# Delving into Sequential Patches for Deepfake Detection

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Recent advances in face forgery techniques produce nearly visually untraceable deepfake videos, which could be leveraged with malicious intentions. As a result, researchers have been devoted to deepfake detection. Previous studies has identified the importance of local low-level cues and temporal information in pursuit to generalize well across deepfake methods, however, they still suffer from robustness problem against post-processings. In this work, we propose the **L**ocal- & **T**emporal-aware **T**ransformer-based Deepfake **D**etection (**LTTD**) framework, which adopts a local-to-global learning protocol with a particular focus on the valuable temporal information within local sequences. Specifically, we propose a Local Sequence Transformer (LST), which models the temporal consistency on sequences of restricted spatial regions, where low-level information is hierarchically enhanced with shallow layers of learned 3D filters. Based on the local temporal embeddings, we then achieve the final classification in a global contrastive way. Extensive experiments on popular datasets validate that our approach effectively spots local forgery cues and achieves state-of-the-art performance.

## 1 Introduction

With the development of face forgery methods [1, 2, 10, 22, 23, 25, 59], an enormous amount of fake videos (a.k.a deepfakes) have raised non-neglectable concerns on privacy preservation and information security. To this end, researches have been devoted to the reliable tagging of deepfakes in order to block the propagation of malicious information. However, it is still an open problem due to the limited generalization of detection methods and the continuous advances in deepfake creation.

Earlier studies [6, 31, 3, 13, 14, 55, 41, 61] devote efforts to enhance general Convolutional Neural Networks (CNN) for identifying clear semantic distortions in deepfakes. Recent methods pay attention to the temporal inconsistency problem [5, 30, 20, 50, 45, 52, 44, 4], yet most of them fall into semantic motion understanding (e. g., detecting abnormal eye blinking, phoneme-viseme mismatches, aberrant landmark fluctuation). These methods are able to learn specific forgery patterns, however, the remarkable visual forgery cues are expected to be gradually eliminated during the continuous army race between forgers and detectors. As a result, the *generalizability* of previous methods is typically unsatisfactory when encountering deepfakes generated by unseen techniques.

On the other hand, the importance of low-level information is identified for tackling the *generalization* problem. A group of studies are developed using hand-made low-level filters (e. g., DCT [40, 19, 26], steganalysis features [58], SRM [35]) to better capture subtle differences between generated textures and the natural ones. However, methods depending on recognizable low-level patterns would become less effective on degraded data with commonly applied post-processing procedures like visual compression [34, 20, 57], which indicates the lack of *robustness*.

As a result, we rethink the appropriate representation that can ensure both *generalizability* and *robustness* in deepfake detection. We take the inspiration from Liu et al. [34], which indicates

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

that deepfakes can be directly distinguished from pieces of skin patches. Containing little uniform semantics, networks are forced to look into the low-level details within local patches. Such practice prevents network from overfitting to global semantic cues, making learned patterns more *generalizable*. In addition, as the creation of deepfakes inevitably relies on frame re-assembling, the substantial temporal differences also arise locally during the *independent local modifications* of forged frames. The underlying temporal patterns are less affected by spatial interference, thus, such local temporal inconsistency will not be easily erased during common perturbations, making the low-level modeling more *robust*.

Motivated by the observations above, in this work, we propose the **L**ocal- & **T**emporal-aware **T**ransformer-based Deepfake **D**etection (**LTTD**) framework, which particularly focuses on patch sequence modeling in deepfake detection with Transformer [17]. Detailedly, we divide the 3D video information spatially into independent local regions (as shown in Fig. 1). Encouraged by the recent success of vision transformer [17], we formulate the patch sequence modeling problem in a self-attention style and propose a *Local Sequence Transformer (LST)* module, which operates on sequences of local patches. Benefited from the attention mechanism, LST is not constrained by the receptive field in CNNs, allowing for better learning of both long and short span temporal patterns. In addition, we hierarchically inject low-level results from shallow 3D convolutions after self-attention layers to enhance the low-level feature learning at multiple scales. Such design particularly emphasizes on the low-level information from a temporal consistency perspective. After modeling each sequence of local patches in LST, two questions remain to be solved: 1) how to model their inherent relationships and 2) how to aggregate their information for final deepfake prediction. We explicitly impose global contrastive supervision on patch embeddings with a *Cross-Patch Inconsistency (CPI)* loss. Then the final predictions is given in the *Global Contrastive Classification (GCC)* module with follow up Transformer blocks. Ablations on these designs show non-trivial improvements.

Our contributions can be summarized as follows: **1)** We propose to address deepfake detection from the perspective of low-level temporal pattern learning, which enjoys better generalizability and robustness. **2)** We propose the Local- & Temporal-aware Transformer-based Deepfake Detection (LTTD) framework, which adopts a local-to-global learning protocol with a focus on the temporal information within patch sequences. **3)** Quantitative experiments show that our approach achieves the state-of-the-art generalizability and robustness. Qualitative results further illustrate the interpretability of our approach.

## 2 Related work

### 2.1 Deepfake detection

In recent years, we witness great progress in deepfake detection, where numerous forgery spotting models are proposed successively to address the practical demands of the application. In the earlier stage, methods [6, 31, 3, 14, 41, 61] are built with a major emphasis on spotting semantic visual artifacts with sophisticated model designs. Dang et al. [13] design a segmentation task optimized with the classification backbone simultaneously to predict the forgery regions. Zhao et al. [55] take the deepfake detection task as a fine-grained image classification problem, and enforce the model to attend to widespread potential forgery regions. While these methods achieve satisfied in-dataset results, the cross-dataset evaluation shows poor generalizability.

Nevertheless, generalizability is considered to be the first point should it be designed for practical application scenarios, since we never have a chance to foresee the attackers' movements. More works [48, 9, 43] begin to notice the vital problem. Wang et al. [48] argue that the lack of generalizability is due to overfitting to significant semantic visual artifacts, and propose a dynamic data argumentation schema to relieve the issue. But considering the CNN always takes the downsampled semantic representation for final classification, their method tends to only focus on more semantic visual artifacts. Another group of works [28, 9, 56, 40, 19, 26, 35, 33, 34, 8] dig deeper into the fundamental differences of deepfakes from the generation process. With the expectation that forgery methods will gradually improve, those works propose to identify deepfakes from low-level image features rather than semantic visual clues, as the latter are disappearing in the latest deepfakes. [28, 9, 56] both notice the content-independent low-level features that can uniquely identify their sources and the identity swapping will destroy the origin consistency. Li et al. [28] propose to detect those low-level features across facial boundary and Zhao et al. [56] and Chen et al. [9] turn to learn the spatial-local

inconsistencies. Other methods look for clues in the frequency domain. With DCT transform, [40, 19, 26] fuse the low-level frequency pattern learning into CNN to improve the generalizability. Liu et al. [33] instead theoretically analyze that phase information are more sensitive to upsampling, therefore, such low-level feature is more crucial than high-level semantic information for our task. The finds from [34] further encourage us to detect deepfakes from local patches, where the generated images are easy to be isolated even only one small skin patch is given. However, there is still a vital issue that low-level features should be even less robust to real-world distortions. Experiments in [34] show that simple smoothing could impair the performance of more than 20%. Drastic performance drops of [28] on suspected videos with degradation also indicate the weakness of low-level pattern learning. Different from current works, we propose to rely on temporal low-level changes with learnable shallow filters to better cope with real-world degradations.

Although temporal methods [5, 30, 20, 50, 44, 57] are also studied, most of they fall into the visual anomalous pattern learning such abnormal eye blinking [30], non-synchronized lip movements [5, 20], inconsistent facial movements [57], etc. In the foreseeable further or even good cases of current arts, we could hardly find these significant patterns. In contrast, we explore the substantial temporal inconsistency of independently frame-wise generation at local patches, which is more generalizable and also more robust to common perturbations, as the underlying temporal patterns are less affected.

### 2.2 Vision transformer

Vaswani et al. [47] first propose to using only self-attention, multilayer perceptron (MLP), and layer norm to establish a new canonical form, coined transformer, for natural language processing (NLP). Promoted by the great advancement in NLP, researchers of vision communities also start to explore the potential of transformer designs. ViT [17] could be the first success to apply pure typical transformer in image classification. After that, many researches also extend transformer into different vision tasks, (e. g., semantic segmentation [18], action recognition [39, 29], video understanding [7, 54], scene graph generation [12], object detection [60, 51, 37]) and achieve exceptional performance.

Recently, transformer structure is also adopted by a few works [57, 16, 49] in deepfake detection. [49] stack multiple ViTs [17] with different spatial embedding size. With the motivation to learn identity, transformer with extra token is proposed in [16]. Besides, The closest work to us, [57] adopt few self-attention layers to aggregate successive frames' features, which are completely downsampled semantic embeddings. While in our method, transformer is introduced to achieve patch-sequential temporal learning in a restricted spatial receptive field with a totally different purpose to identify low-level temporal inconsistency.

## 3 Approach

In this section, we will elaborate our proposed **L**ocal- & **T**emporal-aware **T**ransformer-based Deepfake **D**etection (**LTTD**) framework after formally define the problem in section 3.1. We describe Local Sequence Transformer (**LST**) in section 3.2. Then, the Cross-Patch Inconsistency (**CPI**) loss is introduced with the Global Contrastive Classification (**GCC**) in section 3.3.

### 3.1 Problem statement

Given one image $\mathbf{x} \in \mathbb{R}^{C\times H\times W}$, we spatially split it into local patches $\{\mathbf{x}^i \in \mathbb{R}^{C\cdot P^2} | i = 1, 2, ..., N\}$, where $C$ is the number of image channels, $P$ is the local patch size, and $N = HW/P^2$ is the patch number. In the original setting of ViT [17], the separated patches can be directly sent into the Transformer blocks for semantic understanding. Differently, in our task, we take a video clip $\mathbf{v} \in \mathbb{R}^{C\times T\times H\times W}$ as input, where the extra $T$ indicates the clip length, i. e., with $T$ successive frames. We also split each frame into independent patches for better low-level forgery pattern learning, but with an additional temporal dimension. The patch set is represented as $\mathbf{s} = \{\mathbf{x}^{t,i} \in \mathbb{R}^{C\cdot P^2} | t = 1, 2, ..., T; i = 1, 2, ..., N\}$, where $\mathbf{x}^{t,i}$ stands for the image patch at the $i$-th spatial region of the $t$-th frame. Thus the total patch number becomes $T \cdot N$. After that, our proposed LTTD leverages the patch locality with simple learnable filters and powerful self-attention operations to explore the low-level temporal inconsistency of deepfakes, and finally give a prediction (fake or real) based on global contrast across the whole spatial region.

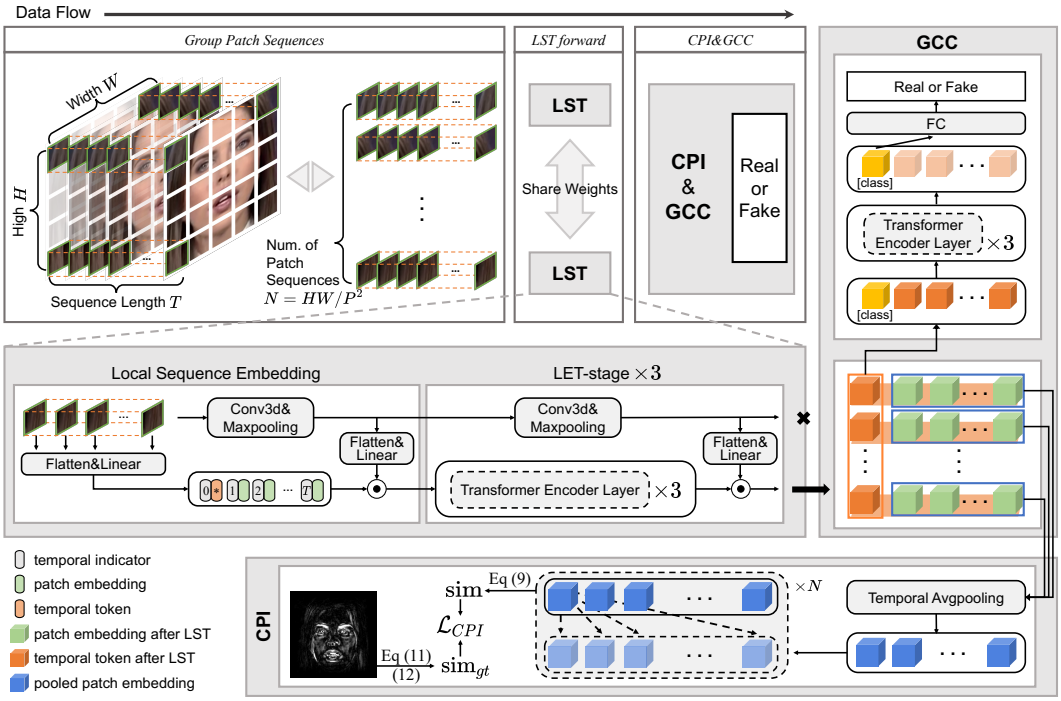


Figure 1: The overall pipeline and details of the proposed LTTD. **Top left**: we divide the whole process into three cascaded parts as *Group Patch Sequences*, *LST forward*, and *CPI&GCC*. **Others**: we illustrate the details of three main components (LST, CPI, and GCC) of our approach.

### 3.2 Local Sequence Transformer

As we discussed previously, in order to learn the low-level temporal patterns, we feed the local patches at the same spatial position into the proposed Local Sequence Transformer (LST) for further temporal encoding, i. e., the input of our LST is a patches set like $\mathbf{s}^i = \{\mathbf{x}^{t,i} \in \mathbb{R}^{C \cdot P^2} | t = 1, 2, ..., T\}$, where $i \in \{1, 2, ..., N\}$, and $\mathbf{s}^i$ includes frame patches at the spatial location of $i$.

We show the details of the LST in the middle of Fig. 1, which is divided into two parts: the Local Sequence Embedding and the Low-level Enhancement Transformer stages. In both parts, low-level temporal enhancement is consistently introduced by shallow 3D convolutions given three intuitions: 1) Learnable shallow filters would be than hand-crafted filters [41,20,36] in capturing low-level information in complex situations. 2) Voxels at one specific spatial location may not be well aligned temporally considering camera movements. Thus 3D filters which covers 3D neighborhood structures are more suitable than 2D ones when handling this kind of situation. 3) The patch embedding in Transformers always projects the patch as a whole without more fine-grained locality emphasis. We make up for this using shallow convolutions to enhance low-level feature learning at multiple scales.

**Local Sequence Embedding**. In addition to commonly used linear patch embedding, we involve 3D convolutions at the beginning to enhance low-level temporal modeling. More formally, we define this part as follows:

$$\mathbf{zs}_0^i = [\mathbf{x}^{1,i}\mathrm{E}_s; \mathbf{x}^{2,i}\mathrm{E}_s; ...; \mathbf{x}^{T,i}\mathrm{E}_s], \quad \mathrm{E}_s \in \mathbb{R}^{(C \cdot P^2) \times D} \tag{1}$$

$$[\mathbf{y}_0^{1,i}; \mathbf{y}_0^{2,i}; ...; \mathbf{y}_0^{T,i}] = \mathrm{Maxpool}(\mathrm{Conv3d}([\mathbf{x}^{1,i}; \mathbf{x}^{2,i}; ...; \mathbf{x}^{T,i}]), k), \tag{2}$$

$$\mathbf{zt}_0^i = [\mathbf{y}_0^{1,i}\mathrm{E}_0; \mathbf{y}_0^{2,i}\mathrm{E}_0; ...; \mathbf{y}_0^{T,i}\mathrm{E}_0], \quad \mathrm{E}_0 \in \mathbb{R}^{(C_t \cdot (P/k)^2) \times D} \tag{3}$$

$$\begin{aligned} \mathbf{z}_0^i &= [\mathbf{x}_{temp}^i; \mathbf{zs}_0^i(1) \cdot \sigma(\mathbf{zt}_0^i(1)); \mathbf{zs}_0^i(2) \cdot \sigma(\mathbf{zt}_0^i(2)); ...; \mathbf{zs}_0^i(T) \cdot \sigma(\mathbf{zt}_0^i(T))] + \mathrm{E}_{pos} \\ &= [\mathbf{z}_{temp}^i; \mathbf{z}^{1,i}; \mathbf{z}^{2,i}; ...; \mathbf{z}^{T,i}], \qquad \mathbf{x}_{temp}^i \in \mathbb{R}^D, \quad \mathrm{E}_{pos} \in \mathbb{R}^{(T+1) \times D} \end{aligned} \tag{4}$$

where in Eq. (1), $\mathrm{E}_s$ is a trainable linear projection with dimension of $D$, and so is $\mathrm{E}_0$ in Eq. (3), but the difference is that $\mathrm{E}_0$ acts on patches after temporal filtered in Eq. (2), i.e., the enhanced patch

representations at the initial stage $\mathbf{y}_0^{t,i}$. The $k$ in Eq. (2) denotes the pooling kernel size, which is always set to 2 for multi-granularity locality emphasis as discussed previously. The the pooling also leads to more efficient training. And $C_t$ in Eq. (3) is the number of the used temporal filters in $\mathrm{Conv3d}$, which is set to 64. Finally, the embedding is given by Eq. (4), where $\mathbf{x}_{temp}^i$ is a learnable temporal token, $\sigma$ indicates $\mathrm{sigmoid}$ function, $\mathbf{z}^{t,i} = \mathbf{zs}^i(t) \cdot \sigma(\mathbf{zt}^i(t))$ represents the patch sequence embedding of spatial location $i$ at timestamp $t$, and learnable position embedding $\mathrm{E}_{pos}$ is also kept, but with different meaning of temporal indicator.

**Low-level Enhancement Transformer stage**. Then, patch sequence embeddings at different spatial locations are independently fed into multiple Low-level Enhancement Transformer stages (LET-stages) for further temporal modeling. Similar to the 3D filters we used before, we propose to enhance the temporal modeling of patch sequence with aids of shallow spatial-temporal convolution at multiple scales. Denote multiheaded self-attention (MSA) [17], MLP, and Layernorm (LN) as :

$$\varepsilon' = \mathrm{MSA}(\mathrm{LN}(\varepsilon)) + \varepsilon, \quad \mathrm{En}(\varepsilon) = \mathrm{MLP}(\mathrm{LN}(\varepsilon')) + \varepsilon'. \tag{5}$$

Given the input embeddings at stage $l-1$ as $\mathbf{z}_{l-1}^i = [\mathbf{z}_{temp}^i; \mathbf{z}^{1,i}; \mathbf{z}^{2,i}; ...; \mathbf{z}^{T,i}]$ and the enhanced patch representations as $[\mathbf{y}_{l-1}^{1,i}; \mathbf{y}_{l-1}^{2,i}; ...; \mathbf{y}_{l-1}^{T,i}]$ , we formally define one LET-stage as:

$$\mathbf{zs}_l^i = [\mathbf{z}_{temp}^{i'}; \mathbf{z}_s^{1,i}; \mathbf{z}_s^{2,i}; ...; \mathbf{z}_s^{T,i}] = \mathrm{En}^3(\mathbf{z}_{l-1}^i), \tag{6}$$

$$[\mathbf{y}_l^{1,i}; \mathbf{y}_l^{2,i}; ...; \mathbf{y}_l^{T,i}] = \mathrm{Maxpool}(\mathrm{Conv3d}([\mathbf{y}_{l-1}^{1,i}; \mathbf{y}_{l-1}^{2,i}; ...; \mathbf{y}_{l-1}^{T,i}]), k), \tag{7}$$

$$\mathbf{zt}_l^i = [\mathbf{y}_l^{1,i}\mathrm{E}_l; \mathbf{y}_l^{2,i}\mathrm{E}_l; ...; \mathbf{y}_l^{T,i}\mathrm{E}_l], \quad \mathrm{E}_l \in \mathbb{R}^{(C_t \cdot (P/k^l)^2) \times D} \tag{8}$$

$$\begin{aligned} \mathbf{z}_l^i = \text{LET-stage}(\mathbf{z}_{l-1}^i) &= [\mathbf{z}_{temp}^{i'}; \mathbf{zs}_l^i(1) \cdot \sigma(\mathbf{zt}_l^i(1)); \mathbf{zs}_l^i(2) \cdot \sigma(\mathbf{zt}_l^i(2)); ...; \mathbf{zs}_l^i(T) \cdot \sigma(\mathbf{zt}_l^i(T))] \\ &= [\mathbf{z}_{temp}^{i'}; \mathbf{z}^{1,i'}; \mathbf{z}^{2,i'}; ...; \mathbf{z}^{T,i'}], \end{aligned} \tag{9}$$

where $\mathrm{En}$ in Eq. (5) denotes one Transformer block [17], and $\mathrm{En}^3$ in Eq. (6) indicates cascaded stacking three layers together. Overall, the proposed LST is formed by a Local Sequence Emebedding stage and three cascaded Low-level Enhancement Transformer stages.

**Discussion**. Compared with our designs, one *straight thought* might be "just leave the work to self-attention", since theoretically patches can progressively find the most relative patches at the same spatial room for temporal modeling. However, this is nearly impracticable considering both short and long span temporal information is important to our task [57] and the self-attention operation has a quadratic complexity with respect to the number of patches. In contrast, by independently modeling the patch sequences with weight-shared LST, we not only reduce the time complexity of the *straight throught* from $\mathcal{O}(T^2 \cdot N^2)$ to $\mathcal{O}(T \cdot N^2)$, but also explicitly avoid semantic modeling of features like facial structure and always focus on low-level temporal learning.

### 3.3 Cross-Patch Inconsistency loss and Global Contrastive Classification

After modeling the temporal relation of sequential patches in LST, we have a set of temporal embeddings of all spatial locations $\{\mathbf{z}^i | i = 1, 2, ..., N\}$. However, how to give a final decision is still nontrivial, since different spatial regions should represent different temporal features with different local changes, e. g., pixel variation in the background region is usually less dramatic than that of the mouth region. In addition, there normally exist real regions (usually the background) in deepfakes. Thus, directly adopting an overall binary classification loss on all patches is not the best choice. Instead, we propose to identify the inconsistency by global contrast, because forgery parts should retain heterogeneous temporal patterns compared with the real ones. Concretely, we achieve this goal through the Cross-Patch Inconsistency loss and the proposed Global Contrastive Classification.

**Cross-Patch Inconsistency loss**. Given the feature set after LST $\{\mathbf{z}^i | i = 1, 2, ..., N\}$, we leave the first temporal token at all spatial locations ($\{\mathbf{z}_{temp}^i \in \mathbb{R}^D | i = 1, 2, ..., N\}$) for latter classification, and use the remaining patch embeddings ($\{\mathbf{z}^{t,i} \in \mathbb{R}^D | t = 1, 2, ..., T; i = 1, 2, ..., N\}$) in this part. We first reduce the temporal dimension of all local regions as: $\mathbf{f}^i = \frac{1}{T}\sum_{t=1}^{T} \mathbf{z}^{t,i}$. Then we calculate the dense cosine similarities of all regions as:

$$\mathbf{sim}^{p,q} = \frac{\langle \mathbf{f}^p, \mathbf{f}^q \rangle}{\|\mathbf{f}^p\| \cdot \|\mathbf{f}^q\|}, \quad p = 1, 2, ..., N; q = 1, 2, ..., N. \tag{10}$$

The sequence of real regions will certainly depict a "natural" variation, while the sequence of fake regions formed by re-assembling will be different. According to the simplest thought that temporal features of real regions should be similar to the real ones, and vice versa. We propose to impose a intra-frame contrastive supervision as:

$$\mathcal{L}_{CPI} = \sum_{p,q} \max\left(\left|\mathbf{sim}^{p,q} - \mathbf{sim}_{gt}^{p,q}\right| - \mu, 0\right)^2, \tag{11}$$

where $\mu$ is a tolerance margin which we set it to 0.1, that allows more diverse feature representations in binary classes. And $\mathbf{sim}_{gt} \in \mathbb{R}^{N\times N}$ is the ground truth similarity matrix that generated from the modification mask sequence $\mathrm{m}_o \in \mathbb{R}^{T\times H\times W}$ as follows:

$$\mathrm{m}_\alpha = \frac{1}{T}\sum_{t=1}^{T} \mathrm{m}_o^t, \quad \mathrm{m}_\beta = \mathrm{Interpolate}(\mathrm{m}_\alpha) \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}}, \quad \mathrm{m} = \mathrm{Flatten}(\mathrm{m}_\beta) \in \mathbb{R}^N, \tag{12}$$

$$\mathbf{sim}_{gt}^{p,q} = 2\cdot(1 - |\mathrm{m}^p - \mathrm{m}^q|) - 1, \quad p = 1, 2, ..., N; q = 1, 2, ..., N. \tag{13}$$

The modification masks $\mathrm{m}_o$ are generated by simply subtracting the fake frame from the corresponding real one, which should be normalize to range of $(0, 1)$ in advance. Thus, the value range of $\mathbf{sim}_{gt}$ is consistent with cosine similarity, i.e., $(-1, 1)$. As for the real clip without mask, the $\mathbf{sim}_{gt} = \mathbf{1}^{N\times N}$.

**Global Contrastive Classification**. For the final classification, we want to determine the final prediction based on the overall consistency. We first insert a final class token $\mathbf{x}_{class} \in \mathbb{R}^D$ in to earlier LST output temporal tokens as $[\mathbf{x}_{class}; \mathbf{z}_{temp}^1; \mathbf{z}_{temp}^2; ..., \mathbf{z}_{temp}^N]$. Based on those location-specific temporal tokens, we then adopt additional three Transformer blocks (similar to Eq. (6)) to achieve cross-patch temporal information aggregation. Then the final prediction is given by a fully connection layer taking the [class] embedding as input. With such practice, our model is empowered to identify the different correlations between real and fake sequences. For the classification loss, we simply use the binary cross-entropy loss denoted as $\mathcal{L}_{BCE}$.

And finally, we can train our LTTD in an end-to-end manner with the two losses as:

$$\mathcal{L} = \mathcal{L}_{BCE} + \lambda \cdot \mathcal{L}_{CPI}, \tag{14}$$

where the parameter $\lambda$ is used to balance the two parts and empirically set to $10^{-3}$.

# 4 Experiments

## 4.1 Setups

**Datasets**. Our experiments are conducted based on several popular deepfake datasets including FaceForensics++ (**FF++**) [41], DeepFake Detection Challenge dataset (**DFDC**) [15], CelebDF-V2 (**CelebDF**) [32], FaceShift dataset (**FaceSh**) [27], and DeeperForensics dataset (**DeepFo**) [21]. FF++ (HQ) is used as train set and the remaining four datasets are used for generalization evaluation. Please find more details of these datasets in the supplementary materials.

**Data preprocessing**. All the used datasets are published in a full-frame format, thus, most of the deepfake detection methods crop the face regions in advance. However, with a motivation to learn the low-level temporal patterns, cropping face regions in advance will lead to artificially introduced jittering due to the independent face detection. Therefore, in our method, we crop the face regions using the same bounding box after randomly determine the clip range on-the-fly, where the box is detected by MTCNN [53]. We leave more details of this processing in the supplementary materials.

**Implementation details**. The spatial input size $H \times W$ and patch size $P$ is set to $224 \times 224$ and 16, respectively. It is worth noting that the division into such small patches has considerably suppress the overall semantic features. The embedding dimension $D$ is set to 384. The kernel size of Conv3d of LST is set to $3 \times 3 \times 3$. For temporal dimension $T$, we empirically set it to 16 and provide more discussion in ablations. We use only the first 128 frames of each video in the experiments, thus the final prediction is averaged from 8 clips. For optimizer, we use Adam [24] with the initial learning rate of $10^{-4}$. When the model performance no longer improves significantly, we gradually decay the learning rate until convergence. Four NVIDIA A100 GPUs are used in our experiments.

**Evaluation metrics**. Following previous works [41, 20, 28], we use binary classification accuracy (ACC) and Area Under the Receiver Operating Characteristic Curve (AUC) as evaluation metrics.

### 4.2 Generalizability evaluation

For practical application, generalizability should be one of the most concerned properties. Nevertheless, it is usually the Achilles' heel of most deepfake detectors. Since deepfakes generated by different forgery methods (in different datasets) hold different kinds of forgery cues, and overfitting on semantic visual artifacts of the train set can easily lead to cross-dataset evaluation collapse. As we show the performance of models trained on FF++ and tested on four unseen datasets in Table. 1, many methods do not perform satisfactorily. In contrast, our approach outperforms all the recently published novel detectors, and achieves a new state-of-the-art of 91.9 AUC% averaged from the four datasets. Note PatchForensics [8] also focuses on local patches, but only narrowed the perceptive field by truncating the CNN without considering the relations between patches globally, it shows limited generalizability. In addition, FTCN-TT reports comparable results, where they also leverage the self-attention mechanism. But different from our low-level temporal view, they use the semantic features of the whole frame for prediction. Thus, when the visual artifacts are less distinguishable in CelebDF and DFDC, our method surpasses it by a clear margin. A similar situation in LipForensics, which is proposed to perform deepfake detection using pretrain priors [36] of high-level semantic understanding. With temporal regularity considered, they also achieve comparable results but show suboptimal performance on CelebDF and DFDC. Moreover, PCL+I2G and Face X-ray also focus on low-level learning. However, without considering temporal properties, they exhibit insufficient robustness and perform poorly in DFDC, in which the videos are filmed under very different circumstances and have been perturbed to some extent.

Table 1: **Generalizability evaluation**. Models are trained on FF++, and test on remaining four datasets. We show the metric of video-level AUC% comparing with the state of the arts.

| Method | CelebDF | DFDC | FaceSh | DeepFo | *Average* |
|---|---|---|---|---|---|
| CNN-GRU [42] | 69.8 | 68.9 | 80.8 | 74.1 | 73.4 |
| Multi-task [38] | 75.5 | 68.1 | 66.0 | 77.7 | 71.9 |
| PatchForensics [8] | 69.6 | 65.6 | 57.8 | 81.8 | 68.7 |
| FWA [31] | 69.5 | 67.3 | 65.5 | 50.2 | 63.1 |
| Face X-ray [28] | 79.5 | 65.5 | 92.8 | 86.8 | 81.2 |
| PCL+I2G [56] | **90.0** | 67.5 | - | **99.4** | 85.6 |
| LipForensics [20] | 82.4 | 73.5 | 97.1 | 97.6 | 87.7 |
| FTCN-TT [57] | 86.9 | 74.0 | 98.8 | 98.8 | 89.6 |
| LTTD (ours) | 89.3 | **80.4** | **99.5** | 98.5 | **91.9** |

### 4.3 Robustness to perturbations

Another vital property for practical application is robustness. For network transmission, videos are always compressed. Also, consider possible attacking against the detectors, we evaluate our approach on different types of perturbed videos.

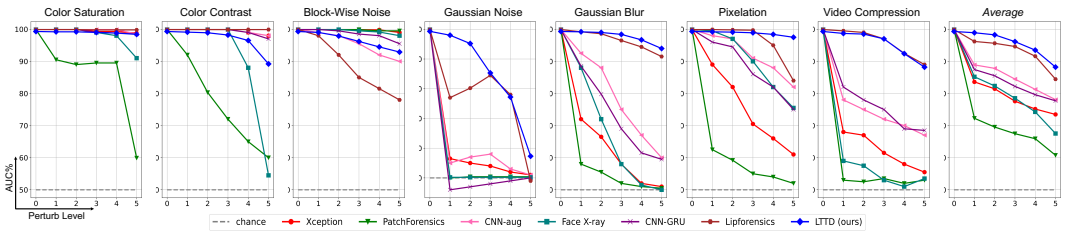


Figure 2: **Robustness evaluation**. Models are trained on the the *clean* train set of FF++ and tested on perturbed test sets, respectively. *Average* indicates the results averaged from all types of perturbations.

As previous works [20, 21], we use the script [1] on FF++ and generate seven types of perturbations at five levels. As we show the comparison diagram in Fig. 2, the seven perturb methods include color saturation change, color contrast change, block-wise noise, gaussian noise, gaussian blur, pixelation, and video compression. Examples of these perturbations can be found at this site.

Comparing with other methods, our approach shows excellent robustness against most of the perturbations. As for the type of gaussian noise at a relatively higher level, the underlying low-level patterns

[1] `https://github.com/EndlessSora/DeeperForensics-1.0/tree/master/perturbation`

are severely disturbed that our method can not work properly. We also present the precise numbers of two state-of-the-arts in Table. 2, where the results are averaged from all five levels. In general, Face X-ray detects low-level boundary artifacts, and LipForensics focuses on high-level semantic features. Thus, the former performs better in block-wise noise type and the latter shows better results in others. In contrast to Face X-ray, we detect low-level patterns in spatial-temporal space, which is better resistant to perturbations, and thus achieves better robustness evaluation performances.

Table 2: **Robustness evaluation**. Average performance evaluated on perturbed videos at five levels. Clean: origin videos, CS: color saturation, CC: color contrast, BW: block-wise noise, GNC: gaussian noise, GB: gaussian blur, PX: pixelation, VC: video compression, Avg: averaged performance on distored videos, Drop: performance drop comparing to Clean. The gray numbers do not reflect robustness, and metrics of video-level AUC% is reported.

| Method | Clean | CS | CC | BW | GNC | GB | PX | VC | *Avg/Drop* |
|---|---|---|---|---|---|---|---|---|---|
| Face X-ray [28] | 99.8 | 97.6 | 88.5 | **99.1** | 49.8 | 63.8 | 88.6 | 55.2 | 77.5/-22.3 |
| LipForensics [20] | 99.9 | **99.9** | **99.6** | 87.4 | 73.8 | 96.1 | 95.6 | **95.6** | 92.6/-7.3 |
| LTTD (ours) | 99.4 | 98.9 | 96.4 | 96.1 | **82.6** | **97.5** | **98.6** | 95.0 | **95.0/-4.3** |

### 4.4 Ablations

**Module effects**. We first compare our method with related baselines and several alternative designs. 1) **Xception** is the commonly used backbone in deepfake detection; 2) **ViT** is the most famous vision transformer backbone, where we use the "small" version with embedding dimension of 384; 3) **LTTD w/o LST** indicates the model that replace the proposed LST with commonly used Patch Embedding and Transformer blocks [17]; 4) **LTTD w/o CPI** is our $L^3$TD framework trained without using $\mathcal{L}_{CPI}$ (Eq. (13)); 5) **LTTD w/o GCC** represents the model that we replace the GCC module with a simple fully connected classification layer after average pooling the temporal embeddings from all spacial locations. From Table. 3, all the models achieve nearly perfect results on FF++. While in the cross-dataset setting, our model with elaborately designed modules exhibits much stronger generalizabilty, and the three components consistently boost the best results.

Table 3: **Module effect**. Models are trained on FF++. Gray numbers reflect in-dataset effectiveness, and others represent cross-dataset generalization. *Cross-Avg*: average from unseen datasets.

| Method | FF++ | | FaceSh | | DFDC | | DeepFo | | *Cross-Avg* | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ACC% | AUC% | ACC% | AUC% | ACC% | AUC% | ACC% | AUC% | ACC% | AUC% |
| Xception [11] | 96.08 | 99.38 | 72.47 | 78.60 | 60.47 | 67.36 | 69.21 | 83.28 | 67.38 | 76.41 |
| ViT [17] | 95.00 | 97.92 | 62.86 | 65.56 | 64.81 | 72.89 | 71.85 | 83.24 | 66.51 | 73.90 |
| LTTD w/o LST | 95.57 | 98.57 | 90.71 | 97.44 | 60.40 | 70.06 | 87.68 | 97.94 | 79.60 | 88.48 |
| LTTD w/o CPI | 97.29 | 99.23 | 95.00 | 98.86 | 67.95 | 77.03 | 90.62 | 97.68 | 84.52 | 91.19 |
| LTTD w/o GCC | 96.14 | 99.00 | 91.79 | 99.35 | 65.29 | 70.64 | 87.39 | 96.62 | 81.49 | 88.87 |
| LTTD | 97.72 | 99.52 | 96.55 | 99.51 | 71.34 | 80.39 | 92.53 | 98.50 | 86.81 | 92.80 |

**Visualization**. To intuitively demonstrate the reason that our approach is more generalizable, we compare the feature representations before the final classification in Fig. 3. Models are trained on FF++ and tested on four subsets of FF++ (Deepfakes, Face2Face, FaceSwap, and NeuralTextures) respectively. Due to the abundant inductive bias of the convolution, Xception clearly split the four kinds of deepfakes into four clusters, even **only binary labels are used** in training. This shows that the strong CNN tends to overfit on method-specific artifacts generated by different deepfake methods, although plausible results are achieved in in-dataset testing, it is harmful to generalize to unseen deepfakes. In addition, similar phenomena can be found in the ViT results, but due to less inductive bias introduced, the four clusters are not as clearly divided as Xception. In contrast, the features of our approach show a completely different outline, where the real videos are also clearly separated, but the remaining four types of deepfakes are compacted in a unified manifold. We attribute this phenomenon to our low-level temporal learning, which can distinguish deepfakes from real depending on more fundamental low-level temporal inconsistencies, thus achieving the best generalizability.

**Sequential input**. As we mentioned previously, we empirically set the clip length to 16. On the one hand, too short a clip does not ensure enough temporal information, e. g., 4 consecutive frames are

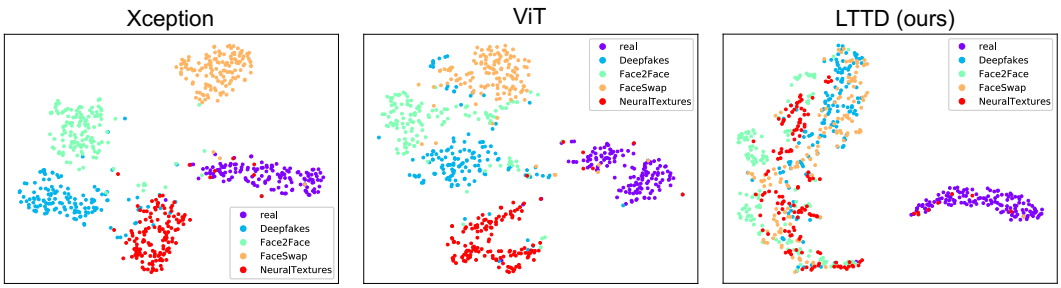


Figure 3: **Feature visualization**. In the t-SNE [46] visualization, every dot represents a compacted features of the corresponding test video, and different color indicates different class.

almost identical to each other. On the other hand, too long a clip is not necessarily better: 1) A longer clip costs more memory, and a smaller batch size also slow down the training convergence, or worse. 2) We found some of the videos in the test datasets contained scene switching. Considering the FPS of 24, a video clip in our experiments will not last 1 second, thus greatly avoiding the scene-switching problem. Here, we further analyze how clip sampling will affect the low-level temporal learning of the proposed LTTD. Two hyper-parameters are investigated: clip length and frame sampling space.

Table 4: **Sequential input ablation**. CL: clip length, FSS: frame sampling space.

| CL | FSS | FF++ | DFDC | DeepFo | *Average* |
|---|---|---|---|---|---|
| 8 | 1 | 99.26 | 77.25 | 96.82 | 91.11 |
| 16 | 1 | 99.52 | 80.39 | 98.50 | 92.80 |
| 32 | 1 | 99.47 | 80.92 | 98.41 | 92.93 |
| 64 | 1 | 99.38 | 79.23 | 98.49 | 92.37 |
| 16 | 2 | 99.15 | 76.17 | 97.32 | 90.88 |
| 16 | 4 | 98.51 | 73.02 | 91.33 | 87.62 |

From Table 4, the first 4 rows show that clip length has little effect on the results. We believe this is closely related to the idea of low-level temporal learning, which does not require the video clip to last long enough (e. g., 3 seconds) but only adequate frames to extract the low-level temporal patterns. Sparse sampling is another practice to aggregate more content in a single clip. When we expand the frame interval, the performance degrades considerably. This phenomenon suggests that more sparse sampling is detrimental to the learning of low-level temporal patterns even when more motion content is included. These findings demonstrate that LTTD is distinctly different from related temporal-based models considering the low-level temporal learning is specially designed for deepfake detection.

## 5 Conclusion and discussion

**Conclusion**. In this paper, we propose a reliable framework to address the practical problem of deepfake detection, which emphasizes on the low-level temporal patterns of sequential patches in the restricted spatial region, but with a whole-range temporal receptive field using Transformer blocks. In addition, we make the final classification in a more general global-contrastive way. Thus better generalizability and robustness are achieved to better support deepfake detection in real-world scenes. Moreover, qualitative results further verify that low-level temporal information can lead to stronger generalizability, which could also be a guideline for developing better approaches in the future.

**Limitation**. Considering the continuous advances in deepfake creation and adversarial training, the performance of our approach when encountering low-level and temporal adversarially enhanced deepfakes in the future is yet unclear.

Moreover, although our method shows favorable performance compared to recent works, it still requires intensive labor in order to handle real-world scenarios. This is a commonly shared limitation that we do not know if the detectors are *calibrated well* for real-world deployment. In addition to identifying deepfakes, how we can ensure the predictions are credible remains an open problem, hindering the application of all deepfake detectors.

**Ethical Concern**. Our work is proposed to mitigate the potential negative social impacts of other works, but there is still a risk of facilitating the forgers to propose better deepfake methods through adversarial training.

# References


[1] Deepfakes github. `https://github.com/deepfakes/faceswap`, 2022. Accessed: 2022-02-28.

[2] Deepfakes github. `https://github.com/MarekKowalski/FaceSwap`, 2022. Accessed: 2022-02-28.

[3] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7. IEEE, 2018.

[4] Shruti Agarwal and Hany Farid. Detecting deep-fake videos from aural and oral dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 981–989, 2021.

[5] Shruti Agarwal, Hany Farid, Tarek El-Gaaly, and Ser-Nam Lim. Detecting deep-fake videos from appearance and behavior. In *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2020.

[6] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and Alberto Del Bimbo. Deepfake video detection through optical flow based cnn. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019.

[7] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021.

[8] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *European Conference on Computer Vision*, pages 103–120. Springer, 2020.

[9] Shen Chen, Taiping Yao, Yang Chen, Shouhong Ding, Jilin Li, and Rongrong Ji. Local relation learning for face forgery detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1081–1088, 2021.

[10] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[11] 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.

[12] Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-temporal transformer for dynamic scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16372–16382, 2021.

[13] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K Jain. On the detection of digital face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5781–5790, 2020.

[14] Sowmen Das, Selim Seferbekov, Arup Datta, Md Islam, Md Amin, et al. Towards solving the deepfake problem: An analysis on improving deepfake detection using dynamic face augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3776–3785, 2021.

[15] 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.

[16] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Ting Zhang, Weiming Zhang, Nenghai Yu, Dong Chen, Fang Wen, and Baining Guo. Protecting celebrities with identity consistency transformer. *arXiv preprint arXiv:2203.01318*, 2022.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[18] Brendan Duke, Abdalla Ahmed, Christian Wolf, Parham Aarabi, and Graham W Taylor. Sstvos: Sparse spatiotemporal transformers for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5912–5921, 2021.

[19] Qiqi Gu, Shen Chen, Taiping Yao, Yang Chen, Shouhong Ding, and Ran Yi. Exploiting fine-grained face forgery clues via progressive enhancement learning. *arXiv preprint arXiv:2112.13977*, 2021.

[20] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Lips don't lie: A generalisable and robust approach to face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5039–5049, 2021.

[21] 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.

[22] 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.

[23] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

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

[25] Mohammad Rami Koujan, Michail Christos Doukas, Anastasios Roussos, and Stefanos Zafeiriou. Head2head: Video-based neural head synthesis. *arXiv preprint arXiv:2005.10954*, 2020.

[26] Jiaming Li, Hongtao Xie, Jiahong Li, Zhongyuan Wang, and Yongdong Zhang. Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In *Proceedings of the*

*IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6458–6467, 2021.

[27] 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.

[28] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5001–5010, 2020.

[29] Xiangyu Li, Yonghong Hou, Pichao Wang, Zhimin Gao, Mingliang Xu, and Wanqing Li. Trear: Transformer-based rgb-d egocentric action recognition. *IEEE Transactions on Cognitive and Developmental Systems*, 2021.

[30] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7. IEEE, 2018.

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

[32] 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.

[33] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 772–781, 2021.

[34] Zhengzhe Liu, Xiaojuan Qi, and Philip HS Torr. Global texture enhancement for fake face detection in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8060–8069, 2020.

[35] 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*, pages 16317–16326, 2021.

[36] Pingchuan Ma, Brais Martinez, Stavros Petridis, and Maja Pantic. Towards practical lipreading with distilled and efficient models. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7608–7612. IEEE, 2021.

[37] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3164–3173, 2021.

[38] Huy H Nguyen, Fuming Fang, Junichi Yamagishi, and Isao Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. In *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8. IEEE, 2019.

[39] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. Spatial temporal transformer network for skeleton-based action recognition. In *International Conference on Pattern Recognition*, pages 694–701. Springer, 2021.

[40] 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.

[41] 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.

[42] 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.

[43] Ke Sun, Taiping Yao, Shen Chen, Shouhong Ding, Rongrong Ji, et al. Dual contrastive learning for general face forgery detection. *arXiv preprint arXiv:2112.13522*, 2021.

[44] Zekun Sun, Yujie Han, Zeyu Hua, Na Ruan, and Weijia Jia. Improving the efficiency and robustness of deepfakes detection through precise geometric features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3609–3618, 2021.

[45] Shahroz Tariq, Sangyup Lee, and Simon S Woo. A convolutional lstm based residual network for deepfake video detection. *arXiv preprint arXiv:2009.07480*, 2020.

[46] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[48] Chengrui Wang and Weihong Deng. Representative forgery mining for fake face detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14923–14932, 2021.

[49] Junke Wang, Zuxuan Wu, Jingjing Chen, and Yu-Gang Jiang. M2tr: Multi-modal multi-scale transformers for deepfake detection. *arXiv preprint arXiv:2104.09770*, 2021.

[50] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265. IEEE, 2019.

[51] Junbo Yin, Jianbing Shen, Chenye Guan, Dingfu Zhou, and Ruigang Yang. Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11495–11504, 2020.
[52] Daichi Zhang, Chenyu Li, Fanzhao Lin, Dan Zeng, and Shiming Ge. Detecting deepfake videos with temporal dropout 3dcnn. IJCAI, 2021.
[53] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.
[54] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13577–13587, 2021.
[55] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. *arXiv preprint arXiv:2103.02406*, 2021.
[56] Tianchen Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. Learning self-consistency for deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15023–15033, 2021.
[57] 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.
[58] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Two-stream neural networks for tampered face detection. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pages 1831–1839. IEEE, 2017.
[59] Hao Zhu, Chaoyou Fu, Qianyi Wu, Wayne Wu, Chen Qian, and Ran He. Aot: Appearance optimal transport based identity swapping for forgery detection. *Advances in Neural Information Processing Systems*, 33:21699–21712, 2020.
[60] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.
[61] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. Wilddeepfake: A challenging real-world dataset for deepfake detection. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2382–2390, 2020.

# Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] The FF++ dataset and FaceSh dataset are released under the FaceForensics Terms of Use. The CelebDF dataset is released under the Terms to Use Celeb-DF. The use of DeepFo dataset is bounded by the Terms of Use: DeeperForensics-1.0 Dataset.

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The datasets we use are face-related open-source datasets that contain personally identifiable information.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]