Domain Invariant Vision Transformer Learning for Face Anti-spoofing Supplementary Material

Chen-Hao Liao[1], Wen-Cheng Chen[2], Hsuan-Tung Liu[3], Yi-Ren Yeh[4], Min-Chun Hu[5], Chu-Song Chen[1]
[1]National Taiwan University, [2]National Cheng Kung University, [3]E.SUN Financial Holding Co., Ltd.,
[4]National Kaohsiung Normal University, [5]National Tsing Hua University
r09922113@csie.ntu.edu.tw, jerrywiston@mislab.csie.ncku.edu.tw, ahare-18342@esunbank.com.tw,
yryeh@nknu.edu.tw, anitahu@cs.nthu.edu.tw, chusong@csie.ntu.edu.tw

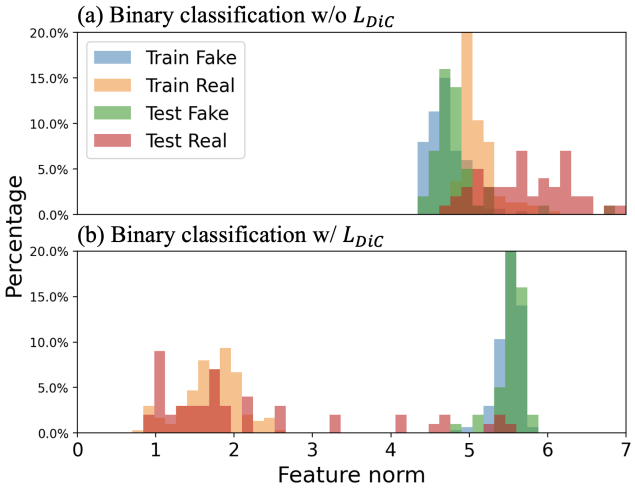

Figure 1. Histogram of the feature norms on O&C&I to M setting. The lower half figure indicates that our concentration loss can effectively pull the real faces' feature embedding from the unseen domain towards the origin.

1. Visualization of Feature Norm

To confirm that our domain-invariant concentration loss can effectively pull real features to the origin, we plot the distribution of feature norms. We randomly sample the same amount of real and fake data in each dataset for better visualization. Figure 1 shows the norm distribution of feature embeddings with or without using L_{DiC} on the binary classification MobileViT-S in the M & C & I to O setting. The feature norm distribution when using concentration loss is shown in Figure 1(b). It shows that the loss effectively pulls real-face embeddings to the origin compared to binary classification models (Figure 1(a)), and can be generalized to unseen domains.

2. Using l_2-norm in L_{DiC}

We have ablated our L_{DiC} loss by changing the l_1-norm in Equation 2 in the main paper to l_2-norm. The results are shown in Table 1. In this case, using l_1-norm in L_{DiC} leads to slightly better performance. This may be because l_1-norm generally encourages sparsity, which brings additional benefits.

3. Pushing fake features

The idea of pulling features to the origin is also investigated in action localization [17]. This approach not only pulls the features of the background class towards the origin, but also pushes the features of other classes away from the origin to make the embeddings more discriminative. Likewise, we also conduct experiments to further push the spoofed features away from the origin. The following loss takes over the L_{DiC},

$$L_{Contra-k} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[x_i \in \mathbf{D^R}] \cdot \|f_i\|_k + \mathbb{1}[x_i \notin \mathbf{D^R}] \cdot \max(0, m - \|f_i\|_k),$$

where k represents the k-norm the loss adopted and m is a hyper-parameter to penalize the spoofed features which are not far enough from the origin. Others follow the same notations in Sec. 3.1 of the main paper.

The results are shown in Table 2. We evaluate the performance of using l_1- and l_2-norm. Although pushing spoofed features away from the origin could make the boundary between real and spoofed features more separated, it does not improve the performance in our case. The reason might be that we already use a separation loss, which can also push real and spoofed faces away from each other; thus the extra loss terms cannot further improve the performance.

4. Effect of random sampling training method

During the training phase, we follow the same setting in [13] to randomly sample one image in each video as training data. We run the experiment five times by sampling different images in each video in these experiments. The mean and standard deviation of the results for each setting on DiVT-M are shown in Table 3. As can be seen, because of the minor discrepancy among different frames in a video, the variances in our approach are relatively small.

5. Different kinds of training strategies in [10]

As described in Sec. 4.4.1 of the main paper, the work in [10] has also done the experiments of adapting the classification layer only and obtained favorable results. We also test this strategy with our method on the leave-one-out protocol. We train the classification layer of the transformer and fix the weights of the backbone by using a binary cross-entropy loss, as shown in Table 4. However, the results show that the performance is much worse when the backbone weights are fixed and only the classification layer is trained. The reason could be that we have multiple domains in the training stage, freezing the backbone severely limits the model's capability on exploiting the information across domains.

Method	O&C&I to M		O&M&I to C		O&C&M to I		I&C&M to O		Average	
	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)
DiVT-M (l_1)	**2.86**	**99.14**	**8.67**	**96.92**	**3.71**	**99.29**	13.06	94.04	**7.07**	**97.34**
DiVT-M (l_2)	4.29	98.91	13.33	94.73	7.07	97.43	**12.81**	**94.61**	9.38	96.42

Table 1. Performance of using l_1- and l_2-norm in our L_{DiC}

Method	O&C&I to M		O&M&I to C		O&C&M to I		I&C&M to O		Average	
	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)
DiVT-M	**2.86**	**99.14**	**8.67**	**96.92**	**3.71**	**99.29**	13.06	94.04	**7.07**	**97.34**
MobileViT + L_{DiA}^{ce} + $L_{Contra-1}$	7.14	98.02	9.44	96.39	10.71	95.60	**12.07**	94.82	9.84	96.20
MobileViT + L_{DiA}^{ce} + $L_{Contra-2}$	6.19	98.12	9.44	96.42	7.14	97.75	12.40	**95.04**	8.79	96.83

Table 2. Performance of pushing spoofed features from the origin by using $L_{Contra-k}$ (k means we are using l_k-norm in the loss).

Method	O&C&I to M		O&M&I to C		O&C&M to I		I&C&M to O	
	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)
DiVT-M	5.43±1.63	98.75±0.30	10.51±1.61	95.88±1.03	6.46±2.03	98.15±1.16	11.98±0.89	95.20±0.85

Table 3. Mean and standard deviation of our DiVT-M's performance.

Methods	O&C&I to M		O&M&I to C		O&C&M to I		I&C&M to O	
	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)
ViT-Base(FC)	25.71	85.29	31.33	76.60	44.29	59.31	28.63	77.94
ViT-Tiny(FC)	21.67	86.99	30.00	76.46	48.57	52.80	33.07	74.83
Swin-T(FC)	30.00	76.64	21.89	85.07	31.36	76.64	34.44	70.43
MobileViT-S(FC)	17.14	87.91	22.00	82.92	35.71	69.05	25.97	80.93

Table 4. Performance of different backbones when only training the classification layer. The weights of backbone models are fixed.