This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TCDS.2022.3171550, IEEE Transactions on Cognitive and Developmental Systems

HMANet: Hyperbolic Manifold Aware Network for Skeleton-Based Action Recognition

Jinghong Chen, Chong Zhao, Qicong Wang, Hongying Meng, *Senior Member, IEEE*

***Abstract*—Skeleton-based action recognition has attracted significant attentions in recent years. To model the skeleton data, most popular methods utilize Graph Convolutional Networks to fuse nodes located in different parts of the graph to obtain rich geometric information. However, these methods cannot be generalized to different graph structures due to their dependencies on the input of the topological structure. In this paper, we design a novel Hyperbolic Manifold Aware Network without introducing a dynamic graph. Instead, it leverages Riemannian geometry attributes of hyperbolic manifold. Specifically, this method utilizes the Poincaré model to embed the tree-like structure of the skeleton into a hyperbolic space to automatically capture hierarchical features, which may explore the underlying manifold of the data. To extract spatio-temporal features in the network, the features in manifold space are projected to a tangent space, and a tangent space features translation method based on the Levi-Civita connection was proposed. In addition, we introduce the geometric knowledge of Riemannian manifolds to further explain how features are transformed in the tangent space. Finally, we conduct experiments on several 3D skeleton datasets with different structures, successfully verifying the effectiveness and advancement of the proposed method.**

***Index Terms*—Action recognition, hyperbolic manifold, Poincaré model, Riemannian geometry, spatio-temporal features.**

I. Introduction

ACTION recognition is one of the most important fields in computer vision research. It utilizes computer vision methods to determine the action category of the camera recorded data. Generally, two different types of data are used in action recognition tasks, RGB video data and skeleton data. Most methods based on RGB video data use Convolutional Neural Networks (CNN) to extract image information or use traditional methods to extract video optical flow trajectory information. RGB video data has the advantages of easy collection and data regularization, but it is susceptible to interference from the shooting environment. With the development of reliable skeleton estimation methods in depth video [1] and RGB video [2], the 3D joint positions of human bones in action videos can be easily obtained in real time, which greatly promotes the research and application of skeleton-based action recognition. In recent years, skeleton-based human action recognition has received widespread attention. It mainly uses the Euclidean coordinates of 3D joint points for modeling. The compact skeleton data makes the model more efficient and robust to changes in perspective and environment. These methods have achieved desirable results.

The methods based on manual features [3–6] capture spatio-temporal or geometric features of the skeleton sequence, while the methods based on deep learning is directly supervised by the action category to learn discriminative spatio-temporal features. Although action recognition methods based on manual features can usually achieve good performance, these methods have intrinsic limitations, especially that they can only extract shallow features. Deep learning provides a way to obtain high semantic representations. For example, taking advantage of the characteristics of RNN being suitable for time series data processing, methods based on RNN have been proposed to improve the ability to learn temporal context. Thus, Long Short-Term Memory (LSTM) was introduced to extract time series features. Zhang et al. [7] applied geometric joint features to multi-layer LSTM networks instead of joint positions. Ma et al. [8] utilized dynamic evolution of time series by introducing differences of time series as inputs to the LSTM. The main drawback of these methods is that they lack spatial modeling capabilities, resulting in poor results. CNN has an excellent ability to extract high-level semantic information. Many approaches [9–11] have utilized the CNN model for action recognition by encoding skeletal joints as pseudo images, and then input it into the network. Zhang et al. [11] mapped a skeleton sequence to an image to facilitate spatio-temporal modeling by CNN. Banerjee et al. [12] proposed four feature representations of the sequence of key joints, and utilized CNNs to encode these features for classificationare. Compared to RNN, an significant challenge of using CNN is how to organize sequential data for natural input of the model. Most methods directly convert the skeleton data into images, which may lead to the loss of spatial information and usually complex calculations, limiting their practical applications. Therefore, the application of CNN for skeleton-based action recognition is still an unsolved research problem. In recent years, graph-based skeleton action recognition has become a research hotspot in the field of computer vision due to its excellent performance. The ST-GCN proposed in [13] applied Graph Convolutional Networks (GCN) to action recognition for the first time which achieved a great improvement in accuracy. In recent researches [14], the authors modeled the skeleton sequence as a graph, and applied GCN to capture spatial and temporal dynamics to provide high performance. Although GCN-based methods have achieved excellent accuracy, they have limited applications and are costly in terms of memory when there are larger numbers of nodes. In addition, they

J. Chen, C. Zhao and Q. Wang are with the Department of Computer Science, Xiamen University, Xiamen 361005, China (e-mail: qcwang@xmu.edu.cn).

H. Meng is with the Department of Electronic and Computer Engineering, Brunel University, London, UK, UB83PH (e-mail: hongying.meng@brunel.ac.uk).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TCDS.2022.3171550, IEEE Transactions on Cognitive and Developmental Systems

cannot be generalized to different structure of graph.

All the previous approaches are defined in Euclidean space. However, the underlying anatomical structure of the data often contains more geometric information in non-Euclidean spaces , so Euclidean space may not be the best choice for modeling hierarchical data. Recent studies have proved that complex types of data (such as graph data) in many fields exhibit topological structures that are closely related to manifolds. Under such circumstances, Euclidean space cannot provide maximum expression ability or meaningful geometric representation. For example, Sala et al. [15] proves that arbitrary tree structures cannot be embedded with arbitrary low distortion (i.e. almost preserving their metric) in the Euclidean space with infinite dimensions, but this task becomes strikingly easy in the hyperbolic space with only two dimensions where the exponential growth of distances matches the exponential growth of nodes with the tree depth. Therefore, neural network operations defined directly in the data-related space [16] may benefit the learning process. Different from learning joints embedding directly in Euclidean space, we explore the modeling space of skeleton graph sequence in non-Euclidean geometry. However, deep learning in these non-Euclidean spaces has been rather limited, the main reason being the non-trivial or impossible principled generalizations of basic operations (e.g. vector addition, matrix-vector multiplication, vector translation, vector inner product). Thus, classic tools such as feedforward networks or recurrent networks have no corresponding representations in these spaces, and it is difficult to find natural mathematical descriptions for basic operations such as convolution. Inspired by research [17], we get idea from the bijection between the hyperbolic space and the tangent space. The classic operations can be generalized to tangent spaces through logarithmic map. In this way, the spatio-temporal features can be obtained by applying Euclidean filters on feature map in tangent space. In this paper, we construct a 3D action recognition framework (HMANet) that leverages hyperbolic space to make spatio-temporal features full of hierarchy. Our contributions can be summarized as follows:

- To the best of our knowledge, our HMANet introduces hyperbolic manifold into the field of 3D action recognition for the first time. It devotes to mining the spatial configuration of the skeleton sequence. For features represented in hyperbolic space, we mix temporal and spatial filters to extract spatio-temporal features in the tangent space.
- Explain how our network learns the features in the tangent space from the perspective of differential geometry, and establish relationship between the metric tensor of the Riemannian manifold and the features in the tangent space through mathematical theory. The introduction of manifold theory into the model makes it more explanatory.
- Propose a hyperbolic aware bias for features in the tangent space of manifold. It utilizes the parallel transport with respect to Levi-Civita connection to translate the tangent vector along the geodesic to make the captured features lie in different tangent spaces of manifold, such that the model can automatically aware of underlying manifold.

The rest of this paper is organized as follows. Section 2 reviews the related approaches and discusses their relationships to the present works. Section 3 gives a detailed description of our method and the corresponding network architecture, while supplying a theoretical analysis. Comprehensive experimental results and analysis are provided in Section IV, and finally, a conclusion is drawn in Section V.

II. Related work

Skeleton-Based Action Recognition with CNN

Most of the methods based on CNN flatten the 3D skeleton sequence into pseudo images with joints and frames as different dimensions, and the feature learning follows the methods in image. Li et al.[18] encoded the pairwise distances between joints into RGB images, and separately trained CNN models in 4 orthogonal planes with empirical fusion schemes account for view invariance. Banerjee et al. [12] propose a CNN model, which leverages features estimated from angular information and kinematics of human to capture complementary characteristics of the sequence of key joints. The approach mentioned in [19] is a CNN-based method that utilizes a gating mechanism for images generated from a specific order of skeletons. The two-stream attention mask in CNN was reported in [20]. Li et al.[21] used the features in methods [9] and [18] and the LSTM network to study the multi-classifier classification model of the maximum, multiplicative and average decision score fusion scheme. These methods are not sensitive to subtle movement changes within the class which can be rectified by using more specialized features. Huynh-The et al. [22] studied specialized geometric feature extraction techniques, including joint orientation, which provided impressive performance. Recent methods [23] and [24] exploit transition geometric features alongside frame-wise geometrical features, which is a very crucial step towards utilizing motion information. The features learned by these methods treat the data as an image, and thus fail to effectively express the long-distance interaction relationship in the skeleton. Although the CNN operator can indeed form an overall feature representation through the local convolution kernel, it neglects the interaction of the longdistance joints. Moreover, the Euclidean distance between joint coordinates cannot accurately describe their geometric distance. For the purpose of learning the features implying underlying manifold, our method attempts to mine this geometric topology in hyperbolic space, which enables the distance between coordinates to express their geometric structure to a certain extent.

Representations in non-Euclidean Space

In order to explore more robust skeleton features in non-Euclidean space, one approach is to directly emloy manifold data as the original input. For example, researchers express rotation relationships as points in the Lie group $SO(3)$, and describe the skeleton motion information through the rotation relationships between each pair of 3D vectors, so as to eliminate the influence of viewing angle changes and learn more robust features. Vemulapalli et al. [25] first proposed performing action recognition by using $SO(3)$ to represent human bones (rotation and translation), LieNet [26] further

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TCDS.2022.3171550, IEEE Transactions on Cognitive and Developmental Systems

realized deep learning curve clusters by defining rotation map transformation, vemulapalli et al. [27] introduced the concept of rolling map in mathematics, which mapped the $SO(3)$ representation of the human skeleton to the tangent space, and utilized SVM for linear classification. These methods manually characterize the data in a specific manifold, which may lose part of the original information, resulting in poor results, and networks specially designed for them often bring a large amount of calculation.

Another more reasonable approach is to generalize the deep neural network to non-Euclidean geometry. Specifically, it uses deep learning to automatically embed the data on the Riemannian manifold. For example, in order to construct a model on a Riemannian manifold, Mathieu et al. [28] proposed Poincaré variational autoencoder and showed a better generalisation for hierarchical structures. In this paper, we focus on hyperbolic manifolds, which is a non-Euclidean space with constant negative Gaussian curvature and has the ability to efficiently model hierarchical structures. In machine learning, hyperbolic representations greatly outperformed Euclidean embeddings for hierarchical, taxonomic or entailment data recently. Disjoint subtrees from the implicit hierarchical structure are well clustered in the embedding space. However, appropriate deep learning tools are needed to embed feature data in this space and use it in downstream tasks. Ganea et al. [29] established the connection between hyperbolic manifold and Euclidean space in the context of neural network and deep learning, and generalizes basic operators, polynomial regression and feedforward network to the Poincaré model of hyperbolic manifold. Ungar [30] combined the gyrovector space and the generalized Möbius transformation with the popular properties of Riemannian geometric, smoothly parametrize basic operations and objects in all spaces of constant negative curvature using a unified framework that depends only on the curvature value. Then, the Euclidean space and hyperbolic spaces can be continuously deformed into each other.

Neural Networks on Hyperbolic Manifold

Recently, there have been some attempts to design neural networks in hyperbolic space. Specifically, the pioneering research on learning representation in hyperbolic spaces was reported in [31]. Then, in the research [29], hyperbolic neural networks were introduced, linking hyperbolic geometry with deep learning. Subsequent related works provided analogies on the hyperbolic manifolds of classic operations, or developed several other algorithms, such as Poincaré GloVe [32] and hyperbolic aware mechanism networks [17]. In addition, their method is also more general for graph sequence data because they are naturally in non-Euclidean space. Chami et al. [33] utilized hyperbolic geometry to construct a graph neural network. Considering that there is a bijection between the hyperbolic space and the tangent space, scholars can first perform the convolution operation on the tangent space, and then project the extracted features back as a trajectory on the manifold. Since the hyperbolic distance between unrelated samples in a hyperbolic manifold will increase exponentially than the distance between similar samples, it may be better to construct classification model for human skeleton on a hyperbolic manifold. To this end, we are dedicated to propose a spatio-temporal manifold-aware network for a specific model of hyperbolic geometry (i.e. Poincaré model). This network does not generate node embeddings by inputing human spatio-temporal graph, but explores more reasonable manifold projections, such that the projection features are more discriminative and the network can be generalized to different skeleton structures. In addition, regarding the interpretability of neural networks, Hauser et al. [34] took feature transformation as the transformation of Riemannian metric tensor on manifold from the perspective of differential geometry. This paper attempts to study these issues on hyperbolic manifolds, and combines the network architecture with hyperbolic space, taking advantages of its good hierarchical structure modeling capabilities to further strengthen the exploration of hierarchical structures.

III. PROPOSED METHOD

In this section, we describe our classification model HMANet in details. The framework is shown in Fig.1. The convolutional layer of our network consists of a spatial filter and a temporal filter. Such blocks are added to capture the spatio-temporal features of the skeleton sequence. The model firstly expands the joint dimension by affine transformation, leveraging exponential function to map coordinates to hyperbolic space and renew position coordinates, then performs affine transformation in Euclidean space by logarithmic map. We will describe important components of our framework in the following sections in details.

TABLE I
NOTATIONS AND DEFINITIONS

Notations	Definitions
$\mathcal{M}$	a smooth manifold
H_2	a 2D Poincaré disc
δ_H	the metric in hyperbolic space
δ_E	the metric in Euclidean space
$\mathcal{D}^n$	an n-dimensional open unit ball in Euclidean space
γ_x	the Riemannian metric tensor at point x of manifold
I_n	the n-order identity matrix
λ_x	the conformal factor on manifold
$\oplus$	the Möbius addition on Poincaré model
V	the number of joint points
T	the number of frames in one action
J_t^v	the 3D coordinate of the v-th joint in the t-th frame
w_t^c	the position vector of the c-th channel in the t-th frame
$\tilde{w}_t^c$	the deviation vector of the c-th channel in the t-th frame
$T_x\mathcal{D}^n$	the tangent space at point x on manifold $\mathcal{D}^n$
U_x	an open set on manifold $\mathcal{M}$
φ_x	A coordinate function that maps elements in U_x to $\mathcal{R}^n$
E_x	a set of basis vectors in tangent space
$T_x^*\mathcal{M}$	the cotangent space of the manifold $\mathcal{M}$
D_v	the directional derivative of the direction v
E_x^*	a set of basis vectors in cotangent space
v^l	a feature vector of layer l
$\mathcal{H}^l$	the Jacobian matrix of mapping between two manifolds
$\mathcal{J}$	a smooth second order tensor on manifold

A. Poincaré Model of Hyperbolic Geometry

3D human skeleton can be represented as a graph composed of nodes and edges due to the spatial topology of joints. Traditional Euclidean space is a linear manifold, any

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TCDS.2022.3171550, IEEE Transactions on Cognitive and Developmental Systems

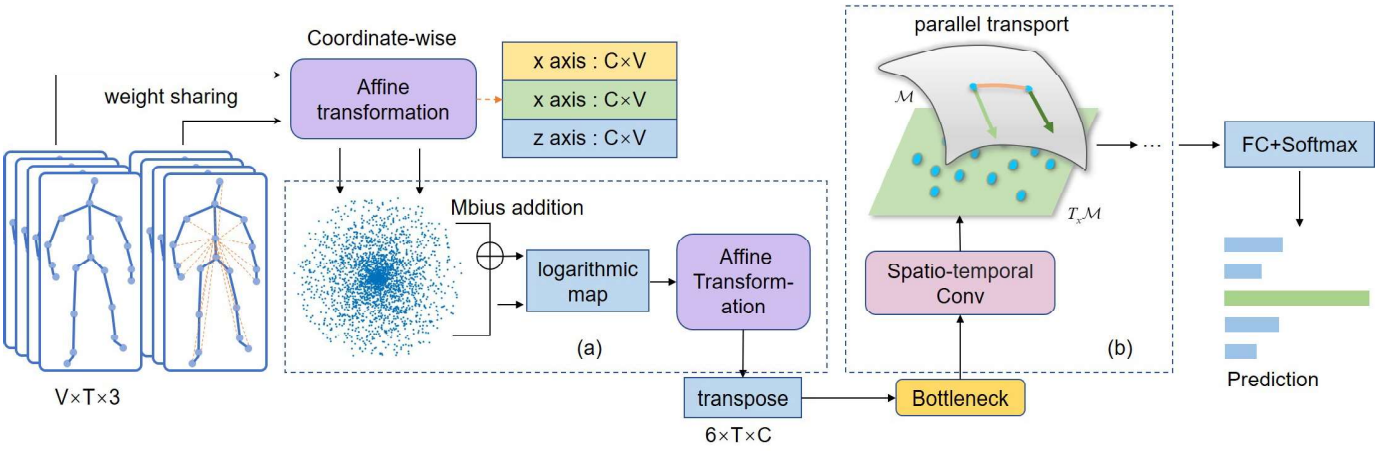

Fig. 1. Illustration of our framework (HMANet). There are mainly two stages in our framework, including (a) Coordinate-wise affine transformation, and (b) Convolution equipped with manifold transaction. At the first stage, we concatenate position tensor with corresponding deviation tensor, and utilize affine transformation to transform coordinates of joints for each dimension. We stack several layers, at the start of next layer, we map the two to hyperbolic space and perform Möbius addition, then utilize logarithmic map to map them back to Euclidean space. The bottleneck followed by is to point-wise expand the dimension. In stage (b), we adopt manifold transaction in convolution layer to make it manifold-aware.

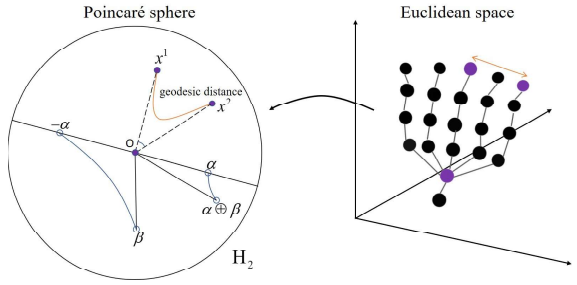

Fig. 2. Left is the 2D Poincaré disk embedded in Euclidean space, where the distance between every two points is the geodesic distance. The right is an illustration of the hand gesture. Embed any two joints into the disk, the distance between them is converted into hyperbolic distance.

parameterization method does not have the ability to represent a graph. Nevertheless, Hyperbolic space provides a more reasonable embedding for human joints due to its unique metric properties. In [35], Cannon gives five isometric models of hyperbolic space. To represent hyperbolic space in a simple way, we choose to study in the Poincaré model. In Fig.2, any two gesture joints can be embedded as two points x^1, x^2 in the Poincaré disc H_2. Poincaré disk $\mathrm{H}_2 := \left\{(x_1, x_2) \mid x_1^2 + x_2^2 < 1\right\}$ is a two-dimensional case of hyperbolic geometry, wherein the distance metric changes. Near each joint, the metric is related to the position of the node, whereas the shortest path between two nodes is not straight line distances. We show that this distance can reflect the topological structure of the joints with the help of the following definitions and formulas.

Hyperbolic space is a Riemannian manifold with constant positive curvature, which is a curved metric space that does not have a distance-preserving relationship with Euclidean space. For nodes represented in hyperbolic space, the metric space expands exponentially with the distance from the original point. Thus it is more advantageous to represent hierarchical information. The Poincaré sphere model $(\mathcal{D}^n, \gamma)$ is an n-dimensional hyperbolic space equipped with a Riemannian metric γ, defined as $\mathcal{D}^n = \{x \in \mathcal{R}^n : \|x\| < 1\}$. The Riemannian metric $\gamma_x : T_x\mathcal{D}^n \times T_x\mathcal{D}^n \to \mathcal{R}$ is a family of positive definite quadratic forms that smoothly vary with point x on the manifold:

$$\gamma_x = \lambda_x^2 \cdot \gamma^E \tag{1}$$

Where $\lambda_x = \frac{2}{1-\|x\|^2}$ is the conformal factor and $\gamma^E = \mathrm{I}_n$ is the Euclidean metric tensor. Therefore, the metric of 2D disc space is defined as:

$$ds^2 = \left(\frac{2}{1 - x_1^2 - x_2^2}\right)^2 \left(dx_1^2 + dx_2^2\right) \tag{2}$$

It can be seen from the formula that the closer the point is to the edge of the disc, the greater the distance represented by the coordinate difference $(\Delta x_1, \Delta x_2)$. Take the gesture skeleton as an example, we illustrate the inspiration for the key technology of embedding skeleton joints in hyperbolic space. In Fig.2, we use a geodesic to show the hyperbolic distance between two points. For the joint points x^1, x^2 in the disc, the geodesic distance is defined as:

$$\delta_H\left(x^1, x^2\right) = \cosh^{-1}\left(1 + 2\frac{\delta_E\left(x^1, x^2\right)^2}{\left(1 - \left\|x^1\right\|^2\right)\left(1 - \left\|x^2\right\|^2\right)}\right) \tag{3}$$

Where δ_H represents the hyperbolic distance, and δ_E represents the Euclidean distance, they can be extended to the case of the 3D skeleton data. Suppose $\left\|x^1\right\| = \left\|x^2\right\| = \tau$, there is:

$$\lim_{\tau \to 1} \delta_H\left(x^1, x^2\right) = \delta_H\left(x^1, 0\right) + \delta_H\left(x^2, 0\right) \tag{4}$$

In other words, the shortest path between x^1 and x^2 is almost the same as the path through the origin. This is analogous to a tree structure, in which the shortest path between two sibling node is the path through their parent node.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TCDS.2022.3171550, IEEE Transactions on Cognitive and Developmental Systems

The tree-like property of hyperbolic space is a key attribute for feature embedding. Given any two points on the disc, no matter how small the angle between them to the center is, this property can be satisfied. Therefore, the hyperbolic distance can well reflect the distance in the sense of joint topology, the natural embedding of the hierarchical structure can be found in the hyperbolic space.

Hyperbolic space is a non-linear space, thus the addition defined in hyperbolic space is different from Euclidean space. It is called Möbius addition, denoted as $\oplus$. For any two points α, β in disc:

$$\alpha \oplus \beta = \frac{\left(1 + 2\langle \alpha, \beta \rangle + \|\beta\|^2\right)\alpha + \left(1 - \|\alpha\|^2\right)\beta}{1 + 2\langle \alpha, \beta \rangle + \|\alpha\|^2\|\beta\|^2} \tag{5}$$

Fig.2 describes the operation from a geometric perspective. As shown in the figure, $\alpha \oplus \beta$ is obtained by translating the triangle along the side $-O\alpha$. Connection can be established by two congruent triangles:

$$\begin{cases} d(-\alpha, \beta) = d(0, \alpha \oplus \beta) \\ d(0, \beta) = d(\alpha, \alpha \oplus \beta) \end{cases} \tag{6}$$

Therefore, combined with the hyperbolic distance formula, it can be known that when β is closer to the center, $\alpha \oplus \beta$ is closer to α, and when β is closer to the edge, the coefficient in the α direction is greater. Simultaneously, if the directions of α and β are closer, $\alpha \oplus \beta$ is farther from the center.

B. The Architecture of HMANet

For the input motion coordinates, we propose an end-to-end deep learning framework. We first represent the skeleton sequence with V joints and T frames as a tensor of shape $V \times T \times 3$. For the skeleton of a person in frame t, we formulate it as $J_t = \left(J_t^1, J_t^2, \ldots, J_t^V\right)^T$, and $J_t^v = \left(J_{tx}^v, J_{ty}^v, J_{tz}^v\right)$ is the 3D joint coordinates. In this way, the skeleton sequence is regarded as an image with 3 channels. Considering that: 1) The two dimensions of the image represent joints and frames, which are usually not equivalent; 2) The movement of a joint is not only related to the local area, but also related to the distant joints. We treat each joint of the skeleton as a channel, and learn the global response of all channels through affine transformation. However, any two channels of the output feature are no longer in a parallel relationship, and they share part of the same information. To this end, we propose a method of transforming features through hyperbolic space, such that the distance of features in new space more accurately reflects their relevance.

Suppose that the coordinate of the human center of gravity in the skeleton is J_t^1, the difference of the coordinates $\tilde{J}_t = J_t - J_t^1$ is calculated in each frame, and these three-dimensional vectors form a tensor with the shape of $V \times T \times 3$, which is called the deviation tensor. We divide the skeleton sequence into 3 parts according to the coordinate dimension, and connect each part with the deviation tensor according to the corresponding dimension to obtain 3 tensors with the shape of $V \times T \times 2$. After that, we use 3 affine transformations to independently aggregate the global features of all joints for each dimension of the coordinate, and use the batchnorm to normalize them, then connect the three dimensions. The obtained tensor is composed of position vectors and deviation vectors. Repeating the learning of the global features, before each subsequent affine transformation, we use the following method to renew the position vectors.

Let $W \in R^{C \times T \times 6}$ be the output of the first layer, where C represents the number of channels. The component of the output tensor at frame t is written as $W_t = (w_{tx}, w_{ty}, w_{tz}, \tilde{w}_{tx}, \tilde{w}_{ty}, \tilde{w}_{tz}) \in R^{C \times 6}$, and channel c contains a position vector $w_t^c = \left(w_{tx}^c, w_{ty}^c, w_{tz}^c\right)$, and a vector $\tilde{w}_t^c = \left(\tilde{w}_{tx}^c, \tilde{w}_{ty}^c, \tilde{w}_{tz}^c\right)$ obtained by affine transformation of the original tensor. Let $A_j \in R^{C \times V}$ and $b_j \in R^C$ $(j = x, y, z)$ be optimizable parameters, the output of each coordinate dimension is obtained by affine transformation:

$$\begin{cases} w_{tx} = A_x J_{tx} + b_x, & \tilde{w}_{tx} = A_x \tilde{J}_{tx} + b_x \\ w_{ty} = A_y J_{ty} + b_y, & \tilde{w}_{ty} = A_y \tilde{J}_{ty} + b_y \\ w_{tz} = A_z J_{tz} + b_z, & \tilde{w}_{tz} = A_z \tilde{J}_{tz} + b_z \end{cases} \tag{7}$$

In order to map them to points in hyperbolic space, we refer to the bijection of hyperbolic space and tangent space at one point proposed by Ganea in [29], called exponential map and logarithmic map. The following are the definitions of exponential map and logarithmic map. $\forall x \in \mathcal{D}^n$:

$$\exp_x(v) = x \oplus \left(\tanh\left(\frac{\lambda_x \|v\|}{2}\right)\phi(v)\right) \tag{8}$$

$$\log_x(y) = \frac{2}{\lambda_x} \tanh^{-1}(\| -x \oplus y\|)\phi(-x \oplus y) \tag{9}$$

Where $\phi(r) = \frac{r}{\|r\|}$ represents vector unitization. We pay attention to case $x = 0$, use the projection function to map the position vectors and the deviation vectors to the hyperbolic space, and perform the Möbius addition to renew the position vectors. However, using affine transformation for features in hyperbolic space will destroy its manifold structure. To this end, we use logarithmic map to project the vectors from the manifold to the tangent space, such that the loss function in the Euclidean space can be employed to optimize the model:

$$w_t^c \leftarrow \log_0\left(\exp_0\left(w_t^c\right) \oplus \exp_0\left(\tilde{w}_t^c\right)\right) \tag{10}$$

We use the deviation vectors to renew the position vectors in the hyperbolic space. Based on the previous discussion, if the included angle of any two deviation vectors is small, the newly obtained corresponding position vector angle will become smaller. Besides, the larger the norm of the deviation vector, it means that in the corresponding position vector, the larger weights are more likely derived from the neighboring points. Therefore, the feature independence is stronger, and the coefficients of this direction are also larger. This is intuitive in the feature space.

Finally, we obtain a tensor with a shape of $C \times T \times 6$. We designate the dimensions of the tangent vectors as channels by transpose, and use bottleneck to increase the feature dimensions before sending it to the convolution layer, which is equivalent to obtaining the coordinate representation of the high-dimensional manifold through manifold immersion. Then

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TCDS.2022.3171550, IEEE Transactions on Cognitive and Developmental Systems

we combine the spatial and temporal filters to extract high-order features. Fig.1 illustrates the entire network framework.

C. Convolution Layer on Tangent Space of Manifold

To discuss the deep convolution block in our model HMANet, we introduce some knowledge about tangent space in this section. As mentioned in section A, Riemannian metric is a quadratic form acting on the tangent space, which can induce the geodesic distance on the manifold.

Metric $\tilde{g}$ and metric g are conformal when they define the identical angle. The Poincaré sphere and Euclidean space are conformal, namely, $\forall x \in \mathcal{D}^n, u, v \in T_x\mathcal{D}^n \backslash \{0\}$, there is

$$\cos(\angle(u,v)) = \frac{g_x^D(u,v)}{\sqrt{g_x^D(u,u)}\sqrt{g_x^D(v,v)}} = \frac{\langle u, v\rangle}{\|u\|\|v\|} \tag{11}$$

Therefore, the length of the tangent vector can be naturally defined as the length in Euclidean space.

In order to transform the manifold features in deep learning based on affine transformation, we map the learned manifold features to the tangent space, thereby obtaining a tangent vector field on the manifold. Since Riemannian manifold $\mathcal{M}$ has a local European structure, there is a family of coordinate charts $\{(U_x, \varphi_x)\}$ that form an open cover of $\mathcal{M}$, and each coordinate function $\varphi_x \in C_x^\infty$ is a homeomorphism from U_x to an open set of $\mathcal{R}^n$. Given a point $x \in \mathcal{M}$ which is mapped to $\mathcal{R}^n$ by coordinate function φ_x, its tangent space $T_x\mathcal{M}$ has a set of natural basis $E_x = \left\{\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n}\right\}$. $\forall v \in T_x\mathcal{M}$, the derivative of the function along the direction v at x is: $D_v[\varphi] = v \cdot d\varphi$, where $[\varphi]$ represents a germ of function at x, that is, all equal functions in a sufficiently small neighborhood. We call $d\varphi$ the cotangent vector, the space $T_x^*\mathcal{M}$ composed of cotangent vectors is called the dual space.

The neural network applies linear transformation to the functional operator in the tangent space through affine transformation. Specifically, considering the cotangent space $T_x^*\mathcal{M}$, there is a dual natural basis $E_x^* = \{dx^1, \ldots, dx^n\}$. We give a smooth tensor $\mathcal{J} : T_x\mathcal{M} \times T_x^*\mathcal{M} \to R$ of type (1,1) which contains a family of Riemannian metrics on $\mathcal{M}$, they can perform inner product on the tangent vector as a special quadratic form. In addition, the Riemannian metric tensor is an endomorphism of the tangent bundle (i.e. $\mathcal{J} : T_x\mathcal{M} \to T_x\mathcal{M}$). The weight matrix of the neural network implies the metric tensor $\mathcal{J}$. Generalizing to a more general case, when the feature dimension is expanded by the network layer, the underlying manifold of features is immersed into the higher-dimensional manifold.

D. Hyperbolic Aware Bias Based on Levi-Civita Connection

Since the tangent space at each point of manifold is not identical, while the captured features are projected to the tangent space of original point (i.e. $T_0\mathcal{M}$) through the logarithmic map, we introduce the following theorem and propose a bias in tangent space to transfer the translation along the geodesic of manifold to the tangent space, converting the tangent vector $v \in T_0\mathcal{M}$ to a tangent vector $v' \in T_x\mathcal{M}, x \neq 0$.

As referred in [29], in the manifold $(\mathcal{D}^n, g)$, the parallel transport w.r.t. the Levi-Civita connection of a vector $v \in T_0\mathcal{D}^n$ to another tangent space $T_x\mathcal{D}^n$ is given by the following isometry:

$$P_{0\to x}(v) = \log_x\left(x \oplus \exp_0(v)\right) = \frac{\lambda_0}{\lambda_x} v = \left(1 - \|x\|^2\right) v \tag{12}$$

This equation is crucial for defining and optimizing the parameters shared between different tangent spaces. Back to our network, the features are regarded as the vectors in tangent space of original point $T_0\mathcal{M}$ through the logarithmic map. A bias is applied to each feature vector to control the distance from the origin point. The distance defined according to the tangent space may reflect its hyperbolic nature. For this reason, we employ the above function to translate the tangent space features. Given a feature in vector form $v_{ti} \in T_x\mathcal{D}^n$, where t represents the temporal dimension and i represents the spatial dimension, we set an optimizable bias parameter $b \in T_0\mathcal{D}^n$ to translate the features in tangent space. Combining Equation 8 and Equation 12, the conformal factor of the feature v_{ti} mapping onto the manifold can be obtained by the tanh function. Therefore, the deformation of bias b on each feature is

$$P_{0\to\exp_0(v_{ti})}(b) = \left(1 - \left(\tanh\|\exp_0(v_{ti})\|\right)^2\right) \cdot b \tag{13}$$

Then, we write the convolution operation on the l-th layer in the network into the following form:

$$v_{ti}^{l+1} = \sigma\left(P_{0\to\exp_0(v_{ti}^l)}(b) + \left(\mathcal{F}_t\left(v_{ti}^l\right) + \mathcal{F}_i\left(v_{ti}^l\right)\right)\right) \tag{14}$$

where $\mathcal{F}_t$ and $\mathcal{F}_i$ represent temporal convolution and spatial convolution respectively. After adding the deformed bias, we make features adapt to different tangent spaces, such that the captured features are located in the tangent space at different points of the manifold, and the features can be represented in the manifold space more accurately. We will demonstrate the effectiveness of this operation by experiments in section IV.

E. Submanifold Immersion and Feature Embedding in Hyperbolic Space

In this section, we explain that after the dimensionality of the features in tangent space is increased by bottleneck, the obtained features can be regarded as vectors in tangent space of a high-dimensional manifold. In deep learning based on Euclidean geometry, the features of the neural network can be regarded as a set of Cartesian coordinates in Euclidean space. The tangent space is a local linear approximation of the manifold, which can be parameterized by the coordinates in the Euclidean space. For the features in manifold, we project them to the tangent space by logarithmic map. In this way, we can learn the tangent space features by performing a linear transformation on the coordinates. If the dimension of features in each layer is constant, the coordinates can be renewed by a full-rank Jacobian matrix, and a positive definite Riemannian metric is maintained. However, the dimension of features in the actual network increases as the layers deepens. From the perspective of differential geometry, if the rank of the map Jacobian matrix is equal to the dimension before the map, the manifold can be immersed in a higher-dimensional space.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TCDS.2022.3171550, IEEE Transactions on Cognitive and Developmental Systems

In our network, in order to more easily separate the data, we embed features into higher-dimensional hyperbolic manifolds. Let $\mathcal{M}$ and $\mathcal{N}$ be m-dimensional and n-dimensional smooth manifolds respectively, where $m \leq n$. $f : \mathcal{M} \to \mathcal{N}$ is a smooth map on manifold. Each tangent space feature in the layer l is represented as $v^l = \log_0(x^l) \in T_0\mathcal{D}^m$ by the logarithmic map on the manifold $\mathcal{M}$, and $v^{l+1} = \log_0(x^{l+1}) \in T_0\mathcal{D}^n$ on the manifold $\mathcal{N}$. Denoting that

$$\begin{aligned} h^l\left(v^l\right) &:= \left(\log_0 \circ f \circ \log_0^{-1}\right)\left(v^l\right) \\ &= \left(\log_0 \circ f \circ \exp_0\right)\left(v^l\right) \end{aligned} \tag{15}$$

where $v^l \in \mathcal{R}^m$, $h^l\left(v^l\right) \in \mathcal{R}^n$, the Jacobian matrix of map f is given by two coordinate functions:

$$\text{Jacobi}\ \left(h^l\right) = \begin{bmatrix} \frac{\partial h_1^l}{\partial v_1^l} & \cdots & \frac{\partial h_1^l}{\partial v_m^l} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_n^l}{\partial v_1^l} & \cdots & \frac{\partial h_n^l}{\partial v_m^l} \end{bmatrix} \tag{16}$$

Denoting this matrix as $\mathcal{H}^l$, the network will learn this matrix to ensure

$$\text{rank}\left(h^l\right) := \text{rank}\left(\mathcal{H}^l\right) = m \tag{17}$$

In this case, $\forall x \in \mathcal{M}$, the Jacobian matrix of the coordinate function h^l is general non-degenerate, then f is an immersion of smooth manifold $\mathcal{M}$ in $\mathcal{N}$. Specific to our network, we utilize bottleneck to transform the features in vector form into a higher-dimensional space and project it to a hyperbolic manifold. This operation can be viewed as the immersion of the manifold space, which ensures that the high-dimensional manifold retains local properties of the manifold of low-dimension, enabling the network to learn the geometric structure of the data.

IV. EXPERIMENTS AND ANALYSIS

This section describes the experiments in terms of datasets, the implementation, the training details, the comparison results and the corresponding analysis.

A. Datasets

We evaluate the performance of HMANet on four benchmark skeleton-based action recognition datasets. In all datasets, we use only the skeleton joint markers.

NTU RGB+D [36]: It is currently one of the largest 3D action recognition datasets, containing RGB+D videos and skeleton data for human action recognition. The motion data was captured from 40 human objects by 3 Microsoft Kinect V2 cameras. There are 56880 samples with 4 million frames in 60 categories, and the maximum number of frames in all samples is 300. Each body skeleton records 25 joints. 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. We follow the original two benchmarks and report the accuracy of Top-1.

Gaming-3D (G3D) [37]: It is a gaming dataset collected with Microsoft Kinect which contains a total of 663 motion sequences. The dataset consists of 20 actions performed by 10 subjects in a controlled indoor environment. Each people performs several times and each sequence may contain multiple actions. As this dataset consists of gaming actions, it has many temporal dependencies and rapid movements of body parts in the video sequences. The dataset provides RGB video data and skeleton data. Skeleton data provides the 3D coordinates of the joints. Each body in a sequence records 20 joints. We use the same protocol as the other works wherein the first five subjects are used for training, and the remaining for testing.

SHREC'17 Track Dataset [38]: The dataset is a public dynamic hand gesture dataset presented for the SHREC'17 Track. It contains sequences of 14 gestures performed between 1 and 10 times by 28 participants in 2 finger configurations, resulting in 2800 sequences. The data is categorized with two levels of granularity, presenting 14 and 28 actions respectively. The coordinates of 22 hand joints in the 3D world space are provided per frame, forming a full hand skeleton. Following the evaluation protocol of SHREC'17 track [38], we trained our model on 1960 samples and evaluated on the other 840 samples.

DHG-14/28 Dataset [38]: The dataset is a public dynamic hand gesture dataset collected by the Intel RealSense short range depth camera. It contains sequences of 14 hand gestures performed 5 times by 20 participants, resulting in 2800 video sequences. The gestures are performed in two ways: using one finger, and using the whole hand. The coordinates of 22 hand joints in the 3D world space are provided per frame, forming a full hand skeleton. Although the DHG-14/28 dataset has the same hand gestures with the SHREC'17 Track dataset, it is more challenging due to the leave-one-subject-out experimental protocol.

B. Implementation

Before the data was fed into the networks, we conduct some pre-processing such that the data structure of each video clip is unified. Since different actions last for various durations, the input sequences are normalized to a fixed length (128 for NTU RGB+D and 64 for others) through bilinear interpolation along the frame dimension. For the single-person sample in dataset with two objects, the second body will be padded with all zeros. Each dimension of the 3D coordinates is put into three channels as inputs. In order to evaluate the effectiveness of our framework more purely, we use relatively primitive data without any preprocessing such as random noise and random cropping.

The basic framework is as illustrated in Fig.1. Take the NTU RGB+D dataset as an example. First, We concatenate each dimension of the three-dimensional position tensor with the deviation tensor and designate joints as channels to extract the global response separately. Then, the position vectors and deviation vectors are mapped to the manifold along the coordinate dimension. They are summed on the manifold space

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TCDS.2022.3171550, IEEE Transactions on Cognitive and Developmental Systems

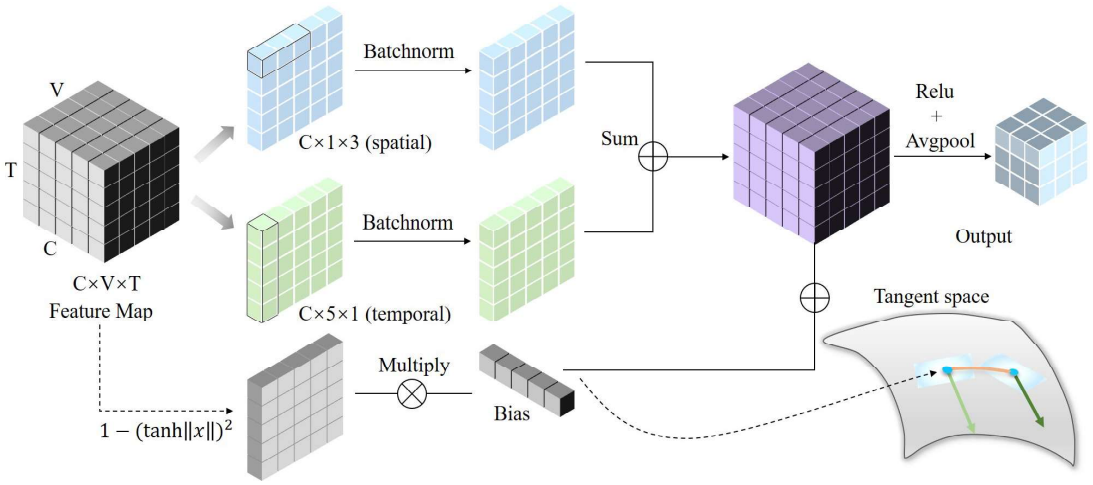

Fig. 3. Illustration of spatio-temporal convolution block equipped with manifold aware bias. Here, the convolution kernel is divided into spatial kernel and temporal kernel and used on the same feature map, the outputs of the two are summed. Both of them are followed by batch normalization (BN) layer. Moreover, a changeable bias based on hyperbolic manifold is utilized to parallel transport the feature. The output is followed by activation layer (ReLU) and Avgpool is fed into next block.

to obtain the new position tensor. To conduct convolution in manifold space, we use logarithmic map to project manifold features to Euclidean space. After the original data being transformed to a 128×128 pseudo image with 6 channels, a bottleneck is utilized to extend the feature to 64D. Then, the corresponding spatio-temporal Conv layers are built in tangent space. As shown in Fig.3, in each convolution layer, 5×1 temporal convolution and 1×3 spatial convolution are coincident, and a bias adapted to manifold is used to move the features. We empirically stack 6 layers on this tangent space and channels at each layer are [64, 64, 128, 256, 256, 256]. Following each layer, a 2×2 Avgpooling is utilized to reduce dimensionality. Finally, the resulted features are averaged along the temporal dimension to 4 vectors in tangent space, and a FC layer followed by a softmax function is utilized to predict a class prediction.

During the training process, the cross-entropy loss is utilized as the classification loss. The learning rate is set as 0.01 and is decreased based on a cosine function. A stochastic gradient descent (SGD) with Nesterov momentum (0.9) is applied as the optimization algorithm for the network. We set the weight decay to 0.0002 as regularization. This model will be trained for 70 epochs and compared to other approaches.

C. Ablation Study

We evaluate the effectiveness of our framework on the datasets of human body and gesture datasets under given two evaluation measures. Firstly, we evaluate how much benefit we obtained from hyperbolic geometry, we implemented our network without hyperbolic geometry. It works directly in Euclidean space without Möbius addtion with its corresponding distance tensor. Simultaneously, the changeable bias based on the hyperbolic manifold is also removed. This network serves as a baseline for comparison with our improved model. The comparison results are shown in Fig.4. It can be seen from our experiments, with the help of hyperbolic space, for a given evaluation, our HMANet can improve the performance without

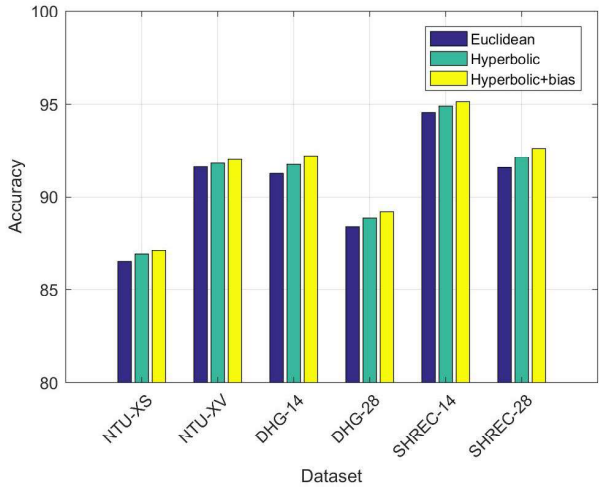

Fig. 4. Comparison of recognition accuracy of using and not using hyperbolic manifold aware mechanism on various datasets.

increasing the parameters. Specifically, under the X-subject and X-view evaluations in NTU RGB+D, our model can overcome the baseline by 0.5% and 0.3%, respectively. Similarly, in SHREC'17, the proposed model defined in hyperbolic space could even outperform it by 1.1% with 28 gesture setting. All of them proves that defining the model on manifold space could benefit greatly.

To further evaluate the effectiveness of the proposed Hyperbolic aware bias, we remove the bias based on the proposed network and conduct comparative experiments. The result is shown by the green bar in the histogram. From the figure, we can notice that using the proposed bias has a certain accuracy improvement on all datasets, especially in gesture data. This is due to the fact that the gesture skeleton is extended from the wrist, which is more tree-like and has a clearer hierarchy between joints. To evaluate how much difference between the original position vectors and that after Möbius addition, we

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TCDS.2022.3171550, IEEE Transactions on Cognitive and Developmental Systems

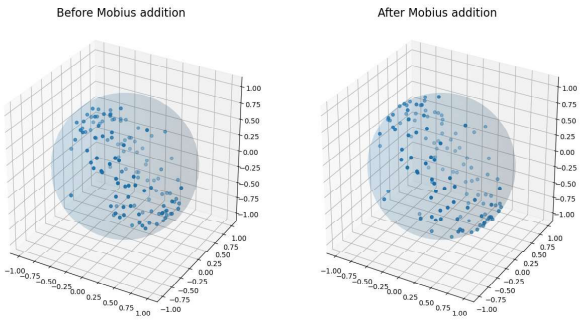

Fig. 5. Left is the projection of the position vectors in hyperbolic space. The right is the result of the Möbius addtion of the position vectors and the deviation vectors.

plotted two spheres containing points distributed in hyperbolic space respectively, as shown in Fig.5. From the figure, we can see that after Möbius addition, the points are more dispersed in the space, and their distribution is more uniform. In other words, the coupling between features is reduced, features become more independent.

D. Comparison With State-of-The-Art

1) G3D: Table II shows comparison with previous methods. Our HMANet is able to achieve superior performance to [39] which extends the Restricted Boltzmann Machine. We are also able to outperform [27] which uses rolling map to project data represented as points on Lie Group to Euclidean space, and the recent method [40]. Work [9] encodes the 3D skeleton data into 2D images, and then utilizes the convolutional network for recognition, increasing the accuracy to 94.24%, which verifies the effectiveness of the convolutional network. We are able to outperform other CNN based methods [9, 12] largely, achieving a state-of-the-art result. It can be seen from the confusion matrix in Fig.6, the recognition errors concentrate on punch right and wave, while our HMANet achieves almost

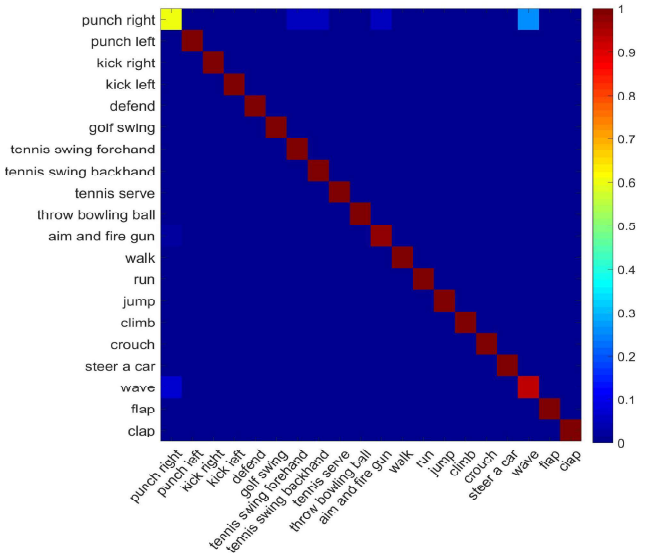

Fig. 6. Confusion matrix of G3D dataset using Cross-Subject protocol.

100% in all other recognizing actions.

TABLE II
PERFORMANCE COMPARISON ON G3D USING CROSS-SUBJECT PROTOCOL

Method	Year	Accuracy(%)
LRBM[39]	2015	90.50
R3D[27]	2016	90.94
JTM[9]	2018	94.24
CNN[9]	2018	96.00
HDM-BG[40]	2019	92.00
FIB-CNN[12]	2020	93.11
KM+TSC[41]	2021	92.91
Proposed HMANet	-	**97.16**

TABLE III
PERFORMANCE COMPARISON ON NTU RGB+D USING CROSS-SUBJECT AND CROSS-VIEW PROTOCOL

Method	Year	Cross-Subject(%)	Cross-View(%)
STA-LSTM[42]	2017	73.4	81.2
GCA-LSTM[43]	2017	74.4	82.8
DS-LSTM[44]	2020	77.80	87.33
CNN+LSTM[21]	2017	82.89	90.10
MTCNN+Rot.Clips[45]	2018	81.09	87.37
HCN[46]	2018	86.5	91.1
TSSI+GLAN+SSAN[20]	2019	82.4	89.1
(P+C)Net[19]	2019	86.1	**93.5**
PoF2I[22]	2019	82.46	89.53
TSRJI[47]	2019	73.3	80.3
POT2I+Inception v3[23]	2020	83.85	90.33
FIB-CNN[12]	2021	84.22	89.71
LAGA-Net[48]	2021	87.07	93.17
ST-GCN[13]	2018	81.5	88.3
GECNNs[49]	2020	85.4	91.1
Proposed HMANet	-	**87.1**	92.0

2) NTU RGB+D: Table III shows comparison of our HMANet with past networks. When compared to the LSTM based approaches [42–44], our network achieves superior performance. This is due to incorporating spatio-temporal features in our model which is sensitive to the geometrical information in the sequence. Furthermore, among all the CNN based methods, we achieve the best result on cross-subject protocol. Our HMANet outperforms approach [21] without employing LSTM networks and [20] without using a specific depth first traversal. The CNN method [46] achieves best performance after learning the tangent space features, which is also considered in our model. We have further achieved comparable performance to some GCN-based method [13, 49], and outperforming the pioneering work [13] on both cross-subject and cross-view protocol by 5.6% and 3.7% respectively. When compared to the GCN based methods, the CNN based methods are unable to take full advantage of the topological structure due to lack of input graph, hence do not perform well. The GCN-based methods utilize local information about specific body parts, while our network does not require any such separate handling of body parts.

3) SHREC'17 Track and DHG-14/28: Table IV shows the recognition accuracy of our framework trained and evaluated on SHREC'17 dataset and DHG-14/28 dataset. It shows that our HMANet achieves state-of-the-art performance under both 14 gesture and 28 gesture settings on the SHREC'17 dataset,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TCDS.2022.3171550, IEEE Transactions on Cognitive and Developmental Systems

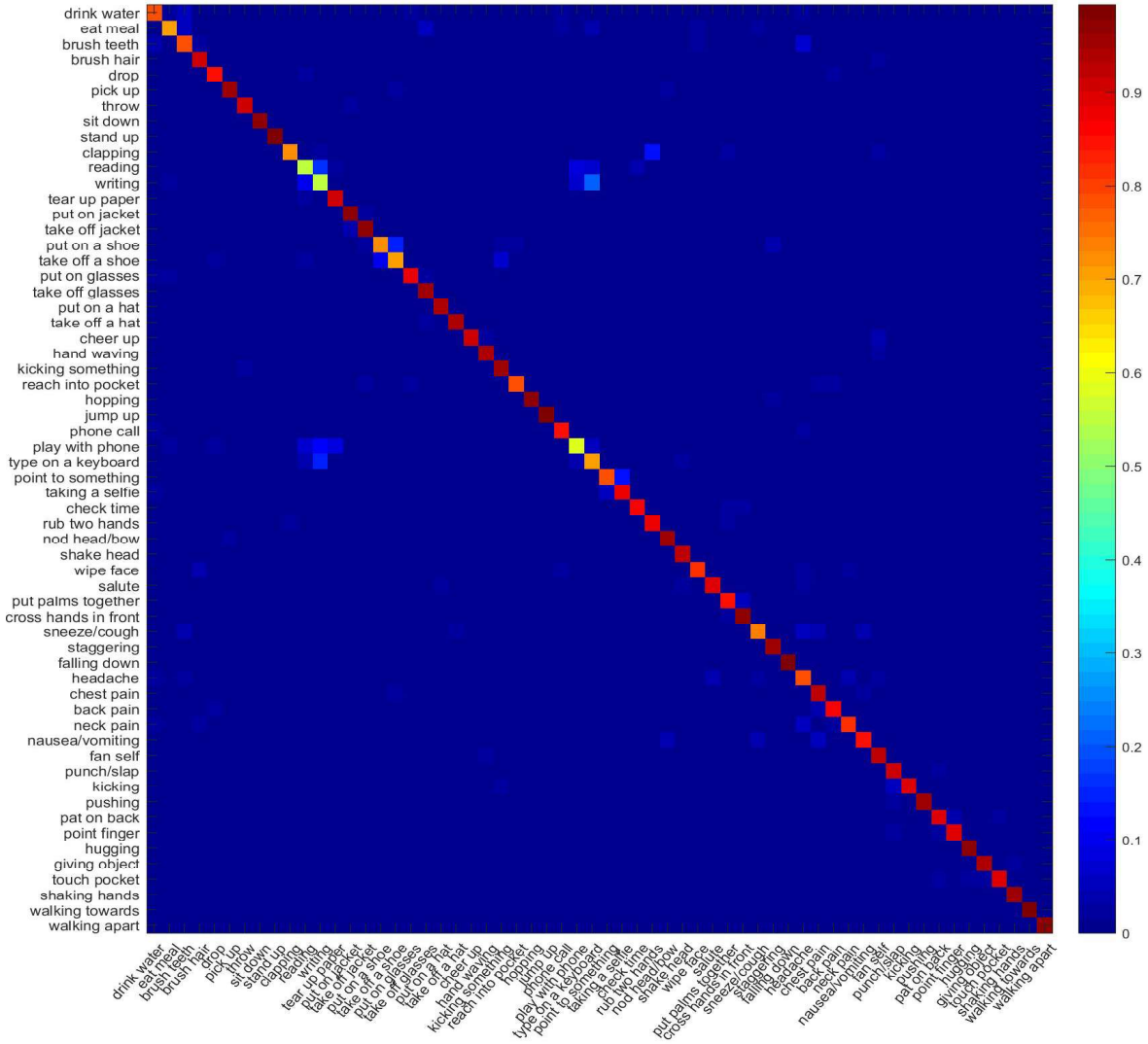

Fig. 7. Confusion matrix of NTU RGB+D dataset in terms of the Cross-Subject protocol.

TABLE IV
PERFORMANCE COMPARISON ON HAND GESTURE DATASETS. 14G AND 28G REPRESENT 14 AND 28 GESTURE SETTINGS.

(a) DHG-14/28 dataset using the Leave-One-Subject-Out protocol

Method	Year	Accuracy(%)	
		14G	28G
CNN+LSTM[21]	2018	85.60	81.10
Res-TCN[50]	2018	86.90	83.60
STA-Res-TCN[50]	2018	89.20	85.00
ST-GCN[13]	2018	91.20	87.10
ST-TS-HGR-NET[51]	2019	87.30	83.40
SPD-NET[51]	2019	92.38	86.31
DG-STA[52]	2019	91.90	88.00
HPEV+HMM[53]	2020	**92.54**	88.86
Proposed HMANet	-	92.21	**89.18**

(b) SHREC'17 Track dataset using Cross-Subject protocol

Method	Year	Accuracy(%)	
		14G	28G
CNN+LSTM[21]	2018	89.8	86.3
Parallel CNN[54]	2018	91.3	84.4
Res-TCN[50]	2018	91.1	87.3
STA-Res-TCN[50]	2018	93.6	90.7
MFA-Net[55]	2019	91.3	86.6
DD-Net[56]	2019	94.6	91.9
HPEV+HMM[53]	2020	94.88	92.26
TCN-Summ[57]	2021	93.57	91.43
Proposed HMANet	-	**95.12**	**92.62**

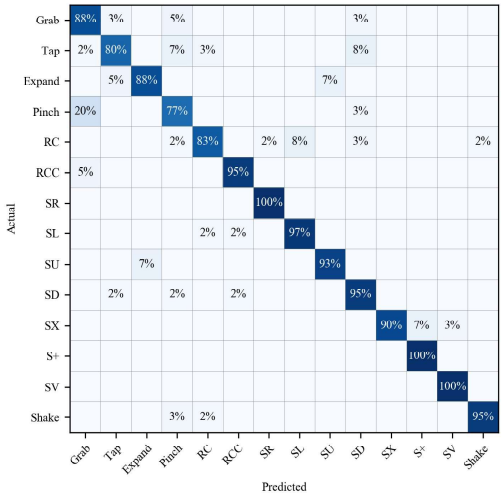

Fig. 8. Confusion matrix of DHG dataset with 14 gestures setting.

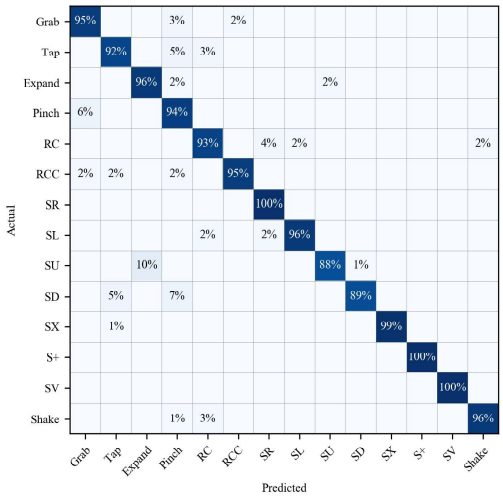

Fig. 9. Confusion matrix of SHREC'17 dataset with 14 gestures setting.

greater accuracy improvement with the more complicated 28 gestures setting, which further validates the effectiveness of our proposed model. Comparing the performance on DHG-14/28 dataset, our proposed hierarchical architecture brings 6.61% and 8.08% accuracy improvement respectively for the 14 gestures setting and 28 gestures setting compared to CNN+LSTM. As shown in Table IV, our network obtains 92.21% on 14 gesture protocol and 89.18% on 28 gesture protocol. It is comparable with the state-of-the-art result in net HPEV+HMM [53] obtaining 92.54% and 88.86% for experiments with 14 and 28 gestures respectively. Particularly, the good performance is more notable with 28 gestures setting than that with 14 gestures setting. Fig.8 and Fig.9 shows the confusion matrix of our network on DHG dataset and the SHREC'17 dataset. The recognition errors concentrate on highly similar actions, e.g. Grap to Pinch. Our network achieves 100% accuracy on both datasets in recognizing actions Swipe Right, Swipe +, Swipe-V.

V. Conclusion

In this paper, we propose a skeleton-based action recognition model using hyperbolic manifold theory. The model is characterized by obtaining joint interaction from the spatial domain for the skeleton sequence, and parametrically representing it in a hyperbolic space. Capturing the coordinate information of global joints through affine transformation, the spatial joint interaction can be fully explored to extract discriminative spatio-temporal features. Since the excellent ability of hyperbolic space to represent hierarchical features, the global interactive features embedded can reflect certain hierarchical relationships. In addition, to extract spatio-temporal features, our method separately uses temporal filter and spatial filter to fuse local information and combine them, while a bias based on hyperbolic manifold is added. On public datasets including human body and gestures, we conduct experiments to prove that the proposed method has a certain versatility while achieving accuracy comparable with mainstream methods, which shows the method can be generalized to different skeleton structures.

References

[1] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 1297–1304.

[2] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–14, 2017.

[3] B. Sun, S. Wang, D. Kong, L. Wang, and B. Yin, "Real-time human action recognition using locally aggregated kinematic-guided skeletonlet and supervised hashing-by-analysis model," *IEEE Transactions on Cybernetics*, 2021.

[4] Z. Shao, Y. Li, Y. Guo, X. Zhou, and S. Chen, "A hierarchical model for human action recognition from body-parts," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 10, pp. 2986–3000, 2018.

[5] J. Weng, C. Weng, and J. Yuan, "Spatio-temporal naive-bayes nearest-neighbor (st-nbnn) for skeleton-based action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4171–4180.

[6] M. M. Arzani, M. Fathy, A. A. Azirani, and E. Adeli, "Switching structured prediction for simple and complex human activity recognition," *IEEE Transactions on Cybernetics*, 2020.

[7] S. Zhang, X. Liu, and J. Xiao, "On geometric features for skeleton-based action recognition using multilayer lstm networks," in *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 148–157.

[8] Q. Ma, Z. Chen, S. Tian, and W. W. Ng, "Difference-guided representation learning network for multivariate time-series classification," *IEEE Transactions on Cybernetics*, 2020.

[9] P. Wang, W. Li, C. Li, and Y. Hou, "Action recognition based on joint trajectory maps with convolutional neural networks," *Knowledge-Based Systems*, vol. 158, pp. 43–53, 2018.

[10] Y. Xu, J. Cheng, L. Wang, H. Xia, F. Liu, and D. Tao, "Ensemble one-dimensional convolution neural networks for skeleton-based action recognition," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 1044–1048, 2018.

[11] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive neural networks for high performance skeleton-based human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1963–1978, 2019.

[12] A. Banerjee, P. K. Singh, and R. Sarkar, "Fuzzy integral based cnn classifier fusion for 3d skeleton action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[13] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proceedings of the AAAI conference on Artificial Intelligence*, 2018.

[14] 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, no. 01, 2019, pp. 8561–8568.

[15] F. Sala, C. De Sa, A. Gu, and C. Ré, "Representation tradeoffs for hyperbolic embeddings," in *Proceedings of the International Conference on Machine Learning*. PMLR, 2018, pp. 4460–4469.

[16] R. Benedetti and C. Petronio, *Lectures on hyperbolic geometry*. Springer Science & Business Media, 2012.

[17] C. Gulcehre, M. Denil, M. Malinowski, A. Razavi, R. Pascanu, K. M. Hermann, P. Battaglia, V. Bapst, D. Raposo, A. Santoro *et al.*, "Hyperbolic attention networks," *arXiv preprint arXiv:1805.09786*, 2018.

[18] C. Li, Y. Hou, P. Wang, and W. Li, "Joint distance maps based action recognition with convolutional neural networks," *IEEE Signal Processing Letters*, vol. 24, no. 5, pp. 624–628, 2017.

[19] C. Cao, C. Lan, Y. Zhang, W. Zeng, H. Lu, and Y. Zhang, "Skeleton-based action recognition with gated convolutional neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 11, pp. 3247–3257, 2018.

[20] Z. Yang, Y. Li, J. Yang, and J. Luo, "Action recognition with spatio–temporal visual attention on skeleton image sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 8, pp. 2405–2415, 2018.

[21] C. Li, P. Wang, S. Wang, Y. Hou, and W. Li, "Skeleton-based action recognition using lstm and cnn," in *Proceedings of the International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2017, pp. 585–590.

[22] T. Huynh-The, C.-H. Hua, and D.-S. Kim, "Encoding pose features to images with data augmentation for 3-d action recognition," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3100–3111, 2019.

[23] T. Huynh-The, C.-H. Hua, T.-T. Ngo, and D.-S. Kim, "Image representation of pose-transition feature for 3d skeleton-based action recognition," *Information Sciences*, vol. 513, pp. 112–126, 2020.

[24] T. Huynh-The, C.-H. Hua, N. A. Tu, and D.-S. Kim, "Learning geometric features with dual–stream cnn for 3d action recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 2353–2357.

[25] 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*, 2014, pp. 588–595.

[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*, 2017, pp. 6099–6108.

[27] 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*, 2016, pp. 4471–4479.

[28] E. Mathieu, C. L. Lan, C. J. Maddison, R. Tomioka, and Y. W. Teh, "Continuous hierarchical representations with poincaré variational auto-encoders," *arXiv preprint arXiv:1901.06033*, 2019.

[29] O.-E. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic neural networks," *arXiv preprint arXiv:1805.09112*, 2018.

[30] A. A. Ungar, "A gyrovector space approach to hyperbolic geometry," *Synthesis Lectures on Mathematics and Statistics*, vol. 1, no. 1, pp. 1–194, 2008.

[31] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," *Proceedings of the International Conference on Neural Information Processing Systems*, vol. 30, pp. 6338–6347, 2017.

[32] A. Tifrea, G. Bécigneul, and O.-E. Ganea, "Poincaré glove: Hyperbolic word embeddings," *arXiv preprint arXiv:1810.06546*, 2018.

[33] I. Chami, Z. Ying, C. R, and J. Leskovec, "Hyperbolic graph convolutional neural networks," *Proceedings of the International Conference on Neural Information Processing Systems*, vol. 32, pp. 4868–4879, 2019.

[34] M. B. Hauser, "Principles of riemannian geometry in neural networks," 2018.

[35] J. W. Cannon, W. J. Floyd, R. Kenyon, W. R. Parry *et al.*, "Hyperbolic geometry," *Flavors of geometry*, vol. 31, no. 59-115, p. 2, 1997.

[36] 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 International Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1010–1019.

[37] V. Bloom, D. Makris, and V. Argyriou, "G3d: A gaming action dataset and real time action recognition evaluation framework," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 7–12.

[38] Q. De Smedt, H. Wannous, J.-P. Vandeborre, J. Guerry, B. Le Saux, and D. Filliat, "Shrec'17 track: 3d hand gesture recognition using a depth and skeletal dataset," in *Proceedings of the 3DOR-10th Eurographics Workshop on 3D Object Retrieval*, 2017, pp. 1–6.

[39] S. Nie, Z. Wang, and Q. Ji, "A generative restricted boltzmann machine based method for high-dimensional motion data modeling," *Computer Vision and Image Understanding*, vol. 136, pp. 14–22, 2015.

[40] R. Zhao, W. Xu, H. Su, and Q. Ji, "Bayesian hierarchical dynamic model for human action recognition," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7733–7742.

[41] G. Paoletti, J. Cavazza, C. Beyan, and A. Del Bue, "Subspace clustering for action recognition with covariance representations and temporal pruning," in *Proceedings of the 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 6035–6042.

[42] 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 *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[43] 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 International Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1647–1656.

[44] X. Jiang, K. Xu, and T. Sun, "Action recognition scheme based on skeleton representation with ds-lstm network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2129–2140, 2019.

[45] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Learning clip representations for skeleton-based 3d action recognition," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2842–2855, 2018.

[46] C. Li, Q. Zhong, D. Xie, and S. Pu, "Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation," *arXiv preprint arXiv:1804.06055*, 2018.

[47] C. Caetano, F. Brémond, and W. R. Schwartz, "Skeleton image representation for 3d action recognition based on tree structure and reference joints," in *Proceedings of the 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, 2019, pp. 16–23.

[48] R. Xia, Y. Li, and W. Luo, "Laga-net: Local-and-global attention network for skeleton based action recognition," *IEEE Transactions on Multimedia*, 2021.

[49] X. Zhang, C. Xu, X. Tian, and D. Tao, "Graph edge convolutional neural networks for skeleton-based action recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 8, pp. 3047–3060, 2019.

[50] J. Hou, G. Wang, X. Chen, J.-H. Xue, R. Zhu, and H. Yang, "Spatial-temporal attention res-tcn for skeleton-based dynamic hand gesture recognition," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[51] X. S. Nguyen, L. Brun, O. Lézoray, and S. Bougleux, "A neural network based on spd manifold learning for skeleton-based hand gesture recognition," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 036–12 045.

[52] Y. Chen, L. Zhao, X. Peng, J. Yuan, and D. N. Metaxas, "Construct dynamic graphs for hand gesture recognition via spatial-temporal attention," *arXiv preprint arXiv:1907.08871*, 2019.

[53] J. Liu, Y. Liu, Y. Wang, V. Prinet, S. Xiang, and C. Pan, "Decoupled representation learning for skeleton-based gesture recognition," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5751–5760.

[54] G. Devineau, W. Xi, F. Moutarde, and J. Yang, "Convolutional neural networks for multivariate time series classification using both inter-and intra-channel parallel convolutions," in *Proceedings of the Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP'2018)*, 2018.

[55] X. Chen, H. Guo, G. Wang, and L. Zhang, "Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition," in *Proceedings of the International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2881–2885.

[56] F. Yang, Y. Wu, S. Sakti, and S. Nakamura, "Make skeleton-based action recognition model smaller, faster and better," in *Proceedings of the ACM Multimedia Asia*, 2019, pp. 1–6.

[57] A. Sabater, I. Alonso, L. Montesano, and A. C. Murillo, "Domain and view-point agnostic hand action recognition," *arXiv preprint arXiv:2103.02303*, 2021.