








# A Robust Lightweight Deepfake Detection Network Using Transformers

Yaning Zhang<sup>1</sup>, Tianyi Wang<sup>2</sup>, Minglei Shu<sup>1</sup>, and Yinglong Wang<sup>1</sup>

<sup>1</sup> Shandong Artificial Intelligence Institute, Qilu University of Technology  
(Shandong Academy of Sciences), Jinan, China  
shum1@sdas.org, wangylscsc@126.com

<sup>2</sup> Department of Computer Science, The University of Hong Kong, Hong Kong, China  
tywang@cs.hku.hk

**Abstract.** Deepfake detection attracts widespread attention in the computer vision field. Existing efforts achieve outstanding progress, but there are still significant unresolved issues. Coarse-grained local and global features are insufficient to capture subtle forgery traces from various inputs. Moreover, the detection efficiency is not powerful enough in practical applications. In this paper, we propose a robust and efficient transformer-based deepfake detection (TransDFD) network, which learns more discriminative and general manipulation patterns in an end-to-end manner. Specifically, a robust transformer module is designed to study fine-grained local and global features based on intra-patch locally-enhanced relations as well as inter-patch locally-enhanced global relationships in face images. A novel plug-and-play spatial attention scaling (SAS) module is proposed to emphasize salient features while suppressing less important representations, which can be integrated into any transformer-based models without increasing computational complexity. Extensive experiments on several public benchmarks demonstrate that the proposed TransDFD model outperforms the state-of-the-art in terms of robustness and computational efficiency.

**Keywords:** Deepfake detection · Spatial attention scaling · Transformer

## 1 Introduction

The threat of face manipulated videos has raised widespread attention, especially after the advent of the deepfake technique that adopts deep learning tools. Deepfake can replace the face in the target video with the face in the source video using deep learning-based technologies such as autoencoder [14] and generative adversarial network (GAN) [8]. With these approaches, face generated videos are exceedingly simple to be generated on the condition that one can access a large amount of data spread widely on the Internet, which brings negative impacts on

---

Y. Zhang and T. Wang—Contributed equally to this work.

individuals, organizations, and governments while greatly threatening the social stability [17]. Furthermore, with the sophistication and development of synthesis techniques, deepfake videos have become more realistic and it is challenging for human eyes to discern authenticity. The above challenges have driven the development of deepfake detection using deep neural networks (DNNs) [4, 9, 18, 30]. Most of the existing efforts in common exploit the powerful data fitting capabilities of neural networks to mine discriminative features for deepfake detection. Deep learning-based detection approaches usually regard deepfake detection as a binary classification problem and employ convolutional neural networks (CNN) to analyze local features. However, the learned representations using CNN are not general enough since CNN seldomly focuses on global information. Furthermore, it is challenging to discern authenticity based on small local regions only. Recent work recognizes this problem and attempts to utilize a transformer-based model [28] to extract global embeddings for capturing long-range manipulation traces. However, it usually analyzes global characteristics in a coarse-grained manner, which may cause some image patches with weak artifacts to be rarely noticed due to face pose transitions. Therefore, coarse-grained global feature learning often serves as a suboptimal solution. In addition, the detection efficiency of the model is increasingly important in practical applications. Recent work has made significant advancements in deepfake detection performance, while state-of-the-art deepfake detectors also become gradually more expensive. For example, the advanced multi-attention (MAT) detector [29] requires 417.63M parameters and 224.38G floating-point operations per second (FLOPs) (20x more than Xception [21]) to realize state-of-the-art performance. Many face forgery detection models depend on on-device computation. Computational overhead is one of the main factors limiting the deployment of current networks in practical applications due to the inadequate computing power, large memory footprint, and severe battery consumption of the device. Based on these real-world resource restrictions, the model efficiency becomes increasingly important for face forgery detection. However, few approaches consider the computational complexity such as the number of parameters and FLOPs. Although some studies utilize the lightweight model Xception [3, 16] to obtain remarkable results, their ability to study general representations is limited due to the coarse-grained local feature learning. As a result, these methods are insufficient to capture weak manipulated patterns owing to the diversity of forgery techniques.

Based on the discussion above, our method mainly solves the following two problems: (1) how to study more discriminative and general features for deepfake detection; (2) how to achieve state-of-the-art detection performance as efficiently as possible. In order to tackle these limitations, we propose a robust lightweight transformer-based deepfake detection (TransDFD) model. In detail, our model consists of two key components: the robust transformer module and the spatial attention scaling (SAS) technique. Robust transformer restricts locally-enhanced multi-head self-attention (LMSA) within each patch and boosts information flow across image patches by the spatial shuffle, thus learning fine-grained local and global representations. SAS flexibly refines spatial features to emphasize more

significant manipulated artifacts, and vice versa. The main contributions of this work are summarized as follows:

- We propose a robust and lightweight TransDFD network for deepfake detection, which captures discriminative and comprehensive forgery traces with much fewer parameters and computational costs.
- The robust transformer is presented to learn fine-grained local and global features via focusing on intra-patch locally-enhanced relations and inter-patch locally-enhanced global relationships in face images.
- We design an innovative plug-and-play SAS technique to suppress less important representations while emphasizing more critical features, via a learnable diagonal matrix, which can be widely applied to boost the representation ability of transformers.
- Extensive experiments on several challenging datasets demonstrate the efficiency and robustness of our proposed model and feature visualizations show the generalizability and interpretability of our method.

## 2 Related Work

Most existing deepfake detection models utilize CNNs or attention mechanisms to capture local discriminative features. Rossler *et al.* [21] used the lightweight Xception, a standard CNN pre-trained on ImageNet, and transferred to the deepfake detection task, to extract local features. The TwoStream framework [18] applies two streams of Xception backbones which analyze the high-frequency feature and RGB content, respectively, for generalized face forgery detection. The representative forgery mining (RFM) [27], an attention-based data augmentation framework, exploits the Xception backbone to guide the detector to refine its attention for capturing local discriminative patterns. The multi-attentional (MAT) architecture [29] establishes a multi-attentional module to combine the low-level textural features and high-level semantic features. Kumar *et al.* [15] adopted multi-streamed CNNs to learn fine-grained local features, considering intra-patch local relations and inter-patch partial relationships within the face image. However, these models only extract local discriminative features and hardly consider the global relations among image patches. To address this problem, a convolutional vision transformer (CViT) framework [28] is proposed to integrate CNN and vision transformer (ViT) [6] for deepfake detection. Specifically, the CNN extracts local features while the ViT analyzes them to capture the inter-patch global dependencies at a coarse-grained level. We noted that, by contrast, our approach is capable of learning fine-grained local and global representations with fewer parameters and computational costs.

## 3 Approach

### 3.1 Network Architecture

The framework of our proposed TransDFD is illustrated in Fig. 1. TransDFD is composed of local feature extraction (LFE), robust transformer, and SAS.

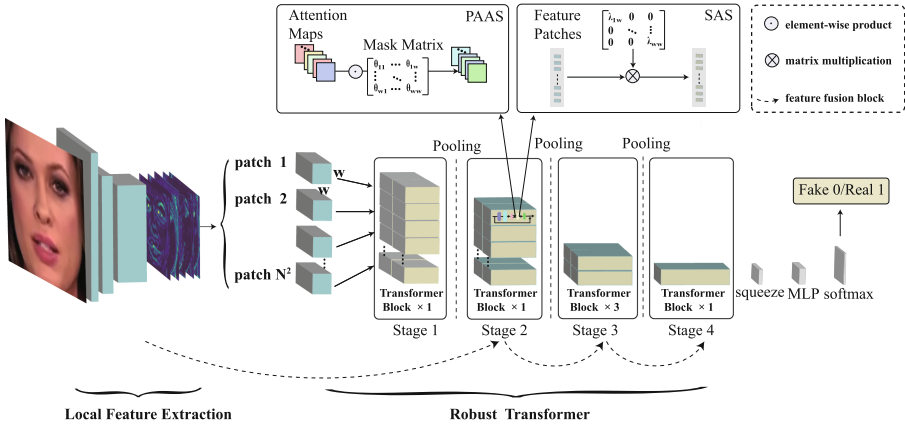


Fig. 1. The overall framework of TransDFD.

LFE adaptively filters the redundant information of a face image to obtain the refined feature map. Robust transformer (Sect. 3.3) utilizes transformer blocks to divide them into  $N^2$  square patches with size  $w \times w$  to encode feature vectors from a patch, thereby capturing fine-grained local and global representations. Meanwhile, the robust transformer employs the feature fusion block to analyze the refined feature map for obtaining local embeddings and supplementing them into fine-grained global representations. After that, SAS (Sect. 3.4) further refines elaborate embeddings using a learnable diagonal matrix. Finally, we squeeze the output of models and flatten them into feature vectors. The multiple layer perceptron (MLP) and softmax generate final detection results.

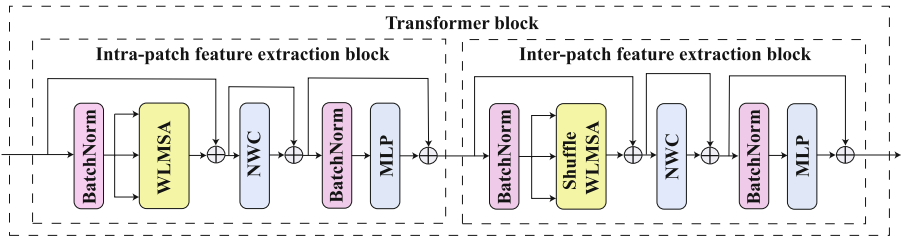


Fig. 2. The structure of the transformer block.

### 3.2 Local Feature Extraction

In order to filter redundant information irrelevant to the detection task in face images, LFE is designed to obtain fine feature maps in a simple and effective manner. In detail, the LFE module consists of the first two sequence blocks of VGG [24]. To save parameters and improve computation efficiency, the output

channels of each convolutional layer in the first and second blocks in VGG are adjusted to 32 and 64, respectively. LFE extracts the delicate feature map  $F_f \in \mathbb{R}^{C \times H \times W}$  as shown in Fig. 1 by inputting a facial image  $F_i \in \mathbb{R}^{3 \times 224 \times 224}$ , where  $H, W, C$  denotes the height, width, and channel of the feature map, respectively, and  $H = W = 56, C = 64$ . The  $F_f$  is then fed into the first transformer block and the first feature fusion block in the robust transformer module, simultaneously.

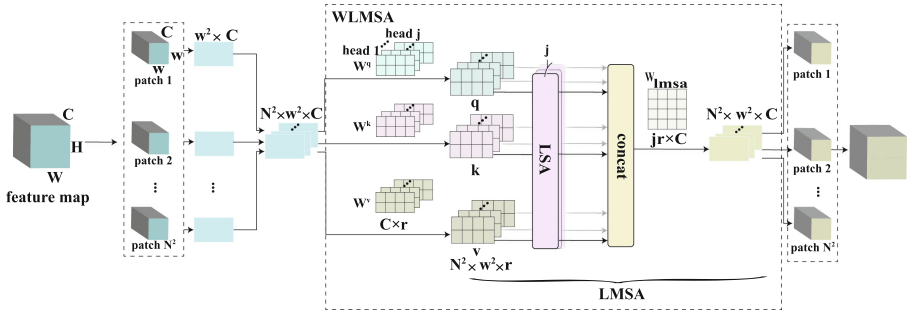


Fig. 3. The main workflow of the intra-patch feature extraction block.

### 3.3 Robust Transformer

Unlike the existing method [28] that utilizes the MSA mechanism to capture coarse-grained global features, inspired by the shuffle transformer [11] architecture with the novel window-based multi-head self-attention (WMSA) [10,16] mechanism and low costs, we propose a robust transformer module to focus on fine-grained local and global features learning. Figure 1 illustrates the architecture of the robust transformer module. Specifically, robust transformer contains four stages. Except for the first stage with  $L$  transformer blocks, each stage consists of a pooling layer,  $L$  transformer blocks, and a feature fusion block. Figure 2 shows the structure of the transformer block. Each transformer block includes two cascaded blocks: intra-patch feature extraction block and inter-patch feature extraction block. In particular, the former captures patch-level local enhancement relations by window-based locally-enhanced multi-head self-attention (WLMSA) module, obtaining fine-grained local representations, and the latter utilizes shuffle window-based locally-enhanced multi-head self-attention (Shuffle WLMSA) module to gain the fine-grained global embeddings via exploring patch-level locally-enhanced global relations. Through two cascaded blocks, transformer blocks analyze the local and global forgery patterns for each patch. To combine the advantages of CNNs in extracting local features and the benefits of transformer blocks in capturing long-range dependencies, the feature fusion block in each stage studies an input feature map through a convolutional layer with a kernel size of 2 and a stride of 2 to obtain downsampling and it is added element-wisely with the output of the last transformer block in this stage.

**Transformer Block.** Different from the traditional transformer block in ViT [6], we perform LMSA computation on each image patch in parallel, and the linear layers are replaced by the convolutional layers, which reduces the number of parameters and computational complexity. As Fig. 2 shows, the transformer block includes two cascaded blocks. The intra-patch feature extraction block aims to model patch-level locally-enhanced relations to obtain fine-grained local representations. The main workflow of the block is shown in Fig. 3. Without losing generality, given a feature map  $F \in \mathbb{R}^{C \times H \times W}$ , we first divide it into  $N^2$  square patches with size  $w \times w$ , and each square patch is reshaped into a succession of flattened 2D feature patches to get  $F_p \in \mathbb{R}^{N^2 \times w^2 \times C}$ , where  $w$  is the width and height of square patches,  $w^2$  is the number of feature patches in a square patch, and  $N^2 = (H/w) \times (W/w)$  is the number of square patches. To study fine-grained local representations and reduce computational costs, we introduce the WLMSA module. In detail, we transform  $F_p$  into three different tensors, *i.e.*, a query  $q = F_p W^q$ , a key  $k = F_p W^k$ , and a value  $v = F_p W^v$ , where query, key, and value tensors are calculated for each square patch from the feature map  $F_p$ , and  $W^q$ ,  $W^k$ , and  $W^v$  are parameters of the convolutional layer. We utilize LMSA to acquire  $F'_p \in \mathbb{R}^{N^2 \times w^2 \times C}$  with fine-grained local embeddings, which implies that square patches are captured variously locally-enhanced relations between the respective internal characteristics in parallel. The LMSA is discussed in detail in the following section. Thereafter, we rearrange the square patches to their original spatial position to obtain  $F_s$ , *i.e.*,  $F'_p \in \mathbb{R}^{N^2 \times w^2 \times C}$  is reshaped to  $F_s \in \mathbb{R}^{C \times H \times W}$ . Finally, we pass  $F_s$  into the neighbor-window connection (NWC) module and MLP module sequentially to obtain  $V \in \mathbb{R}^{C \times H \times W}$ . NWC consists of a convolutional layer with a kernel size equal to the image patch size to enhance connections among neighboring patches. The linear layer in the conventional MLP module [6] is adjusted to a convolutional layer with a kernel of  $1 \times 1$  for economizing parameters. Intra-patch feature extraction block only analyzes patch-level local relationships without taking into account the global relations between image patches. To overcome the limitation, we present the inter-patch feature extraction block whose main workflow is similar to that shown in Fig. 3. In detail, we firstly split  $V \in \mathbb{R}^{C \times H \times W}$  into  $w^2$  square patches with size  $N \times N$  to get  $V_p \in \mathbb{R}^{w^2 \times N^2 \times C}$ . To achieve spatial shuffle and inter-patch information communication, each new square patch with size  $w \times w$  is composed of the feature patches at the same position in  $w^2$  square patches with size of  $N \times N$ , carrying information for the overall patches with size of  $N \times N$ . That is to say, we rearrange  $V_p \in \mathbb{R}^{w^2 \times N^2 \times C}$  to  $V_f \in \mathbb{R}^{N^2 \times w^2 \times C}$ . We introduce the Shuffle WLMSA module which has a similar pipeline to WLMSA and considers locally-enhanced global relations for each image patch in parallel to obtain  $V'_f \in \mathbb{R}^{N^2 \times w^2 \times C}$  with fine-grained global features by inputting  $V_f \in \mathbb{R}^{N^2 \times w^2 \times C}$ . Afterward, we adjust the feature patches to the original positions for spatial alignment. *i.e.*,  $V'_f \in \mathbb{R}^{N^2 \times w^2 \times C}$  is rearranged to  $V_s \in \mathbb{R}^{w^2 \times N^2 \times C}$ . Thereafter, we align image content spatially to obtain the feature map  $I$ . That is,  $V_s \in \mathbb{R}^{w^2 \times N^2 \times C}$  is reshaped to  $I \in \mathbb{R}^{C \times H \times W}$ . Finally, we transfer  $I$  through the NWC and MLP modules to get  $T \in \mathbb{R}^{C \times H \times W}$  which is then fed into the intra-patch feature extraction block in the subsequent transformer block.

**LMSA.** Inspired by [7], we find it beneficial to model locally-enhanced relations between adjacent signals within image patches when given the query, key, and value tensors. Since in the traditional MSA, each feature patch is equally accessible to any other ones and feature patches not in the neighborhood may also attend to each other with relatively large scores, as Fan *et al.* [7] proves mathematically, which potentially introduces noises to semantic modeling and overlooks the link among the surrounding signals. Therefore, a PAAS [20] technique is introduced to remove noise and study the relationships between adjacent feature patches within an image patch. Specifically, the MSA [11] produces the attention maps by formula  $qk^T/\sqrt{r}+B$ , and each value of attention maps denotes the correlation for any two feature patches in a square patch. We introduce a learnable position importance matrix  $W_p \in \mathbb{R}^{w^2 \times w^2}$  to act as a soft attention mask. That is to say, we assign a learnable weight for each element of attention maps to learn the correlations between feature patches, adaptively, thereby eliminating the noises, which is defined as Eq. 1. A locally-enhanced self-attention (LSA), *i.e.*, a feature map with locally-enhanced information, is calculated by Eq. 1. Formally, the LMSA is computed as follows:

$$\text{LSA} = \text{softmax}\left(\left(\frac{qk^T}{\sqrt{r}} + B\right) \odot W_p\right)v, \tag{1}$$

$$\text{LMSA} = [\text{LSA}_1; \text{LSA}_2; \dots; \text{LSA}_j]W_{\text{lmsa}}, \tag{2}$$

where  $q, k, v \in \mathbb{R}^{N^2 \times j \times w^2 \times r}$  are the query, key, and value tensors, respectively.  $w^2$  is the number of feature patches in a square patch.  $j$  denotes the number of attention heads and  $r = C/j$  denotes the dimension of the feature patch in head space.  $B \in \mathbb{R}^{w^2 \times w^2}$  [23] is the relative position matrix.  $\odot$  is the element-wise product.  $W_{\text{lmsa}} \in \mathbb{R}^{jr \times C}$  is the learned parameter.

### 3.4 Spatial Attention Scaling

In order to learn detailed features, we devise the robust transformer module. However, the fine-grained embeddings obtained by Eq. 1 may contain noises as demonstrated by [7], we propose a SAS mechanism to further refine the representations. Specifically, our SAS method denotes that a diagonal matrix right-multiplies the output of LSA, which means that we assign a learnable weight to each spatial feature, and the spatial features of the same position in different channels share the weight. The LMSA in the robust transformer module is modified as follows:

$$F = \text{diag}(\lambda_1, \dots, \lambda_{w^2})\text{LSA}, \tag{3}$$

$$\text{LMSA} = [F_1; F_2; \dots; F_j]W_{\text{lmsa}}, \tag{4}$$

where the parameters  $\lambda_i$  are learnable weights for  $i = 1, \dots, w^2$ . Diagonal matrix is initialized to follow a standard normal distribution.  $F \in \mathbb{R}^{N^2 \times j \times w^2 \times r}$  is the

feature map with refined characteristics. Formally, SAS does not alter the computational overhead of the network by adding these weights since they can be combined into the prior tensor of the LMSA as Eq. 3 demonstrates.

## 4 Experiments

### 4.1 Experiments Setting

**Datasets.** We carried out research on three benchmark databases, *i.e.*, FaceForensics++ (FF++) [21], Deepfake Detection Challenge (DFDC) [5], Deepfake Detection (DFD). FF++ includes 1,000 original videos from YouTube and 4,000 fake videos. The fake videos are generated by four algorithms: DeepFakes (DF) [1], Face2Face (F2F) [26], FaceSwap (FS) [2], and NeuralTextures (NT) [25]. FF++ has three qualities with distinct compression degrees, *i.e.*, raw, high quality (HQ), and low quality (LQ). We applied the HQ-type videos and the official splits, using 740 videos for training, 140 videos for validation, and 140 videos for testing. DFDC is a wide-scale deepfake dataset with a large number of clips and different quality levels. DFD is a deepfake detection dataset that utilizes publicly available deepfake generation methods to create over 3,000 manipulated videos from 28 actors in various scenes. The performance on the test set is reported.

**Evaluation Metrics.** We adopted the accuracy (ACC) and area under the receiver operating characteristic curve (AUC) as our evaluation criteria. Since most previous work rarely presents the metric of computation complexity, as a result, we computed the number of parameters and FLOPs of the models using the same setting.

**Implementation Details.** We used dlib [12] to crop the face regions as input facial images with size  $224 \times 224$ . The size  $w$  of square patches in the robust transformer module is set to 7. The depth  $L$  of the robust transformer is set to 6 with four phases with 1, 1, 3, and 1 transformer blocks and the attention heads  $j$  are set to 2, 4, 8, and 16, respectively. Furthermore, our model is trained with Adam optimizer [13] with learning rate  $1e-4$  and weight decay  $1e-5$ . We utilized the scheduler to drop the learning rate by ten times every 15 epochs.

### 4.2 Comparison with the State of the Art

**Within-Dataset Evaluation.** We used FF++ for training and conducted the within-dataset evaluation. Results are displayed in Table 1. Our method consistently outperforms the recent mainstream models on four manipulation methods. In particular, our model outperforms the state-of-the-art, Xception, by 4.7% AUC, on the most difficult NT forgery technology that barely creates visible fabricated artifacts, illustrating the effectiveness of our proposed model. Furthermore, our method possesses the minimum number of parameters and FLOPs among all compared approaches as shown in Table 2. That is to say, our method is superior in terms of both computing efficiency and detection accuracy.



**Table 1.** Comparison with state-of-the-art methods on within-dataset. We trained on FF++ which consists of four manipulation techniques.

Method	DF		F2F		FS		NT		FF++	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
MAT [29]	90.70	97.43	90.64	97.75	90.82	97.02	77.65	85.56	87.50	94.85
CViT [28]	86.59	96.17	87.75	97.85	92.28	98.71	74.99	82.78	84.86	92.36
TwoStream [18]	91.08	97.39	91.54	97.96	90.82	96.39	79.12	86.65	88.17	94.93
Xception [21]	90.54	97.34	91.93	98.12	95.61	99.28	82.18	90.07	90.08	96.51
TransDFD(Ours)	<b>93.94</b>	<b>98.87</b>	<b>95.24</b>	<b>99.25</b>	<b>97.51</b>	<b>99.70</b>	<b>87.65</b>	<b>94.73</b>	<b>93.60</b>	<b>98.40</b>

**Table 2.** Comparison with state-of-the-art methods on cross-dataset evaluation.

Method	DFDC		DFD		Params(M)	GFLOPs
	ACC	AUC	ACC	AUC		
MAT [29]	63.16	69.56	77.63	85.18	417.63	224.38
CViT [28]	62.79	67.86	72.93	83.24	89.02	6.69
TwoStream [18]	59.93	64.80	75.77	83.79	53.24	13.79
Xception [21]	58.77	66.95	76.84	85.20	20.81	4.59
TransDFD(Ours)	<b>64.12</b>	<b>71.97</b>	<b>84.12</b>	<b>92.23</b>	<b>13.78</b>	<b>4.25</b>

**Cross-Dataset Evaluation.** To evaluate cross-dataset generalization, we trained the networks on FF++ and tested the models on DFDC and DFD. We can see that our proposed model constantly surpasses all of the compared opponents by a significant margin in Table 2. For instance, our method separately exceeds the state-of-the-art Xception which has few parameters and FLOPs by 5.0% and 7.0% AUC on DFDC and DFD, respectively. Different from Xception which merely employs the local information, our model considers the intra-patch relations and inter-patch global relations for fine-grained local and global representation, allowing various artifacts of the manipulated face can be noticed. Furthermore, compared to Xception, the computational costs and the number of parameters are also reduced by 0.3 G and 7.1 M, respectively. In comparison to CViT which also considers both local and global knowledge with transformer, our method confirms excellent performance both in computation overheads and AUC, validating the effectiveness of the fine-grained extraction of global features. Meanwhile, the gains are primarily due to our method’s ability to learn richer forgery traces than compared opponents. Especially for the DFDC dataset, it is a more challenging benchmark since diverse generation technologies are applied to DFDC to achieve larger scale and higher diversity. The AUC of our method is 2.4%, 6.8%, 7.2%, and 5.0% higher than MAT, CViT, Two Stream, and Xception, respectively, on DFDC, which demonstrates the superior robustness of our model.

**Table 3.** Evaluation of each component in TransDFD on FF++. The models are trained from scratch on FF++. ST and RT denote shuffle transformer and robust transformer, respectively.

Datasets	Methods	Params(M)	GFLOPs	ACC	AUC
DF	ST	27.26	4.56	86.78	95.15
	RT	30.42	4.81	87.70	97.73
	LFE+RT	13.75	4.25	93.90	98.76
	LFE+RT+SAS	13.78	4.25	<b>93.94</b>	<b>98.87</b>
F2F	ST	27.26	4.56	83.46	92.98
	RT	30.42	4.81	88.96	97.44
	LFE+RT	13.75	4.25	94.38	99.17
	LFE+RT+SAS	13.78	4.25	<b>95.24</b>	<b>99.25</b>
FS	ST	27.26	4.56	84.80	92.73
	RT	30.42	4.81	93.19	98.10
	LFE+RT	13.75	4.25	96.79	99.48
	LFE+RT+SAS	13.78	4.25	<b>97.51</b>	<b>99.70</b>
NT	ST	27.26	4.56	72.57	78.10
	RT	30.42	4.81	77.61	86.00
	LFE+RT	13.75	4.25	84.92	92.90
	LFE+RT+SAS	13.78	4.25	<b>87.65</b>	<b>94.73</b>
FF++	ST	27.26	4.56	81.93	89.98
	RT	30.42	4.81	87.70	95.25
	LFE+RT	13.75	4.25	92.51	97.87
	LFE+RT+SAS	13.78	4.25	<b>93.60</b>	<b>98.40</b>

### 4.3 Ablation Study

To study the contribution of TransDFD components to learning ability, Table 3 shows the results of our ablation study, which investigates the effect of incrementally adding robust transformer, LFE, and SAS training components.

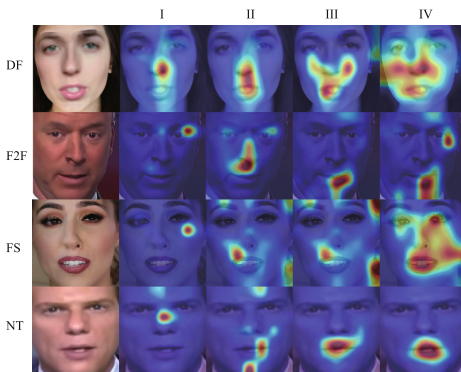
**Effectiveness of Robust Transformer.** We performed the experiments on FF++ to demonstrate that the robust transformer module is necessary. The results are listed in Table 3. It should be noted that the introduction of the robust transformer module consistently improves the ACC and AUC. We believe that the robust transformer module focuses on fine-grained local and global feature learning while paying attention to the local enhancement relationship between fine-grained features, guiding our model to explore more identifiable and comprehensive forgery areas.

**Effectiveness of SAS.** To confirm the effectiveness of our SAS method, our TransDFD model is trained with SAS and without SAS on FF++ and other

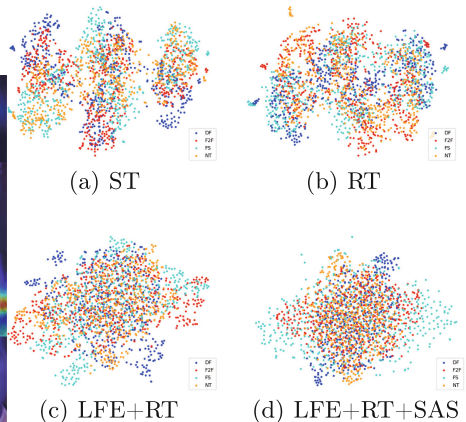
**Table 4.** Ablation results of transformer-based models. We trained on FF++ and tested on FF++, DFDC, and DFD.

Method	FF++		DFDC		DFD		Params(M)	GFLOPs
	ACC	AUC	ACC	AUC	ACC	AUC		
CViT w/o SAS	85.47	94.72	62.79	67.86	72.92	83.24	89.02	6.69
CViT w/ SAS	89.77	96.35	63.64	70.68	80.11	87.96	89.02	6.69
TransDFD w/o SAS	92.51	97.87	62.19	69.95	79.33	87.91	13.75	4.25
TransDFD w/ SAS	<b>93.60</b>	<b>98.40</b>	<b>64.12</b>	<b>71.97</b>	<b>84.12</b>	<b>92.23</b>	13.78	4.25

hyperparameters remain the same. In Table 3, we noticed that due to the introduction of SAS, the AUC of the model is increased by 2.7% on NT. From our perspective, SAS supervises the TransDFD model to concentrate on extensive facial forgery details as shown in Fig. 4. Besides, the parameters of TransDFD with SAS are only increased by 0.03M and the computational complexity is not changed, which lies in our SAS approach can be combined into the prior tensor of the LMSA as the Eq. 3 demonstrates. In order to prove that SAS can boost the performance of transformer-based models, we also conducted ablation experiments on within-dataset and cross-dataset. We show the quantitative results in Table 4, respectively. As we can see, SAS enhances the performance with few parameters and low computational overheads. Assuming that transformer-based models capture diverse global relationships without extra supervision, the SAS approach achieves this by assigning learnable parameters to global features, steering the model to highlight the most important representations and suppress less important ones. As a result, our SAS method boosts the attention of transformer-based models so as to improve their performance.



**Fig. 4.** The heatmap visualizations.



**Fig. 5.** The t-SNE visualizations.

#### 4.4 Visualization

**Visualization of Heatmap.** We visualized the forgery traces captured by different settings using the Grad-CAM [22] on the FF++ dataset, as Fig. 4 illustrates. Each row displays one manipulation approach. From top to bottom, the forgery types are DF, F2F, FS, and NT. The second to fifth columns display the results of four training schemes that have been listed in Table 3. Firstly, we compared the heatmap among different columns (training strategies): robust transformer (II) boosts the ability to capture long-range traces compared to the baseline (I). (III) compared with (II), the LFE module can push the model to locate more potential manipulation areas. In particular, (IV) relative to (III), our SAS technique enhances these candidate regions by exploring more regions of interest. Secondly, in comparison to various rows: It is commonly assumed that the most useful portions to discern are the mouth, nose, and eyes.

**Cluster Visualization of Feature Map.** We visualized the features generated by different models on the same FF++ test set by using the t-SNE [19]. As Fig. 5 shows, each color corresponds to a specific type of synthetic technique. We observe that the features learned by the shuffle transformer for each forgery method are concentrated in their respective regions and are not tightly grouped together. This phenomenon, on the one hand, indicates that different manipulations have various characteristic distributions, and on the other hand, shows that the shuffle transformer will separate fake data created by different forgery types even if we treated all fake samples as one class in the training stage. It clearly reveals that the features which shuffle transformer extracts contain the unique artifacts of each forgery algorithm, affecting its generalization ability. The feature distribution of different manipulations becomes rather compact due to the establishment of robust transformer and LFE. Moreover, owing to the introduction of our SAS mechanism, the fake sample are more mixed together, which proves that the TransDFD network can learn more general representations for each forgery type.

## 5 Conclusion

In this paper, we design a lightweight and robust network using transformers, namely, TransDFD, which applies fine-grained local and global feature learning for deepfake detection. We propose a robust transformer to extract the patch-level local and global embeddings via exploring intra-patch locally-enhanced relations and inter-patch locally-enhanced global relationships. We build a plug-and-play SAS method to identify salient forgery representations without increasing computational complexity, which enhances the performance of transformer-based models. The experiments on FF++, DFDC, and DFD demonstrate that we achieve state-of-the-art performance with few parameters and computational costs. The limitation of our model is that generalization ability needs to be further strengthened. In the future, we intend to explore self-supervised learning to extract critical information from complex datasets containing multiple manipulation techniques.

## References

1. Deepfake. <https://github.com/deepfakes/>. Accessed 03 Sep 2020
2. Faceswap. <https://github.com/MarekKowalski/FaceSwap>. Accessed 03 Sep 2020
3. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1800–1807 (2017). <https://doi.org/10.1109/CVPR.2017.195>
4. Dang, H., Liu, F., Stehouwer, J., Liu, X., Jain, A.K.: On the detection of digital face manipulation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5780–5789, June 2020. <https://doi.org/10.1109/CVPR42600.2020.00582>
5. Dolhansky, B., et al.: The deepfake detection challenge dataset (2020)
6. Dosovitskiy, A., et al.: An image is worth 16 x 16 words: transformers for image recognition at scale. In: International Conference on Learning Representations, Austria (2021)
7. Fan, Z., et al.: Mask attention networks: rethinking and strengthen transformer. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1692–1701. Association for Computational Linguistics, June 2021
8. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
9. Güera, D., Delp, E.J.: Deepfake video detection using recurrent neural networks. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6, November 2018. <https://doi.org/10.1109/AVSS.2018.8639163>
10. Huang, L., Yuan, Y., Guo, J., Zhang, C., Chen, X., Wang, J.: Interlaced sparse self-attention for semantic segmentation. arXiv preprint [arXiv:1907.12273](https://arxiv.org/abs/1907.12273) (2019)
11. Huang, Z., Ben, Y., Luo, G., Cheng, P., Yu, G., Fu, B.: Shuffle transformer: rethinking spatial shuffle for vision transformer. arXiv preprint [arXiv:2106.03650](https://arxiv.org/abs/2106.03650) (2021)
12. King, D.: dlib 19.22.1 (2021). <https://pypi.org/project/dlib/>. Accessed 29 Aug 2021
13. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations (ICLR). San Diego, CA, USA, Conference Track Proceedings, May 2015
14. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: 2nd International Conference on Learning Representations (ICLR), pp. 14–16 (2014)
15. Kumar, P., Vatsa, M., Singh, R.: Detecting face2face facial reenactment in videos. In: 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 2578–2586 (2020). <https://doi.org/10.1109/WACV45572.2020.9093628>
16. Li, L., et al.: Face x-ray for more general face forgery detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5000–5009 (2020). <https://doi.org/10.1109/CVPR42600.2020.00505>
17. London, U.C.: Deepfakes’ ranked as most serious AI crime threat (2021). <https://www.sciencedaily.com/releases/2020/08/200804085908.htm>. Accessed 01 May 2021
18. Luo, Y., Zhang, Y., Yan, J., Liu, W.: Generalizing face forgery detection with high-frequency features. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16312–16321, June 2021. <https://doi.org/10.1109/CVPR46437.2021.01605>

19. Van der Maaten, L., Hinton, G.: Visualizing data using T-SNE. *J. Mach. Learn. Res.* **9**(11), 2579–2605 (2008)
20. Mao, X., et al.: Towards robust vision transformer. arXiv preprint [arXiv:2105.07926](https://arxiv.org/abs/2105.07926) (2021)
21. Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Niessner, M.: Faceforensics++: learning to detect manipulated facial images. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1–11 (2019). <https://doi.org/10.1109/ICCV.2019.00009>
22. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 618–626, October 2017. <https://doi.org/10.1109/ICCV.2017.74>
23. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 2, pp. 464–468, June 2021
24. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations, May 2015
25. Thies, J., Zollhöfer, M., Nießner, M.: Deferred neural rendering: image synthesis using neural textures. *ACM Trans. Graph. (TOG)* **38**(4), 1–12 (2019)
26. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2face: real-time face capture and reenactment of RGB videos. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2387–2395, June 2016. <https://doi.org/10.1109/CVPR.2016.262>
27. Wang, C., Deng, W.: Representative forgery mining for fake face detection. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14918–14927. Nashville, TN, USA (2021). <https://doi.org/10.1109/CVPR46437.2021.01468>
28. Wodajo, D., Atnafu, S.: Deepfake video detection using convolutional vision transformer. arXiv preprint [arXiv:2102.11126](https://arxiv.org/abs/2102.11126) (2021)
29. Zhao, H., Wei, T., Zhou, W., Zhang, W., Chen, D., Yu, N.: Multi-attentional deepfake detection. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2185–2194 (2021). <https://doi.org/10.1109/CVPR46437.2021.00222>
30. Zhou, P., Han, X., Morariu, V.I., Davis, L.S.: Two-stream neural networks for tampered face detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1831–1839, July 2017. <https://doi.org/10.1109/CVPRW.2017.229>