Li LM, Wang BW, Wang X *et al*. Face anti-spoofing with unknown attacks: A comprehensivefeature extraction and representation perspective. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 39(4): 827-840 July 2024. DOI: 10.1007/s11390-024-4164-7

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

Li-Min Li[1] (李利民), Bin-Wu Wang[2, *] (王斌武), Xu Wang[1, 3] (王　旭)
Peng-Kun Wang[1, 3] (王鹏焜), Yu-Dong Zhang[2] (张玉东), *Student Member, IEEE*
and Yang Wang[1, 3, *] (汪　炀), *Senior Member, ACM, IEEE*

[1] *School of Software Engineering, University of Science and Technology of China, Hefei 230000, China*

[2] *School of Data Science, University of Science and Technology of China, Hefei 23000, China*

[3] *Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou 215000, China*

E-mail: lilimin@mail.ustc.edu.cn; wbw1995@mail.ustc.edu.cn; wx309@ustc.edu.cn; pengkun@ustc.edu.cn
zyd2020@mail.ustc.edu.cn; angyan@ustc.edu.cn

Received January 31, 2024; accepted March 29, 2024.

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, therefore 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 fall short of effectively detecting newly emerged attacks. To tackle this problem, we propose an innovative method that effectively captures both the deceptive and discriminative features distinguishing between genuine and spoofing faces. Our method consists of two main components: a two-against-all training strategy and a semantic autoencoder. The two-against-all training strategy is employed to separate deceptive and discriminative features. To address the subsequent invalidation issue of categorical functions and the dominance disequilibrium issue among different dimensions of features after importing deceptive features, we introduce a modified semantic autoencoder. This autoencoder is designed to map all extracted features to a semantic space, thereby achieving a balance in the dominance of each feature dimension. We combine our method with the feature extraction model ResNet50, and experimental results show that the trained ResNet50 model simultaneously achieves a feasible detection of unknown attacks and comparably accurate detection of known spoofing. Experimental results confirm the superiority and effectiveness of our proposed method in identifying the living with the interference of both known and unknown spoofing types.

Keywords face anti-spoofing, spoof detection, zero-shot learning, convolutional neural network, 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 in the accuracy and efficiency of face recogni-

Regular Paper

Special Section of CVM 2024

This paper was supported by the National Natural Science Foundation of China under Grant Nos. 62072427 and 12227901, the Project of Stable Support for Youth Team in Basic Research Field of Chinese Academy of Sciences under Grant No. YSBR-005, and the Academic Leaders Cultivation Program of University of Science and Technology of China.

[*] Corresponding Author (Bin-Wu Wang organized and supervised the entire writing process of this paper. Yang Wang is the leader of the funding projects.)

tion, it has also been widely applied in online payment and banking, bringing safety and reliability issues. The face spoofing attack[3] is usually seen in financial crimes, and cheats face recognition systems with fake faces such as photos, masks, and videos. It is one of the most severe safety threats to face recognition based authentications, while 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 work on this field is 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 (ZSFA) task, in which case models have to discriminate unknown types of spoofing faces. The scenario of zero-shot face anti-spoofing is closer to real application scenarios as new spoofing types keep emerging and one has no idea about newly-emerging spoofing types. Several recent studies[11, 17–20] have made progress in tackling the problem of ZSFA. These studies have put forward carefully crafted deep learning based models and effective learning strategies to extract discriminative features, i.e., existentially significant differences between spoofing and living faces. For instance, DTN[18] proposes a tree-like neural network to extract discriminative features hierarchically. DC-CDN[21] is a dual-cross central difference network with cross-feature interaction modules for dual-stream feature enhancement. AIM-FAS[22] learns the discriminative features to recognize new living and spoofing categories from predefined living and spoofing categories. Although existing work focuses on mining generalizable discriminative features, it is highly uncertain whether the extracted features can be generalized to discriminate unknown types of spoofing faces, since they are primarily based on known types of spoofing in the labeled dataset. 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.

Actually, in the zero-shot task, it is more important to effectively and fully represent a category rather than mining category-specific features. As an intuitive example, if we have tigers, pandas, and horses in our training set and we want to distinguish tigers from other animals, we may find stripes the most discriminative feature of tigers in the training set. However, if we have zebras in our testing set, we will find stripes useless to distinguish tigers from zebras. Similarly, for tackling ZSFA, only extracting discriminative features is not generalizable enough for distinguishing unknown spoofing faces. It is more important to effectively and comprehensively characterize the living faces so that any input that does not match the living face characterization can be considered spoofing faces. Considering the nature of the spoofing faces is to trick anti-spoofing systems by mimicking the living, 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 living. 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 useless for detecting fake facial inputs and their positive roles have been ignored in traditional methods. For instance, as illustrated in Fig.1, printed photos have similar colors and facial identity features to the living ones while masks have similar depth information to the living. 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 living can be discriminative for other spoofing types. Regarding existing ZSFA-targeted methods, they may naturally miss some key features that can be used to well and roundly represent living samples well. 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 imitate the living in terms of the discriminative features among the living and all known types of spoofing faces. Therefore, deceptive features between living faces and all known categories of fake inputs must be the essential ingredients for detecting unknown types of spoofing.

In this paper, we investigate the potential of using deceptive features to improve accuracy. 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

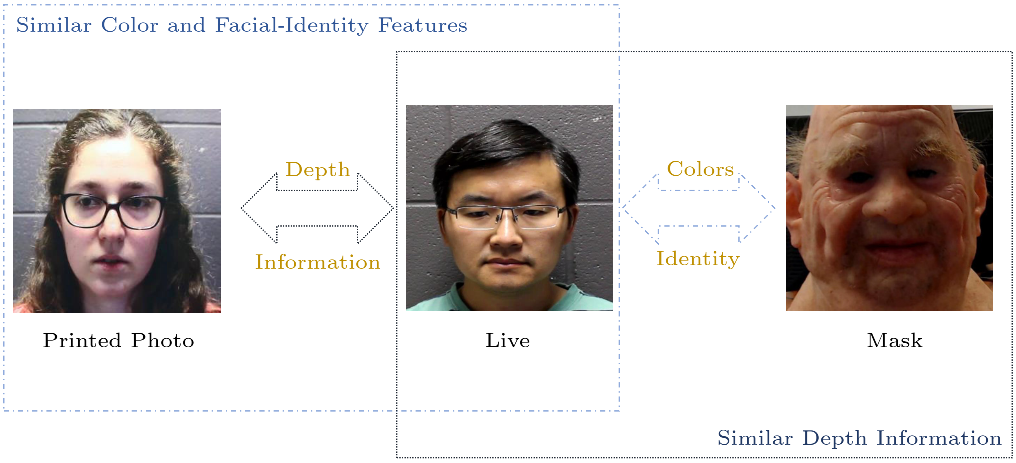

Similar Color and Facial-Identity Features
Depth
Information
Colors
Identity
Printed Photo
Live
Mask
Similar Depth Information

Fig.1. Impacts of deceptive features on anti-spoofing.

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.

We propose a novel method that can achieve a feasible detection of unknown categories of facial spoofing and a comparable accurate detection for known categories of spoofing. The uniquely deceptive features between each category of spoofing faces and living faces can be used to detect other spoofing categories, including unknown ones. To this point, we design a novel two-against-all training strategy. Specifically, a newly designed set of learnable mask modules is introduced to mask partial features of facial images. For each spoofing category, we combine the masked features of it with that of the living faces. The diversity between the combined features and the masked features of other categories of spoofing faces are maximized. Meanwhile, the import of deceptive features in detecting spoofing faces may bring the invalidation issue of categorical functions, and distance-based metrics, 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[23] 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.

- We reveal the fact that deceptive features between known spoofing and living faces are 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 strategy to achieve highly efficient and variable-length filtration of the deceptive feature, 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 method on the datasets of SiW-M for the ZSFA scenario, and extensive experiments demonstrate that, in detecting unknown spoofing, our method can gain up to a 5% improvement in terms of ACER while compared with the advanced ZSFA solutions. Meanwhile, in detecting known spoofing, our method has a practical performance of 98.3% in terms of AUC.

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 method in Section 3. Section 4 uses multiple datasets to evaluate the proposed method, 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 Work

Great efforts have been made in the field of anti-spoofing. Most previous work, which can be divided into two sorts: manual feature based methods[6–10] and deep feature based methods[11–16, 24, 25], regards 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 the HSV space and the local phase quantization (LPQ) features in the YCbCr space. And [10] senses print and replay attacks by analyzing the distortions of both the color and the shape of the input images. These traditional manual feature based methods, which have outstanding performances on some specific datasets, are of generally insufficient generalization ability, and [26] has indicated that the performances of this kind of approach are limited in dealing with 3D face mask attacks.

Recently, deep feature based methods[11–16] have been 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 the proposed models by regarding the face anti-spoofing problem as an anomaly detection mission. [14] first considers spoofing faces as noise-distorted living faces, and extracts 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, focus on specific known types of attacks. Regarding unknown types of attacks, the performance of such technologies is limited.

All previous methods aim at learning specific features from labeled datasets and using 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, to address the challenge of ZSFA, researchers proposed some well-designed deep models and learning strategies that 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] introduces feature generation networks for producing hypotheses for the first time and proposes a deep learning framework for building a generalized face anti-spoofing model. [21] applies patch-wise data augmentation and proposes the 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, therefore they may miss some key features that can be used to well and roundly represent living samples and fall short in detecting unknown attacks.

3 Methodology

In this section, we will describe our method in details. The main contributions of our method include a two-against-all training strategy for extracting deceptive and discriminative features and an autoencoder for learning a robust representation space.

3.1 Method Overview

As shown in Fig. 2, our method for face anti-spoofing consists of three subparts: feature extraction, semantic representation for extracted features, and spoofing detection. In the feature extraction part, we use a CNN-based feature extraction model ResNet50[27] as the backbone with two well-designed classifiers to extract deceptive and discriminative features, respectively. To accurately extract deceptive features, we

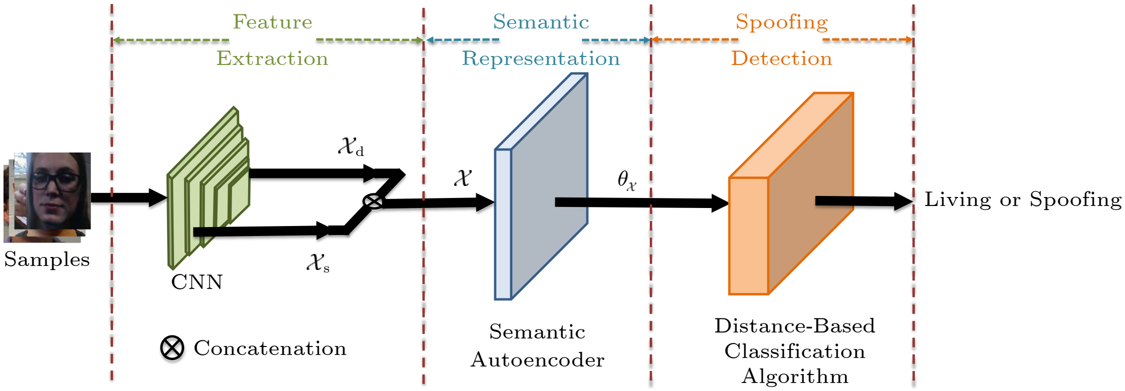


Fig.2. Solution overview.

design a two-against-all training strategy. 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. Finally, a distance-based classification algorithm is applied to detect the spoofing faces.

3.2 Feature Extraction of Facial Images

Previous methods of 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., $\mathcal{X}_\mathrm{d}$; 2) the features that are deceptive between living samples and some types of known spoofing samples, i.e., $\mathcal{X}_\mathrm{s}$. Further, assuming there are m known categories of attacks, we have $\mathcal{X}=(\mathcal{X}^0, \mathcal{X}^1,\dots,\mathcal{X}^m)$ where $\mathcal{X}^i(1\leqslant i\leqslant m)$ indicates the features of the i-th category of attacks and $\mathcal{X}^0$ corresponds to the features of the living samples. Then given $0\leqslant i\leqslant m$, we have $\mathcal{X}^i=\mathcal{X}^i_\mathrm{s}\oplus\mathcal{X}^i_\mathrm{d}$, where $\oplus$ means concatenation, $\mathcal{X}^i_\mathrm{s}$ and $\mathcal{X}^i_\mathrm{d}$ indicate the deceptive and discriminative features between the i-th category of known spoofing samples and the livings, respectively. To supervise feature extraction, as shown in Fig.3, we introduce two classifiers to extract discriminative and deceptive features, respectively. Next, we describe the detailed design of these two classifiers. Notice that we here employ ResNet50[27] in our experiments as the backbone for feature extraction, as shown in Fig.3.

3.2.1 *Classifier for Discriminative Features*

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

$$\mathcal{Y}^i_\mathrm{d}\leftarrow\varphi_\mathrm{d}\left[\mathcal{X}^i_\mathrm{d}\right], i\in\{0,1,\dots,m\},$$

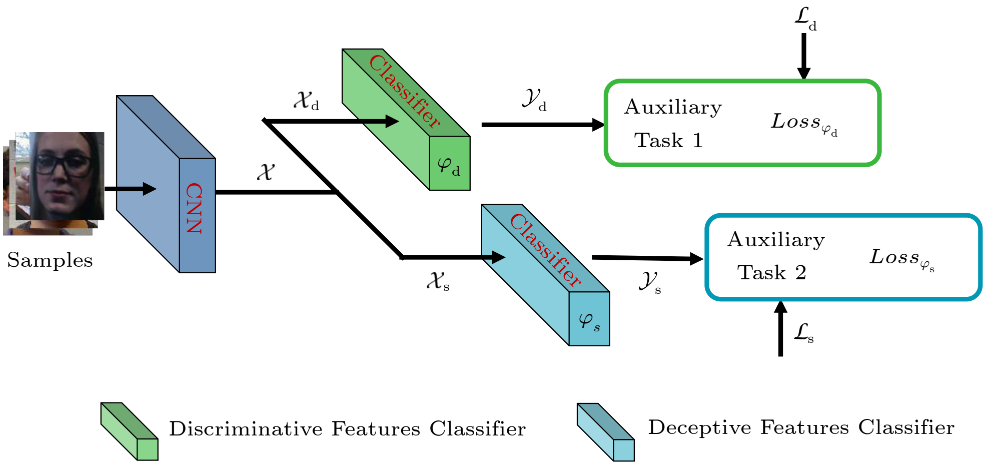


Fig.3. Multi-task learning for feature extraction.

where φ_{d} denotes the binary classification network with parameter ϖ_{d}, and $\mathcal{Y}_{\mathrm{d}}^{i}$ 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.,

$$Loss_{\varphi_{\mathrm{d}}} = \sum_{i=1}^{m} (-\mathcal{L}_{\mathrm{d}}^{i} \log \mathcal{Y}_{\mathrm{d}}^{i} - (1-\mathcal{L}_{\mathrm{d}}^{i}) \log(1-\mathcal{Y}_{\mathrm{d}}^{i})),$$

where $\mathcal{L}_{\mathrm{d}}^{i}$ 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.

3.2.2 Classifier for Deceptive Features

There exist similarities between the livings and every known spoofing category, i.e.,

$$\forall i \in [1,\ldots,m],\ \ \exists\, \boldsymbol{\psi}_i,\ \boldsymbol{\psi}_i \odot \mathcal{X}_{\mathrm{s}}^{0} \sim \boldsymbol{\psi}_i \odot \mathcal{X}_{\mathrm{s}}^{i},$$

where $\odot$ means the Hadamard product, and $\sim$ means $\boldsymbol{\psi}_i \odot \mathcal{X}_{\mathrm{s}}^{0}$ and $\boldsymbol{\psi}_i \odot \mathcal{X}_{\mathrm{s}}^{i}$ follow a same distribution. $\boldsymbol{\psi}_i$ is a vector with the elements of 0 or 1 to extract the deceptive features between the i-th category of known spoofing samples and the livings. To extract the deceptive features between all m categories of known spoofing samples and the livings, we propose a two-against-all training strategy, which employs m-way two-against-all binary classifiers, i.e., $\{\varphi_{\mathrm{s}}^{1},\ldots,\varphi_{\mathrm{s}}^{m}\}$. As demonstrated in Fig.4, regarding one specific spoofing type, we first locate the deceptive features between it and the living by masking partial features of all spoofing types 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. Specifically, the i-th category of spoofing and the livings are classified into one category, which is different from 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,

$$\mathcal{Y}_{\mathrm{s}}^{j}(i) \leftarrow \varphi_{\mathrm{s}}^{i}[\boldsymbol{\psi}_i \odot \mathcal{X}_{\mathrm{s}}^{j}], j \in \{1,\ldots,m\},$$

Here $\mathcal{Y}_{\mathrm{s}}^{j}(i)=1$ if and only if $j=i$. Note that we have $\mathcal{Y}_{\mathrm{s}}^{0}(i)=1$ 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,

$$\mathcal{Y}_{\mathrm{s}}^{j}(i) = \mathrm{Sigmoid}\left(\boldsymbol{w}_{\mathrm{s}}^{i} \otimes (\boldsymbol{\psi}_i \odot \mathcal{X}_{\mathrm{s}}^{j}) + \boldsymbol{b}_{\mathrm{s}}^{i}\right).$$

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

$$\begin{aligned}
\mathcal{Y}_{\mathrm{s}}^{j} &= \mathrm{Sigmoid}\left(\mathcal{W}_{\mathrm{s}} \odot \boldsymbol{\Psi}_{\mathrm{s}} \otimes \mathcal{X}_{\mathrm{s}}^{j} + \boldsymbol{B}_{\mathrm{s}}\right),\\
\mathcal{W}_{\mathrm{s}} &= \left((\boldsymbol{\varpi}_{\mathrm{s}}^{1})^{\mathrm{T}}\,(\boldsymbol{\varpi}_{\mathrm{s}}^{2})^{\mathrm{T}}\ldots(\boldsymbol{\varpi}_{\mathrm{s}}^{m})^{\mathrm{T}}\right)^{\mathrm{T}},\\
\boldsymbol{\Psi}_{\mathrm{s}} &= \begin{pmatrix} \boldsymbol{\psi}_1 & \boldsymbol{\psi}_2 & \ldots & \boldsymbol{\psi}_m \end{pmatrix}^{\mathrm{T}},\\
\boldsymbol{B}_{\mathrm{s}} &= \begin{pmatrix} \boldsymbol{b}_{\mathrm{s}}^{1} & \boldsymbol{b}_{\mathrm{s}}^{2} & \ldots & \boldsymbol{b}_{\mathrm{s}}^{m} \end{pmatrix}^{\mathrm{T}},
\end{aligned}$$

where $\mathcal{Y}_{\mathrm{s}}^{j}$ corresponds to the output of all m classifiers, regarding the inputs of the j-th categories of known spoofing. And we have $\mathcal{Y}_{\mathrm{s}}^{j} = (\mathcal{Y}_{\mathrm{s}}^{j}(1),\ldots,\mathcal{Y}_{\mathrm{s}}^{j}(m))$.

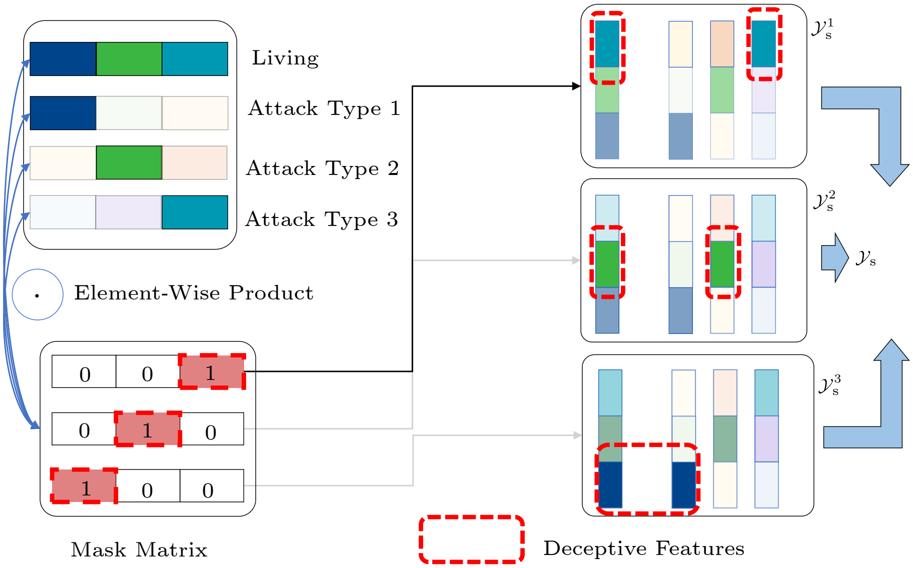

Living
Attack Type 1
Attack Type 2
Attack Type 3
Element-Wise Product
0 0 1
0 1 0
1 0 0
Mask Matrix
$\mathcal{Y}_{\mathrm{s}}^{1}$
$\mathcal{Y}_{\mathrm{s}}^{2}$
$\mathcal{Y}_{\mathrm{s}}^{3}$
$\mathcal{Y}_{\mathrm{s}}$
Deceptive Features

Fig.4. Two-against-all training strategy.

For the deceptive features of the livings $\mathcal{X}_{\rm s}^0$, we have

$$\mathcal{L}_{\rm s}^0 = (\overbrace{1\dots1}^{m})^{\rm T},$$

where $\mathcal{L}_{\rm s}^0$ 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, $\mathcal{L}_{\rm s}^i$ should have the i-th element of 1 and the other elements of 0, i.e.,

$$\mathcal{L}_{\rm s}^i = (\overbrace{0\dots0}^{i-1}\ 1\ \overbrace{0\dots0}^{m-i}).$$

So far, the problem of training the m-way two-against-all classifiers can be transferred to the optimization of the learnable parameter matrix $\mathcal{W}_{\rm s} \odot \boldsymbol{\Psi}_{\rm s}$. This matrix is rather sparse due to the sparsity of $\boldsymbol{\Psi}_{\rm s}$. We can simplify the problem to employ the L1-regularization on matrix $\mathcal{W}_{\rm s}$[28] to replace $\mathcal{W}_{\rm s} \odot \boldsymbol{\Psi}_{\rm s}$. The loss for training the m-way classification neural network can be defined by,

$$Loss_{\varphi_{\rm s}} = \sum_{i=0}^{m} \|\mathcal{Y}_{\rm s}^i - \mathcal{L}_{\rm s}^i\|_2^2 + \lambda_1 \|\mathcal{W}_{\rm s}\|_1,$$

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

3.2.3 Overall Loss for Feature Extraction

As mentioned, we obtain the two kinds of features of living samples by employing ResNet50 and training two kinds of classifiers to check the validity of extracted features. The feature extraction network can be viewed as the combination of ResNet50 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 ResNet50 can extract both the two categories of features. The overall loss for training the feature extraction network can be formulated by,

$$Loss_{\rm feature} = Loss_{\varphi_{\rm d}} + \lambda_2 Loss_{\varphi_{\rm s}},$$

where λ_2 is a hyper-parameter to tune the weight of the corresponding component. Further, as mentioned in [29], 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,

$$Loss_{\rm feature}^{*} = Loss_{\varphi_{\rm d}} + \lambda_2 Loss_{\varphi_{\rm s}} + \lambda_3 \left(\sum_{i=0}^{m} \|\mathcal{X}^i - \epsilon^i\|_2^2\right),$$

where ϵ^i is randomly initiated, and should be subsequently updated by the learnable centers of the i-th category of known spoofing[29]. 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.

3.2.4 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 (3D) space via the t-SNE[30], and the samples of a specific spoofing category should be concentrated in the transferred 3D space. Notice that during training the feature extraction network, we assume the attack category "Makeup_Co" is unknown. As illustrated in Fig.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 functions, and they cannot be directly used to address the detection issue of both known and unknown spoofing. Further, note that in Fig.5, we cannot use distance-based clustering algorithms to directly distinguish known and unknown attacks ei-

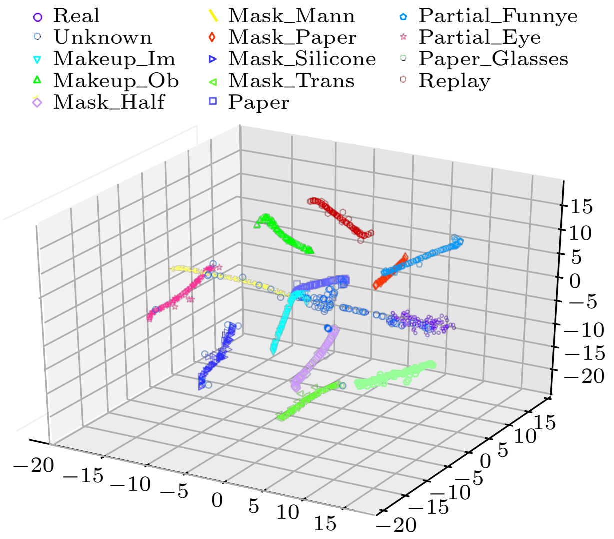


Fig.5. t-SNE visualization of the feature space.

ther, 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.

3.3 Semantic Representation of Extracted Features

As discussed above, we propose a semantic autoencoder to represent all features in a semantic space where the intra-class semantic distances are minimized while the inter-class semantic distances are maximized. Further, to well represent the living, we require that each dimension in the target semantic space should be a combination of owned features of the living. 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 living 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,

$$\boldsymbol{\Theta}_{\mathcal{X}} \leftarrow \mathcal{W}_{\Theta} \otimes \mathcal{X}$$

$$\text{s.t.} \begin{cases} \mathcal{W}_{\Theta} \otimes \mathcal{X}^{0} = (1 \ \ldots \ 1), \\ \mathcal{W}_{\Theta} \otimes \mathcal{X}^{i} = \left(\ \theta_i^1 \ldots \theta_i^{|\Theta_x|} \ \right), \\ \theta_i^j \in (0,1), \end{cases}$$

where $i \in \{1, \ldots, m\}$ and $i \in \{1, \ldots, m\}$. $\boldsymbol{\Theta}_{\mathcal{X}}$ 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 i-th category of spoofing.

Another requirement of the projection from the feature space to the semantic space is that the information loss between the originally 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.,

$$\arg\min_{\mathcal{W}_{\Theta}} \|\mathcal{X} - \mathcal{W}'_{\Theta} \otimes \mathcal{W}_{\Theta} \otimes \mathcal{X}\|_2^2,$$

where $\mathcal{W}'_{\Theta}$ denotes the backward projection from the semantic space to the feature space. Referring to [23, 31], the backward projection $\mathcal{W}'_{\Theta}$ can be simplified as $\mathcal{W}_{\Theta}^{\mathrm{T}}$ with negligible losses. The problem can be written as,

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

Notice here $\mathcal{Y} = 1$ if $\mathcal{X}$ is the features of a living face, otherwise $\mathcal{Y} = 0$. The second term is to maximize the distance between spoofing faces and living faces. To equilibrate the weights of different feature dimensions, a center loss item is introduced, i.e.,

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

where ϵ^i is the center of each category, and λ_4 and λ_5 are also hyper-parameters to tune the weight of the corresponding components.

3.4 Face Detection with Trained Model

As described in Subsection 3.1, we combine ResNet50[27] with our method as our face anti-spoof model where ResNet50 is used for feature extraction. The whole training process of our model is described in Algorithm 1. Here we introduce how to detect whether a face is spoofing or living using ResNet50 trained with our method. Regarding the extracted features $\mathcal{X}$ of a specific sample calculated by our CNN backbone, we can obtain its semantic representation $\Theta_{\mathcal{X}}$, and calculate the distances between $\Theta_{\mathcal{X}}$ and $\mathcal{W}_{\Theta} \otimes \epsilon^i$ for each category of samples. Notice that for $1 \leqslant i \leqslant m$, $\mathcal{W}_{\Theta} \otimes \epsilon^i$ is the center of the i-th category of spoofing faces in the semantic space, and for $i = 0$, $\mathcal{W}_{\Theta} \otimes \epsilon^i$ 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 living if and only if the distance between $\Theta_{\mathcal{X}}$ and $\mathcal{W}_{\Theta} \otimes \epsilon^0$ is the smallest among all calculated distances.

4 Experiments

In this section, we evaluate the proposed method on multiple datasets. And we focus on the following potential questions.

- $Q1$. Compared with the most advanced methods, how accurate is ResNet50 with our method under various scenarios (refer to Subsection 4.2)?

Algorithm 1. Training Process of ResNet50 with Our Method

```
Input: training data $[X, Y]$, CNN backbone $f_\theta : X \rightarrow [\mathcal{X}_{\rm d}, \mathcal{X}_{\rm s}]$ with parameter $\theta$, classifier for discriminative features $\phi_{\rm d}$ with parameter $\mathcal{W}_{\rm d}$, classifier for deceptive features $\boldsymbol{\Phi}_{\rm s} = \{\phi_{\rm s}^i | i \in [1, 2, \ldots, m]\}$ with parameter $\mathcal{W}_{\rm s}$, semantic projection $\mathcal{W}_\Theta$ and category centers $\{\epsilon_0, \epsilon_1, \ldots, \epsilon_m\}$
Output: trained ResNet50
Initialize all trainable parameters
for $[x, y]$ in $[X, Y]$ do
  //extract deceptive and discriminative features
  $\mathcal{X}_{\rm d}, \mathcal{X}_{\rm s} = f_\theta(x)$
  //calculate loss of discriminative features
  $Loss_{\phi_{\rm d}} = -y \log \phi_{\rm d}(\mathcal{X}_{\rm d}) - (1 - y) \log 1 - \phi_{\rm d}(\mathcal{X}_{\rm d})$
  //calculate loss of deceptive features
  Construct label $\mathcal{L}_{\rm s}(y)$ of $x$ for $\boldsymbol{\Phi}_{\rm s}$
  $Loss_{\boldsymbol{\Phi}_{\rm s}} = ||\boldsymbol{\Phi}_{\rm s} - \mathcal{L}_{\rm s}||_2^2 + \lambda_1 ||\mathcal{W}_{\rm s}||_1$
  //loss for features
  $\mathcal{X} = \mathcal{X}_{\rm d} \oplus \mathcal{X}_{\rm s}$
  $Loss_{\rm feature} = Loss_{\varphi_{\rm d}} + \lambda_2 Loss_{\varphi_{\rm s}} + \lambda_3 \left( \|\mathcal{X} - \epsilon^y\|_2^2 \right)$
  //calculate loss of semantic representation
  $Loss_r = \|\mathcal{X} - \mathcal{W}_\Theta^{\rm T} \otimes \mathcal{W}_\Theta \otimes \mathcal{X}\|_2^2$
  $Loss_c = \lambda_4 (1 - \dagger) \|\mathcal{W}_\Theta \otimes \mathcal{X}\|_1 + \lambda_5 \|\mathcal{W}_\Theta \otimes \mathcal{X} - \epsilon^y\|_2^2$
  $Loss_{\rm semantic} = Loss_r + Loss_c$
  //loss for calculating gradient
  $L = Loss_{\rm semantic} + Loss_{\phi_{\rm d}} + Loss_{\rm feature}$
  Gradient back propagation with loss $L$
end
Return trained ResNet50
```

- *Q*2. Does each insight/component contribute to the performance of the model (refer to Subsection 4.3)?
- *Q*3. How does the number of known types affect the performance of the model (refer to Subsection 4.4)?

4.1 Experimental Setups

4.1.1 Datasets

Five datasets are used to evaluate the proposed method, 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 tasks. OULU-NPU is a high-resolution dataset 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.

4.1.2 Quality Measurements

For the OULU-NPU 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 number of livings; 3) average classification error rate (ACER)[34], which equals the average of APCER and BPCER. In the SiW-M dataset, equal error rate (EER) and ACER are employed for evaluation as early work does. For cross-dataset testing, we apply the area under the roc curve (AUC) to evaluate all the methods.

4.1.3 Setting

We extract faces from videos by utilizing the face coordinates given by the datasets themselves. If a dataset does not provide face coordinates, we extract face coordinates by [35] as Replay-Attack does. And then we resize all extracted faces into 224×224 resolution. We employ ResNet50 as the backbone, which takes single images as inputs. Feature extraction networks and semantic representation networks are trained separately. Besides, the optimizer of Adam[36] is used to train ResNet50 with our method, and the learning rate is set to 0.001. The initial values of the hyper-parameters are given and fine-tuned according to the dimensions of the vectors they control. For example, $Loss_{\varphi_d}$ 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 (Q1)

We compare the detection accuracy of the proposed model and SOTA models in three scenarios. First, we evaluate the performance of each model on the ZSFA task, which mainly depends 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 attacks. Finally, we evaluate models for the traditional anti-spoofing task, covering only the types of seen attacks.

4.2.1 Evaluation on SiW-M for ZSFA Testing

We train ResNet50 with our method 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 ResNet50 with our proposed method on ZSFA, we compare it with five state-of-the-art (SOTA) ZSFA methods, Auxiliary[13], DTN[18], SpoofTrace[20], DC-CDN[21], and FGHV[19]. The results are demonstrated in Table 1, where the bolded values mean optimal performance and "Ours" means ResNet50 with our method (the same below). The great variances between the accuracy of models on different spoofing types indicate that diverse spoofing types differ significantly. Thus, detecting unknown spoofing types based on known types is a challenging problem. According to Table 1, we can find that ResNet50 with our method outperforms other alternative methods in terms of both EER and ACER in nearly half of all spoofing types. In particular, our method achieves an overall 25.8% optimization in terms of EER and 28.6% optimization in terms of ACER. Although the performance of our model decreases in some cases, the absolute accuracy of our model in such cases is acceptable and is competitive with those of baselines. In the worst case, the performance of our model is lower than that of the best baseline with a minor margin of less than 1%. These experiments verify the superiority of our proposed method in addressing the ZSFA issue.

To further intuitively illustrate the superiority of our model on ZSFA, as in Fig.6, we here show two samples that are successfully recognized by our model but wrongly recognized by DC-CDN. On the left, we have a living face with a red injury as in the white circle. The injury here could confuse existing models since it does not occur in most living faces so DC-CDN wrongly recognizes this sample as a spoofing face. As for the right figure, the key difference between this spoofing face with living faces is the big eyes. While such a difference could not occur in the training set, existing models may wrongly recognize this image as a living face. From this comparison, we can find clues of the superiority of introducing deceptive features.

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 the three datasets is selected for testing, and the other type for training. Due to the overlap of types in the three datasets, the experiments are usually applied to evaluate 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. This can further confirm the superiority of our proposed method in terms of generalization ability.

Table 1. ZSFA Performances of Different Models on SiW-M

Model	Metric (%)	Replay	Print	Mask Attack					Makeup Attack			Partial Attack			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[20]	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[21]	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[19]	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**

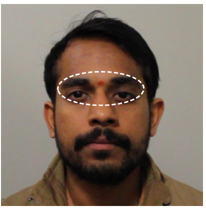
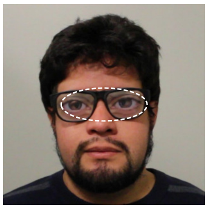
	Living	Funny_Eye
Samples and Corresponding Category		
DC-CDN	×	×
Ours	√	√

Fig.6. Case comparison between our model and DC-CDN. " √ " means that the model can correctly identify this attack. "×" means that the model makes a false identification.

4.2.3 Evaluation for Intra-Dataset Testing

Although our model is proposed for ZSFA tasks, we evaluate the performance of our model in the traditional anti-spoofing task on the OULU-NPU dataset[32]. This series of experiments strictly follows the four protocols that OULU-NPU suggests[32]. 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 a performance decrease on known spoofing faces. This suggests that our model can also handle the traditional anti-spoofing task as SOTA does.

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 (Q2)

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 experiments are conducted in the ZSFA scenario. The SiW-M[18] dataset is employed here for the 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 category of features can effectively help detect unknown category of spoofing faces, and the performances of our method in case of employing only one category of features are almost equivalent. The employment of these two categories of features can significantly enhance the performances of our method in terms of all the metrics of APCER, BPCER, and ACER. This verifies that the employment of deceptive features is effective in detecting unknown spoofing faces.

4.3.2 Impacts of Semantic Representation

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

We then set the dimensionality of semantic space to $1\,920, 1\,024, 512, 256$, and None, respectively, the results are shown in Table 5. Note that in case the dimensionality is set to None, it means that we use the feature space directly to detect unknown spoofing. As shown in this table, increasing the dimensionality of the semantic space can positively enhance the performance of our proposal, and this verifies our previous assumptions. Notice that even though the dimension-

Table 2. AUC (%) of Cross-Dataset Anti-Spoofing on CASIA, Replay, and MSU-MFSD Datasets

Model	CASIA[33]			Replay[4]			MSU-MFSD[6]			Average
	Video	Cut Photo	Wrapped Photo	Video	Digital Photo	Printed Photo	Printed Photo	HR Video	Mobile Video	
Auxiliary[13]	94.2	88.4	79.8	99.7	95.2	78.9	50.6	99.9	93.5	86.7±15.6
DTN[18]	90.0	97.3	97.5	99.9	99.9	99.6	81.6	99.9	97.5	95.9±6.2
SpoofTrace[20]	93.6	99.7	99.1	99.8	99.9	99.8	76.3	99.9	99.1	96.4±7.8
DC-CDN[21]	98.5	99.9	99.8	**100.0**	99.4	99.9	70.8	100.0	99.9	96.5±9.6
FGHV[19]	**98.6**	99.8	**99.9**	99.9	99.1	99.8	73.2	100.0	99.9	96.7±8.8
Ours	**98.8**	**99.9**	99.8	**100.0**	**100.0**	**99.9**	**86.6**	**100.0**	**100.0**	**98.3±4.2**

Table 3. Results of Traditional Anti-Spoofing on Four Protocols of OULU-NPU

Protocol	Model	APCER (%)	BPCER (%)	ACER (%)
1	Auxiliary[13]	1.6	1.6	1.6
	DTN[18]	1.3	1.5	1.4
	SpoofTrace[20]	0.8	1.3	1.1
	DC-CDN[21]	0.5	0.3	0.4
	FGHV[19]	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[20]	2.3	1.6	1.9
	DC-CDN[21]	0.9	1.9	1.3
	FGHV[19]	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[20]	**1.9±1.6**	4.0±5.4	2.8±3.3
	DC-CDN[21]	2.2±2.8	1.6±2.1	1.9±1.1
	FGHV[19]	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[20]	3.3±3.6	5.2±5.4	3.8±4.2
	DC-CDN[21]	5.4±3.3	2.5±4.2	4.0±3.1
	FGHV[19]	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**

Table 4. Impacts of Different Kinds of Features

Feature	APCER (%)	BPCER (%)	ACER (%)
Deceptive only	28.7±21.8	3.14±3.22	14.30±12.1
Discriminative only	20.4±15.2	6.44±3.72	15.70±18.1
Two kinds	**11.9±10.2**	**2.74±1.40**	**6.30±6.1**

Table 5. Impacts of Dimensionality in Semantic Space

Dimension	APCER (%)	BPCER (%)	ACER (%)
None	48.2±17.5	34.10±11.6	41.1±14.3
256	39.0±19.6	11.80±3.6	25.4±10.6
512	28.3±18.2	4.78±9.3	16.5±12.1
1 024	24.6±21.3	5.32±2.7	15.9±10.4
1 920	**11.9±10.2**	**2.74±1.4**	**6.3±6.1**

ality of the semantic space is set to 1 920, the computational complexity of the transformation process is 1.0×10^3 times less than the computational complexity of ResNet50. Thus, in our final implementation, the dimensionality is 1 920.

4.4 Hyperparameter Experiment (Q3)

Given the fact that our proposal can construct better descriptions for the living by considering the deceptive features between the livings 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. The size of the training set is 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 method deteriorates dramatically as the number of known types decreases, and this indicates that a relative 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 the Replay attack, we notice that our method with only eight 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 the Number of Known Types

Number of Known Types	ACPER (%)	BPCER (%)	ACER (%)
12	**8.7**	**1.0**	**4.2**
11	9.4	1.7	6.5
8	10.9	5.9	8.5
5	40.6	38.1	40.1

5 Conclusions

In this paper, we proposed a new method to enhance the identification accuracy of unknown spoofing in the ZSFA scenario. Our method emphasizes extracting dominant features from both deceptive and discriminative perspectives between the living and known spoofing samples. Using ResNet50[27] as the backbone, we introduced a semantic autoencoder to represent all the extracted features in a semantic space, where each dimension carries almost equal importance in distinguishing spoofing attacks. This allows ResNet50 to better differentiate between deceptive and discriminative features, which can aid in identifying attacks from unknown sample classes. Our experimental results demonstrated that our method can significantly improve the detection of unknown spoofing. Specifically, ResNet50 with our method outperformed advanced ZSFA solutions by up to 5% in terms of ACER. Additionally, our method achieved a practical performance of 98.3% in terms of AUC for detecting known spoofing instances. In future work, we plan to further investigate the classification of unknown attacks, particularly when multiple categories of unknown attacks exist.

Conflict of Interest The authors declare that they have no conflict of interest.

References

[1] Li S Z, Jain A K. Handbook of Face Recognition (2nd edition). Springer, 2011.

[2] Zhao W, Chellappa R, Phillips P J, Rosenfeld A. Face recognition: A literature survey. *ACM Computing Surveys*, 2003, 35(4): 399–458. DOI: 10.1145/954339.954342.

[3] Galbally J, Marcel S, Fierrez J. Biometric antispoofing methods: A survey in face recognition. *IEEE Access*, 2014, 2: 1530–1552. DOI: 10.1109/ACCESS.2014.2381273.

[4] Chingovska I, Anjos A, Marcel S. On the effectiveness of local binary patterns in face anti-spoofing. In *Proc. the International Conference of Biometrics Special Interest Group*, Sept. 2012, pp.1–7.

[5] Best-Rowden L, Han H, Otto C, Klare B F, Jain A K. Unconstrained face recognition: Identifying a person of interest from a media collection. *IEEE Trans. Information Forensics and Security*, 2014, 9(12): 2144–2157. DOI: 10.1109/TIFS.2014.2359577.

[6] Wen D, Han H, Jain A K. Face spoof detection with image distortion analysis. *IEEE Trans. Information Forensics and Security*, 2015, 10(4): 746–761. DOI: 10.1109/TIFS.2015.2400395.

[7] Boulkenafet Z, Komulainen J, Hadid A. Face spoofing detection using colour texture analysis. *IEEE Trans. Information Forensics and Security*, 2016, 11(8): 1818–1830. DOI: 10.1109/TIFS.2016.2555286.

[8] Boulkenafet Z, Komulainen J, Hadid A. Face anti-spoofing based on color texture analysis. In *Proc. the 2015 IEEE International Conference on Image Processing*, Sept. 2015, pp.2636–2640. DOI: 10.1109/ICIP.2015.7351280.

[9] Määttä J, Hadid A, Pietikäinen M. Face spoofing detection from single images using micro-texture analysis. In *Proc. the 2011 International Joint Conference on Biometrics*, Oct. 2011, pp.1–7. DOI: 10.1109/IJCB.2011.6117510.

[10] Patel K, Han H, Jain A K. Secure face unlock: Spoof detection on smartphones. *IEEE Trans. Information Forensics and Security*, 2016, 11(10): 2268–2283. DOI: 10.1109/TIFS.2016.2578288.

[11] Arashloo S R, Kittler J, Christmas W. An anomaly detection approach to face spoofing detection: A new formulation and evaluation protocol. *IEEE Access*, 2017, 5: 13868–13882. DOI: 10.1109/ACCESS.2017.2729161.

[12] Nikisins O, Mohammadi A, Anjos A, Marcel S. On effectiveness of anomaly detection approaches against unseen presentation attacks in face anti-spoofing. In *Proc. the 2018 International Conference on Biometrics*, Feb. 2018, pp.75–81. DOI: 10.1109/ICB2018.2018.00022.

[13] Liu Y J, Jourabloo A, Liu X M. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *Proc. the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp.389–398. DOI: 10.1109/CVPR.2018.00048.

[14] Jourabloo A, Liu Y J, Liu X M. Face de-spoofing: Anti-spoofing via noise modeling. In *Proc. the 15th European Conference on Computer Vision*, Sept. 2018, pp.297–315. DOI: 10.1007/978-3-030-01261-8_18.

[15] Atoum Y, Liu Y J, Jourabloo A, Liu X M. Face anti-spoofing using patch and depth-based CNNs. In *Proc. the 2017 IEEE International Joint Conference on Biometrics*, Oct. 2017, pp.319–328. DOI: 10.1109/BTAS.2017.8272713.

[16] Feng L T, Po L M, Li Y M, Xu X Y, Yuan F, Cheung T C H, Cheung K W. Integration of image quality and motion cues for face anti-spoofing: A neural network approach. *Journal of Visual Communication and Image Representation*, 2016, 38: 451–460. DOI: 10.1016/j.jvcir.2016.03.019.

[17] Xiong F, AbdAlmageed W. Unknown presentation attack detection with face RGB images. In *Proc. the 9th IEEE International Conference on Biometrics Theory, Applications and Systems*, Oct. 2018, pp.1–9. DOI: 10.1109/BTAS.2018.8698574.

[18] Liu Y J, Stehouwer J, Jourabloo A, Liu X M. Deep tree learning for zero-shot face anti-spoofing. In *Proc. the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2019, pp.4675–4684. DOI: 10.1109/CVPR.2019.00481.

[19] Liu S C, Lu S T, Xu H Y, Yang J, Ding S H, Ma L Z. Feature generation and hypothesis verification for reliable face anti-spoofing. In *Proc. the 36th AAAI Conference on Artificial Intelligence*, Jun. 2022, pp.1782–1791. DOI: 10.1609/AAAI.V36I2.20071.

[20] Liu Y J, Stehouwer J, Liu X M. On disentangling spoof trace for generic face anti-spoofing. In *Proc. the 16th European Conference on Computer Vision*, Aug. 2020, pp.406–422. DOI: 10.1007/978-3-030-58523-5_24.

[21] Yu Z T, Qin Y X, Zhao H S, Li X B, Zhao G Y. Dual-cross central difference network for face anti-spoofing. In *Proc. the 30th International Joint Conference on Artificial Intelligence*, Aug. 2021, pp.1281–1287. DOI: 10.24963/IJCAI.2021/177.

[22] Qin Y X, Zhao C X, Zhu X Y, Wang Z Z, Yu Z T, Fu T Y, Zhou F, Shi J P, Lei Z. Learning meta model for zero- and few-shot face anti-spoofing. In *Proc. the 34th AAAI Conference on Artificial Intelligence*, Feb. 2020, pp.11916–11923. DOI: 10.1609/AAAI.V34I07.6866.

[23] Kodirov E, Xiang T, Gong S G. Semantic autoencoder for zero-shot learning. In *Proc. the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp.4447–4456. DOI: 10.1109/CVPR.2017.473.

[24] Wang C Y, Lu Y D, Yang S T, Lai S H. Patchnet: A simple face anti-spoofing framework via fine-grained patch recognition. In *Proc. the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2022, pp.20249–20258. DOI: 10.1109/CVPR52688.2022.01964.

[25] Wang Z, Wang Z Z, Yu Z T, Deng W H, Li J H, Gao T T, Wang Z Y. Domain generalization via shuffled style assembly for face anti-spoofing. In *Proc. the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2022, pp.4113–4123. DOI: 10.1109/CVPR52688.2022.00409.

[26] Liu S Q, Yang B Y, Yuen P C, Zhao G Y. A 3D mask face anti-spoofing database with real world variations. In *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 26–Jul. 1, 2016, pp.1551–1557. DOI: 10.1109/CVPRW.2016.193.

[27] He K M, Zhang X Y, Ren S Q, Sun J. Deep residual learning for image recognition. In *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp.770–778. DOI: 10.1109/CVPR.2016.90.

[28] Tibshirani R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1996, 58(1): 267–288. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

[29] Wen Y D, Zhang K P, Li Z F, Qiao Y. A discriminative feature learning approach for deep face recognition. In *Proc. the 14th European Conference on Computer Vision*, Oct. 2016, pp.499–515. DOI: 10.1007/978-3-319-46478-7_31.

[30] Van Der Maaten L, Hinton G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008, 9(86): 2579–2605.

[31] Ranzato M A, Boureau Y L, LeCun Y. Sparse feature learning for deep belief networks. In *Proc. the 20th International Conference on Neural Information Processing Systems*, Dec. 2007, pp.1185–1192.

[32] Boulkenafet Z, Komulainen J, Li L, Feng X Y, Hadid A. OULU-NPU: A mobile face presentation attack database with real-world variations. In *Proc. the 12th IEEE International Conference on Automatic Face & Gesture Recognition*, May 30–Jun. 3, 2017, pp.612-618. DOI: 10.1109/FG.2017.77.

[33] Zhang Z W, Yan J J, Liu S F, Lei Z, Yi D, Li S Z. A face antispoofing database with diverse attacks. In *Proc. the 5th IAPR International Conference on Biometrics*, Mar. 29–Apr. 1, 2012, pp.26-31. DOI: 10.1109/ICB.2012.6199754.

[34] George A, Mostaani Z, Geissenbuhler D, Nikisins O, Anjos A, Marcel S. Biometric face presentation attack detection with multi-channel convolutional neural network. *IEEE Trans. Information Forensics and Security*, 2020, 15: 42–55. DOI: 10.1109/TIFS.2019.2916652.

[35] Froba B, Ernst A. Face detection with the modified census transform. In *Proc. the 6th IEEE International Conference on Automatic Face and Gesture Recognition*, May 2004, pp.91–96. DOI: 10.1109/AFGR.2004.1301514.

[36] Kingma D P, Ba J. Adam: A method for stochastic optimization. In *Proc. the 3rd International Conference on Learning Representations*, May 2015.

Li-Min Li is currently working toward his Ph.D. degree in the School of Software Engineering, University of Science and Technology of China, Hefei. His main research interests include spatiotemporal data mining and computer vision.

Bin-Wu Wang is currently working toward his Ph.D. degree in the School of Data Science, University of Science and Technology of China, Hefei. His main research interests include traffic data mining and continuous learning.

Xu Wang is now a research associate professor at the School of Software Engineering, University of Science and Technology of China, Hefei. He got his Ph.D. degree in 2023. His research interest mainly includes data mining and machine learning.

Peng-Kun Wang is now a research associate professor at the School of Software Engineering, University of Science and Technology of China, Hefei. He got his Ph.D. degree in 2023. His research interests mainly include spatiotemporal data mining and generalized AI for Science.

Yu-Dong Zhang is now a Ph.D. candidate in the School of Data Science, University of Science and Technology of China, Hefei. His current research interests include spatial-temporal data mining and intelligent transportation systems.

Yang Wang is now an associate professor at the School of Software Engineering, University of Science and Technology of China, Hefei. He got his Ph.D. degree at the University of Science and Technology of China, Hefei, in 2007. His research interests mainly include wireless sensor networks, spatial-temporal data mining, and data-driven interdisciplinary research.