

Rethinking Vision-Language Model in Face Forensics: Multi-Modal Interpretable Forged Face Detector

Supplementary Material

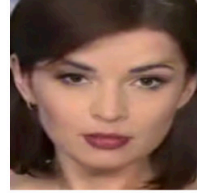
1. Implementation Details

We report implementation details from different aspects, and the source code will be made publicly available upon acceptance.

Architecture Details. In our proposed Multi-Modal Face Forgery Detector (M2F2-Det), we employ the Efficientnet-B4 (Efficient-B4) [7] as the deepfake encoder \mathcal{E}_D . Then, we use CLIP/ViT-L-patch14-336 [2] as the pre-trained CLIP image encoder \mathcal{E}_I and text encoder \mathcal{E}_T . The LLM is Vicuna-7b, denoted as \mathcal{L} . We integrate \mathcal{E}_I and \mathcal{L} in the similar way of LLaVA [5], in which two different MLP layers project the CLIP image feature \mathbf{F}_I and deepfake encoder feature \mathbf{F}^0 into $\mathbf{H}_V \in \mathbb{R}^{576 \times 4096}$ and $\mathbf{H}_F \in \mathbb{R}^{1 \times 4096}$, respectively. Interestingly, we observe \mathbf{F}^0 and \mathbf{f}^0 yield nearly identical performance in text generation, likely because both encode domain-relevant deepfake knowledge, allowing the LLM to access similarly informative visual cues for generating descriptive outputs. \mathcal{E}_A consists of 4 transformer-encoder blocks, and we obtain output features from 10th, 14th, and 22nd layers of \mathcal{E}_I . These features are fed into \mathcal{E}_A , and such design choice is detailed in Tab. 3. Also, we feed \mathcal{E}_A with outputs from last three convolution blocks of \mathcal{E}_D . As for the forgery prompt learning, we set the length of trainable tokens, including general forgery tokens and specific forgery tokens, as 6. First 9 layers of \mathcal{E}_T have 4 trainable layer-wise forgery tokens as inputs.

Training Details. We conduct the three-stage training on 8 A6000 48G GPUs. First, we train the deepfake encoder \mathcal{E}_D , CLIP Text encoder \mathcal{E}_T and Bridge Adapter \mathcal{E}_A using the cross entropy for deepfake detection. The learning rate is $1e-3$, and the optimizer is the Adam optimizer. Secondly, we conduct the feature alignment between \mathbf{H}_V and \mathbf{H}_F on DDVQA. Specifically, only MLP layers are trained, while other components are frozen. We optimize parameters by maximizing the likelihood of target answer tokens in Eq. 7 of the main paper. The learning rate is set as $1e-4$ for 5 epochs. Empirically, such a small training epoch number is enough to converge the training on the DDVQA dataset. Thirdly, we fine-tune the M2F2-Det via instruction-tuning [5] on DDVQA, where only MLP layers and LLM are trained by using LoRA [4]. The supervision is the same as that in the second stage.

Training Samples. The DDVQA dataset is constructed based on common-sense reasoning, which makes the judgment about whether images are real or fake based on visible facial artifacts. In other words, it can fail when the forgery pattern is subtle and cannot be described well



"This face image looks fake. There are stains on eyebrows and eyes region. Based on learned representation, it is a fake image."

(a) An example of DDVQA image-text pair.

User: "Is this image real?"
M2F2-Det: "This face image looks fake."
User: "Can you describe why?"
M2F2-Det: "There are stains on eyebrows and eyes region."
User: "Determine the authenticity of the image."
M2F2-Det: "Based on learned representation, it is a fake image."

(b) The multi-run conversion formulation.

Figure 1. A training sample illustration. (a) A DDVQA image-text pair. In textual descriptions, black and red sentences are original and added contents, respectively. (b) The multi-run formulation used in M2F2-Det’s training procedure.

only by textual explanations. Therefore, we obtain images with subtle artifacts and then add sentences into their corresponding textual descriptions, *i.e.*, ‘‘Based on the learned representation, this image is real/fake.’’. This process is illustrated in Fig. 1. The correlation between this newly added sentence and \mathbf{H}_F is learned by M2F2-Det via training. Also, we formulate DD-VQA samples, *i.e.*, image-text pairs, based on multi-run templates as defined in the LLaVA training and use such updated samples to train the M2F2-Det.

2. Additional Ablation Study

Different Specified Deepfake Encoder. Tab. 1 reports M2F2-Det’s performance using different deepfake backbones. Specifically, Efficient-4B obtains the best performance — 2.55% higher AUC score than XceptionNet on FF++(c40). Therefore, we use Efficient-4B as the specified deepfake encoder of the M2F2-Det. Also, M2F2-Det’s performance is not impacted largely by the choice of different backbones, demonstrating its robustness to different architectural designs.

Universal Forgery Prompts. Universal Forgery Prompts (UF-prompts) are composed of general forgery tokens and specific forgery tokens, which generate forged attention maps that localize forgery regions. We report their forgery localization performance in Tab. 2. More formally, we

Backbones	FF++ (c23)		FF++ (c40)	
	<i>Metric: Acc (% ↑) / AUC (% ↑)</i>			
DenseNet-121	96.54	98.33	90.13	92.31
XceptionNet	97.23	98.10	91.45	94.03
ViT-B	95.13	96.25	89.11	91.97
Efficient-B4	98.79	99.34	93.83	96.58

Table 1. Different backbones used in M2F2-Det’s implementation.

	UF-Prompts		LF	Test set AUC(%)			
	Gen.	Spe.		DF	F2F	FS	NT
1				71.13	55.78	52.34	68.08
2	✓			85.03	80.78	71.11	79.78
3		✓		82.57	77.37	74.57	81.50
4	✓	✓		89.65	83.68	75.20	85.34
5	✓	✓	✓	93.65	89.74	82.74	87.74

Table 2. Forgery localization performance on 4 manipulation types of FF++(c23). Each model is trained on FF++(c23). [Keys: UF-Prompts: universal forgery prompts; Gen.: general forgery token; Spe.: specific forgery token; LF: layer-wise forgery tokens; **Best Results.**]

Fused Layers		FF++
SF	6 th	86.3
SF	10 th	86.8
SF	14 th	89.0
SF	22 nd	89.7
MF	14 th , 22 nd	90.3 ^{+0.6}
MF	10 th , 14 th , 22 nd	91.1 ^{+1.4}

Table 3. Different fusion layers of the Bridge Adapter. [Keys: SF: single fusion, MF: multiple fusion].

download manipulation masks from FF++ and binarize them as the ground truth. Then, we compute the AUC score between such ground truths and forged attention maps generated from different model variants. Specifically, line #1 denotes the localization performance of the pre-trained CLIP, which is used as the baseline. Its performance is worse than using general and specific forgery tokens (*i.e.*, line #2 and #3) on Face2Face (F2F) manipulation by 25.00% and 21.59% AUC scores, respectively. Then, we observe the performance gain from line #4, which indicates the joint usage of specific and general forgery tokens achieves better performance than using them separately. It is worth mentioning that layer-wise trainable tokens further enhance the localization performance of UF-prompts, which is shown by line #5 quantitatively and Fig. 3 qualitatively.

Bridge Adapter Fusion Strategy. The Bridge Adapter

connects the deepfake encoder with the CLIP image encoder, which are CLIP/ViT-L-patch14-336px and Efficient-4B in M2F2-Det, respectively. Tab. 3 reports the performance of different fusion strategies in terms of constructing the Bridge Adapter. Specifically, the single fusion represents only integrating the one single layer output from the pre-trained CLIP image encoder into the Bridge Adapter. The multiple fusion strategy denotes integrating multiple layer outputs from the pre-trained image encoder into the Bridge Adapter. Based on Tab. 3, it shows the latter layer output from the CLIP image encoder helps more on the performance, likely because these layers capture more global information, including forgery patterns. Additionally, the multi-fusion strategy leverages more of the pre-trained CLIP image encoder embeddings, ultimately enhancing overall performance.

	FF++ (c40)		Celeb-DF		WildDeepfake	
	ACC↑	AUC↑	ACC↑	AUC↑	ACC↑	AUC↑
XceptionNet ^{DD}	89.25	92.24	62.41	64.30	62.52	64.53
XceptionNet ^{M2}	91.38	93.40	65.03	67.34	65.00	69.75
HiFi-Net ^{DD}	91.25	95.14	69.37	71.00	69.27	70.03
HiFi-Net ^{M2}	91.48	95.21	71.17	74.68	68.77	72.13
RECCE ^{DD}	92.08	95.36	69.46	70.21	66.57	69.46
RECCE ^{M2}	93.06	95.89	71.46	72.76	65.44	69.57

Table 4. The enhanced deepfake detection performance. ^{DD} and ^{M2} denote detectors enhanced by visual embedding from DDVQA-BLIP [9] and M2F2-Det, respectively. [**Best Performance**]

3. Enhanced Deepfake Detection

Tab. 4 shows the enhanced deepfake detection performance compared with DDVQA-BLIP. Specifically, this table shows visual embedding from our M2F2-Det, *i.e.*, F^0 , outperforms DDVQA-BLIP [9], regardless different detectors. This is because F^0 is more effective in detection, ensembling CLIP embedding, and domain knowledge than visual embedding from DDVQA-BLIP, which is naively fine-tuned for text generation. Even without extract modifications introduced in DDVQA-BLIP, M2F2-Det outputs deepfake scores, which obtain SoTA performance.

4. Additional Generated Textual Explanations

Fig. 2 shows textual explanation generation capabilities of our M2F2-Det can generalize to unseen forged facial images. More formally, we offer additional qualitative results on images generated by StyleGANv2 and recent diffusion models like IP-adapter [8]. The comparison to existing works shows that M2F2-Det is accurate in the textual judgment and convincing in the explanations. Specifically, in the second image, the DDVQA-BLIP [9] fails to identify the forged image and provides erroneous descriptions. In the fourth image, the GPT-4O mistakenly classifies the input image as fake due to the inconsistent texture. In these two scenarios, our proposed M2F2-Det generates the cor-


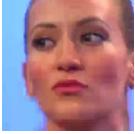


	DDVQA-BLIP	GPT-4O	M2F2-Det
 GT: Fake	The person's skin looks a bit fake because there are stains on the person's face . ✓	This image appears fake due to the unnaturally blurred and textureless eye regions , inconsistent facial details , and overall lack of realistic lighting and depth , all common artifacts of AI-generated forgeries. ✓	The image appears to be an altered version of a person's face . The person has blurry double vision , which is a sign of image duplication and not real. ✓
 GT: Fake	The image looks real . The person has complete face features. Those features are in good shape, size, and positioning . The person has smooth skin, arched eyebrows, round eyes, straight nose, full mouth. ✗	The image shows signs of AI generation , including smooth skin , consistent lighting , irregular hair details , and a blurred, simplistic background. These traits suggest possible AI creation. ✓	This image is fake. This person has overly There are stains or flaws on the person's face . There are areas looking unnaturally brighter . ✓
 GT: Fake	The image looks fake . It looks like a computer generated artist styled image. The person's mouth looks too rigid to convey expressions . There are stains on the person's face . The person's eyebrows look blurry. ✓	The image shows signs of AI generation , including overly smooth skin , poorly blended headgear , blurry background , and unrealistic lighting . These irregularities strongly suggest it was not captured naturally. ✓	The image is not a genuine photograph . The lighting is inconsistent; the President's face is overly bright . The shadows are not consistent with the rest of the scene. The necklace is overly large compared to what we've seen him wear. ✓
 GT: Real	The image looks real . The person has complete face features. Those features are in good shape, size, and positioning . The person has smooth skin, arched eyebrows, round eyes, straight nose, full mouth. ✓	The image shows signs of AI generation . It has overly saturated skin tones , inconsistent textures , and exaggerated facial details . These traits suggest possible post-processing or manipulation via AI technique. ✗	This image appears real due to natural facial asymmetries, realistic skin textures with wrinkles , and consistent lighting and shadows that align with the background scene . This should be a real not AI-generated image. ✓

Figure 2. Additional textual explanation results. From top to bottom, the images are from Celeb-DF, DDVQA, Stable Diffusion, and a real image.

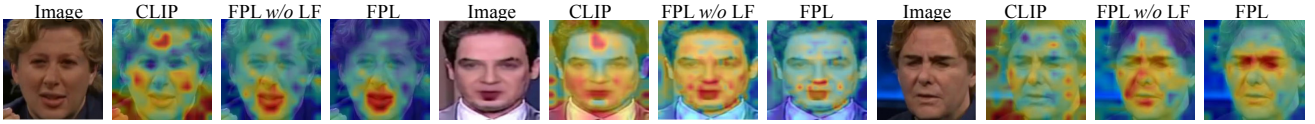


Figure 3. Additional generated forged attention map. [Key: LF: Layer-wise forgery tokens.]

rect judgment along with a convincing description. More qualitative results can be found on our project page.

5. Additional Forged Attention Map

We visualize additional forged attention maps in Fig. 3, where the full version of FPL generates the most accurate forged attention maps. For example, the mouth regions of the first two subjects and the eyes region of the third subject can be identified as forged by the FPL.

6. Limitations and Future Work

We empirically identify two limitations in our proposed method, both of which present opportunities for future research. First, while our model parsing approach delivers excellent performance on the forged face image, it is worth exploring its effectiveness in detecting other forged semantic contents. Also, M2F2-Det’s explanation performance should be evaluated on real faces with unusual decorations, such as funny eyeglasses and silicone masks in the SiW-Mv2 dataset [3].

Secondly, we use a three-stage training strategy to train M2F2-Det, as detailed in Sec. 3.3 of the main paper. Although such a training strategy is effective in optimizing M2F2-Det in both detection and textual explanation performance, it actually would be preferable to have an efficient end-to-end training strategy.

Thirdly, M2F2-Det has two modality outputs, which can have disagreements. For example, the explanation is This image is fake while the fake probability is low from the binary detection branch. We believe this can be a good direction for the future work to resolve.

Fourthly, the full M2F2-Det takes an average of 0.35 seconds per image, but M2F2-Det *without* LLM runs detection 66 frames per second (fps), on par with other detectors (e.g., EfficientNet [7] runs 46 fps). The speed bottleneck is the LLM, which only generates descriptions and can be optional in practice: In general, users can use M2F2-Det *without* the LLM for *efficient* SoTA detection performance; for challenging samples, the full M2F2-Det assists with explanations. In addition, we argue this limitation of inference speed can be alleviated by using a more efficient LLM, e.g.,

MobileVLM [1]. Alternatively, one can propose an alternative approach using the frozen LLM to assist the detection performance [6].

References

- [1] Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, et al. Mobilevlm v2: Faster and stronger baseline for vision language model. *arXiv preprint arXiv:2402.03766*, 2024. 4
- [2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1
- [3] Xiao Guo, Yaojie Liu, Anil Jain, and Xiaoming Liu. Multi-domain learning for updating face anti-spoofing models. In *ECCV*, 2022. 3
- [4] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 1
- [5] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 1
- [6] Xiufeng Song, Xiao Guo, Jiache Zhang, Qirui Li, Lei Bai, Xiaoming Liu, Guangtao Zhai, and Xiaohong Liu. On learning multi-modal forgery representation for diffusion generated video detection. In *NeurIPS*, 2024. 4
- [7] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 1, 3
- [8] Hu Ye, Jun Zhang, Sibao Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 2
- [9] Yue Zhang, Ben Colman, Xiao Guo, Ali Shahriyari, and Gaurav Bharaj. Common sense reasoning for deepfake detection. In *European Conference on Computer Vision*, pages 399–415. Springer, 2024. 2