Generalizable Representation Learning for Mixture Domain Face Anti-Spoofing

Zhihong Chen[1,2*], **Taiping Yao**[2*], **Kekai Sheng**[2], **Shouhong Ding**[2†], **Ying Tai**[2],
Jilin Li[2], **Feiyue Huang**[2], **Xinyu Jin**[1]

[1]College of Information Science & Electronic Engineering, Zhejiang University,
[2] Youtu Lab, Tencent
{zhihongchen, jinxy}@zju.edu.cn, {taipingyao, ericshding, yingtai, jerolinli, garyhuang}@tencent.com,
Shengkekai_D@163.come

Abstract

Face anti-spoofing approach based on domain generalization (DG) has drawn growing attention due to its robustness for unseen scenarios. Existing DG methods assume that the domain label is known. However, in real-world applications, the collected dataset always contains mixture domains, where the domain label is unknown. In this case, most of existing methods may not work. Further, even if we can obtain the domain label as existing methods, we think this is just a sub-optimal partition. To overcome the limitation, we propose domain dynamic adjustment meta-learning (D^2AM) without using domain labels, which iteratively divides mixture domains via discriminative domain representation and trains a generalizable face anti-spoofing with meta-learning. Specifically, we design a domain feature based on Instance Normalization (IN) and propose a domain representation learning module (DRLM) to extract discriminative domain features for clustering. Moreover, to reduce the side effect of outliers on clustering performance, we additionally utilize maximum mean discrepancy (MMD) to align the distribution of sample features to a prior distribution, which improves the reliability of clustering. Extensive experiments show that the proposed method outperforms conventional DG-based face anti-spoofing methods, including those utilizing domain labels. Furthermore, we enhance the interpretability through visualization.

Introduction

Despite recent significant progress, the security of face recognition systems is still vulnerable to presentation attacks (PA), *e.g.,* photo, video replay, or 3D facial mask. To cope with these presentation attacks, face anti-spoofing (Tan et al. 2010; Liu, Jourabloo, and Liu 2018) is deployed as a pre-step prior to face recognition. Various face anti-spoofing methods have been proposed, which assume that there are inherent differences between live and spoof faces, such as color textures (Boulkenafet, Komulainen, and Hadid 2016), image distortion cues (Wen, Han, and Jain 2015), temporal variation (Shao, Lan, and Yuen 2017), or deep features (Yang, Lei, and Li 2014; Zhang et al. 2020). Although these methods achieve promising performance in intra-dataset experiments, the performance dramatically degrades under a
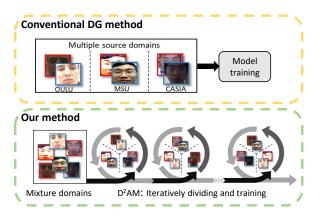

Figure 1: Unlike conventional DG method that requires domain labels, our method can iteratively assign pseudo domain labels and be trained using meta-learning by D^2AM.

cross-domain dataset. To improve generalization ability under unseen situations, a variety of domain generalization (DG)-based methods (Wang et al. 2020; Shao, Lan, and Yuen 2020; Qin et al. 2019; Jia et al. 2020; Saha et al. 2020) are proposed by leveraging domain labels from multiple source domains.

These DG-based methods require domain labels that indicate where each sample in multiple source domains comes from. However, in a more practical scenario, we may obtain a mixture domain dataset, in which the domain label of each sample is unknown as shown in Figure 1. The existing DG methods may not work if the domain label is not available. There are mainly three challenges to relax the constraint: (1) While it could be solved by assigning the domain label manually, it can be very expensive and time-consuming; (2) Even worse, since the domain information of face acti-spoofing is composed of various factors such as illumination, background, camera type, *etc.,* it is not clear how to define the domain and divide the mixture domains; (3) Further, even if we can obtain the domain label which defined as the dataset to which the sample belongs, as existing methods, we think this is just a suboptimal partition. The reason is that the samples among different datasets may partially overlap in distribution, especially when the model can extract a certain degree of domain invariant features in the middle stage of training,

which may cause the model to be unable to focus on better-generalized learning directions due to the small distribution difference between domains.

To relax the constraint that DG-based methods need domain labels, and ensure more difficult and abundant domain differences, as shown in Figure 1, we propose a generalizable face anti-spoofing method, named **d**omain **d**ynamic **a**djustment **m**eta-learning (D^{2}AM), which iteratively divides mixture domains via discriminative domain representation for meta-learning. Specifically, to define the domain information, considering that Instance Normalization (IN) in the networks can alleviate the domain discrepancy (Zhou et al. 2019), we exploit a stack of convolutional feature statistics (*i.e.,* mean and standard deviation) to get domain representation. To extract the domain-discriminative feature, we design a domain representation learning module (DRLM) to extract discriminative domain features under the guidance of the channel attention mechanism. Further, a domain enhancement entropy loss is added to DRLM to enhance the confusion of task discrimination information in domain channels. Once discriminative domain representation is obtained, our method iteratively assigns pseudo domain labels by clustering, and trains a domain-invariant feature extractor by meta-learning. In addition, to prevent outliers from affecting the performance of clustering, we introduce an MMD-based regularization in the adaptation layer, *i.e.,* the previous layer of the output layer, to reduce the distance between the sample feature distribution and the prior distribution. Furthermore, the embedding of MMD-based regularization can encourage the model to learn to correct the distribution of unseen samples through meta-learning.

The main contributions of this work are summarized as follows: (1) We propose a novel and realistic mixture domain face anti-spoofing scenario and design **d**omain **d**ynamic **a**djustment **m**eta-learning (D^{2}AM) to address this scenario. (2) Domain representation learning module (DRLM) and MMD-based regularization are designed for better dynamic adjustment. (3) Extensive experiments and visualizations are presented, which demonstrates the effectiveness of D^{2}AM against the state-of-the-art competitors.

Related Work

Face Anti-Spoofing   Recent face anti-spoofing approaches can be roughly classified into three categories: conventional approaches, deep learning approaches, and domain generalized approaches. Conventional approaches detect attacks by texture cues, which adopt hand-craft features to differentiate real/fake faces, such as LBP (de Freitas Pereira et al. 2014), HOG (Gragnaniello et al. 2015), SURF (Boulkenafet, Komulainen, and Hadid 2016), SIFT (Patel, Han, and Jain 2016), *etc*. With the recent success of deep learning in computer vision, various deep methods are employed. In (Yang, Lei, and Li 2014), discriminative deep features are extracted by CNN for real/fake faces classification. Liu *et al.* (Liu, Jourabloo, and Liu 2018) propose a CNN-RNN architecture, which leverages face depth and rPPG signal estimation as auxiliary supervision to assist in attacks detect. And the work in (Jourabloo, Liu, and Liu 2018) inversely decompose a spoof face into a live face and a noise of spoof

for classification. Although these methods work well under intra-dataset scenarios, their performance becomes degraded in unseen scenarios. In light of this, some domain generalized approaches are proposed. Shao *et al.* (Shao et al. 2019) propose to learn domain-invariant representation by multi-adversarial deep domain generalization for face anti-spoofing, while Jia *et al.* (Jia et al. 2020) design the single-side adversarial learning and the asymmetric triplet loss to further improve the performance. The most related work to ours is proposed in (Shao, Lan, and Yuen 2020), where meta-learning is explored with domain knowledge for generalizable face anti-spoofing. However, this method requires domain labels, which are not satisfied in the novel scenario, *i.e.,* mixture domain face anti-spoofing.

Deep Domain Generalization   Several deep DG methods have been proposed. Ghifary *et al.* (Ghifary et al. 2015) match the feature distributions among multiple source domains by using an auto-encoder. The work in (Li et al. 2018b) aligns multiple domains to a pre-defined distribution via adversarial learning. MLDG (Li et al. 2018a) designs a model-agnostic meta-learning for DG. Note that, the work (Matsuura and Harada 2020) has achieved DG by clustering a mixture of multiple latent domains. However, the domain features they extracted may contain task discriminative information because they did not consider distilling that information, which may hinder the domain clustering. While our method designs a DRLM with style enhancement entropy loss, which encourages model to extract domain-discriminative features for better domain dividing.

Our Approach

Problem Definition and Notations   Conventional generalizable face anti-spoofing methods train the model that accurately works for the unseen domain $\mathcal{D}_t$ by using multiple source domains $\mathcal{D}_{ms} = \{(x_i^s, y_i^s, d_i^s)\}_{i=1}^{n_s}$, where x_i^s is the input image, y_i^s is the task label and d_i^s is the domain label. However, as mentioned above, the dataset may be a mixture of multiple latent domains, in which case it is difficult to get domain labels manually. Our goal is to improve generalization of the model through a mixture domain dataset $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ without the domain label d_i^s.

Overview of D^{2}AM   Figure 2 shows the overall flowchart of our framework. Our method iteratively clusters domain-discriminative representation to reassign pseudo domain label to each sample, which can achieve simultaneously more abundant and more difficult domain shift scenarios for meta-learning. In summary, in each epoch, our method consists of two stages. *In the first stage* (blue flow), the pseudo domain label is assigned by clustering with discriminative domain representation. Specifically, due to the clustering features extracted directly from the network without processing contain more task discrimination information, we process the features extracted by the model based on IN and convert them into domain features. Moreover, we design the DRLM module with domain enhancement entropy loss to encourage the model to extract discriminative domain features that without task discrimination information, so as to avoid the
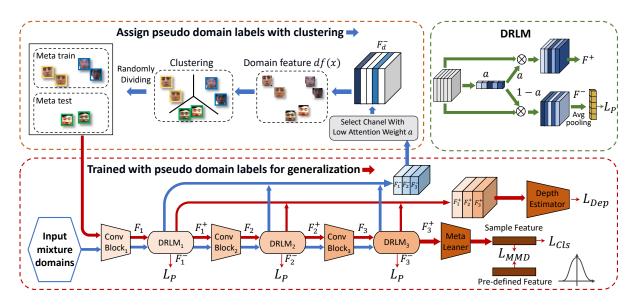
Figure 2: Overview of our method. Each epoch consists of two stages. In the first stage (blue flow), the pseudo domain label is assigned by clustering with discriminative domain representation. In the second stage (red flow), we train a face anti-spoofing model with meta-learning based on pseudo domain label. The green box is DRLM, which utilizes a Squeeze-and-Excitation (SE)-like framework to extract task-discriminative features F^+ and domain-discriminative features F^-.

model's ability to extract domain invariant features from hindering the extraction of domain-discriminative features. *In the second stage* (red flow), we train a face anti-spoofing model with meta-learning based on pseudo domain label. We also incorporate an MMD-based regularization into feature learning process, which regularizes the feature space for better clustering and promotes the model to correct the distribution of unseen samples. The whole process is summarized in Algorithm 1, and details are described as follows.

Dynamically Assigning Pseudo Domain Labels

Domain Feature Define   We assume the domain information of an image can be represented by its style. IN performs a form of style normalization by normalizing feature statistics (Huang and Belongie 2017), which can be formed as:

$$IN(F) = \gamma(\frac{F - \mu(F)}{\sigma(F)}) + \eta, \qquad (1)$$

where $F \in \mathbb{R}^{H \times W \times C}$ is the convolutional feature and H, W, C denote the height, width, and number of channels, respectively, $\gamma, \eta \in \mathbb{R}^C$ are learnable parameters, mean $\mu(\cdot) \in \mathbb{R}^C$ and standard deviation $\sigma(\cdot) \in \mathbb{R}^C$ are computed across spatial dimensions independently for each channel.

$$\mu_c(F) = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (F_{hwc}), \qquad (2)$$

$$\sigma_c(F) = \sqrt{\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (F_{hwc} - \mu_c(F))^2 + \epsilon}. \qquad (3)$$

Since the normalization by μ and σ can alleviate domain discrepancy, we exploit a stack of convolutional feature statistics getting from multiple layers of the feature extractor

to represent domain features. Hence, the domain feature can be defined as: $df(x) = \{\mu(F_1), \sigma(F_1), ..., \mu(F_M), \sigma(F_M)\}$, where M represents the Mth convolutional layer.

Domain Representation Learning Module   To obtain more discriminative domain features $df(x)$, we design the DRLM to extract domain-discriminative convolutional feature. Specifically, as shown Figure 2, it is a SE-like framework, which is expected that channels with high attention contain more *task information*, while channels with low attention contain more *domain information*. Therefore, the module can be used to extract the task-discriminative feature F^+ and domain-discriminative feature F^- by:

$$\boldsymbol{a} = Sigmoid(W_2 ReLu(W_1 pool(F))), \qquad (4)$$

$$F^+ = \boldsymbol{a} \cdot F, \qquad (5)$$

$$F^- = (1 - \boldsymbol{a}) \cdot F, \qquad (6)$$

where $F^+ \in \mathbb{R}^{H \times W \times C}$ and $F^- \in \mathbb{R}^{H \times W \times C}$ are extracted through masking F by a learned channel attention vector $\boldsymbol{a} = [a_1, a_2...., a_C] \in \mathbb{R}^C$. *pool* is a global pooling layer, $W_1 \in \mathbb{R}^{C \times (C/\tau)}$ and $W_2 \in \mathbb{R}^{(C/\tau) \times C}$ are trainable weights, and τ is the dimension reduction ratio.

Minimizing the entropy regularization $-p \log p$ favors a low-density separation between classes and increases task discrimination (Grandvalet and Bengio 2005; Chen et al. 2020). Hence, we utilize reverse entropy loss, named domain enhancement entropy loss, to regularize F^- to discard task discrimination information for better clustering. The loss can be formed as:

$$L_p = P(F^-) \log P(F^-),$$
$$P(F^-) = Sigmoid(W_p pool(F^-)), \qquad (7)$$

where $P(\cdot)$ represents the probability of whether it is a positive sample, and $W_p \in \mathbb{R}^{C \times 1}$ is the trainable weights. Note

that whether it is a positive sample or a negative sample, L_p can constrain $P(\cdot)$ to around 0.5, which means the task discrimination information of F^- may be discarded, so as not to hinder domain clustering.

Clustering with Discriminative Domain Representation
For obtaining pseudo domain labels, since channels with low attention weights contain more domain information, we sort the attention weights in ascending order and select the channel features of F^- corresponding to the first $C/2$ values to form the discriminative domain convolutional feature F_d^-. To further remove task discrimination information, after obtaining domain features $\{df(x_i)\}_{i=1}^{n_s}$ through F_d^- for all samples, we perform K-means clustering (MacQueen et al. 1967) on the positive and negative samples respectively, assign a domain label to each sample, and finally, combine the positive and negative samples with the same domain label into one domain. The problem here is that clustering cannot properly decide which domain label should be assigned to each cluster, which may lead to a mismatch between positive and negative domains and negatively affects the training.

To solve this problem, in the first epoch, we use the ResNet (He et al. 2016) pre-trained on ImageNet to extract domain features to divide all samples into K domains for correct positive and negative domain matching. After that, in each epoch, positive and negative samples are clustered using domain-discriminative features of our face anti-spoofing model and then combined. Based on (Matsuura and Harada 2020), we use Kuhn-Munkres algorithm (Munkres 1957) to ensure the reassigned pseudo domain labels are not shifted largely with those from the previous epoch.

MMD-Based Regularized Meta-Learning
After obtaining pseudo domain labels, we randomly use $K-1$ domains as the meta-train $D_i^s(i = 1, ..., K-1)$ and the remaining one domain as the meta-test D^t in each iteration.

Meta-Train We sample the batches $B_i(i = 1, ..., K-1)$ in each meta-train domain, and perform cross-entropy classification in each meta-train domain as follows:

$$L_{Cls(B_i)}_{\theta_E,\theta_M} = \sum_{(x,y)\in B_i} y \log M(E(x)) + (1-y) \log(1 - M(E(x))),$$
(8)

where θ_E and θ_M represent the parameters of the feature extractor and the meta learner. To avoid the harmful influence of outliers on clustering, we propose the MMD-based regularization to constrain the feature space, which narrows the distance between outliers and sample dense areas, and encourages the meta-learner to correct unseen samples distribution. Specifically, this regularization reduces the distance between the sample feature distribution and the prior distribution on the adaptation layer, *i.e.,* the previous layer of the output layer in the meta leaner, which can be formed as:

$$L_{MMD(B_i)}_{\theta_E,\theta_M} = \|\frac{1}{b_i}\sum_{j=1}^{b_i}\phi(\boldsymbol{h}_{s_j}) - \frac{1}{b_i}\sum_{j=1}^{b_i}\phi(\boldsymbol{h}_{t_j})\|_{\mathcal{H}}^2, \quad (9)$$

where b_i is the batch size, ϕ is the kernel function, $\mathcal{H}$ is the reproducing kernel Hilbert space (RKHS), $\boldsymbol{h}_{s_j}$ is the

---

Algorithm 1 The optimization strategy of our D^{2}AM

1: **Input:** Mixture domain dataset $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$
2: Initialize model parameters $\theta_E, \theta_M, \theta_D$, and determine K by pre-clustering with ResNet
3: **while** not end of epoch **do**
4:   **if** epoch==1
5:     Calculate $\{df(x_i)\}_{i=1}^{n_s}$ using pre-trained ResNet
6:   **else**
7:     Calculate $\{df(x_i)\}_{i=1}^{n_s}$ using feature extractor E
8:   Obtain $\{d_i\}_{i=1}^{n_s}$ by clustering $\{df(x_i)\}_{i=1}^{n_s}$
9:   **while** not end of minibatch **do**
10:     Randomly use $K-1$ domains as the meta-train and the remaining one domain as the meta-test
11:     Meta-train: Sample the batches $B_i(i = 1, ..., K-1)$ in each meta-train domain
12:     **for** each batch B_i **do**
13:       Calculate $L_{Cls(B_i)}(\theta_E,\theta_M)$, $L_{Dep(B_i)}(\theta_E,\theta_D)$, $L_{MMD(B_i)}(\theta_E,\theta_M)$ and $L_{P(B_i)}(\theta_E)$ as Eq. 8, 9, 10, 7, respectively
14:       Inner update $\theta_{M_i'}$ with $L_{Cls(B_i)}(\theta_E,\theta_M)$, $L_{MMD(B_i)}(\theta_E,\theta_M)$
15:     Meta-test: Sample the meta-test batches B_t
16:     Use $\theta_{M_i',E,D}$ to calculate $L_{Dep(B_t)}(\theta_E,\theta_D)$, $\Sigma_{i=1}^{K-1}(L_{Cls(B_t)}(\theta_E,\theta_{M_i'}) + \lambda_m L_{MMD(B_t)}(\theta_E,\theta_{M_i'}))$
17:     Meta-optimization with Eq. 11, 12, 13
18: **Return:** Model parameters $\theta_E, \theta_M, \theta_D$

---

sample feature output by adaptation layer, and $\boldsymbol{h}_{t_j}$ is the feature of the same dimension as $\boldsymbol{h}_{s_j}$, which randomly generated from prior distribution. In each meta-train domain, the inner-update of meta leaner's parameters can be calculated as $\theta_{M'} = \theta_M - \alpha\nabla_{\theta_M}(L_{Cls(B_i)}(\theta_E,\theta_M) + \lambda_m L_{MMD(B_i)}(\theta_E,\theta_M))$, λ_m is the hyper-parameter. Meanwhile, we incorporate face depth maps as auxiliary information to guide learning of extractor, which can be formed as:

$$L_{Dep(B_i)}_{\theta_E,\theta_D} = \sum_{(x,y)\in B_i} \|D(E(x)) - I\|^2, \quad (10)$$

where θ_D is the parameter of depth estimator and I is the face depth map of face image, which estimated by PRNet (Feng et al. 2018) for real face and set zeros for fake face.

Meta-Test We sample batch B_t in the one remaining meta-test domain D^t. We encourage our face anti-spoofing model trained in each meta-train domain can simultaneously perform well on the unseen cross-domain meta-test domain. Hence, we calculate $\Sigma_{i=1}^{K-1}(L_{Cls(B_t)}(\theta_E,\theta_{M'}) + \lambda_m L_{MMD(B_t)}(\theta_E,\theta_{M'}))$ with inner-updated meta learners. Also, $L_{Dep(B_t)}(\theta_E,\theta_D)$ is incorporated like meta-train.

Meta-Optimization We jointly train the three modules in our network in a meta-learning framework as follows:

$$\theta_M \leftarrow \theta_M - \beta\nabla_{\theta_M}(\sum_{i=1}^{K-1}(L_{Cls(B_i)}_{\theta_E,\theta_M} + \lambda_m L_{MMD(B_i)}_{\theta_E,\theta_M} + L_{Cls(B_t)}_{\theta_E,\theta_{M_i'}} + \lambda_m L_{MMD(B_t)}_{\theta_E,\theta_{M_i'}})),$$
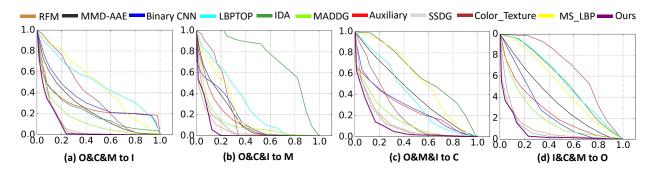(11)

Figure 3: ROC curves of four testing sets for domain generalization on face anti-spoofing.

Method	O&C&M to I		O&C&I to M		O&M&I to C		I&C&M to O	
	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)	HTER(%)	AUC(%)
MS_LBP	50.30	51.64	29.76	78.50	54.28	44.98	50.29	49.31
Binary CNN	34.47	65.88	29.25	82.87	34.88	71.94	29.61	77.54
IDA	28.35	78.25	66.67	27.86	55.17	39.05	54.20	44.59
Color Texture	40.40	62.78	28.09	78.47	30.58	76.89	63.59	32.71
LBPTOP	49.45	49.54	36.90	70.80	42.60	61.05	53.15	44.09
Auxiliary (Depth)	29.14	71.69	22.72	85.88	33.52	73.15	30.17	77.61
Auxiliary (All)	27.6	-	-	-	28.4	-	-	-
MMD-AAE	31.58	75.18	27.08	83.19	44.59	58.29	40.98	63.08
MADDG	22.19	84.99	17.69	88.06	24.5	84.51	27.98	80.02
SSDG-M	18.21	90.61	16.67	90.47	23.11	85.45	25.17	81.83
RFM	17.30	90.48	13.89	93.98	**20.27**	**88.16**	16.45	**91.16**
D²AM	**15.43**	**91.22**	**12.70**	**95.66**	20.98	85.58	**15.27**	90.87

Table 1: Comparison to face anti-spoofing methods on four testing sets for domain generalization on face anti-spoofing.

$$\theta_E \leftarrow \theta_E - \beta \nabla_{\theta_E} (L_{Dep(B_t)}_{\theta_E, \theta_D} + \sum_{i=1}^{K-1} (L_{Cls(B_i)}_{\theta_E, \theta_M} + \lambda_m L_{MMD(B_i)}_{\theta_E, \theta_M}$$

$$+ L_{Cls(B_t)}_{\theta_E, \theta_{M'_i}} + \lambda_m L_{MMD(B_t)}_{\theta_E, \theta_{M'_i}} + \sum_{j=1}^{3} \lambda_p L_{P(B_i)}^{j}_{\theta_E} + L_{Dep(B_i)}_{\theta_E, \theta_D})),$$

(12)

$$\theta_D \leftarrow \theta_D - \beta \nabla_{\theta_D} (L_{Dep(B_t)}_{\theta_E, \theta_D} + \sum_{i=1}^{K-1} (L_{Dep(B_i)}_{\theta_E, \theta_D})), \quad (13)$$

where $L_{P(B_i)}^{j}$ denotes the domain enhancement entropy loss for the jth DRLM in feature extractor E. Since we iteratively re-assign the domain label with the largest domain shift to the sample by clustering with discriminative domain representation in each epoch, this can simulate more abundant and difficult cross-domain scenarios and make the model focus on better-generalized learning directions. The detailed training process is shown in Algorithm 1.

Experiments

Datasets Four public face anti-spoofing datasets are utilized to evaluate the effectiveness of our method: OULU-NPU (Boulkenafet et al. 2017) (denoted as O), CASIA-FASD (Zhang et al. 2012) (denoted as C), Idiap Replay-Attack (Chingovska, Anjos, and Marcel 2012) (denoted as I), and MSU-MFSD (Wen, Han, and Jain 2015) (denoted as M). We randomly select three datasets as a mixture source domain for training, and the remaining one is the unseen domain for testing. Unlike existing methods that assume each dataset represents a domain, the source domain we select is a mixture of multiple latent domains without domain labels.

Implementation Details Our method is implemented via PyTorch and trained with Adam optimizer. We extract the RGB and HSV channels of each input image, thus, the input size is $256 \times 256 \times 6$. The learning rates α, β are set as 1e-3, 1e-4, respectively, and the prior distribution for MMD is defined as the standard normal distribution. For other hyper-parameters, we set λ_p as 0.1 and λ_m as 0.05. In our method, K determines the number of subdomains that the model needs to be divided. We found that converting the convolutional features extracted by pre-trained ResNet into domain features for clustering can clearly divide the sample into several clusters, so we can determine the value of K as 3. We use the Half Total Error Rate (HTER) and the Area Under Curve (AUC) as the evaluation metrics.

Result and Discussion

As shown in Figure 3, Tables 1 and 2, our method outperforms all state-of-the-art methods under most of tasks. We make the following observations from the results. (1) DG-based face anti-spoofing methods perform better than

(a) RFM   (b) D^{2}AM   (c) F^+ (d) F^-
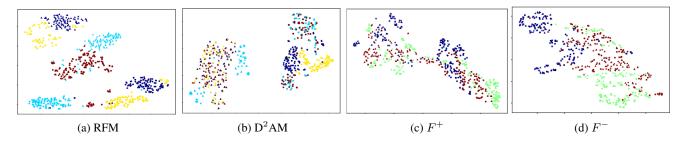
Figure 4: The t-SNE visualization of the O&C&M to I task. Each color represents a domain (blue points in (a), (b) represent the unseen target domain), square points and triangle points represent live faces and fake faces, respectively. Note that, for better comparison, the domain label in the visualization is consistent with RFM, not the pseudo domain label assigned by D^{2}AM.

Method	M&I to C		M&I to O	
	HTER(%)	AUC(%)	HTER(%)	AUC(%)
MS_LBP	51.16	52.09	43.63	58.07
IDA	45.16	58.80	54.52	42.17
CT	55.17	46.89	53.31	45.16
LBPTOP	45.27	54.88	47.26	50.21
MADDG	41.02	64.33	39.35	65.10
SSDG-M	**31.89**	71.29	36.01	66.88
RFM	36.34	67.52	29.12	72.61
D^{2}AM	32.65	**72.04**	**27.70**	**75.36**

Table 2: Comparison to methods with limited domains.

conventional methods. This proves that the distribution of target domain is different from source domain, while conventional methods only focus on the differentiation cues that only fit source domain. (2) The proposed method outperforms other DG-based methods, including RFM utilizing domain labels. We believe the reason is that our method can iteratively cluster mixture domains to find subdomains with the largest distribution difference, which allows the model to find better optimization directions with more abundant and difficult domain shift scenarios. Besides, we jump out of the box and evaluate it on single domain (protocol 3 of Oulu-NPU). The average results are: D^{2}AM (HTER=0.023±0.014, AUC=0.976±0.018) and RFM (HTER=0.031±0.016, AUC=0.947±0.025), which indicate that D^{2}AM has the strong ability to capture domain information.

Method	HTER(%)	AUC(%)
D^{2}AM w/ d	18.24	89.28
D^{2}AM w/o d	27.05	73.57
D^{2}AM w/o select	16.57	89.98
D^{2}AM w/o L_p	15.89	90.81
D^{2}AM w/o L_{MMD}	16.11	90.24
D^{2}AM(K=2)	17.16	90.03
D^{2}AM(K=3)	**15.43**	**91.22**
D^{2}AM(K=4)	16.82	90.57

Table 3: Evaluations of different components of the proposed method on O&C&M to I task.
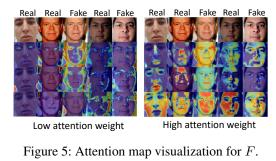
Ablation Study We perform ablation study to verify the efficacy of each component. Several observations can be made from Table 3. (1) D^{2}AM w/o d means that D^{2}AM randomly selects meta-train and meta-test from mixture domains, and its performance is worse than D^{2}AM w/ d, which utilizes domain label as existing methods. This is because there is no domain shift between the randomly selected domains. Hence, we think it is important to simulate difficult and abundant domain shift scenarios for meta-learning. (2) D^{2}AM w/o select means that features of all channels are used instead of only the channel features with low attention weight for clustering, which performs worse than D^{2}AM. These results indicate that features with low attention weights contain more domain information to achieve better clustering. (3) We conduct a sensitivity analysis for K, and the results show that determining K based on the number of clusters divided by pre-trained ResNet can get the best results. (4) D^{2}AM yields the best performance, confirming that each component contributes to the final results.

Visualization and Analysis

Adaptation Feature Visualization We visualize the features of adaptation layer using t-SNE (Donahue et al. 2014). As shown in Figures 4a and 4b, several observations can be made. (1) From the perspective of domain information, comparing Figure 4a and 4b, we can find that D^{2}AM is more powerful to extract domain invariant features, which proves that our method is more robust to unseen samples. (2) From the perspective of task discriminative information, D^{2}AM makes features more dispersed in the feature space compared to RFM. Therefore, a better class boundary can be achieved by D^{2}AM. (3) There are no outliers in the feature space of D^{2}AM, which shows that MMD-based regularization can effectively constrain outliers to dense sample areas.

Convolutional Feature Visualization In Figures 4c and 4d, we visualize the distribution of the features from the 3rd DRLM via t-SNE. It can be seen that $df(x)$ with F^+ contains more task discriminative information, while $df(x)$ with F^- contains more domain information. This result validates that the domain enhancement entropy loss L_p can encourage model to extract domain-discriminative features.

Attention Map Visualization To verify that channels with low attention weight in F contain more domain infor-

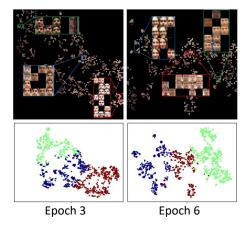Figure 5: Attention map visualization for F.

Figure 6: The t-SNE cluster visualization on O&C&M to I. Each color represents a domain divided by our method.

mation, we visualize attention maps of 3rd DRLM by the Global Average Pooling (GAP) method (Zhou et al. 2016). As shown in Figure 5, we find that attention maps with low attention weight focus on specific attack differentiation cue, backgrounds, overall styles, *etc*, which are not generalized because they will be changed if data comes from a new scene. For example, in the second row of low attention weight, there is a type of attack that mimics eye blinking through two pieces of paper, so the type can be judged by locating a specific difference in the eye region, but other attacks do not have this difference, and in the third row of low attention weight, the background is focused. While attention map with high attention weight always focuses on the region of the internal face, which are more likely to be intrinsic and generalized. Therefore, better domain dividing can be achieved by selecting features with low attention weight.

Cluster Visualization To provide more insights on why our iterative clustering can perform better than other with known domain labels, we randomly sample 10,000 samples and cluster them according to $df(x)$ with F_d^-. The results are shown in Figure 6. We can find that D^2AM can dynamically adjust the basis of division, so as to simulate richer domain shifts for better meta-learning. For example, Epoch 3 is mainly clustered by illumination, while the blue points in the Epoch 6 contain samples of different illumination, so Epoch 6 is mainly clustered by background. The reason why our method can dynamically adjust is that meta-learning learns
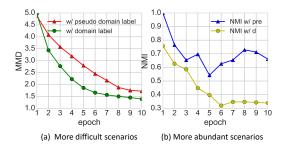

Figure 7: On O&C&M to I, (a) The MMD calculated with actual domains or pseudo domains. (b) NMI between pseudo domain and previous assignments or actual domain.

from the scene of current epoch to improve the robustness of this kind of shift, so in the next epoch, the model will extract domain invariant features for this scene, so as to guide clustering to focus on other domain information that is not yet robust and construct new scenarios.

Cluster Analysis To verify the effectiveness of our iterative clustering, we adopt MMD to calculate the difference between domains, and Normalized Mutual Information (NMI) [1] to measure the changes of pseudo-domain labels to evaluate the difficulty and diversity of domain shift scenarios, respectively. As shown in Figure 7a, compared with domain label which indicates the dataset each sample comes from, the subdomain obtained by pseudo domain label given by D^2AM has a larger domain difference, which shows that our method can simulate more difficult scenarios for better optimization direction. As shown in Figure 7b, we found that NMI between pseudo domain labels and previous assignments is almost between 0.6 and 0.8, indicating that the pseudo-domain label of the sample is constantly changing, especially in the middle stage of training, which indicates that D^2AM can generate richer domain shift scenarios to further improve the generalization. To measure the difference between the pseudo-domain and actual domain, we calculated the NMI between them and found that the pseudo-domain label and domain label only overlap about 64.9%±5.7%, which indicates that the pseudo-domain label is different from the actual domain label.

Conclusion

In this paper, to the best of our knowledge, this is the first wok to address mixture domain face anti-spoofing, where the domain label is unknown. Specifically, we design the D^2AM, which iteratively clusters mixture domains via discriminative domain representation and trains a generalizable feature extractor by meta-learning. A DRLM and MMD-based regularization are designed for better dynamic adjustment to simulate more difficult and abundant domain shift scenes. Comprehensive experiments show that D^2AM outperforms conventional DG-based face anti-spoofing methods, including those utilizing domain labels. Furthermore, we enhance the interpretability through visualization.

---

[1] The smaller the NMI, the greater the difference betwee them.

References

Boulkenafet, Z.; Komulainen, J.; and Hadid, A. 2016. Face spoofing detection using colour texture analysis. *IEEE Transactions on Information Forensics and Security* .

Boulkenafet, Z.; Komulainen, J.; Li, L.; Feng, X.; and Hadid, A. 2017. Oulu-npu: A mobile face presentation attack database with real-world variations. In *FG*, 612–618.

Chen, Z.; Chen, C.; Cheng, Z.; Jiang, B.; Fang, K.; and Jin, X. 2020. Selective transfer with reinforced transfer network for partial domain adaptation. In *CVPR*, 12706–12714.

Chingovska, I.; Anjos, A.; and Marcel, S. 2012. On the effectiveness of local binary patterns in face anti-spoofing. In *BIOSIG*, 1–7.

de Freitas Pereira, T.; Komulainen, J.; Anjos, A.; De Martino, J. M.; Hadid, A.; Pietikäinen, M.; and Marcel, S. 2014. Face liveness detection using dynamic texture. *EURASIP Journal on Image and Video Processing* 2014(1): 2.

Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; and Darrell, T. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, 647–655.

Feng, Y.; Wu, F.; Shao, X.; Wang, Y.; and Zhou, X. 2018. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV*, 534–551.

Ghifary, M.; Bastiaan Kleijn, W.; Zhang, M.; and Balduzzi, D. 2015. Domain generalization for object recognition with multi-task autoencoders. In *ICCV*, 2551–2559.

Gragnaniello, D.; Poggi, G.; Sansone, C.; and Verdoliva, L. 2015. An investigation of local descriptors for biometric spoofing detection. *IEEE transactions on information forensics and security* 10(4): 849–863.

Grandvalet, Y.; and Bengio, Y. 2005. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, 529–536.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 1501–1510.

Jia, Y.; Zhang, J.; Shan, S.; and Chen, X. 2020. Single-Side Domain Generalization for Face Anti-Spoofing. In *CVPR*.

Jourabloo, A.; Liu, Y.; and Liu, X. 2018. Face de-spoofing: Anti-spoofing via noise modeling. In *ECCV*, 290–306.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2018a. Learning to generalize: Meta-learning for domain generalization. In *AAAI*.

Li, H.; Jialin Pan, S.; Wang, S.; and Kot, A. C. 2018b. Domain generalization with adversarial feature learning. In *CVPR*, 5400–5409.

Liu, Y.; Jourabloo, A.; and Liu, X. 2018. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *CVPR*, 389–398.

MacQueen, J.; et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, 281–297.

Matsuura, T.; and Harada, T. 2020. Domain Generalization Using a Mixture of Multiple Latent Domains. In *AAAI*.

Munkres, J. 1957. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics* 5(1): 32–38.

Patel, K.; Han, H.; and Jain, A. K. 2016. Secure face unlock: Spoof detection on smartphones. *IEEE transactions on information forensics and security* 11(10): 2268–2283.

Qin, Y.; Zhao, C.; Zhu, X.; Wang, Z.; Yu, Z.; Fu, T.; Zhou, F.; Shi, J.; and Lei, Z. 2019. Learning meta model for zero-and few-shot face anti-spoofing. *arXiv preprint arXiv:1904.12490* .

Saha, S.; Xu, W.; Kanakis, M.; Georgoulis, S.; Chen, Y.; Pani Paudel, D.; and Van Gool, L. 2020. Domain Agnostic Feature Learning for Image and Video Based Face Anti-spoofing. In *CVPR*.

Shao, R.; Lan, X.; Li, J.; and Yuen, P. C. 2019. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *CVPR*, 10023–10031.

Shao, R.; Lan, X.; and Yuen, P. C. 2017. Deep convolutional dynamic texture learning with adaptive channel-discriminability for 3D mask face anti-spoofing. In *IJCB*.

Shao, R.; Lan, X.; and Yuen, P. C. 2020. Regularized Fine-Grained Meta Face Anti-Spoofing. In *AAAI*, 11974–11981.

Tan, X.; Li, Y.; Liu, J.; and Jiang, L. 2010. Face liveness detection from a single image with sparse low rank bilinear discriminative model. In *ECCV*, 504–517.

Wang, G.; Han, H.; Shan, S.; and Chen, X. 2020. Cross-domain Face Presentation Attack Detection via Multi-domain Disentangled Representation Learning. In *CVPR*.

Wen, D.; Han, H.; and Jain, A. K. 2015. Face spoof detection with image distortion analysis. *IEEE Transactions on Information Forensics and Security* 10(4): 746–761.

Yang, J.; Lei, Z.; and Li, S. Z. 2014. Learn convolutional neural network for face anti-spoofing. *arXiv preprint arXiv:1408.5601* .

Zhang, K.-Y.; Yao, T.; Zhang, J.; Tai, Y.; Ding, S.; Li, J.; Huang, F.; Song, H.; and Ma, L. 2020. Face Anti-Spoofing via Disentangled Representation Learning. *arXiv preprint arXiv:2008.08250* .

Zhang, Z.; Yan, J.; Liu, S.; Lei, Z.; Yi, D.; and Li, S. Z. 2012. A face antispoofing database with diverse attacks. In *ICB*.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *CVPR*, 2921–2929.

Zhou, K.; Yang, Y.; Cavallaro, A.; and Xiang, T. 2019. Omni-scale feature learning for person re-identification. In *ICCV*, 3702–3712.