

# Forensics Adapter: Adapting CLIP for Generalizable Face Forgery Detection

Xinjie Cui<sup>1</sup> Yuezun Li<sup>1,✉</sup> Ao Luo<sup>2</sup> Jiaran Zhou<sup>1</sup> Junyu Dong<sup>1</sup>  
<sup>1</sup> School of Computer Science and Technology, Ocean University of China  
<sup>2</sup> Southwest Jiaotong University

## Abstract

We describe the *Forensics Adapter*, an adapter network designed to transform CLIP into an effective and generalizable face forgery detector. Although CLIP is highly versatile, adapting it for face forgery detection is non-trivial as forgery-related knowledge is entangled with a wide range of unrelated knowledge. Existing methods treat CLIP merely as a feature extractor, lacking task-specific adaptation, which limits their effectiveness. To address this, we introduce an adapter to learn face forgery traces – the blending boundaries unique to forged faces, guided by task-specific objectives. Then we enhance the CLIP visual tokens with a dedicated interaction strategy that communicates knowledge across CLIP and the adapter. Since the adapter is alongside CLIP, its versatility is highly retained, naturally ensuring strong generalizability in face forgery detection. With only **5.7M** trainable parameters, our method achieves a significant performance boost, improving by approximately **7%** on average across five standard datasets. We believe the proposed method can serve as a baseline for future CLIP-based face forgery detection methods. The code is available at <https://github.com/OUC-VAS/ForensicsAdapter>.

## 1. Introduction

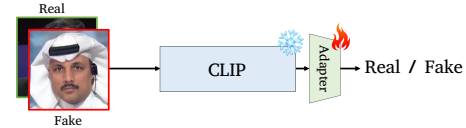
Face forgery techniques<sup>1</sup> have seen remarkable progress in recent years, largely due to significant advancements in generative models [18, 23, 28, 55]. These techniques enable the manipulation of faces with high realism, causing serious concerns to social security, such as privacy violations [7, 34], economic fraud [5, 31], etc. These concerns have motivated a great need for exploration into face forgery detection.

Recent face forgery detection methods are typically de-

<sup>✉</sup>Corresponds to Yuezun Li ([liyuezun@ouc.edu.cn](mailto:liyuezun@ouc.edu.cn))

<sup>1</sup>Face forgery strictly refers to techniques that manipulate local facial content, such as altering lip movement, swapping central face, editing expression, etc. Whole facial image synthesis using GAN or Diffusion models are out of this scope.

(a) Existing methods



(b) Ours

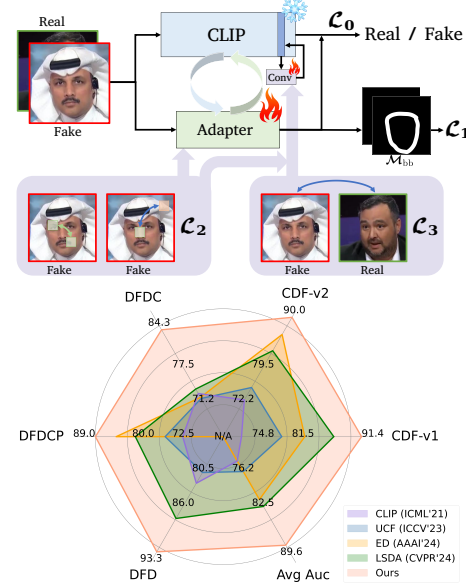


Figure 1. (a) Existing CLIP-based methods. (b) The proposed Forensics Adapter, achieving the best performance compared with several state-of-the-arts on CDF-v1 [40], CDF-v2 [40], DFDC [16], DFDCP [15], and DFD [12] datasets.

veloped on CNNs, benefiting from the learning capacity of these models. They leverage various forgery clues, including biological signals [10, 52, 72], blending artifacts [38, 48, 50], and frequency signals [43, 47, 53]. While these methods can achieve favorable and near-perfect performance on standard datasets, their ability to detect unseen forgeries drops significantly. This is because these methods tend to overfit the specific forgeries present in the training data, limiting their generalization across different distributions. With new forgery techniques constantly emerging in real-world scenarios, these detection methods remain

highly constrained in practical use. To address this issue, several generalizable face forgery detection methods have been proposed, aiming to capture common forgery clues through approaches such as pseudo-fake face augmentations [33, 57], and disentanglement learning [42, 69]. Nevertheless, given the limited variety of forgery categories, these methods still struggle to acquire sufficient forgery priors.

CLIP [54], a recent large vision-language model, contains extensive knowledge priors, including that related to forgery, offering promising potential to enhance generalizable face forgery detection. Inspired by this, several methods [26, 29, 41] are proposed to utilize CLIP for face forgery detection. They commonly use the frozen CLIP image encoder as a feature extractor, with an optional adapter network appended after CLIP. To learn forgery-related knowledge, the adapter network is trained using general objectives (*e.g.*, cross-entropy) on face forgery datasets (see Fig. 1 (a)).

However, this straightforward adaptation has two major limitations: **(1) Task-agnostic adaptation.** The adapter is designed generically, rather than tailored to capture unique task-specific clues in face forgery. **(2) Lack of interaction between CLIP and the adapter.** Since CLIP is not specifically designed for this task, its forgery-related knowledge is intertwined with a broad range of other knowledge. The adapter, therefore, should interact more directly with CLIP to guide it toward forgery-related knowledge, rather than only refining the final output of CLIP. Moreover, the versatility of CLIP can serve as a powerful source of instruction for training the adapter effectively.

In this paper, we propose **Forensics Adapter**, a devoted framework for adapting CLIP to generalizable face forgery detection (see Fig. 1 (b)). Unlike previous CLIP-based methods, our method is grounded in the core nature of face forgery, with a task-specific adaptation that focuses on key forgery-related clues. Our motivation stems from the observation that face forgeries often involve localized manipulations, where synthesized content is blended back into the original regions, creating inconsistencies between the manipulated and original regions. These inconsistencies form blending boundaries, which serve as crucial clues specific to face forgery. To capture these subtle traces, we propose a lightweight adapter network alongside CLIP and introduce task-specific objectives to instruct learning of the adapter. Then we describe the interaction strategy between the adapter and CLIP, assisting the learning of the adapter while enhancing the token effectiveness of CLIP. With this adapter, CLIP is transformed into an effective face forgery detector.

While the blended inconsistencies have been utilized in previous methods [49, 57], the solution of specific adaptation to CLIP is novel and unexplored, offering fresh insights

for future research. Extensive experiments demonstrate that our method outperforms state-of-the-arts by a large margin (+7% in AUC) across five public datasets, with only **5.7M** trainable parameters. We believe our method can establish a strong baseline for CLIP-based face forgery detection methods.

Our contribution can be summarized in threefold: (1) We bridge the gap between CLIP and face forgery task by introducing a Forensics Adapter that effectively adapts CLIP for generalizable face forgery detection. (2) We provide a task-specific adapter design, including both architecture and objectives, and outline an interaction strategy between CLIP and the adapter. (3) The experimental results are promising, demonstrating the potential of our method as a baseline for CLIP-based face forgery detection methods.

## 2. Related Works

**Face Forgery.** Face forgery involves manipulating the local content of the original faces using generative strategies or 3D-based strategies [12, 15, 16, 40, 56]. Its pipeline usually consists of two steps. The first step is to create fake content, *e.g.*, tampered lip or head movement, forged facial identity, etc. The second step is to blend the fake content back to the corresponding position of the original faces while retaining high visual quality. Although the recent face forgery techniques have greatly evolved, this two-step pipeline always introduces inconsistencies between the forgery and original areas, resulting in a blending boundary.

Besides face forgery, whole facial image synthesis is another widely studied forensics topic. It involves synthesizing whole images from scratch using GAN [18] or Diffusion models [23, 55], based on random noises or certain conditioned inputs. It is important to clarify that this paper focuses on face forgery rather than whole facial image synthesis.

**Face Forgery Detection.** Detecting face forgeries is a significant topic in recent years [3, 11, 39, 49, 53, 62]. With the great advancement of deep learning, recent methods are typically designed on Deep Neural Networks (DNNs) to capture the forgery traces, utilizing features ranging from the biological signals (*e.g.*, eye blinking [39], heart-beat rhythm [52], head pose [72], facial action units [1]), frequency signals (*e.g.*, [43, 47, 53]), and auto-learned clues with dedicated training schemes (*e.g.*, [6, 11, 58]). However, these methods usually rely on specific training datasets, which can lead to overfitting on known data distributions and hinder their ability to generalize across different datasets. To enhance detection generalization, many methods have been proposed to create pseudo-fake faces by simulating blending boundaries in real images [3, 33, 37, 49, 57]. While these methods improve the generalization to a certain extent, limited data diversity still restricts detectors

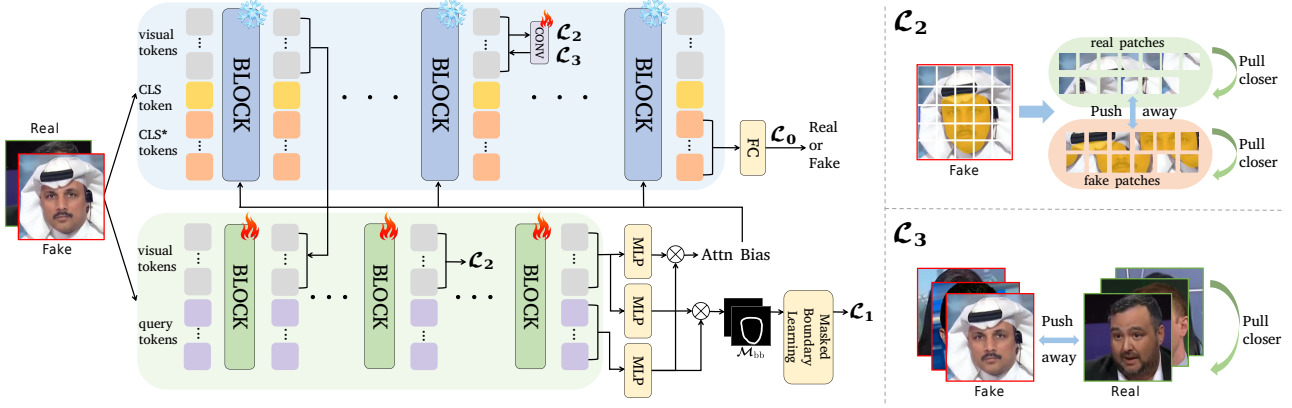


Figure 2. Pipeline of the proposed Forensics Adapter. The top stream denotes CLIP and the bottom stream corresponds adapter. See text for details.

from learning generic features.

More recently, several efforts have started adapting CLIP [54] for face forgery detection [26, 32, 41]. For instance, FFAA [26] creates a dataset comprising real and forged faces with descriptions and forgery reasoning and implements the face forgery through a multi-answer decision strategy based on CLIP. GM-DF [32] attempts to mitigate discrepancies across multiple combined training datasets by leveraging CLIP. FCG [21] leverages CLIP to guide the detection of key facial components over temporal sequences. VLFFD [59] generates mixed forgery images with fine-grained prompts and uses them with original data for joint training. RepDFD [41] reprograms CLIP by merging universal perturbations to the visual input, without adjusting the inner parameters of the VLM model. CLIPping [29] explores the effectiveness of CLIP in combination with recent adaptation methods for universal deepfake detection.

Regardless of how advanced these methods are, they share a common limitation: they solely use CLIP as a feature extractor, overlooking the unique characteristics of face forgery detection. This results in limited task-specific adaptation and incomplete utilization of CLIP’s versatility. Therefore, we describe the Forensics Adapter, designed to capture specific face forgery traces and effectively adapt CLIP into a generalizable face forgery detector.

### 3. Forensics Adapter

To ensure practical usability, the adapter should be lightweight. Thus, we employ the tiny vision transformer [65] as the foundation of the adapter, incorporating only the first eight layers and positioning it parallel to CLIP to unleash its full capacity, inspired by the methods in [44, 66]. Unlike general computer vision tasks, the adapter is instructed to detect the blending boundaries in forged faces, with dedicated objectives (Sec. 3.1). Meanwhile, we describe an interaction strategy between the adapter and

CLIP. On one hand, absorbing CLIP visual tokens can support the learning process of the adapter, while on the other, the task-specific knowledge gained by the adapter can be transferred back into CLIP (Sec. 3.2). With these task-specific objectives and interaction strategies, this adapter can effectively instruct CLIP to focus more on forgery traces while preserving its generalizability. The pipeline of our method is shown in Fig. 2.

#### 3.1. Learning Task-specific Traces

**Input Tokens.** Given an input face image, we first divide it into  $16 \times 16$  patches and then convert them as visual tokens. These tokens are concatenated with  $N$  learnable query tokens as the input for the adapter.

**Prediction Head.** To instruct the adapter in learning blending boundaries, we design a prediction head in the adapter to predict the blending boundary map. The input of this head is query tokens and visual tokens from the adapter respectively. For query tokens, we project it into 128-dimension using MLPs, which is denoted as  $Q \in \mathbb{R}^{N \times 128}$ . Similarly, we project visual tokens into 128-dimension using another MLPs, which is denoted as  $V_{bb} \in \mathbb{R}^{h \times w \times 128}$ , where  $h, w$  is  $1/16$  of input image size. The inner product of  $Q$  and  $V_{bb}$  can generate the blending boundary map as

$$\mathcal{M}_{bb} = \text{Conv}(V_{bb}Q^T), \quad (1)$$

where Conv denotes a couple of convolution operations that gradually transform the dimension of the output as  $\mathcal{M}_{bb} \in \mathbb{R}^{h \times w \times 1}$ . Each element in  $\mathcal{M}_{bb}$  represents the probability of being on a blending boundary. To supervise the prediction and compel CLIP with the adapter to more accurately capture the subtle distinctions between real and fake regions, we generate ground truth blending boundaries for each input and propose three objectives regarding masked blending boundary learning, inter-face contrastive learning, and intra-face patch-wise contrastive learning. The details are elaborated below.

**Masked Blending Boundary Learning.** Following previous works [37], we first perform Gaussian blurring on ground truth manipulation mask  $\mathcal{M}'$  and then generate ground truth blending boundary with  $\mathcal{M}'_{bb} = 4\mathcal{M}'(1 - \mathcal{M}')$ . A straightforward way would be to directly calculate the Mean Square Error (MSE) between  $\mathcal{M}'_{bb}$  and  $\mathcal{M}_{bb}$ . However, since the boundary occupies only a small portion of the map, using standard MSE can easily be influenced by the non-boundary areas, leading to suboptimal results. Therefore, we introduce a masking strategy to highlight the effect of the boundary areas. Specifically, we define a binary mask  $B$  and check all  $16 \times 16$  patches of  $\mathcal{M}'_{bb}$ . If no boundary exists in a patch, the corresponding elements in  $B$  are set to 0. Otherwise, they are set to 1. This objective can be defined as

$$\mathcal{L}_1 = \text{MSE}(\mathcal{M}_{bb} \odot B, \mathcal{M}'_{bb}). \quad (2)$$

**Patch-wise Contrastive Learning.** Note that in a forged face, the representations of the forged region should differ from the authentic regions. Highlighting this difference can assist the model in capturing the task-specific traces. To achieve this, we describe a patch-wise contrastive learning loss performed on the adapter and CLIP.

For the adapter, we extract the intermediate features and perform contrastive learning on them. Each element corresponds to a divided patch, which is labeled as fake if the forgery region covers more than 10% of the corresponding patch, and as real otherwise. Denote  $\mathbf{X} = [\mathbf{X}^1, \mathbf{X}^0]$  as a feature, where  $\mathbf{X}^1, \mathbf{X}^0$  are two sets containing real and fake elements. The patch-wise contrastive learning can be formulated as

$$\mathcal{L}_2 = -\log \frac{\exp(\delta(\mathbf{x}_i, \mathbf{x}_j))/\tau}{\exp(\delta(\mathbf{x}_i, \mathbf{x}_j))/\tau + \sum_{\mathbf{x}_k \in \mathcal{X}^*} \exp(\delta(\mathbf{x}_i, \mathbf{x}_k))/\tau}, \quad (3)$$

where  $\mathbf{x}_i, \mathbf{x}_j$  are elements from same set,  $\mathcal{X}^*$  represents the opposite set of  $\mathbf{x}_i$ , i.e.,  $\mathcal{X}^* = \mathcal{X}^1$  if  $\mathbf{x}_i \in \mathcal{X}^0$ , and  $\mathcal{X}^* = \mathcal{X}^0$  otherwise.  $\tau$  is the temperature parameter and  $\delta$  is the cosine similarity.

For CLIP, we apply the same formulation to refine its visual tokens. Note that the parameters of CLIP are frozen and cannot be directly updated. Therefore, we employ another trainable  $1 \times 1$  convolution on the visual tokens (e.g.,  $\mathbf{X}$ ) and perform patch-wise contrastive learning on the newly obtained tokens (e.g.,  $\mathbf{X}'$ ). Then we add the enhanced tokens back to CLIP with a factor  $\alpha$ , that is  $\mathbf{X} = \mathbf{X} + \alpha\mathbf{X}'$ .

**Sample-wise Contrastive Learning.** In addition to the patch-wise correlation, there are certain associations between real and fake samples. Since the fake samples in this context may belong to different categories (e.g., DF, NT, etc.), we do not consider the similarity among them. In-

stead, we only focus on pulling close real samples while increasing the distance between real and fake samples.

Since CLIP focuses more on the global view, we perform sample-wise contrastive learning on it. Denote a training batch as  $D = [\mathcal{D}^1, \mathcal{D}^0]$ , where  $\mathcal{D}^1, \mathcal{D}^0$  denote the sets with real and fake faces. This learning process can be written as

$$\mathcal{L}_3 = -\log \frac{\exp(\delta(\mathbf{X}_i, \mathbf{X}_j))/\tau}{\exp(\delta(\mathbf{X}_i, \mathbf{X}_j))/\tau + \sum_{\mathbf{X}_k \in \mathcal{D}^*} \exp(\delta(\mathbf{X}_i, \mathbf{X}_k))/\tau}, \quad (4)$$

where  $\mathbf{X}_i, \mathbf{X}_j$  are samples from same set,  $\mathcal{D}^*$  represents the opposite set of  $\mathbf{X}_i$  as in Eq. (3). Since CLIP is frozen, we use the same strategy in patch-wise contrastive learning to update CLIP. Similarly, we employ another trainable  $1 \times 1$  convolution on the visual tokens (e.g.,  $\mathbf{X}$ ) and perform the learning on the newly obtained tokens. Then we add the enhanced tokens back to CLIP with a the same factor  $\alpha$ .

### 3.2. Interaction between Adapter and CLIP

**Absorbing CLIP Visual Tokens.** The features of CLIP are highly versatile and can greatly support the learning process of the adapter. Thus we fuse these features into the adapter. Since CLIP and the adapter have different goals, we limit the feature fusion to the shallow layers of the adapter, avoiding interference with the learning of task-specific features. For a standard 24-layer ViT-L/14 model, we fuse features from layers  $\{1, 8, 16\}$  to the layers  $\{1, 2, 3\}$  of the adapter. Since the feature dimensions of CLIP and the adapter are inconsistent, we employ a  $1 \times 1$  convolution to align them. We then add these two features as the fused ones.

**Enhancing CLIP Visual Tokens.** To make CLIP focus more on blending boundary traces, we convey the knowledge of the adapter to CLIP without modifying any parameters. We achieve this by utilizing an attention bias strategy similar to the one proposed in [66]. Note that CLIP tokens comprise visual tokens and [CLS] token, denoted as  $[\mathbf{X}_{vis}, \mathbf{X}_{[CLS]}]$ . For each layer, we duplicate  $N$  copies of [CLS] token to form an independent token as  $\mathbf{X}_{[CLS^*]} = [\mathbf{X}_{[CLS]}, \dots, \mathbf{X}_{[CLS]}]$ . The entire tokens at a layer can be written as  $[\mathbf{X}_{vis}, \mathbf{X}_{[CLS]}, \mathbf{X}_{[CLS^*]}]$ . Then we update  $\mathbf{X}_{[CLS^*]}$  using self-attention based on the output of the adapter and original visual tokens in CLIP, as

$$\mathbf{X}_{[CLS^*]}^{(\ell+1)} = \text{Softmax}(\mathbf{Q}_{[CLS^*]}^{(\ell)} \mathbf{K}_{vis}^{\top (\ell)} + \Delta^{\top}) \mathbf{V}_{vis}^{(\ell)}, \quad (5)$$

where  $\ell$  represents the layer number,  $\mathbf{Q}_{[CLS^*]}$  denotes the query of  $\mathbf{X}_{[CLS^*]}$ ,  $\mathbf{K}_{vis}$  denotes the key of visual tokens  $\mathbf{X}_{vis}$ , and  $\mathbf{V}_{vis}$  indicates the value of visual tokens  $\mathbf{X}_{vis}$ , respectively. Notably,  $\Delta$  is the attention bias generated by the end of the adapter, following similar operations in generating the blending boundary map  $\mathcal{M}_{bb}$  in Eq. (1). Specifically, we use another MLPs to project visual tokens into



128-dimension as  $V_{ab} \in \mathbb{R}^{h \times w \times 128}$  and generate the attention bias using the inner product of  $Q$  and  $V_{ab}$  as

$$\Delta = V_{ab} Q^\top, \quad (6)$$

where  $Q$  denotes the same query tokens in the prediction head. This attention bias  $\Delta \in \mathbb{R}^{h \times w \times N}$  adjusts the self-attention calculation in CLIP, serving as the bridge to convey knowledge of the adapter. Finally, we use  $X_{[CLS^*]}$  for identification of face authenticity.

### 3.3. Overall Objectives

In the training phase, the main objective is a Cross-entropy loss applied on  $X_{[CLS^*]}$  in the last layer of CLIP, denoted as  $\mathcal{L}_0$ , to achieve forgery detection. Together with the aforementioned objectives in Sec. 3.1, the overall objective can be written as

$$\mathcal{L} = \lambda_0 \mathcal{L}_0 + \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_2 + \lambda_3 \mathcal{L}_3, \quad (7)$$

where  $\lambda_{0 \sim 3}$  are weighting factors to balance each term.

## 4. Experiments

### 4.1. Experimental Settings

**Datasets.** Following previous methods [70, 71], we utilize the following six publicly available datasets: FaceForensics++ (FF++) [56], Deepfake Detection Challenge (DFDC) [16], preview version of DFDC (DFDCP) [15], two versions of CelebDF (CDF-v1, CDF-v2) [40], and DeepfakeDetection (DFD) [12], respectively. To validate the generalizability, our method is trained on c23 compression version of FF++ and tested on other datasets.

**Implementation Details.** Our method is implemented by Pytorch 2.3.0 [51] with one Nvidia GTX 3090 GPU. The adapter is configured with a patch size of 16, an image resolution of 256, and no pre-trained weights are loaded. CLIP architecture is ViT-L/14 with a patch size of 14 and an image resolution of 224, with pre-trained weights loaded. The dimension  $N$  is set to 128. For the adapter, we use the features from intermediate layers 4, 5, and 6 for patch-wise contrastive learning. For CLIP, we add a trainable  $1 \times 1$  convolution to its 13th layer to support both sample-wise contrastive learning and patch-wise contrastive learning, with the factor  $\alpha$  set to 0.05. For overall objectives,  $\lambda_2$  is different on the adapter and CLIP:  $\lambda_2 = 20$  on the adapter and  $\lambda_2 = 10$  on CLIP. Other factors are set to  $\lambda_0 = 10$ ,  $\lambda_1 = 200$ ,  $\lambda_3 = 10$  respectively. The entire model was trained on the FF++ c23 training set, with a batch size of 16, using the Adam optimizer [30] with a learning rate of 0.0002 and weight decay of 0.0005. We follow the configuration of DeepfakeBench [70]: For each video, 32 frames were extracted for training or testing, and RetinaFace [13] was used for face extraction.

### 4.2. Results

**Compared with State-of-the-arts.** For a comprehensive comparison, we compare our method with other state-of-the-arts under both frame-level and video-level evaluations.

Table 1 shows the cross-dataset evaluation results in frame-level AUC. In this scenario, all methods are trained on the FF++ c23 set and evaluated across different datasets. Note that the results of CFM [46] and ED [2] are taken from their original papers. Since SBI [57] does not report its frame-level results, we use the reproduced results from CFM. For all the rest methods, we take the results from the second-best method LSDA [71]. It can be observed that our method outperforms the previous best method LSDA by an impressive 7% improvement in average AUC. Notably, on the widely recognized challenging dataset DFDC, our method significantly surpasses LSDA by 10.7% in AUC, demonstrating the generalizability of our method.

Table 2 shows the cross-dataset evaluation results in video-level AUC. Following previous works [33], the evaluation is performed on CDF-v2, DFDC, and DFDCP datasets. The top section presents video-based methods and the bottom section corresponds to frame-based methods<sup>2</sup>. Note that RealForensics [20], SeeABLE [33], SBI [57], AUNet [3] and LAA-NET [49] are trained on the custom dataset, while others are trained on FF++ c23 set. We can observe that our method significantly outperforms others on all datasets, improving AUC by 0.3%, 8.7%, 6% on CDF-v2, DFDC, and DFDCP, respectively. These results also corroborate the effectiveness of our method in generalizable face forgery detection.

**Compared with CLIP-based Methods.** To fully demonstrate the effectiveness of the adapter, we compare our method with CLIP-based methods. Table 3 shows the comparison results of our method and other CLIP-based methods. Note that VLFFD [59], CLIPping [29], and FFAA [26] are trained on the custom dataset, while the others are trained on FF++ c23 set. Vanilla CLIP refers to the frozen ViT-L/14 model with an additional FC head. We can see that our method notably outperforms others on all datasets, improving the performance by 3.4%, 6%, 2.4% on CDF-v2, DFDC, and DFD respectively. It further demonstrates the superiority of the proposed adapter.

### 4.3. Analysis

**Effect of Losses.** This part studies the effect of each loss ( $\mathcal{L}_1$ ,  $\mathcal{L}_2$ ,  $\mathcal{L}_3$ ). We employ Area Under Curve (AUC), Average Precision (AP), and Equal Error Rate (EER) for evaluation. The results are shown in Table 4. It can be seen that using all losses achieves the best performance. Notably,  $\mathcal{L}_2$

<sup>2</sup>Since several frame-based methods only report either frame-level AUC or video-level AUC, the included methods in Table 1 and Table 2 (bottom section) are not fully consistent.

Method	Venue	CDF-v1	CDF-v2	DFDC	DFDCP	DFD	Avg.
Xception [56]	ICCV'19	0.779	0.737	0.708	0.737	0.816	0.755
EfficientB4 [60]	ICML'19	0.791	0.749	0.696	0.728	0.815	0.756
F3Net [53]	AAAI'20	0.777	0.735	0.702	0.735	0.798	0.749
X-ray [37]	CVPR'20	0.709	0.679	0.633	0.694	0.766	0.696
FFD [11]	CVPR'20	0.784	0.744	0.703	0.743	0.802	0.755
SPSL [43]	CVPR'21	0.815	0.765	0.704	0.741	0.812	0.767
SRM [47]	CVPR'21	0.793	0.755	0.700	0.741	0.812	0.760
Recce [6]	CVPR'22	0.768	0.732	0.713	0.734	0.812	0.752
SBI [57]	CVPR'22	-	0.813	-	0.799	0.774	-
UCF [69]	ICCV'23	0.779	0.753	0.719	0.759	0.807	0.763
ED [2]	AAAI'24	0.818	<u>0.864</u>	0.721	<u>0.851</u>	-	-
LSDA [71]	CVPR'24	<u>0.867</u>	0.830	<u>0.736</u>	0.815	0.880	<u>0.826</u>
CFM [46]	TIFS'24	-	0.828	-	0.758	<u>0.915</u>	-
Ours	-	<b>0.914</b>	<b>0.900</b>	<b>0.843</b>	<b>0.890</b>	<b>0.933</b>	<b>0.896</b>

Table 1. Cross-dataset evaluation results (**Frame-level AUC**). All methods are trained on FF++ and evaluated on other datasets. The **best** results are indicated in bold and the second-best results are underlined.

Method	Venue	CDF-v2	DFDC	DFDCP
RealForensics [20]	CVPR'22	0.869	0.759	-
TALL [67]	ICCV'23	0.908	0.768	-
AltFreezing [64]	CVPR'23	0.895	-	-
SeeABLE [33]	ICCV'23	0.873	0.759	0.863
IID [25]	CVPR'23	0.838	-	0.812
TALL++ [68]	IJCV'24	0.920	<u>0.785</u>	-
SAM [8]	CVPR'24	0.890	-	-
SBI [57]	CVPR'22	0.932	0.724	0.862
AUNet [3]	CVPR'23	0.928	0.738	0.862
CADDm [17]	CVPR'23	0.939	0.739	-
SFDG [63]	CVPR'23	0.758	0.736	-
LAA-NET [49]	CVPR'24	<u>0.954</u>	-	<u>0.869</u>
LSDA [71]	CVPR'24	0.911	0.770	-
CFM [46]	TIFS'24	0.897	-	0.802
Ours	-	<b>0.957</b>	<b>0.872</b>	<b>0.929</b>

Table 2. Cross-dataset evaluation results (**Video-level AUC**). All the results are taken from their original papers. The top section presents video-based methods and the bottom section corresponds to frame-based methods.

Method	Venue	CDF-v2	DFDC	DFD
Vanilla CLIP [54]	ICML'21	0.777	0.742	0.834
VLFFD [59]	arXiv'23	0.848	-	<u>0.948</u>
FFAA [26]	arXiv'24	-	0.740	0.920
GM-DF [32]	arXiv'24	0.832	0.772	-
FCG [21]	arXiv'24	<u>0.923</u>	<u>0.812</u>	-
RepDFD [41]	arXiv'24	0.899	0.810	-
CLIPping [29]	ICMR'24	-	0.719	0.866
Ours	-	<b>0.957</b>	<b>0.872</b>	<b>0.972</b>

Table 3. Cross-dataset evaluation results (**Video-level AUC**). Except for CLIPping referenced from FFAA, all other results are taken directly from the original papers.

seemingly has the largest impact on the performance, which averagely improves 6.4%, 5.0% in AUC and AP, respectively, while reducing the EER by an average of 5.9%.

**Where Using Contrastive Learning.** Theoretically, patch-wise contrastive learning and sample-wise contrastive learning can be used on both CLIP and adapter. Thus, we study where the learning should be added. The results, shown in Table 5, highlight the importance of the proposed task-specific learning strategies. By using the learning on the adapter (ADP) side, we can improve the performance by 4.1%, 2.1% in AUC and AP while reducing EER by 3.0% on average. By applying the learning solely on the CLIP side, we can achieve a performance increase by around 7.2%, 3.3% in AUC and AP while lowering EER by 6.5% on average. When adding the learning strategy to both CLIP and the adapter, our method achieves the best performance.

**Using Various LLMs.** This part investigates the effect of using various LLMs. Specifically, we validate three LLMs: BLIP ViT-L/14 [36], CLIP ViT-B/16 [54], and CLIP ViT-L/14 [54] (the one used in main experiment). The results are shown in Table 6. We observe that BLIP is not suitable for the task, which highly degrades the performance compared to CLIP variants. This is possible because BLIP is trained using only 129 million image-text pairs and designed for image-text retrieval and VQA tasks with a bootstrapping mechanism. In contrast, CLIP is trained on a much larger dataset containing 400 million samples, enabling superior zero-shot capabilities in the modality of images. Straightforwardly, using large ViT can obtain more performance gain than the base version, which aligns with our understanding.

**Using Various Fine-Tuning Methods.** We also study the effect of using various fine-tuning methods, including LoRA [24] and subset fine-tuning [22]. As shown in Table 7, without task-specific adaptation, other fine-tuning

$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_3$	CDF-v1			CDF-v2			DFDC			DFDCP			DFD		
			AUC	AP	EER	AUC	AP	EER	AUC	AP	EER	AUC	AP	EER	AUC	AP	EER
×	✓	✓	0.887	0.925	19.3	0.860	0.926	22.3	0.825	0.857	25.5	0.848	0.921	23.5	0.918	0.990	15.1
✓	×	✓	0.818	0.869	26.1	0.822	0.885	25.8	0.790	0.805	27.8	0.826	0.896	23.9	0.902	0.987	16.9
✓	✓	×	0.904	0.934	18.1	0.899	0.941	18.8	0.840	0.871	24.5	<b>0.893</b>	<b>0.943</b>	<b>19.0</b>	0.920	0.990	15.2
✓	✓	✓	<b>0.914</b>	<b>0.940</b>	<b>15.8</b>	<b>0.900</b>	<b>0.945</b>	<b>18.2</b>	<b>0.843</b>	<b>0.873</b>	<b>23.9</b>	0.890	0.942	19.7	<b>0.933</b>	<b>0.992</b>	<b>13.4</b>

Table 4. Ablation study on the effect of three losses. Evaluation metrics are the frame-level AUC, AP, and EER, respectively.

ADP	CLIP	CDF-v1			CDF-v2			DFDC			DFDCP			DFD		
		AUC	AP	EER	AUC	AP	EER	AUC	AP	EER	AUC	AP	EER	AUC	AP	EER
×	×	0.814	0.895	24.9	0.789	0.880	28.7	0.744	0.778	32.3	0.748	0.868	33.0	0.854	0.982	22.4
✓	×	0.845	0.912	24.0	0.822	0.899	25.8	0.778	0.817	29.8	0.809	0.893	28.3	0.899	0.987	18.5
×	✓	0.881	0.926	20.3	0.875	0.923	21.4	0.817	0.840	26.2	0.833	0.894	24.0	0.903	0.988	17.3
✓	✓	<b>0.914</b>	<b>0.940</b>	<b>15.8</b>	<b>0.900</b>	<b>0.945</b>	<b>18.2</b>	<b>0.843</b>	<b>0.873</b>	<b>23.9</b>	<b>0.890</b>	<b>0.942</b>	<b>19.7</b>	<b>0.933</b>	<b>0.992</b>	<b>13.4</b>

Table 5. Ablation study where using task-specific learning. Evaluation metrics are the frame-level AUC, AP, and EER, respectively.

LLMs	CDF-v2	DFDC	DFDCP
BLIP ViT-L/14 [36]	0.607	0.539	0.548
CLIP ViT-B/16 [54]	0.837	0.775	0.799
CLIP ViT-L/14 [54]	<b>0.900</b>	<b>0.843</b>	<b>0.890</b>

Table 6. Effect of using various LLMs.

Method	CDF-v2	DFDCP
LoRA [24]	0.797	0.800
subset fine-tuning [22]	0.770	0.807
Ours	<b>0.900</b>	<b>0.890</b>

Table 7. Effect of using various fine-tuning methods.

Adapter	CDF-v1	CDF-v2	DFDC	DFDCP	DFD	Avg.
Large	0.900	0.891	0.832	0.840	0.905	0.874
Base	0.895	0.869	0.833	0.862	0.902	0.872
Small	0.897	0.886	0.825	0.861	0.906	0.875
Tiny	<b>0.914</b>	<b>0.900</b>	<b>0.843</b>	<b>0.890</b>	<b>0.933</b>	<b>0.896</b>

Table 8. Effect of using various adapter structures.

methods yield only moderate performance.

**Using Various Adapter Architectures.** In this section, we explore the effect of different adapter architectures, including ViT-tiny (ours), ViT-small, ViT-base, and ViT-large. All adapters are trained using the same configuration as in our main experiment. The results are presented in Table 8. As shown, performance does not improve as the number of trainable parameters increases under the same training configuration. This is likely because, with limited training resources, the smaller ViT-tiny architecture can be fully trained to capture forgery traces, whereas the larger architectures do not show additional benefits.

**Exploration of the Text Modality.** In addition to using the visual encoder of CLIP, we also explore the incorporation of text encoder. Following CoOP [74], we design learnable contexts within the prompts and combine the text tokens

Modality	CDF-v1	CDF-v2	DFDC	DFDCP	DFD	Avg.
T+V	0.738	0.720	0.726	0.719	0.814	0.744
V	<b>0.914</b>	<b>0.900</b>	<b>0.843</b>	<b>0.890</b>	<b>0.933</b>	<b>0.896</b>

Table 9. Effect of adding the text modality. T+V represents the use of both text and visual modalities, while V represents the use of only the visual modality.

with visual tokens. Table 9 presents the results. Surprisingly, the performance after using text modality is notably lower than only using the visual modality. This may be because the forged faces are mainly reliant on visual representations, and the integration of text modality introduces irrelevant noise, which distracts attention from the correct forgery traces.

**Detecting Whole Face Image Synthesis.** Although our method is restricted to detecting face forgeries, we explore its potential for detecting whole face images synthesized by GANs and Diffusion models. Specifically, we select three GANs, including StarGAN [9], CramerGAN [4], and MMDGAN [35], and two Diffusion models, including ADM [14] and PNDM [45]. Table 10 shows the performance of our method compared to IID and LSDA. The results indicate that, despite being designed specifically for face forgery tasks, our method exhibits satisfied performance, and still surpasses the counterparts by a large margin on both GANs and Diffusion models.

**Robustness.** This part studies the performance of our method confronting various perturbations. We employ the same perturbation configurations following [27], which includes five levels of six different types of perturbations: color saturation, color contrast, block wise, gaussian blur, jpeg compression, gaussian noise. We compare our method with LSDA, IID, and vanilla CLIP methods in video-level AUC. The results are shown in Fig. 3. It can be seen that our method is more robust than others. All methods are trained

Method	StarGAN [9]			CramerGAN [4]			MMDGAN [35]			ADM [14]			PNDM [45]		
	AUC	AP	EER	AUC	AP	EER	AUC	AP	EER	AUC	AP	EER	AUC	AP	EER
IID [25]	0.699	0.697	34.7	0.639	0.617	40.3	0.565	0.552	45.3	0.609	0.617	42.0	0.279	0.391	63.3
LSDA [71]	0.772	0.796	29.0	0.675	0.646	35.8	0.621	0.592	40.3	0.577	0.549	42.8	0.572	0.546	40.6
Ours	<b>0.954</b>	<b>0.950</b>	<b>10.9</b>	<b>0.936</b>	<b>0.945</b>	<b>13.7</b>	<b>0.896</b>	<b>0.911</b>	<b>18.6</b>	<b>0.806</b>	<b>0.793</b>	<b>26.3</b>	<b>0.638</b>	<b>0.624</b>	<b>39.3</b>

Table 10. Performance of detecting whole face image synthesized by GANs and Diffusion models. Evaluation metrics are the frame-level AUC, AP, and EER, respectively.

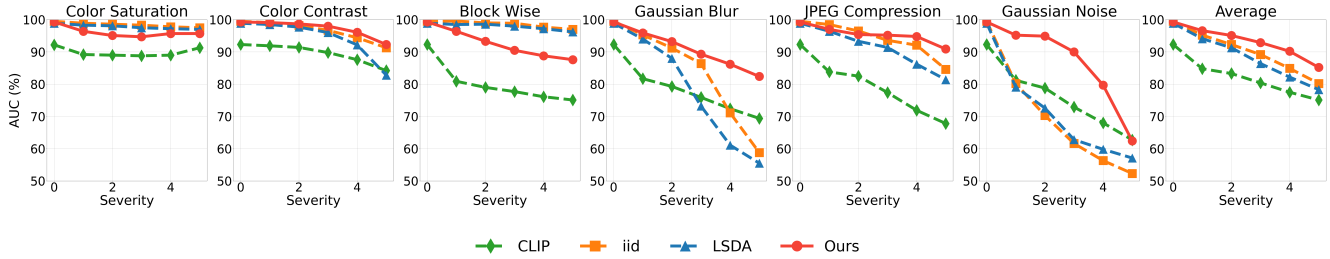


Figure 3. Robustness Analysis. Our method is compared with CLIP [54], IID [25], and LSDA [71] across five levels of six particular types of perturbations in video-level AUC.

Method	# Params	CDF-v2	DFDC
LipForensics [19]	36.0M	0.824	0.735
FTCN [73]	26.6M	0.869	0.740
DCL [58]	19.35M	0.823	0.767
CFM [46]	25.37M	0.897	0.802
Ours	<b>5.7M</b>	<b>0.957</b>	<b>0.872</b>

Table 11. Complexity analysis in video-level AUC.

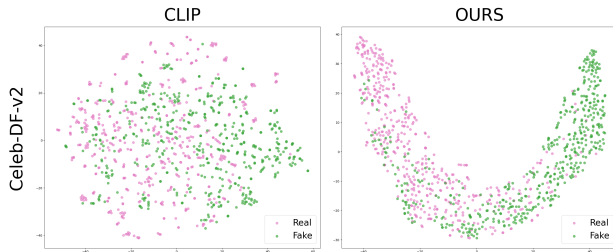


Figure 4. T-SNE Visualizations. The feature distribution before (left) and after (right) using the adapter. The features are extracted from the FC layer after CLIP.

on the FF++ c23 dataset.

**Complexity Analysis.** Since we freeze the CLIP model and only train the lightweight adapter, we significantly reduce the training overhead to 5.7M parameters. Table 11 compares the parameter counts and video-level AUC performance of various methods, all trained on the FF++ c23 dataset. Notably, other methods require three to six times more trainable parameters than ours, yet our method achieves a substantial performance improvement, highlighting the effectiveness of our method.

**T-SNE Visualization.** To demonstrate our adapter greatly improves the effectiveness of CLIP, we visualize the output

of CLIP before and after using the adapter on the CDF-v2 dataset. Specifically, we randomly select 500 real and 500 fake faces from the testing set and employ the T-SNE algorithm [61] for visualization. The results, shown in Fig. 4, reveal that the features of real and fake faces are highly separated with the adapter, indicating its efficacy in capturing generic forgery traces.

**Limitations.** Our method is designed specifically for face forgery detection, rather than whole face image synthesis detection. Thus it only shows decent performance on recent GAN or Diffusion model generated faces. In future work, we may further excavate the potential of CLIP and develop a universal detector capable of addressing both face forgery and whole face image synthesis detection.

## 5. Conclusion

In this paper, we introduce **Forensics Adapter**, an adapter network designed to transform CLIP into a generalizable face forgery detector. Different from existing methods, our adapter is specifically tailored to capture the unique blending boundaries characteristic of forged faces using task-specific objectives. We then introduce an interactive strategy that enables CLIP and the adapter to collaboratively focus on face forgery traces. Notably, our solution of adapting CLIP for face forgery detection is novel, addressing a gap that previous methods have not explored. Extensive experiments demonstrate that, with only 5.7M trainable parameters, our adapter yields substantial performance gains, improving the performance by approximately 7% on average across five standard datasets. We believe our method can serve as a baseline for future CLIP-based face forgery detection techniques.



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

## References

- [1] Shruti Agarwal, Hany Farid, Tarek El-Gaaly, and Ser-Nam Lim. Detecting deep-fake videos from appearance and behavior. In *IEEE international workshop on information forensics and security (WIFS)*. IEEE, 2020. 2
- [2] Zhongjie Ba, Qingyu Liu, Zhenguang Liu, Shuang Wu, Feng Lin, Li Lu, and Kui Ren. Exposing the deception: Uncovering more forgery clues for deepfake detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. 5, 6
- [3] Weiming Bai, Yufan Liu, Zhipeng Zhang, Bing Li, and Weiming Hu. Aunet: Learning relations between action units for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 5, 6
- [4] Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017. 7, 8
- [5] Angela Busacca and Melchiorre Alberto Monaca. Deepfake: Creation, purpose, risks. In *Innovations and Economic and Social Changes due to Artificial Intelligence: The State of the Art*. Springer, 2023. 1
- [6] Junyi Cao, Chao Ma, Taiping Yao, Shen Chen, Shouhong Ding, and Xiaokang Yang. End-to-end reconstruction-classification learning for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 6
- [7] Bobby Chesney and Danielle Citron. Deep fakes: A looming challenge for privacy, democracy, and national security. *Calif. L. Rev.*, 2019. 1
- [8] Jongwook Choi, Taehoon Kim, Yonghyun Jeong, Seungryul Baek, and Jongwon Choi. Exploiting style latent flows for generalizing deepfake video detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 6
- [9] Yunjei Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 7, 8
- [10] Umur Aybars Ciftci, Ilke Demir, and Lijun Yin. Fakecatcher: Detection of synthetic portrait videos using biological signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1
- [11] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K. Jain. On the detection of digital face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 6
- [12] Deepfakedetection. <https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html/> Accessed: 2021-11-13. 1, 2, 5
- [13] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotzia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5
- [14] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2021. 7, 8
- [15] B Dolhansky, R Howes, B Pflaum, N Baram, and CC Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854*, 2019. 1, 2, 5
- [16] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397*, 2020. 1, 2, 5
- [17] Shichao Dong, Jin Wang, Renhe Ji, Jiajun Liang, Haoqiang Fan, and Zheng Ge. Implicit identity leakage: The stumbling block to improving deepfake detection generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6
- [18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 2020. 1, 2
- [19] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Lips don't lie: A generalisable and robust approach to face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 8
- [20] Alexandros Haliassos, Rodrigo Mira, Stavros Petridis, and Maja Pantic. Leveraging real talking faces via self-supervision for robust forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5, 6
- [21] Yue-Hua Han, Tai-Ming Huang, Shu-Tzu Lo, Po-Han Huang, Kai-Lung Hua, and Jun-Cheng Chen. Towards more general video-based deepfake detection through facial feature guided adaptation for foundation model. *arXiv preprint arXiv:2404.05583*, 2024. 3, 6
- [22] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024. 6, 7
- [23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems (NeurIPS)*, 2020. 1, 2
- [24] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 2022. 6, 7
- [25] Baojin Huang, Zhongyuan Wang, Jifan Yang, Jiaxin Ai, Qin Zou, Qian Wang, and Dengpan Ye. Implicit identity driven deepfake face swapping detection. In *Proceedings of*

- the *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6, 8
- [26] Zhengchao Huang, Bin Xia, Zicheng Lin, Zhun Mou, and Wenming Yang. Ffaa: Multimodal large language model based explainable open-world face forgery analysis assistant. *arXiv preprint arXiv:2408.10072*, 2024. 2, 3, 5, 6
- [27] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. Deepforensics-1.0: A large-scale dataset for real-world face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7
- [28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1
- [29] Sohail Ahmed Khan and Duc-Tien Dang-Nguyen. Clipping the deception: Adapting vision-language models for universal deepfake detection. In *Proceedings of the 2024 International Conference on Multimedia Retrieval*, 2024. 2, 3, 5, 6
- [30] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5
- [31] Tyrone Kirchengast. Deepfakes and image manipulation: criminalisation and control. *Information & Communications Technology Law*, 2020. 1
- [32] Yingxin Lai, Zitong Yu, Jing Yang, Bin Li, Xiangui Kang, and Linlin Shen. Gm-df: Generalized multi-scenario deepfake detection. *arXiv preprint arXiv:2406.20078*, 2024. 3, 6
- [33] Nicolas Larue, Ngoc-Son Vu, Vitomir Struc, Peter Peer, and Vassilis Christophides. Seeable: Soft discrepancies and bounded contrastive learning for exposing deepfakes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 5, 6
- [34] Hao-Ping Lee, Yu-Ju Yang, Thomas Serban Von Davier, Jodi Forlizzi, and Sauvik Das. Deepfakes, phrenology, surveillance, and more! a taxonomy of ai privacy risks. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2024. 1
- [35] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. *Advances in neural information processing systems (NeurIPS)*, 2017. 7, 8
- [36] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*. PMLR, 2022. 6, 7
- [37] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 4, 6
- [38] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656*, 2019. 1
- [39] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *IEEE International workshop on information forensics and security (WIFS)*. Ieee, 2018. 2
- [40] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 5
- [41] Kaiqing Lin, Yuzhen Lin, Weixiang Li, Taiping Yao, and Bin Li. Standing on the shoulders of giants: Reprogramming visual-language model for general deepfake detection. *arXiv preprint arXiv:2409.02664*, 2024. 2, 3, 6
- [42] Li Lin, Xinan He, Yan Ju, Xin Wang, Feng Ding, and Shu Hu. Preserving fairness generalization in deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2
- [43] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: Rethinking face forgery detection in frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 6
- [44] Huan Liu, Zichang Tan, Chuangchuan Tan, Yunchao Wei, Jingdong Wang, and Yao Zhao. Forgery-aware adaptive transformer for generalizable synthetic image detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3
- [45] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2022. 7, 8
- [46] Anwei Luo, Chenqi Kong, Jiwu Huang, Yongjian Hu, Xiangui Kang, and Alex C Kot. Beyond the prior forgery knowledge: Mining critical clues for general face forgery detection. *IEEE Transactions on Information Forensics and Security*, 2023. 5, 6, 8
- [47] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 6
- [48] Falko Matern, Christian Riess, and Marc Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In *IEEE Winter Applications of Computer Vision Workshops (WACVW)*. IEEE, 2019. 1
- [49] Dat Nguyen, Nesryne Mejri, Inder Pal Singh, Polina Kuleshova, Marcella Astrid, Anis Kacem, Enjie Ghorbel, and Djamil Aouada. Laa-net: Localized artifact attention network for quality-agnostic and generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 5, 6
- [50] Yuval Nirkin, Lior Wolf, Yosi Keller, and Tal Hassner. Deepfake detection based on discrepancies between faces and their context. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1
- [51] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An im-

- perative style, high-performance deep learning library. *Advances in neural information processing systems*, 2019. 5
- [52] Hua Qi, Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Wei Feng, Yang Liu, and Jianjun Zhao. Deeprrhythm: Exposing deepfakes with attentional visual heartbeat rhythms. In *Proceedings of the 28th ACM international conference on multimedia*, 2020. 1, 2
- [53] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European conference on computer vision*. Springer, 2020. 1, 2, 6
- [54] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 2021. 2, 3, 6, 7, 8
- [55] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2
- [56] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2, 5, 6
- [57] Kaede Shiohara and Toshihiko Yamasaki. Detecting deepfakes with self-blended images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 5, 6
- [58] Ke Sun, Taiping Yao, Shen Chen, Shouhong Ding, Jilin Li, and Rongrong Ji. Dual contrastive learning for general face forgery detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 2, 8
- [59] Ke Sun, Shen Chen, Taiping Yao, Haozhe Yang, Xiaoshuai Sun, Shouhong Ding, and Rongrong Ji. Towards general visual-linguistic face forgery detection. *arXiv preprint arXiv:2307.16545*, 2023. 3, 5, 6
- [60] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, 2019. 6
- [61] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008. 8
- [62] Tianyi Wang, Xin Liao, Kam Pui Chow, Xiaodong Lin, and Yinglong Wang. Deepfake detection: A comprehensive survey from the reliability perspective. *ACM Comput. Surv.*, 57(3), 2024. 2
- [63] Yuan Wang, Kun Yu, Chen Chen, Xiyuan Hu, and Silong Peng. Dynamic graph learning with content-guided spatial-frequency relation reasoning for deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6
- [64] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, and Houqiang Li. Altfreezing for more general video face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6
- [65] Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Tinyvit: Fast pretraining distillation for small vision transformers. In *European conference on computer vision (ECCV)*. Springer, 2022. 3
- [66] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3, 4
- [67] Yuting Xu, Jian Liang, Gengyun Jia, Ziming Yang, Yanhao Zhang, and Ran He. Tall: Thumbnail layout for deepfake video detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 6
- [68] Yuting Xu, Jian Liang, Lijun Sheng, and Xiao-Yu Zhang. Learning spatiotemporal inconsistency via thumbnail layout for face deepfake detection. *International Journal of Computer Vision*, 2024. 6
- [69] Zhiyuan Yan, Yong Zhang, Yanbo Fan, and Baoyuan Wu. Ucf: Uncovering common features for generalizable deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 6
- [70] Zhiyuan Yan, Yong Zhang, Xinhang Yuan, Siwei Lyu, and Baoyuan Wu. Deepfakebench: A comprehensive benchmark of deepfake detection. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*, 2023. 5
- [71] Zhiyuan Yan, Yuhao Luo, Siwei Lyu, Qingshan Liu, and Baoyuan Wu. Transcending forgery specificity with latent space augmentation for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 5, 6, 8
- [72] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019. 1, 2
- [73] Yinglin Zheng, Jianmin Bao, Dong Chen, Ming Zeng, and Fang Wen. Exploring temporal coherence for more general video face forgery detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 8
- [74] Kaiyang Zhou, Jingkan Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 2022. 7