Towards Generalizable AI-Generated Image Detection via Image-Adaptive Prompt Learning

Supplementary Material

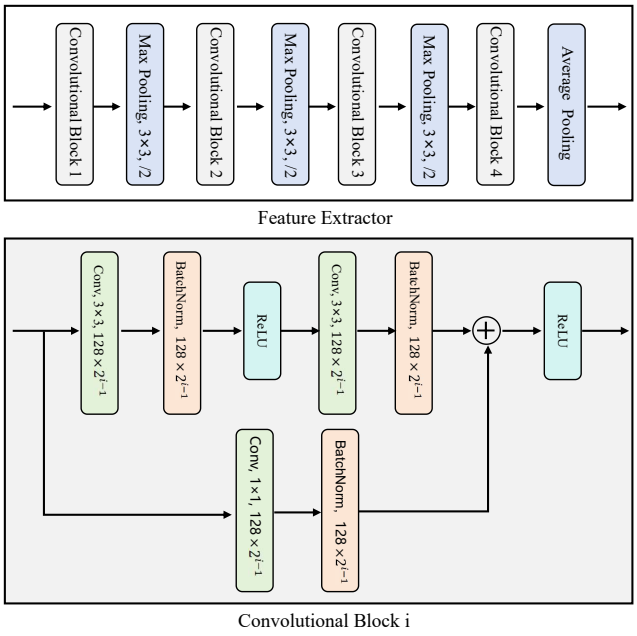

Figure 1. The overall architecture of Feature Extractor.

In the supplementary, we will introduce the detailed structure of Feature Extractor, the detailed calculation process of DCT score, additional visualizations, additional experiments, future work, and failure cases.

1. Feature Extractor

As shown in Fig. 1, we elaborate on the detailed structure of the Feature Extractor used in Conditional Information Learner. It follows these design rules: (1) Several Convolutional Blocks are used to extract features. (2) Max pooling is applied after the first N-1 Convolutional Blocks to reduce the dimensionality of feature maps, where N is the total number of Convolutional Blocks. (3) Average Pooling is used after the final Convolutional Block to obtain global descriptors. In the i-th Convolutional Block, the middle dimension for convolution is set to $128 \times 2^{i-1}$.

2. DCT Score

Following AIDE [6], we use the DCT score to select the rich-texture patch. Specifically, for a patch $p \in R^{M\times M\times 3}$, where M is the size of the patch, N_f different band-pass filters are used to obtain absolute DCT coefficients $C \in R^{N_f\times M\times M\times 3}$. The k-th filters $F_k \in R^{M\times M\times 3}$ could be calculated as Eq. 1.

$$F_{k,ij} = 1,\ if\ \frac{2M}{N_f}\times k \le i+j < \frac{2M}{N_f}\times(k+1), else\ 0, \quad (1)$$

where $F_{k,ij}$ is the weight at the (i, j) position. After that, the obtained DCT coefficients are summed up at all the positions to get the final DCT score as Eq. 2.

$$S = \sum_{k=0}^{N_p} 2^k \times \sum_{t=o}^{2} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} F_{k,ij} \cdot log(|c^k| + 1), \quad (2)$$

where $C^k \in R^{M\times M\times 3}$ is the k-th DCT coefficient, corresponding to $F_{k,ij}$.

Table 1. **Robust Analysis on the UniversalFakeDetect.**

Perturbation	Method	mACC (%)	mAP (%)
Gaussian blurring	SAFE	73.79	65.88
	C2P-CLIP	89.43	97.14
	IAPL(ours)	**90.17**	**97.70**
random cropping	SAFE	87.57	91.22
	C2P-CLIP	93.30	98.70
	IAPL(ours)	**94.92**	**99.26**
Gaussian noising	SAFE	69.76	89.68
	C2P-CLIP	81.57	95.24
	IAPL(ours)	**86.04**	**95.84**
Combined perturbation	SAFE	77.18	76.89
	C2P-CLIP	88.13	96.90
	IAPL(ours)	**90.13**	**97.67**

3. Additional Experiments

Robust Analysis. Inspired by previous method [1], we conduct robust analysis on UniversalFakeDetect as shown in Tab. 1. We adopt Gaussian blurring, random cropping, Gaussian noising, and combined perturbation, with each perturbation applied at a 50% probability. For blurring, the Gaussian kernel size is randomly selected from (3, 5, 7, 9). In the random cropping step, the cropping percentage is uniformly sampled from U(5%, 20%), and the cropped sub-region is upsampled to restore the original resolution. For the Gaussian noising, the variance of the applied Gaussian noise is randomly drawn from the uniform distribution U(5.0, 20.0). For combined perturbation, one of the above-described perturbation methods is randomly selected and applied to the target image. Our method achieves better performance than the recent C2P-CLIP [5] and SAFE

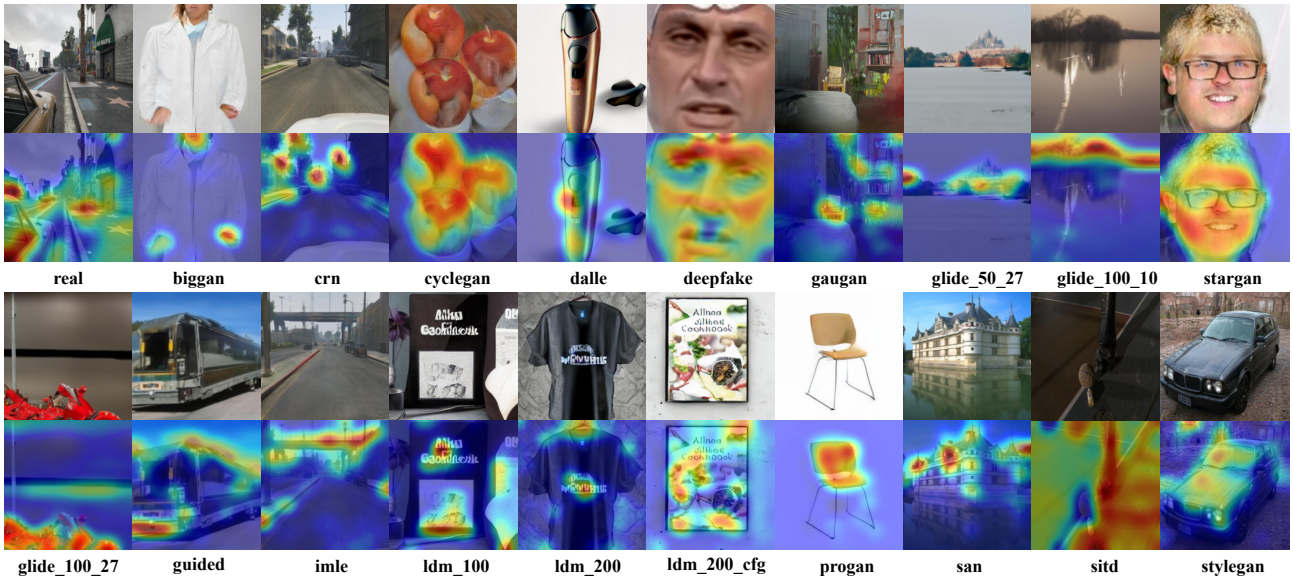

Figure 2. Additional CAM-Grad visualizations.

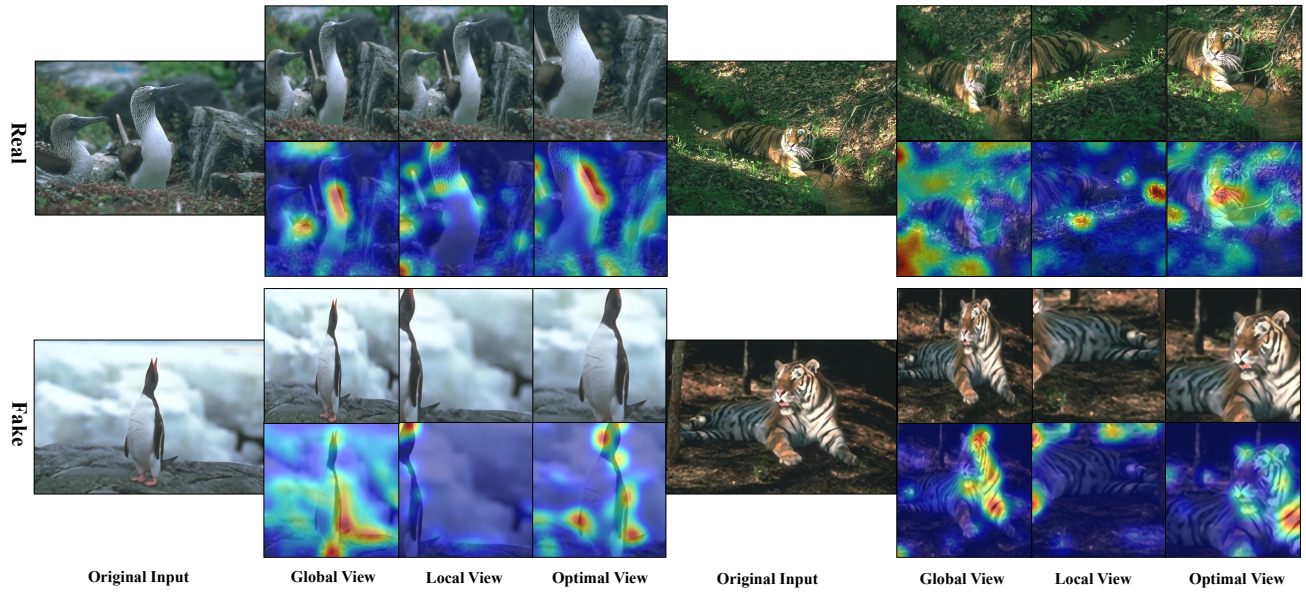

Figure 3. Grad-CAM visualization of different views in Optimal Input Selection. We present the global, local, and optimal views from all generated views. Brighter colors represent the salient region.

[2] across all the aforementioned four settings, and outperforms them by 2.0% and 12.95%, respectively, in terms of mACC under the combined perturbation setting.

4. Inference cost

Table 2. **Inference time cost on the UniversalFake benchmark.**

Metrics	Fatformer	C2P-CLP	Ours (Static)	Ours (T=1)	Ours (T=2)
mAcc (%)	90.86	93.79	93.51	95.25	95.61
Time Cost	0.07s	0.04s	0.04s	0.59s	1.08s

As shown in Tab. 2, on an RTX 3090 + CUDA 11.7 + Torch 2.0.1, our static version achieves 93.51% mAcc at 0.04s per image (matching efficiency of C2P-CLIP and FatFormer), while our dynamic tuning variants (T=1 / T=2) yield notable accuracy gains (95.25% / 95.61% mAcc) with inference times of 0.59s / 1.08s. The dynamic tuning only adjusts 2,048 parameters. Though slower than static methods, this latency is acceptable for synthetic image detection's most typical use cases, e.g., offline content review and forensic analysis, where real-time millisecond

response is not a strict requirement. Furthermore, our method offers a general dynamic strategy that can integrate with advanced static methods for further gains. Finally, we acknowledge inference cost is the main limitation and will optimize it in future work. Additionally, we adopt bf16 for the dynamic test-time tuning.

5. Additional Visualizations

As shown in Fig. 2, we visualize additional CAM-Grad [4] results on the UniversalFakeDetect [3] dataset. As shown in Fig. 3, we visualize the Grad-CAM of global, local, and optimal views in Optimal Input Selection. Across different scenarios, viewpoints, and image categories (real and fake), our method consistently focuses on semantic-rich regions. This suggests that it has learned more discriminative features, which is likely a key reason for its superior performance across various types of forgeries.

6. Future Work and Limitation

To further enhance the generalization ability of forgery detection models and cope with the continuous emergence of novel manipulation techniques, future research can incorporate incremental learning strategies, enabling models to gradually learn new types of forgeries without forgetting previously acquired knowledge. Another important research direction lies in constructing higher-quality datasets that better reflect real-world scenarios, along with exploring corresponding solutions tailored to such data.

Limitation. While performance is improved, the inference time increases due to test-time tuning, which limits the method's applicability in latency-sensitive scenarios. We plan to further optimize in future work.

7. Failure Cases

As shown in Fig. 4, we present several examples of misclassified images. It could be observed that real images captured in simulated environments or from video games are often mistaken as fake ones. Similarly, images taken at night are also likely to be misclassified as fake. This may be due to the model's sensitivity to unnatural textures, lighting conditions, or rendering artifacts that resemble those found in AI-generated images. For fake images, when their texture features closely resemble those of real images and no obvious distortions are present, they are likely to be misclassified as real.

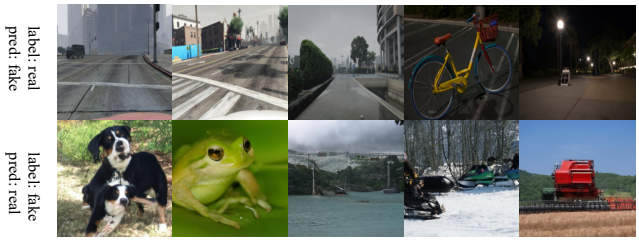

Figure 4. Visualization of failure cases. The first row shows images that are actually real but predicted as fake, while the second row shows images that are actually fake but predicted as real.

References

[1] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*, pages 3247–3258. PMLR, 2020. 1

[2] Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Fuli Feng. Improving synthetic image detection towards generalization: An image transformation perspective. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 2405–2414, 2025. 2

[3] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24480–24489, 2023. 3

[4] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 3

[5] Chuangchuang Tan, Renshuai Tao, Huan Liu, Guanghua Gu, Baoyuan Wu, Yao Zhao, and Yunchao Wei. C2p-clip: Injecting category common prompt in clip to enhance generalization in deepfake detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7184–7192, 2025. 1

[6] Shilin Yan, Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Weidi Xie. A sanity check for ai-generated image detection. *arXiv preprint arXiv:2406.19435*, 2024. 1