Appendix for: Controllable Guide-Space for Generalizable Face Forgery Detection

Ying Guo *, Cheng Zhen *, Pengfei Yan †
Vision AI Department, Meituan
{guoying16, zhencheng02, yanpengfei03}@meituan.com

1. Theoretical proof of the feature purity

In this section, we present the theoretical analysis that higher feature purity (i.e., contains more task-relevant information) will help the generalization.

For the entire forgery detection task, we let $p(x, y)$ represent the ground-truth joint probability distribution corresponding to data x and label y. $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Ideally, we want to get a model $f(x;\theta) : \{\mathcal{X};\Theta\} \rightarrow \mathcal{Y}$, $\theta \in \Theta$, which minimizes the following objective function during the training process [4]:

$$\min_{f} F(f) = \int \mathcal{L}(f(x;\theta), y) dp(x, y) \tag{1}$$

where $\mathcal{L}$ is the loss function in the training.

However, in the actual training process, we cannot know the ground-truth probability distribution $p(x, y)$, but usually use a training set D_{train} that we can obtain, and approximate Eq. (1) through average calculation. Let I denote the number of data, the actual training target of the corresponding model $f_1(x;\theta_1)$ is:

$$\begin{gathered} \min_{f_1} F_{actual}\left(f_1\right) = \frac{1}{I} \sum_{i=1}^{I} \mathcal{L}\left(f_1\left(x_i;\theta_1\right), y_i\right) \\ \text{s.t. } \left(x_i, y_i\right) \in D_{train} \end{gathered} \tag{2}$$

Comparing Eq. (1) and Eq. (2), the model f_1 obtained is not close to the ideal f well due to the deviation of D_{train} to $p(x, y)$ and the average approximation. When D_{train} and $p(x, y)$ are biased, model f_1 may satisfy the goal of Eq. (2) by learning some "shortcut features" [2] which exist in the bias part and are not relevant to the forgery detection task. Therefore, when faced with unseen domain data outside D_{train}, f_1 does not apply well, resulting in weak generalization. On the contrary, if we make the features of f_1 have as few forgery-irrelevant features as possible from the bias part (i.e., the feature purity is as high as possible), then f_1 will be more approximate to f, thus achieving better generalization.

*Equal Contribution. †Corresponding author.

2. Solving details for Eq. (2)

In this section, we show the details of solving Eq. (2) of the paper under the constraints of Eq. (1).

As we mentioned in the paper, $\boldsymbol{g_r}$ represents the guide embedding of the real domain obtained by random initialization, and $\left\{\boldsymbol{g_{f_i}}\right\}_{i=1}^{N}$ is the guide embedding of forgery domains that needs to be solved. We first use $\delta\left(\boldsymbol{g_{f_i}}\right)$ to represent the constraints in Eq. (1) of the original paper:

$$\delta\left(\boldsymbol{g_{f_i}}\right) = e^{\boldsymbol{g_r}^T \boldsymbol{g_{f_i}}} - e^{\cos(\theta_0)} = 0 \quad (i = 1, \cdots, N) \tag{3}$$

Then we aim to minimize Eq. (2) of the original paper, subject to the constraints of $\delta\left(\boldsymbol{g_{f_i}}\right)$, which is formulated as:

$$\begin{gathered} \min\ L\left(\left\{\boldsymbol{g_{f_i}}\right\}_{i=1}^{N}\right) = \frac{1}{N} \sum_{i=1}^{N} \log \sum_{j=1}^{N} e^{\boldsymbol{g_{f_i}}^T \boldsymbol{g_{f_j}} / \tau} \\ \text{s.t. } \delta\left(\boldsymbol{g_{f_i}}\right) = 0 \quad (i = 1, \cdots, N) \end{gathered} \tag{4}$$

We solve this based on the Lagrangian multiplier method [1]. Let ω_i denote the Lagrangian multiplier, then the new solution function $\mathcal{H}(\cdot)$ can be constructed as:

$$\mathcal{H}\left(\left\{\boldsymbol{g_{f_i}}\right\}_{i=1}^{N}\right) = L\left(\left\{\boldsymbol{g_{f_i}}\right\}_{i=1}^{N}\right) + \sum_{i=1}^{N} \omega_i \cdot \delta\left(\boldsymbol{g_{f_i}}\right) \tag{5}$$

By calculating the partial derivatives of $\mathcal{H}$ to $\boldsymbol{g_{f_i}}$ and ω_i and setting them to 0, $\left\{\boldsymbol{g_{f_i}}\right\}_{i=1}^{N}$ can be obtained:

$$\nabla_{\boldsymbol{g_{f_i}}} \mathcal{H} = \frac{\partial \mathcal{H}}{\partial \boldsymbol{g_{f_i}}} = \nabla L + \omega_i \nabla \delta = \mathbf{0} \tag{6}$$

$$\nabla_{\omega_i} \mathcal{H} = \frac{\partial \mathcal{H}}{\partial \omega_i} = \delta\left(\boldsymbol{g_{f_i}}\right) = 0 \tag{7}$$

3. More details on hyper-parameters

***k* in A-DBM:** k is $|K_i|$ in Eq. (5) of the paper. When k=10, 30, 50, 55, 60, 80, and 100, the AUCs on CelebDF are 79.35,

Train Set	DF F2F NT				DF F2F FS			
Test Set	FS (HQ)		FS (LQ)		NT (HQ)		NT (LQ)	
	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC
L_{ce-2}	76.61	85.89	91.82	96.99	79.02	87.28	81.93	90.28
$L_{ce-(1+N)}$	77.65	85.16	91.85	97.07	81.26	88.14	82.63	90.76
L_{guide}	**78.44**	**86.95**	**92.33**	**97.34**	**82.07**	**89.13**	**83.95**	**92.04**

Table 1. Comparisons of methods that increase the discrimination of different domains, including the results of FS and NT as the test set under the cross-test setting within FF++.

Train Set	DF F2F NT				DF F2F FS			
Test Set	FS (HQ)		FS (LQ)		NT (HQ)		NT (LQ)	
	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC
w/o L_{guide}	79.92	88.26	95.05	98.17	82.91	91.49	86.32	94.46
w/o L_{pull}&L_{push}	82.51	90.54	96.93	99.25	84.79	92.17	87.42	95.03
w/o L_{pull}	83.48	92.69	97.05	99.31	86.94	93.86	88.56	95.91
w/o L_{push}	84.35	93.07	97.24	99.40	87.30	94.83	88.67	96.14
w/o A-DBM	80.14	88.79	95.67	98.21	85.73	93.21	87.49	94.75
ours	**86.32**	**94.11**	**97.90**	**99.68**	**88.04**	**96.15**	**89.95**	**97.12**

Table 2. Ablation performance after removing each module of the method, including the results of FS and NT as the test set under the cross-test setting within FF++.

80.65, 83.02, 84.97, 84.91, 84.89, and 84.95. Stability is reached when k=55. When k=200, AUC drops to 81.96. In our training, each forgery domain has about 20,000 data, and the range of $55/20000$=0.275% can be regarded as the nearest neighbor.

The number of clusters: In the decoupling module, we use clustering based on self-supervised features to explore potential similarities between data. When the number of clusters is 100, 300, 500, 700, and 1000, the AUCs on CelebDF are 79.96, 81.78, 84.97, 83.65, and 83.42. When the number is small, it is easy to group less similar data into one cluster, and separating these data does not serve the purpose of decoupling irrelevant similarities well. When the number is large, it will cause similar data to be divided into different clusters, and when we conduct pushing operation, these data are not covered, so the performance will be reduced. When the number is 500, optimal performance is achieved.

4. Computational cost

For FLOPS, A-DBM calculates the nearest neighbor matrix, and this increase is 0.104% of EN-B4 FLOPS, so the time consumption will not increase significantly. For memory consumption, the decoupling model needs to store a feature set V, and this increase is only 10M. Our method focuses on the loss functions, so model parameters are not changed.

5. Additional experiments

In this section, we show more results of our method on cross-test setting within FF++ to demonstrate the effectiveness of our method in multiple experimental settings. In the first two parts of this section, we show the ablation results of using FS and NT as the test set. In the third part, we show the comparison with other recent methods.

5.1. Methods to distinguish forgery domains

For methods of enhancing the forgery domain discrimination, we regard results based on the binary cross-entropy loss L_{ce-2} as the baseline. Based on this, we compare the multi-classification cross-entropy loss $L_{ce-(1+N)}$ that can also distinguish multiple forgery domains. The performance comparisons of L_{ce-2}, $L_{ce-(1+N)}$, and our L_{guide} with FS and NT as the test set are shown in Table 1.

Similar to the results on DF and F2F, on FS and NT, our L_{guide} achieves the best performance among the three losses. For example, on the NT dataset, the AUCs on HQ and LQ are 1.85% and 1.76% higher than L_{ce-2}, and 0.99% and 1.28% higher than $L_{ce-(1+N)}$, respectively. For $L_{ce-(1+N)}$, it outperforms L_{ce-2} in most cases, but on FS (HQ), its AUC is 0.73% lower than L_{ce-2}. This shows that it is not feasible to simply regard distinguishing different domains as an ordinary multi-classification task. To improve generalization, we need to keep the real domain far enough away from forgery domains to cope with the complexity of the

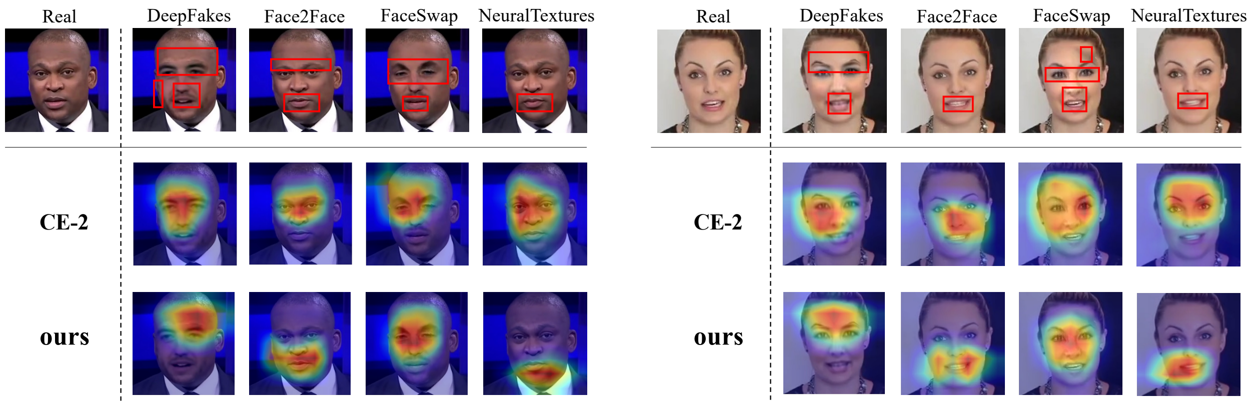

Figure 1. The heatmap comparisons of binary cross-entropy (CE-2) and our method. Forgery artifacts are marked in red frames.

forgery domain, while also ensuring the distinction between the forgery domains. That is, the separation degree between real and forgery should be much larger than the degree between forgery and forgery. Our guide-space based method does this well and thus achieves good performance.

5.2. Importance of different modules

Table 2 lists the performance of our method on FS and NT as the test set when each key module of our method is removed respectively. It can be seen that each module contributes to the overall performance, and its removal will lead to a decrease in performance. Both L_{guide} and $L_{pull}\&L_{push}$ can achieve the separation of different domains and the aggregation of the same domain, but removing L_{guide} has a greater impact. This is because guide embeddings can achieve the controllability of separation and aggregation, and the decoupling model enhances this discriminativeness by reduce the interference of irrelevant similarities between domains. For the A-DBM module, it has different influences on different datasets. For example, on FS (HQ), removing it will reduce AUC by 5.32%, and on NT (LQ), AUC will decrease by 2.37%. Overall, A-DBM focuses on weak samples in the optimization process and plays an important role in the overall performance.

5.3. Cross test on FF++

In cross-test setting within FF++, we compare the performance of our method and the recent methods. In Table 3, we compare the results of DCL [6], Face X-ray [3], and Xception[5]. It can be seen that under DF, F2F, FS, and NT, our method achieves optimal performance. Under NT, DCL [6] achieves the sub-optimal performance, and ours is 2.3% higher than it.

6. More visualizations

In this section, we show more heatmaps of binary cross-entropy (CE-2) and our method, and these visualizations are shown in Figure 1.

Training Set	Train on remaining three			
Testing Set	DF	F2F	FS	NT
Xception [5]	93.9	86.8	51.2	79.7
Face X-ray [3]	99.5	94.5	93.2	92.5
DCL [6]	95.7	98.2	91.5	93.9
Ours	**99.8**	**98.9**	**94.1**	**96.2**

Table 3. Cross-test within FF++ (HQ). Generalization performance AUC (%) when testing on one type after training on the remaining three types.

Similar to the results shown in Figure 6 of the paper, for CE-2, there are certain similarities in the areas that the models focus on under different forgery types, and they are concentrated in the central area of the face. While the areas that our method focuses on are the respective artifacts corresponding to different forgery types. For face-swapping methods (DeepFakes and FaceSwap) that replace the whole face, it is reasonable for the model to focus on either the central area of the face or the boundary artifacts. For the face reenactment methods (Face2Face and NeuralTextures), the forgery traces are mainly in local areas such as the mouth and eyes. But due to the interference of forgery-irrelevant similarities between different forgery methods, CE-2 still focus on the central area of the face similar to the face-swapping methods, and does not extract the distinguishable features of F2F and NT well. In contrast, our method can pay attention to the corresponding forgery traces and extract better forgery-related features.

References

[1] B. Beavis and I. Dobbs. *Optimisation and stability theory for economic analysis*. Cambridge university press, 1990.

[2] K. Hermann and A. Lampinen. What shapes feature representations? exploring datasets, architectures, and training. *Advances*

in Neural Information Processing Systems, 33:9995–10006, 2020.

[3] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5001–5010, 2020.

[4] Y. Li, C. Wang, L. Yangning, H.-T. Zheng, and Y. Shen. Learning purified feature representations from task-irrelevant labels. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 01–08. IEEE, 2022.

[5] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1–11, 2019.

[6] K. Sun, T. Yao, S. Chen, S. Ding, J. Li, and R. Ji. Dual contrastive learning for general face forgery detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2316–2324, 2022.