Hindawi
Security and Communication Networks
Volume 2022, Article ID 3064804, 10 pages
https://doi.org/10.1155/2022/3064804

Research Article

3D Deep Heterogeneous Manifold Network for Behavior Recognition

Jinghong Chen,[1,2] **Li Zhang**,[1,2] **Zhihao Jin**,[1,2] **Chong Zhao**,[1,2] **and Qicong Wang**[1,2]

[1]*Department of Computer Science and Technology, Xiamen University, Xiamen 361005, China*
[2]*Shenzhen Research Institute, Xiamen University, Shenzhen 518057, China*

Correspondence should be addressed to Chong Zhao; zhc@xmu.edu.cn and Qicong Wang; qcwang@xmu.edu.cn

Received 1 February 2022; Accepted 26 February 2022; Published 16 March 2022

Academic Editor: Lu Liu

With the broadening of application scenarios for Internet of Things, intelligent behavior recognition task has attracted more and more attention. Since human behavior is nonrigid motion with strong spatiotemporal topological association, modeling it directly with traditional Euclidean space-based methods may destroy its underlying nonlinearity. Based on the advantages of Riemannian manifold in describing 3D motion, we propose an end-to-end 3D behavior manifold feature learning framework composed of deep heterogeneous networks. This heterogeneous architecture aims to leverage the graph construction to guide manifold backbone network to mine more discriminative nonlinear spatiotemporal features. Therefore, we first model the nonlinear spatiotemporal co-occurrence of 3D behavior in the high-dimensional Riemannian manifold space. Secondly, we implement a non-Euclidean heterogeneous architecture on the Riemannian manifold so that the backbone network can learn deep spatiotemporal features while preserving the manifold topology. Finally, an end-to-end deep graph similarity-guided learning optimization mechanism is introduced to enable the overall model to fully utilize the complex similarity relationship between manifold features. We have verified our 3D deep heterogeneous manifold network on popular skeleton behavior datasets and achieved competitive results.

1. Introduction

Behavior recognition tasks [1–3] receive much attention due to the vigorous development of artificial intelligence and the rise of computer vision. In smart security, human-computer interaction, and immersive games, behavior recognition is playing an increasingly important role. We can perform dangerous behavior warnings, provide more convenient behavior instructions for human-computer interaction, and make immersive games have a rich and exquisite game experience through behavior recognition. With the great improvement of computer and devices for capturing the movement of human skeleton, the acquisition of skeleton sequence data is more convenient, which promotes the development of skeleton-based behavior recognition [4, 5]. The skeleton-based behavior recognition method has the advantages of eliminating the influence of the background and the invariance of the perspective, which brings the ability to pay more attention to the behavior itself. For these reasons, more and more researchers are involved in skeleton-based action recognition research.

There are three main methods of existing behavior recognition: methods based on spatial features of skeleton coordinates, methods based on temporal information of skeleton sequence, and methods based on spatiotemporal features. In the method based on spatial features of skeleton coordinates, the covariance matrix of the joint position trajectory is calculated to build the temporal model of skeleton sequence [2]. In [3], the paired relative positions of joints are also used to describe the posture and joint changes of the skeleton sequence, and the principal component analysis is applied to normalize features to obtain the representation of the principal features. In [4], the rotation and translation between body parts are used as features, and the Fourier temporal pyramid (FTP) is utilized to model the temporal dynamics. These methods pay more attention to

the spatial relationship of the joints in the skeleton behavior, which weakens the attention to the temporal features to a certain extent.

For the temporal information, Wang et al. [1] calculate relative positions of each joint and other joints to represent each frame of the skeleton sequence and then model temporal information. In [6], the histogram of the 3D joint position is calculated to represent each frame of the skeleton sequence, and HMMs are used to model the temporal dynamics. Kim and Reiter [7] propose to use temporal convolutional neural network (TCN) for 3D human behavior recognition. Compared with the popular LSTM-based recurrent neural network model, the TCN-based model is more intuitive and interpretable [7]. These methods can take the spatiotemporal features of behavior into account, but may ignore some spatial features that are globally related and cannot closely link temporal and spatial features.

In the method based on spatiotemporal features, Yan et al. [8] design skeleton sequence graph containing temporal information and use the spatiotemporal graph convolution network to learn the spatiotemporal features in the behavior sequences. Ke et al. [9] use a deep convolutional neural network to obtain the temporal features of the skeleton sequence, use a multitask learning network to process all the frames of the generated fragments, and finally combine the spatial information for behavior recognition. Some scholars use graph convolutional network (GCN) combined with LSTM or dual-stream network structure [5, 10–12] to extract spatiotemporal information in behavior sequences. These methods can pay attention to the close relationship between temporal and spatial features, but since behavior features also have the temporal and spatial co-occurrence, these methods cannot accurately describe this property.

To learn more discriminative spatiotemporal manifold features by the deep model, we need to comprehensively consider the spatiotemporal co-occurrence relationships between the connected and disconnected skeleton parts. To this end, we intend to represent the spatial structure based on the transformation group for each frame of original nonrigid 3D skeleton behavior sequences and use the Riemannian manifold to construct the relative spatial transformation relationships between all pairs of skeleton parts. This spatial structure representation method can describe the relative motion relationship between all pairs of skeleton parts in a frame as a point in the high-dimensional Riemannian manifold space.

Since each action sequence consists of many frames, we employ an interpolation method based on the transformation group to integrate the points in the manifold surface space into a transformation group curve, so as to model the co-occurrence relationship of the spatiotemporal features of original 3D skeleton sequence. However, directly inputting features with manifold constraints into neural network will bring high time and space complexity. Currently, it is difficult to use the neural network to mine rich information contained in manifold input while preserving the manifold constraints. To this end, Wang et al. [13] propose a GCN-based method to solve the problem of edge prediction between nodes. Inspired by this method, we try to treat an action as node, construct similarity graph of all nodes based on its manifold trajectory, use graph convolution to predict connections, and finally achieve the classification of behaviors. With respect to this idea, the difficulty to be solved is how to construct graph of feature nodes in manifold space.

The graph construction method is currently commonly used in determining the similarity of members in social network analysis [14, 15], and the constructed graph is used for intelligent recommendation. In these applications, the multidimensional features of the task are usually data in Euclidean space, and existing methods such as KNN [16] can solve this problem. However, in the application scenario of our problem, we hope to realize the construction of behavior feature nodes on manifold space. Therefore, in this study, a graph construction method based on the Riemannian metric on manifold is proposed. This method can take full advantage of rich information of data on manifold. At the same time, the Riemannian metric method can map behavior nodes isometrically into projected space.

This study proposes a 3D behavior recognition method based on spatiotemporal trajectory graph construction, whose description of framework is shown in Figure 1. This method uses Riemannian metric to measure the spatiotemporal trajectory properties, which make similar nodes closer and dissimilar or different types of nodes far apart. The model mainly has the following stages, data preprocessing, Riemannian metric graph construction, graph convolution, and behavior classification. In the data preprocessing stage, we process the 3D coordinate data of the skeleton sequences into a behavior trajectory curve representing relative behavior relationship between any pair of bones. In order to express as much spatial information as possible to reflect rich spatiotemporal co-occurrence, we calculate the relative behavior relationship between any two bones. In the stage of Riemannian metric graph construction, we roll and expand the processed manifold spatiotemporal trajectory curve along the direction of the trajectory into a corresponding continuous rolling tangent space curve. This process tries to ensure that the distance between any two points in a tangent space curve is equivalent to the distance between two points in the original manifold, use DTW to measure the similarity between curves, and use the similarity between behavior nodes to construct a similarity graph. In the graph convolution stage, through the update between each iteration of graph convolution, similar nodes are pulled closer and different are pushes apart so that behavior nodes of the same category are gathered together. Finally, in the classification stage, the labels are spread from the central point of each cluster to achieve the classification of behaviors. The main contributions of this study are as follows:

(1) For skeleton sequences, we extract rotation and translation relationships from bone pairs and represent them as discrete trajectories in Riemannian manifold, which can describe spatiotemporal co-occurrence and global relative relationships.

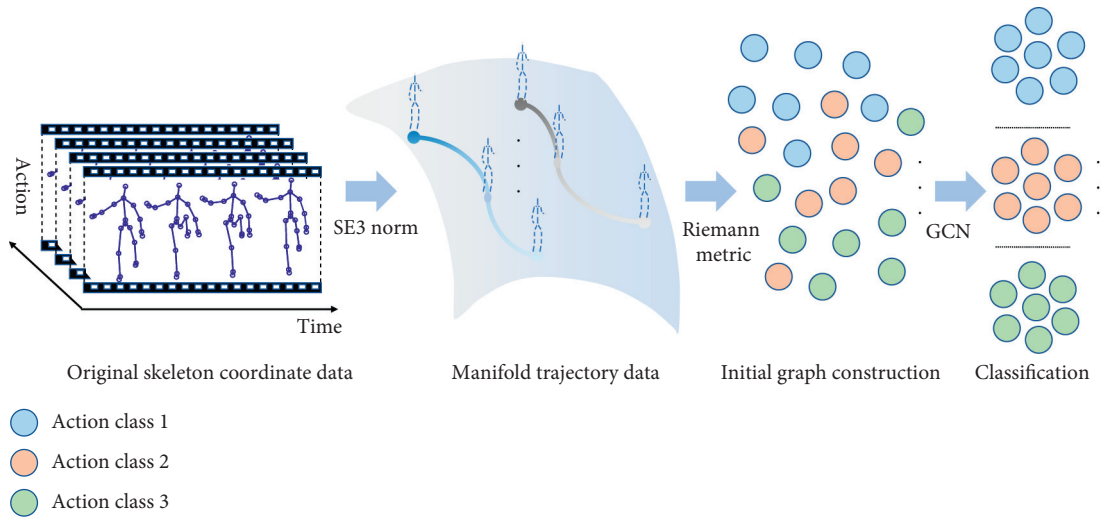

Figure 1: Framework of 3D behavior recognition network based on spatiotemporal trajectory graph construction.

(2) We propose a graph construction method based on continuous projections on Riemannian manifold, which is employed to map the spatiotemporal trajectories on the manifold isometrically to preserve more complex similarity distribution relationship between manifold features.

(3) We propose a deep heterogeneous manifold model consisting of two subnetworks with different structures. It incorporates an end-to-end optimizable manifold backbone network, which exploits the powerful representative ability of Riemannian manifold and can be guided by the subsequent graph-based subnetwork.

2. Spatiotemporal Manifold Trajectory Representation

To fully exploit the nonlinearity of behavior data, we represent them as curves in manifold space. Specifically, we represent it in the Lie manifold SE_3 in the form of Cartesian product, which can contain rich spatiotemporal co-occurrent relationships.

Given 3D coordinates of the joints of the skeleton behavior sequence, we assume that the number of frames of an behavior sequence is F, and the number of joints is N_j, so the coordinate of the nth joint in the frame f is expressed as $X_n^f = (x_n^f, y_n^f, z_n^f)$, and the 3D coordinate of a behavior sequence is represented as $\{X_n^f | n = 1, \ldots, N_j; f = 1, \ldots, F\}$. With these 3D coordinates and the body structure data given in the dataset, i.e., the above joint points are connected in the body structure, here we might as well assume that the joint i and the joint j are the two ends of the bone B_{ij} in the first frame, and this bone can be represented as $B_{ij} = X_i^1 - X_j^1 = (x_i^1 - x_j^1, y_i^1 - y_j^1, z_i^1 - z_j^1)$; in this way, a bone can intuitively be represented as a vector in 3D Euclidean space, and the set of bones $\{B_{ij}^f | 1 < i < j < N_j; f = 1, \ldots, F\}$ can also be obtained. Since the spatiotemporal graph of the body structure in the current skeleton data are all acyclic graphs, the number of bones is $N_j - 1$. In the skeleton of body, the relationship between any two different bones is $(N_j - 1) * (N_j - 2)$ pairs.

The elements in the trajectory manifold have the following constraints:

$$SE_3 = \left\{ T = \begin{bmatrix} R & d \\ 0^T & 1 \end{bmatrix} \in \Re^{4\times 4} | R \in SO_3, d \in \Re^3 \right\}, \quad (1)$$

where SE_3 is special Euclidean group and SO_3 is special orthogonal group.

The manifold trajectory using relative relationships has the following advantages:

(1) The features used to represent the rotation relationship between skeletons are scale invariant; in other words, no matter how large the scale is to represent the skeleton, the rotation relationship between the skeletons is unchanged

(2) The relative relationship of SE_3 has spatial co-occurrence, i.e., we can explore the relationship between not only any two bones but also spatially connected skeleton pairs

(3) Representing the relative relationship of the skeleton based on the trajectory curve can closely combine the spatial information and the temporal information, so different spatial features can be represented point by point to form a discrete curve on manifold space, which helps to increase the similarity of features with the similar temporal information

3. Backbone Network of Deep Heterogeneous Manifold Network

3.1. Riemannian Manifold Preservation Network. Since the input data of our deep Riemannian manifold network is the initialized high-dimensional Riemannian manifold transformation group data, it is necessary to maintain the richness and topology of their nonlinear structures during the feature learning process. The commonly used Euclidean

spatial convolution layer may destroy this property, so we employ a convolution-like Riemannian transform layer that contains transform parameters optimized for deep model learning and whose output still conforms to the Riemannian manifold constraints, which preserve the Riemannian manifold topology of the data.

According to the above description, we know that the feature is a set of points in the motion group SE_3, which is represented by the discrete curves' form on the manifold of the Lie group [17, 18]. We denoted this manifold as $\mathscr{M}$, and the set of points is $\mathbb{S}$; then, the feature of the fth frame in the kth behavior is represented as $\mathbb{S}_f^k$. Since any point on the manifold $\mathscr{M}$ has constraints: if we have any $U \in \mathscr{M}$, then $U \cdot U^T = I$ and $\det(U) = 1$, where I is the identity matrix, which is also the identity element on the manifold, and det is the operation to find the value of the determinant. So, there is

$$\mathrm{SE}_n = \left\{R \in \mathbb{R}^{n\times n} | R^\top R = I_n, \det R = 1\right\}. \tag{2}$$

If we have $V \in \mathscr{M}$, then $V \cdot U \in \mathscr{M}$.

This property can be summarized as

$$\mathrm{SE}(3) \times \mathrm{SE}(3) \longrightarrow \mathrm{SE}(3). \tag{3}$$

The SE_3 matrix has the invertible property $R^{-1} = R^\top$. Therefore, the behavior trajectory curve l is in the form of $\mathrm{SE}(3) \times \mathrm{SE}(3) \times \ldots \times \mathrm{SE}(3)$.

The initialized high-dimensional Riemannian manifold transformation group data are also a spatiotemporal co-occurrence representation of the original 3D data, thus requiring spatial and temporal pooling techniques on the Riemannian manifold. We can not only reduce the data dimension and preserve topology but also further obtain more discriminative spatiotemporal manifold features between action sequence frames.

3.2. Graph Construction Based on Manifold Trajectory. On the obtained manifold trajectory curves, we use the Riemannian similarity metric method to construct graph for the behavior features on Riemannian manifold. The distance on a manifold is obtained by measuring geodesics on the manifold. To ensure that the distance between any two points on the manifold remain constant in the constructed graph, we can map the points on the manifold isometrically to a convenient measurement space. The implementation process of the graph construction method based on Riemannian similarity metric is shown in Algorithm 1.

The dimension of the SE_3 matrix is 6, which brings high computational and space complexity to operations such as multiplication and inverse. Therefore, in this study, we do not use the method of directly calculating the distance between two points on the SE_3 manifold. We explore the use of a certain method that can isometrically map the points on the manifold to a space that is convenient for measurement. If we directly expand the projection at a point, for example, we expand at the pole, the result may be that the closer to the pole, the more similar the curve after projection is to the original curve on the manifold, and the farther away from the pole, the more distorted the curve is after projection. Inspired by methods of geodetic distance [19], we propose a method for measuring the distance of a curves on manifold based on a continuous projection.

Figure 2 shows a continuous projection of a behavior trajectory curve on the manifold along the quasi-average curve to its corresponding tangent space. In the curve l_{ABC} on the manifold, we use the continuous projection method along the average curve of the class (i.e., the dotted line in the figure) to project the points on the curve one by one into the tangent space. The lengths of line segments l_{AB}, l_{BC}, and l_{AC} on the manifold are, respectively, equal to the lengths of l_{ab}, l_{bc}, and l_{ac} of the corresponding tangent space.

Below, we explain this continuous projection process in detail. Specifically, the continuous projection mapping on the manifold is a smooth mapping h: along a smooth average curve α: $[0, T] \longrightarrow \mathscr{M}$:

$$\begin{aligned} h\colon [0,T] &\longrightarrow SE_3 = SO_3 \propto \mathbb{R}^3, \\ t &\mapsto h(t) = (R(t), s(t)). \end{aligned} \tag{4}$$

In particular, this rolling continuous mapping needs to meet the three conditions defined in [20] at any time $t \in [0, T]$, namely, rolling conditions, no-slip conditions, and no-twist conditions. The continuous projection $h(t)$ is a continuous map that satisfies the above three conditions and maps the manifold trajectory to the corresponding tangent space.

Since the area near the point on the Lie group manifold is smooth, any point in this area can be represented by a slight rotation and translation change from a point to its neighbors. Assuming that P is a point on the manifold space of SE_3, α: $[0, \tau] \longrightarrow \mathrm{SE}_3, \alpha(t) = U(t)P_0W(t)^T$ is a curve on SE_3 starting from P_0 when $t = 0$, and at any subsequent time, you can find a point on the curve corresponding to that time. We can find such a smooth curve; then, this meets the continuous projection condition. Since our calculation cannot exhaust every point on the continuous curve, in order to facilitate the calculation, in the following calculation, we will continue to project the points on the curve frame by frame. Under the three constraints of manifold described above, this mapping process can be expressed as

$$\begin{aligned} h\colon [0,\tau] &\longrightarrow G = \mathrm{SE}_3 \times \mathrm{SE}_3 \propto \mathbb{R}^{4\times 4}, \\ t &\mapsto h(t) = \left(U^\top(t), W^\top(t), X(t)\right), \end{aligned} \tag{5}$$

where $\langle i$mes denotes semidirect product symbol and $(U^\top(t), W^\top(t), X(t))$ is the solution of the motion equation in the projection process at time t.

This process is a continuous projection V along the curve $\alpha(t)$ on the Lie group manifold V: $= T_{P_0}^{\mathrm{Aff}}\mathrm{SE}_3 \cong T_{P_0}\mathrm{SE}_3$; the curve $\alpha(t)$ has the following expression:

$$\alpha(t) = U(t)P_0W(t)^\top. \tag{6}$$

$\alpha_{\mathrm{dev}}(t)$ is the expansion of the curve $\alpha(t)$ under the effect of continuous projection $h(t)$ at P_0:

$$\alpha_{\mathrm{dev}}(t) = h(t)^{\circ}\alpha(\mathrm{t}) = \mathrm{U}^\top(\mathrm{t})\alpha(\mathrm{t})\mathrm{W}(\mathrm{t}) + \mathrm{X}(\mathrm{t}) = \mathrm{P}_0 + \mathrm{X}(\mathrm{t}). \tag{7}$$

Suppose we perform continuous projection in the time interval $[0, T]$ on a certain behavior curve. Since the curve on

Input: trajectory curves of all skeletons $\mathbb{S}$; behavior sequence label in training set L; total number of behavior categories M;
(1) **for** Given behavior category $L_i \in [L_1, L_M]$ **do**
(2) Calculate the average trajectory curve of each class on the manifold;
(3) Average trajectory curve $L_i^a vr = DT\ W$ (All train behavior curves $\in L_i$);
(4) **end for**
(5) **for all** Training trajectory curve $\mathbb{S}$ with label L_i **do**
(6) Continuously project training trajectory curve $\mathbb{S}$ along the average trajectory curve $L_i^a vr$,
(7) Obtain the curve features on the tangent space S_{train} after continuous projection;
(8) **end for**;
(9) **for all** Training trajectory curve $\mathbb{S}$ **do**
(10) Given test set trajectory curve $\mathbb{S}$
(11) **for** $i = 1$; $i < M$; $i++$ **do**
(12) Continuously project test set trajectory curve $\mathbb{S}$ along the average trajectory curve $L_i^a vr$;
(13) **end for**
(14) Continuously unfold test set trajectory curve $\mathbb{S}$ along the path of M average curves, obtain a set of curves $\{S_1, S_2 \ldots S_M\}$
(15) Calculate the set of similarity scores between each curve in the curve set and the corresponding average curve Score;
(16) Obtain the features S_{test} under the score reflecting to the highest similarity;
(17) **end for**;
(18) **for all** Training trajectory curve S **do**
(19) Given a curve S_{train} feature, use DTW to calculate the most similar K trajectory curve to this curve;
(20) Get adjacency list T_{train};
(21) **end for**;
(22) **for all** Test track curves S **do**
(23) Given a curve S_{test} feature, use DTW to calculate the most similar K trajectory curve to this curve;
(24) Get adjacency list T_{test};
(25) **end for**;
Output: Curve features of training set S_{train} and test set S_{test} after continuous projection; The adjacency list obtained of the training set T_{train} and test set T_{test};

Algorithm 1: Graph construction method based on Riemannian similarity metric.

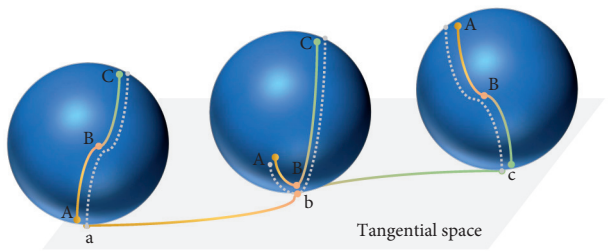

Figure 2: The behavior trajectory curve on the manifold is continuously projected along the curve to its corresponding tangent space.

the manifold we use is discrete on the time axis, we get the corresponding points in the mapping space. It is $\alpha_{\text{dev}}(t), t \in \{0, 1, 2, \ldots, T-1, T\}$.

Using the continuous projection method, the process of obtaining the similarity between the behavior curves from the manifold space is shown in Figure 3. We take the three points A, B, and C of a certain behavior curve on the manifold as an example. After continuous projection, they correspond to the three points a, b, and c in the tangent space. Our method aims to make the distances between AB, BC, and AC on the manifold are basically similar to the mapped distances ab, bc, and ac, especially to ensure that the distances between nodes of the same category are as similar as possible.

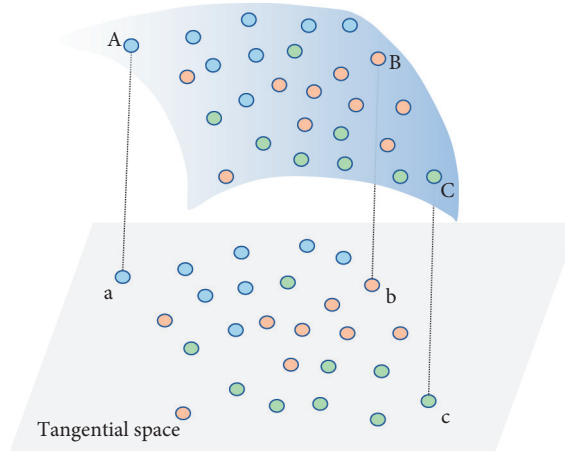

Figure 3: The distribution of nodes in manifold and in projection space by the continuous projection method.

The projection method based on the tangent space of a certain point has a problem, that is, the closer the data to the projection point, the better the retention of features and local similarities between the data. On the contrary, the farther away from the projection point is, the relative distance of the data is pulled away after being projected, which causes the local similarity of the data far from the projection point and the global similarity of the whole data to be destroyed. We keep the local similarity and global similarity between nodes

as much as possible in the projection process, avoiding the distortion of the distance between nodes that affects the subsequent node classification.

Generally, the behavior curves of a certain type on the manifold does not completely coincide with the geodesic. In particular, when this continuous projection curve satisfies certain constraints, the continuous projection curve we get degenerates into a geodesic curve. In a part of the projection of a certain point, the curve on the manifold and the two curves in the corresponding tangent space have the same geodesic curvature. That is to say, the geodesic curve is a projection curve that meets certain constraints, so the applicable range is narrow. Our continuous projection method can be applied to more manifold projection scenes; expanding average curve of a class along the behavior curve can better measure the similarity between different classes.

4. End-to-End Optimizable Graph-Guided Heterogeneous Model

In the previous 3D action recognition methods based on deep learning, most methods usually use a fully connected layer at the end of the backbone network and use cross-entropy loss to complete the task. In the iterative learning process, they do not fully consider the similarities and changes between deep features of similar actions as well as the differences between deep features of different action categories. Since the output of our backbone network is still topologically preserved Riemannian manifold data, we need a construction method of nearest neighbor graph on a high-dimensional Riemannian manifold surface to model local similarities, combined with graph convolutional network to achieve deep global similarity prediction to guide the feature learning of backbone network. This can make full use of the potential local similarity relationship in the local context information of each action sequence so that the whole heterogeneous network can integrate the common features of the same category and suppress their changes and at the same time expand the differences of different categories through the aggregating capability of graph convolution.

Our deep heterogeneous manifold network consists of two subnetworks with different structures. The former is the backbone network for learning deep manifold spatiotemporal features, and the latter is the graph convolution-guided learning subnetwork, which is built on the previous trajectory curves. In the backbone network, two pooling learning submodules are added to learn more discriminative features for further promoting of the graph convolutional network. In an end-to-end manner, the latter subnetwork can guide the feature learning of the former backbone subnetwork. However, its backpropagation will be more complicated, and the whole heterogeneous model is built on the Riemannian manifold, making the optimization problem with manifold constraints. If the manifold is embedded in linear space, the dimension problem will increase, thereby increasing the complexity. It is very difficult to optimize in Euclidean space. However, in some specific Riemannian manifold, the constraints can be eliminated to become unconstrained optimization, so we consider to solve an end-to-end optimization problem directly on the Riemannian manifold.

In the first module of the trajectory curve feature learning part, we set the learning parameter R_S in a Lie group manifold and then perform a spatial pooling on the data that has undergone manifold learning so that we can select more discriminative spatial features learned by the previous layer, and it reduces the computational complexity of spatial features and facilitates the subsequent computation. Similarly, the second module also sets a learning parameter R_T in the Lie group manifold and then performs a temporal pooling on the data. In this way, on the one hand, it is possible to select more discriminative temporal features after learning from the previous layer, and on the other hand, it reduces the computational complexity of temporal features.

Given $R_S \in \mathrm{SE}_3$ and $R_T \in \mathrm{SE}_3$, we suppose that the data passed in each time are $D \in \mathrm{SE}_3$. Due to the retention of Lie group operations, there is

$$\begin{aligned} D \cdot R_S &\in \mathrm{SE}_3, \\ D \cdot R_T &\in \mathrm{SE}_3. \end{aligned} \tag{8}$$

Therefore, in this part, the network parameters' learning is constrained in the Lie group manifold. In the graph-guided convolution module, we loop all behavior nodes, put all nodes into a queue, construct a domain subgraph with each node as the central point, and predict the connection relationship between the included peripheral nodes and the central point. As a result, a set of edges whose weights are the connection probability can be obtained. In order to cluster similar nodes together, a simple method is to prune all edges whose weights are lower than a certain threshold and use breadth-first search method to propagate pseudolabels. In each iteration, the edge is updated below a certain threshold, and in the next iteration, the connected clusters are greater than the predefined maximum value. In the next iteration, the threshold for updating the edge is increased. Repeat this loop process until the queue is empty. At this time, all nodes have been marked with pseudolabels of the category. We take the label of the central node of each cluster to propagate, i.e., the classification of nodes is realized.

5. Experimental Verification

5.1. Dataset Description

5.1.1. G3D Dataset. This dataset is a skeleton-based dataset [21] collected from game data. It contains 10 participants, who perform 20 categories of game behaviors. Most behavior sequences are recorded by a specific camera in a controlled indoor environment. Participants perform basic behaviors in strict accordance with instructions, and each sequence was repeated 3 times by each subject. Nevertheless, participants are free to complete the collection of different exercise sequences according to their own exercise habits. The dataset contains manually labeled behavior category labels for all sequences.

The skeleton in this dataset consists of 20 joints, and the position of the participant's joints is expressed in X, Y, and Z

coordinates in meters. The skeleton data also includes a joint tracking state, including accurately tracked joints, imported joint coordinates, and predicted joint coordinates. In many cases, the predicted joints are accurate, but in some cases, the limbs are occluded and the predicted joints may be inaccurate. Since some joint points in the dataset are obtained through prediction, the accuracy of the final classification will be affected to a certain extent if the predicted joints are inaccurate.

5.1.2. HDM05 Dataset. The behavior sequences in this dataset are performed by 5 nonprofessional actors [22]. Most of the behavior sequences are performed multiple times by all five actors according to the specific instructions in the script. The script contains five parts, and each part is divided into several scenes. Each behavior sequence is only collected in the corresponding single scene. The skeleton in this dataset consists of 31 joints, and the 3D coordinates of the joints are represented in X, Y, and Z coordinates in centimeters.

Although the dataset is small in scale, the behavior categories are more detailed, with a total of 130 behavior categories, some of which may look similar. Therefore, this dataset is also somewhat challenging.

5.1.3. NTU-RGBD Dataset. The NTU-RGBD dataset contains 60 behavior classes and 56880 video samples [23]. This dataset contains RGB video, depth mapping sequence, 3D bone data, and infrared (IR) video for each sample. Each data is captured simultaneously by 3 Kinect V2 cameras. Here, we use three-dimensional skeleton data, and the three-dimensional coordinates of the joints are expressed in X, Y, and Z coordinates. The three-dimensional skeleton data contain the three-dimensional coordinates of 25 human body joints per frame. The original benchmark provides two evaluation methods, namely, cross-subject (CS) and cross-view (CV) evaluation. In CS evaluation, the training set contains 40,320 videos from 20 subjects, and the remaining 16,560 videos are used for testing. In CV evaluation, 37920 videos captured from No. 2 and No. 3 cameras were used for training, and the remaining 18,960 videos from No. 1 camera were used for testing.

This dataset is widely used in skeleton-based behavior recognition. It has several scene categories, including daily behaviors, medical scenes, and multiperson sports. Since it contains both single-person sequences and multiperson interaction sequences, it is quite challenging to perform recognition tasks on this dataset.

Table 1 summarizes the main data distribution characteristics of the above three datasets. It can be seen that the number of joints and the number of bones selected in the three datasets are roughly similar, and the number of frames in each behavior sequence varies widely, ranging from a few frames to a few hundred frames, i.e., it is linearly adjustable within certain limits. From this perspective, it is very important to fully dig out the temporal information to complete the task of behavior recognition. Judging from the number of behavior sequences contained, the scales of the three datasets from small to large are G3D-Gaming, HDM05, and NTU-RGBD; from the perspective of the divided behavior categories, HDM05 has the most behavior categories, indicating the classification of behavior sequences is finer, and the corresponding recognition difficulty is also greater. In addition, in order to further improve the generalization ability of recognition in the future, we have implemented a behavior recognition data acquisition system with multichannel video input. The system can be connected to the mainstream RGBD cameras on the market, and the number of channels is linearly adjustable within a certain range. The collected videos can be processed into the current major formats, for example, AVI, MPEG, and MP4. We can estimate the 3D skeleton sequences as datasets from the collected video data.

In the G3D dataset and HDM05 dataset, we follow the principle of cross-validation experiment, using half of the dataset for training and the remaining half for testing. The experimental settings of the NTU dataset adopts the commonly used cross settings, including the cross subject and cross view. In order to keep the number of frames consistent for all behavior sequences, we downsample the execution frames of the skeleton sequences so that each dataset has a fixed number of frames. The number of frames selected for the G3D dataset is 100, the HDM05 dataset is 300, and the NTU dataset is 300. For the three datasets, we apply similar normalization preprocessing to achieve the invariance of position and view changes.

5.2. Experiment and Comparative Analysis. We first test the classification result of the proposed method on the G3D dataset. The 663 sequences in the dataset are divided into the training set and the test set according to the participating objects. The behavior sequences performed by the participants 1, 3, 5, 7, and 9 are used as the training set, and the behavior sequences performed by the remaining participants are used as the test set; thus, 333 training set sequences and 330 test set sequences are obtained.

Due to the small size of the dataset, we consider that the number of neighbor nodes' set when constructing the graph is relatively small. In the update process of graph convolution, around each node, the closest node and the 11 closest nodes around it are selected. Initially, they are considered to be of the same class, and then, the edge weights are updated.

The experimental results on G3D dataset are shown in Table 2. From the data in the table, it can be seen that the proposed method has better performance than the previous methods. The reason is that the previous method directly expands the manifold data and inputs them into the network for learning. In this process, some manifold constraints are destroyed, making the latter network unable to mine the rich information originally contained on the manifold data. The proposed method continuously projects manifold curves into the corresponding projection space along the average curve of the class, which can keep the distance between the curves projected from manifold curves as consistent as possible. In this way, the subsequent graph convolution can use the similarity between the projected curves to classify.

Table 1: Datasets' summary.

Datasets	Class	Sequence	Joint	Frame	Subject
G3D-gaming	20	663	20	6–330	10
HDM05	130	2343	31	50–721	5
NTU-RGBD	60	56 880	25	50–300	40

Table 2: Performance comparison on the G3D dataset.

Methods	Accuracy (%)
RBM + HMM [24]	86.4
SE3 + FTP [4]	87.23
SO3 [25]	87.95
SO3 + deep [26]	89.10
Ours	90.69

The proposed method has an improvement of 1.59% compared with the method combining deep neural network. This is due to the fact that the spatiotemporal trajectory can mine more abundant co-occurrent features, and using these features, we can achieve better similarity construction. Graph convolution network in the following can improve the classification result through pulling similar nodes closer and pushing others far apart.

In the HDM05 dataset, we randomly select half of the behavior sequences from each class as the training set and the remaining half as the test set. There are a total of 2343 behavior sequences in the dataset and 130 detailed behavior categories. Each category has an average of less than 20 behavior sequences. After dividing the training set and the test set, the training set and test set have about 10 behavior sequences for each category. Therefore, in the update process of graph convolution, one of the closest nodes around each node and the 7 closest nodes around it are selected.

The experimental results on the HDM05 dataset are shown in Table 3. The proposed method is compared with the method that only uses the manifold learning. There is about 20% improvement. We reckon that the continuous projection method based on the manifold curve can learn the features that contain rich spatiotemporal co-occurrence from the manifold data, and the similarity graph between behavior nodes is better constructed; thus, the graph convolution method can be used for further similarity learning. In this process, the method based on continuous projection can maintain the similarity between curves, especially the similarity between curves of the same category. This step is a key step to connect the manifold data and the deep network.

Compared with some methods using deep learning, such as PB-GCN [28], our method also has a certain improvement. The reason may be that the conventional deep learning network just arranges the data according to a certain dimension. For example, the data separated into different body parts are sent to the network for learning. In this process, the local behavior information of most of the skeleton coordinates can be used, but it is difficult to learn the essential complicated features of the relative relationship of the movement in the network. Nonetheless, the proposed network can use this information by learning the features of the manifold trajectory.

Table 3: Performance comparison on HDM05 dataset.

Methods	Accuracy (%)
SPDNet [27]	61.45
SE3 + FTP [4]	70.26
SO3 [25]	71.31
SO3 + deep [26]	75.78
PB-GCN [28]	88.17
Ours	90.05

In NTU-RGBD dataset, we conduct training and testing according to the currently commonly used data division and conduct subject-cross and view-cross experiments, respectively. Due to the large number of behavior sequences for each category in the dataset, each node cannot be directly connected to its peers when constructing a graph. When constructing the graph, 200 nearest neighbor nodes of each node are selected to form the adjacency list. In the update process of graph convolution, one of the closest nodes around each node and the 20 closest neighbors around it are selected.

The experimental results on the NTU dataset are shown in Table 4. The proposed method is greatly improved compared to the method that only uses the Lie group. The reason is that, after the graph construction by continuous projection, the introduced graph convolution module can leverage backpropagation to enhance the learning ability of the Lie group. Compared with some existing deep learning methods such as Deep-LSTM [23], ST-LSTM [29], TCN [7], and GCA-LSTM [30], our method also has some advantages. When these methods are mining behavior sequences, the main focus is on one of the temporal features and spatial features, and our method can organically combine the temporal and spatial features of the behavior characteristics by means of the manifold behavior trajectory. Compared with the current mainstream behavior recognition methods HCN [31], ST-GR [32], ST-GR [32], and ST-GCN [8], our method is still comparable.

5.3. Ablation Study. In order to verify the effectiveness of the proposed method, we performed ablation experiments on HDM05 dataset to validate each module. We have done five experiments to compare the method of directly stretching the manifold data into European data (Stretch), the method of logarithmic mapping (LogMap), the method of continuous projection (Ours/G), and the continuous projection combined with graph convolution.

The results of the ablation experiments on the HDM05 dataset are shown in Table 5. It can be seen from the table that the result of directly stretching the manifold data into the Euclidean data is the worst. In this process, the constraints of manifold data are broken, so a large amount of spatiotemporal information contained is difficult to be utilized by subsequent networks. The logarithmic mapping method can retain part of the data constraints by projecting the data into the tangent space. After projection, the data can still express most of the spatiotemporal feature information. Compared with the logarithmic mapping

Table 4: Performance comparison on the NTU-RGBD dataset using cross-subject and cross-view protocol.

Methods	Accuracy	
	Xsub (%)	Xview (%)
Lie group [4]	50.1	82.8
Deep-LSTM [23]	60.7	67.3
ST-LSTM [29]	69.2	77.7
TCN [7]	74.3	83.1
GCA-LSTM [30]	74.4	82.8
HCN [31]	86.5	91.1
ST-GR [32]	86.9	92.3
ST-GCN [8]	81.5	88.3
DGNN [33]	87.5	94.3
Ours	85.3	93.8

Table 5: Comparison of ablation experiments on the HDM05 dataset.

Methods	Accuracy (%)
Stretch	69.34
Logmap	75.65
Ours/G	82.35
Ours	90.05

method, the method based on continuous projection still has a lot of improvement, which shows that the continuous projection maintains the stronger similarity of the data after the projection than the logarithmic mapping. Finally, the method of continuous projection combined with graph convolution achieves the best results, which shows that the graph convolution method used here can achieve the function of pulling similar nodes closer and pushing others far apart to improve the classification result of the algorithm.

6. Conclusion

In this study, a deep heterogeneous manifold network is proposed. It incorporates a graph construction method based on Riemannian metric, which can preserve the nonlinear constraints of the spatiotemporal trajectory to a large extent and obtain better data projection through continuous projection. The graph nodes of behavior sequences built by this method are input to graph convolutions to realize the clustering and classification, which can improve the classification result of behavior recognition. The whole architecture combines a manifold learning backbone subnetwork and a graph convolutional network. The two parts learn from each other through end-to-end optimization, and manifold-based graph construction can guide the manifold network. The proposed method has been validated on several mainstream skeleton-based datasets and achieved competitive results. In the future, we will investigate how to automatically learn features represented in Riemannian manifold from raw data, which will further improve the discriminativeness of Riemannian representations.

Data Availability

All datasets are public datasets that can be downloaded online. G3D dataset is publicly available at https://dipersec.king.ac.uk/G3D/G3D.html, NTU RGB + D dataset is publicly available at https://rose1.ntu.edu.sg/dataset/actionRecognition/, and HDM05 dataset is publicly available at https://resources.mpi-inf.mpg.de/HDM05/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Jinghong Chen and Li Zhang are contributed equally to this work.

Acknowledgments

This work was supported by the Shenzhen Science and Technology Programs under Grant nos. JCYJ20180306-173210774 and JCYJ20200109143035495.

References

[1] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1290–1297, IEEE, Providence, RI, USA, 16-21 June 2012.

[2] 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 *Proceedings of the Twenty-third international joint conference on artificial intelligence*, Beijing China, August 2013.

[3] X. Yang and Y. L. Tian, "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor," in *Proceedings of the 2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pp. 14–19, IEEE, Providence, RI, USA, 16-21 June 2012.

[4] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 588–595, Columbus, OH, USA, 23-28 June 2014.

[5] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: deep learning on spatio-temporal graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5308–5317, Las Vegas, NV, USA, 27-30 June 2016.

[6] L. Xia, C.-C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 20–27, IEEE, Providence, RI, USA, 16-21 June 2012.

[7] T. S. Kim and A. Reiter, "Interpretable 3d human action analysis with temporal convolutional networks," in *Proceedings of the 2017 IEEE Conference On Computer Vision And Pattern Recognition Workshops (CVPRW)*, pp. 1623–1631, IEEE, Honolulu, HI, USA, 21-26 July 2017.

[8] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in

Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, Seattle WA USA, October 2018.

[9] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3d action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3288–3297, Honolulu, HI, USA, 21-26 July 2017.

[10] R. Zhao, K. Wang, H. Su, and Q. Ji, "Bayesian graph convolution lstm for skeleton based action recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6882–6892, Seoul, Korea (South), 27 Oct.-2 Nov. 2019.

[11] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional lstm network for skeleton-based action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1227–1236, Long Beach, CA, USA, 15-20 June 2019.

[12] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12026–12035, Long Beach, CA, USA, 15-20 June 2019.

[13] Z. Wang, L. Zheng, Y. Li, and S. Wang, "Linkage based face clustering via graph convolution network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1117–1125, Long Beach, CA, USA, 15-20 June 2019.

[14] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.

[15] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[16] I. Mani and I. Zhang, "Knn approach to unbalanced data distributions: a case study involving information extraction," in *Proceedings of the Workshop On Learning From Imbalanced Datasets*, vol. 126, Menlo Park, CA, USA, August 2003.

[17] N. Boumal and P.-A. Absil, "A discrete regression method on manifolds and its application to data on so(n)," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 2284–2289, 2011.

[18] K. Hüper and F. Silva Leite, "On the geometry of rolling and interpolation curves on S n , SO n , and grassmann manifolds," *Journal of Dynamical and Control Systems*, vol. 13, no. 4, pp. 467–502, 2007.

[19] S. Banerjee, "On geodetic distance computations in spatial modeling," *Biometrics*, vol. 61, no. 2, pp. 617–625, 2005.

[20] R. Caseiro, P. Martins, J. F. Henriques, F. Silva Leite, and J. Batista, "Rolling riemannian manifolds to solve the multiclass classification problem," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 41–48, Portland, OR, USA, 23-28 June 2013.

[21] V. Bloom, D. Makris, and V. Argyriou, "G3d: A gaming action dataset and real time action recognition evaluation framework," in *Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 7–12, IEEE, Providence, RI, USA, 16-21 June 2012.

[22] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Kruger, and A. Weber, "Documentation mocap database hdm05," *Computer Graphics Technical Reports*, 2007.

[23] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1010–1019, Las Vegas, NV, USA, 27-30 June 2016.

[24] S. Nie and Q. Ji, "Capturing global and local dynamics for human action recognition," in *Proceedings of the 2014 22nd International Conference on Pattern Recognition*, pp. 1946–1951, IEEE, Stockholm, Sweden, 24-28 Aug. 2014.

[25] R. Vemulapalli and R. Chellapa, "Rolling rotations for recognizing human actions from 3d skeletal data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4471–4479, Las Vegas, NV, USA, 27-30 June 2016.

[26] Z. Huang, C. Wan, T. Probst, and L. Van Gool, "Deep learning on lie groups for skeleton-based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6099–6108, Honolulu, HI, USA, 21-26 July 2017.

[27] Z. Huang and L. Van Gool, "A riemannian network for spd matrix learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, February 2017.

[28] K. Thakkar and P. J. Narayanan, "Part-based graph convolutional network for action recognition," 2018, https://arxiv.org/abs/1809.04983.

[29] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," in *Proceedings of the European Conference on Computer Vision*, pp. 816–833, Springer, Amsterdam, The Netherlands, October 2016.

[30] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention lstm networks for 3d action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1647–1656, Honolulu, HI, USA, 21-26 July 2017.

[31] C. Li, Q. Zhong, D. Xie, and S. Pu, "Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation," 2018, https://arxiv.org/abs/1804.06055.

[32] B. Li, X. Li, Z. Zhang, and F. Wu, "Spatio-temporal graph routing for skeleton-based action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8561–8568, Honolulu, HA, USA, January 2019.

[33] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7912–7921, Long Beach, CA, USA, 15-20 June 2019.