Deep Manifold-to-Manifold Transforming Network for Skeleton-Based Action Recognition

Tong Zhang, *Member, IEEE*, Wenming Zheng, *Senior Member, IEEE*, Zhen Cui, *Member, IEEE*, Yuan Zong, *Member, IEEE*, Chaolong Li, Xiaoyan Zhou, and Jian Yang, *Member, IEEE*

***Abstract*—In this paper, we will investigate skeleton-based action recognition by employing high-order statistics feature and first-order statistics feature, where the high-order statistics feature is characterized by symmetric positive definite (SPD) matrices. Noting that SPD matrices are theoretically embedded on Riemannian manifolds, we propose an end-to-end deep manifold-to-manifold transforming network (DMT-Net), which can make SPD matrices flow from one Riemannian manifold to another one for facilitating the action recognition task. To learn discriminative SPD features from both spatial and temporal dependencies, we propose a neural network model with three novel layers on manifolds: i.e., (1) the local SPD convolutional layer, (2) the non-linear SPD activation layer, and (3) the Riemannian-preserved recursive layer. The SPD property is preserved through all layers without the singular value decomposition (SVD) operation, which has to be conducted in the existing methods with expensive computation cost. Furthermore, a diagonalizing SPD layer is designed to efficiently calculate the final metric for the classification task. Finally, DMT-Net is further fused with a first order layer to capture temporal evolution information. To evaluate our proposed method, we conduct extensive experiments on the task of action recognition, where the input signals are represented as SPD matrices. The experimental results demonstrate that the proposed method is competitive over state-of-the-art methods.**

Manuscript received June 15, 2018; revised March 1, 2019 and September 7, 2019; accepted November 10, 2019. Date of publication January 15, 2020; date of current version October 23, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1305200, in part by the National Natural Science Foundation of China under Grants 61921004, 61906094, 61772276, 61902064, and 81971282, in part by the Jiangsu Provincial Key Research and Development Program under Grant BE2016616, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20190452, and in part by the Fundamental Research Funds for the Central Universities under Grants 30919011232, 2242018K3DN01 and 2242019K40047. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Balakrishnan Prabhakaran. *(Corresponding author: Wenming Zheng.)*

Tong Zhang is with the Key Laboratory of Child Development and Learning Science of Ministry of Education, School of Biological Science and Medical Engineering, Southeast University, Nanjing 210096, China, and also with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: tong.zhang@njust.edu.cn).

Wenming Zheng, Yuan Zong, and Chaolong Li are with the Key Laboratory of Child Development and Learning Science of Ministry of Education, School of Biological Science and Medical Engineering, Southeast University, Nanjing 210096, China (e-mail: wenming_zheng@seu.edu.cn; xhzongyuan@seu.edu.cn; lichaolong@seu.edu.cn).

Zhen Cui and Jian Yang are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: zhen.cui@njust.edu.cn; csjyang@njust.edu.cn).

Xiaoyan Zhou is with the School of Electronic and Information Engineering, Nanjing University of Information Science and Engineering Technology, Nanjing 210044, China (e-mail: xiaoyan_zhou@nuist.edu.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2020.2966878

***Index Terms*—Riemannian manifold, SPD matrix, deep learning, action recognition.**

I. Introduction

SKELETON-BASED action recognition is an active research field of computer vision in recent years and a lot of papers had been presented over the past several years [1], [3], [6], [8], [10], [11], [13], [15], [31], [36], [39], [44], [45]. Among the various action recognition methods, a typical method is based on temporal action dynamics of 3D joint locations. For instance, Xia *et al.* [48] proposed a method to learn high level visual words from skeleton data and employed hidden Markov models (HMMs) to characterize temporal evolutions of those visual words. Another popular action recognition method is to describe the similarities between joints by constructing high-order statistics features lying on manifolds, e.g., covariance matrix and its evolved versions. Based on these high-order features, feature learning methods are designed to learn more discriminative descriptors while preserving the underlying manifold structure. For example, Mehrtash *et al.* [17] proposed a manifold-to-manifold learning method to obtain discriminative symmetric positive definite (SPD) feature in lower dimensional subspace from the original SPD descriptor of covariance of 3D joints (Cov3DJ) [22].

In dealing with the skeleton-based action recognition problem, the first-order statistics feature such as the joint trajectories had been successfully used [28], [29], [37], [38], [51]. Nevertheless, it should be noted that the first-order features mainly focus on depicting temporal evolution of skeleton joints, such that they may fail to capture the relationships among the various skeleton joints. This would be disadvantageous to further improve the action recognition performance. Since the high-order statistics features, e.g., the covariance feature, can well capture the spatial relationships among the various skeletal points, there were a large amount of variants are evolved from the covariance theory in recent years for skeleton-based action recognition. In [45], Wang *et al.* proposed a kernel based method, called kernel-matrix-based (KMB) descriptor, to depict the relationship between skeletal joints for action recognition. In [22], Hussein *et al.* computed the statistical covariance of 3D Joints (Cov3DJ) as spatio-temporal SPD features to encode the relationship between joint movement meanwhile took the temporal variation of action sequences into account. The driving forces to this trend are the powerful representation ability and the behind fundamental mathematical theory of Riemannian manifold spanned by SPD matrices, of which covariance is a special case.

To deal with action recognition based on SPD matrix representation, it is desirable to perform feature learning and dimensionality reduction on Riemannian manifold in order to reduce the computation cost involved in the SVD operation and at the same time improve the recognition performance. However, standard feature learning or dimensionality reduction operations in Euclidean pace, e.g., convolution, recursive unit and activation function, can not be directly employed as they may destroy the Riemannian structure, which results in the distortion of original distribution of SPD matrices. To solve this problem, many methods, such as manifold-to-manifold transformation [17] and geometry-aware methods [16], are proposed to seek for the optimal SPD embedding matrices for dimension reduction on the Riemannian structure. More recently, Huang *et al.* [20] adopted the deep learning technique to propose a Riemannian network to learn more discriminative features from SPD matrices by designing the bilinear mapping (BiMap) layer and eigenvalue rectification (ReEig) layer.

In addition to performing the dimension reduction operation, it is also notable that the temporal dynamics information would be very useful to reduce the obscure of a single matrix descriptor (e.g., covariance) when dealing with sequential data. This had been successfully demonstrated by the recursive learning [9] and convolutional neural network (CNN) [27] methods. However, these methods are developed under the Euclidean distance metric. For manifold learning problem, the traditional Euclidean distance metric would not be applicable and hence new learning method should be developed to this end, e.g., designing SPD descriptor on the manifold. To this end, there are several crucial issues should be addressed:

1) How to perform local convolutional filtering on SPD matrices with manifold preservation?
2) How to perform recursive learning along Riemannian manifolds so as to model temporal dynamics?
3) How to avoid computationally expensive SVD during the computation of metrics in order to reduce computation cost?

Motivated by the recent progress of deep learning, especially, a deep neural network architecture of SPD matrices proposed in the literature [20] recently, in this paper we propose a deep manifold-to-manifold transforming network (DMT-Net) method under the Riemannian manifold scenario to address the aforementioned three major issues.[1] Specifically, we design a SPD convolutional layer in the DMT-Net model for the local convolutional filtering of the SPD matrices, in which we enforce the convolutional filters to be a SPD matix such that the convolutional results are also SPD matrices. The feasibility of the SPD convolution filtering on the manifold is also justified in the paper. In addition, we also design a non-linear activation layer that only needs to perform element-wise operation without the requirement of SVD operation. Moreover, to model the sequence dynamics information, we design a manifold-preserved recursive layer, which aims to encode sequentially SPD matrices of segmented subclips. Finally, we design a diagonalizing layer that converts each SPD map into a positive diagonal matrix so that the log-Euclidean metric distance between two SPD matrices can be efficiently calculated without the high-computational SVD operation.

In addition to the use of high-order statistics features, we also build an additional first-order (FO) layer so as to utilizing the first-order information to model the temporal evolution of joints by directly taking joint trajectories as input. In the FO layer, the classic gated recurrent unit (GRU) [9] is employed to capture temporal dependencies of trajectories. To utilize the learning results of both first-order and high-order features, the FO layer is further fused with DMT-Net to jointly train for recognition. In summary, the major contributions of the proposed DMT-Net model include the following four major points:

1) we propose a true local SPD convolutional layer with a theoretical guarantee of Riemannian manifold preservation which leads to the advantages of sparsity and efficiency; these should also be highlighted, while the literature [20] only employed a bilinear operation (referred to the BiMap layer).
2) we design a non-linear activation layer to avoid SVD, with a theoretical proof of manifold preservation, while the literature [20] still need SVD (referred to the ReEig layer) because its framework is based on the standard logarithm operation of matrix.
3) we design a manifold-preserved recursive layer to encode temporal dependencies of SPD matrices.
4) we develop an elegant diagonalizing trick to bypass the high computation of SVD when applying log-Euclidean mapping, while the literature [20] employed the standard log-Euclidean metric need SVD.

The remainder of this paper is organized as follows: In Section III, we address the proposed DMT-Net method for Action Recognition. In Section IV, we conduct extensive experiments to evaluate the proposed method. In Section V, we conclude our paper.

II. Related Work

In this section, we will briefly review SPD matrix descriptor, which is closely related with the proposed method of this paper.

The use of SPD matrices as a feature representation had been widely used in different patten recognition tasks [22], [25], [45]. As a special case of SPD matrix, covariance matrices are used to encode important relationship among regions in object detection [41], object tracking [35], face recognition [33] and so on. Since the SPD matrix based feature points lie on Riemannian manifolds, most algorithms attempted to extract discriminative features by operating SPD descriptors on manifolds, such as Laplacian Eigenmaps (LE) [7], Locally Linear Embedding (LLE) [14] and manifold-to-manifold transformation [17]. To measure distances between two points of manifold, various Riemannian metrics have been proposed such as affine-invariant metric [34] and Log-Euclidean metric [5]. In addition, kernelized metric versions are also developed in [23], [47], which map data to an RKHS with a family of positive definite kernels. In dealing with the distance calculation between two SPD matrices, it is notable that the distance metric of two SPD matrices defined

[1]This is an extension of our preliminary conference paper of [19].

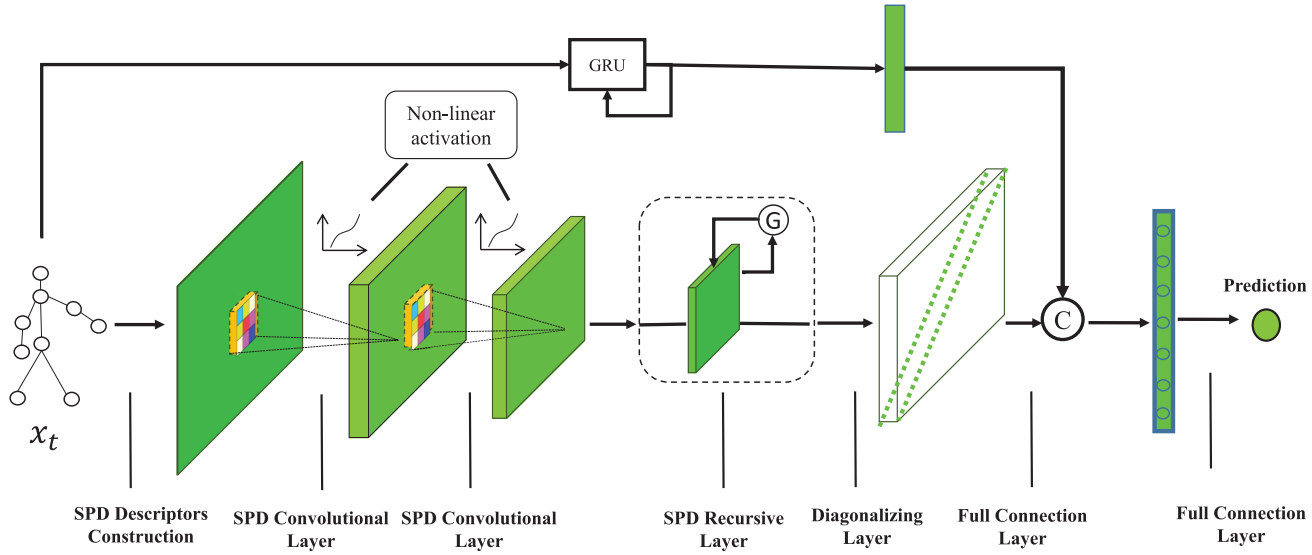

Fig. 1. The architecture of DMT network for action recognition. The raw spatio-temporal SPD features describing the skeleton based actions are fed into the network. The SPD convolutional layers (Section III-A) includes local SPD convolutional filtering and non-linear SPD activation characterizing spatial dependencies. The SPD recursive layer (Section III-B) captures temporal dependencies of sequential SPD descriptors with manifold preservation. The diagonalizing layer (Section III-C) converts SPD matrices to the specific diagonalized SPD matrices so as to implement an efficient metric computation in the next layer. With the theoretical guarantee, the matrix descriptors flow from one Riemannian manifold to another Riemannian manifold for the sake of searching more discriminative manifold spaces. 'C' means the concatenation of features of GRU and DMT-Net.

on the Riemannian manifold is different from the one defined on the Euclidean space. Due to the gap between Euclidean space and Riemannian manifold, directly applying the geometry in Euclidean space, e.g., Frobenius inner product, to measure the distances on Riemannian manifold may lead to undesired effects such as swelling effect [34]. For this reason, various Riemannian metrics operations are proposed for distance measurement, e.g., affine invariant Riemannian metric [34] and Gaussian radial basis function (RBF) kernels [24]. Besides, Log-Euclidean metric [12], [40], [42] is proposed to transform SPD manifolds from Riemannian manifold to tangent space where Euclidean geometry can be employed for measuring similarity.

III. DMT-Net for Action Recognition

In this section, we will firstly address the proposed DMT-Net model in details, and then apply it to the action recognition problem. The framework of DMT-Net is illustrated in Fig. 1, in which the inputs correspond to the raw SPD matrices/descriptors and the model layerwisely extracts discriminative features from one manifold to another manifold. Specifically, the local SPD convolutional layer performs locally convolutional filtering on SPD descriptors extracted from a subclip similar to the standard CNN method. To preserve the SPD property of the transformed matrices, the convolutional kernels should fall into the SPD space rather than a general Euclidean space, and the non-linear activation function should also satisfy the manifold preservation, which is different from the literature [20] using SVD on SPD matrices. The detailed introduction is described in Section III-A. After consecutively stacking SPD convolutional layers, we can extract discriminative SPD features from each subclip, which models the spatial information on manifold. The learned SPD features of each subclip are fed into the designed SPD recursive layer for modeling temporal dependencies. However, different from conventional RNN in Euclidean space, this recursive layer makes each state still flow on Riemannian manifolds. The detailed introduction is given in Section III-B.

Moreover, to simplify the metric computation of the SPD features, we design the diagonalizing layer, which is able to compute log-Euclidean metric efficiently without high-computational SVD operations. In the proposed diagonalizing layer, calculating matrix logarithm is equivalent to calculating the logarithm of scalar elements without any SVD of matrices operations. More details can be found in Section III-C.

Finally, the feature maps are flattened into vectors and passed through a fully connected layer followed by a softmax layer, which is addressed in Section III-D. The above operations are finally integrated to realize a fully end-to-end neural network action recognition.

A. Local SPD Convolutional Layer

This section aims to address the details information of the local SPD convolution calculation. To this end, we firstly provide the definition of multi-channel SPD map.

Definition of Multi-Channel SPD Map 1: Given a multi-channel map $\mathbf{X} \in \mathbb{R}^{C\times D\times D}$, where C is the channel number and D is the spatial dimension. If each sliced channel $\mathbf{X}^{(i)} \in \mathbb{R}^{D\times D}(i = 1, \ldots, C)$ is SPD, then we call that $\mathbf{X}$ is a multi-channel SPD map.

According to the above definition of the multi-channel SPD map, we have the following three theorems with respect to the multi-channel SPD map, which will be used for developing the local SPD convolutional layer:

Theorem 1: Given an SPD matrix $\mathbf{X} \in \mathbb{R}^{D\times D}$. Let $\mathbf{W} \in \mathbb{R}^{K\times K}$ be a convolutional kernel, then the convolutional operation is defined as

$$O_{i,j} = \sum_{p=0}^{K-1}\sum_{q=0}^{K-1} W_{p,q}X_{i+p,j+q}, \tag{1}$$

where $\mathbf{O} \in \mathbb{R}^{(D-K+1)\times(D-K+1)}$ is the filtering output. If the convolutional kernel $\mathbf{W}$ is SPD, then the output map $\mathbf{O}$ is also SPD.

Proof: The proof of Theorem 1 is given in Appendix A. ■

Based on Theorem 1, we have the following Theorem 2:

Theorem 2: Given a multi-channel SPD matrix $\mathbf{X} \in \mathbb{R}^{C\times D\times D}$. Let $\mathbf{W} \in \mathbb{R}^{C'\times C\times K\times K}$ be convolutional kernels, where C' is the kernel number and K is the kernel size. Then the convolutional operation is defined as

$$F_{i,j}^{m} = \sum_{c=0}^{C}\sum_{p=0}^{K-1}\sum_{q=0}^{K-1} W_{p,q}^{m,c}X_{i+p,j+q}^{c}, m = 1,\ldots,C', \tag{2}$$

where $\mathbf{F} \in \mathbb{R}^{C'\times(D-K+1)\times(D-K+1)}$ is the filtering output. If each convolutional kernel $\mathbf{W}^m \in \mathbb{R}^{C\times K\times K}$ is multi-channel SPD, then the output map $\mathbf{F}$ is also multi-channel SPD.

Proof: The proof of Theorem 2 is given in Appendix B. ■

Theorem 3: Given an SPD matrix, the transformation of element-wise activation with $\exp(\cdot), \sinh(\cdot)$ or $\cosh(\cdot)$ is still SPD.

Proof: The proof of Theorem 3 is given in Appendix C. ■

Now, based on the aforementioned theorems, we can develop the local SPD convolutional layer. Specifically, we construct SPD maps by using covariance or its variants. Then, we perform the local convolution filtering just like what the standard CNN does. To preserve the SPD structure, local convolutional kernels are designed for learning the optimal SPD such that they are still lying in the manifold according to Theorem 2. Moreover, to construct an SPD convolutional kernel, we employ the multiplication of one matrix $\mathbf{V}^{m,c} \in \mathbb{R}^{D\times D}$, i.e.,

$$\mathbf{W}^{m,c} = (\mathbf{V}^{m,c})^T(\mathbf{V}^{m,c}) + \epsilon\mathbf{I}, \tag{3}$$

where $\epsilon \to 0^+$ and $\mathbf{I}$ is an identity matrix. Obviously, the constructed $\mathbf{W}$ is SPD. Hence, during network learning, we only need to learn the parameter $\mathbf{V}$, and perform Eqn. (3) to implement the SPD convolution.

For the non-linear activate function, we employ the element-wise operation on some specific functions based on Theorem 3. Specifically, we perform non-linear transformations without the high-computational SVD operation that is often used in the previous methods such as the method in [20].

The convolutional filtering and the non-linear activation form the basic local SPD convolutional layer, and in this process, the SPD matrices are normalized with the Frobenius norm in case that the eigenvalues of them may go unbounded after non-linear activation. The SPD convolutional layer is rather flexible in efficient computation, which can be directly implemented by the conventional matrix operation just like the standard convolutional layer.

B. SPD Recursive Layer

The design of the SPD recursive layer is inspired by the philosophy of the classic gated recurrent unit (GRU) [9]. Specifically, we design the manifold-preserved recursive layer as follows:

$$\mathbf{R}_t^m = \sigma_g((\mathbf{W}_{fr}^m)^T\mathbf{F}_t^m\mathbf{W}_{fr}^m + (\mathbf{W}_{hr}^m)^T\mathbf{H}_{t-1}^m\mathbf{W}_{hr}^m + \epsilon\mathbf{I} + b_r^m), \tag{4}$$

$$\mathbf{Z}_t^m = \sigma_g((\mathbf{W}_{fz}^m)^T\mathbf{F}_t^m\mathbf{W}_{fz}^m + (\mathbf{W}_{hz}^m)^T\mathbf{H}_{t-1}^m\mathbf{W}_{hz}^m + \epsilon\mathbf{I} + b_z^m), \tag{5}$$

$$\widetilde{\mathbf{H}}_t^m = \sinh((\mathbf{W}_{fh}^m)^T\mathbf{F}_t^m\mathbf{W}_{fh}^m + \mathbf{H}_{t-1}^m \odot \mathbf{R}_t^m + \epsilon\mathbf{I} + b_h^m), \tag{6}$$

$$\mathbf{H}_t^m = \mathbf{Z}_t^m \odot \mathbf{H}_{t-1}^m + \widetilde{\mathbf{H}}_t^m,$$
$$\text{s.t. } b_r \geq 0, \quad b_z \geq 0, \quad b_h \geq 0, \epsilon > 0, \tag{7}$$

where $\sigma_g(\mathbf{X}) = \frac{\exp(\mathbf{X})}{\max(\exp(\mathbf{X}))}$, m means the channel number of the input multi-channel SPD matrices, the multi-channel SPD projection matrices denoted as $\mathbf{W}_{fr}, \mathbf{W}_{hr}, \mathbf{W}_{hz}, \mathbf{W}_{fr}$ and $\mathbf{W}_{fh}$ are trainable parameters for generating desirable hidden states, while $\mathbf{W}_{fr}^m, \mathbf{W}_{hr}^m, \mathbf{W}_{hz}^m, \mathbf{W}_{fr}^m$ and $\mathbf{W}_{fh}^m$ are their corresponding m-th channels, the non-linear activation function denoted as $\sigma_g(\cdot)$ is designed to generate two gates, denoted as $\mathbf{R}_t, \mathbf{Z}_t$, with manifold preservation. $\mathbf{R}_t, \mathbf{Z}_t$ have values ranging in [0, 1] and decide whether to memorize the previous output states through Hadamard product, and the non-linear activation function, $\sinh(\cdot)$, is employed to endow flexibility to current hidden state, b_r, b_z and b_h are trainable biases of positive values and $\mathbf{H}_t^m$ denotes the current output state.

From the above formulations of the manifold-preserved recursive layer, we can see that the SPD recursive layer is able to well model the temporal dependencies on manifold by properly memorizing or forgetting the hidden states. Moreover, we also have the following Theorem 4 to guarantee the manifold preservation in the SPD recursive layer:

Theorem 4: Given sequential SPD feature maps denoted as $\mathbf{F}_1, \ldots, \mathbf{F}_T$ where T is the temporal length, the defined model above Eqn. (4)-(7) is manifold-preserved.

Proof: The proof of Theorem 4 is given in Appendix D. ■

C. Diagonalizing Layer

The diagonalizing layer aims to transform the SPD matrices on manifold into the feature points of the Euclidean space, such that the conventional Euclidean distance metric can be applicable in the distance calculation on the manifold.

Let $\mathbf{Z} \in \mathbb{R}^{D\times D}$ be a SPD matrix and

$$\mathbf{Z} = \mathbf{U}^T\mathbf{\Sigma}\mathbf{U}$$

is the singular value decomposition (SVD) of $\mathbf{Z}$, where $\mathbf{U}, \mathbf{\Sigma}$ are eigenvectors and eigenvalues of $\mathbf{Z}$. Then, the logarithm operation of $\mathbf{Z}$, denoted by $\log(\mathbf{\Sigma})$, can be formulated as the following form [5]:

$$\log(\mathbf{Z}) = \mathbf{U}^T\log(\mathbf{\Sigma})\mathbf{U}. \tag{8}$$

It is notable that the distance metric on the Riemannian manifold contains the logarithm operation of the SPD matrices. In

this case, we may face the expensive computational problem in using the DMT-Net model. To solve this problem, we propose a novel method of mapping a Riemannian manifold denoted by $\mathcal{M}_1$ spanned by $D \times D$ SPD matrices, into a diagonal Riemannian manifold, denoted by $\mathcal{M}_2$ spanned by $D^2 \times D^2$ SPD matrices. More specifically, the transformation for the matrix $\mathbf{Z}$ can be formulated as the following form:

$$\tilde{\mathbf{Z}} = \text{diag}(\text{fla}(\delta(\mathbf{Z}))), \tag{9}$$

where the standard non-linear activation δ on the input SPD matrix $\mathbf{Z}$ is used to produce positive activated values, $\text{fla}(\cdot)$ denotes the operation of flattening a matrix into a vector (here the elementwise functions such as $\cosh(\cdot)$ and $\exp(\cdot)$ could be employed), in which the vector is further diagonalized into a diagonal matrix $\tilde{\mathbf{Z}}$, where the diagonal elements of $\tilde{\mathbf{Z}}$ take elements of this vector.

We have done the conversion from a general SPD matrix $\mathbf{Z}$ on the manifold $\mathcal{M}_1$ to another SPD matrix $\tilde{\mathbf{Z}}$, where $\tilde{\mathbf{Z}}$ still lies on a Riemannian manifold denoted as $\mathcal{M}_2$ due to its SPD property. Since $\tilde{\mathbf{Z}}$ is a diagonal SPD matrix, the matrix logarithm only need perform the general element-wise logarithm operation on each diagonal element. This brings two major advantages:

1) The time-consuming operation of SVD is avoided in our method;
2) The computation is very efficient without any additional memory space because we only use the the non-zero elements of the diagonal matrix.

After the diagonalizing layer, the discriminative feature maps of dynamic sequence are obtained. We vectorize these diagonal feature maps with only non-zero elements and remove those uninformative zeros values.

D. The Cross-Entropy Loss

We adopt the cross entropy loss defined as the objective loss function of the proposed DMT-Net model, where the cross entropy loss [43] can be expressed as the following form:

$$E_D = -\sum_{j=1}^{N}\sum_{c=1}^{N_{class}} \tau(y_j, c) \times \log(P_D(c|\mathbf{X}_j)), \tag{10}$$

where E denotes the cross entropy loss, $\mathbf{X}_j$ represents the j-th training sample of the training set, $P_D(c|\mathbf{X}_j)$ denotes the prediction probability of DMT-Net and y_j is the label of the j-th training sample, and $\tau(x, y)$ is defined as

$$\tau(x, y) = \begin{cases} 1, & \text{if } x = y; \\ 0, & \text{otherwise.} \end{cases}$$

E. Fusing DMT-Net With FO Layer

As DMT-Net focuses more on modeling the high-order covariances among joint points, it may ignore the temporal evolution of joint trajectories when being applied to dealing with the action recognition problem. For this reason, we additionally construct an FO layer which directly takes joint trajectories as input. This layer employs GRU to model temporal dependencies of trajectories so that it captures complementary information to DMT-Net, which further promotes the network performance.

To fuse DMT-Net with FO Layer, we concatenate the output features of them and pass it through a common fully connected layer and softmax layer to obtain the corresponding prediction probability. Then, the obtained prediction probability is used to calculate the loss for the whole network training. The network after fusion is named as Fused DMT-Net (F-DMT-Net). Furthermore, we apply a two-stage training on the DMT-Net and FO Layer: (1) the DMT-Net and FO layer are first trained one by one while fix the other; (2) the pretrained two parts are further optimized together with a relative low learning rate. In detail, these two processes can be achieved by weighting two control factors, i.e., β_1 and β_2 to the outputs of the DMT-Net and FO Layer during concatenation, respectively. Then, by iteratively $\beta_1$and β_2 to 1 and 0, the DMT-Net and FO layers can thus be trained one by one. Moreover, by setting $\beta_1$and β_2 to be both 1, the network of the two parts can be jointly optimized.

IV. Experiments

We evaluate our DMT-Net and F-DMT-Net by conducting experiments on four action recognition datasets, i.e., NTU RGB+D (NTU) dataset [37], the Large Scale Combined (LSC) dataset [50], HDM05 database [30] and the Florence 3D Actions dataset [36]. Before the experiments, we will firstly specify the implemental procedures of the proposed method in details, which consist of the following parts:

1) Preprocessing: The preprocessing aims to reduce the variations of skeleton data, e.g., body orientation variation and body scale variation, which contains the following three steps:

a) Downsample the action sequences to a unified length by splitting them into a fixed number of subsequences and randomly choose one frame from each subsequence.
b) Scale the skeletons with different factors ranging in [0.95, 1.05] to improve the adaptive scaling capacity.
c) Rotate the skeletons along x, y and z axis with angles ranging in $[-45, 45]$ during training stage, which make the model be robust to orientation variation. Fig. 2 shows this process with two actions of Florence dataset.

2) SPD Feature Extraction: Let $\mathbf{S}_t = [\mathbf{s}_t^1, \ldots, \mathbf{s}_t^J]^T \in \mathbb{R}^{J\times 3}$ represent the joint locations of the t-th frame, where J is the number of joints and L is the temporal length of action sequence, then the SPD feature of the t-th frame, denoted by $\mathbf{X}_t$, can be calculated as follows:

$$\mathbf{X}_t = \text{fla}(\mathbf{S}_t - \bar{\mathbf{S}})(\text{fla}(\mathbf{S}_t - \bar{\mathbf{S}}))^T + \epsilon * \mathbf{I}, \tag{11}$$

where $\bar{\mathbf{S}}$ can be expressed as

$$\bar{\mathbf{S}} = [\mathbf{s}_p, \ldots, \mathbf{s}_p]^T, \quad \mathbf{s}_p = \frac{1}{LJ}\sum_{t=1}^{L}\sum_{j=1}^{J}\mathbf{s}_t^j. \tag{12}$$

3) Network Architecture and Parameter Setting: In preprocessing process, the sequences are split into 12 subsequences for all datasets. For recognition, we employ the same architecture of DMT-Net when evaluating all the datasets. We train the model for 500 epochs in total with the learning rate of 0.003

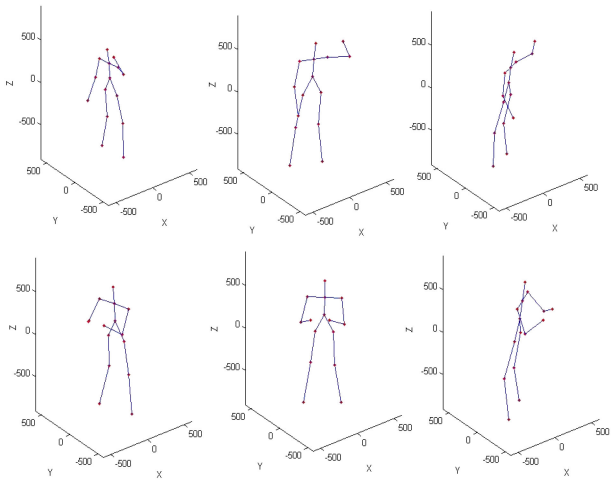

Fig. 2. Examples of rotation around z axis. The first row contains rotated samples of action 'wave' and the second row contains rotated samples of action 'clap'.

to tune our F-DMT-Net, where the optimizer is RMSPropOptimizer with the momentum of 0.9. For the detailed architecture, as we can see from Fig. 1, the DMT-Net model contains two SPD convolutional layers, one SPD recursive layer, one diagonalizing layer, one fully connected layer and one softmax layer. For SPD convolutional layers, we double the number of SPD convolutional kernels in the second convolutional layer comparing to the first layer. When fusing with FO Layer, the number of hidden nodes of GRU is set to be 128.

Specifically, for the parameter setting, we double the number of SPD convolutional kernels beacuse those popular CNN models in Euclidean space, e.g., VGG_Net and Res_Net, set the parameters like this. Moreover, we also set parameters according to the scale of datasets. The three datasets, i.e., the LSC, HDM05 and Florence datasets, are in similar size and thus set the same parameters. In contrast, NTU contains much more samples so we set more parameters.

A. Experiments on NTU Dataset

The NTU dataset consists of 56880 RGB+D video samples executed by 40 different human subjects, whose ages range from 10 to 35. In this dataset, signals of various modalities, including RGB videos, depth sequences, skeleton data, and infrared frames, are collected by three synchronous Microsoft Kinect v2 sensors from three different horizontal angles. Specifically, skeleton data contains 3D locations of 25 major body joints. The large amount of samples, intra-class and view point variations make this dataset great challenging.

We conduct experiments by following two different protocols in [37], named cross-subject and cross-view protocols respectively. For cross-subject protocol, samples are split to a training and a testing sets according to subjects' ID numbers where each set contains 20 subjects. Thus, the training and testing sets have 40320 and 16560 samples, respectively. For cross-view protocol, samples captured by cameras #2 and #3 are used for training while the samples of camera #1 are for testing. This protocol yields 37920 and 18960 samples for training and testing sets respectively. For this dataset, the sizes of the SPD convolutional kernels in the two SPD convolutional layers of DMT-Net are set to be $8 \times 1 \times 5 \times 5$ and $16 \times 8 \times 5 \times 5$ respectively.

TABLE I
THE COMPARISONS ON NTU DATASET

Method	Cross Subject Accuracy (%)	Cross View Accuracy(%)
Lie group [44]	50.08	52.76
HBRNN [11]	59.07	63.97
LieNet [21]	61.37	66.95
Deep LSTM [37]	60.69	67.29
P-LSTM [37]	62.93	70.27
ST-LSTM [28]	69.20	77.70
STA-LSTM [38]	73.40	81.20
GF-LSTM [51]	70.26	82.39
GCA-LSTM [29]	**74.40**	82.80
DMT-Net	69.2	79.3
F-DMT-Net	73.8	**84.4**

The proposed framework is compared with various the-state-of-the-art methods, e.g., different kinds of recurrent neural networks (RNNs) including hierarchical bidirectional recurrent neural networks (HBRNN) [11], part-aware LSTM (P-LSTM) [37], spatio-temporal LSTM (ST-LSTM) [28], spatio-temporal attention LSTM (STA-LSTM) [38], global context-aware attention LSTM (GCA-LSTM) [29], and geometric features LSTM (GF-LSTM) [51]. The experimental results of the various methods are listed in Table I.

From Table I, we can see that, for cross-subject protocol, F-DMT-Net outperforms most algorithms except GCA-LSTM, where the accuracy is 0.6 percent lower. For cross-view protocol, however, F-DMT-Net achieves the best accuracy comparing to previous methods and outperforms GCA-LSTM with the accuracy of 1.6 percent higher.

B. Experiments on Large Scale Combined Dataset

The LSC dataset is created by combing nine existing public datasets with both red, green and blue (RGB) video and depth information. In total, it contains 4953 video sequences of 94 action classes performed by 107 subjects. As these video sequences come from different individual datasets, the variations with respect to subjects, performing manners and backgrounds are very large. Moreover, the number of samples for each action is different. All these factors, i.e., the large size, the large variations and the data imbalance for each class, make LSC dataset challenging for recognition.

We conduct experiments by following two different protocols employed in [50]. The first protocol, named Random Cross Sample (RCSam), consisting of 88 action classes, in which half of the samples of each class are randomly selected as training data while the rest of ones are used as testing data. The second protocol, named Random Cross subject (RCSub), consisting of 88 action classes, in which half of the subjects are randomly selected as training data and the rest of subjects are used as testing data. In both protocols, only skeleton data is used for recognition. Due to the imbalance of samples in each class, the

TABLE II
THE COMPARISONS ON LSC DATASET FOLLOWING RCSAM AND RCSUB PROTOCOLS

Protocol	Method	Precision (%)	Recall (%)
RCSam	HON4D [32]	84.6	84.1
	Dynamic skeleton [50]	85.9	85.6
	P-LSTM [37]	84.2	84.9
	DMT-Net	86.5	85.1
	F-DMT-Net	**87.6**	**85.7**
RCSub	HON4D [32]	63.1	59.3
	Dynamic skeleton [50]	74.5	73.7
	P-LSTM [37]	76.3	74.6
	DMT-Net	80.5	76.3
	F-DMT-Net	**84.2**	**80.2**

values of precision and recall are employed for evaluating the performance instead of accuracy. The sizes of the SPD convolutional kernels in the two SPD convolutional layer are set to be $8 \times 1 \times 5 \times 5$ and $16 \times 8 \times 5 \times 5$. The comparisons on LSC dataset are shown in Table II.

From Table II, we can see that, for both protocols, our DMT-Net and F-DMT-Net achieve competitive performance comparing to previous methods. Especially, for RCSub protocol, F-DMT-Net achieves relatively better performance: the values of precision and recall are respectively almost 8 and 6 percent higher than P-LSTM in [37], which demonstrates that the propose method is more robust to the variation caused by differences of subjects.

C. Experiments on HDM05 Dataset

The HDM05 dataset contains 2,273 sequences of 130 motion classes executed by five actors named 'bd,' 'bk,' 'dg,' 'mm' and 'tr,' and each class has 10 to 50 realizations. The skeleton data contains as many as 31 joints so that it leads to larger size of covariance matrix feature. The large amount of action classes, which may be the most among skeleton based datasets, and also the data imbalance for each class make this dataset difficult for recognition.

To achieve a comprehensive comparison to the state-of-the-art methods on HDM05 dataset, we conduct experiments by following two different protocols employed in previous literatures. For the first protocol employed in [45], actions performed by two subjects named 'bd' and 'mm' are used for training and the remaining samples are for testing. For the second protocol employed in [20], 10 random evaluations are conducted and for each evaluation a half of the samples of each class are randomly selected for training and the rests for testing. The sizes of the SPD convolutional kernels in the two SPD convolutional layer are set to be $8 \times 1 \times 5 \times 5$ and $16 \times 8 \times 5 \times 5$ respectively. The comparisons on HDM05 dataset are shown in Table III.

From Table III, we can see that, for both protocols, our DMT-Net achieves the best performance comparing to the-state-of-the-art methods. Specifically, for the second protocol, DMT-Net is compared with a Riemannian network proposed in [20] which is another kind of deep network on Riemannian manifold. The proposed DMT-Net outperforms the Riemannian network with the recognition rate of 81.52% which is almost 20 percent higher. Moreover, the F-DMT-Net further promotes the performance to the accuracy of 85.30%.

TABLE III
THE COMPARISONS ON HDM05 DATASET

Method	Protocol #1 Accuracy (%)	Protocol #2 Accuracy(%)
RSR-ML [17]	40.00	-
Cov-RP [41]	58.90	-
Ker-RP-RBF [45]	66.20	-
Lie group [44]	-	70.26 ± 2.89
LieNet [21]	-	75.78 ± 2.26
SPDNet [20]	-	61.45 ± 1.12
P-LSTM [37]	70.04	73.42 ± 2.05
DMT-Net-kernel	71.2	-
DMT-Net	77.85	81.52 ± 1.17
F-DMT-Net	**79.72**	**85.30 ± 1.58**

TABLE IV
THE COMPARISONS ON FLORENCE DATASET

Method	Accuracy (%)
Multi-part Bag-of-Poses [36]	82.00
Lie group [44]	90.08
Shape Analysis on Manifold [10]	87.04
Elastic Function Coding [4]	89.67
Graph Based Representation [46]	91.63
Multi-instance Multitask Learning [49]	95.29
P-LSTM [37]	95.35
Tensor Representation [26]	95.47
DMT-Net-kernel	95.35
DMT-Net	96.72
F-DMT-Net	**99.55**

D. Experiment on Florence Dataset

The Florence 3D Actions dataset contains 9 activities: 'wave (WV),' 'drink from a bottle (DB),' 'answer phone (AP),' 'clap (CL),' 'tight lace (TL),' 'sit down (SD),' 'stand up (SU),' 'read watch (RW)' and 'bow(BO)'. These actions are performed by 10 subjects for 2 or 3 times yielding 215 activity samples in total and represented by 15 joints without depth data. Similar actions such as 'drink from a bottle' and 'answer phone' are easy to be confused.

We follow the similar protocol of leave-one-subject-out validation as [46], where skeleton data of nine subjects is used for training and the resting part is for testing. The sizes of the SPD filters in the two SPD convolutional layer are set to be $4 \times 1 \times 5 \times 5$ and $8 \times 4 \times 5 \times 5$ respectively. The result of the proposed DMT-Net dataset on Florence dataset is shown in Table IV, in which we also provide the comparisons with various existed algorithms, such as [4], [44], [46].

TABLE V
COMPARISONS OF THE DMT-NET, FO LAYER AND F-DMT-NET ON NTU, LSC, HDM05 AND FLORENCE DATASETS

Module	NTU dataset (Cross Subject protocol)	NTU dataset (Cross View protocol)	LSC dataset (RCSam protocol)		LSC dataset (RCsub protocol)		HDM05 dataset (Protocol #1)	HDM05 dataset (Protocol #2)	Florence dataset
	Accuracy	Accuracy	Precision	Recall	Precision	Recall	Accuracy	Accuracy	Accuracy
DMT-Net	69.2	79.3	86.5	85.1	80.5	76.3	77.9	81.5 ± 1.17	96.7
FO-Layer	70.7	80.1	84.1	83.5	77.0	75.6	77.9	82.4 ± 1.43	93.0
F-DMT-Net	**73.8**	**84.4**	**87.6**	**85.7**	**84.2**	**80.2**	**79.7**	$\mathbf{85.3 \pm 1.58}$	**99.6**

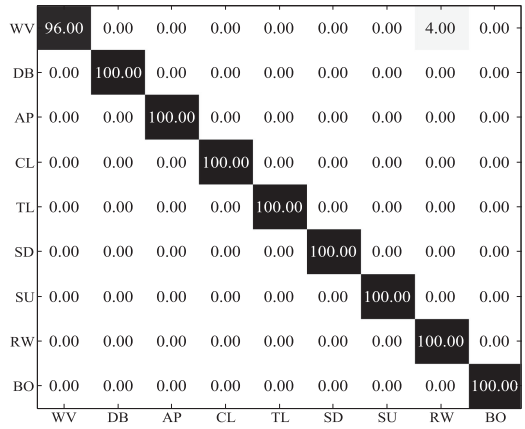

Fig. 3. The experimental results of confusion matrix on Florence dataset.

From Table IV, we can see that the proposed F-DMT-Net achieves the recognition accuracy of 99.55%, which is about 4 percent higher than the algorithm proposed in [26].

Moreover, we also show the the confusion matrices of the different recognition results of actions in Fig. 3, from which we can see that F-DMT-Net performs well on eight actions with 100% recognition rates. Main confusion appears between the pair of actions which are 'wave' versus 'read watch'.

E. Additional Experiments for DMT-Net

In this section, we will conduct additional experiments to dissect our network architecture. The following five baseline experiments are respectively conducted for this purpose:

a) *Comparisons of DMT-Net and FO Layer:* The experiments aim to verify how the two modules in F-DMT-Net, i.e., DMT-Net and FO Layer, promote the recogntion performance. Table V compare the experimental results on the four datasets.

b) *Comparisons of DMT-Net and DMT-Net Without SPD Convolutional Layer:* We remove the two SPD convolutional layers from DMT-Net so that the resulting network mainly contains only one SPD recursive layer. The size of each channel of the hidden state in this SPD recursive layer is 20×20. The experimental result is shown in Table VI.

c) *Comparisons of DMT-Net and DMT-Net Without SPD Recursive Layer:* To verify the effectiveness of SPD recursive layer, we construct a network by removing SPD recursive layer from DMT-Net. The sizes of SPD filters in the two SPD convolutional layers remain the same as DMT-Net, which are $8 \times 1 \times 5 \times 5$ and $16 \times 8 \times 5 \times 5$ respectively. The experimental result is shown in Table VI.

TABLE VI
THE ARCHITECTURE EVALUATION WITH HDM05 DATASET USING PROTOCOL #1

Method	Accuracy (%)
DMT-Net without SPD convolutional layer	69.22
DMT-Net without SPD recursive layer	75.58
DMT-Net without non-linear activation	76.46
DMT-Net	**77.85**

TABLE VII
THE CONVOLUTIONAL KERNEL SIZE EVALUATION WITH HDM05 DATASET FOLLOWING PROTOCOL #1

SPD convolutional kernel size	Accuracy (%)
3×3	74.72
5×5	**77.85**
7×7	77.59

d) *Comparisons of DMT-Net and DMT-Net Without Non-Linear Activation:* We remove the non-linear activation functions in two SPD convolutional layers from DMT-Net to verify the effectiveness of non-linear activation. The experimental result is shown in Table VI.

e) *Results of DMT-Net With SPD Convolutional Kernels of Different Sizes:* We set the sizes of SPD convolutional kernels to be 3×3, 5×5 and 7×7 to see how different kernel sizes influent the recognition results. The experimental result is shown in Table VII.

According to the aforementioned baseline experimental results of Tables V–VII evaluated on HDM05 dataset with the protocol #1, we have the following observations:

a) *F-DMT-Net Outperforms Either DMT-Net or FO Layer:* Either DMT-Net or FO layer benefits the performance because they capture two different types of information, i.e. the first order information depicting trajectory variation and high-order information of joint correlation. The performance gain of either DMT-Net or FO layer also verifies the effectiveness of our fusion strategy. In addition, The superiority of FO-Layer outperforms DMT-Net on NTU and HDM05, which may attribute to the reason that or actions in NTU and HDM05, the trajectory information and its temporal variation may contribute more to the recognition.

b) *Convolutional Filtering Plays a Crucial Role in the Performance Promotion Like the Standard CNN in Euclidean Space:* In manifold space composed of covariance matrices, the local convolution filtering should more extract some bundling features co-occurred for certain task.

c) *The SPD Recursive Layer can Further Improve the Performance Due to the Introduction of Sequence Dynamics:* According to the performances of DMT-Net vs DMT-Net without SPD recursive layer, the accuracy of DMT-Net is almost 2.3 percent higher than the network without recursive layer.
d) *The Non-Linear Activation Function is Effective for Endowing Flexibility:* Without non-linear activation the performance drops about 1.4 percent comparing to original DMT-Net.
e) *The Kernel Sizes of SPD Filters Influence the Performance of Action Recognition:* For HDM05 dataset, best performance is achieved when the sizes are set as 5×5.

V. Conclusion

We have proposed the DMT-Net model for dealing with the action recognition problem of spatio-temporal dynamic sequences, in which the entire sequence is segmented into several clips and each clip is described by a SPD matrix. Since SPD matrices are embedded on Riemannian manifold, we designed a series of novel layers to transform the SPD matrices for extracting the discriminative features while preserving the transformed matrices lying on the manifolds. The constructed layers contain SPD convolutional, non-linear activation, SPD recursive and the diagonalizing layer. All these layers do not need the high computational SVD operation. The proposed model is generic for the representation learning of manifolds, thus it may have a constructive value for deep learning and manifold learning research fields. Moreover, the recognition performance of skeleton based action recognition was further promoted by fusing the DMT-Net with an FO Layer. We conducted experiments on the task of skeleton based action recognition, and achieved the state-of-the-art performance under the same experimental environments.

Appendix A
Proof of Theorem 1

Proof: As $\mathbf{W}$ is SPD, it can be decomposed into:

$$\mathbf{W} = \mathbf{V}\mathbf{V}^T, \tag{13}$$

where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K]$ is a matrix of full rank, and K is the number of both columns and rows of $\mathbf{W}$. Then the convolutional result of an SPD representation matrix $\mathbf{X} \in \mathbb{R}^{D\times D}$ can be written as

$$\mathbf{O} = \mathbf{X} * \mathbf{W} = \mathbf{X} * (\mathbf{V}\mathbf{V}^T) \tag{14}$$

$$= \mathbf{X} * (\mathbf{v}_1\mathbf{v}_1^T) + \cdots + \mathbf{X} * (\mathbf{v}_K\mathbf{v}_K^T) \tag{15}$$

$$= \mathbf{X} * \mathbf{v}_1 * \mathbf{v}_1^T + \cdots + \mathbf{X} * \mathbf{v}_K * \mathbf{v}_K^T, \tag{16}$$

where the derivation from Eqn. (15) to Eqn. (16) uses the property of separable convolution. Suppose that $\mathbf{v}_i = [v_1, v_2, \ldots, v_K]^T, i = [1, 2, \ldots, K]$, the convolution between $\mathbf{X}$ and $\mathbf{v}_i$ can be written as:

$$\mathbf{X} * \mathbf{v}_i = \mathbf{G}_{\mathbf{v}_i}\mathbf{X}, \quad \mathbf{X} * \mathbf{v}_i^T = \mathbf{X}\mathbf{G}_{\mathbf{v}_i}^T,$$

where $\mathbf{G}_{\mathbf{v}_i} \in \mathbb{R}^{(D-K+1)\times D}$ and

$$\mathbf{G}_{\mathbf{v}_i} = \begin{bmatrix} v_1, & v_2, & \ldots, & v_K, & 0, & 0, & 0, & \ldots, & 0 \\ 0, & v_1, v_2, & \ldots, & v_K, & 0, & 0, & \ldots, & & 0 \\ 0, & 0, & v_1, v_2, & \ldots, & v_K, & 0, & \ldots, & & 0 \\ & & & \cdots & & & & & \\ 0, & 0, & \ldots, & 0, & v_1, & v_2, & \ldots, & v_K & \end{bmatrix}. \tag{17}$$

Then we get the following equations:

$$\mathbf{X} * \mathbf{v}_i * \mathbf{v}_i^T = \mathbf{G}_{\mathbf{v}_i}\mathbf{X}\mathbf{G}_{\mathbf{v}_i}^T, \tag{18}$$

and

$$\mathbf{O} = \mathbf{X} * \mathbf{V} = \mathbf{G}_{\mathbf{v}_1}\mathbf{X}\mathbf{G}_{\mathbf{v}_1}^T + \cdots + \mathbf{G}_{\mathbf{v}_K}\mathbf{X}\mathbf{G}_{\mathbf{v}_K}^T. \tag{19}$$

As the rank of $\mathbf{G}_{\mathbf{v}_i}$ equals $D - K + 1$, $\mathbf{G}_{\mathbf{v}_i}\mathbf{X}\mathbf{G}_{\mathbf{v}_i}^T$ is also an SPD matrix. Thus $\forall\, \mathbf{z} \in \mathbb{R}^{D-K+1}, \mathbf{z} \neq \mathbf{0}$, we have

$$\mathbf{z}^T\mathbf{O}\mathbf{z} = \sum_{i=1}^{K} \mathbf{z}^T\mathbf{G}_{\mathbf{v}_i}\mathbf{X}\mathbf{G}_{\mathbf{v}_i}^T\mathbf{z} > 0. \tag{20}$$

So $\mathbf{O}$ is an SPD matrix. ■

Appendix B
Proof of Theorem 2

Proof: The m-th channel of $\mathbf{F}$ can be written as:

$$\mathbf{F}^{(m)} = \sum_{c=1}^{C} \mathbf{X}^{(c)} * \mathbf{W}^{(m,c)}, \tag{21}$$

where $\mathbf{X}^{(c)}$ denotes the c-th channel of input descriptor, apparently $\mathbf{X}^{(c)}$ and $\mathbf{W}^{(m,c)}$ are SPD matrices. According to Theorem 1, $\mathbf{F}^{(m)}$ is an SPD matrix. So $\mathbf{F}$ is also an multi-channel SPD matrix. ■

Appendix C
Proof of Theorem 3

Proof: We take $\exp(\cdot)$ as an example. Let $\mathbf{X} = [X_{ij}]_{D\times D}$ denote an SPD matrix, then the element-wise activation result can be denoted as:

$$\exp(\mathbf{X}) = \begin{bmatrix} e^{X_{11}}, & e^{X_{12}}, & \ldots, e^{X_{1D}} \\ e^{X_{21}}, & e^{X_{22}}, & \ldots, e^{X_{2D}} \\ & \cdots & \\ e^{X_{D1}}, & e^{X_{D2}}, & \ldots, e^{X_{DD}} \end{bmatrix}, \tag{22}$$

$$= \begin{bmatrix} \sum_{i=0}^{\infty} \frac{X_{11}^i}{i!}, & \sum_{i=0}^{\infty} \frac{X_{12}^i}{i!}, & \ldots, \sum_{i=0}^{\infty} \frac{X_{1D}^i}{i!} \\ \sum_{i=0}^{\infty} \frac{X_{21}^i}{i!}, & \sum_{i=0}^{\infty} \frac{X_{22}^i}{i!}, & \ldots, \sum_{i=0}^{\infty} \frac{X_{2D}^i}{i!} \\ & \cdots & \\ \sum_{i=0}^{\infty} \frac{X_{D1}^i}{i!}, & \sum_{i=0}^{\infty} \frac{X_{D2}^i}{i!}, & \ldots, \sum_{i=0}^{\infty} \frac{X_{DD}^i}{i!} \end{bmatrix}, \tag{23}$$

$$= \mathbf{1} + \mathbf{X} + \frac{1}{2}\mathbf{X} \circ \mathbf{X} + \frac{1}{3!}\mathbf{X} \circ \mathbf{X} \circ \mathbf{X} + \cdots, \tag{24}$$

where $\circ$ means Hadamard product of two matrices. According to the Schur product theorem, $\mathbf{X} \circ \mathbf{X} \cdots \mathbf{X}$ is an SPD matrix. So $\exp(\mathbf{X})$, which equals the summation of multiple positive

definite matrices and semi-positive definite matrices, is an SPD matrix. Similarly, we can prove that

$$\sinh(\mathbf{X}) = \sum_{i=0}^{\infty} \frac{\mathbf{X}^{2i+1}}{(2i+1)!} \tag{25}$$

and

$$\cosh(\mathbf{X}) = \sum_{i=0}^{\infty} \frac{\mathbf{X}^{2i}}{(2i)!} \tag{26}$$

are also SPD matrices where $\mathbf{X}^i$ means element-wise power here. ■

APPENDIX D
PROOF OF THEOREM 4

Proof: SPD recursive layer mainly contains three kinds of operations which are bilinear projection with diagonal bias, non-linear activation functions including $\exp(\cdot)$ and $\sinh(\cdot)$, and Hadamard product. According to the definition of SPD, Theorem 3 and Schur product theorem respectively, these three operations can be easily proved to preserve symmetric positive definiteness. Moreover, as $\exp(\cdot)$ preserves symmetric positive definiteness and $\forall \mathbf{X}$, we have $\max(\exp(\mathbf{X})) > 0$, so $\sigma_g(\mathbf{X})$ also preserves symmetric positive definiteness.

Then we can prove Theorem 4 with mathematical induction. When $t = 1$, the initial hidden state denoted as $\mathbf{H}_0$ is set to be zero matrix. Then $\mathbf{R}_1$ can be rewritten as

$$\mathbf{R}_1 = \sigma_g(\mathbf{W}_{fr}^T \mathbf{F}_1 \mathbf{W}_{fr} + b_r + \epsilon * \mathbf{I}), \tag{27}$$

$$\mathbf{Z}_1 = \sigma_g(\mathbf{W}_{fz}^T \mathbf{F}_1 \mathbf{W}_{fz} + b_z + \epsilon * \mathbf{I}), \tag{28}$$

$$\widetilde{\mathbf{H}}_1 = sinh(\mathbf{W}_{fh}^T \mathbf{F}_1 \mathbf{W}_{fh} + b_h + \epsilon * \mathbf{I}). \tag{29}$$

As $\mathbf{F}_1$ is SPD, then apparently $\mathbf{R}_1, \mathbf{Z}_1, \widetilde{\mathbf{H}}_1$ are all SPD. Thus $\mathbf{H}_1 = \widetilde{\mathbf{H}}_1$ is also SPD.

Then $\forall t \in [2, \ldots, T]$, if $\mathbf{H}_{t-1}$ is SPD, similar to the situation of $t = 1$, we can also prove that $\mathbf{R}_t$, $\mathbf{Z}_t$, $\widetilde{\mathbf{H}}_t$ and $\mathbf{H}_t$ in Eqn. (4), (5), (6) and (7) are SPD. This is because all the operations dealing with $\mathbf{H}_{t-1}$ and $\mathbf{F}_t$ preserve symmetric positive definiteness.

Thus, according to mathematical induction, the output states of SPD recursive layer, i.e., $\mathbf{H}_t, t = 1, \ldots, T$, are SPD. ■

REFERENCES

[1] B. B. Amor, J. Su, and A. Srivastava, "Action recognition using rate-invariant analysis of skeletal shape trajectories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 1–13, Jan. 2016.

[2] X. Cai, W. Zhou, L. Wu, J. Luo, and H. Li, "Effective active skeleton representation for low latency human action recognition," *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 141–154, Feb. 2016.

[3] M. Liu, H. Liu, and C. Chen, "Robust 3D action recognition through sampling local appearances and global distributions," *IEEE Trans. Multimedia*, vol. 20, no. 8, pp. 1932–1947, Aug. 2018.

[4] R. Anirudh, P. Turaga, J. Su, and A. Srivastava, "Elastic functional coding of human actions: From vector-fields to latent variables," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 3147–3155.

[5] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Log-Euclidean metrics for fast and simple calculus on diffusion tensors," *Magn. Reson. Medicine*, vol. 56, no. 2, pp. 411–421, 2006.

[6] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *Proc. Int. Workshop Human Behavior Understanding*, 2011, pp. 29–39.

[7] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[8] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, "Bio-inspired dynamic 3D discriminative skeletal features for human action recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, 2013, pp. 471–478.

[9] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[10] M. Devanne *et al.*, "3-D human action recognition by shape analysis of motion trajectories on Riemannian manifold," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1340–1352, Jul. 2015.

[11] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 1110–1118.

[12] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 20, no. 2, pp. 303–353, 1998.

[13] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: Human action recognition using joint quadruples," in *Proc. Int. Conf. Pattern Recognit.*, 2014, pp. 4513–4518.

[14] A. Goh and R. Vidal, "Clustering and dimensionality reduction on Riemannian manifolds," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2008, pp. 1–7.

[15] M. A. Gowayyed, M. Torki, M. E. Hussein, and M. El-Saban, "Histogram of oriented displacements (HOD): Describing trajectories of human joints for action recognition," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013.

[16] M. Harandi, M. Salzmann, and R. Hartley, "Dimensionality reduction on SPD manifolds: The emergence of geometry-aware methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 48–62, Jan. 2018.

[17] M. T. Harandi, M. Salzmann, and R. Hartley, "From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices," in *Proc. Eur. Conf. Comput. Vision*, 2014, pp. 17–32.

[18] M. T. Harandi, C. Sanderson, R. Hartley, and B. C. Lovell, "Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach," in *Proc. Eur. Conf. Comput. Vision*, 2012, pp. 216–229.

[19] T. Zhang, W. Zheng, Z. Cui, and C. Li, "Deep Manifold-to-manifold transforming network," in *Proc. IEEE Int. Conf. Image Process.*, 2018, pp. 4098–4102.

[20] Z. Huang and L. J. Van Gool, "A Riemannian network for SPD matrix learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2036–2042.

[21] Z. Huang, C. Wan, T. Probst, and L. Van Gool, "Deep learning on lie groups for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 6099–6108.

[22] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, vol. 13, pp. 2466–2472.

[23] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, "Kernel methods on the Riemannian manifold of symmetric positive definite matrices," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2013, pp. 73–80.

[24] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, "Kernel methods on Riemannian manifolds with Gaussian RBF kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2464–2477, Dec. 2015.

[25] Z. J. Koles, M. S. Lazar, and S. Z. Zhou, "Spatial patterns underlying population differences in the background EEG," *Brain Topography*, vol. 2, pp. 275–284, 1990.

[26] P. Koniusz, A. Cherian, and F. Porikli, "Tensor representations via kernel linearization for action recognition from 3D skeletons," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 37–53.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Advances Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[28] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 816–833.
[29] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention LSTM networks for 3D action recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1647–1656.
[30] M. Müller T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation mocap database HDM05," Dept. Inst. Comput. Sci. II, Univ. Bonn, Bonn, Germany, Tech. Rep. CG-2007-2, Jun. 2007.
[31] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition," in *Proc. Comput. Vision Pattern Recognit. Workshops*, 2012, pp. 8–13.
[32] O. Oreifej and Z. Liu, "HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2013, pp. 716–723.
[33] Y. Pang, Y. Yuan, and X. Li, "Gabor-based region covariance matrices for face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 7, pp. 989–993, Jul. 2008.
[34] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," *Int. J. Comput. Vision*, vol. 66, pp. 41–66, 2006.
[35] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 2006, vol. 1, pp. 728–735.
[36] L. Seidenari, V. Varano, S. Berretti, A. Bimbo, and P. Pala, "Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, 2013, pp. 479–485.
[37] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+ D: A large scale dataset for 3D human activity analysis," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 1010–1019.
[38] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proc. Assoc. Advancement Artif. Intell.*, 2017, pp. 4263–4270.
[39] L. Tao and R. Vidal, "Moving poselets: A discriminative and interpretable skeletal motion representation for action recognition," in *Proc. IEEE Int. Conf. Comput. Vision Workshops*, 2015, pp. 61–69.
[40] D. Tosato, M. Farenzena, M. Spera, V. Murino, and M. Cristani, "Multi-class classification on Riemannian manifolds for video surveillance," in *Proc. Eur. Conf. Comput. Vision*, 2010, pp. 378–391.
[41] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Proc. Eur. Conf. Comput. Vision*, 2006, pp. 589–600.
[42] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on Riemannian manifolds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1713–1727, Oct. 2008.
[43] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1988–1996.
[44] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 588–595.
[45] 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. Vision*, 2015, pp. 4570–4578.
[46] P. Wang, C. Yuan, W. Hu, B. Li, and Y. Zhang, "Graph based skeleton motion representation and similarity measurement for action recognition," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 370–385.
[47] W. Wang, R. Wang, Z. Huang, S. Shan, and X. Chen, "Discriminant analysis on Riemannian manifold of Gaussian distributions for face recognition with image sets," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 2048–2057.
[48] L. Xia, C.-C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3D joints," in *Proc. Comput. Vision Pattern Recognit. Workshops*, 2012 pp. 20–27.
[49] Y. Yang *et al.*, "Discriminative multi-instance multitask learning for 3D action recognition," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 519–529, Mar. 2017.
[50] J. Zhang *et al.*, "A large scale RGB-D dataset for action recognition," in *Proc. Int. Conf. Pattern Recognit.*, 2016, pp. 101–114.
[51] S. Zhang, X. Liu, and J. Xiao, "On geometric features for skeleton-based action recognition using multilayer LSTM networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vision*, 2017, pp. 148–157.

Tong Zhang (Member, IEEE) received the B.S. degree from the School of Information Science and Engineering, Southeast University, Nanjing, China, in 2011, the M.S. degree from the Research Center for Learning Science, Southeast University, Nanjing, China, in 2014, and the Ph.D. degree from the School of Information Science and Engineering, Southeast University, in 2018. He is currently a Lecturer with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. His current research interests include pattern recognition, machine learning, and computer vision.

Wenming Zheng (Senior Member, IEEE) received the B.S. degree in computer science from Fuzhou University, Fuzhou, China, in 1997, the M.S. degree in computer science from Huaqiao University, Quanzhou, China, in 2001, and the Ph.D. degree in signal processing from Southeast University, Nanjing, China, in 2004. Since 2004, he has been with the Research Center for Learning Science, Southeast University, where he is currently a Professor with the School of Biological Science and Medical Engineering and the Key Laboratory of Child Development and Learning Science of the Ministry of Education. His current research interests include affective computing, pattern recognition, machine learning, and computer vision. He was an Associate Editor of several peer-reviewed journals, such as the IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, *Neurocomputing*, and *Visual Computer*.

Zhen Cui (Member, IEEE) received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2014. He was a Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, from 2014 to 2015. He also spent half a year as a Research Assistant with Nanyang Technological University from June 2012 to December 2012. He is currently a Professor with Nanjing University of Science and Technology, Nanjing, China. His research interests include computer vision, pattern recognition and machine learning, especially focusing on deep learning, manifold learning, sparse coding, face detection/alignment/recognition, object tracking, image super resolution, emotion analysis, etc. He has authored or coauthored several papers in the top conferences NIPS/CVPR/ECCV and some IEEE TRANSACTIONS.

Yuan Zong (Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Nanjing Normal University, Nanjing, China, in 2011 and 2014, respectively, and the Ph.D. degree in biomedical engineering in 2018 from Southeast University, Nanjing, China, where he is currently a Lecturer with the Key Laboratory of Child Development and Learning Science of Ministry of Education, School of Biological Science and Medical Engineering. From 2016 to 2017, he was a Visiting Student with the Center for Machine Vision and Signal Analysis, University of Oulu, Finland. His research interests include affective computing, pattern recognition, and computer vision.

Chaolong Li received the B.S. and M.S. degrees from the Key Laboratory of Child Development and Learning Science of the Ministry of Education, School of Biological Science and Medical Engineering, Southeast University, Nanjing, China, in 2016 and 2019, respectively. He is currently an AI Engineer with the IFLYTEK CO, China. His research interests include computer vision, pattern recognition, and deep learning.

Xiaoyan Zhou received the B.S. degree in communication engineering from Hohai University, Changzhou, China, in 1997, and the M.S. and Ph.D. degrees in signal and information processing from Southeast University, Nanjing, China, in 2005 and 2011, respectively. She is currently an Associate Professor of electrical and information engineering with the Nanjing University of Information and Science and Engineering, Nanjing, China. Her research interests include pattern recognition and machine learning.

Jian Yang (Member, IEEE) received the Ph.D. degree in pattern recognition and intelligence systems from Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002. In 2003, he was a Postdoctoral Researcher with the University of Zaragoza, Zaragoza, Spain. From 2004 to 2006, he was a Postdoctoral Fellow with the Biometrics Centre of Hong Kong Polytechnic University. From 2006 to 2007, he was a Postdoctoral Fellow with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA. He is currently a Chang-Jiang Professor with the School of Computer Science and Technology, NUST. He is the author of more than 200 scientific papers in pattern recognition and computer vision. His papers have been cited more than 5000 times in the Web of Science and 13 000 times in the Google Scholar. His research interests include pattern recognition, computer vision, and machine learning. He is an Associate Editor for *Pattern Recognition*, *Pattern Recognition Letters*, IEEE TRANSACTION ON NEURAL NETWORKS AND LEARNING SYSTEMS, and *Neurocomputing*. He is a Fellow of IAPR.