Feature Generation and Hypothesis Verification for Reliable Face Anti-spoofing

Shice Liu[1*], **Shitao Lu**[1,2*], **Hongyi Xu**[1,3], **Jing Yang**[1†], **Shouhong Ding**[1†], **Lizhuang Ma**[2,3]

[1]Youtu Lab, Tencent, Shanghai, China
[2]East China Normal University, Shanghai, China
[3]Shanghai Jiao Tong University, Shanghai, China
{shiceliu,alonyang,ericshding}@tencent.com; lusto32768@gmail.com; xuhongyi@sjtu.edu.cn; lzma@cs.ecnu.edu.cn

Abstract

Although existing face anti-spoofing (FAS) methods achieve high accuracy in intra-domain experiments, their effects drop severely in cross-domain scenarios because of poor generalization. Recently, multifarious techniques have been explored, such as domain generalization and representation disentanglement. However, the improvement is still limited by two issues: 1) It is difficult to perfectly map all faces to a shared feature space. If faces from unknown domains are not mapped to the known region in the shared feature space, accidentally inaccurate predictions will be obtained. 2) It is hard to completely consider various spoof traces for disentanglement. In this paper, we propose a Feature Generation and Hypothesis Verification framework to alleviate the two issues. Above all, feature generation networks which generate hypotheses of real faces and known attacks are introduced for the first time in the FAS task. Subsequently, two hypothesis verification modules are applied to judge whether the input face comes from the real-face space and the real-face distribution respectively. Furthermore, some analyses of the relationship between our framework and Bayesian uncertainty estimation are given, which provides theoretical support for reliable defense in unknown domains. Experimental results show our framework achieves promising results and outperforms the state-of-the-art approaches on extensive public datasets.

1 Introduction

Nowadays, face recognition (FR) has been widely used in many AI systems in our daily lives. However, endless face presentation attacks continuously threaten the security of face applications. To endow the AI systems with this important defensive capability, FAS techniques must be equipped.

In the past few years, various FAS approaches have been proposed. Some representative methods are handcrafted feature-based (Boulkenafet et al. 2015), movement-based (Pan et al. 2007), distortion-based (Wen et al. 2015), physiological signal-based (Li et al. 2016) and deep feature-based (Yang et al. 2014). Although these methods perform well in intra-domain experiments, the effects decrease severely in cross-domain scenarios due to poor generalization. With the aim of more generalized models, two categories of methods are currently being studied.

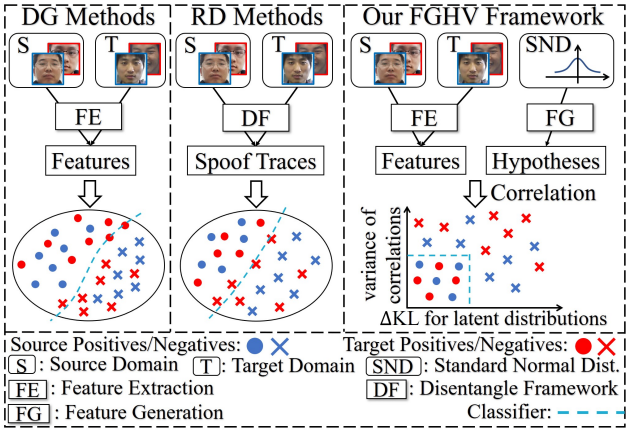


Figure 1: Comparison among domain generalization-based methods (DG), representation disentanglement-based methods (RD) and our FGHV framework.

Domain generalization-based methods are exploited to learn a domain-agnostic feature space. Intuitively, if a model works well in several known domains, it would be more likely to be effective in other unknown domains. To achieve it, commonly used solutions include metric learning (Jia et al. 2020), adversarial training (Liu et al. 2021b), meta-learning (Wang et al. 2021) and special structures (Yu et al. 2021c). Despite using different techniques, all these methods aim to attain a better feature extraction backbone so that regardless of which domain the input comes from, the backbone always outputs generalized features. However, in real scenarios, it is hard to perfectly map real and fake faces from different domains to a generalized shared feature space. Additionally, from the perspective of Bayesian uncertainty estimation, models will be prone to giving wrong results if they are fed with what they do not know.

Representation disentanglement-based methods hold the view that features can be partitioned into liveness-related parts (i.e., spoof trace) and liveness-unrelated parts (e.g., appearance, identification, age), and only the liveness-related parts are leveraged to classify for better generalization. To

*These authors contributed equally.
†These authors are the corresponding authors.

this end, CycleGAN (Zhang et al. 2020) or arithmetic operations on images (Liu et al. 2020) is utilized to extract the spoof trace. However, spoof traces vary with the type and style of attacks. That's to say, these methods might be confused if inputs from unknown domains are given.

In order to tackle such issues, we propose a Feature Generation and Hypothesis Verification (FGHV) framework to verify both real-face feature space and real-face distribution by generating hypotheses. Fig. 1 depicts the comparison to two related methods. Firstly, we leverage two feature generation networks to generate features of real faces and known attacks respectively which are also termed as hypotheses. Different from domain generalization-based methods that construct a shared feature space from raw images in huge RGB space, hypotheses are generated from latent vectors which are sampled from the same distribution during training and testing periods. Compared with representation disentanglement-based methods that mine spoof traces, we mainly care about features of real faces which are more similar across domains. Secondly, we devise a Feature Hypothesis Verification Module (FHVM) to estimate to what extent the input face comes from the real-face feature space. Specifically, after generating enough real-face hypotheses via the feature generation network, the FHVM evaluates the consistency of correlations between each hypothesis and the input face. Thirdly, we design a Gaussian Hypothesis Verification Module (GHVM) to measure the KL divergence between the input face distribution and the real-face distribution in the latent space. Furthermore, from the viewpoint of Bayesian uncertainty estimation, we analyze that our framework actually constructs a more effective prior distribution than Bayesian neural network (Shridhar et al. 2018a,b; Farquhar et al. 2020) to estimate the epistemic uncertainty, which brings greater effects and better reliability.

The main contributions are summarized as follows:

• The FAS problem is modeled as a classification problem of real faces and non-real faces, and to the best of our knowledge, feature generation networks for producing hypotheses are introduced into the FAS task for the first time.

• Two effective hypothesis verification modules are proposed to judge whether the input face comes from the real-face feature space and the real-face distribution respectively.

• The relationship between our framework and Bayesian uncertainty estimation is clearly stated. And comprehensive experiments and visualizations demonstrate the effectiveness and reliability of our approach.

2 Related Work

We briefly review related works on traditional FAS methods, and then detail domain generalization-based methods and representation disentanglement-based methods which are two popular research orientations based on deep learning. Finally, we give a sketch of Bayesian deep learning.

Traditional Face Anti-Spoofing. Early researchers have introduced lots of handcrafted features to achieve FAS task, such as LBP (Boulkenafet et al. 2015; de Freitas Pereira et al. 2012, 2013; Määttä et al. 2011), HOG (Komulainen et al. 2013; Yang et al. 2013) and SIFT (Patel et al. 2016). Since they are too simple to perform well, more liveness cues are explored later, such as eye blinking (Pan et al. 2007), face movement (Wang et al. 2009), light changing (Zhang et al. 2021a) and remote physiological signals (e.g., rPPG (Li et al. 2016; Liu et al. 2018; Yu et al. 2021b; Hu et al. 2021)). However, these methods are always limited by low accuracy or complicated process in video data.

Domain Generalization-Based Face Anti-Spoofing. Although deep learning facilitates the FAS task, the generalization ability for multiple domains still need to be improved. To this end, researchers have tapped the potential of various techniques. Some approaches measured and constrained the distance of features or domains to obtain domain-agnostic features. For instance, Li et al. (2018a) used the MMD distance to make features unrelated with domains. Jia et al. (2020) and Yang et al. (2021) introduced triplet loss (Li et al. 2019a,b) and Zhang et al. (2021b) even constructed a similarity matrix to constrain the distance between features. Also, many meta-learning-based methods were exploited to find a generalized space among multiple domains (Shao et al. 2020; Qin et al. 2020; Kim and Lee 2021; Chen et al. 2021b; Wang et al. 2021; Qin et al. 2021). Besides, Wang et al. (2019a) and Liu et al. (2021b) utilized adversarial training, while Yu et al. (2020a,e,c,b, 2021c,a) and Chen et al. (2021a) proposed some special network structures and loss functions for better generalization. Although the effects were improved from different aspects, all these methods intended to make the feature extraction backbone generalized. Nevertheless, mapping all real faces and attacks of different domains to a shared feature space is difficult and the shared feature space is usually not generalized well. Furthermore, in the view of uncertainty estimation, given some inputs from unknown domains, these models might produce surprisingly bad results because of incapacity to distinguish what they know and what they don't.

Representation Disentanglement-Based Face Anti-Spoofing. Some other representative methods for generalization realized FAS through detecting spoof traces which were disentangled from input faces. Stehouwer et al. (2020) would like to synthesize and identify noise patterns from seen and unseen medium/sensor combinations, and then benefited from adversarial training. Liu et al. (2020) intended to disentangle spoof traces via arithmetic operations, adversarial training and so on. Especially, motivated by CycleGAN (Zhu et al. 2017), Zhang et al. (2020) finished disentangling liveness features of real faces and attacks by exchanging, reclassification and adversarial training. Generally speaking, these methods successfully disentangled liveness-related features and thus had relatively stronger interpretability. However, the set of attacks is an open set and it is almost impossible to construct the distribution of attacks or disentangle diverse spoof traces accurately.

Bayesian Deep Learning. Bayesian deep learning is one of the commonly used uncertainty estimation methods, which can capture both epistemic uncertainty and aleatoric uncertainty. The network combined with Bayesian deep learning is referred to Bayesian Neural Network (BNN) (Denker and LeCun 1990; MacKay 1992). Later, several approximation approaches, such as variational inference

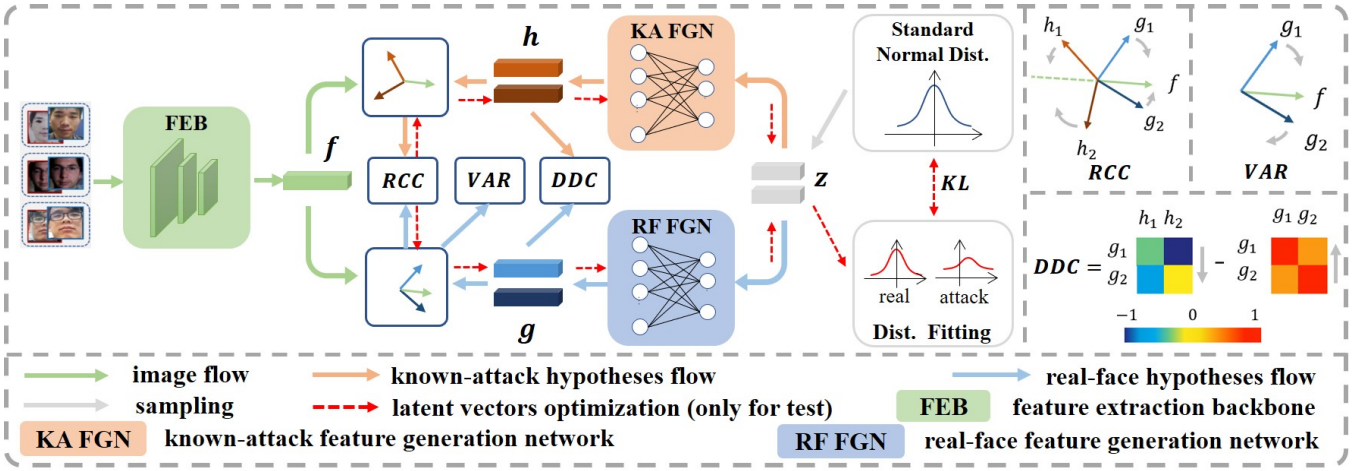


Figure 2: An overview of the proposed Feature Generation and Hypothesis Verification framework. Feature Hypothesis Verification Module introduces the variance constraint (VAR) in both training and testing. Besides, Gaussian Hypothesis Verification Module draws support from Relative Correlation Constraint (RCC) and Distribution Discrimination Constraint (DDC) in training, and then acquires the distribution distance (i.e., KL divergence) via latent vectors optimization in testing. The gray arrows on the right prompt how hypotheses are optimized after a real-face image is input. RCC makes $cos(f, g_i)$ higher and $cos(f, h_i)$ lower. VAR compels all $cos(f, g_i)$ to be equal. DDC urges $cos(h_i, g_j)$ lower and $cos(g_i, g_j)$ higher.

and MC dropout, were presented to model the uncertainty (Graves 2011; Blundell et al. 2015; Hernandez-Lobato et al. 2016; Gal and Ghahramani 2016). Recently, methods incorporated with Bayesian deep learning have been applied to various fields, such as recommender systems (Li and She 2017), object tracking (Kosiorek et al. 2018), health care (Wang et al. 2019b) and salient object detection (Tang et al. 2021). Whereas, there are rare applications of Bayesian deep learning to face anti-spoofing.

3 Proposed Method

As shown in Fig. 2, the **Feature Generation and Hypothesis Verification** framework contains a traditional Feature Extraction Backbone, two novel Feature Generation Networks and two powerful hypothesis verification modules: 1) **Feature Hypothesis Verification Module** estimates the possibility that the input face comes from the real-face feature space. 2) **Gaussian Hypothesis Verification Module** directly measures the KL divergence between the distribution of the input face and that of real faces in the latent space.

3.1 Feature Extraction Backbone

In our framework, the feature extraction backbone takes a single face image $I \in [0, 255]^{3\times H\times W}$ as input and outputs the liveness feature vector $\boldsymbol{f} \in \mathbb{R}^{C_f}$, where $H \times W$ is the spatial size and C_f is the dimension of the feature vector.

In the training period, many existing works prefer to directly constrain this feature vector $\boldsymbol{f}$ to obtain a shared feature space. However, after considering the diversity of multiple domains, we still find it is too difficult to construct a perfectly generalized feature mapping via limited training data. After all, the raw image space $[0, 255]^{3\times H\times W}$ is so large that we can only get a fraction of samples for training.

Instead, we agree only real faces in multiple domains are similar, so we merely attempt to classify real and non-real faces. To achieve it, we mainly construct the real-face distribution and secondarily make the known-attack distribution the auxiliary by means of feature generation networks.

3.2 Feature Generation Network

Inspired by GAN (Goodfellow et al. 2014), we design two generation networks to fit the distribution of real faces and known attacks. Nevertheless, considering that noisy or blurry images might be generated (Di Biase et al. 2021) and the detailed information in images is essential to the FAS task, we urge the two generation networks to directly generate features instead of raw face images.

The two feature generation networks take a latent vector $\boldsymbol{z} \in \mathbb{R}^{C_z}$ as input, where C_z is the dimension of the latent vector and $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ is sampled from the standard multivariate normal distribution. One feature generation network generates a real-face feature vector $\boldsymbol{g} \in \mathbb{R}^{C_f}$ and the other generates a known-attack feature vector $\boldsymbol{h} \in \mathbb{R}^{C_f}$.

For as much as these generated real-face and known-attack features are not extracted from real collected images, it is more appropriate to name the generated features as hypotheses. For the sake of effective hypotheses, the feature generation networks are constrained in regard to both feature space and feature distribution with the assist of the following two hypothesis verification modules.

3.3 Feature Hypothesis Verification Module

In an attempt to optimize the feature extraction backbone and the feature generative network in terms of feature space, Feature Hypothesis Verification Module is proposed, which evaluates the consistency of correlations between each real-face hypothesis and the input face feature to determine

whether the input face comes from real-face feature space. In our definition, the real-face feature space is a space composed of features which have high consistency of correlations with real-face hypotheses.

Intuitively, the correlations between real-face features should be high, while the correlations between real-face features and non-real-face features should be low. In fact, after obtaining the face feature $\boldsymbol{f}$ and the real-face hypothesis $\boldsymbol{g}$, we utilize cosine similarity to measure the correlation:

$$cos(\boldsymbol{f}, \boldsymbol{g}) = \frac{\boldsymbol{f} \cdot \boldsymbol{g}}{\|\boldsymbol{f}\|_2 \cdot \|\boldsymbol{g}\|_2}. \tag{1}$$

In pursuit of robust consistency estimation, we take multiple hypotheses into account. And then there will be three situations: (1) If the input face feature has high correlations with each real-face hypothesis, the input face will be thought to be in the same feature space as the real faces. (2) If the input face feature has low correlations with each real-face hypothesis, the input face will also be regarded as a sample from the real-face space but not from the real-face distribution. This situation will be discussed in Sec. 3.4. (3) If some correlations are high and the others are low, it will mean that the input face feature space and the real-face feature space have some intersecting subspaces. For this situation, we can determine how these two feature spaces match via measuring the consistency of correlations.

Concretely, we simultaneously sample N Gaussian vectors $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_N\}$ to generate N real-face hypotheses $\{\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_N\}$ and calculate N cosine similarities between $\boldsymbol{f}$ and each hypothesis. By reason that mathematical variance can capture the consistency, we make the variance of cosine similarities the indicator of space intersection size as Eq. 2. For real-face inputs, the variance should be small. As shown in Fig. 2, the included angles between the input face feature and each real-face hypothesis tend to be similar. On the contrary, for non-real-face inputs, the variance is usually large and those included angles tend to be different.

$$VAR = \frac{1}{N-1} \sum_{i=1}^{N} (cos(\boldsymbol{f}, \boldsymbol{g}_i) - \frac{\sum_{j=1}^{N} cos(\boldsymbol{f}, \boldsymbol{g}_j)}{N})^2 \tag{2}$$

Even though the variance constraint can not distinguish the first and the second situations above, it has ability to identify the third situation which is the most common situation in actual usage. Moreover, we will partition the first and the second situations in Sec. 3.4. It should be noted that the variance constraint does not apply to known-attack hypotheses, because there are various attack types and any attack is not necessarily highly correlated with other known attacks.

Finally, we conclude the advantages of the FHVM by revisiting some domain generalization methods (Jia et al. 2020; Yang et al. 2021). These methods map real faces and known attacks to a shared space, and agree that the scores of real faces should be close to 1 and those of attacks should be close to 0. But in this way, if any input face from unknown domains is not mapped to the known region in the shared space, the model will probably output an inaccurate score. Instead, the FHVM urges the feature extraction backbone to map real faces and non-real faces to different spaces. Therefore, before the final prediction is given, we can utilize the variance to check whether the input face belongs to the real-face feature space, which brings on better reliability.

3.4 Gaussian Hypothesis Verification Module

Since the distribution of real faces is a distribution composed of features which have high correlations with real-face hypotheses and is only a manifold in the real-face space, not all hypotheses in the real-face space are right real-face features, which explains the second situation illustrated in Sec. 3.3. To alleviate this issue, we introduce Gaussian Hypothesis Verification Module which brings in two constraints and measures KL divergence between the distribution of input faces and that of real faces.

Relative Correlation Constraint. Considering that the real-face distribution is only a part of meaningful regions in the real-face space, we take the attitude that it is necessary to increase constraints to make this distribution more accurate. Intuitively, for real-face inputs, the correlations with real-face hypotheses should be higher than those with known-attack hypotheses. As for non-real-face inputs, the goal is the opposite. Thus, we propose Relative Correlation Constraint (RCC) in a cross-entropy-like form. After formula derivation, the constraint is represented by Eq. 3, where y' is 1 for real-face inputs and is 0 for non-real-face inputs. In particular, Fig. 2 gives an example of how two types of hypotheses are optimized with respect to the given real-face input.

$$\begin{aligned} RCC = \frac{1}{N} \sum_{i=1}^{N} (ln(e^{cos(\boldsymbol{f}, \boldsymbol{g}_i)} + e^{cos(\boldsymbol{f}, \boldsymbol{h}_i)}) \\ -y' cos(\boldsymbol{f}, \boldsymbol{g}_i) - (1 - y') cos(\boldsymbol{f}, \boldsymbol{h}_i)) \end{aligned} \tag{3}$$

Distribution Distance Measurement. We calculate KL divergence between the input face distribution and the standard normal distribution in the latent space where $\boldsymbol{z}$ is sampled. For real-face inputs, the RCC losses are usually small, so the modifications of $\boldsymbol{z}_i$ are minor. Considering $\boldsymbol{z}^{(0)}$ are sampled from the standard normal distribution, $\boldsymbol{z}^{(M)}$ would also obey the standard normal distribution. On the contrary, for non-real-face inputs, the modifications are so major that $\boldsymbol{z}^{(M)}$ would obey another unknown distribution.

Enlightened by Schlegl et al. (2017), we utilize the Gradient Descent approach to search for the corresponding latent vector of the input face. Specifically, we assume that the input face is a real face, i.e., $y' = 1$, and then decrease the RCC loss via optimizing the latent vector $\boldsymbol{z}$ in the hope of acquiring the real-face hypotheses with higher correlations and the known-attack hypotheses with lower correlations. In practice, given an original latent vector $\boldsymbol{z}^{(0)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, the RCC loss can be calculated and a new latent vector $\boldsymbol{z}^{(1)}$ will be obtained via Eq. 4, where α is the step length for Gradient Descent. The latent vector $\boldsymbol{z}^{(1)}$ is closer to the corresponding latent vector of the input face than $\boldsymbol{z}^{(0)}$. And after M iterations, $\boldsymbol{z}^{(M)}$ is basically able to represent the corresponding latent vector of the input face. Moreover, since N latent vec-

tors are sampled before, N corresponding latent vectors of the input face $\{\boldsymbol{z}_1^{(M)}, \boldsymbol{z}_2^{(M)}, \cdots, \boldsymbol{z}_N^{(M)}\}$ are obtained.

$$\boldsymbol{z}^{(1)} = \boldsymbol{z}^{(0)} - \alpha \frac{\partial RCC_{y'=1}^{(0)}}{\partial \boldsymbol{z}^{(0)}} \tag{4}$$

For simplification, we assume the corresponding latent vectors of the input face obey a multivariate normal distribution. KL divergence with the standard normal distribution $\mathcal{N}(0,1)$ can be computed as Eq. 5, where μ and σ^2 are respectively the estimated mean and variance for a certain dimension in these latent vectors. After calculating the KL divergences for all dimensions, we average them to acquire the final KL divergence KL. We use the difference of two KL divergences $\Delta KL = KL^{(M)} - KL^{(0)}$ to judge whether the input face is real, where $KL^{(0)}$ and $KL^{(M)}$ are the KL divergence before the initial iteration and that after the final iteration, respectively. Note that, $KL^{(0)}$ should be zero theoretically. But allowing for the limited number of samples from the standard normal distribution, it is actually a quite small value. Fortunately, in the experiments, we find that ΔKL is usually a small value for real-face inputs, while it is a large value for non-real-face inputs.

$$KL = -log\sigma + \frac{\sigma^2 + \mu^2}{2} - \frac{1}{2} \tag{5}$$

Distribution Discrimination Constraint. For further improving the discriminative ability of the real-face distribution, we not only make real-face hypotheses more concentrated, but also make the distances between real-face hypotheses and known-attack hypotheses farther. Thus, we impose the distribution discrimination constraint as Eq. 6.

$$DDC = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (cos(\boldsymbol{g}_i, \boldsymbol{h}_j) - cos(\boldsymbol{g}_i, \boldsymbol{g}_j)) \tag{6}$$

Overall Loss. The overall loss function Eq. 7 is the combination of the variance constraint, the relative correlation constraint and the distribution discrimination constraint, where λ_1 and λ_2 are the weights for balance.

$$\mathcal{L}_{overall} = (2y' - 1) \cdot VAR + \lambda_1 \cdot RCC + \lambda_2 \cdot DDC \tag{7}$$

3.5 Relation to Bayesian Uncertainty Estimation

Bayesian deep learning obtains epistemic uncertainty and aleatoric uncertainty by formulating probability distributions over the model parameters and outputs. Especially, the epistemic uncertainty is formulated by placing a prior distribution over the model parameters. Given some inputs, the epistemic uncertainty can be estimated by measuring how much the output varies with these sampled parameters. In previous works (Shridhar et al. 2018a,b; Farquhar et al. 2020), the prior distribution is usually set as a multivariate normal distribution with learnable means and variances. As the model parameters are sampled from the learnt distribution, the variance of the outputs is leveraged to approximate the epistemic uncertainty as Eq. 8, where $\hat{\boldsymbol{y}}_t = F^{\hat{W}_t}(\boldsymbol{x})$ is the t-th sampled output for random weights $\hat{W}_t \sim q(W)$, and $q(W)$ is the learnt normal distribution.

$$Epi(\boldsymbol{y}) \approx \frac{1}{T} \sum_{t=1}^{T} \hat{\boldsymbol{y}}_t^2 - (\frac{1}{T} \sum_{t=1}^{T} \hat{\boldsymbol{y}}_t)^2 \tag{8}$$

Coincidentally, our proposed approach quite matches the idea of epistemic uncertainty estimation. The feature generation network is equivalent to the parameter generation network for the fully-connected layers, which generates random weights $\hat{W}_t$. And the cosine similarity can be regarded as the sampled output $\hat{\boldsymbol{y}}_t$.

Nevertheless, compared with the previous uncertainty estimation methods which set $q(W)$ to learnable normal distributions, we make $q(W)$ an arbitrary joint distribution which is directly constructed by a generation network. By removing the strong prior assumption that each model parameter obeys an independent normal distribution, we achieve more accurate epistemic uncertainty estimation.

4 Experiments

4.1 Evaluation Basis

Datasets. Above all, we conduct the cross-dataset testing on four public datasets, i.e., OULU-NPU (denoted as O) (Boulkenafet et al. 2017), CASIA-MFSD (denoted as C) (Zhang et al. 2012), Idiap Replay-Attack (denoted as I) (Chingovska et al. 2012) and MSU-MFSD (denoted as M) (Wen et al. 2015). After that, the cross-type testing is carried out on the rich-type dataset, i.e., SiW-M (Liu et al. 2019).

Classification Bases. There are three bases for classification: (1) The mean of N *softmax* outputs derived from $cos(\boldsymbol{f}, \boldsymbol{g}_i)$ and $cos(\boldsymbol{f}, \boldsymbol{h}_i)$. (2) The variance calculated by Eq. 2. (3) The ΔKL claimed in Sec. 3.4. In cross-dataset testing, we only use the first score for fair comparison. In cross-type testing, we use all three scores to show the effectiveness of our approach.

Implementation Details. All experiments are conducted via PyTorch on a 32GB Tesla-V100 GPU. For fair comparison, the architecture of our feature extraction network is DepthNet (Liu et al. 2018) which is the same as most alternative methods. The structures of two feature generation networks are the same, each of which consists of two fully-connected layers and a leaky ReLU activation function. The input is only an RGB image (resized to 3x256x256) and has no need of HSV image. During the training period, the framework is trained with SGD optimizer where the momentum is 0.9 and the weight decay is 5e-4. The learning rate is initially 1e-3 and drops to 1e-4 after 50 epochs. The hyper-parameters λ_1 and λ_2 are both set to 1. Our source code is available at https://github.com/lustoo/FGHV.

4.2 Comparison to Alternative Approaches

Cross-dataset testing. We strictly follow the previous methods (Jia et al. 2020; Qin et al. 2020) and select one dataset for testing and the other three datasets for training. As for evaluation, we follow the popular metrics, i.e., the Half Total Error Rate (HTER) and the Area Under Curve (AUC), and let the *Softmax* scores derived from real-face correlations and

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 (Määttä et al. 2011)	50.30	51.64	29.76	78.50	54.28	44.98	50.29	49.31
Auxiliary(Depth) (Liu et al. 2018)	29.14	71.69	22.72	85.88	33.52	73.15	30.17	77.61
MMD-AAE (Li et al. 2018b)	31.58	75.18	27.08	83.19	44.59	58.29	40.98	63.08
MADDG (Shao et al. 2019)	22.19	84.99	17.69	88.06	24.50	84.51	27.98	80.02
SSDG-M (Jia et al. 2020)	18.21	94.61	16.67	90.47	23.11	85.45	25.17	81.83
RFM (Shao et al. 2020)	17.30	90.48	13.89	93.98	20.27	88.16	16.45	91.16
NAS-FAS (Yu et al. 2020d)	**11.63**	**96.98**	16.85	90.42	15.21	92.64	**13.16**	**94.18**
DRDG (Liu et al. 2021b)	15.56	91.79	12.43	95.81	19.05	88.79	15.63	91.75
D^2AM (Chen et al. 2021b)	15.43	91.22	12.70	95.66	20.98	85.58	15.27	90.87
Self-DA (Wang et al. 2021)	15.60	90.10	15.40	91.80	24.50	84.40	23.10	84.30
ANRL (Liu et al. 2021a)	16.03	91.04	10.83	96.75	17.85	89.26	15.67	91.90
Ours	16.29	90.11	**9.17**	**96.92**	**12.47**	**93.47**	13.58	93.55

Table 1: Comparison to face anti-spoofing methods on the cross-dataset testing task for domain generalization.

Method	Metrics	Rep.	Pri.	Mask Attacks					Makeup Attacks			Partial Attacks			Average
				Hal.	Sil.	Tra.	Pap.	Man.	Obf.	Imp.	Cos.	Fun.	Gla.	Par.	
Auxiliary (Liu et al. 2018)	ACER	16.8	6.9	19.3	14.9	52.1	8.0	12.8	55.8	13.7	11.7	49.0	40.5	5.3	23.6±18.5
	EER	14.0	4.3	11.6	12.4	24.6	7.8	10.0	72.3	10.1	9.4	21.4	18.6	4.0	17.0±17.7
DTN (Liu et al. 2019)	ACER	9.8	**6.0**	15.0	18.7	36.0	4.5	7.7	48.1	11.4	14.2	19.3	19.8	8.5	16.8±11.1
	EER	10.0	**2.1**	14.4	18.6	26.5	5.7	9.6	50.2	10.1	13.2	19.8	20.5	8.8	16.1±12.2
SpoofTrace (Liu et al. 2020)	ACER	**7.8**	7.3	7.1	12.9	**13.9**	4.3	6.7	53.2	4.6	19.5	20.7	21.0	5.6	14.2±13.2
	EER	**7.6**	3.8	8.4	13.8	14.5	5.3	4.4	35.4	**0.0**	19.3	21.0	20.8	1.6	12.0±10.0
NAS-FAS (Yu et al. 2020d)	ACER	9.3	7.9	11.4	12.1	15.8	**1.9**	2.7	28.5	0.4	15.1	**16.5**	16.0	3.8	10.9±7.8
	EER	9.3	6.8	9.7	11.1	**12.5**	**2.7**	**0.0**	26.1	**0.0**	15.0	**15.1**	13.4	2.3	9.5±7.4
DC-CDN (Yu et al. 2021c)	ACER	12.1	9.7	14.1	**7.2**	14.8	4.5	**1.6**	40.1	**0.4**	11.4	20.1	16.1	**2.9**	11.9±10.3
	EER	10.3	8.7	11.1	**7.4**	**12.5**	5.9	**0.0**	39.1	**0.0**	12.0	18.9	13.5	**1.2**	10.8±10.1
Ours	ACER	8.4	7.3	**5.2**	9.8	14.2	3.2	4.1	**16.7**	1.9	**9.0**	18.2	**8.3**	4.4	**8.5±5.1**
	EER	9.0	8.0	**5.9**	9.9	14.3	3.7	4.8	**19.3**	2.0	**9.2**	18.9	**8.5**	4.7	**9.1±5.4**

Table 2: Comparison to face anti-spoofing methods on the cross-type testing task for domain generalization.

Method	HTER(%)	AUC(%)
Bayesian Neural Network	20.00	88.67
MC Dropout	15.00	92.50
Deep Ensemble	**5.83**	95.75
Ours	9.17	**96.92**

Table 3: Comparison to uncertainty estimation methods.

known-attack correlations become the only basis for classification. Compared to the alternative methods in Table 1, our FGHV framework has overwhelming performance improvement in two cross-dataset settings and comparable accuracy in the left two settings. Although NAS-FAS (Yu et al. 2020d) got a substantial increase in effect by searching a much stronger backbone than DepthNet, our framework is able to enhance the effects of cross-dataset scenarios only with the aid of the three constraints, which indicates better generalization.

It is worth noting that some ideas between our framework and SSDG (Jia et al. 2020) are quite similar. Firstly, the asymmetric triplet loss in SSDG and our DDC loss are both aimed at optimizing distances among features. In fact, SSDG only constrained input-face features whose quantity is restricted by the size of datasets. Instead, we choose to restrain considerable generated hypotheses so that the distribution of real faces and that of non-real faces will be accurately constructed and separated. Secondly, the feature generator in SSDG is actually a feature extractor and the discriminator makes extracted features domain-agnostic, while our feature generation networks are real generators that are optimized by FHVM and GHVM.

Cross-type testing. Strictly following the cross-type testing protocol (13 attacks leave-one-out) on SiW-M, we select out one attack type as the unknown testing type and treat the others as the known training types for each experiment. As for performance metrics, Average Classification Error Rate (ACER) and Equal Error Rate (EER) are utilized. Since ACER describes the practical performance un-

RCC	VAR	DDC	HTER(%)	AUC(%)
✓			14.16	91.12
	✓		33.33	64.50
		✓	19.12	86.83
✓	✓		10.00	93.92
✓		✓	10.83	93.50
	✓	✓	12.50	94.23
✓	✓	✓	**9.17**	**96.92**

Table 4: Comparison of different constraints.

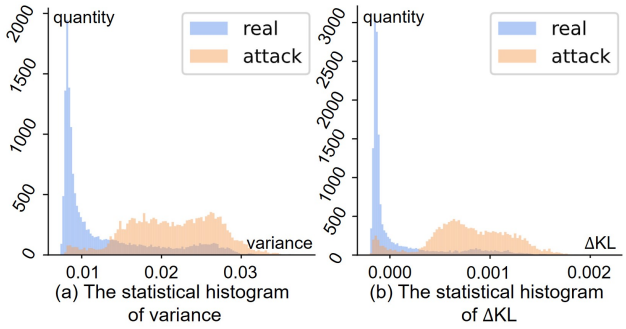


Figure 3: The quantity varies with variance and ΔKL.

der predeterminate thresholds, we additionally take advantage of the variance and ΔKL to assist classification. As shown in Table. 2, our framework can significantly improve the overall performance by 2.4% for ACER and 0.4% for EER. Meanwhile, the results are more stable. Surprisingly, our framework achieves a great promotion in defense for difficult obfuscation makeup attacks. Furthermore, it is worth noting that the variance and ΔKL indeed contribute to detecting attacks (ACERs are overall lower than EERs) so they are quite practical in actual usage. Consequently, excellent reliability is guaranteed.

Comparison to uncertainty estimation methods. Since the FHVM plays a similar role in estimating epistemic uncertainty, we compare our framework with three representative uncertainty estimation methods, i.e., Bayesian Neural Network (Kendall and Gal 2017), MC Dropout (Gal and Ghahramani 2016) and Deep Ensemble (Lakshminarayanan et al. 2017). These methods are reproduced on O&C&I to M setting and the comparison results are shown in Table 3. It has been proved that the strong prior assumption (i.e., normal distribution) in Bayesian Neural Network limits the ability of models. As for MC Dropout, it is too simple to bring exciting improvement. Moreover, Deep Ensemble outperforms in HTER but underperforms in AUC, which means that our method and Deep Ensemble are comparable in terms of overall performance. However, Deep Ensemble constructs the distribution for the whole convolution and fully-connected layers, which explains why it is more time-consuming and memory-consuming.

Method	HTER(%)	AUC(%)
No FGN	25.00	83.08
Only RF FGN	15.83	86.67
Only KA FGN	20.00	90.67
Both FGNs	**9.17**	**96.92**

Table 5: The impacts of both feature generation networks.

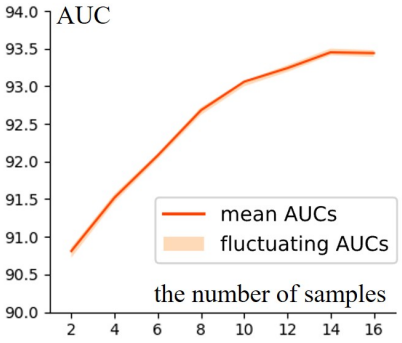


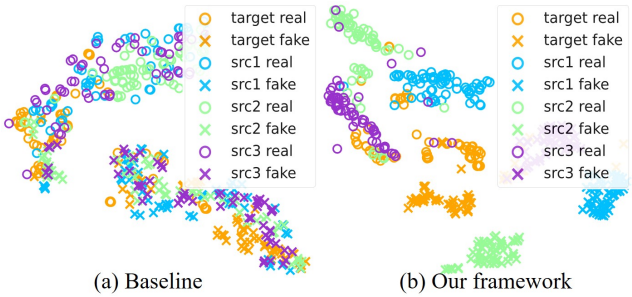
Figure 4: The impact of the number of samples.

4.3 Ablation Study

What is the effect of each constraint? In an effort to explore the effects of three constraints in our framework, we do ablation studies on O&C&I to M setting. For each experiment, we select different constraints to optimize the same framework and then use HTER and AUC for evaluation. As indicated in Table 4, the framework with single RCC constraint can be regarded as the baseline because the RCC is similar with cross entropy loss. The result of single VAR constraint is unsatisfactory on account of not constructing real-face distributions. But if the VAR constraint is accompanied with the RCC, the effect will be improved a lot. Additionally, the usage of the DDC makes the real-face distribution more discriminative so that the final effect reaches the current best. In summary, the three constraints are indispensable and boost the effectiveness from different aspects.

Is the FHVM discriminative? By reason that the FHVM utilizes the consistency of correlations to estimate to what extent the input face belongs to the real-face feature space, we analyze how the quantities of real faces and attacks vary with the variance of correlations on I&C&M to O setting. As shown in Fig. 3(a), the variances of correlations between real-face inputs and real-face hypotheses are much lower than those between attack inputs and real-face hypotheses. The results point out that the uncertainty estimation is indeed carried out and the predictions of real-face inputs are more certain. Hence, the discrimination of the FHVM is proved.

Is the GHVM discriminative? In an attempt to probe the discrimination of the GHVM, we utilize Gradient Decent strategy to find the corresponding latent vectors of the input face on I&C&M to O setting as explained in Sec. 3.4. The number of iterations M is 15 and the step length α is



(a) Baseline (b) Our framework

Figure 5: The t-SNE visualization.

1. After acquiring $\Delta KL = KL^{(15)} - KL^{(0)}$, we create a histogram to reveal how the quantity varies with ΔKL. As shown in Fig. 3(b), it is easy to find that ΔKLs of real faces are generally small while ΔKLs of fake faces are usually large, which is consistent with our expectation. And if the threshold is chosen appropriately, the separation of real faces and attacks can be achieved successfully. Therefore, the results demonstrate that the GHVM is discriminative.

Is it necessary to introduce two feature generation networks? Aiming to explore the necessity of both real-face and known-attack feature generation networks, we launch a probe into the impacts of both feature generation networks (FGN) on O&C&I to M setting. The structure without any FGN (No FGN) is the same with the conventional binary classification network. The structure equipped with known-attack FGN (Only KA FGN) can utilize RCC constraint, while the structure equipped with real-face FGN (Only RF FGN) can benefit from both RCC and VAR constraints. And all three constraints cannot be used until two FGNs (Both FGNs) are introduced at the same time. The results shown in Table 5 prove that two feature generation networks accompanied with three constraints can bring the greatest improvement. Thus, the necessity is self-evident.

Can more sampled latent vectors bring better performance or stability? To answer this question, we change the number of sampled latent vectors to repeat training and testing on I&C&M to O setting for several times (i.e., twenty times). For performance assessment, we average the twenty AUCs on the testset. For stability assessment, we utilize the maximum and the minimum of the twenty AUCs to reflect the fluctuation of effects. As depicted in Fig. 4, with the number of samples increasing, the effects are enhanced a lot. Furthermore, the training process and evaluation results become more stable. Taken time consumption and accuracy into account, the optimal number of samples is 14. Not only is it possible for us to compromise accuracy and speed, but applications can benefit from the reproducible experiments and the robust models.

Will the feature extraction backbone be improved coincidentally? Although we are mainly committed to improving the feature generation networks and attach less significance to the feature extraction backbone, we intend to validate whether the feature extraction backbone is also improved coincidentally. To confirm it, we visualize the features of input faces via t-SNE (Van der Maaten and Hinton 2008), which is depicted in Fig. 5. We conduct the ablation study on I&C&M to O setting and treat the model which is optimized only with cross entropy loss as the baseline method. According to the visualization, the baseline method classifies real faces and attacks well just in the source domains but poorly in the target domain. On the contrary, for our approach, even though attacks are not gathered together, the real-face regions are well constructed, which makes it easier to distinguish real faces and non-real faces. That's to say, our feature extraction backbone is promoted passingly.

5 Conclusion

In this paper, we propose a feature generation and hypothesis verification framework for FAS. Firstly, for the purpose of generalization, we regard the FAS task as the classification of real faces and non-real faces. Then, two feature generation networks are devised for the first time and two hypothesis verification modules are designed to estimate to what extent the input face belongs to the feature space and the distribution of real faces. Finally, we analyze the framework from the viewpoint of Bayesian uncertainty estimation and demonstrate the reliability of the framework. Qualitative and quantitative analyses show our framework outperforms the state-of-the-art approaches.

References

Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. 2015. Weight uncertainty in neural network. In *International Conference on Machine Learning*, 1613–1622.

Boulkenafet, Z.; Komulainen, J.; Hadid, A.; et al. 2015. Face anti-spoofing based on color texture analysis. In *2015 IEEE international conference on image processing (ICIP)*, 2636–2640. IEEE.

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 *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)*, 612–618. IEEE.

Chen, S.; Yao, T.; Zhang, K.; Chen, Y.; Sun, K.; Ding, S.; Li, J.; Huang, F.; and Ji, R. 2021a. A Dual-stream Framework for 3D Mask Face Presentation Attack Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 834–841.

Chen, Z.; Yao, T.; Sheng, K.; Ding, S.; Tai, Y.; Li, J.; Huang, F.; and Jin, X. 2021b. Generalizable Representation Learning for Mixture Domain Face Anti-Spoofing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1132–1139.

Chingovska, I.; Anjos, A.; Marcel, S.; et al. 2012. On the effectiveness of local binary patterns in face anti-spoofing. In *2012 BIOSIG-proceedings of the international conference of biometrics special interest group (BIOSIG)*, 1–7. IEEE.

de Freitas Pereira, T.; Anjos, A.; De Martino, J. M.; and Marcel, S. 2012. LBP-TOP based countermeasure against face spoofing attacks. In *Asian Conference on Computer Vision*, 121–132. Springer.

de Freitas Pereira, T.; Anjos, A.; De Martino, J. M.; and Marcel, S. 2013. Can face anti-spoofing countermeasures work in a real world scenario? In *2013 international conference on biometrics (ICB)*, 1–8. IEEE.

Denker, J. S.; and LeCun, Y. 1990. Transforming neural-net output levels to probability distributions. In *Proceedings of the 3rd International Conference on Neural Information Processing Systems*, 853–859.

Di Biase, G.; Blum, H.; Siegwart, R.; and Cadena, C. 2021. Pixel-wise Anomaly Detection in Complex Driving Scenes. In *CVPR*, 16918–16927.

Farquhar, S.; Osborne, M. A.; Gal, Y.; et al. 2020. Radial Bayesian neural networks: beyond discrete support in large-scale Bayesian deep learning. In *International Conference on Artificial Intelligence and Statistics*, 1352–1362. PMLR.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Graves, A. 2011. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24.

Hernandez-Lobato, J.; Li, Y.; Rowland, M.; Bui, T.; Hernández-Lobato, D.; and Turner, R. 2016. Black-box alpha divergence minimization. In *International Conference on Machine Learning*, 1511–1520. PMLR.

Hu, C.; Zhang, K.-Y.; Yao, T.; Ding, S.; Li, J.; Huang, F.; and Ma, L. 2021. An End-to-end Efficient Framework for Remote Physiological Signal Sensing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2378–2384.

Jia, Y.; Zhang, J.; Shan, S.; and Chen, X. 2020. Single-side domain generalization for face anti-spoofing. In *CVPR*, 8484–8493.

Kendall, A.; and Gal, Y. 2017. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*.

Kim, Y. E.; and Lee, S.-W. 2021. Domain Generalization with Pseudo-Domain Label for Face Anti-Spoofing. *arXiv preprint arXiv:2107.06552*.

Komulainen, J.; Hadid, A.; Pietikäinen, M.; et al. 2013. Context based face anti-spoofing. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 1–8. IEEE.

Kosiorek, A. R.; Kim, H.; Posner, I.; and Teh, Y. W. 2018. Sequential attend, infer, repeat: Generative modelling of moving objects. *arXiv preprint arXiv:1806.01794*.

Lakshminarayanan, B.; Pritzel, A.; Blundell, C.; et al. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *Advances in Neural Information Processing Systems*, 30.

Li, B.; Sun, Z.; Li, Q.; Wu, Y.; and Hu, A. 2019a. Group-wise deep object co-segmentation with co-attention recurrent neural network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8519–8528.

Li, B.; Sun, Z.; Tang, L.; Sun, Y.; and Shi, J. 2019b. Detecting Robust Co-Saliency with Recurrent Co-Attention Neural Network. In *IJCAI*, volume 2, 6.

Li, H.; Li, W.; Cao, H.; Wang, S.; Huang, F.; and Kot, A. C. 2018a. Unsupervised domain adaptation for face anti-spoofing. *IEEE Transactions on Information Forensics and Security*, 13(7): 1794–1809.

Li, H.; Pan, S. J.; Wang, S.; and Kot, A. C. 2018b. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5400–5409.

Li, X.; Komulainen, J.; Zhao, G.; Yuen, P.-C.; and Pietikäinen, M. 2016. Generalized face anti-spoofing by detecting pulse from face videos. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, 4244–4249. IEEE.

Li, X.; and She, J. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 305–314.

Liu, S.; Zhang, K.-Y.; Yao, T.; Bi, M.; Ding, S.; Li, J.; Huang, F.; and Ma, L. 2021a. Adaptive normalized representation learning for generalizable face anti-spoofing. *arXiv preprint arXiv:2108.02667*.

Liu, S.; Zhang, K.-Y.; Yao, T.; Sheng, K.; Ding, S.; Tai, Y.; Li, J.; Xie, Y.; and Ma, L. 2021b. Dual reweighting domain generalization for face presentation attack detection. *arXiv preprint arXiv:2106.16128*.

Liu, Y.; Jourabloo, A.; Liu, X.; et al. 2018. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 389–398.

Liu, Y.; Stehouwer, J.; Jourabloo, A.; and Liu, X. 2019. Deep tree learning for zero-shot face anti-spoofing. In *CVPR*, 4680–4689.

Liu, Y.; Stehouwer, J.; Liu, X.; et al. 2020. On disentangling spoof trace for generic face anti-spoofing. In *European Conference on Computer Vision*, 406–422. Springer.

Määttä, J.; Hadid, A.; Pietikäinen, M.; et al. 2011. Face spoofing detection from single images using micro-texture analysis. In *2011 international joint conference on Biometrics (IJCB)*, 1–7. IEEE.

MacKay, D. J. 1992. A practical Bayesian framework for backpropagation networks. *Neural computation*, 4(3): 448–472.

Pan, G.; Sun, L.; Wu, Z.; and Lao, S. 2007. Eyeblink-based anti-spoofing in face recognition from a generic webcamera. In *2007 IEEE 11th international conference on computer vision*, 1–8. IEEE.

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

Qin, Y.; Yu, Z.; Yan, L.; Wang, Z.; Zhao, C.; and Lei, Z. 2021. Meta-teacher for Face Anti-Spoofing. *IEEE TPAMI*.

Qin, Y.; Zhao, C.; Zhu, X.; Wang, Z.; Yu, Z.; Fu, T.; Zhou, F.; Shi, J.; and Lei, Z. 2020. Learning meta model for zero- and few-shot face anti-spoofing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 11916–11923.

Schlegl, T.; Seeböck, P.; Waldstein, S. M.; Schmidt-Erfurth, U.; and Langs, G. 2017. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, 146–157. Springer.

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.; Yuen, P. C.; et al. 2020. Regularized fine-grained meta face anti-spoofing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 11974–11981.

Shridhar, K.; Laumann, F.; Liwicki, M.; et al. 2018a. Uncertainty estimations by softplus normalization in bayesian convolutional neural networks with variational inference. *arXiv preprint arXiv:1806.05978*.

Shridhar, K.; Laumann, F.; Maurin, A. L.; Olsen, M.; and Liwicki, M. 2018b. Bayesian convolutional neural networks with variational inference. *arXiv preprint arXiv:1806.05978*.

Stehouwer, J.; Jourabloo, A.; Liu, Y.; and Liu, X. 2020. Noise modeling, synthesis and classification for generic object anti-spoofing. In *CVPR*, 7294–7303.

Tang, L.; Li, B.; Zhong, Y.; Ding, S.; and Song, M. 2021. Disentangled high quality salient object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3580–3590.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Wang, G.; Han, H.; Shan, S.; and Chen, X. 2019a. Improving cross-database face presentation attack detection via adversarial domain adaptation. In *2019 International Conference on Biometrics (ICB)*, 1–8. IEEE.

Wang, H.; Mao, C.; He, H.; Zhao, M.; Jaakkola, T. S.; and Katabi, D. 2019b. Bidirectional inference networks: A class of deep bayesian networks for health profiling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 766–773.

Wang, J.; Zhang, J.; Bian, Y.; Cai, Y.; Wang, C.; and Pu, S. 2021. Self-Domain Adaptation for Face Anti-Spoofing. *arXiv preprint arXiv:2102.12129*.

Wang, L.; Ding, X.; Fang, C.; et al. 2009. Face live detection method based on physiological motion analysis. *Tsinghua Science & Technology*, 14(6): 685–690.

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

Yang, B.; Zhang, J.; Yin, Z.; and Shao, J. 2021. Few-Shot Domain Expansion for Face Anti-Spoofing. *arXiv preprint arXiv:2106.14162*.

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

Yang, J.; Lei, Z.; Liao, S.; and Li, S. Z. 2013. Face liveness detection with component dependent descriptor. In *2013 International Conference on Biometrics (ICB)*, 1–6. IEEE.

Yu, Z.; Li, X.; Niu, X.; Shi, J.; and Zhao, G. 2020a. Face anti-spoofing with human material perception. In *European Conference on Computer Vision*, 557–575. Springer.

Yu, Z.; Li, X.; Shi, J.; Xia, Z.; and Zhao, G. 2021a. Revisiting Pixel-Wise Supervision for Face Anti-Spoofing. *IEEE TBIOM*.

Yu, Z.; Li, X.; Wang, P.; and Zhao, G. 2021b. Transrppg: Remote photoplethysmography transformer for 3d mask face presentation attack detection. *IEEE Signal Processing Letters*.

Yu, Z.; Qin, Y.; Li, X.; Wang, Z.; Zhao, C.; Lei, Z.; and Zhao, G. 2020b. Multi-Modal Face Anti-Spoofing Based on Central Difference Networks. In *CVPR Workshops*, 650–651.

Yu, Z.; Qin, Y.; Xu, X.; Zhao, C.; Wang, Z.; Lei, Z.; and Zhao, G. 2020c. Auto-Fas: Searching Lightweight Networks for Face Anti-Spoofing. In *ICASSP*, 996–1000. IEEE.

Yu, Z.; Qin, Y.; Zhao, H.; Li, X.; and Zhao, G. 2021c. Dual-Cross Central Difference Network for Face Anti-Spoofing. *arXiv preprint arXiv:2105.01290*.

Yu, Z.; Wan, J.; Qin, Y.; Li, X.; Li, S. Z.; and Zhao, G. 2020d. NAS-FAS: Static-Dynamic Central Difference Network Search for Face Anti-Spoofing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yu, Z.; Zhao, C.; Wang, Z.; Qin, Y.; Su, Z.; Li, X.; Zhou, F.; and Zhao, G. 2020e. Searching central difference convolutional networks for face anti-spoofing. In *CVPR*, 5295–5305.

Zhang, J.; Tai, Y.; Yao, T.; Meng, J.; Ding, S.; Wang, C.; Li, J.; Huang, F.; and Ji, R. 2021a. Aurora Guard: Reliable Face Anti-Spoofing via Mobile Lighting System. *arXiv preprint arXiv:2102.00713*.

Zhang, K.-Y.; Yao, T.; Zhang, J.; Liu, S.; Yin, B.; Ding, S.; and Li, J. 2021b. Structure Destruction and Content Combination for Face Anti-Spoofing. In *2021 IEEE International Joint Conference on Biometrics (IJCB)*, 1–6. IEEE.

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. In *European Conference on Computer Vision*, 641–657. Springer.

Zhang, Z.; Yan, J.; Liu, S.; Lei, Z.; Yi, D.; and Li, S. Z. 2012. A face antispoofing database with diverse attacks. In *2012 5th IAPR international conference on Biometrics (ICB)*, 26–31. IEEE.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232.