Face Spoofing Detection Using Relativity Representation on Riemannian Manifold

Chengtang Yao, Yunde Jia, *Member, IEEE*, Huijun Di, *Member, IEEE*, and Yuwei Wu

***Abstract*— Face recognition and verification systems are susceptible to spoofing attacks using photographs, videos or masks. Most existing methods focus on spoofing detection in Euclidean space, and ignore the features' manifold structure and interrelationships, thus limiting their capabilities of discrimination and generalization. In this paper, we propose a relativity representation on Riemannian manifold for face spoofing detection. The relativity representation improves generalization capability while ensuring discriminability, at both levels of feature description and classification score. The feature-level relativity representation generalizes information by modeling interrelationships among basic features, and would not depend too much on characteristics of a particular dataset. The score-level relativity representation makes decisions relatively, not absolutely, according to interrelationships (via Riemannian metric) and competitions (via example reweighting) among data samples on Riemannian manifold. The discriminability is ensured by the high-order nature of the feature-level relativity representation as well as Riemannian reweighted discriminative learning of the score-level relativity representation. Moreover, we integrate an attack-sensitive SVM classifier in Euclidean space to improve spoofing detection. Experiments demonstrate the effectiveness of our method on both intra-dataset and cross-dataset testing.**

***Index Terms*— Face spoofing detection, relativity representation, Riemannian manifold, Riemannian reweighted discriminative learning.**

I. Introduction

FACE biometrics, as one of the most popular human identity authentication technologies, has received significant attention owing to its natural, intuitive, and less human-invasive properties. Unfortunately, face recognition and verification systems are vulnerable to spoofing attacks using photographs, videos or 3D masks [1], [2]. Researchers had spared no effort on this severe security problem and many methods have been proposed [3], [4]. Texture-based methods resort to explore very small differences in texture information via hand-crafted features, such as LBP [5]–[7], HOG [8], [9], Haralick [10], and SURF [11]. Motion-based methods distinguish attacks from genuine access through the analysis of physiological signs of life [12], or motion cues like eye blinking [13] and mouth movement [14]. Recently, there is a surge of deep learning methods in face spoofing detection [15]–[21].

The physical generation process of face spoofing artifacts includes recapture and requantization of live face images. The recapture step results in texture degradation and much noise, while the requantization step leads to image distortion. No matter what disturbances occur in these two steps, they will eventually cause some subtle differences between genuine and spoofed faces, which are difficult to distinguish. Traditional methods use low-order features (such as LBP and HOG) that are not enough to describe the complex distribution of the aforementioned subtle changes, deep learning methods provide a way to learn rich features but rely heavily on training data. Also, most of these methods focus on spoofing detection in Euclidean space, and ignore the features' manifold structure and interrelationships, thus further limiting their capabilities of discrimination and generalization.

In this paper, we propose a relativity representation on Riemannian manifold for face spoofing detection. The relativity representation improves generalization capability while ensuring discriminability, at both levels of feature description and classification score. The feature-level relativity representation generalizes information by modeling interrelationships among basic features, and ensures the discriminability as it is also a hyper-feature representation capturing high-order information among basic features. Haralick statistics [22], [23] and kernel correlation [24] are adopted as options to implement this relativity representation. We use Haralick statistics to compute the basic texture feature at each video frame, and use kernel correlation to generate the hyper-feature over basic features extracted from the input video sequence. This relativity representation is also flexible and generates features with a unified dimension for the videos with different resolution or number of frames.

The feature-level relativity representation lies on a symmetric positive definite (SPD) Riemannian manifold that is utilized by the score-level relativity representation, called Riemannian reweighted KNN score, to improve the performance of face spoofing detection. The Riemannian reweighted KNN score makes decisions relatively, not absolutely, according to interrelationships (via Riemannian metric) and competitions (via example reweighting) among data samples on Riemannian manifold. The Riemannian metric and example weights are learned to maximize the discriminability of the score. The weight learning for data examples is driven by their

Manuscript received May 16, 2019; revised December 6, 2019 and May 7, 2020; accepted May 9, 2020. Date of publication June 3, 2020; date of current version July 9, 2020. This work was supported in part by the Natural Science Foundation of China (NSFC) under Grant 61673062 and Grant 61472038. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Julian Fierrez. *(Corresponding author: Huijun Di.)*

The authors are with the Beijing Key Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: yao.c.t.adam@gmail.com; jiayunde@bit.edu.cn; ajon@bit.edu.cn; wuyuweibit@bit.edu.cn).

Digital Object Identifier 10.1109/TIFS.2020.2998956

interrelationships described by Riemannian metric. And the learned weights in turn compensate for the misclassification made by Riemannian metric. Based on this score-level relativity representation, we take the advantages of both Riemannian metric and example reweighting.

We also consider the fusion with other classifiers to further improve the performance of face spoofing detection. We present an attack-sensitive SVM to additionally consider the differences between photo attacks and video attacks, and to make use of the features' distribution information in Euclidean space. An ensemble classifier is finally built upon the Riemannian reweighted KNN and the attack-sensitive SVM. Experiments on four public datasets are conducted to demonstrate the effectiveness of our relativity representation, especially in cross-dataset testing. We also perform a comprehensive ablation study to show the rationality of each part of our method.

The rest of this paper is organized as follows. Section II reviews the related work. Section III presents the formal formulation of our method. Experiment and ablation study are discussed in Section IV. Conclusion and future work are given in Section V.

II. Related Work

A. Face Spoofing Detection

Our method uses texture features as basis for spoofing detection. Previous texture based methods detect face liveness using handcrafted features that encode image quality, texture, and so on. Kim *et al.* [25] utilized a feature based on different diffusion speeds of live and spoofed face images. Agarwal *et al.* [10] used the block-wise Haralick feature for spoofing detection. Boulkenafet *et al.* [6] focused on the analysis of chromatic components and extracted texture features based on the Local Binary Pattern (LBP). Peng *et al.* [7] designed a guided scale texture descriptor via LBP variants. A chromatic co-occurrence of LBP feature (CCoLBP) [26] was also proposed by them for face spoofing detection. In order to improve the discriminability, Boulkenafet *et al.* [11] applied the Fisher vector encoding on SURF features. Yang *et al.* [8] proposed a high-level face representation extracted by pooling the codes of low-level HOG descriptors from each face component. Komulainen *et al.* [9] also used the HOG feature to detect spoofing. Recently, there is a surge of deep learning in face spoofing detection. Yang *et al.* [15] presented the first detection method based on Convolution Neural Network (CNN). Atoum *et al.* [16] proposed a two-stream CNN-based method that extracts the local features and depth maps for face spoofing detection. Jourabloo *et al.* [17] pioneered a novel view to recognize the spoofed face images as a rerendering of live face images with additional noise, and then process them via denoising or deblurring. Due to the lack of training data in face spoofing detection and the data dependence of deep learning, Li *et al.* [18] proposed a deep local binary network to explore the utilization of hand-crafted features in the neural network. Different from the above methods that focus on designing/learning specific texture features to encode differences between live and spoofed face images, we focus on modeling features' interrelationships and manifold structure. We present a relativity representation that is new to the community of face spoofing detection and is helpful to improve generalization capability while ensuring discriminability. The features from the above methods can be used as basic features to calculate our feature-level relativity representation that is a hyper-feature representation modeling high-order interrelationships among basic features. Our score-level relativity representation further considers interrelationships and competitions among data samples on Riemannian manifold.

Motion based methods consider temporal information and detect any physiological sign of life. Motion cues like eye blinking [13] and mouth movement [14] can be analyzed for liveness detection. De Freitas Pereira *et al.* [5] extracted LBP from three orthogonal planes (LBP-TOP) feature to encode spatial-temporal information. Tu and Fang [19] used long short-term memory (LSTM) to explore both spatial information and long-range temporal relationships. The remote photoplethysmography (rPPG) signals, like heart pulse signal [12], [20], were obtained through spatial-temporal analysis. Liu *et al.* [20] learned deep models to estimate both face depth and rPPG signals with auxiliary supervision. Li *et al.* [21] considered the spatial-temporal information via a 3D CNN. Although our intention is not to use motion cues for spoofing detection, our method can also provide an option for temporal information encoding, by modeling interrelationships among video frames.

As argued by Hadid *et al.* [2], score-level fusion methods offer a flexible framework that integrates diverse countermeasure strategies/algorithms to improve the performance of spoofing detection. Yan *et al.* [27] utilized weighted sum rule for the fusion of different countermeasures. De Freitas Pereira *et al.* [28] merged normalized scores of different anti-spoofing methods. Wild *et al.* [29] utilized 1-median filtering in the fusion process. Boulkenafet *et al.* [30] applied average scheme for score fusion. Ning *et al.* [31] proposed a fusion approach to combine the lower CNN models for better predictive accuracy. Score-level fusion is also adopted in our method. We present an ensemble classifier by fusing the Riemannian reweighted KNN and the attack-sensitive SVM. The attack-sensitive SVM can also be viewed as a fusion of two SVMs that handles the differences between photo attacks and video attacks.

B. SPD Manifold Analysis

Kernel correlation matrix [24] has been widely used as a global feature representation in many areas. It is a kind of SPD matrix that lies on a specific Riemannian manifold. Riemannian metric, like affine-invariant metric [32], Log-Euclidean metric [33], or learned Log-Euclidean metric [34], is utilized to measure the geodesic distance between kernel matrices on the manifold, and to make the decision for classification, e.g., a nearest neighbor classifier based on Riemannian metric [34], or kernel subspace clustering [35]. Different from the above methods that use Riemannian metric without considering its inadequacy or the quality of each

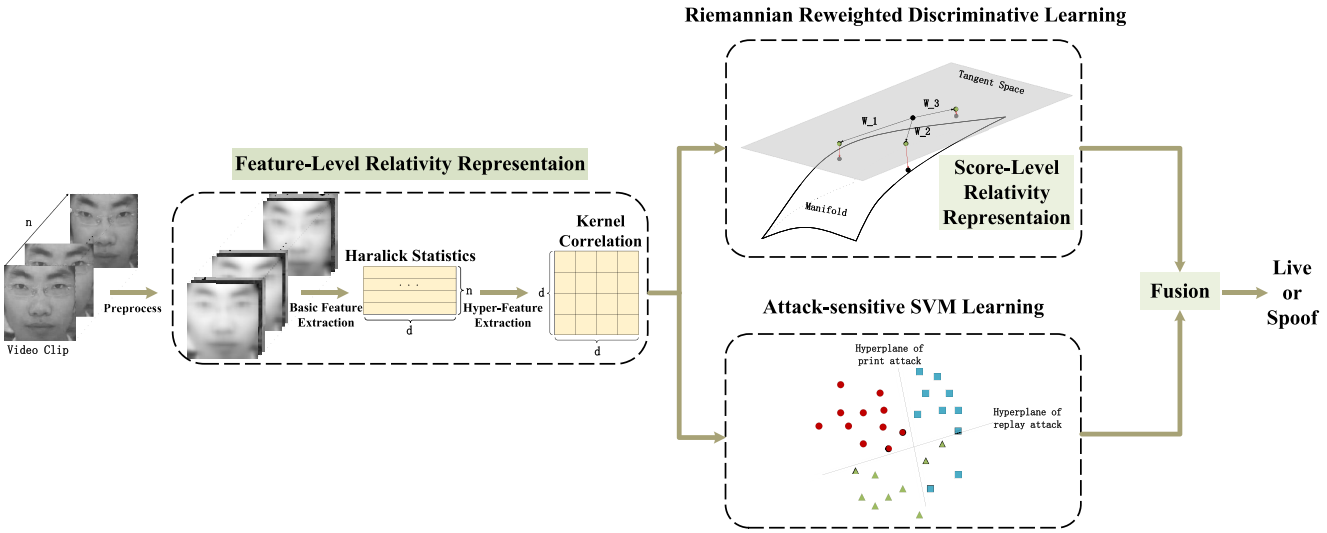

Fig. 1. An overview of the proposed method. We use Haralick statistics to compute the basic feature in each video frame, and use kernel correlation to generate the hyper-feature over basic features extracted from the input video sequence. The final classification is achieved by fusing the results from two classifiers based on Riemannian reweighted discriminative learning and attack-sensitive SVM learning respectively.

training example, we integrate Riemannian metric with example reweighting in the score-level relativity representation. The weights are learned to compensate for the misclassification made by Riemannian metric.

C. Example Reweighting

Example reweighting is used in learning-based methods to consider the importance of examples. Kahn and Marshall [36] reweighted examples to make one distribution match another one. Freund and Schapire [37] used a multiplicative weight-update technique to boost the algorithm. The self-paced learning [38], [39] was used to pick out the examples that are easier to be optimized, and to approximate the learning procedure that is from easy to hard knowledge. The meta-learning [40], [41] was also introduced to compute the weights of examples. The example reweighting was used in hard example mining [42], [43] to solve the imbalanced training data problem. It was also applied to the label noise problem by finding reliable data to robustify the learning process [44], [45]. Different from above example reweighting methods that use only the example weight to determine the importance of examples, the importance of examples in our score is jointly determined by Riemannian metric and example weight. Our score-level relativity representation takes the advantages of both Riemannian metric and example reweighting. Weight learning for our score is driven by samples' interrelationships characterized by Riemannian metric, while the learned weight adjusts the contribution of examples to reduce the misclassification made by Riemannian metric.

III. Proposed Method

The pipeline of our method is depicted in Fig. 1. Due to the effectiveness of color space [6], [46] and nonlinear diffusion [25], we first preprocess the video frames through color space transformation and channel-wise nonlinear diffusion. Then, our feature-level relativity representation is generated by computing the hyper-feature over basic features extracted from the input video sequence. Specifically, we use Haralick statistics to compute the basic feature in each video frame, and use kernel correlation to generate the hyper-feature.

After feature extraction, the classification pipeline becomes two subbranches, Riemannian reweighted discriminative learning on SPD manifold and attack-sensitive SVM learning in Euclidean space. We utilize LEML [34] to learn a Riemannian discriminative metric. Our score-level relativity representation, Riemannian reweighted KNN score, is formulated by integrating the Riemannian metric with example reweighting. The weights of examples are learned to maximize the discriminability of the score. Apart from this subbranch, we also design an attack-sensitive SVM to additionally consider the differences between photo attacks and video attacks, and to explore the features' distribution information in Euclidean space. Finally, an ensemble classifier is built upon the two classifiers via score-level fusion. More concrete details will be given in the following subsections.

A. Feature-Level Relativity Representation

1) Haralick Statistics: As aforementioned, we use Haralick statistics [22], [23] to extract basic feature for each video frame. For the sake of understanding, we give a brief description of it, but no specific calculation formula is presented here. For more details please refer to [22].

The Haralick statistics are computed from the co-occurrence matrix that describes the co-occurrence probability of the values from two pixels with a certain spatial relationship (i.e., with a certain displacement distance and direction). We consider only four (8/2) 2D neighborhood relationships (horizontal, vertical, diagonal, antidiagonal) to define the gray co-occurrences, and thirteen (26/2) 3D neighborhood relationships to define the color co-occurrences.

TABLE I
FOURTEEN HARALICK STATISTICS

Contrast	Entropy
Correlation	Diference Variance
Sum of Squares	Difference Entropy
Inverse Diference Moment	Information Measures of Correlation 1
Sum Average	Information Measures of Correlation 2
Sum Variance	*Maximal Correlation Coefficient*

Fourteen Haralick statistics, as shown in Table I, are then computed from the co-occurrence matrix under each neighborhood relationship. The Haralick statistics characterize the distribution information of the values in the co-occurrence matrix. Similar to [23], [47], we drop the last statistic, maximal correlation coefficient, as it is ordinarily considered computationally unstable. The size of the final Haralick feature for a color image is 169 (13 statistics $\times$ 13 neighborhood relationships), which is independent of the image resolution. Different from the resolution-dependent features, we do not need to downsample the high-resolution images anymore, where the resolution has a considerable influence in face spoofing detection as discussed in [48].

2) Kernel Correlation Matrix: We use kernel correlation [24] to generate the hyper-feature over basic features extracted from the input video sequence. The kernel correlation matrix is a kind of SPD matrix and has been widely used as a global feature representation in many areas. It can model the nonlinear interrelationships among the basic feature descriptors.

More concretely, suppose there are n frames in the video sequence, and the basic feature obtained in each frame is denoted as x_t, $t = 1, 2, \ldots, n$. We reform $X = [x_1, x_2, \ldots, x_n]^T \in R^{n\times d}$ as $X = [f_1, f_2, \ldots, f_d]$, where d is the feature dimension. The $(i, j)^{th}$ entry of kernel matrix $H_{d\times d}$ is defined as

$$h_{i,j} =< \phi(f_i), \phi(f_j) >= \kappa(f_i, f_j), \tag{1}$$

where $\phi(\cdot)$ is an implicit nonlinear mapping implied by a kernel function $\kappa(\cdot, \cdot)$. Here, we adopt the Gaussian RBF (radial basis function) kernel

$$\kappa(f_i, f_j) = \exp(-\gamma \| f_i - f_j \|^2). \tag{2}$$

It has a high complexity of $O(nd^2)$ to compute all the entries $h_{i,j}(i, j = 1, \ldots, d)$. To reduce the computational cost, we calculate the RBF kernel via integral images. Note that $\| f_i - f_j \|^2 = f_i^\top f_i - 2 f_i^\top f_j + f_j^\top f_j$, and d^2 integral images can be precomputed for the inner product of any two feature dimensions.

The dimension of our feature-level relativity representation is fixed as $d \times d$ and is independent of the resolution or frame number n in the video. Such a feature-level relativity representation lies on an SPD Riemannian manifold allowing us to further explore the manifold properties to improve face spoofing detection.

B. Score-Level Relativity Representation

To explore the manifold properties of the feature-level relativity representation, our score-level relativity representation, Riemannian reweighted KNN score, is formulated by integrating Riemannian metric with example reweighting. Specifically, the Riemannian reweighted KNN score of input feature H is defined as

$$Sc(H) = \sum_{i=1}^{K} w_i m(H_i, H) l_i, \tag{3}$$

where w_i, H_i and l_i are the weight, feature and label ($l_i \in \{1, -1\}$) of the training example i, respectively. The similarity measurement $m(H_i, H)$ is defined as

$$m(H_i, H) = \frac{D_{max}(H) - D_{le}(H_i, H)}{D_{max}(H) - D_{min}(H)}, \tag{4}$$

where D_{le} is a Riemannian metric (i.e., the geodesic distance on the Riemannian manifold), $D_{max/min}(H)$ is the maximum/minimum value of $D_{le}(H_i, H)$ over all training examples. The score $Sc(H)$ is evaluated from top K training examples sorted by $w_i m(H_i, H)$. The final predicted label for spoofing detection of input feature H is obtained by the sign function on the score:

$$Pred(H) = sign(Sc(H)). \tag{5}$$

The score-level relativity representation defined in Equation (3) considers interrelationships between the input feature and training features according to their Riemannian metric. We will learn a discriminant Riemannian metric to take the data distributions on the manifold into consideration to distinguish the spoofing attacks from genuine faces. The score also considers the competitions among the training examples by learning their contribution to the classification. The example weight w_i will be learned to maximize the discriminability of the score. The learning of the Riemannian metric and example weight is given in the next subsection.

C. Riemannian Reweighted Discriminative Learning

1) Riemannian Discriminative Metric Learning: We adopt LEML framework [34] to learn a discriminative Riemannian metric. The goal of the LEML is to seek a transformation W that defines a map from the original tangent space to a more discriminative one. Following the derivations in [34], the geodesic distance on the new transformed SPD manifold is formulated as

$$D_{le}^{W}(H_i, H_j) = ||W^T \log(H_i) W - W^T \log(H_j) W||_F^2, \tag{6}$$

where D_{le}^{W} is the geodesic distance under Log-Euclidean metric, $log(\cdot)$ is the matrix logarithm, H is our feature-level relativity representation, and $|| \cdot ||_F$ represents the Frobenius norms. The Equation (6) can be rewritten as

$$D_{le}^{Q}(H_i, H_j) = trace(Q(T_i - T_j)(T_i - T_j)), \tag{7}$$

where T is equal to $log(H)$, and Q is equal to $WW^T WW^T$. Then the goal of metric learning is transferred to learning the matrix Q to improve the discrimination of the spoofed and live faces.

Specifically, the distance measured by $D_{le}^{Q}(H_i, H_j)$ between paired attacks or paired live faces should be less than a small value, while the distance between attack and live face should

be greater than a large value. According to this, the objective function for discriminative metric learning is given by

$$\min_{Q,\xi} \; D_{ld}(Q, Q_0) + \eta D_{ld}(diag(\xi), diag(\xi_0)),$$
$$s.t.\ \delta_{ij} D_{le}^{Q}(H_i, H_j) \le \xi_{ij}, \tag{8}$$

where $D_{ld}(Q, Q_0) = trace(QQ_0^{-1}) - logdet(QQ_0^{-1}) - d$, d is the feature dimension size, Q_0 and ξ_0 are the initialization of Q and ξ individually, ξ is the slack bound, and δ_{ij} is 1 if the feature H_i and H_j are both from live or spoof faces, otherwise δ_{ij} is -1. Then based on [34], the Q is iteratively updated and optimized according to following equations

$$\xi_{ij}^{t+1} = \frac{\eta \xi_{ij}^{t}}{\eta + \delta_{ij}\alpha_{ij}\xi_{ij}^{t}}, \tag{9}$$

$$Q^{t+1} = Q^t + \frac{\delta_{ij}\alpha_{ij} Q^t A Q^t}{1 - \alpha_{ij} trace(Q^t A)}, \tag{10}$$

$$\alpha_{ij} = min(\frac{\delta_{ij}\eta}{\eta + 1}(\frac{1}{trace(Q^t A)} - \frac{1}{\xi_{ij}^{t}}), \lambda_{ij}), \tag{11}$$

$$\lambda_{ij} = \lambda_{ij} - \alpha_{ij}, \tag{12}$$

where $A = (T_i - T_j)(T_i - T_j)$.

2) Discriminative Learning of Example Weight: The score-level relativity representation defined in Equation (3) takes the advantages of both Riemannian metric and example reweighting. With the discriminative Riemannian metric obtained in the previous subsection, the example weights are learned to further boost the discriminability of the score. We use a validation set to learn the weight w of the training examples, by minimizing a loss function defined as

$$L(w) = \sum_{v=1}^{M} (|Sc(H_v)|G_v - Sc(H_v))^2, \tag{13}$$

where M is the size of the validation set, H_v and G_v are the feature and ground truth label of the v_{th} validation sample, respectively. It's important to have such a form of loss function with $Sc(H_v)$ instead of directly using $Pred(H_v)$, as we want to punish more on the validation samples seriously deviating from the ground truth, like $G = 1, Sc = -100$. In contrast, other forms such as $sigmoid(Sc)$, $tanh(Sc)$, and $Sc/|Sc|$ will have an opposite effect.

Gradient descent is adopted to learn the weight w_i for the training example i, and the corresponding derivative and update formula are

$$\Delta w_i = 2\sum_{v=1}^{M} \delta_{iv} m(H_i, H_v) l_i Sc(H_v)(G_v - Pred(H_v))^2, \tag{14}$$

$$w_i = w_i - \Delta w_i * lr, \tag{15}$$

where δ_{iv} is an indicator function, and lr is the learning rate. δ_{iv} takes 1 when the training example i is the top K nearest neighbors for the v_{th} validation sample, and takes 0 otherwise.

To generate the validation set when it is not provided, we randomly split 20% of the training set for validation and keep the original rate between attacks and genuine faces. After that, the weights of examples in the split training set is learned and updated. We continue this random spitting and learning 50 times to ensure all the training examples getting their updated weights that are initialized to 1 at first.

D. Ensemble Classifier

We also design an attack-sensitive SVM learning to additionally consider the differences between photo attacks and video attacks, and explore the features' distribution information in Euclidean space. Ultimately, an ensemble classifier is built upon the Riemannian reweighted KNN and the attack-sensitive SVM by a score-level fusion. Specifically, we generate the predicted scores from the two classifiers separately, and then compute the final result via a voting scheme. The details of the attack-sensitive SVM learning are described below.

Previous researches treat the face spoofing detection as a binary classification problem, i.e., live face vs. spoofed face. However, the artifacts introduced by photo attacks and video attacks are different. The mixture of many complicated patterns would hinder the classification. Therefore, it is better to split the recognition of photo attacks and video attacks at the beginning, and then recombine them to facilitate the detection. At least, this splitting strategy, which we call attack-sensitive SVM learning, would not deteriorate the recognition of live and spoofed faces compared to single SVM learning.

Specifically, as illustrated in Fig. 1, we train two SVM classifiers against photo attacks and video attacks separately. During the testing, they both receive the vectorized feature as input. Distance from the hyperplane is treated as the classification score. The testing example is finally predicted to be an attack if any of the two classifiers recognize it as a spoofed face.

IV. Experimental Results

In this section, four datasets are used to evaluate our method. They are CASIA-FASD [49], Replay-Attack [50], OULU-NPU [30] and SiW [20]. We compare our method with prior methods on both intra-dataset and cross-dataset testing and achieve impressive results, especially in cross-dataset testing. We further supply a comprehensive ablation study to reveal the effectiveness of each component in our method.

A. Datasets

The CASIA-FASD [49] dataset has 600 video clips of 50 different subjects with 150 real-access videos and 450 spoofing attack videos. It contains three quality levels of captured videos: the low-quality video captured by a long-time-used USB camera with a resolution of 480×640, the normal quality video captured by a newly bought USB camera with a resolution of 480×640 and the high-quality video captured by a Sony NEX-5 camera with maximum resolution of 1920×1080. It also contains three types of spoofing attacks, including warped photo attack, cut photo attack, and video attack.

The Replay-Attack [50] dataset has 1,300 video clips of 50 different subjects and considers the light of a fluorescent lamp and the day-light. Additionally, the videos are captured

TABLE II

PERFORMANCE COMPARISON USING HTER MEASURE ON CASIA-FASD AND REPLAY-ATTACK DATASETS (D: DEEP LEARNING METHODS, T: TRADITIONAL METHODS)

Type	Methods	Replay-Attack		CASIA-FASD(%)
		Devel(%)	Test(%)	
D	CNN [15]	4.33	2.66	6.18
	DPCNN [55]	-	6.10	4.50
	CNNs Fusion [16]	-	0.72	**2.27**
	CNN LBP-TOP [56]	-	2.68	9.94
	Ultra Net [19]	-	1.18	**1.22**
	DSGN [31]	0.13	0.63	3.42
	GDFR [21]	0.30	1.20	**1.40**
	lsCNN [57]	-	2.50	4.44
	LBP Net [58]	-	1.30	-
T	Motion+LBP [59]	-	5.11	-
	LBP-TOP [5]	-	7.60	-
	DMD+LBP [60]	-	3.75	21.75
	diffusion speed [25]	13.72	12.50	-
	BIQE [61]	-	5.38	12.70
	VLBC [62]	-	-	6.48
	LGBP [7]	0.69	3.31	3.05
	Our method	**0.00**	**0.00**	**2.96**

by an Apple 13-inch MacBook laptop with a resolution of 320×240.

The OULU-NPU [30] dataset consists of 4950 real access and attack videos from different presentation attack instruments with a resolution of 1080×1920. It also contains 4 protocols to evaluate the robustness of the face spoofing detection method. Protocol I evaluates the performance over unseen environment conditions including illumination and background. Protocol II evaluates the influence of different presentation attack instruments. Protocol III is designed to verify the generalization on various cameras. And Protocol IV considers all the factors above.

The SiW [20] dataset pays more attention to the emerging high-quality spoofing mediums. It contains 3 protocols and 4,478 videos with variations of distance, pose, illumination and expression. Protocol I evaluates the generalization over different head poses and facial expressions. Protocol II examines the robustness on diverse spoofing medium. Protocol III is proposed for unseen attack, where different attack types are used for training and testing.

B. Parameter Setting

We preprocess the image sequences with face detection and face alignment. Specifically, we use the JDA [51] to get the face bounding box and crop the face. Then we align the face with 68 feature points extracted by the work of [52].

Similar to other traditional face spoofing detection methods, we conduct a series of experiments to determine the best parameter settings. As it is difficult to determine all the parameters at once, we split out each part to find best initial choices respectively, and then obtain the overall best set of parameters via training on the holistic model. If no validation set is provided, we randomly split 20% of the training set into the validation set and keep the original rate between attacks and genuine faces.

In our method, we used RGB+HSV as the input color space. We then diffuse the input image with an additive operator splitting scheme [53] and set the iteration as 3, the time step size as 30. The γ in the kernel function is set as 0.0001 in Replay-Attack and CASIA-FASD, 0.001 in OULU and 0.01 in SiW. As for Riemannian reweighted discriminative learning, we set K as 9, $lr = 0.0001e^{-0.01*i}$ where i is the current iteration, and the other parameters are the same with [41]. In the final fusion of scores from the Riemannian reweighted KNN and the attack-sensitive SVM, we set the weights of them as 0.3, 0.7 in intra-dataset testing and 0.6, 0.4 in cross-dataset testing.

C. Evaluation Metrics

On CASIA and Replay-Attack datasets, we evaluate the performance with the commonly used False Accept Rate (FAR), False Reject Rate (FRR) and Half Total Error Rate (HTER):

$$\begin{aligned} FAR &= \frac{\sum_{i=1}^{N_{spoof}} (Pred_i == live)}{N_{spoof}}, \\ FRR &= \frac{\sum_{i=1}^{N_{live}} (Pred_i == spoof)}{N_{live}}, \\ HTER &= (FAR + FRR)/2, \end{aligned} \tag{16}$$

where N_{live} is the number of live face examples, N_{spoof} is the number of spoofed face examples, and $Pred_i$ represents the predicted label.

On OULU-NPU and SiW datasets, we use attack presentation classification error rate (APCER), bona fide presentation classification error rate (BPCER) and average classification error rate (ACER) [54] to evaluate the performance:

$$\begin{aligned} APCER &= \max_{k \in \mathcal{S}_A} (1 - \frac{1}{N_k} \sum_{i=1}^{N_k} Res_i), \\ BPCER &= \frac{\sum_{i=1}^{N_{BF}} Res_i}{N_{BF}}, \\ ACER &= (APCER + BPCER)/2, \end{aligned} \tag{17}$$

where N_k is the number of attacks for the k-th type of presentation attack in the overall set of attack instruments $\mathcal{S}_A$, N_{BF} is the number of bona fide presentations, and Res_i takes 1 if i-th example is classified as attack otherwise is assigned to 0.

D. Intra-Dataset Testing

We conduct the intra-dataset testing on CASIA-FASD, Replay-Attack, OULU-NPU and SiW datasets. For CASIA-FASD and Replay-Attack datasets, we opt for the last protocol to evaluate the general performance, just like the mainstream approaches did. For OULU-NPU and SiW datasets, we follow the protocols defined in each of them.

As reported in Table II, we get impressive results on Replay-Attack and CASIA-FASD datasets. For Replay-Attack dataset, we achieve the best result with the only training set, which mirrors our method can find the underlying distributions of live and spoof faces from limited data. For CASIA-FASD

TABLE III

PERFORMANCE COMPARISON ON THE PROTOCOL I OF OULU-NPU DATASET (D: DEEP LEARNING METHODS, T: TRADITIONAL METHODS)

Type	Method	APCER (%)	BPCER (%)	ACER (%)
D	CPqD [63]	**2.9**	10.8	6.9
	SZUCVI [63]	11.3	65.0	38.1
	MixedFASNet [63]	**0.0**	17.5	8.8
	NWPU [63]	8.8	21.7	15.2
	Recod [63]	3.3	13.3	8.3
	VSS [63]	20.0	41.7	30.8
	Auxiliary [20]	**1.6**	**1.6**	**1.6**
	Face De-Spoofing [17]	**1.2**	**1.7**	**1.5**
T	baseline [64]	5.8	21.3	13.5
	GRADIANT [63]	**1.3**	12.5	6.9
	Massy HNU [63]	5.4	20.8	13.1
	LBP+GS-LBP [7]	-	-	15.4
	CCoLBP [26]	-	-	10
	Our method	**3.33**	**2.92**	**3.13**

TABLE IV

PERFORMANCE COMPARISON ON THE PROTOCOL II OF OULU-NPU DATASET(D: DEEP LEARNING METHODS, T: TRADITIONAL METHODS)

Type	Method	APCER (%)	BPCER (%)	ACER (%)
D	CPqD [63]	14.7	3.6	9.2
	SZUCVI [63]	**3.9**	9.4	6.7
	MixedFASNet [63]	9.7	2.5	6.1
	NWPU [63]	12.5	26.7	19.6
	Recod [63]	15.8	4.2	10.0
	VSS [63]	25.3	23.9	24.6
	Auxiliary [20]	**2.7**	**2.7**	**2.7**
	Face De-Spoofing [17]	**4.2**	4.4	**4.3**
T	baseline [64]	21.5	7	14.2
	GRADIANT [63]	**3.1**	1.9	**2.5**
	Massy HNU [63]	26.1	3.9	15
	LBP+GS-LBP [7]	-	-	11.3
	CCoLBP [26]	-	-	14.9
	Our method	**8.61**	**0.50**	**4.56**

TABLE V

PERFORMANCE COMPARISON ON THE PROTOCOL III OF OULU-NPU DATASET (D: DEEP LEARNING METHODS, T: TRADITIONAL METHODS)

Type	Method	APCER (%)	BPCER (%)	ACER (%)
D	CPqD [63]	6.8±5.6	8.1±6.4	7.4±3.3
	SZUCVI [63]	12.1±10.6	16.1±8.0	14.1±4.4
	MixedFASNet [63]	**5.3±6.7**	7.8±5.5	6.5±4.6
	NWPU [63]	**3.2±2.6**	33.9±10.3	18.5±4.4
	Recod [63]	10.1±13.9	8.9±9.3	9.5±6.7
	VSS [63]	21.4±7.7	25.3±9.6	23.3±2.3
	Auxiliary [20]	**2.7±1.3**	3.1±1.7	**2.9±1.5**
	Face De-Spoofing [17]	**4.0±1.8**	3.8±1.2	**3.6±1.6**
T	baseline [64]	13.1±7.6	11.0±6.8	12.1±3.7
	GRADIANT [63]	**2.6±3.9**	5.0±5.3	**3.8±2.4**
	Massy HNU [63]	19.3±26.5	14.2±13.9	16.7±10.9
	LBP+GS-LBP [7]	-	-	16.6±10.9
	CCoLBP [26]	-	-	14.6±11.90
	Our method	**6.1±4.4**	**2.6±4.5**	**4.4±2.2**

TABLE VI

PERFORMANCE COMPARISON ON THE PROTOCOL IV OF OULU-NPU DATASET (D: DEEP LEARNING METHODS, T: TRADITIONAL METHODS)

Type	Method	APCER (%)	BPCER (%)	ACER (%)
D	CPqD [63]	32.5±37.5	11.7±12.1	22.1±20.8
	SZUCVI [63]	**0.8±2.0**	80.8±28.5	40.8±13.5
	MixedFASNet [63]	**10.0±7.7**	35.8±26.7	22.9±15.2
	NWPU [63]	30.8±7.4	44.2±23.3	37.5±9.4
	Recod [63]	35.0±37.5	10.0±4.5	22.5±18.2
	VSS [63]	21.7±8.2	44.2±11.1	32.9±5.8
	Auxiliary [20]	**9.3±5.6**	10.4±6.0	**9.5±6.0**
	Face De-Spoofing [17]	**5.1±6.3**	6.1±5.1	**5.6±5.7**
T	baseline [64]	32.5±35.6	21.9±14.1	27.2±14.3
	GRADIANT [63]	**5.0±4.5**	15.0±7.1	10.0±5.0
	Massy HNU [63]	35.8±35.3	8.3±4.1	22.1±17.6
	LBP+GS-LBP [7]	-	-	36.7±22.6
	CCoLBP [26]	-	-	22.9±18.1
	Our method	**13.3±7.5**	**5.8±7.4**	**9.6±3.7**

TABLE VII

PERFORMANCE COMPARISON ON SIW DATASET

Protocol	Method	APCER (%)	BPCER (%)	ACER (%)
1	Auxiliary [20]	3.58	3.58	3.58
	Our method	**3.35**	**2.34**	**2.84**
2	Auxiliary [20]	0.57+0.69	0.57+0.69	0.57+0.69
	Our method	1.08+1.16	**0.49+0.20**	0.79+0.67
3	Auxiliary [20]	8.31+3.81	8.31+3.80	8.31+3.81
	Our method	56.99+8.25	8.52+4.51	32.75+1.88

dataset, we are better than all traditional methods, comparable to 3 deep learning methods and better than the rest deep learning methods.

We also test our method on OULU-NPU dataset with four protocols. The first three protocols aim to evaluate the influence of light conditions, attack mediums, camera variations separately, and the last protocol considers all the above three factors simultaneously. As shown in Table III∼VI, we obtain the best results among traditional methods, similar results to 2 deep learning methods and better results than all the other deep learning methods.

Although we do not outperform all deep learning methods, especially on OULU-NPU dataset, we don't aim to completely beat the deep learning in this paper but to provide new idea and alternatives for face spoofing detection. From this point, we supply a discussion to show that the current trend, end-to-end deep learning methods, may also benefit from our view on face spoofing detection. As illustrated in the above experiments, apart from the traditional approaches, our method gets better performance than approaches with deep-features + traditional classifiers (like NWPU [63] on OULU-NPU dataset, CNN [15], DPCNN [55], and CNN LBP-TOP [56] on CASIA dataset), and all the methods on Replay-Attack dataset. Our method also exceeds deep learning methods using traditional descriptors (like LBP Net [58] and CNN LBP-TOP [56] on CASIA dataset), and even exceeds a large part of end-to-end networks (like DSGN [31] and lsCNN [57] on CASIA dataset, SZCVI [63], MixedFASNet [63], Recod [63], VSS [63], and CPqD [63] on OULU-NPU dataset).

We also evaluate our method on SiW dataset in Table VII. We achieve state-of-the-art performance on both Protocol I and Protocol II, and a worse result in protocol III. The protocol III is designed to evaluate the influence of unseen attack types,

TABLE VIII

PERFORMANCE COMPARISON USING HTER MEASURE IN CROSS-DATASET TESTING BETWEEN CASIA-FASD AND REPLAY-ATTACK DATASETS (D: DEEP LEARNING METHODS, T: TRADITIONAL METHODS). ABLATION STUDY OF THE ENSEMBLE CLASSIFIER AND EVERY SINGLE CLASSIFIER IS ALSO SHOWN HERE (RKNN IS THE RIEMANNIAN KNN WITHOUT EXAMPLE REWEIGHTING, RRKNN IS THE RIEMANNIAN REWEIGHTED KNN, ASSVM IS THE ATTACK-SENSITIVE SVM AND OUR METHOD IS THE ENSEMBLE CLASSIFIER COMBINING THE RRKNN AND THE ASSVM

Type	Methods	Train CASIA FASD	Test Replay Attack	Train Replay Attack	Test CASIA FASD
D	CNN [15]	48.50%		45.50%	
	Auxiliary [20]	27.60%		28.40%	
	Face De-Spoofing [17]	28.50%		41.10%	
T	Motion [65]	50.20%		47.90%	
	LBP-TOP [5]	49.70%		60.00%	
	Motion-Mag [66]	50.10%		47.00%	
	Color LBP [6]	37.90%		35.40%	
	Color texture [46]	30.30%		37.70%	
	LBP+GS-LBP [7]	48.36%		40.26%	
	Dictionary learning [67]	22.80%		**27.40%**	
	CTMF [68]	32.30%		45.90%	
	Motion-base [69]	33.70%		49.30%	
	Domain adaptation [70]	27.40%		36.00%	
	CCoLBP [26]	23.10%		**27.28%**	
	Our method	**16.88%**		**27.41%**	
	Our baseline: RKNN	20.25%		31.48%	
	Our baseline: RRKNN	18.13%		27.59%	
	Our baseline: ASSVM	21.00%		34.44%	

which requires the method to be trained on the photo (video) attack data and test on the video (photo) attack data. It is tough to solve such a challenging generalization across attacks. The completely different patterns between video attacks and photo attacks seriously deteriorate the performance, especially for the method based only on texture features, like us. We think there is a long way to solve the problem that deserves a lot of effort to design a more systematic method.

E. Cross-Dataset Testing

In real applications, the face spoofing detection system not only needs to accurately detect the differences between live faces and attacks in specific situations, but also is expected to be robust when meeting different data and demands. We provide cross-dataset testing to validate the generalization capability of our method. The cross testing experiments are conducted on both CASIA-FASD and Replay-Attack datasets, just like mainstream methods did. The final results and comparisons are shown in Table VIII. State-of-the-art methods can only achieve an error rate of around 25%, which discloses the significant challenge of such cross-dataset testing.

Notably, our method achieves state-of-the-art performance in the cross-dataset testing, revealing the superior capability of generalization. While our improvement of generalization is not at the sacrifice of discriminability. In intra-dataset testing discussed above, our method can still perform better than traditional methods and is comparable to deep learning methods (except for the testing on Protocol 3 of SiW dataset).

TABLE IX

PERFORMANCE COMPARISON IN CROSS-DATASET TESTING BETWEEN CASIA-FASD AND OULU-NPU DATASETS. WE USE ACER (%) ON PROTOCOL IV TO EVALUATE THE PERFORMANCE WHEN TRAINING ON CASIA-FASD AND TESTING ON OULU-NPU. WE USE HTER (%) TO EVALUATE THE PERFORMANCE WHEN TRAINING ON OULU-NPU AND TESTING ON CASIA-FASD

Methods	Train CASIA FASD	Test OULU NPU	Train OULU NPU	Test CASIA FASD
Our method	34.17±9.32		18.15	

TABLE X

THE COMPARISON BETWEEN THE BASIC FEATURE AND HYPER-FEATURE ON CASIA-FASD AND REPLAY-ATTACK DATASETS

Type	CASIA-FASD			Replay-Attack		
	FAR	FRR	HTER	FAR	FRR	HTER
Basic+SVM	16.67%	22.22%	19.44%	15.75%	11.25%	13.50%
Hyper+SVM	12.96%	11.11%	12.04%	0.00%	0.00%	0.00%

As shown in Table IX, we also provide another cross-testing between CASIA-FASD and OULU-NPU. As there is no existing research does such an experiment, we present it here with only our results and for future comparison by other methods. Our cross-testing result on OULU-NPU by using training data from CASIA-FASD is still better than one traditional and two deep-learning methods in the intra-dataset testing shown in Table VI. This reveals our excellent capability of generalization. We can also observe that using OULU-NPU can considerably improve the performance on CASIA-FASD than using Replay-Attack. This is probably due to the higher fidelity and greater difficulty of OULU-NPU dataset. This inspires us that it might be possible to pre-train a spoofing detector on high-fidelity and carefully collected dataset (like the role of ImageNet), and then use it under the situation with low-cost sensors, or as a start point for online/offline fine-tuning.

F. Ablation Study

1) Effectiveness of Feature-Level Relativity Representation: To show the necessity of our feature-level relativity representation, we compare the result of pure basic feature (Haralick statistics) and the feature-level relativity representation (hyper-feature over basic features). For the situation of pure basic feature, we get the final representation via averaging the computed feature over the input video sequence. For both situations, we trained a SVM classifier on CASIA and Replay Attack datasets separately, which suffices to show the significant differences. As illustrated in Table X, the performance of the feature-level relativity representation is much better than that of the basic feature. This experiment provides evidence on how relativity helps to improve the discriminability, by modeling high-order interrelationships among basic features.

2) Effectiveness of Example Reweighting: As aforementioned, we integrate the idea of example reweighting in our score representation. To prove its rationality, we conduct

TABLE XI
ACER OF THE COMBINED CLASSIFIER AND EVERY SINGLE CLASSIFIER ON OULU-NPU DATASET (RKNN IS THE RIEMANNIAN KNN WITHOUT EXAMPLE REWEIGHTING, RRKNN IS THE RIEMANNIAN REWEIGHTED KNN, SVM IS THE CLASSIFIER TRAINED WITHOUT DIFFERENT TREATMENT OF THE ATTACKS, ASSVM IS THE ATTACK-SENSITIVE SVM AND ENCLS IS THE ENSEMBLE CLASSIFIER COMBINING THE RRKNN AND THE ASSVM)

Type	Protocol 1	Protocol 2	Protocol 3	Protocol 4
RKNN	12.64%	19.31%	16.39%±2.56%	21.67%±3.03%
RRKNN	11.46%	12.22%	14.38%±2.40%	18.33%±4.08%
SVM	7.50%	12.22%	12.60%±3.32%	20.00%±8.06%
ASSVM	4.17%	5.69%	4.58%±2.21%	11.25%±5.18%
EnCls	3.12%	4.56%	4.37%±2.15%	9.58%±3.68%

comparative tests on OULU-NPU dataset (Table XI) and in the cross-dataset testing (Table VIII). We compare the classifiers between the Riemannian KNN (RKNN) and the proposed Riemannian reweighted KNN (RRKNN). The superior performance of the RRKNN is clear in both the intra-dataset and cross-dataset testing. This study demonstrates how example reweighting can compensate for the misclassification made by Riemannian metric.

3) Effectiveness of Attack-Sensitive Learning: In this part, we verify the effectiveness of the proposed attack-sensitive learning scheme that considers the differences between photo attacks and video attacks. As shown in Table XI, we can find that the attack-sensitive learning scheme (ASSVM) considerably improves the performance of the single SVM classifier trained without different treatment of the attacks.

4) Effectiveness of Ensemble Classifier: Our ensemble classifier is built upon the Riemannian reweighted KNN and the attack-sensitive SVM. To show how such a fusion can improve the spoofing detection in further, we compare the ensemble classifier with every single classifier in the intra-dataset testing and cross-dataset testing. In the intra-dataset testing shown in Table XI, the attack-sensitive SVM performs better than the Riemannian reweighted KNN. While in the cross-dataset testing shown in Table VIII, we can find that the results of the Riemannian reweighted KNN are better than the attack-sensitive SVM's. The final result of ensemble classifier is better than two classifiers in each testing, by a fusion of them. The fusion weights can be chosen according to the situation in practical applications. When the training data does not reflect the actual situation (the case of cross-dataset testing), more fusion weight can be added to the Riemannian reweighted KNN. While under the opposing situation (the case of intra-dataset testing), the fusion weight of the attack-sensitive SVM can be strengthened to obtain a benefit from the available training data.

G. Discussions

1) Relativity Representation: The above experiments validate that the proposed relativity representation is helpful to improve generalization capability while ensuring discriminability, at both levels of feature description and classification score. Without the score-level relativity representation, the other classifier based only on the feature-level relativity representation (i.e., the SVM/ASSVM classifier) still achieves great performance in the cross-dataset testing shown in Table VIII, and obtains comparable results in the intra-dataset testing through comparing the SVM/ASSVM in Table XI with other methods in Tables III∼VI. Such results reveal the discriminability and generalization capability of the feature-level relativity representation. In the cross-dataset testing shown in Table VIII, we observe that the performance of the Riemannian reweighted KNN is obviously better than the attack-sensitive SVM, which mirrors the generalization capability of the score-level relativity representation. In the intra-dataset testing shown in Table XI, the Riemannian reweighted KNN helps to further improve the performance when we combine it with the attack-sensitive SVM to construct an ensemble classifier. This demonstrates the discriminability of the score-level relativity representation.

2) Fusion Scheme: Existing score-level fusion methods usually integrate countermeasures that are based on different clues, features, or modalities. In this paper, we present other fusion schemes that use the same feature input but treat the feature/data differently. Our ensemble classifier intends to make use of the features' distribution information on Riemannian manifold and in Euclidean space. Experiments validate the complementarity of these two kinds of distribution information. Previous methods treat the face spoofing detection as a binary classification problem, i.e., live face vs. spoofed face. The effectiveness of the proposed attack-sensitive learning scheme shows that it is necessary to consider the problem of fine-grained attack classification (i.e., to recognize the type of attack), which can in turn improve the performance of spoofing detection by fusing the fine-grained results.

3) Generalization Across Attacks: As presented in the experiment, the performance of spoofing detection under unseen attacks is significantly reduced due to the completely distinct patterns between the different attack types in training and testing. As pointed by Hadid *et al.* [2], spoofing attacks are unpredictable and evolving. Therefore, generalization across attacks is necessary. It is worth designing new strategies/algorithms to improve such generalization capability for spoofing detection in the wild.

4) Quality of Dataset: The quality of data really matters. As shown in Table VIII and IX, almost all the models trained on the dataset containing higher resolution and more complicated influence factors have better generalization capability. Thus, it would be better to collect a larger and better quality dataset for future research. One can pre-train a spoofing detector on such a dataset, and then transfer it into other situations. Moreover, the effectiveness of example reweighting shows that training examples contribute differently to spoofing detection. Data screening, by removing/suppressing the noisy or ineffective samples, may also be useful to improve the quality of dataset or to guide the data collection.

V. CONCLUSION

We present a novel method for face spoofing detection, by using relativity representation on Riemannian manifold. Our method achieves state-of-the-art performance in the cross-dataset testing, revealing the superior capability of generalization. Our improvement of generalization is not at

the sacrifice of discriminability. In the intra-dataset testing, our method can still perform better than traditional methods and is comparable to deep learning methods (except for unseen-attack testing). The capabilities of our method come from the proposed relativity representation that is new to the community of face spoofing detection. Currently, only the texture information is modeled in our method, limiting its ability. In future research, our idea of relativity representation can be extended to model other types of cues for spoofing detection. We also plan to explore the utilization of our relativity representation in deep learning methods, to boost their generalization capability and to reduce their data dependence. Our method has a low performance under the unseen-attack testing, and meta-learning method based on our relativity representation can be adopted to improve the performance.

ACKNOWLEDGMENT

The authors want to thank Zhi Gao for his great help.

REFERENCES

[1] J. Hernandez-Ortega, J. Fierrez, A. Morales, and J. Galbally, "Introduction to face presentation attack detection," in *Handbook of Biometric Anti-Spoofing*. Cham, Switzerland: Springer, 2019, pp. 187–206.

[2] A. Hadid, N. Evans, S. Marcel, and J. Fierrez, "Biometrics systems under spoofing attack: An evaluation methodology and lessons learned," *IEEE Signal Process. Mag.*, vol. 32, no. 5, pp. 20–30, Sep. 2015.

[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] R. Ramachandra and C. Busch, "Presentation attack detection methods for face recognition systems: A comprehensive survey," *ACM Comput. Surveys*, vol. 50, no. 1, p. 8, 2017.

[5] T. De Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel, "LBP-TOP based countermeasure against face spoofing attacks," in *Proc. Asian Conf. Comput. Vis. Workshops*, in Lecture Notes in Computer Science, vol. 7728, no. 1. Berlin, Germany: Springer, 2013, pp. 121–132.

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

[7] F. Peng, L. Qin, and M. Long, "Face presentation attack detection using guided scale texture," *Multimedia Tools Appl.*, vol. 77, no. 7, pp. 8883–8909, Apr. 2018.

[8] J. Yang, Z. Lei, S. Liao, and S. Z. Li, "Face liveness detection with component dependent descriptor," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2013, pp. 1–6.

[9] J. Komulainen, A. Hadid, and M. Pietikainen, "Context based face anti-spoofing," in *Proc. IEEE Int. Conf. Biometrics Theory, Appl. Syst.*, Sep. 2013, pp. 1–8.

[10] A. Agarwal, R. Singh, and M. Vatsa, "Face anti-spoofing using haralick features," in *Proc. IEEE 8th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Sep. 2016, pp. 1–6.

[11] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face antispoofing using speeded-up robust features and Fisher vector encoding," *IEEE Signal Process. Lett.*, vol. 24, no. 2, pp. 141–145, Feb. 2017.

[12] S. Liu, P. C. Yuen, S. Zhang, and G. Zhao, "3D mask face anti-spoofing with remote photoplethysmography," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 85–100.

[13] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–2.

[14] K. Kollreider, H. Fronthaler, M. I. Faraj, and J. Bigun, "Real-time face detection and motion analysis with application in 'liveness' assessment," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 548–558, Aug. 2007.

[15] J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," 2014, *arXiv:1408.5601*. [Online]. Available: http://arxiv.org/abs/1408.5601

[16] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based CNNs," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2017, pp. 319–328.

[17] A. Jourabloo, Y. Liu, and X. Liu, "Face de-spoofing: Anti-spoofing via noise modeling," in *Proc. Eur. Conf. Comput. Vis.*, Jul. 2018, pp. 290–306.

[18] L. Li, X. Feng, X. Jiang, Z. Xia, and A. Hadid, "Face anti-spoofing via deep local binary patterns," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 101–105.

[19] X. Tu and Y. Fang, "Ultra-deep neural network for face anti-spoofing," in *Proc. Int. Conf. Neural Inf. Process.*, in Lecture Notes in Computer Science, vol. 10635. Cham, Switzerland: Springer, Nov. 2017, pp. 686–695.

[20] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 389–398.

[21] H. Li, P. He, S. Wang, A. Rocha, X. Jiang, and A. C. Kot, "Learning generalized deep feature representation for face anti-spoofing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 10, pp. 2639–2652, Oct. 2018.

[22] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[23] E. Miyamoto and T. Merryman, Jr., "Fast calculation of haralick features," Dept. Electr. Comput. Eng., Human Comput. Interact. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. 15213, 2011, pp. 1–6.

[24] L. Wang, J. Zhang, L. Zhou, C. Tang, and W. Li, "Beyond covariance: Feature representation with nonlinear kernel matrices," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4570–4578.

[25] W. Kim, S. Suh, and J.-J. Han, "Face liveness detection from a single image via diffusion speed model," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2456–2465, Aug. 2015.

[26] F. Peng, L. Qin, and M. Long, "CCoLBP: Chromatic co-occurrence of local binary pattern for face presentation attack detection," in *Proc. 27th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2018, pp. 1–9.

[27] J. Yan, Z. Zhang, Z. Lei, D. Yi, and S. Z. Li, "Face liveness detection by exploring multiple scenic clues," in *Proc. 12th Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, Dec. 2012, pp. 188–193.

[28] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel, "Can face anti-spoofing countermeasures work in a real world scenario?" in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2013, pp. 1–8.

[29] P. Wild, P. Radu, L. Chen, and J. Ferryman, "Robust multimodal face and fingerprint fusion in the presence of spoofing attacks," *Pattern Recognit.*, vol. 50, pp. 17–25, Feb. 2016.

[30] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid, "OULU-NPU: A mobile face presentation attack database with real-world variations," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2017, pp. 612–618.

[31] X. Ning, W. Li, M. Wei, L. Sun, and X. Dong, "Face anti-spoofing based on deep stack generalization networks," in *Proc. 7th Int. Conf. Pattern Recognit. Appl. Methods*, 2018, pp. 317–323.

[32] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," *Int. J. Comput. Vis.*, vol. 66, no. 1, pp. 41–66, Jan. 2006.

[33] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric means in a novel vector space structure," *SIAM J. Matrix Anal. Appl.*, vol. 29, no. 1, pp. 328–347, 2007.

[34] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen, "Log-Euclidean metric learning on symmetric positive definite manifold with application to image set classification," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 720–729.

[35] M. Yin, Y. Guo, J. Gao, Z. He, and S. Xie, "Kernel sparse subspace clustering on symmetric positive definite manifolds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5157–5164.

[36] H. Kahn and A. W. Marshall, "Methods of reducing sample size in Monte Carlo computations," *J. Oper. Res. Soc. Amer.*, vol. 1, no. 5, pp. 263–278, Nov. 1953.

[37] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[38] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1189–1197.

[39] J. S. Supancic, III, and D. Ramanan, "Self-paced learning for long-term tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2379–2386.

[40] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels," vol. 4, 2017, *arXiv:1712.05055*. [Online]. Available: http://arxiv.org/abs/1712.05055

[41] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4331–4340.
[42] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 761–769.
[43] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1367–1381, Jun. 2019.
[44] S. Azadi, J. Feng, S. Jegelka, and T. Darrell, "Auxiliary image regularization for deep CNNs with noisy labels," in *Proc. Int. Conf. Learn. Represent.*, 2016.
[45] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, "Using trusted data to train deep networks on labels corrupted by severe noise," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 10456–10465.
[46] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face anti-spoofing based on color texture analysis," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 2636–2640.
[47] M. V. Boland, "Quantitative description and automated classification of cellular protein localization patterns in fluorescence microscope images of mammalian cells," Ph.D. dissertation, Dept. Biomed. Eng., Carnegie Mellon Univ., Pittsburgh, PA, USA, 1999.
[48] J. Hernandez-Ortega, J. Fierrez, A. Morales, and P. Tome, "Time analysis of pulse-based face anti-spoofing in visible and NIR," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 544–552.
[49] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face antispoofing database with diverse attacks," in *Proc. 5th IAPR Int. Conf. Biometrics (ICB)*, Mar. 2012, pp. 26–31.
[50] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *Proc. Int. Conf. Biometrics Special Interes Group*, 2012, pp. 1–7.
[51] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, "Joint cascade face detection and alignment," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 109–122.
[52] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks)," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1021–1030.
[53] J. Weickert, B. M. T. H. Romeny, and M. A. Viergever, "Efficient and reliable schemes for nonlinear diffusion filtering," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 398–410, Mar. 1998.
[54] *Information Technology-Biometric Presentation Attack Detection-Part 3: Testing and Reporting*, Standard ISO/IEC 30107-3:2017, International Organization for Standardization, Sep. 2017.
[55] L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li, and A. Hadid, "An original face anti-spoofing approach using partial convolutional neural network," in *Proc. 6th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Dec. 2016, pp. 1–6.
[56] M. Asim, Z. Ming, and M. Y. Javed, "CNN based spatio-temporal feature extraction for face anti-spoofing," in *Proc. 2nd Int. Conf. Image, Vis. Comput.*, Jun. 2017, pp. 234–238.
[57] G. B. de Souza, J. P. Papa, and A. N. Marana, "On the learning of deep local features for robust face spoofing detection," in *Proc. 31st SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Oct. 2018, pp. 258–265.
[58] L. Li, X. Feng, Z. Xia, X. Jiang, and A. Hadid, "Face spoofing detection with local binary pattern network," *J. Vis. Commun. Image Represent.*, vol. 54, pp. 182–192, Jul. 2018.
[59] J. Komulainen, A. Hadid, M. Pietikainen, A. Anjos, and S. Marcel, "Complementary countermeasures for detecting scenic face spoofing attacks," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2013, pp. 1–7.
[60] S. Tirunagari, N. Poh, D. Windridge, A. Iorliam, N. Suki, and A. T. S. Ho, "Detection of face spoofing using visual dynamics," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 4, pp. 762–777, Apr. 2015.
[61] C.-H. Yeh and H.-H. Chang, "Face liveness detection based on perceptual image quality assessment features with multi-scale analysis," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 49–56.
[62] X. Zhao, Y. Lin, and J. Heikkila, "Dynamic texture recognition using volume local binary count patterns with an application to 2D face spoofing detection," *IEEE Trans. Multimedia*, vol. 20, no. 3, pp. 552–566, Mar. 2018.
[63] Z. Boulkenafet *et al.*, "A competition on generalized software-based face presentation attack detection in mobile scenarios," in *Proc. IEEE Int. Joint Conf. Biometrics*, Jan. 2018, pp. 688–696.
[64] T. V. Nguyen, R. K. W. Wong, and C. Hegde, "A provable approach for double-sparse coding," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3852–3859.
[65] T. De Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel, "Can face anti-spoofing countermeasures work in a real world scenario?" in *Proc. Int. Conf. Biometrics*, 2013, pp. 1–8.
[66] S. Bharadwaj, T. I. Dhamecha, M. Vatsa, and R. Singh, "Computationally efficient face spoofing detection with motion magnification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 105–110.
[67] I. Manjani, S. Tariyal, M. Vatsa, R. Singh, and A. Majumdar, "Detecting silicone mask-based presentation attack via deep dictionary learning," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 7, pp. 1713–1723, Jul. 2017.
[68] L.-B. Zhang, F. Peng, L. Qin, and M. Long, "Face spoofing detection based on color texture Markov feature and support vector machine recursive feature elimination," *J. Vis. Commun. Image Represent.*, vol. 51, pp. 56–69, Feb. 2018.
[69] T. Edmunds and A. Caplier, "Motion-based countermeasure against photo and video spoofing attacks in face recognition," *J. Vis. Commun. Image Represent.*, vol. 50, pp. 314–332, Jan. 2018.
[70] H. Li, W. Li, H. Cao, S. Wang, F. Huang, and A. C. Kot, "Unsupervised domain adaptation for face anti-spoofing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 7, pp. 1794–1809, Jul. 2018.

Chengtang Yao received the B.S. degree in computer science from the Beijing Institute of Technology (BIT) in 2018, where he is currently pursuing the M.S. degree in computer science. His current research interests include computer vision, pattern recognition, and machine learning.

Yunde Jia (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Beijing Institute of Technology (BIT) in 1983, 1986, and 2000, respectively. He was a Visiting Scientist with the Robotics Institute, Carnegie Mellon University (CMU), from 1995 to 1997. He is currently a Professor with the School of Computer Science, BIT, and the Team Head of BIT innovation on vision and media computing. He serves as the Director of the Beijing Lab of Intelligent Information Technology. His research interests include computer vision, vision-based HCI and HRI, and intelligent robotics.

Huijun Di (Member, IEEE) received the B.E. degree and the Ph.D. degree in computer science from Tsinghua University in 2002 and 2009, respectively. He was a Visiting Scholar with Siemens Corporate Research, Munich, Germany, from 2008 to 2009. His post-doctoral research was carried out at the Department of Computer Science and Technology, Tsinghua University, from 2009 to 2012. He joined the School of Computer Science, Beijing Institute of Technology, in Fall 2012. His research interests include computer vision, pattern recognition, and machine learning.

Yuwei Wu received the Ph.D. degree in computer science from the Beijing Institute of Technology (BIT), Beijing, China, in 2014. He is currently an Assistant Professor with the School of Computer Science, BIT. From August 2014 to August 2016, he was a Post-Doctoral Research Fellow with the Rapid-Rich Object Search (ROSE) Lab, School of Electrical and Electronic Engineering (EEE), Nanyang Technological University (NTU), Singapore. He has strong research interests in computer vision, information retrieval, and machine learning. He received the outstanding Ph.D. Thesis Award from BIT and the Distinguished Dissertation Award Nominee from the China Association for Artificial Intelligence (CAAI).