Supplementary Material for AUNet: Learning Relations Between Action Units for Face Forgery Detection

Weiming Bai[1,2*] Yufan Liu[1,2*] Zhipeng Zhang[3*] Bing Li[1,4†] Weiming Hu[1,2,5]
[1]State Key Laboratory of Multimodal Artificial Intelligence Systems,
Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]DiDiChuxing [4]People AI, Inc.
[5]CAS Center for Excellence in Brain Science and Intelligence Technology
{baiweiming2019,yufan.liu}@ia.ac.cn zhipeng.zhang.cv@outlook.com {bli,wmhu}@nlpr.ia.ac.cn

1. Details of AU correlation calculation

In Sec. 1 of the main text, we present the average correlation intensity between an AU and other AUs under different number of samples from Real and corresponding Deepfakes videos. Here we perform experiments on different data volumes (*i.e.,* 20%, 80%) of other different forgery categories (*i.e.,* Face2Face, FaceSwap, NeuralTextures). Results are shown in Fig. 1. The calculation process is shown as follows.

We first randomly select m real videos and the corresponding m fake videos from the FF++ dataset (*e.g.,* 200 Youtube videos and their corresponding 200 Deepfakes videos). In each frame of each video, we utilize the open-source tools OpenFace [1] to extract AU Label $\mathbf{L} \in \mathbb{R}^n$, which responds to the occurrence of n AUs. $L_i = 1$ implies that the ith Action Unit appears in this frame. Then we calculate the correlation intensity between ith Action Unit and jth Action Unit by the similar method in [10]:

$$C_{ij} = P(L_j = 1 | L_i = 1), \tag{1}$$

where $\mathbf{C} \in \mathbb{R}^{n \times n}$ is the correlation intensity matrix. Let $\mathbf{C}^r$ and $\mathbf{C}^f$ denote the correlation intensity matrices computed on m Real videos and m corresponding Deepfakes videos, respectively. To measure the degree of association between ith AU and other AUs, we calculated the average correlation intensity $\mathbf{A}^r, \mathbf{A}^f \in \mathbb{R}^n$ by:

$$\mathbf{A}^{\{r,f\}} = \sum_{j=1}^{n} C_{ij}^{\{r,f\}} \Big/ n. \tag{2}$$

*Equal contribution
†Corresponding Author

2. More Implementation Details

In Image-level Tampering, color transformations are performed by shifting the values of RGB channels, hue, saturation, value, brightness, and contrast of input images. Frequency transformations are implemented by downsampling or sharpening input images. At each training step, the model randomly chooses one of the four layers, $block1$, $block3$, $block5$, $block7$ of the Xception backbone to perform Feature-level Mixing. The parameter λ is sampled from $\beta(10, 1)$.

In the inference process, given an image $\mathbf{I}$, we denote the output of the last ART encoder as $[\mathbf{x}_c, \mathbf{x}_1, \cdots, \mathbf{x}_n]$, where $\mathbf{x}_c$ and $\mathbf{x}_1, \cdots, \mathbf{x}_n$ correspond to the class token and n patch tokens, respectively. During inference, we combine the predicted results based on the class token and patch tokens by:

$$\mathbf{I}_{out} = fc(\mathbf{x}_c) + \frac{1}{2} max(fc_s(\mathbf{x}_i)), \tag{3}$$

where fc_s is the shared full-connected layer performed on each patch token.

In the pre-processing, we extracted the regions related to action units by a similar process as in method [7, 11]. Figure 2 shows the examples.

3. Additional Experiments

3.1. More in-dataset evaluation results

In Sec. 4.2 of the main text, we implement in-dataset experiments on FF++. Here we present more in-dataset evaluation results on CDF [9] and DFDCP [4]. The results are shown in Table 1, which demonstrates that our approach achieves superior performance on CDF and the best results on DFDCP.

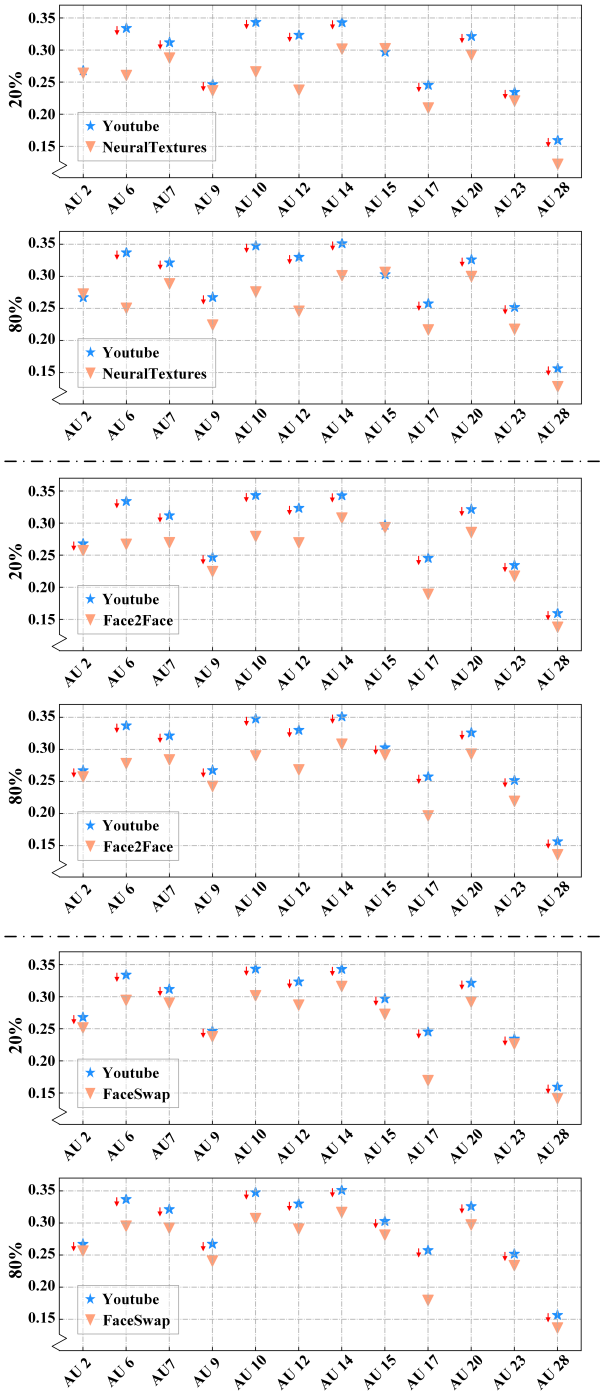

Figure 1. The average correlation intensity (y-axis) between an AU (x-axis) and other AUs under different data volumes, which is calculated on other different manipulation methods (*i.e.,* NeuralTextures, Face2Face, FaceSwap).

3.2. More analysis results

Adaptation to CNN architecture. Our proposed TAP can also be applied to CNN-based networks after serializing their output features. Concretely, we extract the features $F \in R^{C\times H\times W}$ before the global average pooling layer of CNN, and then acquire serialized feature by reshaping feature F into the size of (C, HW). Here, we investigate the performance of different common architectures, i.e., ResNet-34 [6], Xception [3], EfficientNet-b4 [12] trained with TAP. As shown in Table 2, all architectures achieve good results on CDF, DFDCP, and FFIW.

Performing FM on different layers. In the main text, we randomly choose one of the four layers (*i.e.,* $block1$, $block3$, $block5$, $block7$) of the backbone to perform Feature-level Mixing at each training step. Here we train different models with FM applied to different layers. For notation, $block13$ means we randomly choose one of the layers in $block1$, $block3$ of the backbone to apply FM at each training step. The results shown in Table 3 demonstrate that applying FM to multiple shallow layers generally achieves better performance. For instance, $block135$ is better than $block1$. We obtain the best average performance

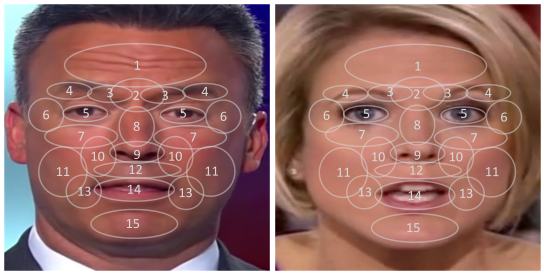

Figure 2. Examples of the extracted AU-related regions.

Method	Testing Set (AUC (%))	
	CDF	DFDCP
Tolosana *et al.* [13]	83.60	91.10
S-MIL-T [8]	98.84	85.11
PCL+I2G [14]	**99.98**	94.38
Gu *et al.* [5]	99.61	92.79
RECCE [2]	99.94	91.33
Ours	99.94	**95.21**

Table 1. **In-dataset evaluation results on CDF, DFDCP.** Our method achieves competitive performance in terms of AUC.

Method	Testing Set AUC (%)			Avg
	CDF	DFDCP	FFIW	
ResNet-34 [6]	90.19	84.85	73.64	82.89
Xception [3]	91.31	83.06	76.60	83.66
EfficientNet-b4 [12]	95.10	85.12	85.84	88.69

Table 2. **Results of applying proposed TAP on CNN architectures.**

Method	Testing Set AUC (%)			Avg
	CDF	DFDCP	FFIW	
$block1$	90.87	81.94	78.43	83.75
$block13$	91.20	84.63	79.98	85.27
$block135$	91.96	85.12	80.78	85.95
$block1357$	92.77	86.16	81.45	86.79
$block13579$	92.04	84.01	81.99	86.01

Table 3. **Results of applying proposed FM on different layers.**

with $block1357$.

References

[1] Openface 2.2.0: a facial behavior analysis toolkit. https://github.com/TadasBaltrusaitis/OpenFace. Accessed 2022-11-15. 1

[2] 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*, pages 4113–4122, 2022. 2

[3] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 2

[4] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854*, 2019. 1

[5] Zhihao Gu, Yang Chen, Taiping Yao, Shouhong Ding, Jilin Li, and Lizhuang Ma. Delving into the local: Dynamic inconsistency learning for deepfake video detection. AAAI, 2022. 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[7] Wei Li, Farnaz Abtahi, Zhigang Zhu, and Lijun Yin. Eac-net: Deep nets with enhancing and cropping for facial action unit detection. *IEEE transactions on pattern analysis and machine intelligence*, 40(11):2583–2596, 2018. 1

[8] Xiaodan Li, Yining Lang, Yuefeng Chen, Xiaofeng Mao, Yuan He, Shuhui Wang, Hui Xue, and Quan Lu. Sharp multiple instance learning for deepfake video detection. In *Proceedings of the 28th ACM international conference on multimedia*, pages 1864–1872, 2020. 2

[9] 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*, pages 3207–3216, 2020. 1

[10] Xuesong Niu, Hu Han, Shiguang Shan, and Xilin Chen. Multi-label co-regularization for semi-supervised facial action unit recognition. *Advances in neural information processing systems*, 32, 2019. 1

[11] Zhiwen Shao, Zhilei Liu, Jianfei Cai, and Lizhuang Ma. Deep adaptive attention for joint facial action unit detection and face alignment. In *Proceedings of the European conference on computer vision (ECCV)*, pages 705–720, 2018. 1

[12] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 2

[13] Ruben Tolosana, Sergio Romero-Tapiador, Julian Fierrez, and Ruben Vera-Rodriguez. Deepfakes evolution: Analysis of facial regions and fake detection performance. In *International Conference on Pattern Recognition*, pages 442–456. Springer, 2021. 2

[14] Tianchen Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. Learning self-consistency for deepfake detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15023–15033, 2021. 2