International Journal of Computer Vision
https://doi.org/10.1007/s11263-024-02274-6

DeepFake-Adapter: Dual-Level Adapter for DeepFake Detection

Rui Shao[1] · Tianxing Wu[2] · Liqiang Nie[1] · Ziwei Liu[2]

Received: 24 May 2023 / Accepted: 30 September 2024

Abstract
Existing deepfake detection methods fail to generalize well to unseen or degraded samples, which can be attributed to the over-fitting of low-level forgery patterns. Here we argue that high-level semantics are also indispensable recipes for generalizable forgery detection. Recently, large pre-trained Vision Transformers (ViTs) have shown promising generalization capability. In this paper, we propose the first parameter-efficient tuning approach for deepfake detection, namely **DeepFake-Adapter**, to effectively and efficiently adapt the generalizable high-level semantics from large pre-trained ViTs to aid deepfake detection. Given large pre-trained models but limited deepfake data, DeepFake-Adapter introduces lightweight yet dedicated dual-level adapter modules to a ViT while keeping the model backbone frozen. Specifically, to guide the adaptation process to be aware of both global and local forgery cues of deepfake data, **1)** we not only insert **Globally-aware Bottleneck Adapters** in parallel to MLP layers of ViT, **2)** but also actively cross-attend **Locally-aware Spatial Adapters** with features from ViT. Unlike existing deepfake detection methods merely focusing on low-level forgery patterns, the forgery detection process of our model can be regularized by generalizable high-level semantics from a pre-trained ViT and adapted by global and local low-level forgeries of deepfake data. Extensive experiments on several standard deepfake detection benchmarks validate the effectiveness of our approach. Notably, DeepFake-Adapter demonstrates a convincing advantage under cross-dataset and cross-manipulation settings.

Keywords DeepFake Detection · Parameter-Efficient Transfer Learning · Generalization Ability · Adapter

1 Introduction

With recent advances in deep generative models, increasing hyper-realistic face images or videos can be readily generated, which can easily cheat human eyes. This leads to serious misinformation and fabrication problems in politics (Shao et al., 2023, 2024, 2022, b, 2023), entertainment and society once these techniques are maliciously abused. This threat is known as *DeepFake*.

To address this security issue, various deepfake detection methods have been proposed and obtain promising performance when training and testing forgery data are from identical manipulation types with good quality. Nevertheless, their performance degrades once countering unseen or low-quality forgeries (Luo et al., 2021; Chai et al., 2020; Shao et al., 2022a, 2020). This may be because most of existing deepfake detection methods merely focus on exploiting low-level forgery features from local textures (Chen et al., 2021; Gu et al., 2022; Zhao et al., 2021; Liu et al., 2020; Shao et al., 2019, 2018), blending boundary (Li et al., 2020), or frequency information (Li et al., 2021; Qian et al., 2020). These features have the following commonness in practice: **1)** different forgeries tend to have quite distinct low-level characteristics and thus testing data with unseen forgery types would present quite distinct forgery patterns compared to training data and **2)** a portion of low-level forgery patterns are likely to be altered and covered by post-processing steps such as compression, blur and noise in

Communicated by Maja Pantic.

✉ Ziwei Liu
ziwei.liu@ntu.edu

Rui Shao
shaorui@hit.edu.cn

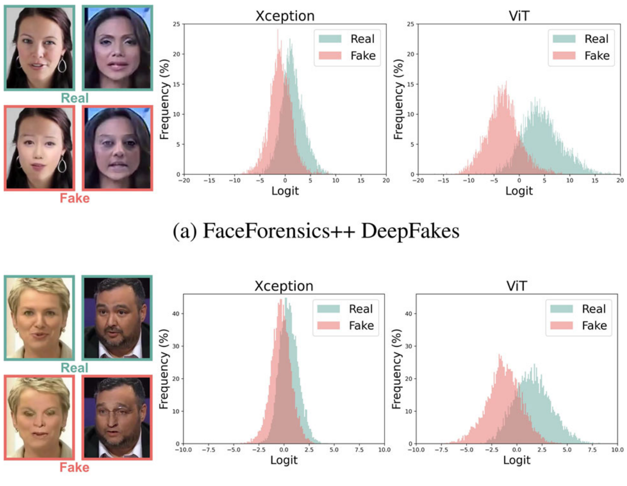
Tianxing Wu
twu012@ntu.edu.sg

Liqiang Nie
nieliqiang@gmail.com

[1] School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China

[2] S-Lab, Nanyang Technological University, Singapore, Singapore

Published online: 24 January 2025



(a) FaceForensics++ DeepFakes

(b) FaceForensics++ FaceSwap

Fig. 1 Example images and distributions of pre-trained Xception and ViT features after linear-probe on **a**DeepFake and **b** FaceSwap splits of FaceForensics++ dataset

low-quality data. These factors degenerate the generalization ability of extracted forgery representations. To address these issues, this paper explores high-level semantics to facilitate a generalizable deepfake detection. In particular, as shown in example images of Fig. 1, we can observe that apart from distinct low-level patterns such as textures between real and fake faces, some generic high-level semantics of real faces such as face style and shape are also altered by some face manipulation methods (*e.g.,* DeepFake and FaceSwap in FaceForensics++ dataset (Rossler et al., 2019)). Thus, these high-level semantics could be exploited for deepfake detection as they are robust to variation of low-level features.

Recently, Vision Transformer (ViT) (Dosovitskiy et al., 2020) and its variants have demonstrated remarkable success in a broad range of computer vision tasks. Various large ViTs pre-trained on massive labeled data are able to learn representations with rich semantics. We preliminarily verify the efficacy of high-level semantic features of large pre-trained ViT for deepfake detection in Fig. 1. Following the setting of linear-probe evaluation in Radford et al. (2021), we compare the linear separability regarding real/fake faces of FaceForensics++ dataset (Rossler et al., 2019) based on features extracted by Xception (Chollet, 2017) pre-trained on ImageNet-1K (Deng et al., 2009) and ViT-Base (Dosovitskiy et al., 2020) pre-trained on ImageNet-21K, respectively. As illustrated in Fig. 1, high-level semantic features from both pre-trained Xception and ViT have the potential to discriminate face forgeries through a simple linear-probe. Furthermore, the separation between real/fake distributions of ViT features are substantially larger than that of Xception features on both manipulation types. These observations demonstrate that **1)** high-level semantic features are useful for deepfake detection and **2)** features from larger pre-trained ViT model are more effective for deepfake detection. This motivates us to dig the power of large pre-trained ViTs for our task.

A straightforward way to adapt the pre-trained ViT for deepfake detection is full fine-tuning (full-tuning) with face forgery data. However, the performance of full-tuning would be severely affected by two factors **1)** given a large volume of pre-trained ViT's parameters (*e.g.,* ViT-Base (Dosovitskiy et al., 2020) with 85.8M parameters) and a relatively smaller amount of deepfake detection data, full-tuning is very likely to result in over-fitting and thus damages the generalization ability of ViT and **2)** as proved in Kumar et al. (2022), full-tuning could distort pre-trained features and leads to worse performance in the presence of large distribution shift. Therefore, to effectively and efficiently adapt generalizable high-level semantics from large pre-trained ViTs to deepfake detection, this paper proposes to explore a fast adaptation approach, namely **DeepFake-Adapter**, in a parameter-efficient tuning manner. DeepFake-Adapter allows a small amount (16.9M, 19% of ViT-Base parameters. Refers to Table 1) of model parameters, *i.e.,* adapter modules, to be trained whereas the vast majority of pre-trained parameters in the model backbone are kept frozen.

Notably, DeepFake-Adapter consists of dual-level modules. First, to adapt ViT with global low-level features, we insert **Globally-aware Bottleneck Adapters (GBA)** in parallel to Multilayer Perception (MLP) layers of the pre-trained ViT. It explores global low-level forgeries, *e.g.,* blending boundary (Li et al., 2020), in a bottleneck structure. Second, to capture more local low-level forgeries in the adaption process, *e.g.,* local textures (Chen et al., 2021; Gu et al., 2022; Zhao et al., 2021; Liu et al., 2020), we devise **Locally-aware Spatial Adapters (LSA)** to extract local low-level features and lead them to interact with features from the pre-trained ViT via a series of cross-attention. In this way, the forgery detection is regularized by generalizable high-level semantics from a pre-trained ViT and adapted with global and local low-level forgeries by the dual-level adapter. This organic interaction between high-level semantics and global/local low-level forgeries contributes to better generalizable forgery representations for deepfake detection. Main contributions of our paper are:

- We argue that high-level semantics of large pre-trained ViTs could be beneficial for deepfake detection. To make use of these semantics, we are the first work to introduce the *adapter* technique into the field of deepfake detection, which fast adapts a pre-trained ViT for our task.
- We propose a novel **DeepFake-Adapter**, which is a dual-level adapter composed of **Globally-aware Bottleneck Adapters (GBA)** and **Locally-aware Spatial Adapters (LSA)**. **DeepFake-Adapter** can effectively adapt a pre-

Table 1 Configurations of pre-trained ViT and proposed Deepfake-Adapter

Settings of ViT (Dosovitskiy et al., 2020)					Settings of GBA			Settings of LSA			Total Param
Blocks	Width	MLP	Heads	#Param	N	Width	#Param	N	Heads	#Param	
12	768	3072	12	85.8M	12	64	1.19M	3	6	15.73M	102.7M

trained ViT by enabling high-level semantics from ViT to organically interact with global and local low-level forgeries from adapters. This contributes to more generalizable forgery representations for deepfake detection.

- Extensive quantitative and qualitative experiments demonstrate the superiority of our method for deepfake detection. Notably, DeepFake-Adapter outperforms the full-tuning adaptation method by only tuning less than 20% of all model parameters. We hope that our approach can facilitate future research on generalizable deepfake detection in the era of larger vision models.

2 Related Work

2.1 DeepFake Detection

Current deepfake detection methods can be mainly categorized into spatial-based and frequency-based forgery detection. The majority of spatial-based deepfake detection methods pay attention to capturing low-level visual cues from the spatial domain. The blending boundary caused by face forgery operations is detected as the visual artifacts for deepfake detection (Li et al., 2020). Various local textures (Chen et al., 2021; Gu et al., 2022; Zhao et al., 2021; Zhu et al., 2021; Liu et al., 2020) are intensively analyzed and explored to highlight the appearance differences between real and forged faces. Besides, direct light (Zhu et al., 2021) is disentangled by a 3D decomposition method and fused with other features using a two-stream network for forgery detection. Patch diffusion (Zhang et al., 2022) and patch inconsistency (Zhao et al., 2021) are also studied to explore the distinct correlation consistency among local patch features between real and forgery faces. Moreover, motion artifacts are dig out from mouth movements as the face forgery patterns by fine-tuning a temporal network pre-trained on lipreading (Haliassos et al., 2021). This method targets at detecting fake videos based on mouth movements without overfitting to low-level, manipulation-specific artefacts. RealForensics (Haliassos et al., 2022) is also another work to exploit generalizable high-level temporal features by studying the natural correspondence between the visual and auditory modalities based on a self-supervised cross-modal manner. In addition, noise characteristics are exploited as the forgery clues in works (Gu et al., 2022; Zhou et al., 2017).

On the other hand, some methods focus on frequency domain for detecting spectrum artifacts. High-frequency part of Discrete Fourier Transform (DFT) (Durall et al., 2019; Dzanic et al., 2020) are extracted to detect distinct spectrum distributions and characteristics between real and fake images. Local frequency statistics based on Discrete Cosine Transform (DCT) are exploited by F^3-Net (Qian et al., 2020) to mine forgery cues. Up-sampling artifacts in phase spectrum are explored by a Spatial-Phase Shallow Learning method (Liu et al., 2021). To capture generalizable forgery features, high-frequency features are integrated with regular RGB features with a two-stream model (Luo et al., 2021). What's more, a frequency-aware discriminative feature learning framework is proposed to perform metric learning in frequency features (Li et al., 2021).

Most of the above deepfake detection methods only study low-level spatial or frequency artifacts. Instead, this paper performs interaction between high-level semantics from a large pre-trained ViT and dual levels of forgeries from DeepFake-Adapter, unveiling better generalizable forgery representations. -5 mm

2.2 Parameter-Efficient Transfer Learning

Parameter-efficient tuning methods have drawn increasing attention starting from the natural language processing (NLP) community. Unlike most of the popular transfer learning methods such as full-tuning and linear-probe (Zhuang et al., 2020), parameter-efficient tuning methods only need to train a small portion of model parameters in consideration of the rapid increase in model size of large pre-trained language models (Li et al., 2024; Chen et al., 2024; Shen et al., 2024; Ye et al., 2024). Prompt learning (Liu et al., 2023; Lester et al., 2021) is wide-used in NLP which prepends learnable tokens into the input text. Adapter (Houlsby et al., 2019) and LoRA (Hu et al., 2022) add tiny learnable modules into NLP transformers. Some follow-up parameter-efficient tuning works in the computer vision field (Jia et al., 2022; Chen et al., 2022b, 2023a, b) have also been proposed recently. This paper is the first work to introduce the *adapter* technique into deepfake detection with a dedicated dual-level adapter.

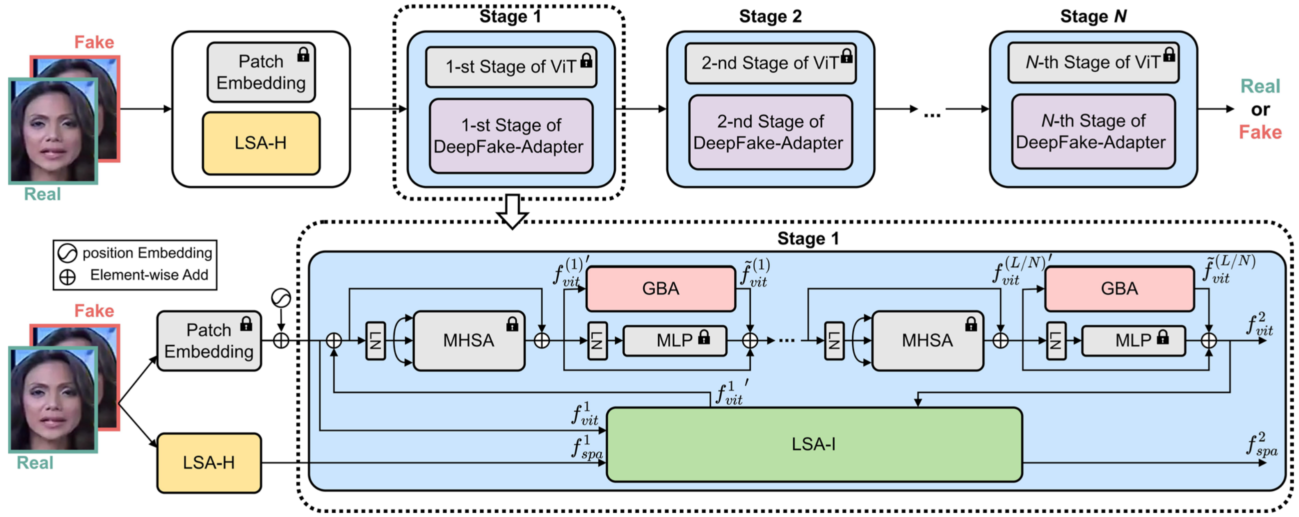


Fig. 2 Overall architecture of proposed model. The model consists of N stages. Each stage contains MHSA and MLP layers of pre-trained ViT, GBA and LSA (LSA-H and LSA-I) of proposed DeepFake-Adapter

3 Our Approach

3.1 Overview

The overall architecture of the proposed network is illustrated in Fig. 2. As depicted in the first row of Fig. 2, the whole network is composed of N stages. Each stage contains one stage of pre-trained ViT whose parameters are frozen during the training, and one stage of deepfake-adapter with trainable parameters for fast adaptation. Moreover, the patch embedding layer of ViT is also frozen and the head part of the proposed Locally-aware Spatial Adapter (LSA-H) is inserted at the beginning of the network.

Specifically, we take Stage 1 as an example shown in the second row of Fig. 2. Given a pre-trained ViT with total L blocks (each block consists of a Multi-Head Self-Attention (MHSA) layer and a MLP layer), we evenly group these blocks into N stages and thus the above one stage of ViT contains L/N blocks. The corresponding stage of DeepFake-Adapter consists of L/N Globally-aware Bottleneck Adapters (GBA) and one interaction part of Locally-aware Spatial Adapter (LSA-I), which are organically interacted with ViT for adaptation. Details of each module in the whole model are introduced in the following sections.

3.2 Vanilla ViT

We adopt a pre-trained vanilla ViT as the frozen backbone of our network. As mentioned above, it basically consists of a patch embedding layer and following L blocks. Given an input image $x \in \mathrm{R}^{3\times H\times W}$, we feed it into the patch embedding layer of ViT. It firstly divides the image into non-overlapping $P \times P$ patches and then flattens them into sequential patches $x_p \in \mathrm{R}^{K\times(P^2C)}$, where (H, W) is the resolution of the input image; C is the number of channels; (P, P) is the resolution of each image patch, and $K = HW/P^2$ is the resulted number of patches. All of these image patches are projected to D-dimensional embedding and added with position embedding. This produces the patch embedding of ViT as the input of Stage 1, denoted as $f_{vit}^{1} \in \mathrm{R}^{(P^2C)\times D}$. After that, the patch embedding passes through MHSA layers and MLP layers in every block of ViT to carry out a series of self-attention. Specifically, MHSA in l-th block of ViT is performed on normalized query (Q), key (K), and value (V) features as follows,

$$\begin{aligned} f_{vit}^{(l)'} &= \text{Attention}(Q = \widehat{f_{vit}^{(l)}}, K = \widehat{f_{vit}^{(l)}}, V = \widehat{f_{vit}^{(l)}}) \\ &= \text{Softmax}(K^T Q/\sqrt{D})V \end{aligned} \qquad (1)$$

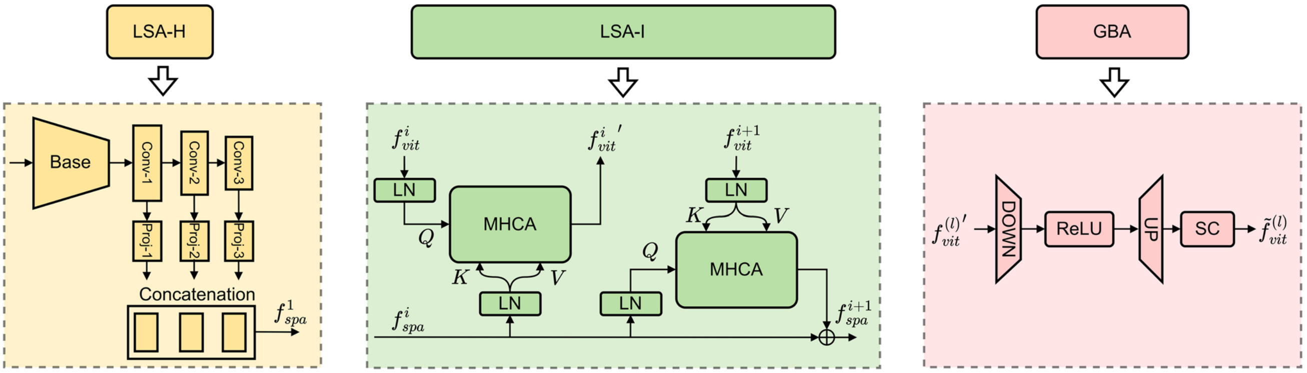
where $\widehat{f_{vit}^{(l)}} = \text{LN}(f_{vit}^{(l)})$, which is the feature normalized by the LayerNorm layer (Ba et al., 2016) as the input of MHSA layer in l-th block. Its output $f_{vit}^{(l)'}$ is then fed into the following MLP layer as follows,

$$f_{vit}^{(l+1)} = \text{MLP}(\widehat{f_{vit}^{(l)'}}) + f_{vit}^{(l)'} \qquad (2)$$

where $f_{vit}^{(l+1)}$ is the output of l-th block of ViT.

3.3 Globally-aware Bottleneck Adapter

Considering that MHSA layers of ViT tend to extract global features (Dosovitskiy et al., 2020), we insert Globally-aware Bottleneck Adapter (GBA) after each MHSA layer and in



(a) Head part of Locally-aware Spatial Adapter (LSA-H)

(b) Interaction part of Locally-aware Spatial Adapter (LSA-I) in stage i

(c) Globally-aware Bottleneck Adapter (GBA) in l-th block of ViT

Fig. 3 Details of GBA and LSA of proposed DeepFake-Adapter. **a** Head part and **b** Interaction part of LSA capture local low-level forgeries that interact with features from pre-trained ViT via a series of cross-attention. **c** GBA adapts the pre-trained ViT with global low-level forgeries in a bottleneck structure

parallel to MLP layers of ViT as illustrated in second row of Fig. 2. It attempts to adapt the pre-trained ViT with more global low-level forgery features, such as blending boundary (Li et al., 2020). Specifically, inspired by (Chen et al., 2022b; He et al., 2022), as shown in Fig. 3 (c), GBA is devised as a bottleneck structure in purpose of saving parameters for fast adaptation, which consists of a down-projection linear layer (DOWN) and an up-projection linear layer (UP). In addition, a ReLU layer (Agarap, 2018) is incorporated between two projection layers for non-linear transformation. To adaptively weigh the importance of global low-level features in the adaptation, one more learnable scale function (SC) is added after two projection layers. The whole adaptation process of GBA is as follows,

$$\tilde{f}_{vit}^{(l)} = \mathrm{SC} \cdot \mathrm{UP} \cdot \mathrm{ReLU}(\mathrm{DOWN}(f_{vit}^{(l)\prime})) \tag{3}$$

where $\tilde{f}_{vit}^{(l)}$ is the adapted global low-level features from corresponding GBA in l-th block of ViT, which can be further fused with the original output of MLP layer as follows,

$$f_{vit}^{(l+1)} = \mathrm{MLP}(\widehat{f_{vit}^{(l)\prime}}) + f_{vit}^{(l)\prime} + \tilde{f}_{vit}^{(l)} \tag{4}$$

3.4 Locally-aware Spatial Adapter

It is well known that ViT has much less image-specific inductive bias, *e.g.,* spatial locality, than Convolutional Neural Networks (CNNs) (Dosovitskiy et al., 2020). This makes a ViT less likely to differentiate local low-level features between real and fake faces. To address this issue, we introduce Locally-aware Spatial Adapter (LSA) in this section, which is composed of head and interaction parts. It aims to adapt more local low-level forgery features, such as local textures (Chen et al., 2021; Gu et al., 2022; Zhao et al., 2021; Liu et al., 2020) for our task.

Head part (LSA-H). Inspired by recent works (Yuan et al., 2021; Wu et al., 2021) that integrate convolutional operations of CNNs into a ViT, we introduce the convolution-based head part of LSA. It locates in parallel to the patch embedding layer of ViT, attempting to capture more local low-level forgeries of input images from the beginning. To be specific, as depicted in Fig. 3 (a), following the structure of ResNet (He et al., 2016), we employ a standard CNN as the base network to extract base feature maps, which consists of three Convolution-BatchNorm-ReLU blocks and a max-pooling layer. Then, three similar convolutional blocks are used to extract several intermediate feature maps. They are composed of various pyramid resolutions, $1/r_1$, $1/r_2$, and $1/r_3$ resolutions, corresponding to the size of original input images. After that, all of them are projected into the same dimension D via three projectors and concatenated into a feature vector denoted as $f_{spa}^1 \in \mathrm{R}^{(\frac{HW}{r_1^2}+\frac{HW}{r_2^2}+\frac{HW}{r_3^2})\times D}$. Based on this, LSA-H aggregates features with diverse spatial resolutions, capturing fine-grained and rich local forgeries.

Interaction part (LSA-I). Given the aggregated features f_{spa}^1, we intend to enable the whole adaption process to sufficiently be aware of local low-level forgeries captured from these features. To this end, as illustrated in the second row of Fig. 2, we devise the interaction part of LSA which leads these features (*e.g.,* f_{spa}^1 in Stage 1) to interact with features from the beginning and end of ViT in each stage (*e.g.,* f_{vit}^1 and f_{vit}^2 in Stage 1). In greater detail, in Stage i, the first interaction is performed by a multi-head cross-attention (MHCA) between feature f_{spa}^i and the feature from the beginning of ViT f_{vit}^i, as depicted in Fig. 3 (b). Here, we treat normalized f_{vit}^i as query and normalized f_{spa}^i as key and value in this MHCA as follows,

$$f_{vit}^{i}{}' = f_{vit}^{i} + \text{Attention}(Q = \widehat{f_{vit}^{i}}, K = \widehat{f_{spa}^{i}}, V = \widehat{f_{spa}^{i}}) \tag{5}$$

where $f_{vit}^{i}{}'$ is the result of the first interaction. It will be fed back into ViT and go through the following MHSA, MLP layers and GBA modules in ViT. This adaptation process injects local low-level features into the forward process of ViT. Once obtaining the feature (denoted as f_{vit}^{i+1}) through the whole forward process of ViT in Stage i, we perform the second interaction at the end of ViT by conducting MHCA between f_{spa}^{i} and f_{vit}^{i+1}. We switch the K, Q, V by taking normalized f_{spa}^{i} as query and normalized f_{vit}^{i+1} as key and value in this MHCA as follows,

$$f_{spa}^{i+1} = f_{spa}^{i} + \text{Attention}(Q = \widehat{f_{spa}^{i}}, K = \widehat{f_{vit}^{i+1}}, V = \widehat{f_{vit}^{i+1}}) \tag{6}$$

where f_{spa}^{i+1} is the updated low-level features which will be forwarded to interaction with new features of ViT f_{vit}^{i+1} in the next stage. As such, the influence of local low-level features regarding face forgery data could be further strengthened in the adaptation process at the end of each stage.

After we extract the feature f_{spa}^{N+1} through N stages of our model, we feed it into a classifier based on a linear layer CLS and calculate a cross-entropy loss as follows,

$$\mathcal{L} = \mathbf{H}(\text{CLS}(f_{spa}^{N+1}), y) \tag{7}$$

where y are labels for corresponding samples and $\mathbf{H}(\cdot)$ is the cross-entropy function. We train all the above adapters with this loss function $\mathcal{L}$ in an end-to-end manner.

In summary, through making high-level semantic features from the pre-trained ViT interacted with global low-level features in Eq. 4 and local low-level features in Eq. 5-6, our model based on such dual-level adaptation could exploit better generalizable forgery representations.

3.5 Merits and Limitations

As aforementioned, the proposed DeepFake-Adapter performs fast adaption for a large pre-trained ViT via GBA and LSA simultaneously. This dual-level adaptation contributes to a better discriminative and generalizable deepfake detection. Furthermore, this adaptation is devised in a parameter-efficient tuning manner. We only need to train remarkably fewer parameters in GBA and LSA (less than 20% of the original large pre-trained ViT). This makes our method easily scale up to various deepfake datasets and deployed with affordable GPU machines for training.

On the other hand, we need to point out one main limitation of the proposed method. Since our approach is designed specifically to adapt a large pre-trained ViT, the detection process is regularized by high-level semantics from it. While its advantage facilitates a generalizable deepfake detection, it brings a negative impact that the current model is likely to be unavailable for detecting the forgery based on face reenactment (*e.g.,* Face2Face and NeuralTexture in FaceForensics++ dataset). This is because these types of face manipulation only present very minor low-level forgery patterns without much modification on high-level semantics. We argue that it is impractical to address all types of deepfake manipulation in a single model. Consequently, this paper mainly focuses on detecting one of the most popular and the highest risky face forgery methods based on face swapping.

4 Experiments

4.1 Experimental Settings

Datasets. Experiments are conducted on several existing public deepfake datasets, *e.g.,* **FaceForensics++ (FF++)** (Rossler et al., 2019), **Celeb-DF** (Li et al., 2020), **Deepfake Detection Challenge (DFDC)** (Dolhansky et al., 2020), and **DeeperForensics−1.0 (DF1.0)** (Jiang et al., 2020). As one of the most widely-used datasets in deepfake detection, FF++ collects 1,000 original videos, and 4000 fake videos generated by corresponding four face manipulation techniques: Deepfakes (DF) (Li et al., 2020), Face2Face (F2F) (Thies et al., 2016), FaceSwap (FS) (Thies et al., 2016), and NeuralTextures (NT) (Thies et al., 2019). In contrast, most of the manipulation types in Celeb-DF, DFDC, and DF1.0 datasets are based on face swapping. Considering factors mentioned in section of Merits and Limitations and the most prevailing face forgery in practice, we mainly train our model on manipulation types of DF and FS in FF++ dataset. This evaluates the generalization ability of our method on forgeries related to face swapping. Moreover, we adopt both c23 (high-quality) and c40 (low-quality) versions of FF++ data in our experiments, examining deepfake detection on forgeries with various qualities.

4.2 Evaluation on Discrimination Ability

Implementation Details. We tabulate configurations of the used pre-trained ViT and proposed DeepFake-Adapter in Table 1. We adopt ViT-Base (Dosovitskiy et al., 2020) pre-trained on ImageNet-21K as our frozen backbone in this paper, which is equipped with 12 blocks. These blocks are evenly split into 3 stages and thus there exist 4 blocks of ViT in each stage. In each block, every MHSA layer has 12 heads

Table 2 Structure details of all components in LSA-H

Layer	Chan./Stri	Out.Size
Base		
Input: image		
conv0-1	64/2	112
conv0-2	64/1	112
conv0-3	64/1	112
pool0-1	-/2	56
Conv-1		
Input: pool0-1		
conv1-1	128/2	28
Conv-2		
Input: conv1-1		
conv2-1	256/2	14
Conv-3		
Input: conv2-1		
conv3-1	256/2	7
Proj-1		
Input: conv1-1		
conv4-1	768/1	28
Proj-2		
Input: conv2-1		
conv5-1	768/1	14
Proj-3		
Input: conv3-1		
conv6-1	768/1	7

and the embedding size of every MLP layer is 3072. Since each MLP layer of ViT is paralleled with a GBA, 12 GBA are inserted in total, where the embedding size of bottleneck is 64. Moreover, we place one LSA in each stage of our network and thus the total number of LSA is 3, where each MHCA has 6 heads. In all, parameter numbers of ViT, GBA and LSA are 85.8M, 1.19M, and 15.73M. This implies the trainable dual-level adapter is much smaller than pre-trained ViT (only **19.72%** of pre-trained ViT parameters), which statistically validates the proposed model is parameter-efficient.

All of our experiments are performed on 4 NVIDIA V100 GPUs with PyTorch framework (Paszke et al., 2017). For the training schedule, we employ a 10-epochs warm-up strategy. The initial learning rate is set as $1e-1$, with a cosine learning rate decay. We use the SGD momentum optimizer with weight decay set as $1e-4$. The batch size is set as 64.

We also provide the structure details of LSA-H (as shown in Fig. 3 (a)) in Table 2. Specifically, each convolutional layer in blocks of Base Network, Conv-1, Conv-2, and Conv-3 is followed by a batch normalization layer and a ReLU activation function.

Evaluation Metrics. We evaluate the proposed method and other baselines using the most commonly used metrics in related works (Cao et al., 2022; Dong et al., 2022; Chen et al., 2022a; Li et al., 2020; Chen et al., 2021; Qian et al., 2020; Zhao et al., 2021), including Accuracy (ACC), Area Under the Receiver Operating Characteristic Curve (AUC), and Equal Error Rate (EER).

In this section, to examine the discrimination ability of the proposed method, we carry out an intra-dataset evaluation where training and test data are from the same FF++ dataset. Following (Gu et al., 2022), we compare the proposed method with a few state-of-the-art (SOTA) approaches applied in deepfake detection. The evaluation is performed on both c23 (high-quality) and c40 (low-quality) data versions and we tabulate the comparison results in Table 3. From Table 3, it can be seen that in the easier case of evaluation on c23, some latest baselines have already achieved saturated performance over 99% AUC when dealing with four manipulation types, especially in DF and FS forgeries. In such a case, although the proposed method (Ours) is not able to reach 100% detection accuracy, it still obtains the second best performance compared to other baselines. Furthermore, in the harder case of c40, the proposed method substantially outperforms other baselines by 1%-2% AUC improvement in DF, FS and F2F respectively and obtains comparable results in NT. These experimental results demonstrate that the proposed method not only performs well in detection of high-quality forgeries but also is discriminative and robust in detecting low-quality forgeries filled with blur, compression and noise. This verifies exploiting high-level semantics improves the discrimination and robustness of forgery detection in presence of various post-processing. To further improve the discriminative ability in intra-manipulation scenarios for all types of deepfake detection, we unfreeze the self-attention layer in the first block of ViT for training, which only increases a small number of trainable parameters (increase from 16.92M to 19.28M). We denote this version of the proposed method as Ours* in Table 3. It can be seen from Table 3 that two versions of the proposed method can obtain the best or the second-best results in all settings. Notably, Ours* achieves the SOTA performance in 6/8 benchmarks and performs better than SIM under NT of c40. This further indicates that the proposed method can discriminate well all types of face manipulation methods, especially for the harder cases in c40 of FF++.

4.3 Evaluation on Generalization Ability

Cross-Manipulation Evaluation. To evaluate the generalization ability of our method on unseen forgeries, following RECCE (Cao et al., 2022), we first perform cross-manipulation experiments by training and testing on different face manipulation methods in c40 version of FF++ dataset.

Table 3 Performance of intra-dataset evaluation. Best results are in bold. Second-best results are in underline

Methods	FaceForensics++ (c23)				FaceForensics++ (c40)			
	DF	FS	F2F	NT	DF	FS	F2F	NT
ResNet-50 (He et al., 2016)	98.93	99.64	98.57	95.00	95.36	94.64	88.93	87.50
Xception (Chollet, 2017)	98.93	99.64	98.93	95.00	96.78	94.64	91.07	87.14
LSTM (Hochreiter & Schmidhuber, 1997)	99.64	98.21	99.29	93.93	96.43	94.29	88.21	88.21
C3D (Tran et al., 2014)	92.86	91.79	88.57	89.64	89.29	87.86	82.86	87.14
I3D (Carreira & Zisserman, 2017)	92.86	96.43	92.86	90.36	91.07	91.43	86.43	78.57
TEI (Liu et al., 2020)	97.86	97.50	97.14	94.29	95.00	94.64	91.07	90.36
DSANet (Wu et al., 2021)	99.29	99.64	99.29	95.71	96.79	95.36	93.21	91.78
V4D (Zhang et al., 2020)	99.64	99.64	99.29	96.07	97.86	95.36	93.57	92.50
FaceNetLSTM (Sohrawardi et al., 2019)	89.00	90.00	87.00	-	-	-	-	-
Co-motion (Wang et al., 2020)	99.10	98.30	93.25	90.45	-	-	-	-
DeepRhythm (Qi et al., 2020)	98.70	97.80	98.90	-	-	-	-	-
ADD-Net (Zi et al., 2020)	92.14	92.50	83.93	78.21	90.36	80.00	78.21	69.29
S-MIL (Li et al., 2020)	98.57	99.29	99.29	95.71	96.79	94.64	91.43	88.57
S-MIL-T (Li et al., 2020)	99.64	100	9.64	94.29	97.14	96.07	91.07	86.79
STIL (Gu et al., 2021)	99.64	100	99.29	95.36	98.21	97.14	92.14	91.78
SIM (Gu et al., 2022)	**100**	**100**	99.29	6.43	99.28	97.86	95.71	4.28
Ours	99.84	99.76	99.33	95.97	9.57	9.00	7.50	91.25
Ours*	9.85	9.83	**99.67**	**96.52**	**99.65**	**99.20**	**97.61**	**94.30**

Bold values indicate the best results

Table 4 Cross-manipulation evaluation trained with DF and FS

Methods	Train	DF	FS
Xception (Chollet, 2017)	DF	98.44	68.67
RFM (Wang & Deng, 2021)		98.80	72.69
Add-Net (Zi et al., 2020)		98.04	68.61
Freq-SCL (Li et al., 2021)		98.91	66.87
MultiAtt (Zhao et al., 2021)		99.51	67.33
RECCE (Cao et al., 2022)		**99.65**	4.29
Ours		9.57	**79.51**
Xception (Chollet, 2017)	FS	79.54	97.02
RFM (Wang & Deng, 2021)		81.34	98.26
Add-Net (Zi et al., 2020)		72.82	97.56
Freq-SCL (Li et al., 2021)		75.90	98.37
MultiAtt (Zhao et al., 2021)		82.33	98.82
RECCE (Cao et al., 2022)		2.39	8.82
Ours		**88.57**	**99.04**

Bold values indicate the best results

We compare our method with several SOTAs in Table 4. It can be observed from Table 4 that the proposed method achieves better generalization performance on unseen face manipulation methods compared with other competitors, yielding about 5% and 6% AUC gains in two cross-manipulation evaluation settings. At the same time, the proposed method remains very competitive when training and testing data are from identical manipulation types. These results under c40 of FF++ validate that better generalizable forgery representations for deepfake detection can be captured by our method even facing highly post-processing scenarios.

We further tabulate cross-manipulation evaluation with respect to F2F as shown in Table 5. Our model trained with F2F can also achieve second best average generalization performance when testing on unseen manipulation types, with very close average result to RECCE (Cao et al., 2022) and substantially surpassing the other SOTAs. To be specific, the other version of the proposed method (Ours* in Table 5) further improves the cross-manipulation performance and can achieve the best average generalization ability compared to all the other SOTAs. The experimental results in Table 3 and Table 5 demonstrate our model have great potential to deal with all types of face manipulation methods including face reenactment methods like F2F and NT.

Cross-Dataset Evaluation. To verify the generalization ability of our method on unseen forgeries with larger variations, we further conduct cross-dataset evaluations where training and testing data are from different deepfake datasets. Firstly, we perform a normal cross-dataset experiment, where we train deepfake detection models with data of c40 version on FF++, and test them on Celeb-DF and DFDC datasets, respectively. We tabulate the obtained performance of this experiment in Tables 6. Table 6 shows that the proposed method exceeds all the other considered baselines by a large margin in terms of both AUC and EER metrics. In particular, our model can reach 71.74 % and 72.66 % AUC when

Table 5 Performance of cross-manipulation evaluation trained with F2F

Methods	Train	DF	FS	NT	Avg
Xception (Chollet, 2017)	F2F	72.93	64.26	70.48	69.22
RFM (Wang & Deng, 2021)		67.80	64.67	64.55	65.67
Add-Net (Zi et al., 2020)		70.24	59.54	69.74	66.51
Freq-SCL (Li et al., 2021)		67.55	55.35	66.66	63.19
MultiAtt (Zhao et al., 2021)		73.04	65.10	71.88	70.01
RECCE (Cao et al., 2022)		**75.99**	64.53	2.32	0.95
Ours		72.24	7.26	71.37	70.29
Ours*		3.30	**67.73**	**72.39**	**71.14**

Bold values indicate the best results

Table 6 Performance of cross-dataset evaluation

Methods	Train	Celeb-DF		DFDC	
		AUC ↑	EER ↓	AUC ↑	EER ↓
Xception (Chollet, 2017)	FF++	61.80	41.73	63.61	40.58
RFM (Wang & Deng, 2021)		65.63	38.54	66.01	39.05
Add-Net (Zi et al., 2020)		65.29	38.90	64.78	40.23
F3-Net (Qian et al., 2020)		61.51	42.03	64.60	39.84
MultiAtt (Zhao et al., 2021)		67.02	37.90	68.01	37.17
RECCE (Cao et al., 2022)		8.71	5.73	9.06	6.08
Ours		**71.74**	**33.98**	**72.66**	**32.68**

Bold values indicate the best results

Table 7 Cross-dataset evaluation with single manipulation method training

Methods	DF			FS			Avg.
	DFDC	Celeb-DF	DF1.0	DFDC	Celeb-DF	DF1.0	
Xception (Chollet, 2017)	65.4	68.1	61.7	70.8	60.1	60.5	64.4
Face X-ray (Li et al., 2020)	60.9	55.4	66.8	64.6	69.7	79.5	66.1
F3-Net (Qian et al., 2020)	68.2	66.4	65.8	67.9	63.6	65.1	66.1
RFM (Wang & Deng, 2021)	75.8	72.3	71.7	71.4	59.1	71.4	70.2
SRM (Luo et al., 2021)	67.9	65.0	72.0	67.1	64.3	7.1	68.9
SLADD (Chen et al., 2022a)	7.2	3.0	4.2	4.2	**80.0**	69.5	4.6
Ours	**77.6**	**84.7**	**91.2**	**75.9**	3.6	**81.8**	**80.8**

Bold values indicate the best results

tested on Celeb-DF and DFDC respectively, which surpass the SOTA method RECCE by about 3%.

To justify the generalization of the proposed method more comprehensively, we further perform another cross-dataset evaluation by training the model with a single type of manipulation method (*e.g.,* DF and FS) in c23 of FF++ and testing it on the unseen DFDC, Celeb-DF and DF1.0 datasets, following the setting proposed in SLADD (Chen et al., 2022a). We tabulate experimental results in Table 7. It is evident from Table 7 that the proposed method achieves the best performance regarding the cross-dataset task in most cases, by nontrivial margins improvement in some cases such as evaluation on DF1.0. Notably, our model is able to surpass other compared methods by more than 6% AUC averaged across all cases. In all, DeepFake-Adapter can obtain quite promising performance in above two kinds of cross-dataset evaluations. This clearly demonstrates that regularized by generalizable high-level semantics of pre-trained ViT and adapted with global and local low-level forgeries via dual-level adapter, a better generalizable deepfake detection across different datasets can be obtained by our method than other existing deepfake detection methods.

4.4 Experimental Analysis

Comparison of Different Adaptation Methods. To highlight the advantage of DeepFake-Adapter compared to other existing adaptation approaches, we compare it with the most widely-used tuning approaches, *e.g.,* full-tuning and linear-probe (Zhuang et al., 2020), and two latest parameter-efficient adaptation methods named Visual Prompt Tuning (VPT) (Jia et al., 2022) and ViT-Adapter (Chen et al., 2023b)

Table 8 Comparison with adaptation methods

Methods	Train	DF	FS
Full-tuning	DF	9.38	74.25
Linear-probing (Zhuang et al., 2020)		91.41	67.20
VPT (Jia et al., 2022)		99.37	5.93
ViT-Adapter (Chen et al., 2023b)		99.19	73.16
Ours		**99.66**	**76.85**
Full-tuning	FS	7.89	98.31
Linear-probing (Zhuang et al., 2020)		75.49	80.41
VPT (Jia et al., 2022)		86.30	97.25
ViT-Adapter (Chen et al., 2023b)		87.65	8.55
Ours		**88.57**	**99.04**

Bold values indicate the best results

Table 9 Ablation study of Deepfake-Adapter

Modules		Train	DF	FS
GBA	LSA			
✗	✗	DF	91.41	67.20
✗	✓		99.19	73.16
✓	✗		99.33	75.03
✓	✓		**99.57**	**79.51**
✗	✗	FS	75.49	80.41
✗	✓		87.65	98.55
✓	✗		84.90	98.99
✓	✓		**88.57**	**99.04**

Bold values indicate the best results

Table 10 Comparison of different pre-trained weights

Methods	Pre-train	Train	DF	FS
Ours	MAE (He et al., 2022)	DF	99.52	73.12
	Supervised		**99.57**	**79.51**
Ours	MAE (He et al., 2022)	FS	83.97	98.88
	Supervised		**88.57**	**99.04**

Bold values indicate the best results

in Table 8. As analyzed in Introduction, full-tuning a large ViT with limited deepfake data will result in over-fitting and distorting the pre-trained features. This is proved by the experimental results in Table 8, where full-tuning all the parameters of ViT obtains lower performance than our method. Since tuning a small number of parameters (less than 20% of all model parameters) can exceed tuning all the parameters of ViT, DeepFake-Adapter is proved to be both efficient and effective for deepfake detection. In addition, DeepFake-Adapter also substantially outperforms the other two parameter-efficient tuning methods, linear-probe and VPT. To achieve parameter-efficient tuning, only the last layer of whole network is adapted and tuned in linear-probe, while VPT simply prepends trainable tokens in the input space. The comparison with linear-probe and VPT in Table 8 further proves through organically interacting with features from different intermediate layers of the pre-trained ViT, the proposed dual-level adapter can attain more fine-grained and more sufficient fast adaptation. Furthermore, the proposed method also surpasses ViT-Adapter (Chen et al., 2023b). This is because our method constructs a more comprehensive dual-level adaptation, introducing LSA and GBA and thus guiding adaption process to be aware of both global and local forgery cues.

Ablation Study. In this sub-section we investigate the impact of two key adapter modules in our DeepFake-Adapter, GBA and LSA, to the overall performance. The considered components and the corresponding results obtained for each case are tabulated in Table 9. As evident from Table 9, removing either GBA or LSA will degrade the overall performance. This validates that both global low-level features exploited by GBA and local low-level features extracted by LSA promote the discrimination and generalization of deepfake detection. These two modules complement each other to produce overall better performance. In addition, removing both of two adapter modules is equal to linear-prob of pre-trained ViT. It can be observed from Table 9 that adding either GBA or LSA will dramatically boost performance than linear-prob, justifying that any one of the proposed adapters is apparently more effective than linear-prob in terms of fast adaptation for deepfake detection. In other words, two dedicated adapters are critical and necessary to enable the whole model to achieve superior performance.

Comparison of Different Pre-training Weights. This section studies the effect of different pre-training weights for our model. We tabulate comparison results with MAE (He et al., 2022) pre-trained weights in Table 10. Table 10 shows our backbone with supervised pre-training weights improves performance compared to self-supervised pre-training weights by MAE (He et al., 2022). These results indicate supervised pre-training can provide better high-level semantics for deepfake detection and thus more suitable for adaptation. Besides, this experiment suggests our method can attain performance benefits by freely selecting the most suitable pre-training manner without additional training cost.

Comparison of Numbers of Trainable Parameters and Inference Time. We further compare the number of trainable parameters and inference time of our method with some representative baselines in Table 11. The measurement of inference time is performed based on GPU: Tesla V100-PCIE-32GB with 32GB space, and CPU: Intel(R) Xeon(R) Gold 6278C CPU @ 2.60GHz. Comparison results in Tables 3–7 and Table 12 have demonstrated our method outperforms these baselines. Table 11 here further indicates such better performance of our method is achieved by using

Table 11 Comparison of numbers of trainable parameters and inference time

Methods	#Param	Inference Time
MultiAtt (Zhao et al., 2021)	417.52 M	22.81 ms
SRM (Luo et al., 2021)	53.36 M	11.25 ms
Xception (Chollet, 2017)	20.81 M	**4.46 ms**
LipForensics (Haliassos et al., 2021)	35.99 M	12.63 ms
RealForensics (Haliassos et al., 2022)	25.34 M	16.53 ms
Ours	**16.92 M**	0.88 ms
Ours*	9.28 M	0.88 ms

Bold values indicate the best results

Table 12 Robustness to low-level corruptions

Method	Saturation	Contrast	Block	Noise	Blur	Pixel	Compress	Avg
Xception (Chollet, 2017)	99.3	98.6	**99.7**	53.8	60.2	74.2	62.1	78.3
CNN-aug (Wang et al., 2020)	99.3	9.1	95.2	54.7	76.5	91.2	72.5	84.1
Patch-based (Chai et al., 2020)	84.3	74.2	9.2	50.0	54.4	56.7	53.4	67.5
Face X-ray (Li et al., 2020)	97.6	88.5	99.1	49.8	63.8	88.6	55.2	77.5
CNN-GRU (Sabir et al., 2019)	99.0	98.8	97.9	47.9	71.5	86.5	74.5	82.3
LipForensics (Haliassos et al., 2021)	**99.9**	**99.6**	87.4	73.8	96.1	95.6	95.6	92.5
RealForensics (Haliassos et al., 2022)	9.8	**99.6**	98.9	79.7	95.3	**98.4**	97.6	5.6
Ours	97.6	97.2	97.4	6.0	**96.9**	95.8	7.6	95.5
Ours*	97.9	97.5	98.0	**86.3**	6.8	6.2	**98.0**	**95.8**

Bold values indicate the best results

significantly fewer trainable parameters and less inference time (only slightly more than Xception), implying the efficacy and efficiency of the proposed DeepFake-Adapter.

Robustness to low-level corruptions. We follow (Haliassos et al., 2021) to assess robustness to various unseen low-level perturbations. Specifically, following (Haliassos et al., 2021), we compare the proposed method with other SOTA deepfake detection methods trained on FF++ c23 dataset on seven unseen low-level perturbations, such as saturation, contrast, block, noise, blur, pixel, and compress, as illustrated in Table 12. It can be seen that the proposed method (Ours and Ours*) can achieve the best and the second-best performance under most of the low-level perturbations, and Ours* attains the best average performance compared to all the other baselines. This further implies that DeepFake-Adapter aided by high-level semantic understanding capability from large vision models is more robust to unseen low-level corruptions.

Detection of deepfake samples generated by diffusion modes. We further test our model on deepfake samples generated by 5 representative diffusion modes, such as DDPM (Ho et al., 2020a), IDDPM (Quinn & Dhariwal, 2021), ADM (Dhariwal & Nichol, 2021a), PNDM (Liu et al., 2022), and LDM (Rombach et al., 2022). Specifically, our model and Xception (Chollet, 2017) are trained on the FF++ c40 dataset and evaluated under the setting of (Ricker et al., 2024). Note that both our model and Xception are evaluated in a zero-shot testing setup. We conducted testing using the officially provided evaluation scripts. The PD metric refers to the probability of detection at a fixed false alarm rate, which is defined as the true positive rate at a specific false alarm rate. PD@10% indicates the probability of our model detecting fake images while allowing a 10% false alarm rate, a higher value is desirable. The same applies to PD@5% and PD@1%.

We tabulate the comparison between our model and Xception (Chollet, 2017) in Table 13. It can be seen that the proposed model outperforms the baseline method in detecting deepfake samples generated by all the 5 representative diffusion modes, under all evaluation settings. This suggests the proposed method can be applied and generalized well to more realistic deepfake images generated by diffusion models.

Comparison of Different ViT Architectures. This section studies the proposed DeepFake-Adapter with various ViT architectures, in the cross-manipulation evaluation setting same as in Section 4.3. Particularly, we tabulate the cross-manipulation performance of DeepFake-Adapter based on ViT-Base and ViT-Large architectures in Table 14. It shows that DeepFake-Adapter based on various ViT architectures can obtain the best or second-best performance compared to the SOTA method RECCE in both intra- and cross-manipulation settings. This suggests the proposed DeepFake-Adapter with various ViT architectures can simultaneously yield promising performance. Meanwhile, the t-SNE plots of

Table 13 Performance of detection of deepfake samples generated by various diffusion modes

DM	Methods	AP	AUROC	PD@10%	PD@5%	PD@1%
DDPM (Ho et al., 2020b)	Xception (Chollet, 2017)	51.4	50.9	11.9	6.0	1.1
	Ours	**59.9**	**61.9**	**18.4**	**10.6**	**2.6**
IDDPM (Quinn & Dhariwal, 2021)	Xception (Chollet, 2017)	52.8	52.4	13.7	6.6	1.2
	Ours	**61.8**	**64.2**	**20.3**	**11.5**	**2.8**
ADM (Dhariwal & Nichol, 2021b)	Xception (Chollet, 2017)	47.9	47.7	8.3	3.7	0.5
	Ours	**54.5**	**56.9**	**13.2**	**6.6**	**1.1**
PNDM (Liu et al., 2022)	Xception (Chollet, 2017)	48.7	46.8	10.6	5.4	1.1
	Ours	**53.4**	**54.8**	**12.7**	**6.8**	**1.3**
LDM (Robin et al., 2022)	Xception (Chollet, 2017)	49.8	49.3	10.4	5.1	1.0
	Ours	**57.5**	**59.4**	**16.3**	**8.9**	**1.8**

Bold values indicate the best results

Table 14 Performance of cross-manipulation evaluation based on various ViT architectures

Methods	Train	DF	FS
RECCE (Cao et al., 2022)	DF	**99.65**	74.29
Ours (ViT-Base)		99.57	**79.51**
Ours (ViT-Large)		9.62	8.60
RECCE (Cao et al., 2022)	FS	82.39	98.82
Ours (ViT-Base)		**88.57**	9.04
Ours (ViT-Large)		5.08	**99.10**

Bold values indicate the best results

Table 15 Comparison between different numbers of Stages and Blocks in each Stage

#Stages and Blocks	#Param	DF	FS
2 Stages with 6 Blocks	12.19 M	99.34	75.44
6 Stages with 2 Blocks	31.11 M	9.53	8.69
3 Stages with 4 Blocks	16.56 M	**99.57**	**79.51**

Bold values indicate the best results

Training data is DF of FF++ dataset and testing data is DF and FS of FF++ dataset

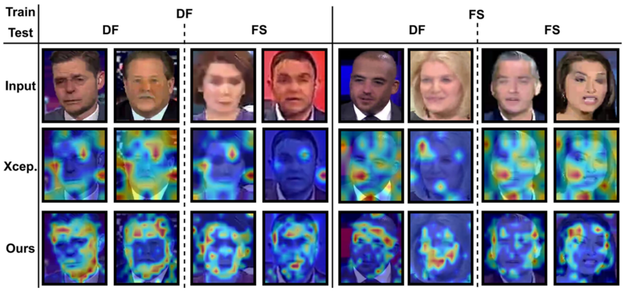


Fig. 4 Comparison of Grad-CAM visualizations between Xception and the proposed model in cross-manipulation evaluation. (Best viewed in color)

different ViT architectures' features are displayed in Fig. 7. All these qualitative and quantitative results validate that the proposed DeepFake-Adapter can be compatible with various ViT architectures for the task of deepfake detection.

Comparison between Different Numbers of Stages and Blocks. This section studies the choice regarding the number of Stages and Blocks. As mentioned above, the proposed DeepFake-Adapter splits the pre-trained ViT into 3 Stages with 4 Blocks in each Stage. This choice is determined by the consideration of both effectiveness and efficiency. The core claim of the proposed DeepFake-Adapter is a parameter-efficient tuning approach for deepfake detection. As shown in Table 1, LSA possesses much more trainable parameters than GBA. Therefore, employing LSA on every block will significantly increase the trainable parameters and thus make the proposed method less parameter-efficient. For this reason, we introduce the concept of Stage and deploy only one LSA in each Stage, which saves many trainable parameters. This is proved by the fact that DeepFake-Adapter outperforms the full-tuning adaptation method by only tuning less than 20% of all model parameters. On the other hand, decreasing the number of GBA and LSA would also affect the performance.

To validate the efficacy of this choice, we tabulate the comparison between different number of Stages and Blocks in Table 15. It follows the setting of cross-manipulation evaluation in Sec. 4.3. As illustrated in Table 15, the setting of 6 Stages with 2 Blocks dramatically increases the number of trainable parameters but still achieves less effective performance. Meanwhile, the setting of 2 Stages with 6 Blocks yields fewer trainable parameters at the cost of degraded performance. In contrast, the setting (3 Stages with 4 Blocks) adopted by DeepFake-Adapter can simultaneously achieve moderate number of trainable parameters and best performance. This indicates the more optimal choice regarding the number of Stages and Blocks by the proposed method.

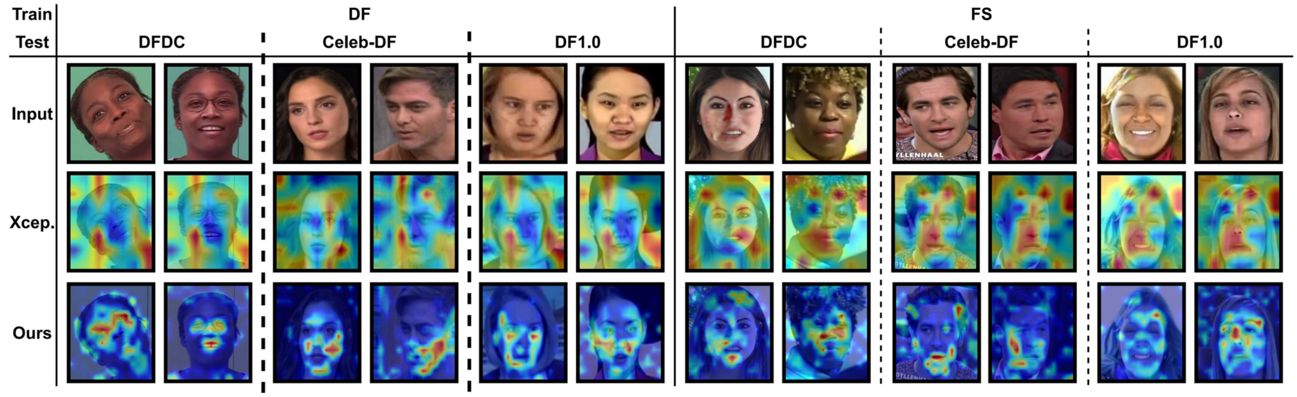


Fig. 5 Comparison of Grad-CAM visualizations between Xception and the proposed model in cross-dataset evaluation among DFDC, Celeb-DF and DF1.0 datasets. (Best viewed in color)

4.5 Visualization

Attention map compared with Xception. To provide a deeper understanding about the decision-making mechanism of our method, we compare Grad-CAM (Selvaraju et al., 2017) visualizations between our model and Xception (Chollet, 2017) on FF++ as shown in Fig. 4. Some critical observations can be derived from Fig. 4 that **1)** Xception sometimes pays attention to semantically irrelevant regions for real/fake classification such as background, especially when it is trained with DF and tested on unseen manipulation FS. In contrast, in both intra and cross-manipulation evaluations, our model is more likely to focus on facial regions or facial contours. Since face swapping is bound to leave some manipulation traces in local textures of facial regions or blending boundaries around facial contours, these visualizations demonstrate that our model is able to locate such generic regions to exploit generalizable forgery patterns and **2)** our model can generate more fine-grained and adaptive attention heatmaps than Xception. This indicates more sufficient discrimination cues can be captured by the proposed method.

We further show more samples in the cross-dataset scenario as illustrated in Fig. 5. As illustrated in Fig. 5, Xception usually pays attention to background or random large regions irrelevant for face forgery detection, while the proposed model adaptively detects more fine-grained forgery patterns in the facial regions. These visualizations further demonstrate the generalization ability of our model for deepfake detection.

Attention map regarding GBA and LSA. To facilitate the understanding about the function of GBA and LSA, we integrate the pre-trained ViT only with GBA or LSA and visualize Grad-CAM (Selvaraju et al., 2017) of the two cases in Fig. 6, respectively. It can be observed from Fig. 6 that **1)** ViT adapted by GBA tends to focus on facial contours to exploit global low-level features related to the blending boundary, while **2)** ViT adapted with LSA pays attention to some facial regions to adaptively capture local low-level features about textures. This means the adaption by GBA and LSA could complement each other and the combination of them would contribute to a more sufficient and comprehensive adaption.

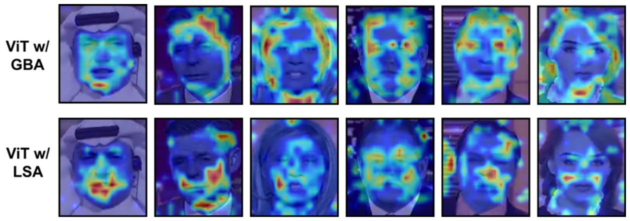
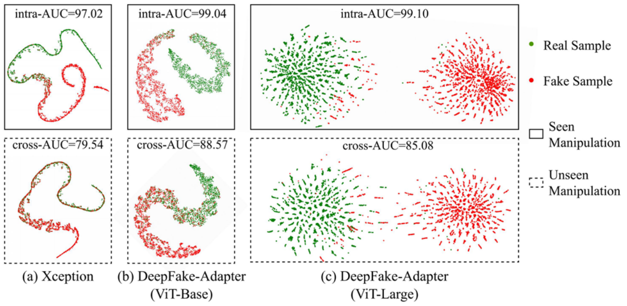


Fig. 6 Comparison of Grad-CAM visualizations between ViT backbone with GBA and LSA. (Best viewed in color)



(a) Xception (b) DeepFake-Adapter (ViT-Base) (c) DeepFake-Adapter (ViT-Large)

Fig. 7 t-SNE visualization of features encoded by **a** Xception **b** DeepFake-Adapter (ViT-Base) and **c** DeepFake-Adapter (ViT-Large) in intra and cross-manipulation settings. (Best viewed in color)

Feature Distribution. We apply t-SNE (Van der & Hinton, 2008) to visualize feature embeddings under the cross-manipulation setting, as illustrated in Fig. 7. Plots in solid-lined boxes, such as plots in the first row of Fig. 7, visualize features of testing samples from seen manipulations encoded by Xception and DeepFake-Adapter (based on ViT-Base and ViT-Large architectures), respectively. Meanwhile, plots in dotted-lined boxes, such as plots in the second row of Fig. 7, visualize features of testing samples from unseen manipulations encoded by Xception and DeepFake-Adapter, respectively.

As illustrated in the first row of Fig. 7, when training and testing manipulation types are identical, both Xception and our model can attain clear classification boundary with two separated class clusters. However, the second row of Fig. 7 shows that feature embeddings of Xception become heavily overlapped between real/fake samples once facing unseen manipulations, while our method can remain much clearer clusters with smaller overlaps. This demonstrates that our model can learn superior generalizable representations for deepfake detection.

Furthermore, it can be seen from (b) and (c) of Fig. 7 that our model based on both ViT-Base and ViT-Large architectures can achieve clear classification boundaries with two separated class clusters. This demonstrates the proposed DeepFake-Adapter can be compatible with various ViT architectures for the task of deepfake detection.

5 Conclusion

This paper studies high-level semantics for deepfake detection and first introduces the adapter approach to efficiently tune large pre-trained ViT to our task. A powerful DeepFake-Adapter is devised with GBA and LSA, which effectively and efficiently leads high-level semantics of ViT to interact with global and local low-level features in a dual-level fashion. Various quantitative and qualitative experiments demonstrate the effectiveness of our model for deepfake detection. Valuable observations pave the way for future research on generalizable deepfake detection in the era of large vision models.

Potential Negative Impact. Although some face forgery data is used, this work is designed to help people better fight against the abuse of deepfake technology. Through our study and releasing our code, we hope to draw greater attention towards generalizable deepfake detection.

Acknowledgements This study is supported by National Natural Science Foundation of China (Grant No. 62306090 and Grant No. 62236003); Natural Science Foundation of Guangdong Province of China (Grant No. 2024A1515010147); Shenzhen College Stability Support Plan (Grant No. GXWD20220817144428005). This study is supported by the Ministry of Education, Singapore, under its MOE AcRF Tier 2 (MOET2EP20221-0012), NTU NAP, and under the RIE2020 Industry Alignment Fund - Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

Data Availability Statement The datasets analysed during this study are all publicly available for the research purpose - the FaceForensics++, Celeb-DF, DeepfakeDetection Challenge, and DeeperForensics−1.0 datasets.

References

Agarap, A.F. (2018). Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.

Ba, J.L., Kiros, J.R., & Hinton, G.E. (2016). Layer normalization. arXiv preprint arXiv:1607.06450.

Cao, J., Ma, C., Yao, T., Chen, S., Ding, S., & Yang, X. (2022). End-to-end reconstruction-classification learning for face forgery detection. In: CVPR, pp. 4113–4122.

Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR, pp. 6299–6308.

Chai, L., Bau, D., Lim, S.N., & Isola, P. (2020). What makes fake images detectable? understanding properties that generalize. In: ECCV, pp. 103–120.

Chen, G., Shen, L., Shao, R., Deng, X., & Nie, L. (2024). Lion: Empowering multimodal large language model with dual-level visual knowledge. In: CVPR, pp. 26540–26550.

Chen, L., Zhang, Y., Song, Y., Liu, L., & Wang, J. (2022). Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection. In: CVPR, pp. 18710–18719.

Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., & Luo, P. (2022). Adaptformer: Adapting vision transformers for scalable visual recognition. *NeurIPS, 35*, 16664–16678.

Chen, S., Yao, T., Chen, Y., Ding, S., Li, J., & Ji, R. (2021). Local relation learning for face forgery detection. *AAAI, 35*, 1081–1088.

Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., & Qiao, Y. (2023). Vision transformer adapter for dense predictions. In: ICLR.

Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., & Qiao, Y. (2023). Vision transformer adapter for dense predictions. In: ICLR.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In: ICCV, pp. 1251–1258.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In: CVPR, pp. 248–255.

Dhariwal, P., & Nichol, A. (2021). Diffusion models beat gans on image synthesis. *NeurIPS, 34*, 8780–8794.

Dhariwal, P., & Nichol, A. (2021). Diffusion models beat gans on image synthesis. *NeurIPS, 34*, 8780–8794.

Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., & Ferrer, C.C. (2020). The deepfake detection challenge (dfdc) dataset. arXiv preprint arXiv:2006.07397.

Dong, X., Bao, J., Chen, D., Zhang, T., Zhang, W., Yu, N., Chen, D., Wen, F., & Guo, B. (2022). Protecting celebrities from deepfake with identity consistency transformer. In: CVPR, pp. 9468–9478.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR.

Durall, R., Keuper, M., Pfreundt, F.J., & Keuper, J. (2019). Unmasking deepfakes with simple features. arXiv preprint arXiv:1911.00686.

Dzanic, T., Shah, K., & Witherden, F. (2020). Fourier spectrum discrepancies in deep network generated images. *NeurIPS, 33*, 3022–3032.

Gu, Q., Chen, S., Yao, T., Chen, Y., Ding, S., & Yi, R. (2022). Exploiting fine-grained face forgery clues via progressive enhancement learning. *AAAI, 36*, 735–743.

Gu, Z., Chen, Y., Yao, T., Ding, S., Li, J., Huang, F., & Ma, L. (2021). Spatiotemporal inconsistency learning for deepfake video detection. In: ACM MM, pp. 3473–3481.

Gu, Z., Chen, Y., Yao, T., Ding, S., Li, J., & Ma, L. (2022). Delving into the local: Dynamic inconsistency learning for deepfake video detection. *AAAI, 36*, 744–752.

Haliassos, A., Mira, R., Petridis, S., & Pantic, M. (2022). Leveraging real talking faces via self-supervision for robust forgery detection. In: CVPR, pp. 14950–14962.

Haliassos, A., Vougioukas, K., Petridis, S., & Pantic, M. (2021). Lips don't lie: A generalisable and robust approach to face forgery detection. In: CVPR, pp. 5039–5049.

He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., & Neubig, G. (2022). Towards a unified view of parameter-efficient transfer learning. In: ICLR.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. In: CVPR, pp. 16000–16009.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: CVPR, pp. 770–778.

Ho, J., Jain, A., & Abbeel, P. (2020). *Denoising diffusion probabilistic models. NeurIPS, 33*, 6840–6851.

Ho, J., Jain, A., & Abbeel, P. (2020). *Denoising diffusion probabilistic models. In: NeurIPS, 33*, 6840–6851.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Supervised sequence labelling with recurrent neural networks pp. 37–45.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., & Gelly, S. (2019). Parameter-efficient transfer learning for nlp. In: ICML, pp. 2790–2799.

Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). Lora: Low-rank adaptation of large language models. In: ICLR.

Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., & Lim, S.N. (2022). Visual prompt tuning. In: ECCV, pp. 709–727.

Jiang, L., Li, R., Wu, W., Qian, C., & Loy, C.C. (2020). Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. In: CVPR, pp. 2889–2898.

Kumar, A., Raghunathan, A., Jones, R., Ma, T., & Liang, P. (2022). Fine-tuning can distort pretrained features and underperform out-of-distribution. In: ICLR.

Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. In: EMNLP, pp. 3045–3059.

Li, J., Xie, H., Li, J., Wang, Z., & Zhang, Y. (2021). Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In: CVPR, pp. 6458–6467.

Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., & Guo, B. (2020). Face x-ray for more general face forgery detection. In: CVPR, pp. 5001–5010.

Li, X., Lang, Y., Chen, Y., Mao, X., He, Y., Wang, S., Xue, H., & Lu, Q. (2020). Sharp multiple instance learning for deepfake video detection. In: ACM MM, pp. 1864–1872.

Li, Y., Yang, X., Sun, P., Qi, H., & Lyu, S. (2020). Celeb-df: A new dataset for deepfake forensics. In: CVPR, pp. 3207–3216.

Li, Z., Xie, Y., Shao, R., Chen, G., Jiang, D., & Nie, L. (2024). Optimus-1: Hybrid multimodal memory empowered agents excel in long-horizon tasks. In: NeurIPS.

Liu, H., Li, X., Zhou, W., Chen, Y., He, Y., Xue, H., Zhang, W., & Yu, N. (2021). Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In: CVPR, pp. 772–781.

Liu, L., Ren, Y., Lin, Z., & Zhao, Z. (2022). Pseudo numerical methods for diffusion models on manifolds. In: ICLR.

Liu, L., Ren, Y., Lin, Z., & Zhao, Z. (2022). Pseudo numerical methods for diffusion models on manifolds. In: ICLR.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys, 55*(9), 1–35.

Liu, Z., Luo, D., Wang, Y., Wang, L., Tai, Y., Wang, C., Li, J., Huang, F., & Lu, T. (2020). Teinet: Towards an efficient architecture for video recognition. *AAAI, 34*, 11669–11676.

Liu, Z., Qi, X., & Torr, P.H. (2020). Global texture enhancement for fake face detection in the wild. In: CVPR, pp. 8060–8069.

Luo, Y., Zhang, Y., Yan, J., & Liu, W. (2021). Generalizing face forgery detection with high-frequency features. In: CVPR, pp. 16317–16326.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. Journal of machine learning research **9**(11).

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). Automatic differentiation in pytorch.

Qi, H., Guo, Q., Juefei-Xu, F., Xie, X., Ma, L., Feng, W., Liu, Y., & Zhao, J. (2020). Deeprhythm: Exposing deepfakes with attentional visual heartbeat rhythms. In: ACM MM, pp. 4318–4327.

Qian, Y., Yin, G., Sheng, L., Chen, Z., & Shao, J. (2020). Thinking in frequency: Face forgery detection by mining frequency-aware clues. In: ECCV, pp. 86–103.

Quinn, N.A., & Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In: ICML, pp. 8162–8171.

Quinn, N.A., & Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In: ICML, pp. 8162–8171.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In: ICML, pp. 8748–8763.

Ricker, J., Damm, S., Holz, T., & Fischer, A. (2024). Towards the detection of diffusion model deepfakes. In: VISAPP, pp. 446–457.

Robin, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In: CVPR, pp. 10684–10695.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In: CVPR, pp. 10684–10695.

Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). Faceforensics++: Learning to detect manipulated facial images. In: ICCV, pp. 1–11.

Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., & Natarajan, P. (2019). Recurrent convolutional strategies for face manipulation detection in videos. *Interfaces (GUI), 3*(1), 80–87.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In: CVPR, pp. 618–626.

Shao, R., Lan, X., Li, J., & Yuen, P.C. (2019). Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In: CVPR, pp. 10023–10031.

Shao, R., Lan, X., & Yuen, P. C. (2018). Joint discriminative learning of deep dynamic textures for 3d mask face anti-spoofing. *IEEE Transactions on Information Forensics and Security, 14*(4), 923–938.

Shao, R., Lan, X., & Yuen, P. C. (2020). Regularized fine-grained meta face anti-spoofing. *AAAI, 34*, 11974–11981.

Shao, R., Perera, P., Yuen, P. C., & Patel, V. M. (2022). Federated generalized face presentation attack detection. *IEEE Transactions on Neural Networks and Learning Systems, 35*(1), 103–116.

Shao, R., Perera, P., Yuen, P. C., & Patel, V. M. (2022). Open-set adversarial defense with clean-adversarial mutual learning. *International Journal of Computer Vision, 130*(4), 1070–1087.

Shao, R., Wu, T., & Liu, Z. (2022). Detecting and recovering sequential deepfake manipulation. In: ECCV, pp. 712–728.

Shao, R., Wu, T., & Liu, Z. (2023). Detecting and grounding multi-modal media manipulation. In: CVPR, pp. 6904–6913.

Shao, R., Wu, T., & Liu, Z. (2023). Robust sequential deepfake detection. arXiv preprint arXiv:2309.14991.

Shao, R., Wu, T., Wu, J., Nie, L., & Liu, Z. (2024). Detecting and grounding multi-modal media manipulation and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 46*(8), 5556–5574.

Shen, L., Chen, G., Shao, R., Guan, W., & Nie, L. (2024). Mome: Mixture of multimodal experts for generalist multimodal large language models. In: NeurIPS.

Sohrawardi, S.J., Chintha, A., Thai, B., Seng, S., Hickerson, A., Ptucha, R., & Wright, M. (2019). Poster: Towards robust open-world detection of deepfakes. In: ACM CCS, pp. 2613–2615.

Thies, J., Zollhöfer, M., & Nießner, M. (2019). Deferred neural rendering: Image synthesis using neural textures. *Acm Transactions on Graphics, 38*(4), 1–12.

Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., & Nießner, M. (2016). Face2face: Real-time face capture and reenactment of rgb videos. In: CVPR, pp. 2387–2395.

Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., & Paluri, M. (2014). C3d: generic features for video analysis. CoRR, abs/1412.0767.

Wang, C., & Deng, W. (2021). Representative forgery mining for fake face detection. In: CVPR, pp. 14923–14932.

Wang, G., Zhou, J., & Wu, Y. (2020). Exposing deep-faked videos by anomalous co-motion pattern detection. arXiv preprint arXiv:2008.04848.

Wang, S.Y., Wang, O., Zhang, R., Owens, A., & Efros, A.A. (2020). Cnn-generated images are surprisingly easy to spot... for now. In: CVPR, pp. 8695–8704.

Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., & Zhang, L. (2021). Cvt: Introducing convolutions to vision transformers. In: ICCV, pp. 22–31.

Wu, W., Zhao, Y., Xu, Y., Tan, X., He, D., Zou, Z., Ye, J., Li, Y., Yao, M., Dong, Z., et al. (2021). Dsanet: Dynamic segment aggregation network for video-level representation learning. In: ACM MM, pp. 1903–1911.

Ye, Q., Yu, Z., Shao, R., Xie, X., Torr, P., & Cao, X. (2024). Cat: Enhancing multimodal large language model to answer questions in dynamic audio-visual scenarios. In: ECCV.

Yuan, K., Guo, S., Liu, Z., Zhou, A., Yu, F., & Wu, W. (2021). Incorporating convolution designs into visual transformers. In: CVPR, pp. 579–588.

Zhang, B., Li, S., Feng, G., Qian, Z., & Zhang, X. (2022). Patch diffusion: A general module for face manipulation detection. *AAAI, 36*, 3243–3251.

Zhang, S., Guo, S., Huang, W., Scott, M.R., & Wang, L. (2020). V4d: 4d convolutional neural networks for video-level representation learning. In: ICLR.

Zhao, H., Zhou, W., Chen, D., Wei, T., Zhang, W., & Yu, N. (2021). Multi-attentional deepfake detection. In: CVPR, pp. 2185–2194.

Zhao, T., Xu, X., Xu, M., Ding, H., Xiong, Y., & Xia, W. (2021). Learning self-consistency for deepfake detection. In: ICCV, pp. 15023–15033.

Zhou, P., Han, X., Morariu, V.I., & Davis, L.S. (2017). Two-stream neural networks for tampered face detection. In: CVPRW, pp. 1831–1839.

Zhu, X., Wang, H., Fei, H., Lei, Z., & Li, S.Z. (2021). Face forgery detection by 3d decomposition. In: CVPR.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE, 109*(1), 43–76.

Zi, B., Chang, M., Chen, J., Ma, X., & Jiang, Y.G. (2020). Wilddeepfake: A challenging real-world dataset for deepfake detection. In: ACM MM, pp. 2382–2390.