Vulnerability-Aware Spatio-Temporal Learning for Generalizable Deepfake Video Detection

Supplementary Material

Function	Test set AUC (%)						
	CDF	DFD	DFDCP	DFDC	DFW	DiffSwap	Avg.
MinMax	91.6	95.9	88.1	71.1	70.1	92.1	84.8
Unnormalized 3D Gaussian	**92.5**	91.8	86.4	72.4	**76.3**	91.7	85.2
MeanStd	92.4	**98.5**	**90.0**	**74.6**	74.2	**96.9**	**87.8**

Table 1. **Comparison of different normalization functions**. We consider three normalization functions, i.e., Standardization (MeanStd), MinMax, and Unnormalized 3D Gaussian. Among these, Standardization gives the best overall performance.

1. Appendix

1.1. Details of the Ground-Truth Derivative

Calculation formula. To generate the ground truth data for the temporal branch h, we compute the derivative on $\tilde{\mathbf{B}} = (\tilde{\mathbf{B}}(t))_{t\in[[1,T]]}$ with respect to the temporal dimension. Specifically, we calculate the absolute value of the difference between two consecutive patch-level vulnerability values $\tilde{\mathbf{B}}(t)$ and $\tilde{\mathbf{B}}(t-1)$ with $t \geq 2$ such that,

$$\mathbf{D}(t) = \left|\tilde{\mathbf{B}}(t) - \tilde{\mathbf{B}}(t-1)\right|. \tag{1}$$

The process is iterated for every pair of consecutive frames of $(\tilde{\mathbf{B}}(t))_{t\in[[1,T]]}$ to obtain a derivative matrix $\mathbf{D} \in \mathbb{R}^{T\times\sqrt{N}\times\sqrt{N}}$. For $t = 1$, we insert a matrix $\mathbf{0}$, i.e., $\mathbf{D}(1) = \mathbf{0}$, indicating no temporal change at the first frame.
Normalization functions. Employing a normalization function is important for stabilizing the training of our temporal branch h. Therefore, we consider three different normalization functions including Standardization (MeanStd), MinMax, and Unnormalized 3D Gaussian. Specifically, for the Standardization and MinMax, we respectively compute the std($\mathbf{D}$)-mean($\mathbf{D}$), and min($\mathbf{D}$)-max($\mathbf{D}$), while we follow the work of [17] to adapt an unnormalized Gaussian map from 2D to 3D for normalization. We report in Table. 1 cross-evaluation results on six datasets [6–8, 15, 31, 33] with the use of the three investigated functions, using a model trained on FF++ [19]. It can be noted that the model is robust to various types of normalization functions with the best performance recorded for the Standardization approach.

1.2. SBV: Pseudo-code and Visual Samples

Algorithm. To enhance the clarity and reproducibility of the SBV generation process, we provide the overall algorithm in the form of pseudo-code, as detailed in Algorithm 1.

Algorithm 1: Pseudo-code for SBV Generation

Input: Real video $\mathbf{X}_i^r \in \mathcal{V}^r$ of size (C, T, H, W), facial landmarks $\mathbf{L}_i \triangleq \cup_{t=1}^{T}\{\mathbf{l}_{ij}(t)\}_{1\leq j\leq n}$ of size $(T, n, 2)$, a distance $\bar{d}$, a threshold τ
Output: Self-blended video $\mathbf{X}_i^{\tilde{r}} \in \mathcal{V}^{\tilde{r}}$ of size (C, T, H, W), blending mask $\mathbf{M}_i$ of size (T, H, W)

Initialize $\theta^{(sbi)}$ as an empty dictionary
Initialize $\mathbf{X}_i^{\tilde{r}}$ as an empty array
Initialize $\mathbf{M}_i$ as an empty array
for $j = 1$ **to** T **do**
  if $j = 1$ **then**
    $\mathbf{X}_i^{\tilde{r}}(t_0), \mathbf{M}_i(t_0), \{\theta^{(c)}, \theta^{(m)}, \theta^{(b)}, \dots\} \leftarrow$ SBI($\mathbf{X}_i^r(t_0), \mathbf{l}_i(t_0)$)
    $\mathbf{X}_i^{\tilde{r}} \leftarrow \mathbf{X}_i^{\tilde{r}} \cup \{\mathbf{X}_i^{\tilde{r}}(t_0)\}$
    $\mathbf{M}_i \leftarrow \mathbf{M}_i \cup \{\mathbf{M}_i(t_0)\}$
    $\theta^{(sbi)} \leftarrow \{\theta^{(c)}, \theta^{(m)}, \theta^{(b)}, \dots\}$
  end
  else
    $\mathbf{l}_i(t) \leftarrow$ LandmarkInterpolation($\mathbf{l}_i(t)$, $\mathbf{l}_i(t-1)$, $\bar{d}$, τ)
    $\mathbf{X}_i^{\tilde{r}}(t), \mathbf{M}_i(t), \_ \leftarrow$ SBI($\mathbf{X}_i^r(t), \mathbf{l}_i(t), \theta^{(sbi)}$)
    $\mathbf{X}_i^{\tilde{r}} \leftarrow \mathbf{X}_i^{\tilde{r}} \cup \{\mathbf{X}_i^{\tilde{r}}(t)\}$
    $\mathbf{M}_i \leftarrow \mathbf{M}_i \cup \{\mathbf{M}_i(t)\}$
  end
end
return $\mathbf{X}_i^{\tilde{r}}, \mathbf{M}_i$

Visual Samples. To visually demonstrate the benefits of the Consistent Synthesized Parameters (CSP) and the Landmark Interpolation (LI) module (Section 3.1 in the main paper) in generating high-quality pseudo-fake videos, we show some SBV samples, their blending boundaries, original landmarks, and those modified by the proposed modules in Figure 1. In the top part of the figure, we compare data generated using only CSP to data generated with both CSP and the Landmark Interpolation module. We observe that the Landmark Interpolation module ensures smooth transitions of facial landmarks between consecutive frames ($t \rightarrow t+1$). In the bottom part of the figure, we compare data generated with only CSP to data generated without any of the proposed SBV components. We observe significant variations in the manipulated facial areas when CSP is omitted. Therefore, the proposed CSP and Landmark Interpolation module effectively enhances the temporal coherence of

Figure 1. **Illustration of the facial landmarks, the generated SBV, and the blending boundaries** with and without applying the Consistent Synthesized Parameters (CSP) module and the Landmark Interpolation (LI) module. The lack of applied CSP and LI indicates simply stacked SBIs (BottomRight).

the generated SBV.

1.3. Impact of SBV

To verify the advantage of using SBV for improving the generalization of detectors, we conduct several experiments using different binary classifiers trained either with SBV or with one of the four types of forgeries forming FF++ [19] (DF [4], F2F [22], FS [11], NT [21]). For a fair comparison, a widely-used CNN-based Resnet3D [9] and a Transformer-based TimeSformer [1] are employed. We note that both selected models are trained from *Scratch (S)* without pretrained initialization. Table. 2 presents the generalization performance in terms of AUC (%) on five datasets [6–8, 15, 33] respectively when trained with different manipulation methods from FF+. Notably, training with SBV significantly increases the overall generalizability capability of binary models as compared to those trained on using one specific manipulation. This indicates the importance of highly realistic, naturally consistent generated pseudo-fake videos.

1.4. Robustness to Unseen Perturbations

In the main manuscript, we report in Figure. 3 the "Average" performance under different corruptions. This section complements this experiment by reporting the mean performance across different severity levels for each degradation type, as detailed in Table. 3. Except for a slight decrease in effectiveness under "Change Saturation" compared to LAA-Net [18], FakeSTormer is generally more robust to the unseen perturbations as compared to other augmented-based methods [12, 14, 18, 20].

1.5. Multi-shot Inferences

Models can sometimes be overconfident in their predictions, which negatively impacts the generalizability aspect [16, 29]. To address this issue, we explore the possi-

Method	Pretrain	Training set					Test set AUC (%)						Avg.
		Real	DF	FS	F2F	NT	FF++	CDF	DFD	DFDCP	DFDC	DFW	
ResNet3D [9]	×	✓	✓	×	×	×	72.5	58.5	51.3	53.4	59.4	65.0	60.0
TimeSFormer [1]	×	✓	✓	×	×	×	65.4	59.3	66.1	53.5	61.4	57.5	60.5
ResNet3D [9]	×	✓	×	✓	×	×	70.6	61.1	50.6	59.2	55.8	51.5	58.1
TimeSFormer [1]	×	✓	×	✓	×	×	76.4	51.7	43.7	44.6	54.5	43.9	52.5
ResNet3D [9]	×	✓	×	×	✓	×	78.0	63.8	54.5	63.4	55.7	50.1	60.9
TimeSFormer [1]	×	✓	×	×	✓	×	81.1	64.4	60.1	64.5	52.0	50.5	62.1
ResNet3D [9]	×	✓	×	×	×	✓	72.7	63.7	75.6	69.1	59.6	62.6	67.2
TimeSFormer [1]	×	✓	×	×	×	✓	75.5	65.8	84.7	70.3	62.7	65.5	70.8
ResNet3D [9] + SBV	×	✓	×	×	×	×	<u>90.2</u>	<u>85.9</u>	<u>85.0</u>	<u>82.8</u>	<u>66.4</u>	<u>67.5</u>	<u>79.6</u>(↑12.4)
TimeSFormer [1] + SBV	×	✓	×	×	×	×	**94.7**	**89.5**	**95.6**	**88.6**	**72.5**	**70.9**	**85.3**(↑14.5)

Table 2. **Cross-dataset generalization**. Performance comparison in terms of AUC (%) on multiple datasets of different binary classification models [1, 9] trained using our video synthesis (SBV) and normal fake data [19]. All models are trained on FF++(c23) [19] from ***Scratch (S)*** and are tested on other datasets [6–8, 15, 33]. Gray indicates the use of normal fake data for training. **Bold** and <u>underline</u> highlight the best and the second-best performance, respectively.

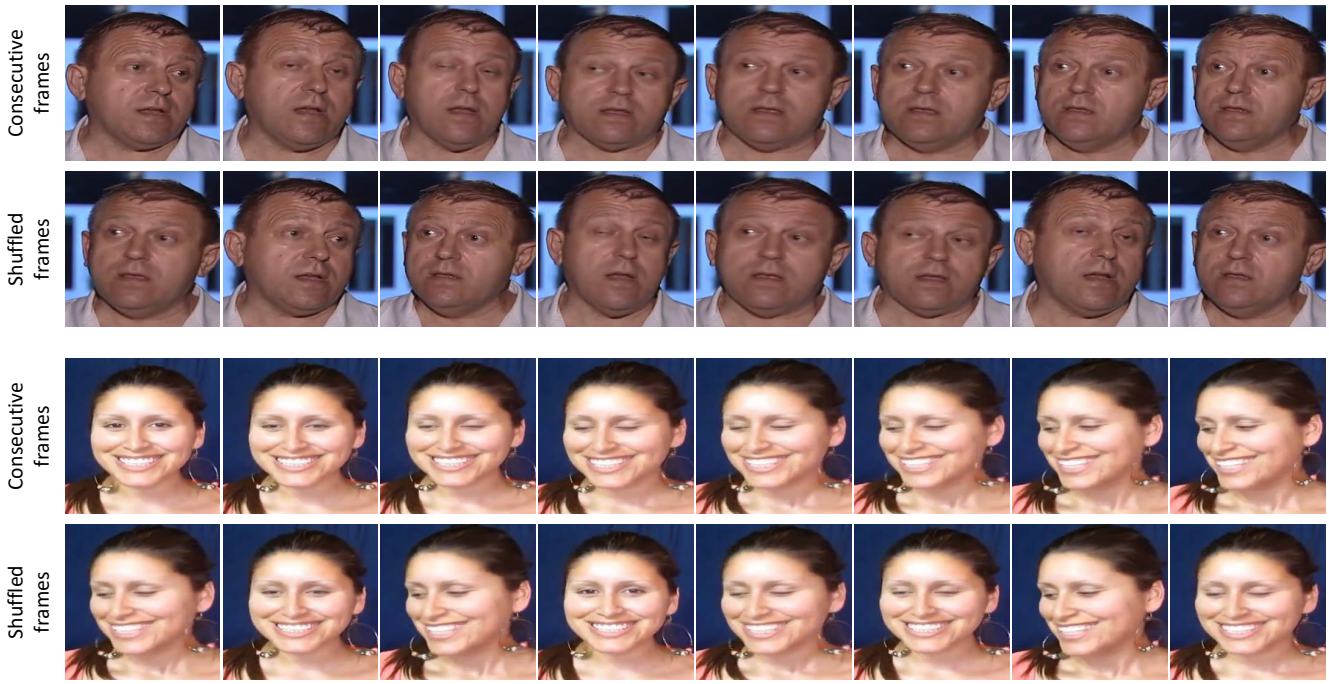

Figure 2. **Shuffled frames can produce obvious temporal inconsistencies.**

bility of regularizing the input during testing. Specifically, we propose multi-shot inference, leveraging Vulnerability-Driven Cutout Augmentation by utilizing the temporal head output $\tilde{\mathbf{D}}$. We use $\tilde{\mathbf{D}}$ because the most significant temporal changes in vulnerable areas over time, from $t \rightarrow (t+1)$, are likely to occur at the spatial locations corresponding to the highest values of $\bar{\mathbf{B}}$ (Eq.3 in the main manuscript).

In particular, given a test video, after the first shot of inference, the prediction map $\tilde{\mathbf{D}}$ can be leveraged to generate a new masked video through the proposed Cutout augmentation for the second inference shot. Specifically, we select $\tilde{\mathbf{D}}$ at $t = 2$ (capturing the temporal transition from the first $\rightarrow$ the second frame) to define the set $\mathcal{P}$ (Section. 3.1 of the main manuscript) for determining Cutout positions. This iterative process can be repeated for multiple inference shots.

Table. 4 presents the cross-dataset evaluation results with five shots of inference on five unseen datasets [6–8, 15, 33] using the model trained on FF++ [19]. The results indicate a gradual improvement in generalization performance after each iteration. This suggests that the prediction outputs are not only interpretable but also can be used potentially to enhance the model performance.

Method	Real	Fake	Contrast	Saturation	Gaussian Blur	Gaussian Noise	JPEG Compression	Block Wise	Avg
DSP-FWA [14]	✓	✓	80.7	79.6	67.3	61.8	68.0	76.6	72.3
FaceXray [12]	✓	✓	88.9	96.0	70.0	58.0	62.2	94.7	78.3
SBI [20]	✓	×	92.3	92.0	72.7	62.2	79.1	92.2	81.7
LAA-Net [18]	✓	×	95.0	**97.0**	73.2	57.5	75.2	94.9	82.1
Ours	✓	×	**97.6**	96.3	**81.6**	**65.4**	**92.9**	**96.8**	**88.4**

Table 3. **Robustness to unseen perturbations**. Average AUC scores (%) across all levels for each degradation type.

No. shots	Test set AUC (%)					Avg.
	CDF	DFD	DFDCP	DFDC	DFW	
1	92.35	98.47	90.02	74.56	74.19	85.92
2	92.34	98.51	90.11	74.60	74.25	85.96(↑0.04)
3	**92.38**	**98.52**	**90.13**	**74.61**	74.28	85.98(↑0.06)
4	**92.38**	**98.52**	**90.13**	**74.61**	**74.30**	**85.99**(↑0.07)
5	92.36	**98.52**	**90.13**	**74.61**	**74.30**	85.98(↑0.06)

Table 4. **Multi-shot inferences.** AUC (%) comparison of our model using different numbers of inference shots in the cross-dataset setup. The AUC slightly increases with a higher number of shots.

1.6. Visualization of Auxiliary Branches' Outputs

In addition to the probability output of the standard classification branch, FakeSTormer can provide more valuable insights from our auxiliary branches that might be conductive to prediction's post-analyses. Specifically, the spatial and temporal branches output the intensity of the spatial artifacts encoded in each video frame and the vulnerability change over time, respectively. The spatial branch provides frame-level scores, while the temporal branch offers more fine-grained insights. As shown in Figure 3, the spatial outputs (denoted by the numbers in each frame) denote high values for fake data and low values for real data. For the temporal outputs, the heatmaps show the change in vulnerability between the instant frame t and the previous one. It can be observed that it primarily focuses around the blending boundaries. We note that the change between $t-1$ and t is visualized at the t^{th} frame; hence, there is a *blank* heatmap at the 1^{st} frame.

1.7. STC [13]: Shuffled Frames can produce obvious Temporal Inconsistencies

We propose SBV to generate subtler artifacts for pseudo-fakes compared to the STC approach used in [13]. We believe that STC may produce obvious (low-quality) temporal artifacts, as it shuffles frames in the temporal domain, leading to significant inconsistencies. Figure 2 illustrates how shuffling creates noticeable discrepancies between frames. In contrast, our SBV leverages consecutive frames to produce subtler temporal artifacts while simulating these artifacts in a different manner (as detailed in Section 3.1 of the main manuscript).

1.8. Details on the Datasets

Datasets. For our experiments, we select datasets that haven typically used as benchmarks in previous works [2, 3, 23–26, 30, 32]. For both training and validation, we employ **FaceForensics++**(FF++) [19], which consists of 1,000 real videos and 4,000 fake videos generated using four manipulation methods: (Deepfakes (DF) [4], FaceSwap (FS) [11], Face2Face (F2F) [22], and NeuralTextures (NT) [21]). It can be noted that, for training, we use only the real videos and generate pseudo-fake data using our synthesized method, SBV. By default, the c23 version of FF++ is adopted, following the recent literature [3, 24, 25, 30, 32].

For further validation, we also evaluate on the following datasets: (1) **Celeb-DFv2** (CDF) [15], a well-known benchmark with high-quality deepfakes; (2) **DeepfakeDetection** (DFD) [8], which includes 3,000 forged videos featuring 28 actors in various scenes; (3) **Deepfake Detection Challenge Preview** (DFDCP) [6] and (4) **Deepfake Detection Challenge** (DFDC) [7], a large-scale dataset containing numerous distorted videos with issues such as compression and noise; (5) **WildDeepfake** (DFW) [33], a dataset fully sourced from the internet, without prior knowledge of manipulation methods; (6) **DiffSwap** generated in the similar protocol as in LFGDIN [28] by using a recent diffusion-based swapping method [31] on 250 real videos selected from CDF [15]; and (7) **DF40** [27], a highly diverse and large-scale dataset comprising 40 distinct deepfake techniques, enables more comprehensive evaluations for the next generation of deepfake detection.

Data Pre-processing. Following the splitting convention [19], we extract 256, 32, and 32 consecutive frames for training, validation, and testing, respectively. Facial regions are cropped using Face-RetinaNet [5]. These bounding boxes are conservatively enlarged by a factor of 1.25 around the center of the face and then resized to a fixed resolution of 224×224. Additionally, we store 81 facial landmarks for each frame, extracted using Dlib [10]. Finally, the preserved landmark keypoints are utilized to dynamically generate pseudo-fakes during each training iteration.

1.9. Revisited TimeSformer: Implementation Details

We choose TimeSformer [1] as our feature extractor given its ability to effectively capture separate long-range

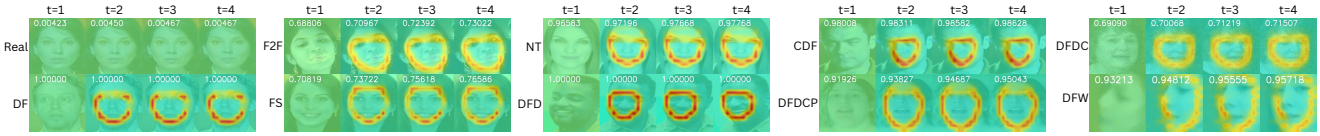

Figure 3. **Visualization of Auxiliary Branches' Outputs.** We visualize the additional auxiliary spatial and temporal branches' outputs on different unseen datasets. As shown, the number on each frame denotes the output of the spatial branch g, while the heatmap visualizes the output of the temporal branch h.

temporal information and spatial features. First, given a video $\mathbf{X} \in \mathbb{R}^{C\times T\times H\times W}$, its frames in each time step are split into N number of non-overlapping patches of size $P \times P$, i.e., $N = \frac{H\times W}{P^2}$. Each patch is flatten as $\mathbf{x}_{(t,p)} \in \mathbb{R}^{C.P^2}$, and is then linearly mapped into D-dimensional embedding vector $\mathbf{z}^0_{(t,p)} \in \mathbb{R}^D$ by means of a learnable matrix $E \in \mathbb{R}^{D\times C.P^2}$ where $t = [[1, T]]$ indexes temporal positions, and $p = [[1, N]]$ indexes spatial positions. The process results in an input patch embedding matrix $\mathbf{Z}^0 \in \mathbb{R}^{T\times N\times D}$.

In TimeSformer, a global class token $\mathbf{z}_{cls}$ attends to all patches and then is used for classification. This mechanism implicitly captures mixed spatial-temporal features at the same time, which might lead to overfitting on a specific type of domain artifacts [24, 32]. We revisit it slightly in order to decouple the spatial and temporal information by considering two sorts of additional tokens (one spatial and one temporal).

For that purpose, we attach in each dimension of $\mathbf{Z}^0$, a spatial token $\mathbf{z}^0_s \in \mathbb{R}^D$ and a temporal token $\mathbf{z}^0_t \in \mathbb{R}^D$, respectively. These tokens will independently interact only with patch embeddings belonging to their dimension axis by leveraging the decomposed SA [1]. This mechanism not only facilitates the disentanglement learning process of spatio-temporal features but is also beneficial to optimize the computational complexity of $\mathcal{O}(T^2 + N^2)$ as compared to $\mathcal{O}(T^2 \cdot N^2)$ in vanilla SA. Those tokens will be then fed into L ($L = 12$ as default) transformer encoder blocks in which each block contains a multi-head temporal SA (TSA), a multi-head spatial SA (SSA), LayerNorm (LN), and a multi-layer perception (MLP). Note that, for the sake of matrix compatibility, a placeholder embedding $\mathbf{z}^0_{(0,0)}$ is attached. Formally, the feature extraction process can be summarized as follows,

$$[\mathbf{Z}^L, \mathbf{z}^L_s, \mathbf{z}^L_t] = \Phi(\mathbf{X}), \tag{2}$$

where $\mathbf{Z}^L$ is the final patch embedding matrix, $\mathbf{z}^L_s$ the resulting set of spatial tokens, and $\mathbf{z}^L_t$ the resulting set of temporal tokens that will be respectively sent to the temporal head h, the spatial head g, and the classification head f. Our overall framework is illustrated in Figure. 2-I of the main paper.

References

[1] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 2, 3, 4, 5

[2] Junyi Cao, Chao Ma, Taiping Yao, Shen Chen, Shouhong Ding, and Xiaokang Yang. End-to-end reconstruction-classification learning for face forgery detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4103–4112, 2022. 4

[3] Jongwook Choi, Taehoon Kim, Yonghyun Jeong, Seungryul Baek, and Jongwon Choi. Exploiting style latent flows for generalizing deepfake video detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1133–1143, 2024. 4

[4] Deepfakes. Faceswapdevs. https://github.com/deepfakes/faceswap, 2019. 2, 4

[5] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *CoRR*, abs/1905.00641, 2019. 4

[6] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton-Ferrer. The deepfake detection challenge (DFDC) preview dataset. *CoRR*, abs/1910.08854, 2019. 1, 2, 3, 4

[7] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton-Ferrer. The deepfake detection challenge dataset. *CoRR*, abs/2006.07397, 2020. 4

[8] Nick Dufour and Andrew Gully. Contributing data to deepfake detection research. https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html, 2019. 1, 2, 3, 4

[9] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6546–6555, 2018. 2, 3

[10] Davis E. King. Dlib-ml: A machine learning toolkit. *J. Mach. Learn. Res.*, 10:1755–1758, 2009. 4

[11] Marek Kowalski. Faceswap. https://github.com/MarekKowalski/FaceSwap, 2018. 2, 4

[12] 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 *IEEE/CVF Conference on*

Computer Vision and Pattern Recognition (CVPR), 2020. 2, 4

[13] Maosen Li, Xurong Li, Kun Yu, Cheng Deng, Heng Huang, Feng Mao, Hui Xue, and Minghao Li. Spatio-temporal catcher: A self-supervised transformer for deepfake video detection. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8707–8718, 2023. 4

[14] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. 2, 4

[15] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3, 4

[16] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? *CoRR*, abs/1906.02629, 2019. 2

[17] Dat Nguyen, Marcella Astrid, Enjie Ghorbel, and Djamila Aouada. Fakeformer: Efficient vulnerability-driven transformers for generalisable deepfake detection. *arXiv preprint arXiv:2410.21964*, 2024. 1

[18] Dat Nguyen, Nesryne Mejri, Inder Pal Singh, Polina Kuleshova, Marcella Astrid, Anis Kacem, Enjie Ghorbel, and Djamila Aouada. Laa-net: Localized artifact attention network for quality-agnostic and generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17395–17405, 2024. 2, 4

[19] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 3, 4

[20] 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

[21] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *CoRR*, abs/1904.12356, 2019. 2, 4

[22] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of RGB videos. *CoRR*, abs/2007.14808, 2020. 2, 4

[23] 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 (CVPR)*, pages 7278–7287, 2023. 4

[24] 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 (CVPR)*, pages 4129–4138, 2023. 4, 5

[25] Yuting Xu, Jian Liang, Gengyun Jia, Ziming Yang, Yanhao Zhang, and Ran He. Tall: Thumbnail layout for deepfake video detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22658–22668, 2023. 4

[26] Zhiyuan Yan, Yong Zhang, Yanbo Fan, and Baoyuan Wu. Ucf: Uncovering common features for generalizable deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22412–22423, 2023. 4

[27] Zhiyuan Yan, Taiping Yao, Shen Chen, Yandan Zhao, Xinghe Fu, Junwei Zhu, Donghao Luo, Chengjie Wang, Shouhong Ding, Yunsheng Wu, and Li Yuan. Df40: Toward next-generation deepfake detection. In *Advances in Neural Information Processing Systems*, pages 29387–29434. Curran Associates, Inc., 2024. 4

[28] Pengfei Yue, Beijing Chen, and Zhangjie Fu. Local region frequency guided dynamic inconsistency network for deepfake video detection. *Big Data Mining and Analytics*, 7(3): 889–904, 2024. 4

[29] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *CoRR*, abs/1710.09412, 2017. 2

[30] Cairong Zhao, Chutian Wang, Guosheng Hu, Haonan Chen, Chun Liu, and Jinhui Tang. Istvt: Interpretable spatial-temporal video transformer for deepfake detection. *IEEE Transactions on Information Forensics and Security*, 18: 1335–1348, 2023. 4

[31] Wenliang Zhao, Yongming Rao, Weikang Shi, Zuyan Liu, Jie Zhou, and Jiwen Lu. Diffswap: High-fidelity and controllable face swapping via 3d-aware masked diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8568–8577, 2023. 1, 4

[32] 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 (ICCV)*, pages 15044–15054, 2021. 4, 5

[33] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. Wilddeepfake: A challenging real-world dataset for deepfake detection. *Proceedings of the 28th ACM International Conference on Multimedia*, 2020. 1, 2, 3, 4