

Face Presentation Attack Detection by Excavating Causal Clues and Adapting Embedding Statistics

Meiling Fang¹, Naser Damer^{1,2}

¹Fraunhofer Institute for Computer Graphics Research IGD, Darmstadt, Germany

²Department of Computer Science, TU Darmstadt, Darmstadt, Germany

Email: meiling.fang@igd.fraunhofer.de

Abstract

Recent face presentation attack detection (PAD) leverages domain adaptation (DA) and domain generalization (DG) techniques to address performance degradation on unknown domains. However, DA-based PAD methods require access to unlabeled target data, while most DG-based PAD solutions rely on a priori, i.e., known domain labels. Moreover, most DA-/DG-based methods are computationally intensive, demanding complex model architectures and/or multi-stage training processes. This paper proposes to model face PAD as a compound DG task from a causal perspective, linking it to model optimization. We excavate the causal factors hidden in the high-level representation via counterfactual intervention. Moreover, we introduce a class-guided MixStyle to enrich feature-level data distribution within classes instead of focusing on domain information. Both class-guided MixStyle and counterfactual intervention components introduce no extra trainable parameters and negligible computational resources. Extensive cross-dataset and analytic experiments demonstrate the effectiveness and efficiency of our method compared to state-of-the-art PADs. The implementation and the trained weights are publicly available¹.

1. Introduction

Face recognition [2, 3, 9] has become a remarkable technique for identity authentication and has been widely deployed in our daily lives. However, face recognition is vulnerable to presentation attacks (PAs) [1, 13, 55, 56], such as printed images, replayed videos, and 3D masks. Therefore, face presentation attack detection (PAD) plays an important role in securing face recognition from such PAs. Despite the promising performance in intra-dataset evaluation by deep learning-based face PAD techniques [10, 51, 52], the PAD performance significantly degrades under the more realistic cross-domain scenario.

Recently, a variety of domain adaptation (DA) and do-

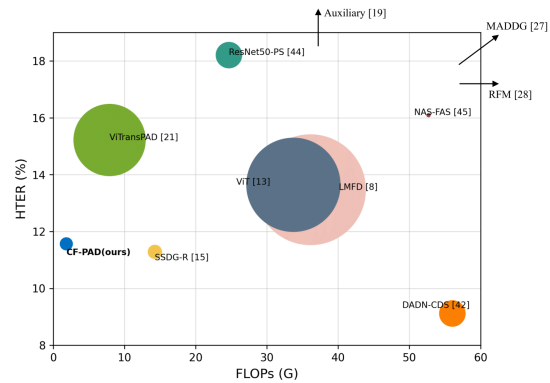


Figure 1. PAD performance vs. Computational complexity. x-axis indicates the complexity of the model by FLOPs, y-axis indicates the PAD performance represented by HTER (%), and the radius of the circle indicates the number of trainable parameters. An ideal PAD tends to be located in the bottom left corner with a small circle, as achieved by our CF-PAD solution (in blue).

main generalization (DG) based PAD methods [22, 34, 35, 44, 49] are proposed to boost the PAD generalizability. DA-based PAD methods [23, 41] learn a discriminative feature space by accessing the labeled source domains and unlabeled target domains. However, this target data is typically unavailable in real-world scenarios. In contrast with DA-based methods leveraging the target testing data, DG-based PAD methods [5, 22, 34, 35] aim to generalize the knowledge obtained from multiple source datasets to the unseen target domain and has achieved remarkable progress. However, these efforts only attempt to alleviate the problems caused by different data distributions by modelling the statistical dependencies between inputs and outputs. They do not excavate the intrinsic causal mechanisms. In addition, DG-based PAD methods rely on a priori knowledge that the domain label of each source dataset is known. This is challenging to be satisfied in practical situations and it is hard to define the representation of a *domain* well enough. Moreover, most existing DG-based face PAD methods are computationally intensive due to complex model architec-

¹<https://github.com/meilingfang/CF-PAD>

tures [20, 49] or multi-stage/multi-network training [5, 25].

To address the above issues, this work presents a structural causal model (SCM) to formulate the face PAD problem, aiming to uncover the causality hidden in the high-level representative features (as shown in Fig. 2) and enhance the PAD generalizability, without requiring complex architectures. The contributions of our work are as follows: 1) We model the compound DG-based PAD from a causality-based view linking it to the model optimization. Our method relaxes the need for a priori domain knowledge compared to most DA/DG based PADs, which rely on domain labels or target data. 2) we propose to boost the PAD generalizability by enriching the feature-level training data distribution via our proposed class-guided MixStyle and separating the causal factors from high-level representation via counterfactual interventions. Both components require no extra trainable parameters and very little computation complexity, as illustrated in Fig. 1. 3) We conduct extensive experiments under various cross-dataset scenarios and the analytical results demonstrate the effectiveness and efficiency of our method in comparison to existing works.

2. Related Work

Face Presentation Attack Detection Many face PAD solutions [7, 11, 12, 26, 51, 52] have been proposed and shown good performance with the recent advances of deep-learning techniques, especially in intra-dataset evaluations. Recently, a number of PAD works [22, 34, 35, 44, 49] turned to leverage DA and DG techniques to target the challenging cross-domain scenario. DA transfers the knowledge from the source domain to the target domain, where the unlabeled target data is accessed in the training process. DA-based face PAD methods align the feature space between source and target domains by minimizing the maximum mean discrepancy (MMD) [23] and adversarial training [41]. However, collecting unlabeled target domain data is very difficult and laborious. Moreover, in real-world scenarios, there is usually no information available for the target domain during training. In comparison with DA-based methods, DG-based PAD methods are more practical for real-world deployment because they do not require the acquisition of target data. DG-based PAD methods learn a model from multiple datasets to obtain a generalized feature representation that allows generalizing well to the unseen target domain. There are various types of DG-based PAD methods: 1) adversarial training [22, 34], 2) meta-learning [5, 35], simulating the domain shift by dividing source data into meta-train and meta-test sets. Most existing DA- or DG-based PAD solutions are computationally intensive because of the use of large model architectures such as ViT [20], DADN-CDS [49] and MADDG [34] consisting of multiple feature extractors and discriminators. Different from the above-mentioned face PAD methods, we approach the face

PAD from a causal perspective and focus on learning the causal representations by analyzing the contribution of features to PAD decisions. Our method is an efficient model that boosts the generalizability of a simple single network for face PAD by incorporating two components. These two components require no extra parameters during training and thus eliminating the need for multiple networks and/or computationally costly components (e.g., attention blocks). In addition, these two components are removed during the inference.

Causal Inference Causal inference [16, 30], analyzing the correlation between the variables and the final results, has gained increasing attention in recent years. The causal inference has been successfully applied in many computer vision tasks to enhance the generalizability of models. For example, Rao *et al.* [32] proposed to learn the attention with counterfactual causality by measuring the attention quality and encouraging the model to generate high-quality attention for fine-grained image recognition. Chen *et al.* [4] proposed a counterfactual analysis approach for human trajectory prediction. They investigated the causality between the predicted trajectories and inputs and mitigated the negative effect brought by the environment bias.

A recent work, Causal Intervention for Face Anti-spoofing (CIFAS) method [25], was proposed to target the generalizability of face PAD. CIFAS [25], as DG PAD solution, was facilitated based on a priori that domain labels are known and assumed that data from various domains affecting the PAD performance. Therefore, the PAD performance was improved by the preconceived domain-specific features in CIFAS, i.e., the pre-learned domain features are used for backdoor adjustment. In contrast, our goal is to automatically discover the intrinsic causality from the learned high-level features without any priori knowledge, i.e. as a compound DG PAD task. Moreover, CIFAS [25] is a three-stage face PAD framework including two networks and an additional domain feature extraction process. Compared to the computationally intensive CIFAS, our proposed solution is an end-to-end model containing two components without additional trainable parameters other than a lightweight feature extractor and a classifier.

3. Methodology

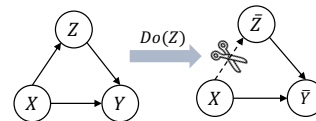


Figure 2. SCM with variables X (face input), Y (prediction), and Z (high-level feature), representing the PAD task $(X, Z) \rightarrow Y$. Performing CI on feature Z , cutting the link between $X \rightarrow Z$, can force the Z independent of its causal parent X . The detailed CI is presented in Sec. 3.3.

3.1. PAD in the view of contextual reasoning

We provide a background on causal inference [16, 30] and model optimization [39] that will form the basis for understanding the rest of this paper.

We first introduce the SCM \mathcal{M} to help formulate the underlying causal structure of the PAD task. A convention SCM consists of a set of causal variables and causal links. As shown in Fig. 2, X is the input (i.e. face images), Y is the final prediction (i.e. bona fide or attack), and Z is the intermediate high-level features. The PAD task here is to predict Y from X , relying on the high-level features Z hidden in the inputs, presented as $(X, Z) \rightarrow Y$. The PAD models are commonly optimized by supervising the final prediction Y , overlooking the effect of Z on Y . Causal inference [16], the analysis of causal correlation between these variables, facilitates thinking outside the box. In practice, we can analyze the causalities by manipulating the values of variables and observing the caused effect. This manipulation is termed *intervention* in causal inference [16] and represented by an idiomatic symbol $do(\cdot)$.

Next, we link the intervention with the model optimization by recalling the conventional empirical risk minimization (ERM) [39]. Assume a model h , accessing training data from a distribution $p(X, Y) \in \mathcal{P}$, is trained to minimize the empirical risk \hat{R} :

$$\hat{R}_{p(X,Y)}(h) = \hat{\mathbb{E}}_{p(X,Y)}[\mathcal{L}(Y, h(X))] \quad (1)$$

where \mathcal{L} refers to loss function. The objective of intervention risk minimization associating to an intervention I is:

$$\hat{R}_{p(X,Y|I)}^{\mathcal{M}}(h) = \hat{\mathbb{E}}_{p(X,Y|I)}^{\mathcal{M}}[\mathcal{L}(Y, h(X))] \quad (2)$$

where $p(X, Y|I)$ indicates the intervened data distribution. When a set of intervention operations $I \in \mathcal{I}$ are performed over the SCM, the worst-case intervened data distribution is:

$$\hat{R}_{p(X,Y|I)}^{\mathcal{M}}(h) = \max_{I \in \mathcal{I}} \hat{\mathbb{E}}_{p(X,Y|I)}^{\mathcal{M}}[\mathcal{L}(Y, h(X))] \quad (3)$$

Finally, a generalized model can be obtained by solving

$$h^* = \arg \min_h \max_{I \in \mathcal{I}} \hat{\mathbb{E}}_{p(X,Y|I)}^{\mathcal{M}}[\mathcal{L}(Y, h(X))] \quad (4)$$

As the min-max optimization problem is very challenging, Equation 4 can be approximated by 1) enriching the diversity of the training data distribution and 2) increasing the diversity of interventions [54]. This theoretical basis motivates us to develop a generalized PAD by solving this approximation problem from both perspectives. The pipeline of the proposed CF-PAD solution is shown in Fig. 3, containing the two components we proposed in this work, class-guided MixStyle (CGMixStyle) and Counterfactual Intervention (CI). CGMixStyle aims to enrich the diversity of training data (on the feature-level), while CI mines causal

related features by performing a set of counterfactual interventions. The details are presented in Sec. 3.2 and 3.3, respectively.

3.2. Class-guided MixStyle

Due to the privacy concerns and the labor-intensive nature of data collection, most existing face PAD datasets are limited in identity, capture environment, and attack type. Therefore, increasing the training data diversity by incorporating more training data is the less realistic option. Inspired by the recently introduced MixStyle [57], we propose a class-guided MixStyle to increase the distribution of the available limited training data by mixing the feature statistics within the same class. The MixStyle [57] is formulated as the following:

$$\begin{aligned} \gamma &= \lambda\sigma(x) + (1 - \lambda)\sigma(\tilde{x}) \\ \beta &= \lambda\mu(x) + (1 - \lambda)\mu(\tilde{x}) \end{aligned} \quad (5)$$

Where $x \in \mathbb{R}^{B \times C \times H \times W}$ refers to the input batch feature and \tilde{x} indicates the corresponding reference batch from x . $\sigma(\cdot), \mu(\cdot) \in \mathbb{R}^{B \times C}$ are mean and standard deviation computed across the spatial dimension within each channel of each sample. $\lambda \in \mathbb{R}^B$ are weights sampled from the Beta distribution. The final mixed feature statistic is applied to the styled normalized x as:

$$MixStyle(x) = \gamma \frac{x - \mu(x)}{\sigma(x)} + \beta \quad (6)$$

In MixStyle [57], two shuffle operations are considered to generate \tilde{x} from x , domain label guided and random shuffle, as shown in Fig. 4 (a) and (b). After shuffling, data can be produced in new stylized domains by following the equations 5 and 6. Meanwhile, the discriminative features, such as outline and shape, for general computer vision classification task, is unchanged. However, face PAD relies on the subtle intrinsic attack clues that are assumed to be part of the image style. Therefore, both shuffle operations pose a potential risk of degradation in PAD performance, as the style transfer may be performed between bona fide and attack samples. Consequently, we propose to modify the MixStyle concept by restricting the shuffle operation within the same class, as illustrated in Fig. 4 (c). Specifically, in Eq. 5, each sample in \tilde{x} , generated by class-guided shuffle, belongs to the same class as each sample in x . This "class-guided" MixStyle is designed to produce more diverse features without a confusion of bona fide and attack features, and thus resulting in a better generalized representations. The effectiveness of class-guided shuffling is demonstrated in later in Tab. 1.

3.3. Counterfactual Intervention

In an optimal situation, a well-generalized PAD should be independent of some factors, such as sensor type, environment, and identity, as represented by the dashed rectangular box in Fig. 3. However, PAD solutions suffer from the

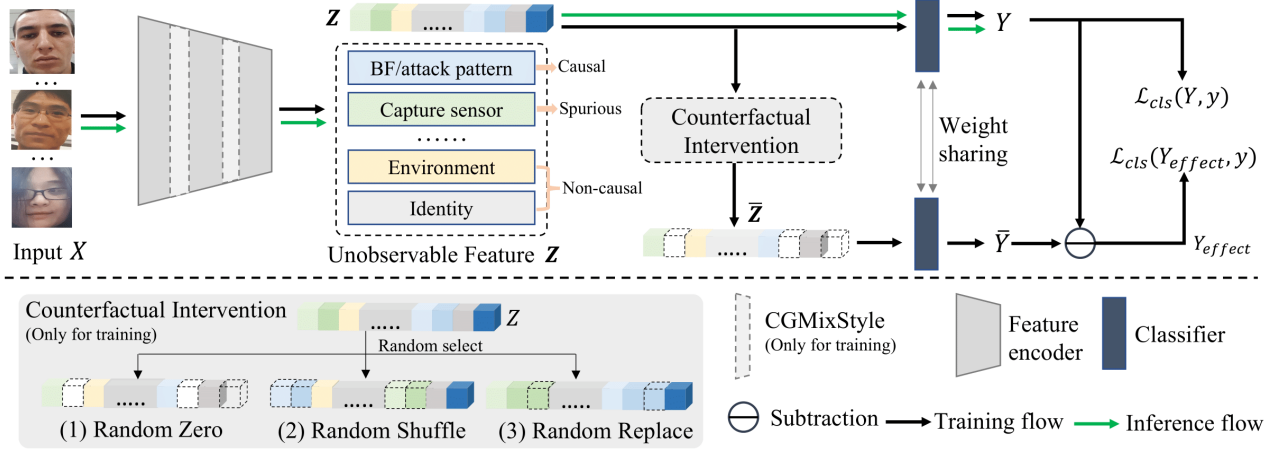


Figure 3. The workflow of the proposed CF-PAD solution. During the training process, CGMixStyle enriches the diversity of feature-level training data distribution and three CIs guides the model to uncover the causal factors that truly contributes to the PAD decision. Both CGMixStyle and CI are removed in the inference phase.

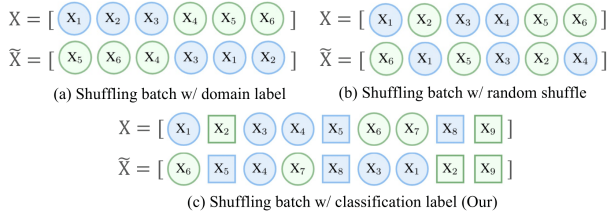


Figure 4. The comparison of shuffling operation between MixStyle (a,b) and our CGMixStyle (c). Color and shape refer to domain and class, respectively. Class-guided shuffle restrict the shuffling within the same class (bona fide or attack in PAD) without the prior domain information.

low generalization on unknown data [14, 31, 48, 55]. Therefore, we build an assumption regarding the high-level feature Z that each feature, representing for input images X , is constructed from a mix of different factors: 1) The class-dependent information encoded in Z is causal factors, associated with a universal and invariant patterns of bona fide or attack. 2) The information independent of class in Z is considered as non-causal factors, such as identity information and capture environments. 3) Capture sensors is considered to be the spurious factor as several existing cross-domain PADs [25, 49, 55] stated that the PAD performance is affected by the capture devices.

Our goal is to identify the factors in Z that truly contribute to the class Y . As mentioned in Sec. 3.1 and illustrated in Fig. 2, we can perform intervention on the learned feature Z to measure its contribution and encourage model to produce more generalized features. Specifically, We propose to adopt *counterfactual intervention* [19, 45] to analyse the effect of high-level features. Counterfactual intervention ($do(Z = \bar{Z})$) refers to generate a counterfactual sample where the causal factors are removed. However, generating (true) counterfactual features is impractical, because

the causal factors in high-level feature is unobservable and unformulated.

In our work, we present three simple interventions, random zero, random shuffle, and random replace, to encourage the model to uncover the causal factors from the confounding feature Z . This uncovering is facilitated by comparing the effect of the original features Z and counterfactual features \bar{Z} , as shown in Fig. 3. The causal effect is then formulated based on the difference of the original prediction and its counterfactual alternative prediction:

$$Y_{effect} = Y(Z) - Y(\bar{Z}) \quad (7)$$

where $Y(\bar{Z}) = f(do(Z = \bar{Z}), X = X)$ and $f(\cdot)$ is the final classifier, catching the changes in final decision caused by the interventions. The loss function is formulated as:

$$\mathcal{L} = \mathcal{L}(Y, y) + \lambda \mathcal{L}(Y_{effect}, y) \quad (8)$$

where y is the ground-truth label and λ is the loss weight. \mathcal{L} is the cross-entropy loss in our case. The former loss optimizes the model with the actual correct predictions and the latter loss penalizes the *wrong* results based on the counterfactual samples, and thus guiding the model to mine the causal factors.

3.4. Enabling computational efficiency

During the training phase, CGMixStyle and CI both have no extra trainable parameters. In the case of CGMixStyle, the gradients of μ and σ in Eq. 5 are blocked in the computation graph, i.e. no back-propagation. In the case of CI, the introduced extra computational cost is an additional feed-forward path of the counterfactual features to the final classifier, which shares the weight of the original features, i.e., no extra gradient computation. Moreover, during the inference phase, CGMixStyle and CI are both removed, as

the green flow shown in Fig. 3. Therefore, the whole computation cost of CF-PAD is very low (whether for training or inference) and minimalistic when compared to other DA- and DG-based PAD framework which requires the complicated model architectures [20, 34, 49], multi-stage training process [25], or iterative meta-learning steps [5, 35]. The comparison results are discussed later in Sec. 5.3 and illustrated in Fig. 1.

4. Experiments

4.1. Datasets

Following the [12, 24, 25, 29, 34, 35], we conduct experiments on five publicly available face PAD datasets: MSU-MFSD [46] (denoted as M), CASIA-FASD [56] (denoted as C), Idiap Replay-Attack [6] (denoted as I), OULU-NPU [1] (denoted as O), and CelebA-Spoof [55] (denoted as CA).

The MSU-MFSD [46] dataset is comprised of 440 videos across from 35 subjects and contains two types of attacks, printed photo attacks and replay attacks. The CASIA-FASD [56] dataset consists of 600 videos from 50 subjects and includes three types of attacks: warped photo attack, cut photo attack, and video replay attack. The **Idiap Replay-Attack** dataset [6] contains 300 videos from 50 subjects and includes two attack types: print attacks and replay attacks. The **OULU-NPU** [1] is a mobile face PAD dataset, consisting of 5940 videos from 55 subjects using six different mobile phones. The **CelebA-Spoof** [55] is a large-scale face PAD dataset collecting from the web, comprising 625,537 images from 10,177 subjects. The dataset is diverse in terms of subjects, illumination, sensors, and attack types and contains print, replay, 3D mask, and paper cut attacks.

All the experiments are cross-dataset scenarios and can be categorized into two groups based on the number of training datasets: multi-source, limited-source, single-source scenarios. Multi-source scenario contains five cases (training dataset(s) \rightarrow testing dataset): O&C&I \rightarrow M, O&M&I \rightarrow C, O&C&M \rightarrow I, I&C&M \rightarrow O, O&C&M \rightarrow CA, and limited-source scenarios contains M&I \rightarrow C and M&I \rightarrow O cases by following works [5, 22, 25, 42]. The first four cases are widely-used evaluation protocols for cross-domain PAD [12, 28, 51, 52] and thus utilized for the following analytical experiments in Sec. 5.1. Single-source scenario includes 12 cases by all the combinations of selecting one dataset from O, C, I, and M as the training data and testing separately on the other three datasets, by following the protocol in [22, 35, 42, 43].

4.2. Implementation Details

Following [12, 15, 18], we sampled evenly 25 frames per video across the duration of each video in O, C, I, and M datasets. Then, we used MTCNN [53] to detect face for each frame and resized each face to $256 \times 256 \times 3$ pix-

els. In each mini-batch, we sampled the training data to keep a bona fide-attack ratio of 1:1 [36]. Our feature extractor is ResNet-18 initialized by the pre-trained weight on ImageNet [8]. The initial learning of feature extractor and classifier were set to 0.001 and 0.01, respectively. The learning rates were halved at 30 and 45 epochs with a maximum epoch of 60. During the training process, we used the Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9 and weight decay of $5e-4$. The loss weight λ was empirically set to 2 for balancing two losses. In the inference phase, a final PAD decision score of a video is a fused score (mean-rule fusion) of all frames by following [12, 18, 51, 52]. The hyper-parameters of CGMixStyle followed the setting of MixStyle [57] for a fair comparison. The hyper-parameters of CI was investigated in Sec. 5.1.

4.3. Evaluation Metrics

Following existing cross-domain face PAD methods [12, 24, 25, 34, 35], we report the Half Total Error Rate (HTER), which is the mean of Bona fide Presentation Classification Error Rate (BPCER) [21] and Attack Presentation Classification Error Rate (APCER) [21] and Area under the Receiver Operating Characteristic (ROC) Curve (AUC) value for comparison.

5. Results

5.1. Analytical Experiments

Comparison of CGMixStyle and MixStyle: Tab. 1 compares the results of MixStyle with random and cross-domain shuffling operation and our CGMixStyle with class-guided shuffle. The results in Tab. 1 show that the random Mixstyle fails to improve the results in one case. For example, in the case O&M&I \rightarrow C, the HTER value is increased from 14.67% (without MixStyle) to 19.00% (with random MixStyle). Moreover, CGMixStyle outperform the MixStyle using random and cross-domain shuffle operations. The results suggest that the class-guided operation helps to enrich the style of the training data without confusing bona fide and attack patterns.

Effect of different types of CIs: We investigated the effect of using three CIs to produce counterfactual features. The results are presented in Tab. 2, where 'All' refers to randomly select one from three interventions (random zero, random replace, random shuffle) in each training mini-batch. The results indicate that applying a combination of different interventions outperform solutions using single intervention. This is consistent with the theoretical basis in Sec. 3.1 that increasing diversity of counterfactual interventions can enhance the PAD generalizability.

Effect of Hyperparameter in CIs: In addition to demonstrate the effectiveness of the different types of CIs, we conducted a set of experiments to analyze the sensitivity

Method	O&C&I → M		O&M&I → C		O&C&M → I		I&C&M → O		Average	
	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑
w/o MixStyle	15.48	91.50	14.67	93.17	22.70	81.63	16.77	91.03	17.41	89.33
cross-domain	11.67	94.93	14.00	92.20	21.65	83.45	13.61	93.71	15.23	91.07
random	10.71	94.71	19.00	88.41	22.55	83.37	13.64	93.57	16.48	90.02
CGMixStyle (ours)	10.95	95.34	12.56	93.15	20.95	85.88	12.10	95.07	14.14	92.36

Table 1. Comparison results of our CGMixStyle and MixStyle. cross-domain and random shuffle operations are presented by MixStyle [57], while our CGMixStyle leverages the class-guided shuffle. The CGMixStyle obtains the best average performance, demonstrating the importance of the class-guided operation.

Method	O&C&I → M		O&M&I → C		O&C&M → I		I&C&M → O		Average	
	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑
Baseline (w/ CGMixStyle)	10.95	95.34	12.56	93.15	20.95	85.88	12.10	95.07	14.14	92.36
+ Random Zero	9.52	94.00	11.89	94.23	18.30	87.90	12.80	94.15	13.13	92.57
+ Random Replace	8.81	96.12	14.56	93.12	18.15	86.05	12.12	93.89	13.41	92.30
+ Random Shuffle	8.33	97.12	12.67	93.25	19.55	89.50	12.39	94.69	13.24	93.64
All	8.11	96.43	11.78	95.64	16.50	91.50	9.87	95.13	11.57	94.68

Table 2. Effect of different types of CIs. All refers to the random execution of one intervention selected from random zero/replace/shuffle in each training mini-batch. 'All' outperforms other single CIs, proving that increasing the diversity of interventions helps model optimization.

of hyperparameter of CIs. This hyperparameter control the feature destruction degree of CIs, i.e., percentage of random zeroing or replacing. For example, 0.2 in x-axis in Fig. 5 indicates that 20% values in the feature Z (of dimension 512) are manipulated by either setting it to zero or replacing it randomly with the value of the previous neighbour. 0.0 indicates no intervention is performed, representing also as dashed lines. In most cases, performing CI using any hyperparameters achieved better PAD performance than no CI (i.e. solid lines underneath dashed lines of the same colour). Moreover, PAD performance achieves its best (lowest HTER values) mostly at 0.2. In the following experiments, we therefore set this hyperparamters at 0.2.

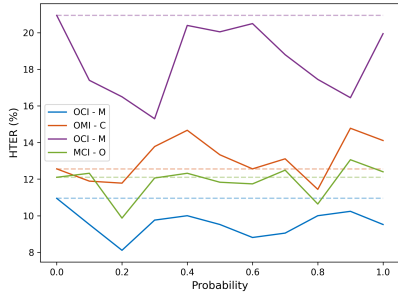


Figure 5. Effect of Hyperparameter in CIs. x-axis indicates the degree of counterfactual intervention in features and y-axis is the HTER (%). Performing CI with any degree enhances the PAD performance in most cases, as solid lines underneath dashed lines of the same colour. The best overall performance is achieved mostly at 0.2.

Effect of CGMixStyle and CIs in CF-PAD: After a detailed analysis of CGMixStyle and CI, we discuss the influence of these two modules in the proposed CF-PAD approach, as the results presented in Tab. 3. We have the following observations: 1) both module enhances the face PAD generalizability, respectively. 2) CF-PAD, combining the CGMixStyle and CI, achieves the best performance. The results point out that two modules are mutually comple-

mentary and strengthening, consisting with the theoretical rationale in Sec. 3.1.

5.2. Comparison with State-of-the-Art methods

The following presents the achieved performances on the considered evaluations in comparison to the SOTA that reported on each setup to capture the widest range of comparisons and the different levels of training data availability.

Multi-source cross-dataset evaluation: Tab. 4 shows results of the following four evaluation protocols: O&C&I → M, O&M&I → C, O&C&M → I, I&C&M → O, by comparing with several SOTA face PAD solutions. From Tab. 4, we observe that: 1) The proposed CF-PAD approach outperforms baseline method (without CGMixStyle and CI) by a drop of 5.84 percentage points on HTER and an enhancement of 5.35 percentage points on AUC. 2) Our proposed CF-PAD method obtains competitive and even better PAD performance than SOTA approaches. Under the challenging case of I&C&M → O, our CF-PAD approach outperforms the SOTA methods with an HTER value of 9.87% and an AUC value of 95.13%.

Furthermore, Tab. 5 shows the results of most challenging case O&C&M → CA, as CA poses a high diversity in terms of the attack types, identity, and capture environments. Our CF-PAD solution achieves the best performance in comparison to the SOTA methods. Both results suggest that our CF-PAD method can identify the causal and generalized discriminative features for cross-dataset scenarios.

Limited-source cross-dataset evaluation: Tab. 6 shows the results of models trained on the limited two training datasets. Our CF-PAD approach outperforms the SOTA face PAD methods, further proving the generalizability of CGMixStyle and CI modules.

Single-source cross-dataset evaluation: To further verify the effectiveness of the proposed method, we evaluate the CF-PAD under the highly challenging scenario where only one dataset is available for training. Tab. 7 and Fig.

Method	O&C&I → M		O&M&I → C		O&C&M → I		I&C&M → O		Average	
	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑
Baseline	15.48	91.50	14.67	93.17	22.70	81.63	16.77	91.03	17.41	89.33
+ CGMixStyle	10.95	95.34	12.56	93.15	20.95	85.88	12.10	95.07	14.14	92.36
+ CI	9.29	94.35	12.44	93.89	18.50	85.61	13.74	90.01	13.49	90.97
CF-PAD (all)	8.11	96.43	11.78	95.64	16.50	91.50	9.87	95.13	11.57	94.68

Table 3. Ablation studies on two components, CGMixStyle and CI, in our CF-PAD solution. The bold number indicates the best performance for each evaluation protocol, demonstrating the importance of both components in CF-PAD.

Method	O&C&I → M		O&M&I → C		O&C&M → I		I&C&M → O		Average	
	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑	HTER(%) ↓	AUC(%) ↑
Binary CNN [48]	29.25	82.87	34.88	71.94	34.47	65.88	29.61	77.54	32.05	74.56
Auxiliary [26]	22.72	85.88	33.52	73.15	29.14	71.69	30.17	77.61	28.89	77.08
ResNet50-PS [51]	14.32	94.51	18.23	89.75	18.86	89.63	21.44	87.56	18.21	90.36
NAS-FAS [52]	19.53	88.63	16.54	90.18	14.51	93.84	13.8	93.43	16.10	91.52
LMFD [12]	10.48	94.55	12.50	94.17	18.49	84.72	12.41	94.95	13.47	92.10
ViTransPAD [28]	8.39	-	21.27	-	16.83	-	15.63	-	15.53	-
PatchNet [40]	7.10	98.64	11.33	94.58	14.60	92.51	11.82	95.07	11.21	95.20
MADDG [34]	17.69	88.06	24.50	84.51	22.19	84.99	27.89	80.02	23.07	84.40
RFM [35]	17.30	90.48	13.89	93.98	20.27	88.16	16.45	91.16	16.98	90.95
SSDG-R [22]	7.38	97.17	10.44	95.94	11.71	96.59	15.61	91.54	11.29	95.31
D ² AM [5]	12.70	95.66	20.98	85.58	15.43	91.22	15.27	90.87	16.10	90.83
ViT [20]	4.75	98.59	15.70	92.76	17.68	86.66	16.46	90.37	13.65	92.10
TransFAS [44]	7.08	96.69	9.81	96.13	10.12	95.53	15.52	91.10	10.63	94.86
DADN-CDS [49]	5.24	98.06	6.84	97.95	10.64	95.14	13.77	93.09	9.12	96.06
CIFAS [25]	5.95	96.32	10.66	95.30	8.50	97.24	13.17	93.44	9.57	95.58
Baseline	15.48	91.50	14.67	93.17	22.70	81.63	16.77	91.03	17.41	89.33
CF-PAD (ours)	8.11	96.43	11.78	95.64	16.50	91.50	9.87	95.13	11.57	94.68

Table 4. The results of the four multi-source cross-dataset evaluations comparing our CF-PAD and SOTA PAD solutions. Methods in the first block are conventional PAD solution (i.e., no domain adaptation), while methods in the second block are explicitly designed to target the domain shift problem. The bold number indicates the best performance under each protocol. Our CF-PAD demonstrates a very competitive PAD performance.

Method	HTER(%) ↓	AUC(%)
GRL Layer [17]	29.1	76.4
ADDA [37]	33.7	70.3
DA-FAS [33]	27.1	79.2
UCDA-FAS [29]	26.1	80.0
CIFAS [25]	24.6	83.2
Baseline	27.1	80.3
CF-PAD (ours)	23.5	84.2

Table 5. Comparison results of models trained on the O&C&M and tested on the large-scale CA.

Method	M&I → C		M&I → O	
	HTER(%) ↓	AUC(%)	HTER(%) ↓	AUC(%)
MS-LBP [27]	51.16	52.09	43.63	58.07
IDA [47]	45.16	58.80	54.52	42.17
MADDG [34]	41.02	64.33	39.35	65.10
RFM [35]	36.34	67.52	29.12	72.61
SSDG-R [22]	31.89	71.29	36.01	66.88
DR-MD-Net [42]	31.67	75.23	34.02	72.65
D ² AM [5]	32.65	72.04	27.70	75.36
CIFAS [25]	22.67	83.39	24.63	81.48
Baseline	25.78	80.56	23.24	84.66
CF-PAD (ours)	22.11	85.06	19.71	89.01

Table 6. Limited-source cross-dataset results in terms of HTER (%) and AUC (%).

6 presents the results of our method compared to the SOTA method. Our method improves the PAD generalizability in most cases and achieves the lowest average error rates. In addition, CF-PAD improves the worst case, where the highest HTER value (worst) among 12 cases is 34.00% obtained by CF-PAD and 38.50% by DR-UDA [43]. These results

implies that enriching the diversity of feature-level training data and the diversity of CIs leads to a better causal and representative feature learning as rationalized by Eq. 4. Fig. 6 illustrates the mean and standard deviation of the HTER values of the results showed in Tab. 7. The plot stresses that CF-PAD performs more consistently (the lowest average error rates, the lowest deviation, and the lowest errors of the worst-case) than other PAD solutions in the challenging scenarios.

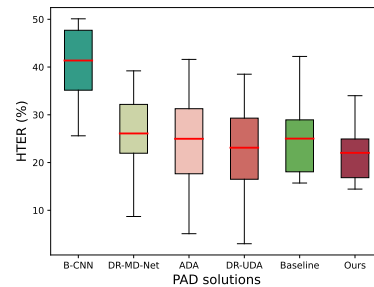


Figure 6. The Box and Whisker plot of the performance variation under single-source cross-dataset scenarios reported in Tab. 7. Each box refers to a model evaluated on a total of 12 protocols (B-CNN is binary CNN method). The red line within the box is the mean value of 12 testing cases, the box range represents the standard deviation, and the top straight line extended of the box is the worst-case. Note the lowest average HTER value of our CF-PAD, along with its low deviation and the optimal worst-case.

Method	C → I	C → M	C → O	I → C	I → M	I → O	M → C	M → I	M → O	O → I	O → M	O → C	Average	Worst
Binary CNN [50]	45.80	25.60	36.40	44.40	48.60	45.40	50.10	49.90	31.40	47.40	30.20	41.20	41.37 ± 8.42	50.01
ADA [41]	17.50	9.30	29.10	41.60	30.50	39.60	17.70	5.10	31.20	26.80	31.50	19.80	24.98 ± 11.28	41.60
DR-MD-Net [42]	26.10	20.20	24.70	39.20	23.20	33.60	34.30	8.70	31.70	27.60	22.00	21.80	26.09 ± 7.70	39.20
DR-UDA [43]	15.60	9.00	28.70	34.20	29.00	38.50	16.80	3.00	30.20	25.40	27.40	19.50	23.11 ± 10.50	38.50
Baseline	38.85	18.10	17.94	42.22	18.81	28.42	27.11	16.30	30.49	23.10	15.71	23.11	25.01 ± 8.74	42.22
CF-PAD (ours)	24.80	17.14	19.43	34.00	24.76	31.70	14.44	15.90	25.34	21.50	15.00	20.33	22.03 ± 6.33	34.00

Table 7. Single-source cross-dataset results in terms of HTER (%). The bold number indicates the lowest error rates, indicating a better generalizability of our CF-PAD.

Methods	Param. (M)	FLOPs (G)	HTER (%)
Auxiliary [26]	2.20	47.41	28.89
ResNet50-PS [51]	23.54	24.66	18.21
NAS-FAS [52]	2.94	52.67	16.10
LMFD [12]	101.72	36.11	13.47
ViTransPAD [28]	66.00	7.88	15.22
MADDG [34]	17.81	189.61	23.07
RFM [35]	3.87	95.79	16.89
SSDG-R [22]	12.23	14.26	11.29
ViT [20]	86.39	33.69	13.65
DADN-CDS [49]	23.61	56.00	9.12
CF-PAD (ours)	11.18	1.82	11.57

Table 8. Computational complexity vs. PAD Performance. HTER (%) refers to the average HTER values reported in Tab. 4. The proposed CF-PAD achieves the lowest complexity (FLOPs) and a very competitive PAD performance.

5.3. Performance vs. Computational Complexity

To enable the efficiency comparison, we account the trainable parameters of models and the floating point operations (FLOPs) of our method and the PAD works listed in Tab. 4 when it is feasible. Tab. 8 and Fig. 1 demonstrate the efficiency of the propose CF-PAD method by reporting the performance as the HTER, and computational complexity in terms of the number of trainable parameters and the FLOPs. Five PAD methods in Tab. 4 were ignored for efficiency comparison due to no open source implementation and no detailed model architecture available for accounting. Fig. 1 presents the number of FLOPs (x-axis) vs. the PAD performance (y-axis) vs. the trainable parameters (radius of the circle) of our method and the SOTA PADs. The ideal solution will tend to be placed on the bottom left corner (low error rates and low complexity) as well as small circle (less trainable parameters). The proposed CF-PAD method achieves the lowest FLOPs and very competitive PAD performance, indicating its relative high efficiency. The detailed values presented in Tab. 8 supports this conclusion.

5.4. Visualization and Analysis

We visualized the feature distribution learned by the baseline and our CF-PAD model under the protocol I&C&M → O in Fig. 7. To make a clear observation and avoid possible overlapping region, we randomly select 300 samples from each dataset and illustrate their distribution by using t-SNE [38]. Each color in the plots represent the different datasets (purple is unseen testing dataset OULU-NPU) and each shape refers to different classes (square and cross represent bona fide and attack samples, respectively).

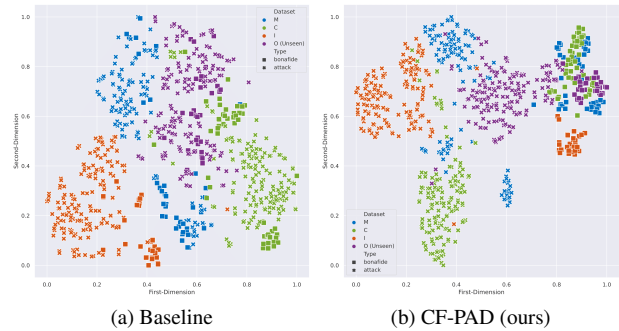


Figure 7. Feature visualization of the case I&C&M → O by baseline (a) and our CF-PAD solution (b), respectively. Different colors indicates the different dataset, where O (purple) is unseen testing data, and different shape represents the different classes, where square is bona fide and cross is attack. It is clear that CF-PAD offers a more discriminative feature space.

Comparing Fig. 7 (a) and (b), we observe that samples from the same dataset (same color) are more clustered by the baseline model and sample from the same classes (same shape) are more clustered by our CF-PAD, especially for the unseen O data. This observation proves that our CF-PAD learns a causal and generalized discriminative feature space for face PAD instead of focusing on features from different domains. Moreover, this observation implies that the proposed CF-PAD without accessing the target testing data is effective and reliable for complex practical scenarios.

6. Conclusion

This paper models the face PAD task from a causal view, aiming to enhance the generalizability of PAD under cross-dataset scenarios. The main idea is to approximate the model optimization from two aspects, 1) enriching the feature-level training data diversity via CGMixStyle and 2) performing a set of CIs on high-level features, to learn a causal and generalized feature representation. In addition, CGMixStyle and CI components introduce no extra trainable parameters, negligible computation complexity during training, and no computation resources in the inference phase. A wide set of experiments and comparisons reveal the effectiveness and efficiency of our proposed method.

Acknowledgment: This research work has been funded by the German Federal Ministry of Education and Research and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

- [1] Zinelabidine Boulkenafet, Jukka Komulainen, Lei Li, Xiaoyi Feng, and Abdenour Hadid. OULU-NPU: A mobile face presentation attack database with real-world variations. In *FG*, pages 612–618. IEEE Computer Society, 2017. 1, 5
- [2] Fadi Boutros, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. Elasticface: Elastic margin loss for deep face recognition. In *CVPR Workshops*, pages 1577–1586. IEEE, 2022. 1
- [3] Fadi Boutros, Vitomir Struc, Julian Fierrez, and Naser Damer. Synthetic data for face recognition: Current state and future prospects. *Image and Vision Computing*, 135:104688, 2023. 1
- [4] Guangyi Chen, Junlong Li, Jiwen Lu, and Jie Zhou. Human trajectory prediction via counterfactual analysis. In *ICCV*, pages 9804–9813. IEEE, 2021. 2
- [5] Zhihong Chen, Taiping Yao, Kekai Sheng, Shouhong Ding, Ying Tai, Jilin Li, Feiyue Huang, and Xinyu Jin. Generalizable representation learning for mixture domain face anti-spoofing. In *AAAI*, pages 1132–1139. AAAI Press, 2021. 1, 2, 5, 7
- [6] Ivana Chingovska, André Anjos, and Sébastien Marcel. On the effectiveness of local binary patterns in face anti-spoofing. In *BIOSIG*, volume P-196 of *LNI*, pages 1–7. GI, 2012. 5
- [7] Naser Damer and Kristiyan Dimitrov. Practical view on face presentation attack detection. In *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*. BMVA Press, 2016. 2
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE Computer Society, 2009. 5
- [9] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4690–4699. Computer Vision Foundation / IEEE, 2019. 1
- [10] Meiling Fang, Hamza Ali, Arjan Kuijper, and Naser Damer. Patchswap: Boosting the generalizability of face presentation attack detection by identity-aware patch swapping. In *IJCB*, pages 1–10. IEEE, 2022. 1
- [11] Meiling Fang, Fadi Boutros, Arjan Kuijper, and Naser Damer. Partial attack supervision and regional weighted inference for masked face presentation attack detection. In *16th IEEE FG, Jodhpur, India, December 15-18, 2021*, pages 1–8. IEEE, 2021. 2
- [12] Meiling Fang, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. Learnable multi-level frequency decomposition and hierarchical attention mechanism for generalized face presentation attack detection. In *IEEE/CVF WACV, Waikoloa, HI, USA, January 3-8, 2022*, pages 1131–1140. IEEE, 2022. 2, 5, 7, 8
- [13] Meiling Fang, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. Real masks and spoof faces: On the masked face presentation attack detection. *Pattern Recognition*, 123:108398, 2022. 1
- [14] Meiling Fang, Marco Huber, and Naser Damer. Synthaspoof: Developing face presentation attack detection based on privacy-friendly synthetic data. In *CVPR Workshops*, pages 1061–1070. IEEE, 2023. 4
- [15] Meiling Fang, Wufei Yang, Arjan Kuijper, Vitomir Struc, and Naser Damer. Fairness in face presentation attack detection. *CoRR*, abs/2209.09035, 2022. 5
- [16] Norman E. Fenton, Martin Neil, and Anthony C. Constantinou. The book of why: The new science of cause and effect, judea pearl, dana mackenzie. basic books (2018). *Artif. Intell.*, 284:103286, 2020. 2, 3
- [17] Yaroslav Ganin and Victor S. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1180–1189. JMLR.org, 2015. 7
- [18] Anjith George and Sébastien Marcel. Deep pixel-wise binary supervision for face presentation attack detection. In *ICB*, pages 1–8. IEEE, 2019. 5
- [19] York Hagmayer, Steven A Sloman, David A Lagnado, and Michael R Waldmann. Causal reasoning through intervention. *Causal learning: Psychology, philosophy, and computation*, pages 86–100, 2007. 4
- [20] Hsin-Ping Huang, Deqing Sun, Yaojie Liu, Wen-Sheng Chu, Taihong Xiao, Jinwei Yuan, Hartwig Adam, and Ming-Hsuan Yang. Adaptive transformers for robust few-shot cross-domain face anti-spoofing. In *ECCV (13)*, volume 13673 of *Lecture Notes in Computer Science*, pages 37–54. Springer, 2022. 2, 5, 7, 8
- [21] International Organization for Standardization. ISO/IEC DIS 30107-3:2016: Information Technology – Biometric presentation attack detection – P. 3: Testing and reporting, 2017. 5
- [22] Yunpei Jia, Jie Zhang, Shiguang Shan, and Xilin Chen. Single-side domain generalization for face anti-spoofing. In *CVPR*, pages 8481–8490. Computer Vision Foundation / IEEE, 2020. 1, 2, 5, 7, 8
- [23] Haoliang Li, Wen Li, Hong Cao, Shiqi Wang, Feiyue Huang, and Alex C. Kot. Unsupervised domain adaptation for face anti-spoofing. *IEEE Trans. Inf. Forensics Secur.*, 13(7):1794–1809, 2018. 1, 2
- [24] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain generalization with adversarial feature learning. In *CVPR*, pages 5400–5409. IEEE Computer Society, 2018. 5
- [25] Yuchen Liu, Yabo Chen, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. Causal intervention for generalizable face anti-spoofing. In *ICME*, pages 1–6. IEEE, 2022. 2, 4, 5, 7
- [26] Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *CVPR*, pages 389–398. IEEE Computer Society, 2018. 2, 7, 8
- [27] Jukka Määttä, Abdenour Hadid, and Matti Pietikäinen. Face spoofing detection from single images using micro-texture analysis. In *IJCB*, pages 1–7. IEEE Computer Society, 2011. 7

- [28] Zuheng Ming, Zitong Yu, Musab Al-Ghadi, Muriel Visani, Muhammad Muzzamil Luqman, and Jean-Christophe Burie. Vitranspad: Video transformer using convolution and self-attention for face presentation attack detection. In *ICIP*, pages 4248–4252. IEEE, 2022. 5, 7, 8
- [29] Ankush Panwar, Pratyush Singh, Suman Saha, Danda Pani Paudel, and Luc Van Gool. Unsupervised compound domain adaptation for face anti-spoofing. In *FG*, pages 1–8. IEEE, 2021. 5, 7
- [30] Judea Pearl. Causal inference. In *NIPS Causality: Objectives and Assessment*, volume 6 of *JMLR Proceedings*, pages 39–58. JMLR.org, 2010. 2, 3
- [31] Sandip Purnapatra, Nic Smalt, Keivan Bahmani, Priyanka Das, David Yambay, Amir Mohammadi, Anjith George, Thirimachos Bourlai, Sébastien Marcel, Stephanie Schuckers, Meiling Fang, Naser Damer, Fadi Boutros, Arjan Kuijper, Alperen Kantarci, Basar Demir, Zafer Yildiz, Zabi Ghafoory, Hasan Dertli, Hazim Kemal Ekenel, Son Vu, Vassilis Christophides, Dashuang Liang, Guanghao Zhang, Zhanlong Hao, Junfu Liu, Yufeng Jin, Samo Liu, Samuel Huang, Salieri Kuei, Jag Mohan Singh, and Raghavendra Ramachandra. Face liveness detection competition (livdet-face) - 2021. In *International IEEE Joint Conference on Biometrics, IJCB 2021, Shenzhen, China, August 4-7, 2021*, pages 1–10. IEEE, 2021. 4
- [32] Yongming Rao, Guangyi Chen, Jiwen Lu, and Jie Zhou. Counterfactual attention learning for fine-grained visual categorization and re-identification. In *ICCV*, pages 1005–1014. IEEE, 2021. 2
- [33] Suman Saha, Wenhao Xu, Menelaos Kanakis, Stamatios Georgoulis, Yuhua Chen, Danda Pani Paudel, and Luc Van Gool. Domain agnostic feature learning for image and video based face anti-spoofing. In *CVPR Workshops*, pages 3490–3499. IEEE, 2020. 7
- [34] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C. Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *CVPR*, pages 10023–10031. Computer Vision Foundation / IEEE, 2019. 1, 2, 5, 7, 8
- [35] Rui Shao, Xiangyuan Lan, and Pong C. Yuen. Regularized fine-grained meta face anti-spoofing. In *AAAI*, pages 11974–11981. AAAI Press, 2020. 1, 2, 5, 7, 8
- [36] Ryota Shimizu, Kosuke Asako, Hiroki Ojima, Shohei Morinaga, Mototsugu Hamada, and Tadahiro Kuroda. Balanced mini-batch training for imbalanced image data classification with neural network. In *AIAL*, pages 27–30. IEEE, 2018. 5
- [37] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 2962–2971. IEEE Computer Society, 2017. 7
- [38] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. 8
- [39] Vladimir Vapnik. Principles of risk minimization for learning theory. In *NIPS*, pages 831–838. Morgan Kaufmann, 1991. 3
- [40] Chien-Yi Wang, Yu-Ding Lu, Shang-Ta Yang, and Shang-Hong Lai. Patchnet: A simple face anti-spoofing framework via fine-grained patch recognition. In *CVPR*, pages 20249–20258. IEEE, 2022. 7
- [41] Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Improving cross-database face presentation attack detection via adversarial domain adaptation. In *ICB*, pages 1–8. IEEE, 2019. 1, 2, 8
- [42] Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Cross-domain face presentation attack detection via multi-domain disentangled representation learning. In *CVPR*, pages 6677–6686. Computer Vision Foundation / IEEE, 2020. 5, 7, 8
- [43] Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Unsupervised adversarial domain adaptation for cross-domain face presentation attack detection. *IEEE Trans. Inf. Forensics Secur.*, 16:56–69, 2021. 5, 7, 8
- [44] Zhuo Wang, Qiangchang Wang, Weihong Deng, and Guodong Guo. Face anti-spoofing using transformers with relation-aware mechanism. *IEEE Trans. Biom. Behav. Identity Sci.*, 4(3):439–450, 2022. 1, 2, 7
- [45] Tyler Vander Weele. Explanation in causal inference: Methods for mediation and interaction. *Oxford University Press*, 2015. 4
- [46] Di Wen, Hu Han, and Anil K. Jain. Face spoof detection with image distortion analysis. *IEEE Trans. Inf. Forensics Secur.*, 10(4):746–761, 2015. 5
- [47] Di Wen, Hu Han, and Anil K. Jain. Face spoof detection with image distortion analysis. *IEEE Trans. Inf. Forensics Secur.*, 10(4):746–761, 2015. 7
- [48] Zheng Xu, Wen Li, Li Niu, and Dong Xu. Exploiting low-rank structure from latent domains for domain generalization. In *ECCV (3)*, volume 8691 of *Lecture Notes in Computer Science*, pages 628–643. Springer, 2014. 4, 7
- [49] Wenjun Yan, Ying Zeng, and Haifeng Hu. Domain adversarial disentanglement network with cross-domain synthesis for generalized face anti-spoofing. *IEEE Trans. Circuits Syst. Video Technol.*, 32(10):7033–7046, 2022. 1, 2, 4, 5, 7, 8
- [50] Jianwei Yang, Zhen Lei, and Stan Z. Li. Learn convolutional neural network for face anti-spoofing. *CoRR*, abs/1408.5601, 2014. 8
- [51] Zitong Yu, Xiaobai Li, Jingang Shi, Zhaoqiang Xia, and Guoying Zhao. Revisiting pixel-wise supervision for face anti-spoofing. *IEEE Trans. Biom. Behav. Identity Sci.*, 3(3):285–295, 2021. 1, 2, 5, 7, 8
- [52] Zitong Yu, Jun Wan, Yunxiao Qin, Xiaobai Li, Stan Z. Li, and Guoying Zhao. NAS-FAS: static-dynamic central difference network search for face anti-spoofing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(9):3005–3023, 2021. 1, 2, 5, 7, 8
- [53] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Proc. Lett.*, 23(10):1499–1503, 2016. 5
- [54] Xiheng Zhang, Yongkang Wong, Xiaofei Wu, Juwei Lu, Mohan S. Kankanhalli, Xiangdong Li, and Weidong Geng. Learning causal representation for training cross-domain pose estimator via generative interventions. In *ICCV*, pages 11250–11260. IEEE, 2021. 3

- [55] Yuanhan Zhang, Zhenfei Yin, Yidong Li, Guojun Yin, Junjie Yan, Jing Shao, and Ziwei Liu. Celeba-spoof: Large-scale face anti-spoofing dataset with rich annotations. In *ECCV. Glasgow, UK, August 23-28, 2020.*, volume 12357 of *Lecture Notes in Computer Science*, pages 70–85. Springer, 2020. [1](#), [4](#), [5](#)
- [56] Zhiwei Zhang, Junjie Yan, Sifei Liu, Zhen Lei, Dong Yi, and Stan Z. Li. A face antispoofing database with diverse attacks. In *ICB*, pages 26–31. IEEE, 2012. [1](#), [5](#)
- [57] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*. OpenReview.net, 2021. [3](#), [5](#), [6](#)