Exploiting Style Latent Flows for Generalizing Deepfake Video Detection

Supplementary Material

Jongwook Choi[1,2] Taehoon Kim[1] Yonghyun Jeong[3] Seungryul Baek[4] Jongwon Choi[1,2*]

[1]Dept. of Advanced Imaging, Chung-Ang Univ, Korea [2]GS. of AI, Chung-Ang Univ, Korea
[3]Image Vision, NAVER Cloud, Korea [4]AI Graduate School, UNIST, Korea

{cjw, kimth}@vilab.cau.ac.kr, yonghyun.jeong@navercorp.com, srbaek@unist.ac.kr, choijw@cau.ac.kr

The supplementary details provide specific explanations about the settings that were not fully described in the main paper, including various configuration values. Additionally, we present supplementary information that covers experimental results and visualizations which could not be included in the paper due to space limitations. As a result, we provide additional support to the validity of the results presented in our original paper.

1. More Implementation Details

We employed two preprocessing methods to detect and crop faces in the videos. First, we use RetinaFace [5] to detect and align the faces for each video. We use landmarks to determine the average face region and then use that region to crop the faces. Each clip consists of 32 cropped faces resized to 224× 224 and is used as input for the 3D CNN in Stage 2.

All experiments were conducted using four Nvidia A6000 48GB GPUs and an AMD Ryzen Threadripper PRO 3955WX 16-Cores CPU.

1.1. Stage 1

In Stage 1, we perform preprocessing to prepare the input for the pre-trained pSp encoder. We align and crop the faces using dlib[10]. These face images are resized to 256×256.

We use the total loss L in training StyleGRU, where λ sets to 1.

$$L = L_{tri} + \lambda L_{cls}, \tag{1}$$

1.2. Stage 2

We conduct data augmentation through the cutout. When applying the cutout, n square regions are randomly selected, ranging in size from 20% to 80% of the total image area. These cutout regions are applied uniformly to all frames within the clip. We employed the same 3D CNN architecture and Temporal Transformer Encoder structure as FTCN [21]. For cross-dataset experiments, we utilized the pretrain weights provided by the existing FTCN model to facilitate effective learning. During this process, we employed the SGD optimizer with momentum and trained with a learning rate of 5e-7.

2. Additional Experiments

2.1. Style Latent modeling

Latent	CDF	FSh
coarse	88.1	98.5
middle	87.8	98.5
fine	88.1	98.7
total (Ours)	89.0	99.0

Table 1. **Style Latent ablation study.**

According to [4], training with specific latents is effective for deepfake detection. To test this argument for our model, we divide the extracted **total** style latent (18×512) into **coarse** (3×512), **middle** (4×512), and **fine** (11×512) segments, following the suggestion of StyleGAN[9], and conduct separate experiments on each segment. To evaluate our model's generalization performance, we train it on the FF++ [15] dataset and then conduct performance evaluations on the CDF [12] and FSh [11] datasets.

Our model's ablation study performance undergoes assessment through the presentation of AUC scores (%) on the coarse, middle, fine, and total style latent vectors. In Table 1, we can observe that utilizing all latent variables ultimately demonstrates higher generalization performance.

Metric	CDF	Fsh
No differencing	88.4	98.4
2nd-order differencing	88.8	98.8
Ours(1st-order differencing)	**89.0**	**99.0**

Table 2. **Flow modeling metric comparison experiment.**

As presented in Table 2, it reveals that the StyleGRU achieves superior performance when utilizing a metric based on first-order differencing, compared to other approaches such as neglecting differencing or employing second-order differencing. Nonetheless, the small performance gaps suggest that the GRU layer lets the style feature independent of the metric.

Condition	CDF	FSh
Self-supervised	88.5	98.4
Supervised (Ours)	89.0	99.0

Table 3. **Contrastive learning ablation study.**

[*] corresponding author

Method	Clean	Saturation	Contrast	Block	Noise	Blur	Pixel	Compress	Avg
Xception [2]	99.8	99.3	98.6	99.7	53.8	60.2	74.2	62.1	78.3
CNN-agu [19]	99.8	99.3	99.1	95.2	54.7	76.5	91.2	72.5	84.1
Patch-based [1]	99.9	84.3	74.2	99.2	50.0	54.4	56.7	53.4	67.5
CNN-GRU [16]	99.9	99.0	98.8	97.9	47.9	71.5	86.5	74.5	82.3
FTCN* [21]	99.5	98.0	93.7	90.1	53.8	95.0	94.8	83.7	87.0
AltFreezing* [20]	99.8	99.4	98.9	91.8	60.9	98.3	97.8	89.9	91.0
Ours	99.6	99.2	95.8	92.2	55.0	97.3	97.3	86.3	90.4

Table 4. **Robustness to Perturbations.** We evaluate the average performance change based on the video-level AUC scores when applying distortions at five different degradation levels. The asterisk(*) denotes that we have reproduced the results using officially provided weights. The perturbation follows the approach provided by DeeperForensics [8].

2.2. Contrastive learning comparison

Training the StyleGRU in Stage 1 applies a supervised contrastive learning manner using anchor, positive and negative samples with given labels. On the other hand, for representation learning, self-supervised contrasitive learning approaches without providing labels are often employed. We compared the performance difference between our supervised contrastive learning and a self-supervised contrastive learning approach in which clips extracted from the same video as the anchor clip were used as positive clips, and clips from different videos were used as negative clips, without using label. We conduct an ablation study on the representation learning methodology. We train on the FF++ dataset and verify performance through video-level AUC scores on the CDF and FSh datasets.

Table 3 illustrates that when employing supervised contrastive learning, it exhibits superior generalization performance compared to models utilizing self-supervised contrastive learning, thereby showcasing the efficacy of our approach.

2.3. More about Perturbation Robustness

We conduct a comparison study of the robustness of our proposed model to perturbations, which was not adequately presented in the main paper due to space constraints. As Table 4 illustrates, it demonstrates suboptimal performance for a specific perturbation. Specifically, performance is vulnerable to ‘ Noise ’ type perturbations, which is attributed to the relatively sensitive response of the pSp encoder [14] to noise. However, it exhibits a high level of performance compared to the majority of methods, and there is potential for improvement through future model enhancements.

Method	raw	c23	c40
FTCN	99.5	99.0	70.2
Ours	**99.6**	**99.1**	**71.8**

Table 5. **Experiment on low-quality images.**

As shown in Table 5, we performed an experiment to verify if our algorithm generalizes well to low-quality samples even when trained on high-quality ones. In the results, it can be observed that our method exhibits relatively robust performance against low-quality samples.

Method	CDF	DFD	KoDF	Avg
FTCN	86.9	94.4	69.4	83.6
Altfreezing	**89.0**	93.7	69.8	84.2
Ours	**89.0**	**96.1**	**71.1**	**85.4**

Table 6. **Generalization to distinct datasets from training data.**

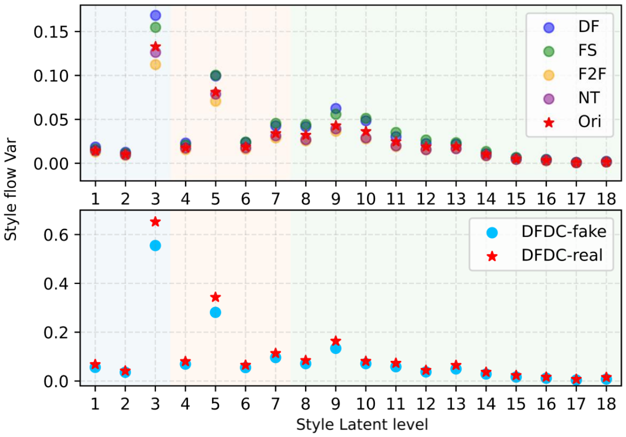

Figure 1. **Visualization about Style Latent variance.** The x-axis represents the levels of style latent vectors for detailed style representations.The plot above visualizes the results for the FF++ dataset, which we primarily used as the Train dataset, and the plot below presents the visualization results for the DFDC dataset.

3. Visualization

3.1. More about style latent variance

We provide additional explanations regarding the motivation behind the variance in style latent and conduct experiments on different datasets in our paper. We utilized the style latent vectors with differencing from the first clip of each video in the dataset. To examine temporal changes, we

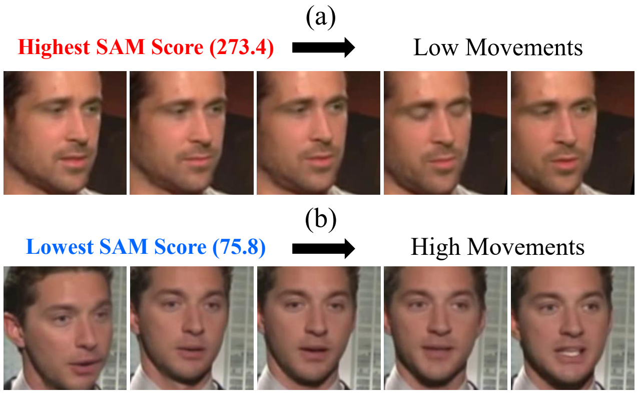

Figure 2. **Qualitative comparison with SAM score.** The displayed frames are extracted from correctly classified synthesis videos in the CDF dataset, with each frame taken at intervals of 6 frames within a single clip.

computed the variance and visualized it as a means of statistical analysis.

According to the results in Figure 1, a noticeable distinction between real and fake videos is evident in both the FF++ [15] and DFDC[6] datasets. The DF [3], FS [13], and DFDC datasets, which involve the identity swap method, exhibit differences in a manner similar to what was observed in the visualizations of DFD [7] and CDF [12]. On the other hand, the F2F [17] and NT [18] datasets, which involve the expression swap method, show reduced variance due to the fixed identity throughout the manipulation.

3.2. Qualitative comparison with SAM score

In Figure 2, a qualitative evaluation is carried out for the SAM scores. Figure 2(a) illustrates frames extracted from fake clips with notable SAM scores. When considering classification based on low-level temporal cues, it becomes challenging to classify clips demonstrating slow movement. The SAM proposed in our work generates high responses for clips containing slow movement, actively utilizing high-level temporal cues. On the other hand, as shown in Figure 2(b), for clips with significant motion, it is observed that low responses are generated to focus on low-level temporal cues. This result suggests that the style latent vector flow proposed in this study can be used as a complementary feature to the conventional image-based temporal artifact.

References

[1] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, pages 103–120. Springer, 2020. 2

[2] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 2

[3] Deepfakes. faceswap. `https://github.com/deepfakes/faceswap`, 2019. [Accessed 27-07-2023]. 3

[4] Matthieu Delmas, Amine Kacete, Stephane Paquelet, Simon Leglaive, and Renaud Seguier. Latentforensics: Towards lighter deepfake detection in the stylegan latent space. *arXiv preprint arXiv:2303.17222*, 2023. 1

[5] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5203–5212, 2020. 1

[6] 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. 3

[7] Nick Dufour and Andrew Gully. Contributing Data to Deepfake Detection Research — ai.googleblog.com. `https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html`, 2019. [Accessed 30-07-2023]. 3

[8] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2889–2898, 2020. 2

[9] 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. 1

[10] Davis E King. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10:1755–1758, 2009. 1

[11] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Advancing high fidelity identity swapping for forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5074–5083, 2020. 1

[12] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3207–3216, 2020. 1, 3

[13] MarekKowalski. Faceswap. `https://github.com/MarekKowalski/FaceSwap`, 2019. [Accessed 30-07-2023]. 3

[14] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021. 2

[15] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In

Proceedings of the IEEE/CVF international conference on computer vision, pages 1–11, 2019. 1, 3

[16] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. *Interfaces (GUI)*, 3(1):80–87, 2019. 2

[17] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016. 3

[18] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *Acm Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 3

[19] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8695–8704, 2020. 2

[20] 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

[21] 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 international conference on computer vision*, pages 15044–15054, 2021. 1, 2