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

Dat NGUYEN[◁], Marcella ASTRID[◁], Anis KACEM[◁], Enjie GHORBEL[◁,⋊], Djamila AOUADA[◁]
CVI[2], SnT, University of Luxembourg[◁]
Cristal Laboratory, National School of Computer Sciences, University of Manouba[⋊]
{dat.nguyen,marcella.astrid,anis.kacem,djamila.aouada}@uni.lu enjie.ghorbel@isamm.uma.tn

Abstract

Detecting deepfake videos is highly challenging given the complexity of characterizing spatio-temporal artifacts. Most existing methods rely on binary classifiers trained using real and fake image sequences, therefore hindering their generalization capabilities to unseen generation methods. Moreover, with the constant progress in generative Artificial Intelligence (AI), deepfake artifacts are becoming imperceptible at both the spatial and the temporal levels, making them extremely difficult to capture. To address these issues, we propose a fine-grained deepfake video detection approach called FakeSTormer that enforces the modeling of subtle spatio-temporal inconsistencies while avoiding overfitting. Specifically, we introduce a multi-task learning framework that incorporates two auxiliary branches for explicitly attending artifact-prone spatial and temporal regions. Additionally, we propose a video-level data synthesis strategy that generates pseudo-fake videos with subtle spatio-temporal artifacts, providing high-quality samples and hand-free annotations for our additional branches. Extensive experiments on several challenging benchmarks demonstrate the superiority of our approach compared to recent state-of-the-art methods. The code is available at https://github.com/10Ring/FakeSTormer.

1. Introduction

With the advances in generative modeling [22, 43], deepfake videos have become alarmingly realistic. Despite their interest in several applications, such as entertainment and education, this type of technology also raises societal concerns [5, 45, 53]. There is therefore an urgency for developing effective deepfake detection methods.

In the literature, several deepfake detection techniques aim to model spatial artifacts by treating each frame independently [3, 6, 7, 15, 33, 39, 41, 42, 46, 48, 69]. While this is reasonable when dealing with frame-level generation methods [32, 51, 73], it becomes less adequate in the presence of video-level manipulation techniques [1, 19, 58], where temporal and spatial artifacts are *intertwined*.

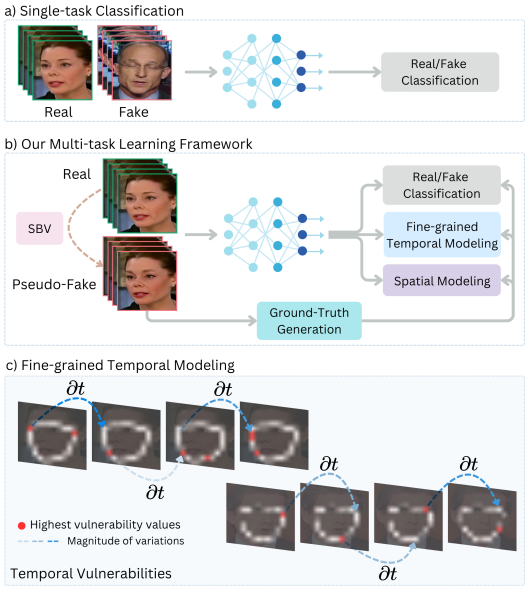

Figure 1. a) Traditional video-based methods [20, 23, 25, 56, 57, 64, 67, 71] versus b) the proposed multi-task learning framework; c) Visualization of the temporal vulnerabilities. Note that only some temporal locations are shown.

For that reason, researchers have explored video-based deepfake detection methods capable of modeling spatio-temporal artifacts [20, 23, 25, 56, 64, 67, 71]. Those methods mainly rely on a deep neural network formed by a single binary classification branch that is trained using a fixed dataset with real and fake data (see Figure 1-a). As a result, they suffer from two main limitations, namely: (1) **The lack of generalizability -** As highlighted in [33, 39, 42, 48], models trained with a standalone binary classifier tend to overfit the type of deepfakes they are trained on, resulting in poor generalization to unseen manipulations; (2) **The lack of robustness to high-quality (HQ) deepfake videos -** The

quality of deepfake videos is improving continuously, resulting in subtle spatio-temporal artifacts. As such, vanilla single-branch architectures trained solely with binary supervision fail to fully capture them, necessitating the design of appropriate attention mechanisms.

To address the generalization issue, video-level data synthesis approaches [24, 34, 62] have been introduced to encourage models to learn more generic representations. However, these methods usually simulate exaggerated temporal variations that are inherently different from artifacts in hyper-realistic deepfake videos. On the other hand, to model localized spatio-temporal inconsistencies, some recent methods [63, 67] have used dedicated architectures integrating an implicit attention mechanism. Nevertheless, these models still rely solely on a binary classifier with no guarantee of extracting artifact-prone *fine-grained* traces.

Interestingly, in image-based deepfake detection, it has been recently demonstrated that the use of a tailored multi-task learning framework for explicitly attending artifact-prone small regions coupled with a subtle data synthesis strategy can be a way to enhance generalization and, at the same time, robustness to high-quality deepfakes [41, 42]. Nevertheless, such an approach has been disregarded in the field of video-level deepfake detection, as its extension to the video level is not straightforward. In particular, it would necessitate the characterization of subtle temporal artifacts that are inherently different from spatial ones, within both the multi-task learning framework and the data synthesis.

In this paper, we redefine deepfake video detection as a fine-grained detection task by proposing a multi-branch network that leverages synthesized data and incorporates specialized learning objectives specifically targeting both subtle spatial and temporal artifacts. As shown in Figure 1-b, a novel multi-task learning framework, termed FakeSTormer is introduced. It is formed by two auxiliary parallel branches in addition to the standard classification head, namely: **(1) a regression temporal branch** incorporating an explicit attention that aims at locating the vulnerability-prone temporal locations. It has been shown that regressing spatial vulnerabilities in specific points [42] or patches [41] can help improve the generalizability of a deepfake detector model. We refer to the definitions given in [41, 42] which describe: *"vulnerable patches/points as the patches/points that are the most likely to embed blending artifacts"*. To generalize this concept to the temporal domain, we propose locating temporal high changes in spatial vulnerable patches (see Figure 1-c). **(2) a spatial branch** to ensure a balance between the spatial and the temporal domains. In fact, detecting spatial artifacts in addition to temporal ones is crucial [56, 67]. For that purpose, we propose predicting frame-wise spatial vulnerabilities.

To create hand-free ground truths for the proposed branches, we introduce a HQ video-level data synthesis algorithm, called "Self-Blended Video (SBV)", inspired by "Self-Blended Image (SBI) [48]", enforcing temporal coherence using two proposed modules on top of SBI (detailed in Section. 3.1). Our experiments demonstrate that simply training a baseline classification model on SBV enables achieving on par performance w.r.t. state-of-the-art (SOTA), highlighting the effectiveness of SBV. Finally, for enhancing spatial and temporal modeling, we revisit the TimeSformer [2] architecture that we use as our backbone. In particular, we leverage TimeSformer's decomposed temporal and spatial attention on embedded patches, appending classification tokens for each frame and for each patch across frames, rather than a single token for the entire video. These classification tokens are then used within the spatial and classification heads, while the embedded patches are used within the temporal head. Extensive experiments on several well-known deepfake detection benchmarks show that our method outperforms the existing SOTA approaches.

Contributions. In summary, we propose in this paper:

- A novel multi-task learning framework using only real data for fine-grained video-based deepfake detection.
- Two auxiliary branches that capture both temporal and spatial vulnerabilities, that are fined-grained by definition.
- A video-level data synthesis technique called SBV that generates high-quality pseudo-fakes and is supported by a vulnerability-driven cutout augmentation strategy to avoid overfitting specific artifact-prone regions.
- A revisited version of the TimeSformer [2], specifically tailored for the proposed video-based deepfake detector.
- Extensive experiments and analyses conducted on several challenging datasets.

Paper organization. Section 2 reviews related work on video deepfake detection. Section 3 describes the proposed FakeSTormer method. Section 4 presents experiments and results. Finally, Section 5 concludes with future work.

2. Related Work

Video-based Deepfake Detection. As highlighted in [56, 71], using a naive spatio-temporal binary classification model for video-level deepfake detection can lead the model to overfit obvious artifacts, resulting in poor generalization to unseen manipulations. To address this, FTCN [71] proposes a fully temporal convolution network by reducing the spatial kernel size to one, hence decreasing the likelihood of focusing only on spatial artifacts. LipForensics [25] considers solely the mouth region, while spatio-temporal dropout [64] randomly removes parts of the input frames in both spatial and temporal domains. AltFreezing [56] separates convolution layers into spatial and temporal ones, failing to model long-term dependencies. Instead of using convolution layers, ISTVT [67] utilizes a video-

based Vision Transformer [16] with self-attention to extract longer-range correlations. Meanwhile, [9] decomposes features into spatial and temporal components. TALL [57] employs an image-level deepfake detector by converting video frames into a thumbnail layout. Despite being promising, most of the aforementioned methods solely rely on a single binary classifier that implicitly guides the feature extraction. As highlighted in the literature on image-based deepfake detection [6, 15, 33, 42], this approach might lead to overfitting specific artifacts present in training datasets. Moreover, the absence of an explicit attention mechanism to spatio-temporal artifact-prone regions can lead to poor robustness to high-quality artifacts.

Data Synthesis. A highly effective approach for enhancing the generalizability of deepfake detectors is training models with synthesized data. While frame-level solutions have been extensively studied [6, 15, 33, 48, 68], video-level augmentations remains relatively underexplored. In recent works, STC [34] generates pseudo-fake samples via time-shuffling, VB [62] perturbs landmarks per frame without imposing temporal coherence, while ST-SBV [24] injects temporal artifacts through random face scaling and blurring over time. However, these methods often introduce exaggerated temporal distortions that differ from HQ deepfakes typically exhibiting finer temporal inconsistencies.

3. Methodology

Let $\mathcal{V} \triangleq \cup_{i=1}^{N}\{(\mathbf{X}_i, y_i)\}$ be a training dataset formed by N videos, where $\mathbf{X}_i$ denotes the i^{th} video sample and y_i its associated label indicating whether the clip is real ($y_i = 0$) or fake ($y_i = 1$). Traditional methods [20, 23, 25, 56, 57, 64, 67, 71] aim to learn jointly a feature extractor $\Phi : \mathcal{V} \mapsto \mathcal{F}$ and a binary classifier $f : \mathcal{F} \to \{0, 1\}$ by minimizing the standard binary cross-entropy (BCE) loss $\mathcal{L}_{BCE}(f(\Phi(\mathbf{X}_i)), y_i)$ using the entire training set $\mathcal{V}$, with $\mathcal{F}$ being the learned feature space. As previously discussed in [33, 39, 42, 48] and also highlighted in Section 1, such a strategy might lead to poor generalization capabilities to unseen generation methods while providing only binary outputs that are not interpretable.

To tackle these issues, inspired by the literature on image-level deepfake detection [6, 15, 33, 42, 68], we introduce a novel multi-task learning framework called FakeSTormer that only relies on the real data subset denoted as $\mathcal{V}^r \subset \mathcal{V}$. Specifically, in addition to the binary classifier f, our framework includes two additional branches $h : \mathcal{F} \to \mathcal{H}$ and $g : \mathcal{F} \to \mathcal{G}$ that aim at triggering the learning of localized temporal and spatial artifact-prone features, respectively, through relevant auxiliary tasks. Note that $\mathcal{H}$ and $\mathcal{G}$ denote respectively the output spaces of h and g. The proposed branches are depicted further in Section 3.2. To provide ground truth to those branches and at the same time avoid overfitting to specific manipulations, we apply to each video belonging to $\mathcal{V}^r$ a data synthesis method described in Section 3.1, resulting in a pseudo-fake subset denoted as $\mathcal{V}^{\tilde{r}}$. Hence, our framework is trained using $\tilde{\mathcal{V}} \triangleq \{\mathcal{V}^r \cup \mathcal{V}^{\tilde{r}}\}$.

3.1. Video-Level Data Synthesis and Augmentation

Self-Blended Video. Blending-based data synthesis methods have demonstrated great performance in image-based deepfake detection [6, 7, 33, 41, 42, 48, 68]. In fact, as the blending step is common to different manipulation types, they contribute to the improvement of the generalization aspect in deepfake detection [33, 42]. Nevertheless, such an approach has been overlooked in the context of video-based deepfake detection. Hence, we propose to extend blending-based data synthesis to the video level. In particular, we revisit Self-Blended Image (SBI) [48], given its ability to produce high-quality pseudo-fake images. The proposed data synthesis approach, termed Self-Blended Video (SBV), is constituted of two main components building on top of SBI, *i.e.*, a Consistent Synthesized Parameters (CSP) module followed by a Landmark Interpolation module (LI) for preserving the temporal coherence of synthesized videos, which is essential for producing high-quality synthesized videos.

Specifically, given a real video $\mathbf{X}_i^r \in \mathcal{V}^r$ formed by T consecutive frames, we start by extracting a set of 2D landmarks $\mathbf{L}_i(t) = \{\mathbf{l}_{ij}(t)\}_{1\leq j\leq n}$ at each instant t from $\mathbf{X}_i^r(t)$, where n refers to the number of landmarks and $\{\mathbf{l}_{ij}(t)\} \in \mathbb{R}^2$. Then, we apply SBI to the 1st video frame denoted as $\mathbf{X}_i^r(t_0)$ to obtain a pseudo-fake image, i.e., $\mathbf{X}_i^{\tilde{r}}(t_0)$ and a blending mask $\mathbf{M}_i(t_0)$. All related blending parameters $\theta^{(sbi)}$ (e.g., ConvexHull type, Mask deformation kernels, blending ratio, etc.) are then conserved for synthesizing the remaining video frames. However, using those parameters solely cannot guarantee the temporal consistency of pseudo-fake videos since the geometry of landmarks can significantly vary over time. To mitigate the issue, we propose to re-interpolate each landmark $\mathbf{l}_i(t)$ based on $\mathbf{l}_i(t-1)$ for $t > t_0$ as follows,

$$\mathbf{l}_i(t) = \begin{cases} \mathbf{l}_i(t-1) + \dfrac{\mathbf{l}_i(t) - \mathbf{l}_i(t-1)}{\text{round}(d/\bar{d})}, \text{if } d > \tau \\ \quad \text{where } d = \|\mathbf{l}_i(t) - \mathbf{l}_i(t-1)\|_2/n \\ \mathbf{l}_i(t), \text{otherwise,} \end{cases} \tag{1}$$

where d represents the normalized distance between the position of a given landmark at the instants t and $t-1$, τ is a constant threshold for determining when to interpolate (interpolation intervenes only in the presence of drastic changes), and $\bar{d}$ is empirically chosen. Overall, a higher d value will push the updated point to move closer to the previous landmark position $\mathbf{l}_i(t-1)$, hence contributing to smooth the landmark position over time. However, excessive smoothing can be disadvantageous, as it can discard temporal artifacts. To address this, we use a round operator

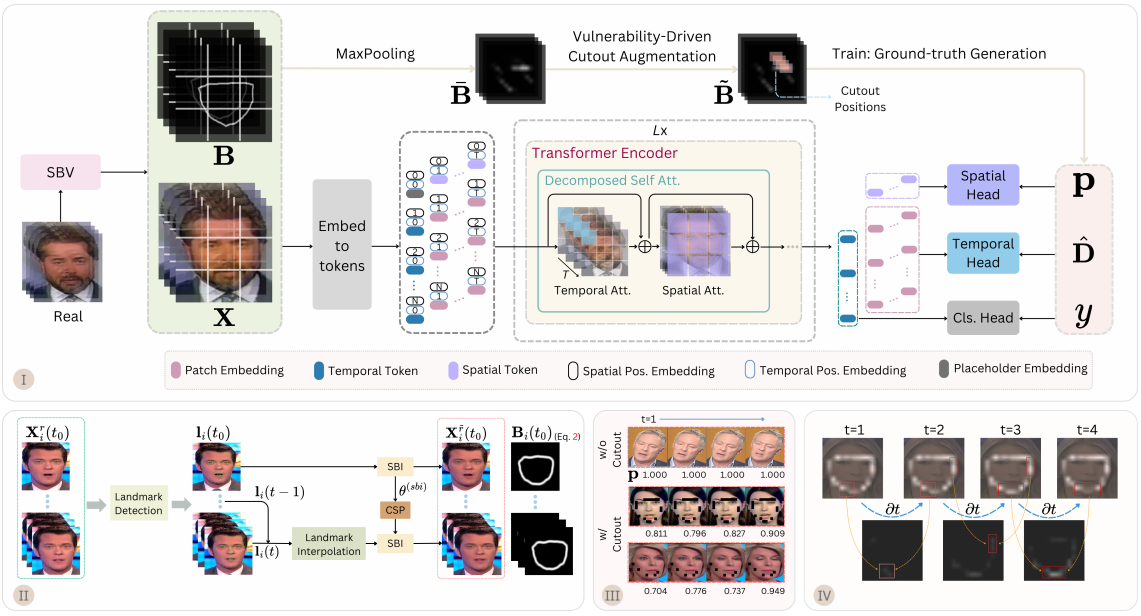

Figure 2. **I) Overview of the proposed framework:** Our multi-task learning framework, FakeSTormer, consists of three branches, i.e., the temporal branch (h), the spatial branch (g), and the standard classification branch (f). Those branches are specially designed to facilitate the disentanglement learning of spatial-temporal features. The hand-free ground-truth data to train the framework are generated based on our proposed video-level data synthesis algorithm coupled with a vulnerability-driven Cutout strategy. **II) Overview of generating a self-blended video:** It contains two main components, including a landmark interpolation module (*LI*) and the consistent utilization of synthesized parameters (*CSP*). **III) Examples of pseudo-fake videos:** with(*w/*) and without(*w/o*) vulnerability-driven Cutout and their corresponding soft labels. We apply the Cutout data augmentation at the same spatial locations throughout video frames. **IV) Extraction of temporal vulnerabilities:** We compute derivatives of the spatial vulnerabilities over time.

to incorporate slight errors. Hence, the proposed SBV data synthesis produces high-quality pseudo-fake videos incorporating subtle temporal artifacts.

As a result, we obtain the pseudo-fake video $\mathbf{X}_i^{\tilde{f}} \in \mathcal{V}^{\tilde{r}}$ and its blending mask sequence $\mathbf{M}_i \in \mathbb{R}^{T\times H\times W}$, with H and W being the image height and width, respectively. An illustration of SBV is given in Figure 2-II. Additional samples, as well as the detailed algorithm, are provided in supplementary materials. It is important to note that, despite being simple, SBV is generic and applicable to any existing video-level deepfake detection approach.

Vulnerability-Driven Cutout Augmentation. Previous works [54, 72] have demonstrated that deep learning methods are often impacted by overfitting. Deepfake detectors might even be more sensitive to this phenomenon as deepfakes are typically characterized by localized artifacts [42]. One solution to regularize training is data augmentation. As such, we propose, in addition to SBV, a novel Cutout data augmentation driven by vulnerable patches, i.e., image patches that are prone to blending artifacts [41]. We posit that by masking the most vulnerable regions, overfitting risks will be reduced, as the model will be pushed to learn from other areas. This masking strategy has already been explored in other computer vision fields such as Classification [8, 12], Object Detection [21] demonstrating great potential.

Specifically, similar to [33], we create a set of blending boundaries $\mathbf{B}$ using a randomly generated blending mask $\mathbf{M}$ as follows,

$$\mathbf{B} = (\mathbf{1} - \mathbf{M}) * \mathbf{M} * 4, \mathbf{B} \in \mathbb{R}^{T\times H\times W}, \tag{2}$$

with $*$ being the element-wise multiplication and $\mathbf{1}$ an all-one matrix. Inspired by [41], vulnerability values are then quantified at the patch level in a non-overlapping manner by applying a MaxPooling function as follows,

$$\bar{\mathbf{B}} = \text{MaxPooling}(\mathbf{B}), \quad \bar{\mathbf{B}} \in \mathbb{R}^{T\times\sqrt{N}\times\sqrt{N}}, \tag{3}$$

where N indicates the number of patches.

After that, we define a threshold τ_{cutout} that is randomly selected from the range $(0.5, 1.0]$. We use the latter to define the set of patches to be masked $\mathcal{P} = \{(l,m) \mid \bar{\mathbf{B}}_{l,m}(t_0) > \tau_{cutout}\}$ within the first frame. The set $\mathcal{P}$ is then used to mask out patches at those locations not only in the first frame but also in the entire video to enforce temporal consistency that is crucial for generating high-quality pseudo-fakes. After masking those patches over time, we finally obtain the masked blending boundary denoted as $\tilde{\mathbf{B}}$. This results in masking the most vulnerable regions, i.e., the regions that are the most likely to include blending artifacts. Figure 2-III shows some examples of the proposed cutout augmentation.

3.2. FakeSTormer

Our multi-task framework, called FakeSTormer, is inspired by [41, 42], where auxiliary branches are designed to push the feature extractor to focus on vulnerabilities. As discussed earlier, the vulnerability is defined in [41, 42] as the pixels/patches that are the most likely to be impacted by blending artifacts. This strategy is therefore claimed to allow the detection of subtle artifacts that are generic across different types of manipulations. While such a vulnerability-driven approach has shown very promising results [41, 42], it does not take into account the temporal nature of videos. Therefore, in addition to spatial vulnerabilities, we argue that there is a need to model temporal vulnerabilities, which we define as significant temporal changes in the blending boundary. Specifically, we introduce two additional branches, namely a temporal head h and a spatial one g. The branch h predicts the derivatives of the blending boundary over time which can reflect high changes, typically characterizing temporal artifacts. Moreover, we suggest the use of a spatial branch g which enables predicting soft labels representing the forgery intensity encoded in each frame, computed from vulnerability information. The proposed framework relies on the TimeSformer backbone [2], which we revisit for better modeling spatial and temporal information.

Herein, we first describe the proposed revisited TimeSformer-based feature extractor Φ in Section 3.2.1. We then detail the two additional temporal and spatial heads in Section 3.2.2 and Section 3.2.3, respectively. Finally, we give the overall training details in Section 3.2.4.

3.2.1. Backbone: Revisited TimeSformer

We choose TimeSformer [2] as our feature extractor given its ability to effectively capture separate long-range temporal information and spatial features. In TimeSformer, a video input $\mathbf{X} \in \mathbb{R}^{C\times T\times H\times W}$ results in an embedding matrix input $\mathbf{Z}^0 \in \mathbb{R}^{T\times N\times D}$. A global class token $\mathbf{z}_{cls}$ attends all patches and is then used for classification. This mechanism implicitly captures mixed spatio-temporal features, which might lead to overfitting one type of artifact. 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 $\mathbf{z}_s^0 \in \mathbb{R}^D$ and a temporal token $\mathbf{z}_t^0 \in \mathbb{R}^D$, respectively. These tokens will independently interact only with patch embeddings belonging to their dimension axis by leveraging the decomposed SA [2]. 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, as described in Figure 2-I. Formally, the feature extraction process can be summarized as follows,

$$[\mathbf{Z}^L, \mathbf{z}_s^L, \mathbf{z}_t^L] = \Phi(\mathbf{X}), \tag{4}$$

where $\mathbf{Z}^L$ is the final patch embedding matrix, $\mathbf{z}_s^L$ the resulting set of spatial tokens, and $\mathbf{z}_t^L$ 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. More details about the implementation of the proposed revisited TimeSformer are given in supplementary materials.

3.2.2. Temporal Head h

Ground Truths. Our temporal head h aims to model fine-grained temporal vulnerabilities in deepfake videos through a regression task. First, to generate ground truth data for the branch h, we hypothesize that temporal high-changes in the blending boundary can reflect the presence of temporal artifacts (see Figure 2-IV). To achieve this, we compute $\mathbf{D}$ based on $\tilde{\mathbf{B}}$ such that:

$$\mathbf{D} = \frac{\partial \tilde{\mathbf{B}}}{\partial t}, \quad \mathbf{D} \in \mathbb{R}^{T\times\sqrt{N}\times\sqrt{N}}. \tag{5}$$

More details regarding the derivative calculation are provided in supplementary materials. To stabilize training, $\mathbf{D}$ is standardized resulting in $\hat{\mathbf{D}} \in \mathbb{R}^{T\times\sqrt{N}\times\sqrt{N}}$. Experiments with different normalization strategies are reported in supplementary materials.

Architecture Design. In order to construct the regression head for predicting $\hat{\mathbf{D}}$, we take the patch embedding matrix $\mathbf{Z}^L$ as input and process them to produce 3D features as follows,

$$\mathbf{F} = \text{Reshape}(\mathbf{Z}^L), \mathbf{F} \in \mathbb{R}^{D\times T\times\sqrt{N}\times\sqrt{N}}. \tag{6}$$

To estimate temporal derivatives, we employ two 3D convolution blocks (3DCnvB) with 3-dimensional temporal kernels and 1-dimensional spatial kernels [71] as follows,

$$\tilde{\mathbf{D}} = h(\mathbf{F}) = \text{3DCnvB}_{3\times1\times1}(\text{3DCnvB}_{3\times1\times1}(\mathbf{F})), \tag{7}$$

where $\tilde{\mathbf{D}} \in \mathbb{R}^{T\times\sqrt{N}\times\sqrt{N}}$. Each convolution block comprises a 3D convolution layer, followed by a BatchNorm and a GELU layer.

Objective Function. For training the temporal branch, we optimize the following Mean Squared Error (MSE) loss,

$$\mathcal{L}_h = \frac{1}{T\times N}\|\hat{\mathbf{D}} - \tilde{\mathbf{D}}\|_2^2, \tag{8}$$

with $\|.\|_2$ referring to the L_2 norm.

3.2.3. Spatial Head g

Ground Truths. To avoid overfitting one type of artifact, we enforce the model to explicitly predict soft labels representing the intensity level of spatial artifacts for each video

frame. Note that several works [4, 40, 49, 52, 66] have leveraged soft labels for training regularization. Given a pseudo-fake video $\mathbf{X} = (\mathbf{X}(t))_{t\in[[1,T]]}$ formed by T frames and $\tilde{\mathbf{B}} = (\tilde{\mathbf{B}}(t))_{t\in[[1,T]]}$ its associated cutout blending boundary, the ground truth for these soft labels is generated for each frame t as follows,

$$p(t) = \max_{l,m\in[[1,\sqrt{N}]]} (\tilde{\mathbf{B}}(t)), \tag{9}$$

resulting in the ground truth for training the spatial branch denoted as $\mathbf{p} = (p(t))_{t\in[[1,T]]}$. We note that $\mathbf{p} = \mathbf{1}^T$ if cutout is not applied and $\mathbf{p} = \mathbf{0}^T$ for a real video.

Architecture Design. To predict the proposed soft labels, a Multi-Layer Perceptron (MLP) is applied to the set of spatial tokens $\mathbf{z}_s^L$, as follows,

$$\tilde{\mathbf{p}} = g(\mathbf{z}_s^L) = \text{MLP}(\mathbf{z}_s^L), \quad \tilde{\mathbf{p}} \in \mathbb{R}^T. \tag{10}$$

Objective Function. To train the spatial branch, we optimize the following Binary Cross Entropy (BCE) loss similar to [52, 66],

$$\mathcal{L}_g = \text{BCE}(\tilde{\mathbf{p}}, \mathbf{p}). \tag{11}$$

3.2.4. Overall Training Objective

Finally, for the standard classification head f, we use the set of temporal tokens $\mathbf{z}_t^L$ such that the predicted label $\tilde{y}$ is given by,

$$\tilde{y} = f(\mathbf{z}_t^L) = \text{MLP}(\mathbf{z}_t^L). \tag{12}$$

The classification loss $\mathcal{L}_c$ is then given by applying a BCE between the ground-truth label y and the predicted label $\tilde{y}$.

Overall, the network is trained by optimizing the following loss:

$$\mathcal{L} = \lambda_c\mathcal{L}_c + \lambda_h\mathcal{L}_h + \lambda_g\mathcal{L}_g, \tag{13}$$

where $\lambda_c, \lambda_h, \lambda_g$ are hyper-parameters to balance the training of the three branches.

4. Experiment

4.1. Settings

Datasets. We set up our datasets following several works [3, 9, 55–57, 59, 67, 71]. For both training and validation, we employ **FaceForensics++** (FF++) [46], which consists of four manipulation methods for the fake data (Deepfakes (DF) [10], FaceSwap (FS) [31], Face2Face (F2F) [51], and NeuralTextures (NT) [50]). 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 [9, 56, 57, 67, 71]. For further validation, we also evaluate on the following datasets: **Celeb-DFv2** (CDF) [36], **DeepfakeDetection** (DFD) [17], **Deepfake Detection Challenge Preview** (DFDCP) [13], **Deepfake Detection Challenge** (DFDC) [14], **WildDeepfake** (DFW) [74], **DF40** [61], and **DiffSwap** [63, 70] generated using a recent diffusion-based approach [70]. Further details on these datasets are provided in supplementary materials.

Data Pre-processing. Following the splitting convention [46], we extract 256, 32, and 32 consecutive frames for training, validation, and testing, respectively. Facial regions are cropped using Face-RetinaNet [11] and resized to a fixed resolution of 224×224. Additionally, we store 81 facial landmarks for each frame, extracted using Dlib [30].

Method	Training		Test set AUC (%)					
	Real	Fake	CDF	DFD	DFDCP	DFDC	DFW	DiffSwap
Xception [46]	✓	✓	73.7	-	-	70.9	-	-
MATT [69]	✓	✓	68.3	92.9	63.0	-	65.7	-
RECCE [3]	✓	✓	70.9	98.2	-	-	68.2	-
SBI [48]	✓	×	90.6	-	-	72.4	-	-
SFDG [55]	✓	✓	75.8	88.0	73.6	-	69.3	-
LSDA [60]	✓	✓	91.1	-	77.0	-	-	-
STIL [23]	✓	✓	75.6	-	-	-	-	-
LipForensics [25]	✓	✓	82.4	-	-	73.5	-	-
RealForensics [26]	✓	✓	86.9	82.2	75.9	-	-	-
FTCN [71]	✓	✓	86.9	94.4	74.0	71.0	-	-
ISTVT [67]	✓	✓	84.1	-	74.2	-	-	-
AltFreezing [56]	✓	✓	89.5	**98.5**	-	-	-	-
Swin+TALL [57]	✓	✓	90.8	-	76.8	-	-	-
StyleLatentFlows [9]	✓	✓	89.0	96.1	-	-	-	-
LFGDIN [63]	✓	✓	90.4	-	80.8	-	-	85.7
FakeSTormer ($T = 4$)	✓	×	**92.4**	**98.5**	**90.0**	**74.6**	**74.2**	**96.9**
FakeSTormer ($T = 8$)	✓	×	92.4	98.2	90.0	74.9	75.9	97.1
FakeSTormer ($T = 16$)	✓	×	92.8	98.6	90.2	75.1	75.3	97.2

Table 1. **Generalization to unseen datasets**. AUC (%) comparisons at *video-level* on multiple unseen datasets [13, 14, 17, 36, 70, 74]. All detectors are trained on FF++(c23). Results are directly extracted from the original papers and from [25, 42]. **Bold** and Underlined text, respectively highlight the best and the second best performance, excluding the variants of our framework with $T = 8$ and $T = 16$.

Method	Training set		Cross-dataset			DF40 subset		
	Real	Fake	CDF	DFDCP	DiffSwap	BlendFace	FSGAN	MobileSwap
Face X-ray [33]	✓	✓	79.5	-	-	-	-	-
PCL+I2G [68]	✓	✓	90.0	74.3	-	-	-	-
SLADD [6]	✓	✓	79.7	-	-	-	-	-
SBI [48]	✓	×	93.2	86.2	90.6	86.5	85.4	86.6
LAA-Net [42]	✓	×	95.4	86.9	92.1	**91.2**	94.2	93.9
STC-Scratch [34]	✓	×	83.4	86.8	-	-	-	-
STC-Pretrain [34]	✓	×	95.8	89.4	-	-	-	-
ST-SBV [24]	✓	×	90.3	91.2	-	-	-	-
StA+VB [62]	✓	×	94.7	90.9	-	90.6	**96.4**	94.6
TimeSformer [2] + SBV	✓	×	94.9	93.0	93.3	89.7	94.6	94.6
FakeSTormer	✓	×	**96.5**	**94.1**	**97.7**	91.1	**96.4**	**95.0**

Table 2. **AUC(%) comparison at video-level with other data synthesis methods.** For fair comparison, we train our FakeSTormer on raw data of FF++(c0), and test *cross-dataset* on [13, 36, 70] and *cross-manipulation* on three subsets of [61].

Method	Training set		FF++ LQ (%)	
	Real	NT	DF	FS
Xception [46]	✓	✓	58.7	51.7
Face X-ray [33]	✓	✓	57.1	51.0
F3Net [44]	✓	✓	58.3	51.9
RFM [54]	✓	✓	55.8	51.6
SRM [37]	✓	✓	55.5	52.9
SLADD [6]	✓	✓	62.8	56.8
TALL-Swin [57]	✓	✓	63.2	51.4
ResNet3D* [27]	✓	✓	66.8	60.6
TimeSformer* [2]	✓	✓	73.3	54.4
Ours	✓	×	**85.3**	**62.1**

Table 3. **Generalization on heavily compressed data (LQ).** AUC (%) comparisons on FF++ (LQ) [46] with a high compression level (c40). The results for comparison are directly extracted from [6, 38]. The symbol $*$ denotes our implementation.

Method	Training set		FF++ (%)				
	Real	Fake	DF	FS	F2F	NT	Avg.
Xception [46]	✓	✓	93.9	51.2	86.8	79.7	77.9
Face X-ray [33]	✓	✓	99.5	93.2	94.5	92.5	94.9
SBI [48]	✓	×	98.6	95.4	92.6	82.3	92.2
LSDA [60]	✓	✓	96.9	95.1	96.4	94.9	95.8
LipForensics [25]	✓	✓	99.7	90.1	99.7	99.1	97.1
FTCN [71]	✓	✓	99.8	99.6	98.2	95.6	98.3
RealForensics [26]	✓	✓	**100**	97.1	99.7	99.2	99.0
AltFreezing [56]	✓	✓	99.8	99.7	98.6	96.2	98.6
StyleLatentFlows [9]	✓	✓	99.7	98.8	98.6	96.4	98.4
NACO [65]	✓	✓	99.9	99.7	99.8	99.4	99.7
LFGDIN [63]	✓	✓	96.2	80.5	90.5	81.7	87.2
Ours (c23)	✓	×	99.9	97.8	98.5	97.2	98.4
Ours (c0)	✓	×	**100**	**99.8**	**99.9**	**99.7**	**99.9**

Table 4. **Generalization to unseen manipulations**. AUC (%) comparisons on FF++ [46], which consists of four manipulation methods (DF, FS, F2F, NT).

Further details are provided in the supplementary materials.

Evaluation Metrics. For fair comparisons with SOTA methods, we use the widely adopted Area Under the Curve (AUC) metric at the video level [9, 25, 26, 56, 57, 67, 71].

Implementation Details. Our framework is initialized with pretrained MAE weights [28] and trained for 100 epochs using the SAM optimizer [18] with a weight decay of 10^{-4} and a batch size of 32. The learning rate starts at 5×10^{-4} for the first quarter of training and decays to 0 thereafter. The backbone is frozen for the first 5 epochs for warm-up, then all layers are unfrozen. Data augmentation includes ColorJittering at the video level and our proposed Cutout. Experiments are conducted on four NVIDIA A100 GPUs, with $\tau = 0.35$ and $\bar{d} = 0.2$ (Eq. (1)), and $T = 4$ frames in most experiments.

4.2. Comparison with State-of-the-art Methods

Generalization to Unseen Datasets. To assess the generalization capabilities of our method, we conduct evaluations using the challenging *cross-dataset* setup [3, 42, 55, 56, 71], validating on unseen datasets (i.e., datasets other than FF++). The results are detailed in Table 1 and Table 2.

As shown, our method achieves comparable results on DFD while surpassing SOTA methods on other datasets. Specifically, it significantly outperforms prior video deepfake detection techniques, including spatio-temporal learning-based methods like AltFreezing [56] and ISTVT [67], as well as various data synthesis approaches. Moreover, our method exhibits superior performance on the large-scale DFDC dataset and the challenging in-the-wild DFW dataset. These results further confirm the enhanced generalization ability of FakeSTormer compared to recent methods.

Generalization on Heavily Compressed Data. Following previous work [6, 38], we also evaluate FakeSTormer on heavily compressed FF++(c40) data. In addition to comparing with several SOTA methods, we train ResNet3D [27], commonly used in deepfake video detection [9, 56, 71], and TimeSformer [2] on NT, then test on DF and FS. The comparison results are presented in Table 3. Our method achieves notably higher AUC scores than other methods across both testing subsets, highlighting its robust generalization capability under various data compression conditions.

SBV	V-CutOut	g	h	Test set AUC (%)						
				CDF	DFD	DFDCP	DFDC	DFW	DiffSwap	Avg.
×	×	×	×	61.5	62.8	59.4	58.5	65.2	71.6	63.2
✓	×	×	×	90.7	95.7	87.9	72.2	70.9	92.9	85.1(↑21.9)
✓	✓	×	×	91.1	96.0	87.6	72.6	71.0	93.1	85.2(↑22.0)
✓	✓	✓	×	92.2	95.4	88.5	72.8	71.3	93.8	85.7(↑22.5)
✓	×	×	✓	**93.4**	**98.5**	88.5	72.8	69.6	**97.3**	86.7(↑23.5)
✓	✓	✓	✓	92.4	**98.5**	**90.0**	**74.6**	**74.2**	96.9	**87.8**(↑24.6)

Table 5. **Ablation study of framework's components**. Gray indicates the use of original fake data for training.

Generalization to Unseen Manipulations. Table 4 compares our framework with SOTA methods on FF++. Other methods [9, 25, 56, 71] use a cross-manipulation setup, training on three forgery types and evaluating on the remaining one. In contrast, our approach trains only on real videos, treating all manipulations as unseen. Despite this, our method shows competitive performance with the others, even without being trained on specific forgery types.

Robustness to Unseen Perturbations. Deepfakes are widely shared on social media, where various perturbations can affect their appearance. Following [29], we evaluate FakeSTormer's robustness across six unseen degradation types at five levels, comparing it with other augmented-based methods [33, 35, 42, 48]. Figure 3 shows AUC scores for each method on these perturbations, using models trained on FF++. Our results demonstrate that FakeSTormer outperforms prior methods on most distortions, with a slight drop compared to LAA-Net [42] for Change Saturation. Nonetheless, FakeSTormer achieves higher performance on average, especially at higher severity levels, highlighting its superior generalization and robustness. Detailed scores are in supplementary materials.

4.3. Additional Discussions

Ablation Study of the FakeSTormer's Components. We conduct ablation studies to assess the impact of each component in our framework, as shown in Table 5. Using TimeSformer trained on FF++ as the baseline, we experiment with different combinations of components: Self-Blended Video (SBV), Vulnerability-driven CutOut (V-CutOut), the spatial branch (g), and the temporal branch (h). Each component improves performance, with SBV providing the most significant boost by generating high-quality pseudo-fake data that aids generalization to unseen datasets.

Ablation Study of the SBV's Components. SBV enhances SBI [48] with CSP and LI for robust pseudo-fake generation in video data. Table 6 shows that without these components, simply stacking frame-wise SBIs fails to produce consistent temporal features, leading to overfitting on more obvious artifacts and poor generalization [56]. A qualitative comparison is provided in supplementary materials.

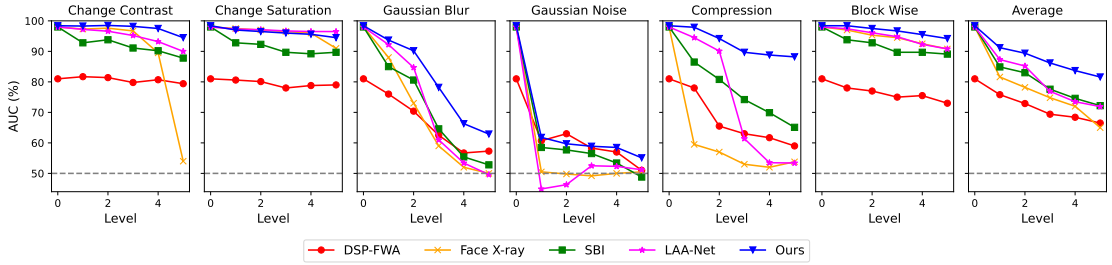

Figure 3. **Robustness to unseen perturbations.** AUC (%) under five different degradation levels for various types of perturbations [29] on FF++ [46]. "Average" denotes the mean across all corruptions at each level. Best viewed in color.

Method	Comp.	CSP	LI	Test set AUC (%)					
				CDF	DFD	DFDCP	DFDC	DFW	DiffSwap
Stacked SBIs	c23	×	×	48.4	49.1	48.9	51.7	52.8	55.5
+ CSP	c23	✓	×	84.1	89.2	86.1	69.5	65.5	85.7
SBV	c23	✓	✓	90.7	95.7	87.9	72.2	70.9	92.9
SBV	c0	✓	✓	**94.9**	**97.6**	**93.0**	**76.4**	**75.3**	**93.3**

Table 6. **Ablation study of SBV's components**. Performance analyses of different SBV's components using cross-evaluation on multiple datasets [13, 14, 17, 36, 70, 74].

λ_c	λ_h	λ_g	Test set AUC (%)						
			CDF	DFD	DFDCP	DFDC	DFW	DiffSwap	Avg.
0.9	1	0.1	89.5	91.0	**93.2**	71.5	**74.7**	90.6	85.1
0.9	10	0.1	**92.5**	95.7	86.8	71.7	72.7	94.5	85.7
0.9	100	0.1	91.6	98.0	87.3	73.6	70.6	96.2	86.2
0.8	100	0.2	92.4	**98.5**	90.0	**74.6**	74.2	96.9	**87.8**
0.5	100	0.5	92.4	98.0	88.1	74.5	72.1	**97.1**	87.0

Table 7. **Impact of loss balancing factors**. AUC (%) comparisons of FakeSTormer trained with different values of λ_c, λ_h, and λ_g on cross-dataset setup, demonstrating robustness to varying hyperparameter settings.

Influence of Number of Frames. Increasing the number of frames T provides more fine-grained temporal information. In Table 1, we vary T values by fixing it to 4, 8, and 16. Our results show a consistent performance improvement with more frames, confirming our hypothesis. However, increasing T also incurs a higher computational cost.

Impact of Loss Balancing Factors. We introduce three hyperparameters, λ_c, λ_g, and λ_h in Eq (13) to balance the training among the three branches of our framework. In Table 7, we analyze the impact of these hyperparameters using various values. Our results show that the method is robust to a range of hyperparameter values, with the best performance achieved when λ_c, λ_g, and λ_h are set to 0.8, 0.2, and 100, respectively.

4.4. Visualization of Saliency Maps

To analyze the contribution of the two proposed branches h and g in the detection performance of FakeSTormer, we visualize the input regions activated by those branches. For that purpose, we adopt Grad-CAM [47] for the temporal branch h and utilize the final SA scores of spatial tokens for the spatial branch g. The visualization results from various datasets are presented in Figure 4. It can be observed that FakeSTormer can discriminate between real and fake videos by focusing on very few, different local areas, even without having seen those types of forgeries during training.

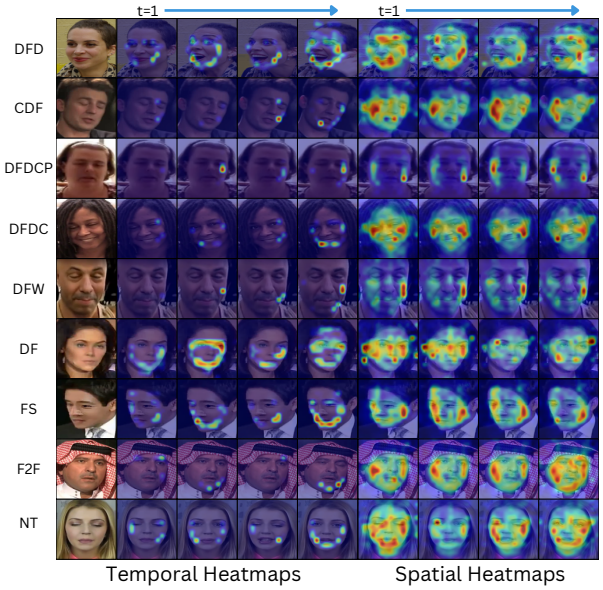

Figure 4. **Visualization of Saliency Maps**. The second-fifth and sixth-ninth columns represent temporal heatmaps and spatial heatmaps on different frames in the video, respectively. All datasets are unseen during validation.

5. Conclusion

This paper introduces a fine-grained approach for generalizable deepfake video detection with two main contributions. First, we propose a multi-task learning framework that targets both subtle spatial and fine-grained temporal vulnerabilities in high-fidelity deepfake videos, incorporating a standard classification branch along with two new auxiliary branches (temporal and spatial). These proposed branches help the model focus on vulnerable regions and provide more valuable insights into how the network *sees* the data while offering more robustness to high-quality deepfakes. This framework is further supported by the introduction of a high-quality pseudo-fake generation technique. Extensive experiments on several challenging benchmarks demonstrate that FakeSTormer achieves superior performance compared to SOTA methods.

Acknowledgment

This work is supported by the Luxembourg National Research Fund, under the BRIDGES2021/IS/16353350/FaKeDeTeR, and by POST Luxembourg. Experiments were performed on the Luxembourg national supercomputer MeluXina. The authors gratefully acknowledge the LuxProvide teams for their expert support.

References

[1] Yuval Alaluf, Or Patashnik, Zongze Wu, Asif Zamir, Eli Shechtman, Dani Lischinski, and Daniel Cohen-Or. Third time's the charm? image and video editing with stylegan3. *CoRR*, abs/2201.13433, 2022. 1

[2] 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, 5, 6, 7

[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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4103–4112, 2022. 1, 6, 7

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *CoRR*, abs/2104.14294, 2021. 6

[5] Heather Chen and Kathleen Magramo. Finance worker pays out $25 million after video call with deepfake "chief financial officer". https://edition.cnn.com/2024/02/04/asia/deepfake-cfo-scam-hong-kong-intl-hnk/index.html, 2024. [Online; accessed 4-February-2024]. 1

[6] 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 (CVPR)*, pages 18710–18719, 2022. 1, 3, 6, 7

[7] Liang Chen, Yong Zhang, Yibing Song, Jue Wang, and Lingqiao Liu. Ost: Improving generalization of deepfake detection via one-shot test-time training. In *Advances in Neural Information Processing Systems*, pages 24597–24610. Curran Associates, Inc., 2022. 1, 3

[8] Juhwan Choi and YoungBin Kim. Colorful cutout: Enhancing image data augmentation with curriculum learning, 2024. 4

[9] 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. 3, 6, 7

[10] Deepfakes. Faceswapdevs. https://github.com/deepfakes/faceswap, 2019. 6

[11] 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. 6

[12] Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout, 2017. 4

[13] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton-Ferrer. The deepfake detection challenge (DFDC) preview dataset. *CoRR*, abs/1910.08854, 2019. 6, 8

[14] 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. 6, 8

[15] 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 (CVPR)*, pages 3994–4004, 2023. 1, 3

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. 3

[17] Nick Dufour and Andrew Gully. Contributing data to deepfake detection research. https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html, 2019. 6, 8

[18] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *CoRR*, abs/2010.01412, 2020. 7

[19] Tsu-Jui Fu, Xin Eric Wang, Scott T. Grafton, Miguel P. Eckstein, and William Yang Wang. M3l: Language-based video editing via multi-modal multi-level transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10513–10522, 2022. 1

[20] Ipek Ganiyusufoglu, L. Minh Ngô, Nedko Savov, Sezer Karaoglu, and Theo Gevers. Spatio-temporal features for generalized detection of deepfake videos. *CoRR*, abs/2010.11844, 2020. 1, 3

[21] Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. Keepaugment: A simple information-preserving data augmentation approach. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1055–1064, 2021. 4

[22] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. 1

[23] Zhihao Gu, Yang Chen, Taiping Yao, Shouhong Ding, Jilin Li, Feiyue Huang, and Lizhuang Ma. Spatiotemporal inconsistency learning for deepfake video detection. *Proceedings of the 29th ACM International Conference on Multimedia*, 2021. 1, 3, 6

[24] Weinan Guan, Wei Wang, Bo Peng, Jing Dong, and Tieniu Tan. St-sbv: Spatial-temporal self-blended videos for deepfake detection. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 274–288. Springer, 2024. 2, 3, 6

[25] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Lips don't lie: A generalisable and robust approach to face forgery detection. *CoRR*, abs/2012.07657, 2020. 1, 2, 3, 6, 7

[26] 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 (CVPR)*, pages 14950–14962, 2022. 6, 7

[27] 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. 6, 7

[28] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, 2022. 7

[29] 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 *CVPR*, 2020. 7, 8

[30] Davis E. King. Dlib-ml: A machine learning toolkit. *J. Mach. Learn. Res.*, 10:1755–1758, 2009. 6

[31] Marek Kowalski. Faceswap. https://github.com/MarekKowalski/FaceSwap, 2018. 6

[32] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Advancing high fidelity identity swapping for forgery detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5073–5082, 2020. 1

[33] 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. 1, 3, 4, 6, 7

[34] 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. 2, 3, 6

[35] 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. 7

[36] 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. 6, 8

[37] 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 (CVPR)*, pages 16317–16326, 2021. 6

[38] Qingxuan Lv, Yuezun Li, Junyu Dong, Sheng Chen, Hui Yu, Huiyu Zhou, and Shu Zhang. Domainforensics: Exposing face forgery across domains via bi-directional adaptation. *IEEE Transactions on Information Forensics and Security*, 19:7275–7289, 2024. 6, 7

[39] Nesryne Mejri, Enjie Ghorbel, and Djamila Aouada. Untag: Learning generic features for unsupervised type-agnostic deepfake detection. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023. 1, 3

[40] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? *CoRR*, abs/1906.02629, 2019. 6

[41] 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, 2, 3, 4, 5

[42] 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. 1, 2, 3, 4, 5, 6, 7

[43] Yuval Nirkin, Yosi Keller, and Tal Hassner. FSGANv2: Improved subject agnostic face swapping and reenactment. IEEE, 2022. 1

[44] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European conference on computer vision*, pages 86–103. Springer, 2020. 6

[45] Reuters. South korea to criminalize watching or possessing sexually explicit deepfakes. https://edition.cnn.com/2024/09/26/asia/south-korea-deepfake-bill-passed-intl-hnk/index.html, 2024. [Online; accessed 26-September-2024]. 1

[46] 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, 6, 7, 8

[47] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR*, abs/1610.02391, 2016. 8

[48] 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. 1, 2, 3, 6, 7

[49] Guocong Song and Wei Chai. Collaborative learning for deep neural networks. In *Neural Information Processing Systems*, 2018. 6

[50] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *CoRR*, abs/1904.12356, 2019. 6

[51] 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. 1, 6

[52] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *CoRR*, abs/2012.12877, 2020. 6

[53] Jane Wakefield. Deepfake presidents used in Russia-Ukraine war. https://www.bbc.com/news/technology-60780142, 2022. [Online; accessed 7-March-2023]. 1

[54] Chengrui Wang and Weihong Deng. Representative forgery mining for fake face detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4, 6

[55] 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. 6, 7

[56] 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. 1, 2, 3, 6, 7

[57] 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. 1, 3, 6, 7

[58] Zhiliang Xu, Zhibin Hong, Changxing Ding, Zhen Zhu, Junyu Han, Jingtuo Liu, and Errui Ding. Mobilefaceswap: A lightweight framework for video face swapping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 1

[59] 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. 6

[60] Zhiyuan Yan, Yuhao Luo, Siwei Lyu, Qingshan Liu, and Baoyuan Wu. Transcending forgery specificity with latent space augmentation for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8984–8994, 2024. 6, 7

[61] 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. 6

[62] Zhiyuan Yan, Yandan Zhao, Shen Chen, Mingyi Guo, Xinghe Fu, Taiping Yao, Shouhong Ding, Yunsheng Wu, and Li Yuan. Generalizing deepfake video detection with plug-and-play: Video-level blending and spatiotemporal adapter tuning. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 12615–12625, 2025. 2, 3, 6

[63] 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. 2, 6, 7

[64] Daichi Zhang, Fanzhao Lin, Yingying Hua, Pengju Wang, Dan Zeng, and Shiming Ge. Deepfake video detection with spatiotemporal dropout transformer. In *Proceedings of the 30th ACM international conference on multimedia*, pages 5833–5841, 2022. 1, 2, 3

[65] Daichi Zhang, Zihao Xiao, Shikun Li, Fanzhao Lin, Jianmin Li, and Shiming Ge. Learning natural consistency representation for face forgery video detection. In *European Conference on Computer Vision*, pages 407–424. Springer, 2024. 7

[66] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *CoRR*, abs/1710.09412, 2017. 6

[67] 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. 1, 2, 3, 6, 7

[68] Eric Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. Learning self-consistency for deepfake detection. In *ICCV 2021*, 2021. 3, 6

[69] 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*, pages 2185–2194, 2021. 1, 6

[70] 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. 6, 8

[71] 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. 1, 2, 3, 5, 6, 7

[72] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *CoRR*, abs/1708.04896, 2017. 4

[73] Yuhao Zhu, Qi Li, Jian Wang, Chengzhong Xu, and Zhenan Sun. One shot face swapping on megapixels. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 4834–4844, 2021. 1

[74] 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. 6, 8