Face Anti-spoofing with Unknown Attacks: A Comprehensive Feature Extraction and Representation Perspective

Xu Wang[1], Pengkun Wang[1], Yudong Zhang[1], Binwu Wang[1]

[1]University of Science and Technology of China

Abstract Face anti-spoofing aims at detecting whether the input is a real photo of a user (living) or a fake (spoofing) image. As new types of attacks keep emerging, the detection of unknown attacks, known as Zero-Shot Face Anti-spoofing (ZSFA), has become increasingly important in both academia and industry. Existing ZSFA methods mainly focus on extracting discriminative features between spoofing and living faces. However, the nature of the spoofing faces is to trick anti-spoofing systems by mimicking the livings, so the deceptive features between the known attacks and the livings, which have been ignored by existing ZSFA methods, are essential to comprehensively represent the livings. Therefore, existing ZSFA models are incapable of learning the complete representations of living faces and thus falling short on effectively detecting newly-emerged attacks. To address this issue, we propose a novel feature extraction framework that can capture both the deceptive and discriminative features between living and existing spoofing faces. This framework is composed of a learnable masking mechanism and a two-against-all training scheme. To address the subsequent invalidation issue of categorical functions and dominance disequilibrium issue among different dimensions of features after importing deceptive features, we employ a newly modified semantic autoencoder to represent all extracted features to a semantic space to equilibrate the dominance of each feature dimension. As a result, our method simultaneously achieves a feasible detection on unknown attacks and a comparably accurate detection on known spoofing. Experimental results confirm the superiority and effectiveness of our proposed method in identifying the livings with the interference of both known and unknown spoofing types.

Key Words: Face antispoofing, Spoof detection, Deep learning

1 Introduction

Face anti-spoofing is becoming a popular method of authentication, with widespread use in mobile applications such as account login and unlocking cell phones, which has greatly enhanced the convenience of people's daily lives [1,2]. With the continuous increase of the accuracy and efficiency of face recognition, it has also been widely applied in online payment and banking, bringing safety and reliability issues. Face spoofing attack [3], which can usually be seen in financial crimes and cheats face recognition systems with fake faces such as photos, masks and videos, is one of the most severe safety threats to face recognition based authentications, and traditional face recognition technologies are incapable of distinguishing the authenticity of input faces [4, 5]. To this end, face anti-spoofing has been raised and extensively studied during recent years, which aims at detecting whether an input face is a fake image, e.g., a photo of one's printed photo, or a real photo of a user.

Early works on this field are mainly based on manual features [6–10] or deep features learned by neural networks [11–16]. Those methods have achieved promising performance in intra-domain experiments, i.e., the training sets cover all the spoofing types in the testing sets. However, their performances decrease severely on the zero-shot face anti-spoofing task, which is closer to real application scenarios as new spoofing types keep emerging. Several recent studies [11, 17–20] have made progress in tackling the problem of Zero-Shot Face Anti-spoofing (ZSFA). These studies have put forward carefully crafted deep-learning based model and effective learning strategies to extract discriminative features that are existentially significant differences between spoofing and living faces.

Although these features are effective, they are primarily based on known types of spoofing in the labeled

dataset. However, it is uncertain whether these features can be generalized to unknown types of spoofing. For example, if a new attack type is substantially different from the known attack types and does not possess these distinguishing features, the model may mistakenly classify it as a genuine face. Furthermore, a living face that is not present in the training data may exhibit these distinctive features and be incorrectly identified as an attack.

Considering the nature of the spoofing faces is to trick anti-spoofing systems by mimicking the livings, similar features between living faces and one category of fake ones, may be valuable for detecting other types of attacks and may be crucial in accurately representing livings. In this paper, we define features as ***deceptive features***, which are specific to certain types of attacks and are shared with the live entity, excluding other types of attacks. Deceptive features are considered as useless for detecting fake facial inputs and their positive roles have been ignored in traditional methods.For instance, as illustrated in Figure 1, depth information, which belongs to deceptive features between living faces and masks, is useful for distinguishing print photos from living ones. Regarding existing ZSFA-targeted methods, they may naturally miss some key features which can be used to well and roundly represent living samples and fall short in detecting unknown attacks. To this end, we can rethink the ZSFA task from a new perspective that a new type of spoofing faces, which can successfully deceive existing anti-spoofing systems, must be imitating the livings in terms of the discriminative features among the livings and all known types of spoofing faces. Therefore, deceptive features between living faces and all known categories of fake inputs must be the essential and key ingredients for detecting unknown types of spoofing.

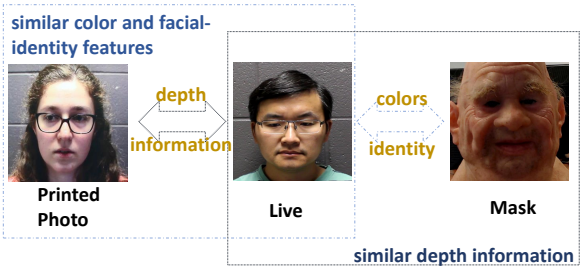

Fig. 1. **Impacts of deceptive features on anti-spoofing.** Printed photos have similar colors and facial identity features as the living ones while masks have similar depth information to the livings. Colors and facial identity features can be utilized to detect masks and depth information can be used to detect printed photos, meaning that the deceptive features between a spoofing type and the livings can be discriminative for other spoofing types.

In this paper, we integrate such deceptive features to provide model more insights, which are apparently excluded in previous works. We decouple the feature space into two orthogonal types of features: deceptive features and discriminative features. Deceptive features are exclusive features shared between a specific attack and living faces, which can help the model detect other types of attacks. Thus, this can boost the generalization of the model against unknown attacks. On the other hand, by cross-checking these features, the recognition of easily confused living faces would be beneficial. Discriminative features can be further used to identify livings and attacks. These two features work synergistically to improve the accuracy of the model.

In this paper, we propose a novel anti-spoofing framework which can achieve a feasible detection on unknown categories of facial spoofing and a comparable accurate detection on known categories spoofing. Specifically, to extract uniquely deceptive features between each spoofing and living faces, which can be used to detecting other categories of spoofing including unknown ones, we design a novel two-against-all training scheme, and this strategy uses a newly designed set of learnable mask module to masked partial features of all spoofing and the living, minimizing the diversity

between it and the living and simultaneously maximizing the diversity between other spoofing types and the combined set of it and the living.Meanwhile, the import of deceptive features in detecting spoofing faces may bring the invalidation issue of categorical functions, and distance-based metric, which can naturally address the invalidation issue may cause serious disequilibrium of dominance among different features. To address these subsequent issues of employing deceptive features in anti-spoofing, in this paper, we apply a modified semantic auto-encoder [21] to represent all extracted features to a semantic space where each dimension has almost equal dominance for distinguish spoofing, hence a feasible detection on unknown categories of spoofing and accurate detection on known categories of spoofing.

The main contributions can be summarized as follows.

- To the best of our knowledge, for the first time, in this paper, we reveal the fact deceptive features between known spoofing and living faces are key and essential for detecting unknown spoofing, and take an initial step on simultaneously detecting both unknown and known spoofing by concerning both deceptive and discriminative features between living and spoofing samples with one integrated network.
- To extract effective deceptive features, we propose a novel two-against-all training scheme to achieve high-efficient and variable-length filtration of deceptive features, and propose a novel idea of employing a modified semantic auto-encoder to equilibrate the dominance among different features, hence the detection on both unknown and known spoofing.
- We evaluate our proposed approach on the dataset of SiW-M for ZSFA scenario, and extensive experiments demonstrate that, in detecting unknown spoofing, our framework can at least gain a 5% improvement in terms of ACER while comparing with the advanced ZSFA solutions. Meanwhile, in detecting known spoofing, our method have a practical performance of 96.6% in terms of AUC, which is comparative with alternative anti-spoofing solutions.

The remainder of this paper is organized as follows. We introduce the existing studies on anti-spoofing and review methodology limitations in Section 2. Next, we describe the details of our proposed model in Section 3. Section 4 uses multiple datasets to evaluate the proposed model, which mainly includes two parts: the detection accuracy and the contribution measure of each component. Finally, we make a conclusion for this paper in Section 5.

2 Related Works

Great efforts have been achieved in the field of anti-spoofing. Most previous works, which can be divided into two sorts: manual feature based methods [6–10] and deep feature based methods [11–16, 22, 23], regard this issue as a classification problem.

Early manual feature based methods [6–10] distinguish living faces and spoofing inputs by exploiting specific handcrafted features with traditional image processing methods. Specifically, [8] extracts color textures to detect attacks by integrating the luminance and the chrominance in HSV (Hue, Saturation, Value) space. [6] first abstracts and aggregates four different features including specular reflection, blurriness, chromatic moment, and color diversity, and uses SVM to achieve dichotomies. Based on the analysis of living face inputs, [9] exploits and utilizes the Local Binary Pattern (LBP) features to detect fake ones from inputs. [7] carries out the detection based on both the

multi-level LBP features in HSV space and the Local Phase Quantization (LPQ) features in YCbCr space. And [10] senses print and replay attacks by analyzing the distortions of both color and shape of the input images. These traditional manual feature based methods, which have outstanding performances on some specific data sets, is of insufficient generalization ability generally, and [24] has indicated that the performances of this kind of approaches are limited in dealing with 3D face mask attacks.

Recently, deep feature based methods [11–16] are proposed to address the issue of face anti-spoofing by exploiting deep features with deep learning technologies. In particular, [13] first designs a deep Convolutional Neural Network (CNN) to estimate the depth map and rPPG signals, and fuses them to execute an end-to-end detection. [11,12] aim at improving the generalization abilities of proposed models by regarding the face anti-spoofing problem as an anomaly detection mission. [14] first considers spoofing images as noise-distorted living images and abstracts the noises with a deep neural network, and subsequently makes classification decisions based on learned noise pattern features. [15] extracts local and global features based on randomly collecting patches within face regions and the depth maps of entire input faces respectively, and fuses these two results to achieve accurate anti-spoofing. [16] considers both local features and additional optical-flow-based motion cues to improve the accuracy of face anti-spoofing. Also, these methods, which aim at effectively learning the feature combination patterns of attacks, are focusing on specific known types of attacks. Regarding unknown types of attacks, the performances of such technologies are limited.

All previous methods aim at learning specific features from labeled dataset, and use the learned patterns to detect attacks. Without exception, these methods take the awareness of the characteristics of attacks as an essential ingredient, and cannot be directly used to address the challenge of unknown spoofing detection.

Moreover, addressing the challenge of Zero-Shot Anti-Spoofing (ZSFA), some latest works propose some well-designed deep models and learning strategies which aim at learning generalized face anti-spoofing models. Specifically, [11, 17] aim at addressing ZSFA by representing known living samples with carefully and manually designed features, and [18] distinguishes living faces from fake ones by using a tree CNN to confirm living samples. More recently, [19] trains a meta-learner to learn the discrimination for detecting new spoofing category, from the support set where contains predefined living and spoofing faces and a few or none data of the new living and spoofing categories. [20] introduces feature generation networks for producing hypotheses for the first time and proposes a deep learning framework for building generalized face anti-spoofing model. [25] applies patch-wise data augmentation and proposed DC-DCN model which consists of horizontal/vertical and diagonal sparse convolution C-CDC. Nevertheless, all these ZSFA-targeted methods have never considered the similar features between known spoofing and living samples, so they may miss some key features which can be used to well and roundly represent living samples and fall short in detecting unknown attacks.

3 Methodology

Our approach consists of three sub-parts: feature extraction, semantic representation for extracted features, and spoofing detection. In the feature extraction part, we use CNN-based model as the backbone with two well-designed classifiers to extract deceptive and discriminative features, respectively. To accurately extract deceptive features, we design a training strategy, namely two-against-all training scheme. Further, we

employ an improved semantic autoencoder to represent all extracted features into a robust semantic representation space where each dimension has almost equal advantages to distinguish spoofing. F inally, distance-based classification a lgorithm i s a pplied t o d etect the spoofing faces.

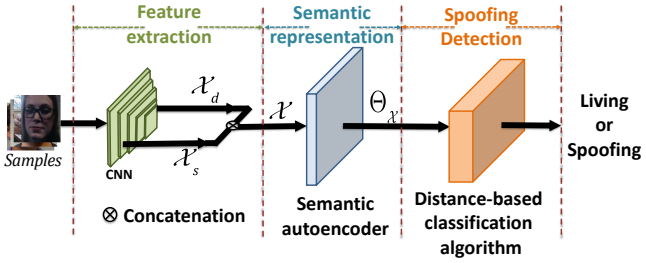

Fig.2. **Solution Overview.** In feature extraction, the deceptive and discriminative features are extracted based on the supervision of our purpose-designed loss functions. Then, extracted features are represented in a learnable semantic space to balance the importance among different dimensions of feature vectors. Finally, a distance-based classification algorithm is applied to detect the spoofing faces.

3.1 Feature extraction of facial images

Previous methods on face anti-spoofing mostly focus on mining and exploiting discriminative features between living samples and known spoofing samples. Considering the fact that all attacks, including known and unknown ones, are to cheat anti-spoofing models by imitating some features of living face images, we divide all features $\mathcal{X}$ obtained from living samples into two categories: 1) the features that are discriminative between living samples and all known spoofing samples, i.e., $\mathcal{X}_d$; 2) the features that are deceptive between living samples and some types of known spoofing samples, i.e., $\mathcal{X}_s$. Further, assuming there are m known categories of attacks, we have $\mathcal{X} = \{\mathcal{X}^0, \mathcal{X}^1, \cdots, \mathcal{X}^m\}$ where $\mathcal{X}^i (1 \leq i \leq m)$ indicates the features of the i-th category of attacks and $\mathcal{X}^0$ corresponds to the features of the living samples. Then given $0 \leq i \leq m$, we have $\mathcal{X}^i = \mathcal{X}_s^i \oplus \mathcal{X}_d^i$, here $\oplus$ means concatenation, $\mathcal{X}_s^i$ and $\mathcal{X}_d^i$ respectively indicate the deceptive and discriminative features between the $i-$th category of known spoofing and the livings. To supervise feature extraction, as shown in Figure 3, we introduce two classifiers to respectively extract discriminative and deceptive features. In the subsequent subsections, we describe the detailed design of these two classifiers. Notice that we here employ ResNet50 [26] in our experiments as the CNN backbone for feature extraction, as shown in Figure 3.

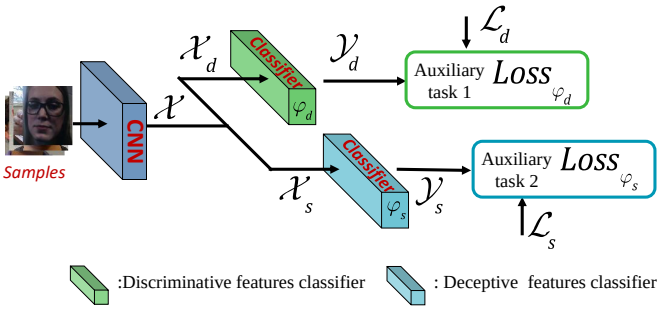

Fig.3. **Multi-task learning for feature extraction.** we employ the CNN backbone to extract deceptive and discriminative features with a same dimensionality. The extraction of deceptive features is based on a learnable masking mechanism and a two-against-all training scheme, and the extraction of discriminative features is supervised by a binary classifier.

Classifier for discriminative features: Since the discriminative features $\mathcal{X}_d$ are diverse between living samples and all known spoofing samples, they can be mapped into two different categories by a binary classifier, i.e.,

$$\mathcal{Y}_d^i \leftarrow \varphi_d \left[\mathcal{X}_d^i\right] \ \ where\ i \in \{0, 1, \cdots, m\} \tag{1}$$

where φ_d denotes the binary classification network with parameter ϖ_d, and $\mathcal{Y}_d^i$ is the classification result of the $i-$th category of known attacks (here 0 indicates the living). The cross entropy loss is employed for training the binary classification network, i.e.,

$$Loss_{\varphi_d} = \sum_{i=1}^{m} \left[-\mathcal{L}_d^i \cdot \log \mathcal{Y}_d^i - (1 - \mathcal{L}_d^i) \log(1 - \mathcal{Y}_d^i)\right] \tag{2}$$

where $\mathcal{L}_d^i$ is the label with regard to the output of the binary classifier φ_d with the input of the $i-$th category

known spoofing samples. It is equal to 0 for any known spoofing sample categories and 1 for living samples.

Classifier for deceptive features: There exist similarities between the livings and every known spoofing category, i.e.,

$$\forall i \in [1, \cdots, m], \ \ \exists\, \psi_i, \ \psi_i \cdot \mathcal{X}_s^0 \sim \psi_i \cdot \mathcal{X}_s^i \qquad (3)$$

where $\sim$ means $\psi_i \cdot \mathcal{X}_s^0$ and $\psi_i \cdot \mathcal{X}_s^i$ follow a same distribution. ψ_i is a vector with the elements of 0 or 1 to extract the deceptive features between the $i-$th category of known spoofing and the livings. Further, here $\cdot$ corresponds the element-wise multiplication. To extract the deceptive features between all m categories of known spoofing and the livings, we employ m-way two-against-all binary classifiers, i.e., $\{\varphi_s^1, \cdots, \varphi_s^m\}$. As demonstrated in Figure 4 (b), the $i-$th category of spoofing and the livings are classified into one category, which is different to other spoofing types, by the $i-$th classifier φ_s^i. Therefore, the output of the $i-$th two-against-all binary classifier φ_s^i for the $j-$th category of spoofing can be calculated as,

$$\mathcal{Y}_s^j(i) \leftarrow \varphi_s^i \left[\psi_i \cdot \mathcal{X}_s^j\right] \ where \ j \in \{1, \cdots, m\} \qquad (4)$$

Here $\mathcal{Y}_s^j(i) = 1$ if and only if $j = i$. Note that we have $\mathcal{Y}_s^0(i) = 1$ when the input is a living face. For the sake of efficiency, we here employ a single FC layer, the two-against-all classification of φ_s^i with regard to all m categories of known spoofing and the livings can be rewritten as,

$$\mathcal{Y}_s^j(i) = \text{Sigmoid}\left(w_s^i \otimes (\psi_i \cdot \mathcal{X}_s^j) + b_s^i\right) \qquad (5)$$

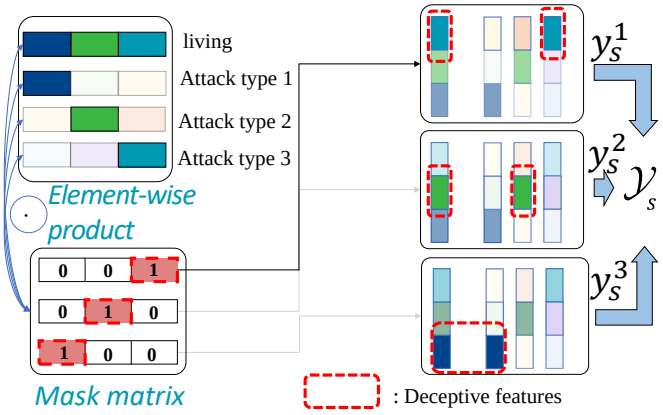

Fig.4. Two-against-all training scheme means, regarding one specific spoofing type, we first locate the deceptive features between it and the living by masking partial features of all spoofing and the living in a learnable manner to minimize the diversity between it and the living and simultaneously maximize the diversity between other spoofing types and the combined set of it and the living.

Note here we have $j \in \{0, \cdots, m\}$, $j = 0$ indicates the livings. And $\otimes$ means the vector multiplication. The m-way two-against-all classifications can be formulated by,

$$\begin{aligned} &\mathcal{Y}_s^j = \text{Sigmoid}\left(\mathcal{W}_s \cdot \Psi_s \otimes \mathcal{X}_s^j + B_s\right) \\ &where \begin{cases} \mathcal{W}_s = \left[\ (\varpi_s^1)^T \ (\varpi_s^2)^T \ \cdots \ (\varpi_s^m)^T \ \right]^T \\ \Psi_s = \left[\ \psi_1 \ \psi_2 \ \cdots \ \psi_m \ \right]^T \\ B_s = \left[\ b_s^1 \ b_s^2 \ \cdots \ b_s^m \ \right]^T \end{cases} \end{aligned} \qquad (6)$$

$\mathcal{Y}_s^j$ corresponds to the output of all m classifiers, regarding the inputs of the $j-$th categories of known spoofing. And we have $\mathcal{Y}_s^j = \{\mathcal{Y}_s^j(1), \cdots, \mathcal{Y}_s^j(m)\}$. For the deceptive features of the livings $\mathcal{X}_s^0$, we have

$$\mathcal{L}_s^0 = [\overbrace{1 \ \cdots \ 1}^{m}]^T \qquad (7)$$

where $\mathcal{L}_s^0$ corresponds to the label with regard to the output of all m two-against-all classifiers. Regarding the input of the $i-$th category of spoofing, $\mathcal{L}_s^i$ should have the $i-$th element of 1 and the other elements of 0, i.e.,

$$\mathcal{L}_s^i = \{\overbrace{0 \ \cdots \ 0}^{i-1} \ 1 \ \overbrace{0 \ \cdots \ 0}^{m-i}\} \qquad (8)$$

So far, the problem of training the m-way two-against-all classifiers can be transferred to the optimization of the learnable parameter matrix $\mathcal{W}_s \cdot \Psi_s$. This matrix is rather sparse due to the sparsity of Ψ_s. We can simplify the problem to employ the L1-regularization on matrix $\mathcal{W}_s$ [27] to replace $\mathcal{W}_s \cdot \Psi_s$. The loss for training the m-way classification neural network can be defined by,

$$Loss_{\varphi_s} = \sum_{i=0}^{m} \left\| \mathcal{Y}_s^i - \mathcal{L}_s^i \right\|_2^2 + \lambda_1 \|\mathcal{W}_s\|_1 \tag{9}$$

where λ_1 is a hyper-parameter to adjust the weight of the corresponding component.

Overall loss for feature extraction: As mentioned, we obtain the two kinds of features of living samples by employing a CNN backbone, and train two kinds of classifiers to check the validity of extracted features. The feature extraction network can be viewed as the combination of the CNN backbone and the two kinds of classifiers, and for training the integrated feature extraction network, we combine the losses of the two kinds of classifiers so that the CNN backbone can extract both the two categories of features. The overall loss for training the feature extraction network can be formulated by,

$$Loss_{feature} = Loss_{\varphi_d} + \lambda_2 Loss_{\varphi_s} \tag{10}$$

λ_2 is a hyper-parameter to tune the weight of the corresponding component. Further, as mentioned in [28], the inter-class variation of a specific category of features may be large, leading to a greater inter-class variation in subsequent semantic representations. We hence modified the loss function for training the feature extraction network as,

$$Loss^*_{feature} = Loss_{\varphi_d} + \lambda_2 Loss_{\varphi_s} + \lambda_3 \left(\sum_{i=0}^{m} \left\| \mathcal{X}^i - \varepsilon^i \right\|_2^2 \right) \tag{11}$$

where ε^i is randomly initiated, and should be subsequently updated by the learnable centers of the i category of known spoofing [28]. Notice that in case $i = 0$, the parameter of ε^i should be updated by the learnable center of the livings. Also, λ_3 is a hyper-parameter to tune the weights.

Visualization of feature space: To demonstrate the effectiveness of our proposed feature extraction network, regarding all extracted 1920-dimensional features of all samples including both deceptive and discriminative features, we transform them into a three-dimensional space via the t-SNE [29], and the samples of a specific spoofing category should be concentrated in the transferred three-dimensional space. Notice that during training the feature extraction network, we assume the attack category "Makeup_Co" is unknown. As illustrated in Figure 5, known categories of samples are nicely clustered by separated categories, and this verifies that the extracted deceptive and discriminative features can be further used to distinguish known spoofing. However, the unknown samples are relatively scattered in this figure, and this indicates that the import of deceptive features may bring distractions to traditional anti-spoofing classification function, and they cannot be directly used to address the detection issue of both known and unknown spoofing. Further, note that in Figure 5, we cannot use distance based clustering algorithms to directly distinguish known and unknown attacks either, due to the bar-like inner-class distributions of samples and the possible intersections among the bar-like distributions of different categories of spoofing. To this end, we still need to seek a better semantic representation of all extracted features to equilibrate the weights of different feature dimensions.

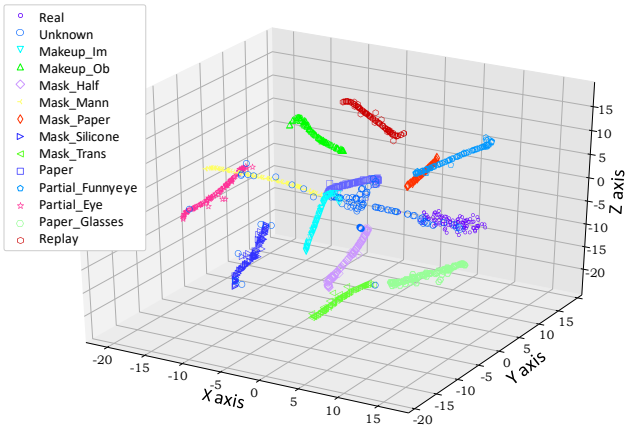

Fig.5. t-SNE Visualization of the Feature Space

3.2 Semantic representation of extracted features

As discussed above, we should represent all features to a semantic space where the intra-class semantic distances are minimized while the inter-class semantic distances are maximized. Further, to well represent the livings, we require that each dimension in the target semantic space should be a combination of owned features of the livings. To balance the disequilibrium among different features, we normalize the value of each dimension in the target semantic space to be $[0, 1]$. Each dimension of the semantic representation of a living sample should be 1. Regarding a spoofing sample, the value of each dimension of its semantic representation, which corresponds to the similarity between this sample and the livings in terms of the corresponding feature combination, should be a real number within $[0, 1]$. The projection from extracted features $\mathcal{X}$ to the semantic space can be written as,

$$\begin{aligned} &\Theta_{\mathcal{X}} \leftarrow \mathcal{W}_\Theta \otimes \mathcal{X} \\ &\text{s.t.} \begin{cases} \mathcal{W}_\Theta \otimes \mathcal{X}^0 = [\quad 1 \cdots 1\,] \\ \mathcal{W}_\Theta \otimes \mathcal{X}^i = \left[\, \theta_i^1 \ \cdots \ \theta_i^{|\Theta_x|} \,\right] \\ \theta_i^j \in [0,1] \end{cases} \\ &where \quad \begin{cases} i \in \{1, \cdots, m\} \\ j \in \{1, \cdots, |\Theta_x|\} \end{cases} \end{aligned} \tag{12}$$

Where $\Theta_{\mathcal{X}}$ denotes the corresponding semantic representation of feature $\mathcal{X}$. The matrix $\mathcal{W}_\Theta$ means the projection between the semantic and feature spaces. Note that $\mathcal{X}^0$ indicates the extracted features of a living sample, and $\mathcal{X}^i$ corresponds to the extracted features of a sample within the ith category of spoofing.

Another requirement of the projection from feature space to semantic space is that the information loss between the original extracted features and their corresponding semantic representations should be minimized, and this loss minimization problem can be converted to a reconstruction problem from the semantic space to the feature space, i.e.,

$$\arg\min_{\mathcal{W}_\Theta} \left\| \mathcal{X} - \mathcal{W}'_\Theta \otimes \Theta_{\mathcal{X}} \right\|_2^2 = \arg\min_{\mathcal{W}_\Theta} \left\| \mathcal{X} - \mathcal{W}'_\Theta \otimes \mathcal{W}_\Theta \otimes \mathcal{X} \right\|_2^2 \tag{13}$$

$\mathcal{W}'_\Theta$ denotes the backward projection from the semantic space to the feature space. Referring to [21,30], the backward projection $\mathcal{W}'_\Theta$ can be simplified as $\mathcal{W}_\Theta^T$ with negligible losses. The problem can be written as,

$$\arg\min_{\mathcal{W}_\Theta} \left\| \mathcal{X} - \mathcal{W}_\Theta^T \otimes \mathcal{W}_\Theta \otimes \mathcal{X} \right\|_2^2 + \lambda_4 (1 - \mathcal{Y}) \left\| \mathcal{W}_\Theta \otimes \mathcal{X} \right\|_1 \tag{14}$$

Notice here $\mathcal{Y} = 1$ if $\mathcal{X}$ is the features of a living face, otherwise $\mathcal{Y} = 0$. To equilibrate the weights of different feature dimensions, a center loss item is introduced, i.e.,

$$
\begin{aligned}
Loss = & \left\| \mathcal{X} - \mathcal{W}_{\Theta}^{T} \otimes \mathcal{W}_{\Theta} \otimes \mathcal{X} \right\|_2^2 \\
& + \lambda_4 (1 - \mathcal{Y}) \left\| \mathcal{W}_{\Theta} \otimes \mathcal{X} \right\|_1 \\
& + \lambda_5 \sum_{i=0}^{m} \left\| \mathcal{W}_{\Theta} \otimes \mathcal{X}^i - \mathcal{W}_{\Theta} \otimes \varepsilon^i \right\|_2^2
\end{aligned} \tag{15}
$$

Here λ_4 and λ_5 are also hyper-parameters to tune the weight of the corresponding components.

3.3 Face detection with trained model

Regarding the extracted features $\mathcal{X}$ of a specific sample , we can obtain its semantic representation $\Theta_{\mathcal{X}}$, and calculate the distances between $\Theta_{\mathcal{X}}$ and $\mathcal{W}_{\Theta} \otimes \varepsilon^i$ for each category of samples. Notice that for $1 \leq i \leq m$, $\mathcal{W}_{\Theta} \otimes \varepsilon^i$ is the center of the i-th category of spoofing faces in the semantic space, and for $i = 0$, $\mathcal{W}_{\Theta} \otimes \varepsilon^i$ corresponds to the center of the livings. After calculating the Euclidean distance within the semantic space, we use it to determine whether a sample is living or not, i.e., a sample is considered as the living if and only if the distance between $\Theta_{\mathcal{X}}$ and $\mathcal{W}_{\Theta} \otimes \varepsilon^0$ is smallest among all calculated distances.

4 Experiments

In this section, We evaluate the proposed model on multiple data sets. And focus on the following potential questions.

- **Q1.** Compared with the most advanced methods, how accurate is the proposed method under various scenarios. Please refer to Section 4.2.
- **Q2.** Does each insight/component proposed contribute to the performance of the model. Please refer to Section 4.3.
- **Q3.** How does the number of known types affect the performance of the model. Please refer to Section 4.4.

4.1 Experimental setups

Databases: Five databases are used to evaluate the proposed approach, including SiW-M [18], OULU-NPU [32], Replay-Attack [4], MSU-MFSD [6] and CASIA [33]. SiW-M contains rich spoofing types and is designed for zero-shot face anti-spoofing task. OULU-NPU is a high-resolution database, and provides four protocols for traditional intra-domain experiments. Additionally, following the protocol proposed in [11], we apply cross-domain testing on Replay-Attack, MSU-MFSD and CASIA.

Quality measurements: In Oulu dataset, all methods are evaluated with the following widely accepted metrics: 1) Attack Presentation Classification Error Rate (APCER) [34], which indicates the ratio of the amount of false livings to the amount of spoofing; 2) Bona fide Presentation Classification Error Rate (BPCER) [34], which corresponds to the ratio of the amount of false spoofing to the amount of livings; 3) Average Classification Error Rate (ACER) [34], which equals to the average of APCER and BPCER. In SiW-M dataset, Equal Error Rate (EER) and ACER are employed for evaluation as early works do. For cross-dataset testing, we apply Area Under the ROC Curve (AUC) to evaluate all the methods.

Other setups: We extract faces from videos by utilizing the face coordinates given by the datasets themselves [1], and then resize all extracted faces into 224*224 resolution. Our proposed model take single images as inputs and employ ResNet50 as backbone. Feature extraction network and semantic representation network are trained separately. Besides, the optimizer of Adam [36] is used to train our model with learning rate as 0.001. The initial values of the hyper-parameters are given and fine-tuned according to the dimensions of

[1] For databases which do not provide face coordinates, we extract face coordinates by [35] as Replay-Attack does.

Table 1. ZSFA performances of different models on SiW-M.

Method	Metrics(%)	Replay	Print	Mask Attacks				Makeup Attacks				Partial Attacks			Average
				Half	Silicone	Trans.	Paper	Manne.	Obfusc.	Imperson.	Cosmetic	Funny Eye	Glasses	Partial	
Auxiliary [13]	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 [18]	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 [31]	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
DC-CDN [25]	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
FGHV [20]	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
Ours	ACER	**4.2**	**2.9**	5.3	7.7	**12.1**	**1.9**	**1.6**	17.1	**1.5**	**0.9**	18.8	**7.3**	**1.2**	**6.3±6.1**
	EER	**4.7**	**2.6**	7.1	7.8	**11.2**	**2.1**	1.9	**18.6**	**1.3**	**1.1**	19.0	**6.8**	**0.8**	**6.5±6.3**

the vectors they control, e.g., $Loss_{\varphi_d}$ and $Loss_{\varphi_s}$ have the same dimensions, thus the initial value of λ_2 is 1. Analogously,in our experiments, the initial values of the parameters of λ_1,λ_2,λ_3, λ_4, and λ_5 are fine-tuned and finally set to 0.001, 1, 0.001, 0.01 and 1, respectively. All hyper-parameters are tuned with grid search.

4.2 Main Experiments (Q.1)

We compare the detection accuracy of the proposed model and SOTA models in three scenarios. First We evaluated the performance of each model against the ZSFA task, which mainly depended on the model's ability to detect unknown attacks. Second, we evaluate the model's performance against hybrid attacks through a series of experiments on cross-data datasets, which mix known and unknown attack. Finally, we evaluate models for traditional anti-spoofing task, covering only the types of seen attacks.

4.2.1 Evaluation on SiW-M for ZSFA testing

We train our model on SiW-M [18] in a leave-one-out testing manner as it suggests, which means each time we split one kind of spoofing images and 20% of the living images as the testing set, and train our model with the rest. To evaluate the performance of our proposed model on ZSFA, we compare it with five State-Of-The-Art (SOTA) ZSFA methods, Auxiliary [13], DTN [18], SpoofTrace [31], DC-CDN [25] and FGHV [20]. The results are demonstrated in Table 1. The great variances between accuracy of models on different spoofing types indicate that diverse spoofing types differ significantly. Thus, detecting unknown spoofing type based on known types is a difficult and challenging problem. According to Table 1, we can find our method outperforms other alternative methods in terms of both EER and ACER in nearly half of all spoofing types. In particular, our approach achieves an overall 25.8% optimization in terms of EER and 28.6% optimization in terms of ACER. Althogh the performance of our model decreases in some cases, the absolute accuracy of our model in such cases is acceptable and is competitive to that of baselines. In the worst case, the performance of our model is lower than the best baseline with a minor margin less than 1%. This experiments verify the superiority of our proposed method in address the ZSFA issue.

4.2.2 Evaluation for cross-dataset testing

To further evaluate the generalization ability of our proposed method, by following the protocol proposed by [11], we conduct a series of cross-dataset evaluations on three alternative datasets, including CASIA, MSU-MFSD and Replay-Attack. Based on the protocol, the performances of all methods are reported with another widely used metric of Area Under the ROC Curve (AUC). Each time, one spoofing type of

Table 2. AUC (%) of cross-dataset anti-spoofing on CASIA, Replay, and MSU-MFSD.

Method	CASIA			Replay			MSU-MFSD			Average
	Video	Cut Photo	Wrapped Photo	Video	Digital Photo	Printed Photo	Printed Photo	HR Video	Mobile Video	
Auxiliary [13]	94.16	88.39	79.85	99.75	95.17	78.86	50.57	99.93	93.54	86.7±15.6
DTN [18]	90	97.3	97.5	99.9	99.9	99.6	81.6	99.9	97.5	95.9±6.2
SpoofTrace [31]	93.6	99.7	99.1	99.8	99.9	99.8	76.3	99.9	99.1	96.4±7.8
DC-CDN [25]	98.5	99.9	99.8	**100**	99.43	99.9	70.8	100	99.9	96.5±9.6
FGHV [20]	**98.6**	99.8	**99.9**	99.9	99.1	99.8	73.2	100	99.9	96.7±8.8
Ours	**98.8**	**99.9**	99.8	**100**	**100**	**99.9**	**86.6**	**100**	**100**	**98.3±4.2**

the three datasets are selected for testing, and other types for training. Due to the overlap of types in the three datasets, the experiments are usually applied for evaluating the performance of models when fed into images collected in different places by different devices. The results are reported in Table 2. As observed, our proposed method can outperform other alternative approaches in most scenarios. The better performance of our model indicates that when faced with spoofing faces in diverse environments, the proposed model achieves more robust anti-spoofing accuracy. And this can further confirm the superiority of our proposed method in terms of the generalization ability.

4.2.3 Evaluation for intra-dataset testing

Although our model are proposed for ZSFA tasks, we evaluate the performance of our model in traditional anti-spoofing task on OULU-NPU. This series of experiments strictly follow the four protocols that OULU-NPU suggests. As in Table 3, the performances of our model are competitive with or better than the performances of SOTA solutions in traditional anti-spoofing. This result indicates that introducing deceptive features will not lead to performance decrease on known spoofing faces. This suggests that our model can also handle the traditional anti-spoofing task as SOTA works.

Table 3. The results of traditional anti-spoofing on four protocols of OULUNPU.

Protocol	Method	APCER	BPCER	ACER
1	Auxiliary [13]	1.6	1.6	1.6
	DTN [18]	1.3	1.5	1.4
	SpoofTrace [31]	0.8	1.3	1.1
	DC-CDN [25]	0.5	0.3	0.4
	FGHV [20]	0.5	0.2	0.4
	Ours	**0.4**	**0.2**	**0.3**
2	Auxiliary [13]	2.7	2.7	2.7
	DTN [18]	2.3	2.0	2.2
	SpoofTrace [31]	2.3	1.6	1.9
	DC-CDN [25]	0.9	1.9	1.3
	FGHV [20]	0.8	1.6	1.2
	Ours	**0.8**	**1.6**	**1.2**
3	Auxiliary [13]	2.7±1.3	3.1±1.7	2.9±1.5
	DTN [18]	2.5±1.4	3.0±2.1	2.8±1.9
	SpoofTrace [31]	**1.9±1.6**	4.0±5.4	2.8±3.3
	DC-CDN [25]	2.2±2.8	1.6±2.1	1.9±1.1
	FGHV [20]	2.1±1.9	1.6±2.4	**1.8±2.1**
	Ours	2.0±1.5	**1.5±2.5**	1.8±2.0
4	Auxiliary [13]	9.3±5.6	10.4±6.0	9.5±6.0
	DTN [18]	8.6±4.3	8.0±5.4	8.3±4.8
	SpoofTrace [31]	3.3±3.6	5.2±5.4	3.8±4.2
	DC-CDN [25]	5.4±3.3	2.5±4.2	4.0±3.1
	FGHV [20]	4.6±2.8	3.4±5.3	4.0±4.0
	Ours	**3.1±2.8**	**3.3±4.0**	**3.2±3.4**

Summary. Based on the analysis conducted above, it can be concluded that the proposed model demonstrates competitive performance in both traditional anti-spoofing tasks and more challenging ZSFA tasks. This highlights the versatility of our model in the face anti-spoofing domain.

4.3 Ablation Study (Q.2)

In this section, we evaluate the effects of each individual component through a series of ablation experiments. It is important to note that all ablation experi-

ments conducted in the ZSFA scenario. Siw-M dataset is employed here for ablation study.

4.3.1 Impacts of deceptive features

In the proposed model, we divide all features into two categories: discriminative and deceptive features. To investigate the impacts of deceptive features on detection, we carry out a series of ZSFA experiments by ablatively taking one category of discriminative and deceptive features away at each round of evaluations, and the results are given in Table 4, as we can see, each individual category of features can effectively help detect unknown category of spoofing faces, and the performances of our approach in case of employing only one category of features are almost equivalent. And the employment of these two categories of features can significantly enhance the performances of our approach in terms of all the metrics of APCER, BPCER, and ACER. This verifies that the employment of deceptive features is effective on detecting unknown spoofing faces.

Table 4. Impacts of different kinds of features

Features	APCER(%)	BPCER(%)	ACER(%)
deceptive Only	28.7±21.8	3.14±3.22	14.3±12.1
Discriminative Only	20.4±15.2	6.44±3.72	15.7±18.1
Two Kinds	**11.9±10.2**	**2.74±1.4**	**7.3±5.2**

4.3.2 Impacts of semantic representation

The dimensionality of semantic space determines the representation ability of semantic space, i.e., a larger dimensionality can help maintain more information from feature space, hence minimizing the information loss. However, when the feature dimension is None, this means that the semantic representation of this component is invalid.

We then set the dimensionality of semantic space to $1920, 1024, 512, 256$, and None, the results are shown in Table 5. Note that in case that the dimensionality is set to None, it means that we use the feature space directly to detect unknown spoofing. As shown in this table, the increasing of the dimensionality of semantic space can positively enhance the performances of our proposal, and this verifies our previous assumptions. Notice that even though the dimensionality of semantic space is set to 1920, the computational complexity of the transformation process is 10^3 times less than the computational complexity of the CNN backbone. Thus, in our final implementation, the dimensionality is 1920.

Table 5. Impacts of dimensionality in semantic space

Dimension	APCER(%)	BPCER(%)	ACER(%)
None	48.2±17.5	34.1±11.6	41.1±14.3
256	39.0±19.6	11.8±3.6	25.4±10.6
512	28.3±18.2	4.78±9.3	16.5±12.1
1024	24.6±21.3	5.32±2.7	15.9±10.4
1920	**11.9±10.2**	**2.74±1.4**	**7.3±5.2**

4.4 Hyperparameter experiment (Q.3)

Given the fact that our proposal can construct better descriptions for the livings by considering the deceptive features between living and known spoofing, *the impacts of the number of known spoofing types* should be investigated. For the sake of fairness, in each round of the experiments, we fix a specific category of spoofing faces as the unknown spoofing type, i.e., Replay. And the size of training sets are fixed by selecting the same number of samples from known attacks for fair comparison, no matter how many categories of known spoofing types are included. Table 6 shows the results. The performance of our approach deteriorates dramatically as the number of known types decrease, and this indicates that a relatively number of known categories of spoofing is essential for extracting enough deceptive features to well represent the livings. Also, combined with Table 1, with respect to Replay attack, we notice that our method with only 8 known categories of

spoofing can achieve an equivalent level of performance to those SOTA solutions with 12 known categories of spoofing.

Table 6. Impacts of number of known types

Number of Known Types	12	11	8	5
ACPER(%)	**8.67**	9.42	10.86	40.62
BPCER(%)	**1.04**	1.72	5.91	38.14
ACER(%)	**4.81**	6.54	8.49	40.11

5 Conclusion

To accurately detect unknown types of spoofing remains challenging in the field of anti-spoofing. In this paper, we investigate the detection of unknown attacks generally, propose a novel approach to extract dominating features from both deceptive and discriminative perspectives between the living and known spoofing, devise a semantic autoencoder to represent all extracted features to a semantic space where each dimension has almost equal dominance for distinguish spoofing. Experimental results demonstrate the superiority of our proposed approach in the mission of detecting both known and unknown attacks. In future, we will further investigate the classification issue of unknown attacks while there exist multiple categories of unknown attacks, and this may benefit the whole anti-spoofing community.

References

[1] A. K. Jain and S. Z. Li, *Handbook of face recognition*, vol. 1. Springer, 2011.

[2] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM computing surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.

[3] J. Galbally, S. Marcel, and J. Fierrez, "Biometric antispoofing methods: A survey in face recognition," *IEEE Access*, vol. 2, pp. 1530–1552, 2014.

[4] I. Chingovska, A. Anjos, and S. Marcel, "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)*, pp. 1–7, IEEE, 2012.

[5] L. Best-Rowden, H. Han, C. Otto, B. F. Klare, and A. K. Jain, "Unconstrained face recognition: Identifying a person of interest from a media collection," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2144–2157, 2014.

[6] D. Wen, H. Han, and A. K. Jain, "Face spoof detection with image distortion analysis," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 746–761, 2015.

[7] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face spoofing detection using colour texture analysis," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1818–1830, 2016.

[8] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face anti-spoofing based on color texture analysis," in *2015 IEEE international conference on image processing (ICIP)*, pp. 2636–2640, IEEE, 2015.

[9] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *2011 international joint conference on Biometrics (IJCB)*, pp. 1–7, IEEE, 2011.

[10] K. Patel, H. Han, and A. K. Jain, "Secure face unlock: Spoof detection on smartphones," *IEEE*

transactions on information forensics and security, vol. 11, no. 10, pp. 2268–2283, 2016.

[11] S. R. Arashloo, J. Kittler, and W. Christmas, "An anomaly detection approach to face spoofing detection: A new formulation and evaluation protocol," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2017.

[12] O. Nikisins, A. Mohammadi, A. Anjos, and S. Marcel, "On effectiveness of anomaly detection approaches against unseen presentation attacks in face anti-spoofing," in *2018 International Conference on Biometrics (ICB)*, pp. 75–81, IEEE, 2018.

[13] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 389–398, 2018.

[14] A. Jourabloo, Y. Liu, and X. Liu, "Face despoofing: Anti-spoofing via noise modeling," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 290–306, 2018.

[15] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based cnns," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 319–328, IEEE, 2017.

[16] L. Feng, L.-M. Po, Y. Li, X. Xu, F. Yuan, T. C.-H. Cheung, and K.-W. Cheung, "Integration of image quality and motion cues for face anti-spoofing: A neural network approach," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 451–460, 2016.

[17] F. Xiong and W. AbdAlmageed, "Unknown presentation attack detection with face rgb images," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–9, IEEE, 2018.

[18] Y. Liu, J. Stehouwer, A. Jourabloo, and X. Liu, "Deep tree learning for zero-shot face anti-spoofing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4680–4689, 2019.

[19] Y. Qin, C. Zhao, X. Zhu, Z. Wang, Z. Yu, T. Fu, F. Zhou, J. Shi, and Z. Lei, "Learning meta model for zero- and few-shot face anti-spoofing," in *AAAI2020*, 2019.

[20] S. Liu, S. Lu, H. Xu, J. Yang, S. Ding, and L. Ma, "Feature generation and hypothesis verification for reliable face anti-spoofing," *arXiv preprint arXiv:2112.14894*, 2021.

[21] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3174–3183, 2017.

[22] C.-Y. Wang, Y.-D. Lu, S.-T. Yang, and S.-H. Lai, "Patchnet: A simple face anti-spoofing framework via fine-grained patch recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20281–20290, June 2022.

[23] Z. Wang, Z. Wang, Z. Yu, W. Deng, J. Li, T. Gao, and Z. Wang, "Domain generalization via shuffled style assembly for face anti-spoofing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4123–4133, June 2022.

[24] S. Liu, B. Yang, P. C. Yuen, and G. Zhao, "A 3d mask face anti-spoofing database with real world variations," in *Proceedings of the IEEE conference*

on computer vision and pattern recognition workshops, pp. 100–106, 2016.

[25] Z. Yu, Y. Qin, H. Zhao, X. Li, and G. Zhao, "Dual-cross central difference network for face anti-spoofing," *arXiv preprint arXiv:2105.01290*, 2021.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[27] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[28] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*, pp. 499–515, Springer, 2016.

[29] V. D. M. Laurens and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2605, pp. 2579–2605, 2008.

[30] M. Ranzato, Y.-L. Boureau, and Y. L. Cun, "Sparse feature learning for deep belief networks," in *Advances in neural information processing systems*, pp. 1185–1192, 2008.

[31] Y. Liu, J. Stehouwer, and X. Liu, "On disentangling spoof trace for generic face anti-spoofing," in *European Conference on Computer Vision*, pp. 406–422, Springer, 2020.

[32] Z. Boulkenafet, J. Komulainen, L. Lei, X. Feng, and A. Hadid, "Oulu-npu: A mobile face presentation attack database with real-world variations," in *IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, 2017.

[33] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Li, "A face antispoofing database with diverse attacks," pp. 26–31, 03 2012.

[34] I. J. S. . Biometrics, "Biometric presentation attack detection - part 1: Framework." ISO/IEC 30107-1:2016, 1 2016.

[35] B. Fröba and A. Ernst, "Face detection with the modified census transform," *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pp. 91–96, 2004.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.