Supplementary for 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

1. Overview

This supplementary material provides additional details regarding our architecture, implementation settings, and additional experimental results, including:

- Detailed Network Architecture (see Section. 2).
- Further Implementation Details (see Section. 3).
- More Ablation Studies (see Section. 4).

2. Detailed Network Architecture

Real Encoder The real encoder (**R**) proposed in the main paper is a pre-trained Arcface network [7]. We freeze the majority of the layers in our pre-trained face recognition model because we believe that these layers have already acquired knowledge about real faces. However, we keep the last block of layers unfrozen so that we can utilize this part of the model for fine-tuning or further training. During the training process, we employ the network as a teacher model to distill the knowledge of real faces to the student model (*i.e.,* a binary classification model). This knowledge transfer is achieved through feature map alignment (see Figure. 2 in the main paper). We aim to encourage the student model to generate feature maps that are near to those produced by the real encoder when presented with real face inputs. The effectiveness of this strategy has been verified through experiments, as detailed in Table. 6 of the manuscript. The details of the real encoder are shown in Figure. 1.

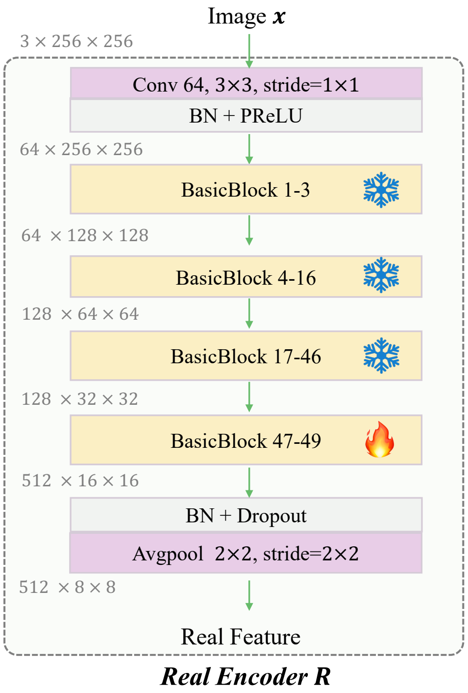

Figure 1. Teacher Real Encoder architecture, an Arcface network with 101 layers. The "BasicBlock" is a typical residual block in the ResNet [11]. "BN" is the short for Batch Normalization.

3. Further Implementation Details

Face-swapping Dataset According to the Section. 2 in our main paper, deepfake can be typically classified into face-swapping forgery (*e.g.,* face-replacement and face-reenactment) and entire facial image synthesis (*e.g.,* GAN and diffusion-generated images). In this work, we mainly focus on the detection of the face-swapping forgery and also show the potential to detect entire image synthesis. We adopt several widely used face-swapping deepfake datasets in the DeepfakeBench [27]: FF++ [19], DFD [5], CDF [16], DFDCP [8], and DFDC [9].

- FF++ [19] is a well-known database used for deepfake detection. The real videos in this dataset are almost interviews or speeches by a single person. FF++ utilizes

[†] Corresponding Author

Ablation	DFD			CelebDF-v1			CelebDF-v2			DFDCP			DFDC			Avg.		
	AUC ↑	AP ↑	EER ↓	AUC ↑	AP ↑	EER ↓	AUC ↑	AP ↑	EER ↓	AUC ↑	AP ↑	EER ↓	AUC ↑	AP ↑	EER ↓	AUC ↑	AP ↑	EER ↓
wo AdT	**0.884**	**0.985**	**19.8**	0.860	0.907	**21.7**	0.824	0.893	**25.7**	0.800	0.890	27.3	0.740	**0.764**	32.7	0.814	0.888	**25.4**
wo AFT	0.869	0.982	21.4	0.847	0.904	22.1	0.809	0.884	26.6	**0.815**	**0.898**	**26.2**	**0.744**	0.762	**32.6**	0.817	0.886	25.8
wo CT	0.881	0.984	19.9	0.847	0.898	23.1	0.793	0.877	28.4	0.793	0.878	28.0	0.734	0.756	33.5	0.810	0.879	26.6
Ours	0.880	0.984	20.0	**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 1. Detailed performance metrics of different ablation studies. The values represent AUC, AP, and EER for each method across various datasets. The average performance (Avg.) across all datasets is also reported. The best results are highlighted in bold.

Encoder Architecture	DFD			CDF-v1			CDF-v2			DFDCP			DFDC			Avg.		
	AUC ↑	AP ↑	EER ↓	AUC ↑	AP ↑	EER ↓	AUC ↑	AP ↑	EER ↓	AUC ↑	AP ↑	EER ↓	AUC ↑	AP ↑	EER ↓	AUC ↑	AP ↑	EER ↓
Xception	**0.888**	**0.986**	20.3	0.823	0.884	23.5	0.806	0.882	27.1	0.796	0.887	28.5	0.729	0.752	34.0	0.808	0.878	26.7
EfficientNet-B1	0.870	0.983	21.4	0.821	0.885	25.1	0.809	0.882	27.2	0.789	0.881	28.6	0.733	0.756	33.5	0.804	0.877	27.2
EfficientNet-B4	0.880	0.984	**20.0**	**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**
EfficientNet-B5	0.877	0.984	21.0	0.848	0.919	22.0	**0.833**	0.898	25.7	0.812	0.892	27.1	**0.748**	**0.772**	32.4	0.824	**0.893**	25.6

Table 2. Performance evaluation of different encoder architectures. 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 is also reported. The best results are highlighted in bold.

four different forgery technologies (*i.e.,* DF [6], F2F [24], FS [10], and NT [25]) to separately generate fake videos from the same 1000 pristine videos. We adopt the official data splits and use 740 videos for training, 140 for validation, and 140 for testing. To evaluate the generalization ability, we adopt the evaluation strategy by training models on the FF++ and testing them on other previously unseen datasets. Note that FF++ has three versions of datasets with different levels of compression. Following previous work [1, 21, 26], we adopt the c23 (light compression/high quality) version for training.

- DFD [5] is a database released by Google, which contains 363 source videos from 28 actors and about 3,000 forged videos. This dataset includes different scenes and characters, more consistent with the face-swapping in the real and complex scenes. Since this dataset does not have the official split settings of the training and testing, we follow DeepfakeBench [27] to use the whole DFD dataset for evaluation. We use the c23 version for the DFD dataset for testing.
- CDF [16] is a large-scale challenging deepfake video database toward celebrities that is generated using the improved synthesis process of DeepFake algorithm [6]. Compared to the DeepFake algorithm applied in the FF++ (FF-DF), CDF utilizes more post-processing technologies to eliminate the visual artifacts, such as the blur, blending boundary, *etc*. This dataset has two versions with similar data sources but different data quantities. CDF-v1 contains 408 real videos and 795 synthesized videos, while CDF-v2 has 5639 fake videos. CDF widely serves as a benchmark for evaluating the generalization performance. In this work, we train our model on FF++ and evaluate it on both CDF-v1 and CDF-v2.
- DFDCP [8] is the preview version of DFDC [9] that is released with the same-named challenge which is held by several corporations and academics to build innovative new technologies for deepfake detection. DFDC dataset contains a lot of disturbed videos, *e.g.,* noise, downsampling, and compression. So far, DFDC is considered as the most challenging deepfake dataset for generalization evaluation. In this work, we train our model on FF++ and evaluate it on both DFDCP and DFDC.

Entire Image Synthesis Dataset In this work, we adopt four typical entire image synthesis datasets generated by GANs or Diffusion models: StarGAN [3], DDPM [12], DDIM [22], and Stable Diffusion [18]. In this work, we train our model on FF++_c23 and use it for the detection of these entire image synthesis datasets (see Table. 4 in the manuscript). This setting could be challenging because both the data source and the manipulation artifacts are relatively distinct. For example, face-swapping forgery methods can produce blending artifacts, but the entire image synthesis does not. Here, we introduce the four entire image synthesis technologies:

- StarGAN [3] is a typical GAN (generative adversarial network) model designed for image-to-image translations across multiple domains with diverse attributes. In this work, we use the dataset from link[1].
- DDPM (Denoising Diffusion Probabilistic Models) [12] is one of the most classical diffusion generative models that creates images by gradually denoising random noise. We use the code of *DiffusionPipeline* from Diffusers[2]. We load the pre-trained model from the *celeba_hq_256*. We perform the image-to-image translation using the original

[1] https://github.com/peterwang512/CNNDetection.

[2] https://github.com/huggingface/diffusers.

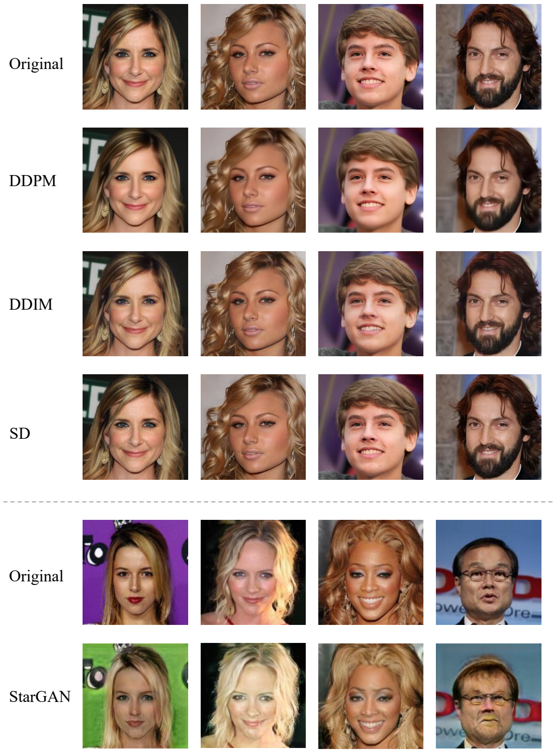

Figure 2. Examples of the entire synthesis images using the generative models. "SD" is the short for stable diffusion [18].

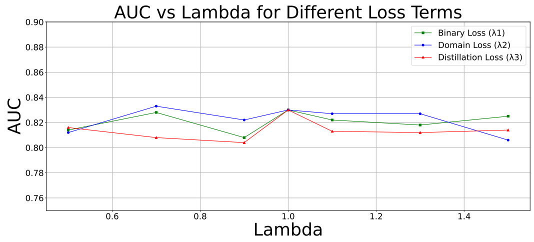

Figure 3. Exploration of loss terms: Binary loss (λ_1), Domain loss (λ_2), Distillation loss (λ_3).

images from CelebA [17].

- DDIM (Denoising Diffusion Implicit Models) [22] offers a non-Markovian variant of diffusion models with faster sampling times. We use the code of *DDIMScheduler* from Diffusers. We load the pre-trained model from the *celeba_hq_256*. We perform the image-to-image translation using the original images from CelebA [17].
- Stable Diffusion [18] is a recent state-of-the-art diffusion generative model that generates high-quality images based on textual descriptions. We use the code of *StableDiffusionPipeline* from Diffusers. We load the pre-trained model from the *celeba_hq_256*. We perform the image-to-image translation using the original images from CelebA [17].

Pre-processing Details In the pre-processing phase, the face extraction and alignment are conducted using DLIB [20], and the resulting aligned faces are resized to dimensions of 256 × 256 for both training and testing. All faces for training and testing are cropped with the margin rate fixed at 1.3 (consistent with DeepfakeBench [27]). Also, 32 frames are sampled with the same interval for each video for both the training and testing.

Training Details In the training phase, the Adam optimizer [13] is employed with a fixed learning rate of 0.0002. We collect and organize videos along with their corresponding fake counterparts into groups. Specifically, we utilize the FF++ dataset for training, which encompasses four distinct forgery methods for each real video. Consequently, for every real video, we have one real image paired with four fake images, resulting in a single group. In each training batch, we assemble four such groups. Our chosen batch size is set to 8, leading to a total of 40 images within one mini-batch. Furthermore, some widely used data augmentations, including image compression, horizontal flip, and down-sampling, are applied to the training data.

4. More Ablation Studies

In this section, we present supplementary ablation studies to provide a more comprehensive evaluation of our approach.

Further ablation for the within-domain augmentation In our manuscript, we conduct ablation studies to explore the effectiveness of within-domain augmentation (WD). There are three key transformations within the WD, *i.e.,* Centrifugal Transformation (CT), Affine Transformation (AfT), and Additive Transformation (AdT). Here, we further explore the effectiveness of these three main transformations toward the generalization performance of our model. To assess the individual contributions of these transformations, we conduct experiments where we remove each of them from the WD. Note that we do not employ the CD in these experiments. Results in Table. 1 show that each component of WD contributes positively to the final results. Removing each of them could produce a lower performance on average, which indicates all three transformations within the WD are important for general deepfake detection.

Exploration of the architecture for forgery encoders In our manuscript, we employ the EfficientNet-B4 as the backbone architecture for the forgery encoders. Here, we conduct additional investigations to explore other backbone architectures. We apply other three variants: Xception [4], EfficientNet-B1/B5 [23]. The results, presented in Table. 2, indicate that EfficientNet-B4 achieves the overall best results on average, making it the optimal choice for our

Method	Testing Datasets		
	CDF-v1	CDF-v2	DFDC
FWA [15]	0.790	0.668	0.613
Face X-ray [14]	0.709	0.679	0.633
OST [2]	-	0.748	-
SLADD [1]	-	0.797	-
SBI* [21]	**0.872**	0.827	0.720
Ours	0.867	**0.830**	**0.736**

Table 3. Comparison with other augmentation-based methods using the frame-level AUC metric. All methods are trained on FF++_c23. The results are generally cited from the benchmark [27]. * donates our reproduction with the official code to obtain the frame-level AUC results.

forgery encoders. Thus, we select EfficientNet-B4 as the default architecture for our forgery encoders.

Comparison with other augmentation-based methods. We conduct a comparative experiment with other approaches that are based on data augmentation. We utilize the frame-level AUC for comparison. Notably, SBI initially reported video-level results in the original paper. We reproduce the results of SBI [21] using the official code[3] to obtain the frame-level AUC results. The results for other detectors are directly sourced from DeepfakeBench. As indicated in Table. 3, we observe that our method consistently demonstrates superior generalization across these benchmarks, underscoring the effectiveness of our proposed method.

Model performance against hyper-parameters' variations. We adjust each hyper-parameter within the final objective function in a wide range. Fig. 3 shows there are slight performance fluctuations *w.r.t.* each parameter.

References

[1] 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, 4

[2] 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. 4

[3] 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. 2

[4] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 3

[5] Deepfakedetection, 2021. https://ai.googleblog.com/2019/09/contributing-data-to-deepfakedetection.html Accessed 2021-11-13. 1, 2

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

[7] 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. 1

[8] 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. 1, 2

[9] 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. 1, 2

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

[11] 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. 1

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

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[14] 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. 4

[15] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656*, 2018. 4

[16] 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. 1, 2

[17] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE/CVF Conference on International Conference on Computer Vision*, 2015. 3

[18] 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, 3

[3] https://github.com/mapooon/SelfBlendedImages.

[19] 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

[20] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: Database and results. *Journal of Image and Vision Computing*, 47:3–18, 2016. 3

[21] 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, 4

[22] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 3

[23] 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. 3

[24] 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

[25] 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

[26] 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

[27] 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. 1, 2, 3, 4