with the augmented. **Student module** contains a single student encoder with an FC layer. This encoder benefits from the learned features of the teacher module.

3.2. Training Procedure

The overall training process is summarized in Fig. 2. In the proposed framework, fake and real features are separately learned using distinct teacher encoders, facilitated by the **domain loss** (see "Training Step 1" in Fig. 2). In this step, the **latent augmentation module** is applied to augment the forgery types. Subsequently, the learned features from both real and fake teacher encoders are combined to distill a student encoder with a binary classifier, guided by the **distillation loss** (see "Training Step 2" in Fig. 2). This student encoder is then encouraged to detect deepfakes (via the **binary loss**) using the features acquired from the teachers. During the whole training process, all teacher and student encoders are **trained jointly in an end-to-end manner.** The rationale is that we aim to perform latent augmentation only within the forgery space. By maintaining this separation, we aim to avoid the unintended combination of features from both real and fake instances. This approach aligns with our objective of expanding the forgery space without introducing real features.

3.3. Latent Space Augmentation

Suppose that we have a training dataset $\mathcal{D} = \bigcup_{i=0}^{m} d_i$, which contains m type forgery images $\bigcup_{i=1}^{m} d_i$ and corresponding real type images d_0. First, we sample a batch of identities (face identities) and collect their image from each type of the dataset $\mathcal{D}$, where $\{\mathbf{x}_i \in \mathbb{R}^{B\times H\times W\times 3} | \mathbf{x}_i \in d_i, i = 0, 1, ..., m\}$. After inputting different types of images into the corresponding teacher encoder f_i, we perform our proposed latent space augmentation on the features $\mathbf{z}_i = f_i(\mathbf{x}_i)$, where $\mathbf{z}_i \in \mathbb{R}^{B\times C\times h\times w}$ and $i = 0, 1, ..., m$.

As depicted in Fig. 2, there are three different within-domain transformations, including the Centrifugal transformation (CT), Additive transformation (AdT), Affine transformation (AfT), and the cross-domain transformation. We will introduce these augmentation methods as follows.

3.3.1 Within-domain Augmentation

The within-domain augmentation (WD) contains three specific techniques: centrifugal, affine, and additive transformations. The Centrifugal transformation serves to create hard examples (far away from the centroid) that could encourage models to learn a more general decision boundary, as also indicated in [42]. The latter two transformations are designed to help models learn a more robust representation by adding different perturbations.

Centrifugal Transformation We argue that incorporating challenging examples effectively enlarges the space within each forgery domain. Challenging examples, in this context, refer to samples that are situated far from the domain centroid. Therefore, transforming samples into challenging examples is to drive them away from the domain centroid $\boldsymbol{\mu}_i \in \mathbb{R}^{C\times h\times w}$, which can be computed by

$$\boldsymbol{\mu}_i = \frac{1}{B}\sum_{j=1}^{B}(\mathbf{z}_i)_j, i = 1, ..., m, \tag{1}$$

where $(\mathbf{z}_i)_j \in \mathbb{R}^{C\times h\times w}$ represents the j-th identity features within the batch B of domain i. We propose two kinds of augmentation methods that achieve our purpose in a **direct** and **indirect** manner, respectively.

- **Direct manner**: We force $\mathbf{z}_i$ to move along the centrifugal direction as follows:

$$\hat{\mathbf{z}}_i = \mathbf{z}_i + \beta(\mathbf{z}_i - \boldsymbol{\mu}_i), i = 1, ..., m, \tag{2}$$

  where β is a scaling factor randomly sampled between 0 and 1.
- **Indirect manner**: We push $\mathbf{z}_i$ towards existing hard examples $\mathbf{a}_i \in \mathbb{R}^{C\times h\times w}$, the sample with the largest Euclidean distance from the center $\boldsymbol{\mu}_i$. We then transform $\mathbf{z}_i$ move towards hard examples by:

$$\hat{\mathbf{z}}_i = \mathbf{z}_i + \beta(\mathbf{a}_i - \mathbf{z}_i), i = 1, ..., m. \tag{3}$$

  Here, β is a scaling factor randomly sampled between 0 and 1.

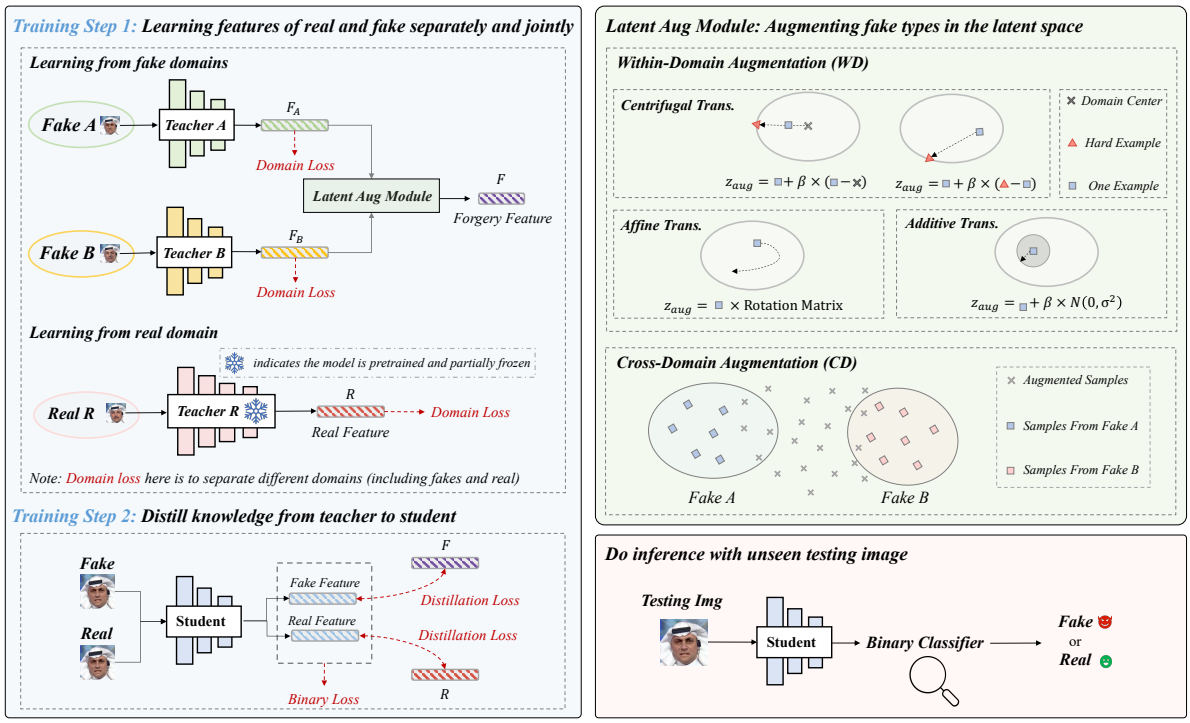

Figure 2. The overall pipeline of our proposed method (two fake types are considered as an example). **(1)** In the training phase, the student encoder is trained to learn a generalizable and robust feature by utilizing the distribution match to distill the knowledge of the real and fake teacher encoders to the student encoder. **(2)** In the inference phase, only the student encoder is applied to detect the fakes from the real. **(3)** For the learning of the forgery feature, we apply the latent space within-domain (WD) and cross-domain (CD) augmentation. **(4)** For the learning of the real feature, the pre-trained and frozen ArcFace face recognition model is applied. **(5)** WD involves novel augmentations to fine-tune domain-specific features, while CD enables the model to seamlessly identify transitions between different types of forgeries.

Affine Transformation Affine transformation is proposed to transform the element-wise position information, creating neighboring samples. Specifically, when we perform an affine rotation on $\mathbf{z}_i$ with rotation angle $\boldsymbol{\theta}$ in radians, we can derive the corresponding affine rotation matrix $\mathbf{A}$ as:

$$\mathbf{A} = \begin{bmatrix} \cos(\boldsymbol{\theta}) & -\sin(\boldsymbol{\theta}) & 0 \\ \sin(\boldsymbol{\theta}) & \cos(\boldsymbol{\theta}) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

After multiplying $\mathbf{A}$ with $\mathbf{P}$, the position information of $\mathbf{z}_i$ (*i.e.,* the coordinate of each element in $\mathbf{z}_i$), the rotated position information $\hat{\mathbf{P}}$ is given by $\hat{\mathbf{P}} = \mathbf{AP}$. Then, we can obtain the rotated feature $\hat{\mathbf{z}_i}$ by rearranging elements' positions according to $\hat{\mathbf{P}}$.

Additive Transformation Adding perturbation is a traditional and effective augmentation, we apply this technique in latent space. By adding random noise, for example, Gaussian Mixture Model noise with zero mean, $\mathbf{z}_i$ can be perturbed with the scaling factor β as follows:

$$\hat{\mathbf{z}_i} = \mathbf{z}_i + \beta\boldsymbol{\epsilon}, \tag{5}$$

where $\boldsymbol{\epsilon} \sim \sum_{k=1}^{G} \pi_k \mathcal{N}(\boldsymbol{\epsilon}|0, \boldsymbol{\Sigma}_k)$ and $\sum_{k=1}^{G} \pi_k = 1$.

3.3.2 Cross-domain Augmentation

To create and interpolate the variants between different forgery domains, we utilize the Mixup augmentation technique [58] in the latent space for cross-domain augmentation. This approach encourages the model to learn a more robust decision boundary and capture the general features shared across various forgeries. Specifically, we compute a linear combination of two latent representations: $\mathbf{z}_i$ and $\mathbf{z}_k$ that belong to different fake domains ($i \neq k$). The weight between two features is controlled by α, which is randomly sampled between 0 and 1. The augmentation can be formally expressed as:

$$\hat{\mathbf{z}_i}^c = \alpha\mathbf{z}_i + (1-\alpha)\mathbf{z}_k, i \neq k \in \{1, ..., m\}, \tag{6}$$

where i and k are distinct forgery domains and $\hat{\mathbf{z}_i}^c$ stands for cross-domain augmented samples.

3.3.3 Fusion layer

Within each mini-batch, we perform both within-domain and cross-domain augmentation on $\mathbf{z}_i$ and obtain corresponding augmented representation $\hat{\mathbf{z}_i} \in \mathbb{R}^{B\times C\times h\times w}$ and

$\hat{\mathbf{z}}_i^{\,c} \in \mathbb{R}^{B\times C\times h\times w}$, respectively. Then, we apply a learnable convolutional layer to bring augmentation results together to align the shape with the output of the student encoder:

$$\hat{\mathbf{z}}_i^{aug} = \mathit{Conv}(\hat{\mathbf{z}}_i \parallel \hat{\mathbf{z}}_i^{\,c}), i = 1, ..., m, \tag{7}$$

where $\parallel$ represents the concatenation operation along the channel dimension. Thus the final latent representation $\mathbf{F}_i \in \mathbb{R}^{B\times C\times h\times w}$ of forgery augmentation can be obtained by combining the original forgery representations and the augmented representations:

$$\mathbf{F}_i = \mathit{Conv}(\hat{\mathbf{z}}_i^{aug} \parallel \hat{\mathbf{z}}_i), i = 1, ..., m. \tag{8}$$

3.4. Objective Function

Domain Loss The domain loss is designed to encourage teacher encoders to learn domain-specific features (with each forgery type and the real category considered as distinct domains). After teacher encoders compress images $\mathbf{x}_i \in \mathbb{R}^{B\times H\times W\times 3}$ to $\mathbf{z}_i \in \mathbb{R}^{B\times C\times h\times w}$ in the latent space, we apply a multi-class classifier to estimate the confidence score $\mathbf{s}_i \in \mathbb{R}^{B\times(m+1)}$ that the feature is recognized as each domain. The domain loss, given as a multi-class classification loss, can be represented by the Cross-Entropy Loss. At first, we turn the confidence score $\mathbf{s}_i$ into the likelihood $\mathbf{p}_i \in \mathbb{R}^B$: after the softmax, taking the i-th result, which is formulated as $\mathbf{p}_i = \mathrm{softmax}(\mathbf{s}_i)[i]$. Then we compute the domain loss as follows:

$$\mathcal{L}_{domain} = -\frac{1}{B\times(m+1)}\times \sum_{j=1}^{B}\left[\log(1-(\mathbf{p}_0)_j) + \sum_{i=1}^{m}\log((\mathbf{p}_i)_j)\right], \tag{9}$$

where $(\mathbf{p}_i)_j \in \mathbb{R}$ represents the forgery probability of j-th identity features within the batch B of domain i (0 is the real type).

Distillation Loss The distillation loss is the key loss to improve the generalization ability of the inference model by transferring augmented knowledge to the student: align the student's feature $\mathbf{F}_i^s$ with augmented latent representation $\mathbf{F}_i$. This alignment process is quantified using a distance measurement function $M(\cdot)$, which is formally as:

$$\mathcal{L}_{distill} = \sum_{i=0}^{m} M(\mathbf{F}_i, \mathbf{F}_i^s). \tag{10}$$

In the context of fake samples, the goal is to adjust the student model's feature map $\mathbf{F}_i^s, i = 1, ..., m$ to approximate the comprehensive forgery representation $\mathbf{F}_i, i = 1, ..., m$, where $\mathbf{F}_i$ is obtained by Eq. (8). Similarly, we align the student's feature map of the real $\mathbf{F}_0^s$ to the teacher's real representation $\mathbf{F}_0$, where $\mathbf{F}_0$ is obtained by utilizing the pre-trained ArcFace [10] model.

Binary Classification Loss To finally achieve the Deepfake detection task, we add a binary classifier to the student encoder for detecting fakes from the real. The binary classification loss, commonly known as Binary Cross-Entropy, is formulated as follows:

$$\mathcal{L}_{binary} = -\frac{1}{B\times(m+1)}\times \sum_{j=1}^{B}\left[\log(1-(\mathbf{p}_0)_j) + \sum_{i=1}^{m}\log((\mathbf{p}_i)_j)\right]. \tag{11}$$

In this equation, B represents the batch size of observations, and $\mathbf{p}_i$ is the predicted probability that observation $\mathbf{x}_i$ belongs to the class indicative of a deepfake, where $i = 0, 1, ..., m$.

Overall Loss The final loss function is obtained by the weighted sum of the above loss functions.

$$\mathcal{L} = \lambda_1\mathcal{L}_{binary} + \lambda_2\mathcal{L}_{domain} + \lambda_3\mathcal{L}_{distill}, \tag{12}$$

where λ_1, λ_2, and λ_3 are hyper-parameters for balancing the overall loss.

4. Experiments

4.1. Settings

Datasets. To evaluate the generalization ability of the proposed framework, our experiments are conducted on several commonly used deepfake datasets: FaceForensics++ (FF++) [39], DeepfakeDetection (DFD) [8], Deepfake Detection Challenge (DFDC) [12], preview version of DFDC (DFDCP) [11], and CelebDF (CDF) [31]. FF++ [39] is a large-scale database comprising more than 1.8 million forged images from 1000 pristine videos. Forged images are generated by four face manipulation algorithms using the same set of pristine videos, *i.e.,* DeepFakes (DF) [9], Face2Face (F2F) [47], FaceSwap (FS) [16], and NeuralTexture (NT) [48]. Note that there are three versions of FF++ in terms of compression level, *i.e.*, raw, lightly compressed (c23), and heavily compressed (c40). Following previous works [4, 5, 28], the c23 version of FF++ is adopted.

Implementation Details. We employ EfficientNet-B4 [46] as the default encoders to learn forgery features. For the real encoder, we employ the model and pre-trained weights of ArcFace from the code[2]. The model parameters are initialized through pre-training on the ImageNet. We also explore alternative network architectures and their respective results, which are presented in the **supplementary**. We employ MSE loss as the feature alignment function (M in eq. (10)). Empirically, the λ_1, λ_2, and λ_3 are set to be

[2] https://github.com/mapooon/BlendFace.

Method	Detector	Backbone	CDF-v1	CDF-v2	DFD	DFDC	DFDCP	Avg.
Naive	Meso4 [1]	MesoNet	0.736	0.609	0.548	0.556	0.599	0.610
Naive	MesoIncep [1]	MesoNet	0.737	0.697	0.607	0.623	0.756	0.684
Naive	CNN-Aug [21]	ResNet	0.742	0.703	0.646	0.636	0.617	0.669
Naive	Xception [39]	Xception	0.779	0.737	<u>0.816</u>	0.708	0.737	0.755
Naive	EfficientB4 [46]	EfficientNet	0.791	0.749	0.815	0.696	0.728	0.756
Spatial	CapsuleNet [35]	Capsule	0.791	0.747	0.684	0.647	0.657	0.705
Spatial	FWA [29]	Xception	0.790	0.668	0.740	0.613	0.638	0.690
Spatial	Face X-ray [28]	HRNet	0.709	0.679	0.766	0.633	0.694	0.696
Spatial	FFD [7]	Xception	0.784	0.744	0.802	0.703	0.743	0.755
Spatial	CORE [36]	Xception	0.780	0.743	0.802	0.705	0.734	0.753
Spatial	Recce [3]	Designed	0.768	0.732	0.812	0.713	0.734	0.752
Spatial	UCF [54]	Xception	0.779	0.753	0.807	<u>0.719</u>	<u>0.759</u>	0.763
Frequency	F3Net [37]	Xception	0.777	0.735	0.798	0.702	0.735	0.749
Frequency	SPSL [33]	Xception	<u>0.815</u>	<u>0.765</u>	0.812	0.704	0.741	<u>0.767</u>
Frequency	SRM [34]	Xception	0.793	0.755	0.812	0.700	0.741	0.760
Ours	EFNB4 + LSDA	EfficientNet	**0.867** (↑5.2%)	**0.830** (↑6.5%)	**0.880** (↑6.4%)	**0.736** (↑1.7%)	**0.815** (↑5.6%)	**0.826** (↑5.9%)

Table 1. Cross-dataset evaluations using the **frame-level AUC** metric on the deepfake benchmark [55]. All detectors are trained on FF++_c23 [39] and evaluated on other datasets. The best results are highlighted in bold and the second is underlined.

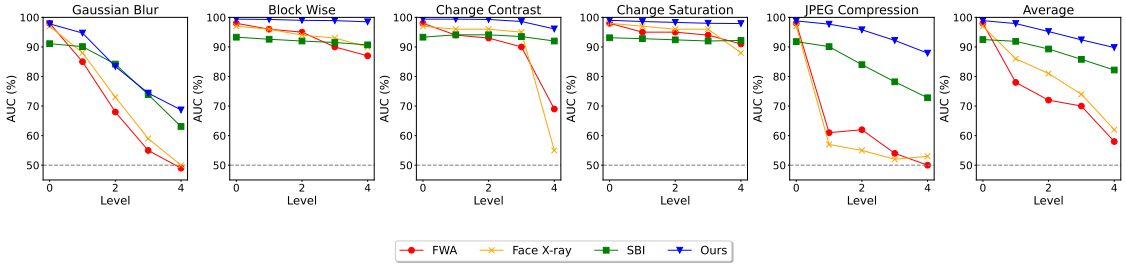

Figure 3. Robustness to Unseen Perturbations: We report video-level AUC (%) under five different degradation levels of five specific types of perturbations [25]. We compare our results with three RGB-based augmentation-based methods to demonstrate our robustness. Best viewed in color.

0.5, 1, and 1 in Eq. (12). We explore other variants in **supplementary**. To ensure a fair comparison, all experiments are conducted within the DeepfakeBench [55]. All of our experimental settings adhere to the default settings of the benchmark. More details are in the **supplementary**.

Evaluation Metrics. By default, we report the **frame-level** Area Under Curve (AUC) metric to compare our proposed method with prior works. Notably, to compare with other state-of-the-art detectors, especially the video-based methods, **we also report the video-level AUC to compare with.** Other evaluation metrics such as Average Precision (AP) and Equal Error Rate (EER) are also reported for a more comprehensive evaluation.

4.2. Generalization Evaluation

All our experiments follow a commonly adopted generalization evaluation protocol by training the models on the FF++_c23 [39] and then evaluating on other previously untrained/unseen datasets (*e.g.*, CDF [31] and DFDC [12]).

Comparison with competing methods. We first conduct generalization evaluation on a unified benchmark (*i.e.*, DeepfakeBench [55]). The rationale is that although many previous works have adopted the same datasets for training and testing, the pre-processing, experimental settings, *etc*, employed in their experiments can vary. This variation makes it challenging to conduct fair comparisons. Thus, we implement our method and report the results using DeepfakeBench [55]. For other competing detection methods, we directly cite the results in the DeepfakeBench and use the same settings in implementing our method for a fair comparison. The results of the comparison between different methods are presented in Tab. 1. It is evident that our method consistently outperforms other models across all tested scenarios. On average, our approach achieves a

Model	Publication	CDF-v2	DFDC
LipForensics [19]	CVPR'21	0.824	0.735
FTCN [61]	ICCV'21	0.869	0.740
PCL+I2G [60]	ICCV'21	0.900	0.744
HCIL [18]	ECCV'22	0.790	0.692
RealForensics [20]	CVPR'22	0.857	<u>0.759</u>
ICT [14]	CVPR'22	0.857	-
SBI* [42]	CVPR'22	<u>0.906</u>	0.724
AltFreezing [53]	CVPR'23	0.895	-
Ours	-	**0.911** (↑0.05%)	**0.770** (↑1.1%)

Table 2. Comparison with recent state-of-the-art methods on CDF-v2 and DFDC using the **video-level AUC**. We report the results directly from the original papers. All methods are trained on FF++_c23. * denotes our reproduction with the official code. The best results are in bold and the second is underlined.

Method	Testing Datasets			
	StarGAN [6]	DDPM [22]	DDIM [43]	SD [38]
SBI [42]	0.787	0.744	0.648	0.478
Ours	**0.810**	**0.854**	**0.748**	**0.506**

Table 3. Results in detecting GAN-generated images and Diffusion-generated images. We compare our results with SBI [42]. We utilize its official code for evaluation. These models are trained on FF++_c23. "SD" is the short for stable diffusion.

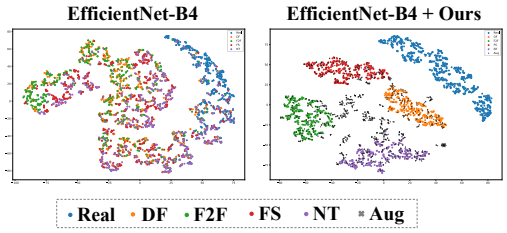

Figure 4. t-SNE visualization of latent space *w* and *wo* augmentations.

notable 5% improvement in performance.

Comparison with state-of-the-art methods. In addition to the detectors implemented in DeepfakeBench, we further evaluate our method against other state-of-the-art models. We report the video-level AUC metric for comparison. We select the recently advanced detectors for comparison, as listed in Tab. 2. Generally, the results are directly cited from their original papers. In the case of SBI, it is worth noting that the original results are obtained from training on the raw version of FF++, whereas other methods are trained on the c23 version. To ensure a fair and consistent comparison, we reproduce the results for SBI under the same conditions as the other methods. The results, as shown in Tab. 2, show the effective generalization of our method as it outperforms other methods, achieving the best performance on both CDF-v2 and DFDC.

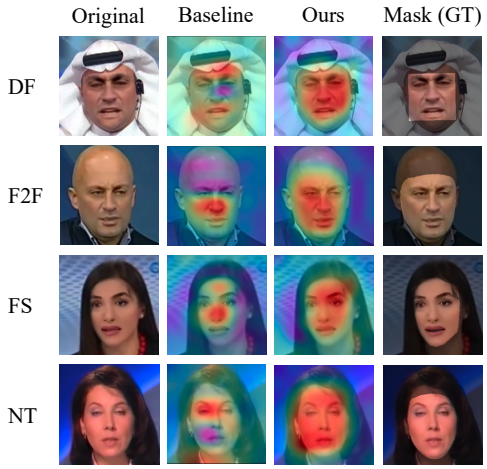

Figure 5. GradCAM visualizations [41] for fake samples from different forgeries. We compare the baseline (EFNB4 [46] with ours. "Mask (GT)" highlights the ground truth of the manipulation region. Best viewed in color.

Comparison with RGB-based augmentation methods. To show the advantages of the latent space augmentation method (ours) over RGB-based augmentations (*e.g.*, FWA [29], SBI [42]), we conduct several evaluations as follows. **Robustness:** RGB-based methods typically rely on subtle low-level artifacts at the pixel level. These artifacts could be sensitive to unseen random perturbations in real-world scenarios. To assess the model's robustness to such perturbations, we follow the approach of previous works [19]. Fig. 3 presents the video-level AUC results for these unseen perturbations, utilizing the model trained on FF++_c23. Notably, our method exhibits a significant performance advantage of robustness over other RGB-based methods. **Extensibility**: RGB-based methods classify an image as "fake" if it contains evidence of a face-swapping operation, typically blending artifacts. Beyond the evaluations on face-swapping datasets, we have extended our evaluation to include the detection in scenarios of entire face synthesis, which do not encompass blending artifacts. For this evaluation, we compare our method SBI [42] that mainly relies on blending artifacts. The models are evaluated on both GAN-generated and Diffusion-generated data. Remarkably, our method consistently outperforms SBI across all testing datasets (see Tab. 3). This observation shows the better extensibility of our detectors, which do not rely on specific artifacts like blending.

4.3. Ablation Study

Effects of the latent space augmentation strategy. To evaluate the impact of the two proposed augmentation strategies (WD and CD), we conduct ablation studies on

WD	CD	CDF-v1			CDF-v2			DFDCP			DFDC			Avg.		
		AUC	AP	EER	AUC	AP	EER	AUC	AP	EER	AUC	AP	EER	AUC	AP	EER
×	×	0.775	0.843	28.6	0.752	0.847	31.3	0.737	0.846	32.9	0.697	0.721	36.6	0.755	0.846	31.1
×	✓	0.862	0.902	21.1	0.819	0.888	26.0	0.807	0.891	27.6	0.733	0.760	33.5	0.821	0.885	25.5
✓	×	**0.887**	**0.925**	**18.5**	**0.833**	0.903	**24.6**	0.787	0.869	28.6	0.729	0.750	33.2	0.819	0.885	25.4
✓	✓	0.867	0.922	21.9	0.830	**0.904**	25.9	**0.815**	**0.893**	**26.9**	**0.736**	**0.760**	**33.0**	**0.825**	**0.893**	**25.5**

Table 4. Ablation studies regarding the effectiveness of the within-domain (WD) and cross-domain (CD) augmentation strategies. All models are trained on the FF++_c23 dataset and evaluated across various other datasets with metrics presented in the order of AUC | AP | EER (the frame-level). The average performance (Avg.) across all datasets are also reported. The best results are highlighted in bold.

Real Encoder	CDF-v1			CDF-v2			DFDCP			DFDC			Avg.		
	AUC	AP	EER	AUC	AP	EER	AUC	AP	EER	AUC	AP	EER	AUC	AP	EER
EFNB4 [46]	0.857	0.908	22.4	0.822	0.893	25.8	0.805	0.885	27.3	0.733	0.759	33.3	0.804	0.861	27.2
iResNet101 [15]	0.854	0.908	23.0	0.792	0.874	28.1	0.797	0.872	27.2	0.715	0.743	35.7	0.790	0.849	28.5
ArcFace [10]	**0.867**	**0.922**	**21.9**	**0.830**	**0.904**	**25.9**	**0.815**	**0.893**	**26.9**	**0.736**	**0.760**	**33.0**	**0.812**	**0.870**	**26.9**

Table 5. Ablation studies regarding the effectiveness of the ArcFace pre-trained before the real encoder. The experimental settings are similar to Table. 4.

Figure 6. Visual examples of the original and augmented data.

several datasets. The evaluated variants include the baseline EfficientNet-B4, the baseline with the proposed within-domain augmentation (WD), the cross-domain augmentation (CD), and our overall framework (WD + CD). The incremental enhancement in the overall generalization performance with the addition of each strategy, as evidenced by the results in Tab. 4, shows the effectiveness of these strategies. We also conduct ablation studies for each WD method in the **Supplementary**.

Effects of face recognition prior. To assess the impact of the face recognition network (ArcFace [10]), we perform an ablation study comparing the results obtained using ArcFace (*w* iResNet101 as the backbone) as the real encoder, to those achieved with the default backbone (*i.e.*, EFNB4) and iResNet101 as the real encoder. As shown in Tab. 5, employing ArcFace as the real encoder results in notably better performance compared to using EFNB4 and iResNet101 (*wo* face recognition pretraining) as the real encoder. This highlights the importance of utilizing the knowledge gained from face recognition, as offered by ArcFace, for deepfake detection tasks. Our findings align with those reported in our previous studies [19, 20].

5. Visualizations

Visualizations of the captured artifacts. We further use GradCAM [62] to localize which regions are activated to detect forgery. The visualization results shown in Fig. 5 demonstrate that the baseline captures forgery-specific artifacts with a similar and limited area of response across different forgeries, while our model could locate the forgery region precisely and meaningfully. In contrast, our method makes it discriminates between real and fake by focusing predominantly on the manipulated face area. This visualization further identifies that LSDA encourages the baseline to capture more general forgery features.

Visualizations of learned latent space. We utilize t-SNE [49] for visualizing the feature space. We visualize the results on the FF++_c23 testing datasets by randomly selecting 5000 samples. Results in Fig. 4 show our augmented method (the right) indeed learns a more robust decision boundary than the un-augmented baseline (the left).

6. Conclusion

In this paper, we propose a simple yet effective detector that can generalize well in unseen deepfake datasets. Our key is that representations with a wider range of forgeries should learn a more adaptable decision boundary, thereby mitigating the overfitting to forgery-specific features. Following this idea, we propose to enlarge the forgery space by constructing and simulating variations within and across forgery features in the latent space. Extensive experiments show that our method is superior in generalization and robustness to state-of-the-art methods. We hope that our work will stimulate further research into the design of data augmentation in the deepfake detection community.

Acknowledgment. Baoyuan Wu was supported by the National Natural Science Foundation of China under grant No.62076213, Shenzhen Science and Technology Program under grant No.RCYX20210609103057050, and the Longgang District Key Laboratory of Intelligent Digital Economy Security. Qingshan Liu was supported by the National Natural Science Foundation of China under grant NSFC U21B2044. Siwei Lyu was supported by U.S. National Science Foundation under grant SaTC-2153112.

References

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *Proceedings of the IEEE International Workshop on Information Forensics and Security*, 2018. 6

[2] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and Alberto Del Bimbo. Deepfake video detection through optical flow based cnn. In *Proceedings of the IEEE/CVF Conference on International Conference on Computer Vision*, pages 0–0, 2019. 2

[3] Junyi Cao, Chao Ma, Taiping Yao, Shen Chen, Shouhong Ding, and Xiaokang Yang. End-to-end reconstruction-classification learning for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4113–4122, 2022. 2, 6

[4] 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. 2, 3, 5

[5] Liang Chen, Yong Zhang, Yibing Song, Jue Wang, and Lingqiao Liu. Ost: Improving generalization of deepfake detection via one-shot test-time training. In *Proceedings of the Neural Information Processing Systems*, 2022. 5

[6] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. 7

[7] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K Jain. On the detection of digital face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2, 6

[8] Deepfakedetection, 2021. https://ai.googleblog.com/2019/09/contributing-data-to-deepfakedetection.html Accessed 2021-11-13. 5

[9] DeepFakes, 2020. www.github.com/deepfakes/faceswap Accessed 2020-09-02. 2, 5

[10] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 2, 5, 8

[11] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854*, 2019. 5

[12] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397*, 2020. 5, 6

[13] Shichao Dong, Jin Wang, Renhe Ji, Jiajun Liang, Haoqiang Fan, and Zheng Ge. Implicit identity leakage: The stumbling block to improving deepfake detection generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3994–4004, 2023. 2

[14] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Ting Zhang, Weiming Zhang, Nenghai Yu, Dong Chen, Fang Wen, and Baining Guo. Protecting celebrities from deepfake with identity consistency transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9468–9478, 2022. 7

[15] Ionut Cosmin Duta, Li Liu, Fan Zhu, and Ling Shao. Improved residual networks for image and video recognition. In *Proceedings of the IEEE International Conference on Pattern Recognition*, pages 9415–9422. IEEE, 2021. 8

[16] FaceSwap, 2021. www.github.com/MarekKowalski/FaceSwap Accessed 2020-09-03. 2, 5

[17] Qiqi Gu, Shen Chen, Taiping Yao, Yang Chen, Shouhong Ding, and Ran Yi. Exploiting fine-grained face forgery clues via progressive enhancement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 735–743, 2022. 2

[18] Zhihao Gu, Taiping Yao, Yang Chen, Shouhong Ding, and Lizhuang Ma. Hierarchical contrastive inconsistency learning for deepfake video detection. In *Proceedings of the European Conference on Computer Vision*, pages 596–613. Springer, 2022. 7

[19] 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*, 2021. 2, 7, 8

[20] Alexandros Haliassos, Rodrigo Mira, Stavros Petridis, and Maja Pantic. Leveraging real talking faces via self-supervision for robust forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14950–14962, 2022. 2, 7, 8

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2, 7

[23] Baojin Huang, Zhongyuan Wang, Jifan Yang, Jiaxin Ai, Qin Zou, Qian Wang, and Dengpan Ye. Implicit identity driven deepfake face swapping detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2023. 2

[24] Shan Jia, Reilin Lyu, Kangran Zhao, Yize Chen, Zhiyuan Yan, Yan Ju, Chuanbo Hu, Xin Li, Baoyuan Wu, and Siwei Lyu. Can chatgpt detect deepfakes? a study of using multimodal large language models for media forensics. *arXiv preprint arXiv:2403.14077*, 2024. 2

[25] 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*, 2020. 6

[26] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2

[27] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 2

[28] 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*, 2020. 2, 5, 6

[29] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656*, 2018. 1, 2, 6, 7

[30] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *Proceedings of the IEEE International Workshop on Information Forensics and Security*, 2018. 2

[31] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A new dataset for deepfake forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2, 5, 6

[32] Jiahao Liang, Huafeng Shi, and Weihong Deng. Exploring disentangled content information for face forgery detection. In *Proceedings of the European Conference on Computer Vision*, pages 128–145. Springer, 2022. 2

[33] 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*, 2021. 2, 6

[34] 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*, 2021. 2, 6

[35] Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. Capsule-forensics: Using capsule networks to detect forged images and videos. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2019. 6

[36] Yunsheng Ni, Depu Meng, Changqian Yu, Chengbin Quan, Dongchun Ren, and Youjian Zhao. Core: Consistent representation learning for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, pages 12–21, 2022. 6

[37] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *Proceedings of the European Conference on Computer Vision*, 2020. 1, 2, 6

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2, 7

[39] 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 Conference on International Conference on Computer Vision*, 2019. 1, 5, 6

[40] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, 2019. 1

[41] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 618–626, 2017. 7

[42] 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. 2, 3, 7

[43] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 7

[44] Zekun Sun, Yujie Han, Zeyu Hua, Na Ruan, and Weijia Jia. Improving the efficiency and robustness of deepfakes detection through precise geometric features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[45] Chuangchuang Tan, Ping Liu, RenShuai Tao, Huan Liu, Yao Zhao, Baoyuan Wu, and Yunchao Wei. Data-independent operator: A training-free artifact representation extractor for generalizable deepfake detection. *arXiv preprint arXiv:2403.06803*, 2024. 2

[46] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 5, 6, 7, 8

[47] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 2, 5

[48] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *Journal of ACM Transactions on Graphics*, 38(4):1–12, 2019. 2, 5

[49] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 2008. 8

[50] 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*, 2021. 2

[51] Run Wang, Felix Juefei-Xu, Lei Ma, Xiaofei Xie, Yihao Huang, Jian Wang, and Yang Liu. Fakespotter: A simple yet robust baseline for spotting ai-synthesized fake faces. *arXiv preprint arXiv:1909.06122*, 2019. 2

[52] Yuan Wang, Kun Yu, Chen Chen, Xiyuan Hu, and Silong Peng. Dynamic graph learning with content-guided spatial-frequency relation reasoning for deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7278–7287, 2023. 2

[53] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, and Houqiang Li. Altfreezing for more general video face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4129–4138, 2023. 2, 7

[54] Zhiyuan Yan, Yong Zhang, Yanbo Fan, and Baoyuan Wu. Ucf: Uncovering common features for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on International Conference on Computer Vision*, pages 22412–22423, 2023. 2, 6

[55] Zhiyuan Yan, Yong Zhang, Xinhang Yuan, Siwei Lyu, and Baoyuan Wu. Deepfakebench: A comprehensive benchmark of deepfake detection. *Advances in Neural Information Processing Systems*, 2023. 2, 6

[56] Tianyun Yang, Juan Cao, Qiang Sheng, Lei Li, Jiaqi Ji, Xirong Li, and Sheng Tang. Learning to disentangle gan fingerprint for fake image attribution. *arXiv preprint arXiv:2106.08749*, 2021. 2

[57] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing.*, 2019. 1

[58] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2, 4

[59] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2

[60] Tianchen Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. Learning self-consistency for deepfake detection. In *Proceedings of the IEEE/CVF Conference on International Conference on Computer Vision*, 2021. 2, 3, 7

[61] Yinglin Zheng, Jianmin Bao, Dong Chen, Ming Zeng, and Fang Wen. Exploring temporal coherence for more general video face forgery detection. In *Proceedings of the IEEE/CVF Conference on International Conference on Computer Vision*, pages 15044–15054, 2021. 2, 7

[62] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016. 8

[63] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Two-stream neural networks for tampered face detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, 2017. 1

[64] Xiangyu Zhu, Hao Wang, Hongyan Fei, Zhen Lei, and Stan Z Li. Face forgery detection by 3d decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2929–2939, 2021. 2