Texture, Shape and Order Matter: A New Transformer Design for Sequential DeepFake Detection

Yunfei Li[1], Yuezun Li[1,*], Xin Wang[2], Baoyuan Wu[3], Jiaran Zhou[1], Junyu Dong[1]
[1] School of Computer Science and Technology, Ocean University of China
[2] University at Albany, SUNY
[3] School of Data Science, The Chinese University of Hong Kong, Shenzhen, China

Abstract

Sequential DeepFake detection is an emerging task that predicts the manipulation sequence in order. Existing methods typically formulate it as an image-to-sequence problem, employing conventional Transformer architectures. However, these methods lack dedicated design and consequently result in limited performance. As such, this paper describes a new Transformer design, called TSOM, by exploring three perspectives: Texture, Shape, and Order of Manipulations. Our method features four major improvements: ❶ we describe a new texture-aware branch that effectively captures subtle manipulation traces with a Diversiform Pixel Difference Attention module. ❷ Then we introduce a Multi-source Cross-attention module to seek deep correlations among spatial and sequential features, enabling effective modeling of complex manipulation traces. ❸ To further enhance the cross-attention, we describe a Shape-guided Gaussian mapping strategy, providing initial priors of the manipulation shape. ❹ Finally, observing that the subsequent manipulation in a sequence may influence traces left in the preceding one, we intriguingly invert the prediction order from forward to backward, leading to notable gains as expected. Extensive experimental results demonstrate that our method outperforms others by a large margin, highlighting the superiority of our method.

1. Introduction

Recent advancements in deep generative techniques have significantly improved the visual quality of generated faces. One such technique, DeepFake, allows for effortless manipulation of faces, with results that are often imperceptible to the naked eye. This technique has found applications in virtual reality and movie special effects, but it also raises severe security concerns [9]. To address these concerns, various forensic methods have been proposed to expose DeepFakes, garnering considerable attention [14, 15, 34].

Conventional forensics methods mainly focus on determining whether a given face image is authentic or forged, which typically falls into the binary classification task (see Fig. 1(a)). However, with the rapid evolution of face editing tools, the manipulation process has become more versatile and user-friendly, enabling the use of multiple specific operations iteratively until the desired effects are reached. In this context, existing methods can only identify the final step of manipulation, providing limited insights into the manipulation details. To overcome this limitation, a new forensics task called **Sequential DeepFake Detection (SDD)** [23] has emerged recently. In contrast to the conventional DeepFake detection (*i.e.*, One-step DeepFake Detection (ODD)), SDD attempts to uncover the sequences of face manipulations in order. For example, users might create a DeepFake face by sequentially editing regions like hair, eyes, and lips. Given such a DeepFake face, SDD can predict the manipulation sequence as "Hair-Eye-Lip" (see Fig. 1(b)). This emerging task improves the practicality of DeepFake detection in real-world scenarios, offering more comprehensive evidence for traceability and interoperability (*e.g.*, pinpointing where, when, and what manipulations are applied). This endeavor expands the horizon of DeepFake detection and poses a fresh challenge to the field.

In general, the task of SDD can be conceptualized as an image-to-sequence problem [27], where the typical solution involves leveraging Transformer-based architectures to predict the manipulation sequences [23, 31]. Concretely, a spatial encoder with conventional self-attention mechanisms is utilized to extract spatial manipulation traces. Then, a sequential decoder is employed to establish conventional cross-attention [26] with the encoder features and manipulation annotation sequence, facilitating the capture of sequential manipulation traces (see Fig. 1(c)).

Despite these methods showing certain effectiveness, many critical perspectives are lacking in further exploration: **1) Neglecting texture details:** Generative models often exhibit imperfect semantic disentanglement, wherein one

*Corresponds to Yuezun Li (liyuezun@ouc.edu.cn)

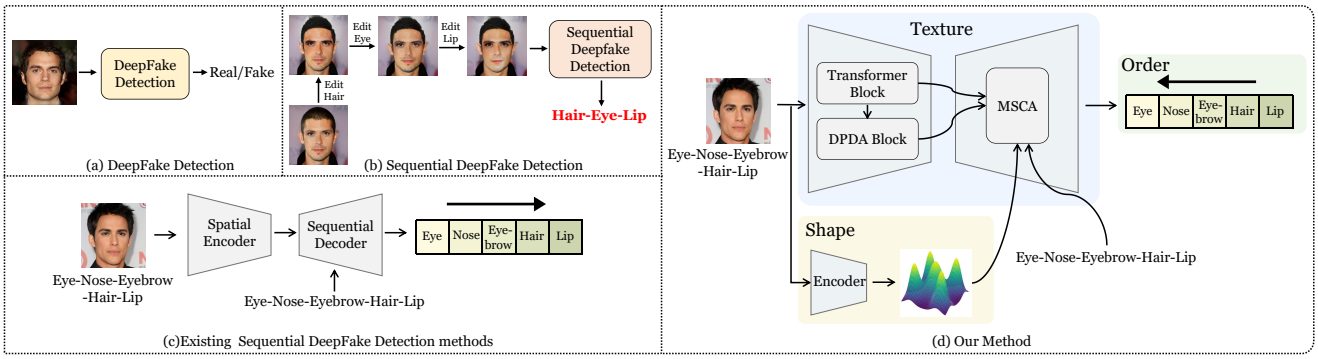

Figure 1. (a) and (b) correspond to One-step DeepFake Detection (ODD) and sequential DeepFake Detection (SDD). (c) illustrates the process of typical Sequential DeepFake Detection (SDD) architecture. (d) represents our proposed method TSOM. In contrast to (c), our method focuses on exploring three perspectives: Texture, Shape, and Order of Manipulations, and introduces four major improvements: a Diversiform Pixel Difference Attention (DPDA) module, a Multi-source Cross-attention (MSCA) module, a Shape-guided Gaussian Mapping (SGM) strategy and an Inverted Order Prediction (IOP).

specific manipulation may affect the entire facial structure, leaving subtle traces on unintended manipulated regions. These subtle traces often manifest more prominently in the textures [17, 33]. However, the existing methods rely on conventional self-attention mechanisms that concentrate more on global correlations, thus overlooking fine-grained features like textures. **2) Lack of manipulation shape guidance:** The existing pipeline accomplishes detection from a holistic perspective, directly predicting the manipulations without considering their specific shapes. Note that the manipulation regions often exhibit various shapes and irregularities. Incorporating knowledge about manipulation shapes can guide the network to gain a deeper understanding of the manipulations. While existing method [23] incorporates the scale and position of manipulation regions, they could not effectively characterize irregular manipulation regions. **3) Improper prediction order:** Existing methods predict the manipulation sequences in a forward order, the same as the actual manipulation order. However, the subsequent manipulation usually affects the appearance of the preceding ones (see Fig. 2)), *i.e.*, the preceding manipulation "hides beneath" the subsequent one, akin to the layers of an "onion". The first manipulation step (*e.g.*, Hair as shown in Fig. 1(b)) acts as the core of the onion. Extracting this core without peeling away its outer layers (*e.g.*, Eye and lip as shown in Fig. 1(b)) would be improper.

Delving into these perspectives, we propose a new Transformer called **TSOM** (Texture, Shape and Order of Manipulations) to expose sequential DeepFakes (see Fig. 1(d)). Our method introduces four major improvements: ❶ In the encoder, we develop two branches for capturing different types of manipulation traces. One branch, termed the vanilla branch, employs conventional Transformer blocks to capture global relationships. The other branch, a new texture-aware branch, is designed to capture subtle manipulation traces. Within this branch, we describe

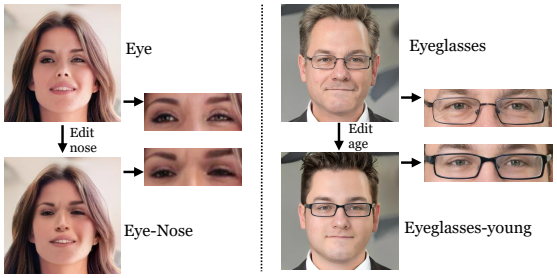

Figure 2. The impact of the sequential manipulation on facial attributes. The second manipulation impacts the first manipulation: (left) Eye and (right) Eyeglasses.

a Diversiform Pixel Difference Attention (DPDA) module, which dynamically integrates various texture extraction operations into the self-attention mechanism, instructed by the knowledge from the vanilla branch. ❷ Then we propose a new Multi-source Cross-attention (MSCA) module to effectively model correlations between spatial features from the encoder and sequential manipulation annotations. Unlike traditional cross-attention, our module processes multiple features simultaneously and employs a multi-level fusion strategy for more effective correlations. ❸ To further enhance cross-attention, we propose a Shape-guided Gaussian Mapping (SGM) strategy to provide priors on manipulation shapes. These priors are estimated using a Variational Autoencoder (VAE) [11]. ❹ During training, we invert the order of the manipulation sequences. This simple "trick" notably improves the detection performance, as expected.

Our method is evaluated on a sequential DeepFake dataset [23] and compared with several state-of-the-art methods. The results demonstrate its superiority in detecting sequential DeepFakes. Furthermore, we conduct a comprehensive analysis of each component within our method, corroborating the efficacy of the proposed strategies.

2. Related Works

One-step DeepFake Detection (ODD). To combat the threats posed by DeepFake, a considerable amount of methods have been proposed for DeepFake detection [3, 6, 7, 14, 30, 32, 34]. These methods can typically be classified into different categories in terms of the way of feature capturing. The methods of [1, 14, 20, 34, 35] expose DeepFake by capturing abnormal biological signals, such as eye blinking, behavior patterns, or heart rhythm. Other methods focus on extracting the spatial artifacts introduced either by face generation or the blending operations [13, 15, 24, 40]. Instead of capturing the spatial artifacts, the methods of [4, 12, 16] demonstrate the effectiveness of frequency information and capture representative traces in the frequency domain. Moreover, many recent methods design dedicated models to capture the traces automatically [7, 30, 38]. However, these methods can only predict authenticity, which can not handle and analyze the multi-step alterations in practice.

Sequential DeepFake Detection (SDD). Sequential DeepFake detection is an emerging task focused on predicting the sequential manipulations in order. This task is typically formulated as the image-to-sequence problem, utilizing architectures tailored for sequential tasks in detection. The work of [23] is pioneering in addressing this problem, employing the Transformer architecture [26], which includes a conventional encoder and decoder with vanilla self-attention mechanisms. Subsequently, MMNet [31] improved its performance through the incorporation of multi-collaboration and multi-supervision modules. Despite these promising results, this task remains largely unexplored.

3. Method

Our method is developed on a Transformer-based architecture comprising a spatial encoder and a sequential decoder. The encoder is designed to extract spatial manipulation features, including a vanilla branch and a texture-aware branch. The vanilla branch employs several conventional Transformer blocks to capture the global correlations. In the texture-aware branch, we introduce the **Diversiform Pixel Difference Attention (DPDA)** module to capture subtle manipulation traces (Sec. 3.1). The decoder models the sequential relation based on the spatial features and corresponding sequence annotations to predict the facial manipulation sequence. In this decoder, we propose a **Multi-source Cross-attention (MSCA)** strategy to fuse the extracted spatial features with the sequence annotation embeddings, enabling effective modeling of sequential relations (Sec. 3.2). Moreover, we introduce **Shape-guided Gaussian Mapping (SGM)** to enhance the effect of cross-attention in sequential modeling (Sec. 3.3). We then introduce the procedure for **Inverted Order Prediction (IOP)** (Sec. 3.4) and outline the overall loss functions (Sec. 3.5).

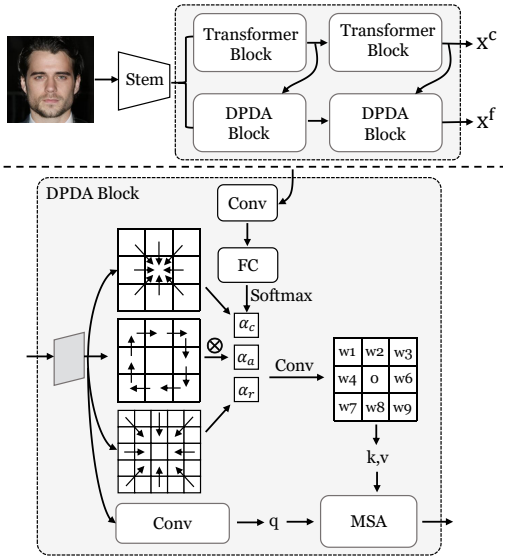

Figure 3. Overview of the spatial encoder (top). Illustration of DPDA module (bottom). MSA denotes Multi-head Self-attention.

3.1. Texture-aware Branch

Denote a face image as $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{H\times W\times 3}$. We first tokenize this face image using a simple CNN stem $\mathcal{F}$ following [8] as $\mathbf{x}^0 = \mathcal{F}(\boldsymbol{\mathcal{I}}) + \mathbf{p}, \mathbf{x}^0 \in \mathbb{R}^{h\times w\times c}$, where $\mathcal{F}(\boldsymbol{\mathcal{I}})$ indicates the feature maps from CNN stem and $\mathbf{p}$ is a positional embedding. These tokens are then flattened in spatial dimension as the input of the vanilla branch and texture-aware branch in the spatial encoder.

To extract these subtle manipulation traces, we develop a new Transformer Block for extracting texture features. Our block draws inspiration from central difference operators [10, 37], which are convolutions used for detecting local textures. In our method, we derive and extend their essence from the convolution style to the self-attention mechanism, and propose a Diversiform Pixel Difference Attention (DPDA) module that captures fine-grained spatial traces while retaining the global view within the vanilla self-attention.

Revisit of Central Difference Convolution. Central difference convolution is a special case of convolution, which emphasizes the center-oriented gradients in the local receptive field region. Given a feature map $\mathbf{x}$, this operation can be formulated as

$$\mathbf{x}'(p_0) = \sum_{p_n\in\mathcal{O}} \mathbf{w}(p_n)(\mathbf{x}(p_0+p_n) - \mathbf{x}(p_0)) \tag{1}$$

where $\mathbf{w}$ is the convolution kernel, $\mathcal{O}$ denotes the local receptive field, p_0 indicates the current location of input feature map $\mathbf{x}$ and output feature map $\mathbf{x}'$, and p_n is the relative position in $\mathcal{O}$.

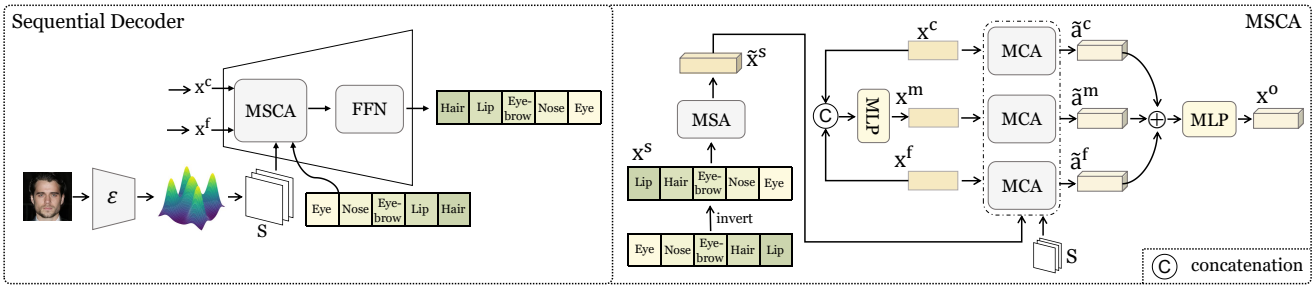

Figure 4. Overview of the sequential decoder (left). Illustration of Multi-source Cross-attention(MSCA). MCA denotes Multi-head Cross-attention.

Diversiform Pixel Difference Attention (DPDA). Inspired by the central difference operations, we extend these operations and integrate such spirit into the self-attention mechanism following [19, 36, 37]. Specifically, given the feature map $\mathbf{x}$, we first perform a vanilla convolution on it to generate the query feature, which is defined as

$$\mathbf{x}^{Q}(p_0) = \sum_{p_n \in \mathcal{O}} \mathbf{w}(p_n)\mathbf{x}(p_0 + p_n) \tag{2}$$

Then we would like to capture the fine-grained spatial features as the source of key and value features. Since the central difference operations focus on the center-oriented gradients, they may overlook other texture patterns, which hinders the effectiveness of capturing manipulation traces. To obtain a diverse range of gradient information, we leverage the concept of Extended Local Binary Pattern (ELBP), which can encode the pixel relations from various perspectives, including central, angular, and radial directions [25]. The angular direction calculates the difference between adjacent pixels in the outline of the kernel in a certain order, while the radial direction calculates the difference between radial adjacent pixels, see Fig. 3. To obtain the key and value features, we adaptively ensemble these operations using a learnable coefficient vector, which is formulated as

$$\mathbf{x}^{K,V}(p_0) = \sum_{p_n \in \mathcal{O}} \mathbf{w}(p_n) \left(\underbrace{[\alpha_c, \alpha_a, \alpha_r]}_{\boldsymbol{\alpha}} \begin{bmatrix} \mathcal{D}_c \\ \mathcal{D}_a \\ \mathcal{D}_r \end{bmatrix} \right) \tag{3}$$

In this equation, $\mathcal{D}_c, \mathcal{D}_a, \mathcal{D}_r$ denote central, angular, and radial difference operations respectively, which can be defined as

$$\begin{aligned} \mathcal{D}_c &= \mathbf{x}(p_0 + p_n) - \mathbf{x}(p_0), \\ \mathcal{D}_a &= \mathbf{x}(p_0 + p_n) - \mathbf{x}(p_{n-1}), \\ \mathcal{D}_r &= \mathbf{x}(p_0 + p_n) - \mathbf{x}(p_{\phi(n)}) \end{aligned} \tag{4}$$

We use 3×3 as the local receptive field $\mathcal{O}$ for instance and set $\mathcal{O} = \{(-1,-1),(-1,0),...,(1,1)\}$. Thus $\phi(n)$ in radial difference operation can defined as $\phi(n) = (2x, 2y)$ if $n = (x, y)$. $\boldsymbol{\alpha}$ is the coefficient vector to balance the contribution of each operation.

Adaptive Balance. Since various DeepFake faces exhibit diverse manipulation traces, the coefficient vector $\boldsymbol{\alpha}$ should be different. Thus, we design a strategy to dynamically regulate the influence of different DPDA operations. As the vanilla branch concentrates on global correlations, it has deeper insight into the manipulation composition. Hence, we set $\boldsymbol{\alpha}$ trainable and adapt it based on the knowledge from the vanilla branch. Let $\mathbf{x}'$ be the intermediate feature of a block within the vanilla branch. We develop a sub-network $\mathcal{H}$ to learn $\boldsymbol{\alpha}$ as

$$\boldsymbol{\alpha} = \text{Softmax}(\mathcal{H}(\mathbf{x}')) \tag{5}$$

After obtaining $\mathbf{x}^{Q}$ and $\mathbf{x}^{K,V}$, we project the query feature $\mathbf{x}^{Q}$ into query tokens, and project $\mathbf{x}^{K,V}$ into key and value tokens. Within each head, $Q_i = \mathbf{x}^{Q}W_i^{Q}, K_i = \mathbf{x}^{K,V}W_i^{K}, V_i = \mathbf{x}^{K,V}W_i^{V}$. Then the multi-head self-attention is calculated using

$$\begin{aligned} h_i &= \text{Softmax}(Q_i K_i^{\top}/\sqrt{d_i})V_i, \\ \text{MSA}(\mathbf{x}) &= \text{Concat}(h_1, ..., h_n)W^{o}, \end{aligned} \tag{6}$$

where h_i is the self-attention of i-th head. Then we concatenate all these heads and perform a linear transformation with W^{o} to obtain the final output.

3.2. Multi-source Cross-attention (MSCA)

Revisit of Conventional Cross-attention. Typically, the cross-attention [26] is employed for modeling such correlation [23]. However, these mechanisms usually process the features from only two sources $\mathbf{x}_1$ and $\mathbf{x}_2$ (*e.g.*, the encoder features and sequential embedding). The conventional Multi-head Cross-attention (MCA) can be defined as

$$\begin{aligned} &Q_i = \mathbf{x}_1 W_i^{Q}, K_i = \mathbf{x}_2 W_i^{K}, V_i = \mathbf{x}_2 W_i^{V}, \\ &h_i = \text{Softmax}(Q_i K_i^{\top}/\sqrt{d_i})V_i, \\ &\text{MCA}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_2) = \text{Concat}(h_1, ..., h_n)W^{o} \end{aligned} \tag{7}$$

Operations of MSCA. Let $\mathbf{x}^{c}$ and $\mathbf{x}^{f}$ represent the features from the coarse-grained and fine-grained branches. Let $\mathbf{x}^{s}$ be the embedding of inverted sequential manipulation annotations. In our task, it is intuitive to perform separate MCA operations on $\mathbf{x}^{s}$ with $\mathbf{x}^{c}$ and $\mathbf{x}^{f}$. However, this integration fails to explore deeper connections between these three features. As such, we propose a Multi-source Cross-attention

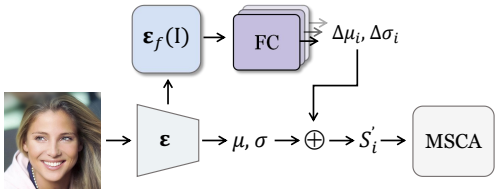

Figure 5. Integration with MSCA.

(MSCA) module to capture the correlations among multiple sources. The overview of the entire module is illustrated in Fig. 4.

Firstly, we concatenate coarse-grained features $\mathbf{x}^c$ and fine-grained features $\mathbf{x}^f$ along the channel dimension. Subsequently, we facilitate interactions between the two different levels of fabricated features throuth an MLP structure, yielding shallowly fused sequencetial features $\mathbf{x}^m$. Then, by performing self-attention computations on $\mathbf{x}^s$, we extract features that exploit correlations among annotations of sequence tampering. These operations can be formulated as follows:

$$\mathbf{x}^m = \mathrm{MLP}(\mathrm{Concat}(\mathbf{x}^c, \mathbf{x}^f)),\ \tilde{\mathbf{x}}^s = \mathrm{MSA}(\mathbf{x}^s) \tag{8}$$

Next, we implement a second-stage fusion strategy, performing multi-head cross-attention between $\tilde{\mathbf{x}}^s$ and $\mathbf{x}^c$, $\mathbf{x}^m$, $\mathbf{x}^f$ respectively, as:

$$\begin{aligned}
\tilde{\mathbf{a}}^c &= \mathrm{MCA}(\tilde{\mathbf{x}}^s, \mathbf{x}^c, \mathbf{x}^c), \mathbf{a}^c = \mathrm{LN}(\tilde{\mathbf{a}}^c + \tilde{\mathbf{x}}^s), \\
\tilde{\mathbf{a}}^m &= \mathrm{MCA}(\tilde{\mathbf{x}}^s, \mathbf{x}^m, \mathbf{x}^m), \mathbf{a}^m = \mathrm{LN}(\tilde{\mathbf{a}}^m + \tilde{\mathbf{x}}^s), \\
\tilde{\mathbf{a}}^f &= \mathrm{MCA}(\tilde{\mathbf{x}}^s, \mathbf{x}^f, \mathbf{x}^f), \mathbf{a}^f = \mathrm{LN}(\tilde{\mathbf{a}}^f + \tilde{\mathbf{x}}^s)
\end{aligned} \tag{9}$$

Finally, we directly sum up the three types of features and then pass them through an MLP to filter out irrelevant noise, thereby uncovering identical manipulation patterns and obtaining the final sequential deepfake features, following the operations as

$$\mathbf{A} = \tilde{\mathbf{a}}^c + \tilde{\mathbf{a}}^m + \tilde{\mathbf{a}}^f,\ \tilde{\mathbf{x}}^o = \mathrm{MLP}(A),\ \mathbf{x}^o = \mathrm{LN}(\tilde{\mathbf{x}}^o + \mathbf{A}) \tag{10}$$

3.3. Shape-guided Gaussian Mapping

We propose a Shape-guided Gaussian Mapping (SGM) strategy to leverage the strong spatial position priors associated with each manipulation component in the faces. Since each manipulation has its specific characteristic, its manipulation shape likely follows a certain distribution. By learning these distributions, we can predict the probability of each position on a face image being manipulated. This probability can serve as shape priors to further enhance the effectiveness of MSCA.

Distribution Estimation. Since no ground truth is provided, estimating these distributions is non-trivial. Inspired by unsupervised learning strategies, we construct a Variational Auto-encoder (VAE) [11] to estimate these distributions. Specifically, we construct an encoder $\mathcal{E}$, which transforms the input image $\mathcal{I}$ into means $\boldsymbol{\mu}$ and standards $\boldsymbol{\sigma}$ of these distributions as $\boldsymbol{\mu}, \boldsymbol{\sigma} = \mathcal{E}(\mathcal{I})$. Note that $\boldsymbol{\mu}, \boldsymbol{\sigma}$ are vectors containing mean and standard of each manipulation. To ensure these distributions align with the input image, we employ a decoder $\mathcal{D}$ for reconstruction based only on the estimated distributions.

Integration with MSCA. We create a grid map M that is composed of the x- and y- coordinates of each position. This grid map is then used to predict the probability of each position in different facial manipulations. This process can be formulated as $S = \mathcal{N}(M \mid \boldsymbol{\mu}, \boldsymbol{\sigma})$, where S is the probability map for each spatial position. Considering different heads in MSCA have different focuses. To properly integrate SGM with MSCA, we craft different probability maps $\{S_i'\}_{i=1}^h$ for each head. Specifically, we use $\boldsymbol{\mu}, \boldsymbol{\sigma}$ as the base of all heads and predict the head-specific offset $\Delta\boldsymbol{\mu}_i, \Delta\boldsymbol{\sigma}_i$ for i-th head. These offsets are obtained as $\Delta\boldsymbol{\mu}_i, \Delta\boldsymbol{\sigma}_i = \mathrm{FC}(\mathcal{E}_f(\mathcal{I}))$, where $\mathcal{E}_f(\mathcal{I})$ denotes the last feature of encoder $\mathcal{E}$. Therefore, the probability map S_i' for i-th head can be defined as

$$S_i' = \mathcal{N}(M \mid \boldsymbol{\mu} + \Delta\boldsymbol{\mu_i}, \boldsymbol{\sigma} + \Delta\boldsymbol{\sigma_i}) \tag{11}$$

This process is illustrated in Fig. 5. Then these maps are integrated into $\mathrm{MCA}(\tilde{\mathbf{x}}^s, \mathbf{x}^c, \mathbf{x}^c)$ and $\mathrm{MCA}(\tilde{\mathbf{x}}^s, \mathbf{x}^f, \mathbf{x}^f)$ operations in MSCA, by calculating the attention as

$$h_i = \mathrm{Softmax}(Q_i K_i^\top / \sqrt{d_i} + \eta S_i')V_i \tag{12}$$

3.4. Inverted Order Prediction

In the training phase, we invert the annotation of the manipulation sequence. For instance, as shown in Fig. 4, the manipulated face is annotated as "eye-nose-eyebrow-hair-lip", we rearrange it to "lip-hair-eyebrow-nose-eye" as the input of the MSCA module for sequential modeling. This strategy enables the model to first predict the final manipulation (*e.g.*, hair) and understand the consequent disturbance caused by it. Then the model concentrates on the preceding manipulation (*e.g.*, lips). The rationale behind this strategy is that the subsequent manipulation (*e.g.*, noise) may affect the preceding ones (*e.g.*, eye), making it more challenging to detect them (recall Fig. 2). In contrast, preceding manipulation (*e.g.*, hair) is not influenced by subsequent ones, making it more easily detectable. In summary, this strategy offers the advantage that, compared to forward sequence predictions, each step in the inverted sequence predictions is only influenced by itself and remains independent of other manipulations.

3.5. Objectives

In the training phase, the auto-regressive mechanism [5] is employed to model sequence relationships with the cross-entropy loss L_{ce}. This loss measures the error of the predicted manipulation sequence and ground truth. To learn the shape priors, we perform the image reconstruction loss

Table 1. Performance of different methods on detecting sequential DeepFakes.

Methods	Facial-components		Facial-attributes		Average	
	FACC	AACC	FACC	AACC	FACC	AACC
Multi-Cls (ResNet-34) [28]	69.66	50.54	66.99	46.68	68.33	48.61
Multi-Cls (ResNet-50) [28]	69.65	50.57	66.66	46.00	68.16	48.29
DETR (ResNet-34) [2]	69.87	50.63	67.93	48.15	68.90	49.39
DETR (ResNet-50) [2]	69.75	49.84	67.62	47.99	68.69	48.92
DRN [29]	66.06	45.79	64.42	43.20	65.24	44.50
DRN* [29]	69.38	49.76	67.66	47.31	68.52	48.54
MA [39]	71.31	52.94	67.58	47.48	69.45	50.21
MA* [39]	70.12	51.82	68.91	49.19	69.52	50.51
Two-Stream [18]	71.92	53.89	66.77	46.38	69.35	50.14
Two-Stream* [18]	72.00	53.97	65.20	44.39	68.60	49.18
SeqFake-Former (ResNet-34) [23]	72.13	54.80	67.99	48.32	70.06	51.56
SeqFake-Former (ResNet-34)* [23]	70.97	52.65	68.40	48.93	69.69	50.79
SeqFake-Former (ResNet-50) [23]	72.65	55.30	68.86	49.63	70.76	52.47
SeqFake-Former (ResNet-50)* [23]	69.12	50.03	67.93	48.65	68.53	49.34
MMNet (ResNet-50) [31]	73.93	56.83	69.27	50.44	71.60	53.64
TSOM (ResNet-34)	75.21	**59.74**	69.18	50.54	72.20	55.14
TSOM (ResNet-50)	**75.53**	59.67	**69.80**	**51.21**	**72.67**	**55.44**

Table 2. Performance of different methods on challenging scenarios.

Methods	Facial-components†		Facial-attributes†		Average	
	FACC	AACC	FACC	AACC	FACC	AACC
Two-Stream	24.00	12.74	31.55	16.92	27.78	14.83
DRN	48.11	14.11	50.35	20.62	49.23	17.37
MA	59.08	33.38	67.20	47.07	63.14	40.23
SeqFake-Former (ResNet-34)	58.70	31.43	62.88	41.88	60.79	36.66
SeqFake-Former (ResNet-50)	57.35	29.50	64.81	44.49	61.08	37.00
TSOM (ResNet-34)	**61.79**	**36.80**	**67.44**	**47.14**	**64.62**	**41.97**
TSOM (ResNet-50)	**62.52**	**37.21**	**67.33**	**48.03**	**64.93**	**42.62**

L_{rec} and KLD loss L_{kld} on constructed auto-encoder $(\mathcal{E}, \mathcal{D})$ as in [11].

The overall loss function is expressed as

$$L = L_{\text{ce}} + \lambda_1 L_{\text{rec}} + \lambda_2 L_{\text{kld}}. \tag{13}$$

where λ_1, λ_2 are weights for balancing these loss terms.

4. Experiments

4.1. Experimental Settings

Datasets. To the best of our knowledge, there is only one public dataset for sequential DeepFake detection that proposed in [23]. This dataset contains two types of sequential DeepFake manipulations, the facial components manipulation and the facial attributes manipulation. The first track comprises $35,166$ face images that are subjected to replacing operations on specific facial components. There are 28 manipulation types in this track, with a maximum sequence length of five. The second track contains $49,920$ face images with 26 types of attribute manipulations. The maximum length of each manipulation is also five. Its split of training, validating, and testing set is $8:1:1$.

Evaluation Metrics. In line with previous work [23], we employ Fixed Accuracy (FACC) and Adaptive Accuracy (AACC) for evaluation. To calculate the FACC score, the sequence annotations shorter than five are padded with the label "no manipulation" and then compare the predictions with the padded sequence annotation. On the other hand, due to the nature of the auto-regress mechanism, the model can stop predicting upon encountering the End-of-Sequence (EOS) symbol, resulting in various lengths of the predicted sequence. The AACC score is calculated by comparing the predictions with the sequence annotations in terms of dynamic lengths.

Implementation Details. Each branch in the spatial encoder has two Transformer blocks. Each block contains four attention heads. The sequential decoder consists of two Transformer blocks. In the training phase, the image size is set to 256×256 with augmentations such as horizontal Flipping and adjusting the brightness of the image. We employ the AdamW optimizer with a weight decay of 10^{-4}. The initial learning rates are set to 10^{-4} for CNN stem and 10^{-4} for Transformer architecture. We utilize a warm-up strategy of 20 epochs, with a batch size of 32. The total epochs are set to 170, with a 10% reduction in learning rate

Table 3. Effect of each component.

Setting	Resnet-34		Resnet-50		Average	
	FACC	AACC	FACC	AACC	FACC	AACC
BL	70.45 (+0.00)	52.18 (+0.00)	69.47 (+0.00)	50.89 (+0.00)	69.96 (+0.00)	51.34 (+0.00)
BL+DPDA	71.07 (+0.62)	52.84 (+0.66)	71.06 (+1.59)	52.80 (+1.91)	71.07 (+1.11)	52.82 (+1.48)
BL+DPDA+MSCA	71.74 (+1.29)	54.05 (+1.87)	72.04 (+2.57)	54.42 (+3.53)	71.89 (+1.93)	54.24 (+2.90)
BL+DPDA+MSCA+IOP	74.22 (+3.77)	58.19 (+6.01)	74.30 (+4.83)	58.05 (+7.16)	74.26 (+4.30)	58.12 (+6.78)
BL+DPDA+MSCA+IOP+SGM	**75.21** (+4.76)	**59.74** (+7.56)	**75.53** (+6.06)	**59.67** (+8.78)	**75.37** (+5.41)	**59.71** (+8.37)

Table 4. Performance of applying IOP on various methods.

Methods	Facial-components		Facial-attributes		Average	
	FACC	AACC	FACC	AACC	FACC	AACC
Two-Stream	72.00	53.97	65.20	44.39	68.60	49.18
Two-Sream+IOP	73.55(+1.55)	58.03(+4.06)	67.60(+2.40)	47.78(+3.39)	70.58(+1.98)	52.91(+3.73)
DRN	69.38	49.76	67.66	47.31	68.52	48.54
DRN+IOP	71.81(+2.43)	54.21(+4.45)	68.89(+1.23)	49.74(+2.43)	70.35(+1.83)	51.98(+3.44)
MA	70.12	51.82	68.91	49.19	69.52	50.51
MA+IOP	73.41(+3.29)	57.30(+5.48)	68.13(-0.78)	48.58(-0.61)	70.77(+1.25)	52.94(+2.43)
SeqFake-Former	70.97	52.65	68.40	48.93	69.69	50.79
SeqFake-Former+IOP	73.34 (+2.37)	56.93 (+4.28)	69.09(+0.69)	48.40 (-0.53)	71.22 (+1.53)	52.67(+1.88)

every 50 epochs. The parameters in the loss function are set as: $\lambda_1 = 0.1, \lambda_2 = 0.01$.

4.2. Results

Table 1 compares the performance of our method with other counterparts, including Multi-Cls, DETR [2], DRN [29], MA [39], Two-Stream [18], SeqFake-Former [23] respectively. Multi-Cls and DETR serve as baseline methods adapted from models for general vision tasks. Multi-Cls performs multi-label classification by directly classifying the manipulated images into multiple classes. DETR is a modified object detector that replaces object queries with manipulation annotation. DRN, MA, and Two-Stream are DeepFake detection methods adapted by substituting the binary classification head with a multi-classification head. SeqFake-Former [23] is a dedicated method for sequential DeepFake detection. MMNet [31] is the latest work achieving state-of-the-art performance. The results of Multi-Cls, DETR, DRN, MA, and Two-stream are referred from [23]. **Note that only DRN, MA, Two-stream, and SeqFake-Former have released their codes.** For better evaluation, we reproduce them with their released codes (marked by $*$) *(More details in Supplementary)*. As shown in Table 1, our method outperforms others in both tracks, averaging 72.67% and 55.44% in FACC and AACC scores on ResNet-50. Compared to SeqFake-Former, our method improves by $2.88\%, 4.37\%$ and $0.94\%, 1.58\%$ in these two tracks on ResNet-50. Moreover, our method averagely surpasses MMNet by 1.07% and 1.8% in FACC and AACC.

In Challenging Scenarios. In real-world scenarios, images often undergo various post-processing operations. To these scenarios, we perform post-processing operations on the facial-components and facial-attributes datasets. Specifically, for each image, we randomly select one or two processes from a set that includes adding Gaussian noises, shifting RGB channels, compression, converting to gray, and jittering colors. This increases the difficulty of original datasets, which we refer to as Facial-components† and Facial-attributes†. We then evaluate different methods under these challenging scenarios. As shown in Table 2, while all methods suffer a notable performance drop, our model still outperforms others, demonstrating its robustness against external disturbances.

4.3. Analysis

Effect of Each Component. Our method consists of several key components, including the Diversified Pixel Difference Attention module (DPDA), Multi-source Cross-attention (MSCA), Inverted Order Prediction (IOP), and Shape-guided Gaussian Mapping (SGM). To evaluate the effectiveness of each component, we conduct a series of ablation studies on the facial-components track using the stem of ResNet-34 and ResNet-50. Specifically, 1) the baseline model (BL) contains only basic encoders and decoders. 2) Adding the DPDA module to the baseline model and directly feeding the features from the encoder into the sequential decoder. 3) Employ the MSCA strategy for feature integration. 4) Incorporating inverted order prediction. 5) Inserting the SGM module into the MSCA strategy. The results shown in Table 3 indicate the contribution of each component to the overall performance. For instance, with Resnet-34, the DPDA module improves the FACC and AACC by 0.59% and 1.44%, respectively. The MSCA strategy further enhances the performance by 0.76% in FACC and 0.8% in AACC. The IOP significantly boosts FACC by

Table 5. Different loss weights λ_1, λ_2.

λ_1	λ_2	FACC	AACC
0.1	0.1	72.21	56.02
0.01	0.01	73.32	56.84
0.5	0.01	74.20	58.21
1	0.01	73.71	57.24
0.1	0.01	**75.21**	**59.74**

Table 6. Performance on FF++.

Methods	ACC
Two-stream	82.40
DRN	81.42
MA	82.64
SeqFake-Former	82.43
TSOM	**83.70**

2.32% and AACC by 3.98%, showing its effectiveness in gradually capturing manipulation traces. Finally, the integration of all components leads to the best performance.

Generalizability of IOP. The proposed IOP strategy is applicable to many architectures. To verify this, we evaluate the performance of different methods with/without IOP. As shown in Table 4, all methods have gained enhancement on average. Even the methods of MA, DRN, and two-stream that are not specifically designed for sequence detection tasks surprisingly improve their performance by a notable margin. In particular, by applying IOP, Two-Stream even achieves competitive performance with MMnet on the facial-components track, demonstrating the generic effect of IOP for SDD.

Study on Loss Weights λ_1, λ_2. We study the effect of various loss weight λ_1 and λ_2. The results are shown in Table 5. It can be seen that our method is not very sensitive to the values of loss weights ($\pm 3\%$). This is because the classification loss primarily influences the learning direction during training. Compared to the KLD loss λ_2, the reconstruction loss λ_1 seemly has more impact, as it directly instructs the training of encoder $\mathcal{E}$ in a self-supervised manner. In the main experiment, we use $\lambda_1 = 0.1, \lambda_2 = 0.01$.

Adaptation on One-step DeepFake Detection (ODD). Since one-step detection does not require sequential clues, existing SDD methods can not be directly applied to this task. For more comprehensive studies, we adapt these methods by replicating the one-step label $N = 5$ times as the sequential annotations, where N is the maximum manipulation length in dataset. For example, the real faces are assigned with annotations of $[0, 0, 0, 0, 0]$, while forged faces correspond to $[1, 1, 1, 1, 1]$. For comparison, we train these methods on the FF++ dataset [21] under this configuration. Note the basenetwork of SeqFake-Former and ours is ResNet-34. The results are shown in Table 6, revealing that our method still achieves the best results.

Attention Visualization. Fig. 6 shows the Grad-CAM [22] of different methods. We observe that under the same annotation, our method can identify the manipulated regions more precisely. In contrast, Seq-FakeFormer can localize the major manipulated regions but may overlook small manipulations with subtle traces. This result highlights the effectiveness of our method in capturing sequential manipulation traces. Furthermore, we also observe that our method can highlight regions that are not annotated as manipulation. We attribute this to the imperfect nature of GAN in the feature space, causing subtle influences on some neighboring areas during the multi-step facial manipulation process. Our method can capture this subtle alternation as evidence for exposing the sequential manipulations.

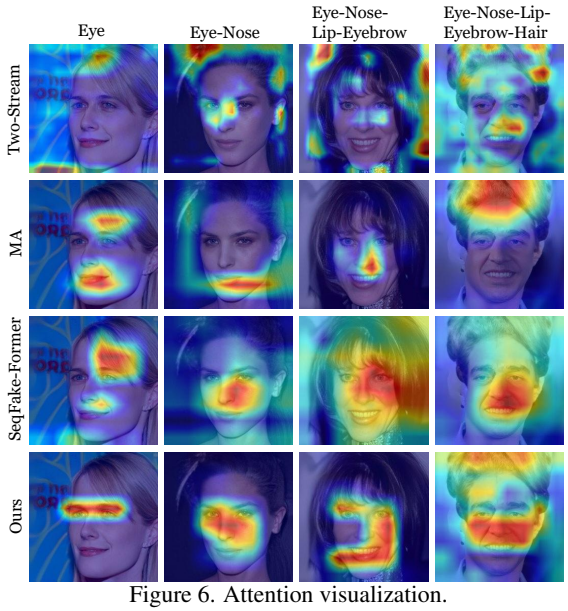

Figure 6. Attention visualization.

5. Conclusion

This paper introduces a new Transformer design, called TSOM, for sequential DeepFake detection. Our method is inspired by three perspectives: Texture, Shape, and Order of Manipulations, leading to four key improvements. Firstly, we propose a text-aware branch featuring new Diversiform Pixel Difference Attention modules (DPDA) to capture subtle manipulation traces. Secondly, we introduce a Multi-source Cross-attention module (MSCA) to explore deep correlations between spatial and sequential features. To enhance the cross-attention further, we describe a Shape-guided Gaussian Mapping (SGM) strategy to incorporate priors on manipulation shapes. Lastly, we improve performance using a simple Inverted Order prediction (IOP) strategy, which reverses the manipulation annotation order from forward to backward. The results on public datasets demonstrate the efficacy of our method in sequential DeepFake detection.

Acknowledgement. This work is supported in part by the National Natural Science Foundation of China (No.62402464), Shandong Natural Science Foundation (No.ZR2024QF035), China Postdoctoral Science Foundation (No.2021TQ0314, No.2021M703036).

References

[1] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting world leaders against deep fakes. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, 2020.

[3] 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 *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2023.

[4] Tarik Dzanic, Karan Shah, and Freddie Witherden. Fourier spectrum discrepancies in deep network generated images. *Advances in neural information processing systems*, 33:3022–3032, 2020.

[5] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[6] Zhihao Gu, Taiping Yao, Yang Chen, Ran Yi, Shouhong Ding, and Lizhuang Ma. Region-aware temporal inconsistency learning for deepfake video detection. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022.

[7] Ying Guo, Cheng Zhen, and Pengfei Yan. Controllable guide-space for generalizable face forgery detection. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 20818–20827, 2023.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[9] Yang He, Ning Yu, Margret Keuper, and Mario Fritz. Beyond the spectrum: Detecting deepfakes via re-synthesis. In *30th International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[10] Felix Juefei-Xu, Vishnu Naresh Boddeti, and Marios Savvides. Local binary convolutional neural networks. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[12] 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 *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 6458–6467, 2021.

[13] 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 Computer Vision and Pattern Recognition (CVPR)*, pages 5001–5010, 2020.

[14] 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.

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

[16] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 772–781, 2021.

[17] Zhengzhe Liu, Xiaojuan Qi, and Philip HS Torr. Global texture enhancement for fake face detection in the wild. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 8060–8069, 2020.

[18] Yuchen Luo, Yong Zhang and1 Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2021.

[19] Changtao Miao, Zichang Tan, Qi Chu, Huan Liu, Honggang Hu, and Nenghai Yu. F 2 trans: High-frequency fine-grained transformer for face forgery detection. *IEEE Transactions on Information Forensics and Security*, 18:1039–1051, 2023.

[20] Hua Qi, Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Wei Feng, Yang Liu, and Jianjun Zhao. Deeprhythm: Exposing deepfakes with attentional visual heartbeat rhythms. In *Proceedings of the 28th ACM international conference on multimedia*, pages 4318–4327, 2020.

[21] 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.

[22] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *International Journal of Computer Vision (IJCV)*, 2019.

[23] Rui Shao, Tianxing Wu, and Ziwei Liu. Detecting and recovering sequential deepfake manipulation. In *European Conference on Computer Vision (ECCV)*, 2022.

[24] Kaede Shiohara and Toshihiko Yamasaki. Detecting deepfakes with self-blended images. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 18720–18729, 2022.

[25] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikainen, and Li Liu. Pixel difference networks for efficient edge detection. In *International Conference on Computer Vision (ICCV)*, 2021.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[27] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell:a neural image caption generator. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2015.

[28] Haoran Wang, Weitang Liu, Alex Bocchieri, and Yixuan Li. Can multi-label classification networks know what they don't know? *Advances in Neural Information Processing Systems*, 2021.

[29] Sheng-Yu Wang, Oliver Wang, Andrew Owens, Richard Zhang, and Alexei A. Efros. Detecting photoshopped faces

by scripting photoshop. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2019.

[30] 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 *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 7278–7287, 2023.

[31] Ruiyang Xia, Decheng Liu, Jie Li, Lin Yuan, Nannan Wang, and Xinbo Gao. MMNet: Multi-collaboration and multi-supervision network for sequential deepfake detection. *IEEE Transactions on Information Forensics and Security*, 2024.

[32] Yuting Xu, Jian Liang, Gengyun Jia, Ziming Yang, Yanhao Zhang, and Ran He. Tall: Thumbnail layout for deepfake video detection. In *International Conference on Computer Vision (ICCV)*, 2023.

[33] Jiachen Yang, Aiyun Li, Shuai Xiao, Wen Lu, and Xinbo Gao. MTD-Net: learning to detect deepfakes images by multi-scale texture difference. *IEEE Transactions on Information Forensics and Security*, 16:4234–4245, 2021.

[34] 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.

[35] Zitong Yu, Rizhao Cai, Zhi Li, Wenhan Yang, Jingang Shi, and Alex C Kot. Benchmarking joint face spoofing and forgery detection with visual and physiological cues. *IEEE Transactions on Dependable and Secure Computing*, 2024.

[36] Zitong Yu, Jun Wan, Yunxiao Qin, Xiaobai Li, Stan Z Li, and Guoying Zhao. NAS-FAS: Static-dynamic central difference network search for face anti-spoofing. *IEEE transactions on pattern analysis and machine intelligence*, 43(9):3005–3023, 2020.

[37] Zitong Yu, Chenxu Zhao, Zezheng Wang, Yunxiao Qin, Zhuo Su, Xiaobai Li, Feng Zhou, and Guoying Zhao. Searching central difference convolutional networks for face anti-spoofing. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2020.

[38] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2185–2194, 2021.

[39] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2021.

[40] Tianchen Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. Learning self-consistency for deepfake detection. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 15023–15033, 2021.