HYPERBOLIC SPATIAL TEMPORAL GRAPH CONVOLUTIONAL NETWORKS

Abdelrahman Mostafa,Wei Peng,Guoying Zhao

Center for Machine Vision and Signal Analysis, University of Oulu, Finland

2022 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-9620-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIP46576.2022.9897522

ABSTRACT

Spatial-temporal graph convolutional networks (ST-GCNs) have been successfully applied for dynamic graphs representation learning, such as modeling skeleton-based human actions. However, ST-GCNs embed these non-Euclidean graph structures into Euclidean space, which is not the natural space to represent such structures as embedding them in this space incurs a large distortion. In this work, we make use of hyperbolic non-Euclidean geometry and construct compact ST-GCNs in the hyperbolic space. It can be shown that hyperbolic ST-GCNs (HST-GCNs) outperform the corresponding Euclidean counterparts. Additionally, these compact hyperbolic models can be used to increase the performance of large complex Euclidean models. Moreover, we show that the same or even better performance of large Euclidean models can be achieved by fusing the scores of smaller Euclidean models and a compact hyperbolic model. This in turn leads to reducing the total number of model parameters and hence model size. To validate the performance of these hyperbolic networks, we conducted extensive experiments on NTU RGB+D, NTU RGB+D 120 and Kinectics-Skeleton datasets for human action recognition.

Index Terms— Hyperbolic geometry, dynamic graphs, graph convolutional networks, human action recognition

1. INTRODUCTION

Human action recognition is an active area of research with many applications such as surveillance and human-computer interaction. Human actions can be predicted from RGB videos, depth maps or skeleton data. In particular, skeleton-based human action recognition has increasingly gained attention due to its robustness against different changes like illumination or viewpoints [1]. Skeleton data provide abstract information about the locations of human joints.

Earlier deep learning skeleton-based human action recognition approaches considered the human joints as a set of independent features and ignored the relationships between joints. These features are then fed into Convolutional Neural Networks (CNNs) [2, 3] or Recurrent Neural Networks (RNNs) [4, 5] to predict the action class.

A human skeleton can be represented as a graph structure. Recently, GCNs were proposed to generalize the convolution operation to any structured graph data [6, 7]. GCNs were used to model the skeleton data in the spatial domain to build the ST-GCN [8]. Many subsequent works achieved state-of-the-art performance on human action recognition datasets using some variants based on ST-GCN to model human joints sequences [9, 10, 11, 12].

However, all these models embed the features into the Euclidean space which has been shown to incur a large distortion [13]. That is because the ball volume V grows polynomially with respect to the radius r in Euclidean space ($V = \frac{\pi^{n/2} r^n}{\Gamma(n/2+1)}$ for n-dimensional Euclidean space where $\Gamma(z)$ is the Euler gamma function) whereas it grows exponentially in hyperbolic space which leads to lower distortion. This can make the model compact as low-dimensional embeddings are needed without losing much information due to distortion. The hyperbolic space is ideal for embedding trees as the number of tree nodes is growing exponentially with respect to the tree depth. The Gromovs or δ-hyperbolicity ($\delta \geq 0$) [14] is used to measure the tree-likeness of data (trees have $\delta = 0$) and the smaller the value is, the more hyperbolic the data. Since the commonly used graph topologies for human skeleton are with δ-hyperbolicity $= 0$, we were motivated to use the hyperbolic space as the embedding space for the human action features. Extensive experiments are performed on three large scale human action recognition datasets, namely, NTU RGB+D, NTU RGB+D 120 and Kinectics-Skeleton and the results demonstrate that using HST-GCNs has great advantages.

2. RELATED WORK

2.1. Skeleton-based action recognition

Human action recognition has been studied extensively and the first approaches used hand-crafted features for this task [15]. With the advances achieved in deep learning, the next approaches used CNNs or RNNs [4, 3, 2, 5, 16]. For example, the work in [3] uses a multi-scale CNN and fine-tune pre-trained CNNs, e.g., AlexNet, ResNet on human action datasets. The work in [4] divides the human skeleton into five parts which are fed into five subnets that can be fused to form a hierarchical RNN.

GCNs model the spatial relationships in graphs and generalize the convolution operation to any graph. ST-GCN [8] used GCNs to model the spatial relationship between joints. Adaptive GCN (AGCN) [9] used a learnable matrix to model the connections for a general graph for all sequences and also used an input-dependent matrix to learn the connections for each input sequence. In a subsequent work [10], the authors added attention modules in the spatial, temporal and channel dimensions to improve the performance. MS-G3D [11] designed a multi-scale sophisticated model which has parallel branches with multi-scale disentangled feature aggregation. The authors introduced across space-time connections to extract more relevant and enhanced features. There are many other works that built models based on GCNs [17, 18, 19, 20, 12]. For example, the work in [20] proposes shift-GCN which uses shift graph operations and point-wise convolutions. Dynamic GCNs were introduced in [18] to learn the skeleton topology automatically. However, all GCN-based previous works use the Euclidean space to embed the tree-like non-Euclidean human action recognition data. In this work, we exploit the hyperbolic geometry and use it to embed the features in the natural non-Euclidean hyperbolic space.

2.2. Hyperbolic neural networks

Using the Lorentz model of hyperbolic space gives enhanced embeddings specially for spaces with small dimensions [21]. HGCNs

were proposed by [13] and they were able to achieve better performance on node classification and link prediction tasks when compared to the Euclidean analogs. Concurrently to this work, [22] proposed the Hyperbolic Graph Neural Networks (HGNNs) which performed well on graph classification tasks. However, both methods only considered the spatial configurations between graph nodes on static graph tasks. In this work, we also consider the temporal domain for dynamic graphs. For dynamic graphs, it is also important to build compact models for computation efficiency. Another work which is most related to our work is Poincaré-GCN [23] which used Poincaré geometry. However, this work is not mathematically rigorous since they assumed that the input embedding features lie on the Poincaré model, which is not naturally satisfied. Besides, the learned model is much bigger than ours. The work in [24] presented an interesting survey on hyperbolic neural networks.

3. METHODS

3.1. Notations

A human skeleton can be represented by a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of n human joints (graph nodes) in the human body and $\mathcal{E}$ is the set of connections or bones (graph edges) between human joints. The edge set $\mathcal{E}$ can be encoded in an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ where $\mathbf{A}_{i,j} \in [0, 1)$ if there is a link between v_i and v_j otherwise, $\mathbf{A}_{i,j} = 0$. Each node v_i has a feature vector $x_i \in \mathbb{R}^d$ of dimension d. Initially, the feature vector is the $2D$ or $3D$ location of the human joint. $\mathbf{X}_t = \{x_1, x_2, \ldots, x_n\}$ is the set of feature vectors of all nodes at time step t where $\mathbf{X}_t \in \mathbb{R}^{n\times d}$. A human action can be observed over T frames.

3.2. Spatial Temporal Graph Convolutional Networks

For the spatial domain, the GCN update step can be formulated as:

$$\mathbf{X}_t^{out} = \sigma(\mathbf{\Lambda}^{-1/2}(\mathbf{A}+\mathbf{I})\mathbf{\Lambda}^{-1/2}(\mathbf{X}_t^{in}\mathbf{W}^{in} + \mathbf{B}^{in})) \quad (1)$$

where σ is an activation function. $\Lambda^{ii} = 1 + \sum_j A^{ij}$ and $\mathbf{\Lambda}$ is a diagonal matrix. $\mathbf{I}$ is the identity matrix to keep identity features. $(\mathbf{A}+\mathbf{I})$ then makes the output feature vector x_i^{out} for every node i as a function of its input feature vector x_i^{in} and the feature vector x_j^{in} for any node $j \in$ neighboring set for node i which is encoded in matrix $\mathbf{A}$. $\mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{\Lambda}^{-\frac{1}{2}}$ is the normalized adjacency matrix to normalize the weights for the nodes in the neighboring set. $\mathbf{W^{in}}$ is the weight matrix corresponding to $\mathbf{X}_t^{in}$ and $\mathbf{B^{in}}$ is the bias translation matrix. To achieve better performance and to increase model capacity, a partitioning strategy can be applied and $\mathbf{A}+\mathbf{I}$ can be decomposed into a number of matrixes $\mathbf{A}_j$ such that $\mathbf{A}+\mathbf{I} = \sum_j \mathbf{A}_j$. For the temporal domain, a simple convolution can be applied to nodes in consecutive frames.

3.3. HST-GCNs

A hyperbolic space is a non-Euclidean space with a constant negative curvature. Many models were introduced to represent and model a hyperbolic space such as the Lorentz model, the Poincaré model and the Klein model. We use the Lorentz model (also called the hyperboloid model) as it is simple and numerically more stable[21].

Let $\langle ., . \rangle_{\mathcal{L}} : \mathbb{R}^{d+1} \times \mathbb{R}^{d+1} \to \mathbb{R}$ represents the Minkowski inner product where $\langle x, y\rangle_{\mathcal{L}} := \sum_{i=1}^{d} x_i y_i - x_0 y_0$. Let $\mathbb{H}^{d,K}$ be a d dimensional hyperboloid model with a constant negative curvature $^{-1}/_K$ where $K > 0$. Then we have:

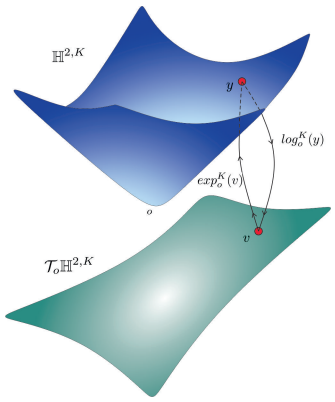

Fig. 1. Mapping between a 2D hyperboloid and the tangent 2D hyperplane (translated down for illustration purposes) at the origin.

$$\mathbb{H}^{d,K} := \{x \in \mathbb{R}^{d+1} : \langle x, x\rangle_{\mathcal{L}} = -K, x_0 > 0\} \quad (2)$$

Note that $x_0 > 0$ to indicate the upper half of the hyperboloid manifold. Let $\mathcal{T}_x\mathbb{H}^{d,K}$ be the Euclidean tangent space centered at point $x \in \mathbb{H}^{d,K}$. Then we have:

$$\mathcal{T}_x\mathbb{H}^{d,K} := \{v \in \mathbb{R}^{d+1} : \langle v, x\rangle_{\mathcal{L}} = 0\} \quad (3)$$

To map a point $y \in \mathbb{H}^{d,K}$ to the tangent space $\mathcal{T}_x\mathbb{H}^{d,K}$ centered at point $x \in \mathbb{H}^{d,K}$ such that $x \neq y$, the logarithmic map can be used which is defined as:

$$log_x^K(y) = d_{\mathcal{L}}^K(x, y)\frac{y + {}^1/_K\langle x, y\rangle_{\mathcal{L}} x}{\|y + {}^1/_K\langle x, y\rangle_{\mathcal{L}} x\|_{\mathcal{L}}} \quad (4)$$

where $\|x\|_{\mathcal{L}} = \sqrt{\langle x, x\rangle_{\mathcal{L}}}$ is the norm of x, $d_{\mathcal{L}}^K(x, y)$ is the Minkowskian distance between two points x and y in $\mathbb{H}^{d,K}$ and is given by:

$$d_{\mathcal{L}}^K(x, y) = \sqrt{K}\,\text{arcosh}(^{-\langle x,y\rangle_{\mathcal{L}}}/_K) \quad (5)$$

To map a point $v \in \mathcal{T}_x\mathbb{H}^{d,K}$ where $x \in \mathbb{H}^{d,K}$ to the hyperboloid manifold such that $v \neq 0$, we use the exponential map defined as:

$$exp_x^K(v) = \cosh(\frac{\|v\|_{\mathcal{L}}}{\sqrt{K}})x + \sqrt{K}\sinh(\frac{\|v\|_{\mathcal{L}}}{\sqrt{K}})\frac{v}{\|v\|_{\mathcal{L}}} \quad (6)$$

The logarithmic and exponential maps represent a bijection between the tangent space at a point and the hyperboloid. Figure 1 illustrates the mapping between the hyperbolic space $\mathbb{H}^{d,K}$ and the tangent space at the origin o which is $\mathcal{T}_o\mathbb{H}^{d,K}$ for $d = 2$.

Parallel transport is used to perform translation in the hyperbolic space. $P_{x\to y}(.)$ maps a point $u \in \mathcal{T}_x\mathbb{H}^{d,K}$ to a point $u' \in \mathcal{T}_y\mathbb{H}^{d,K}$ for $x, y \in \mathbb{H}^{d,K}$ and $x \neq y$. The parallel transport of a point $u \in \mathcal{T}_x\mathbb{H}^{d,K}$ to the tangent space $\mathcal{T}_y\mathbb{H}^{d,K}$ is:

$$P_{x\to y}(u) = u - \frac{\langle log_x^K(y), u\rangle_{\mathcal{L}}}{d_{\mathcal{L}}^K(x, y)^2}(log_x^K(y) + log_y^K(x)) \quad (7)$$

The hyperbolic bias addition can then be defined as:

$$x^H \oplus^K b := exp_{x^H}^K(P_{o\to x_H}(b)) \quad (8)$$

where o is the origin and the bias b is a learnable Euclidean vector defined at the tangent space of the origin.

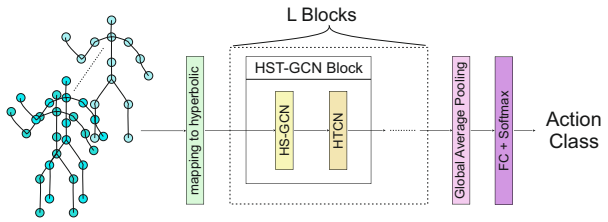

Fig. 2. The full HST-GCN model architecture. The HST-GCN block consists of HS-GCN module (Eq. 9) and HTCN module. FC is the fully connected layer.

For the hyperbolic space, Eq. 1 can be rewritten as:

$$\mathbf{X}_t^{out\_H} = exp_o^K(\sigma(log_o^{K_J}(\sum_j^{\oplus_{K_j}} exp_o^{K_j}(\mathbf{\Lambda}_j^{-1/2}\mathbf{A}_j\mathbf{\Lambda}_j^{-1/2} (log_o^{K_{l-1}}(\mathbf{X}_t^{in\_H})\mathbf{W}_j^{in})) \oplus^{K_j} \mathbf{B}_j^{in}))) \quad (9)$$

where the input and output features are in the hyperbolic space (denoted by _H), $\sum_j^{\oplus_{K_j}}$ is the summation in the hyperboloid model (Möbius addition of points on the hyperboloid) over the partitioning sets where $j \in \{1, 2, \ldots, J\}$. We call this module the Hyperbolic Spatial GCN (HS-GCN). Note that for the 2D or 3D input skeleton joints positions or any initial input feature X_t^{in}, we have $(0, X_t^{in}) \in \mathcal{T}_o\mathbb{H}^{d,K}$ as from Eq. 3, we get $\langle (0, X_t^{in}), o\rangle_{\mathcal{L}} = 0$. Using the exponential map (Eq. 6 at the origin o), we can obtain the initial hyperbolic input feature $\mathbf{X}_t^{in\_H}$. For $j = 1$, $log_o^{K_{l-1}}(\mathbf{X}_t^{in\_H})$ maps $\mathbf{X}_t^{in\_H}$ to the tangent space of the manifold used in the previous layer which can be the initial input feature vector.

To enlarge the receptive field and obtain features from farther away joints, we use a disentangling approach [11] to get features from up to M-hop neighbors which effectively enlarge the receptive field of nodes. The m-hop adjacency matrix $\mathbf{A}_m$ is then given by:

$$[\mathbf{A}_m]_{i,j} = \begin{cases} 1 & \text{if } \; d(v_i, v_j) = m, \\ 0 & \text{otherwise}, \end{cases} \quad (10)$$

where $d(v_i, v_j)$ is the shortest distance between node v_i and node v_j. To obtain features from up to M-hop neighbors, we use $\mathbf{A}_m$ for $m = 0, \ldots, M$. Using the spatial configuration partitioning strategy, we get $2M+1$ subsets for M-hop neighbors. These adjacency matrixes can be used in the HS-GCN module (Eq. 9) to enlarge the receptive field and to obtain enhanced features.

For the temporal domain, the hyperbolic features are mapped to the tangent space ($exp_o^K(\mathbf{X}_t^{out\_H})$) where the convolution across time domain is performed then the resulting features are mapped back to the hyperboloid using $exp_o^{K_l}(\ldots)$ to generate the output for the next layer. This module is the Hyperbolic Temporal Convolutional Network (HTCN). Figure 2 shows the full model architecture.

4. EXPERIMENTS

Here, we provide extensive experiments on three skeleton datasets.

4.1. Datasets

NTU RGB+D 60 is a large scale dataset for 3D human activity analysis which has 60 human action classes. The authors recommend two benchmarks for this dataset: (1) Cross-Subject (X-Sub) and (2) Cross-View (X-View). NTU RGB+D 120 extends NTU RGB+D 60 to have 120 action classes and he authors recommend replacing the Cross-View setting with a Cross-Setup (X-Set) setting. Kinetics-Skeleton is a large-scale human action dataset that has 400 classes. The top-1 and top-5 accuracy on the testing set are reported.

M =	1	2	3	4	5	6
Acc(%)	58.2	75.3	75.1	**76.1**	74.3	74.3
Params(M)	0.01	0.02	0.02	0.02	0.02	0.02

Table 1. Ablation study on the number of neighbors (M) used for feature aggregation using NTU RGB+D 60 Cross-Subject benchmark. Params(M) is the parameters in millions.

C,L	EST-GCN	HST-GCN	Params(M)
16,1	53.7	59.4	0.01
32,1	63.5	66.3	0.01
48,1	64.8	71.3	0.03
64,1	65.0	64.1	0.04
16,2	67.6	73.5	0.01
24,2	70.9	**76.1**	0.02
32,2	71.9	73.8	0.03

Table 2. Performance comparison between hyperbolic models and the corresponding Euclidean ones on the NTU RGB+D 60 Cross-Subject benchmark. C is the number of channels and L is the number of layers in the network.

4.2. Ablation study

4.2.1. Number of neighbors for feature aggregation

Table 1 shows the accuracy and number of parameters for a 2-layer HST-GCN to determine M. Using a 4-hop neighbors for disentangled feature aggregation gives the best performance. More information can be captured from further joints away which is particularly important for shallow models.

4.2.2. Network configuration on different datasets

Table 2 shows the performance of HST-GCNs using different configurations. For the NTU RGB+D 120 and the Kinetics-Skeleton 400 datasets, we conducted similar experiments and found that the best performance can be obtained using the configuration (C,L) = (40,2).

4.3. Performance of hyperbolic vs Euclidean models

We use the NTU RGB+D 60 Cross-Subject benchmark in this experiment. Table 2 shows this comparison for different network configurations. HST-GCNs clearly outperform the corresponding Euclidean ST-GCNs (EST-GCNs) by about 5% for most of the network configurations specially the low-dimensional ones. This shows that enhanced discriminative features can be obtained and embedded in the hyperbolic manifold which increases model performance.

Table 3 shows the performance of HST-GCNs on the NTU datasets and Kinetics dataset. These are light-weight compact models when compared to large existing Euclidean models. For example, the number of parameters in HST-GCN is only about 0.5% and 0.6%

Model	Dataset	NTU RGB+D 60		NTU RGB+D 120		Kinetics-Skeleton 400	
		X-Sub	X-View	X-Sub	X-Set	Top-1	Top-5
EST-GCN	**Acc(%)**	70.9	80.2	63.2	64.8	12.1	29.2
HST-GCN	**Acc(%)**	**76.1**	**81.1**	**67.8**	**69.4**	**17.9**	**38.1**
	Params(M)	0.02	0.02	0.05	0.05	0.06	0.06

Table 3. HST-GCNs and the Euclidean counterparts (EST-GCNs) performance on different datasets.

Model	C,L	Params(M)	NTU RGB+D 60		NTU RGB+D 120		Kinetics-Skeleton 400	
			X-Sub	X-View	X-Sub	X-Set	Top-1	Top-5
AAGCN	32,6	0.26	81.8/83.7	90.8/91.6	74.0/76.5	77.8/79.7	24.2/24.8	46.1/46.6
	48,6	0.56	83.7/85.8	91.1/91.8	75.2/78.0	78.9/80.4	26.9/27.2	49.3/49.6
	32,9	0.98	84.7/85.8	91.6/92.7	75.7/77.0	79.6/81.0	29.4/29.5	52.1/52.1
	48,9	2.15	85.8/86.8	92.5/93.0	76.7/78.4	79.5/81.3	31.1/31.5	53.5/53.7
	64,9	3.87	86.6/87.7	94.8/94.8	78.3/80.1	81.3/82.7	31.7/32.0	54.5/54.5
MS-G3D	56,3	1.36	88.8/88.9	94.5/94.5	82.3/82.8	84.2/84.6	31.3/31.3	53.9/53.9
	72,3	1.98	89.2/89.5	94.8/95.0	82.5/83.0	84.3/84.6	31.9/31.9	54.8/54.8
	96,3	3.20	89.3/89.4	95.0/95.0	83.3/83.5	84.4/84.7	35.8/35.8	58.8/58.8
Shift-GCN	64,9	0.69	87.8/88.2	95.1/95.1	80.8/81.5	83.2/83.7	33.8/33.8	56.4/56.4

Table 4. Boosting the performance of existing methods using HST-GCNs. The numbers separated by / represent the Euclidean model accuracy and the boosted accuracy using HST-GCN model, respectively.

of the total number of parameters in AAGCN model [10]) and MS-G3D model [11]), respectively. HST-GCNs outperform EST-GCNs by more than 5% for most of the datasets benchmarks, which shows the superiority of HST-GCNs.

4.4. Boosting the performance of existing methods

We show that HST-GCNs can be used to boost the performance of other methods. In addition, we show that by using smaller versions of these Euclidean models combined with HST-GCNs is comparable to or outperforms the larger versions of these models. Table 4 shows the performance of different sizes of AAGCN models [10] and MS-G3D models [11] on different datasets. The table also shows the boosted performance of these models and shift-GCN model [20] using the corresponding HST-GCN model from Table 3. For each method, the last row is the original model introduced by the authors. For the AAGCN model, a comparable or better performance with 45% parameters reduction. For the MS-G3D model, we achieved comparable or better performance with 40% parameters reduction. For the NTU RGB+D 120 cross-set benchmark, a better performance was achieved with 60% parameters reduction. Similarly, HST-GCNs can be used to boost the performance of any other models.

4.5. Comparison with SOTA

Table 5 shows the comparison between different methods on the NTU RGB+D 60 dataset. Our method achieved comparable or better performance using smaller or comparable number of parameters.

Method	X-Sub	X-View	Params(M)
ST-GCN [8]	81.5	88.3	3.10
SR-TSL [16]	84.8	92.4	19.07
RAGCN [25]	85.9	93.5	6.21
AAGCN [10]	86.6	94.8	3.87
AS-GCN [26]	86.8	94.2	9.50
NAS-GCN [17]	87.4	94.6	6.57
Poincaré-GCN [23]	87.8	95.0	2.62
Shift-GCN [20]	87.8	**95.1**	**0.69**
DC-GCN+ADG [19]	**88.2**	**95.2**	1.24
Ours (small)	**88.2**	**95.1**	**0.71**
AGC-LSTM [27]	89.2	95.0	22.89
PL-GCN [28]	89.2	95.0	20.70
MS-G3D [11]	89.3	95.0	3.20
Ours	**89.5**	95.0	**2.00**

Table 5. Comparison between different methods on the NTU RGB+D 60 dataset.

5. CONCLUSION

In this work, we showed that using the hyperbolic space to embed human action features is more superior than using the Euclidean space as in classical ST-GCNs. At the same time, HST-GCNs can be used with existing methods to build compact models to achieve comparable performance. We believe that this work has a great potential and hope it motivates researchers to take advantage of the hyperbolic embedding space in different research fields.

6. ACKNOWLEDGEMENTS

This work was supported by the Academy of Finland for Academy Professor project EmotionAI (grants 336116, 345122) and ICT 2023 project (grant 328115), as well as the CSC-IT Center for Science, Finland, for computational resources.

7. REFERENCES

[1] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang 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*, 2016, pp. 1010–1019.

[2] Tae Soo Kim and Austin Reiter, "Interpretable 3d human action analysis with temporal convolutional networks," in *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*. IEEE, 2017, pp. 1623–1631.

[3] Bo Li, Yuchao Dai, Xuelian Cheng, Huahui Chen, Yi Lin, and Mingyi He, "Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep cnn," in *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2017, pp. 601–604.

[4] Yong Du, Wei Wang, and Liang Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1110–1118.

[5] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng, "View adaptive recurrent neural networks for high performance human action recognition from skeleton data," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2117–2126.

[6] Thomas N. Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," 2017.

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, pp. 3844–3852, 2016.

[8] Sijie Yan, Yuanjun Xiong, and Dahua Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[9] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *CVPR*, 2019.

[10] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing LU, "Skeleton-Based Action Recognition with Multi-Stream Adaptive Graph Convolutional Networks," *arXiv:1912.06971 [cs]*, Dec. 2019.

[11] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 143–152.

[12] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang, "Constructing stronger and faster baselines for skeleton-based action recognition," 2021.

[13] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec, "Hyperbolic graph convolutional neural networks," *Advances in neural information processing systems*, vol. 32, pp. 4868–4879, 2019.

[14] Mikhael Gromov, "Hyperbolic groups," in *Essays in group theory*, pp. 75–263. Springer, 1987.

[15] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 588–595.

[16] Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan, "Skeleton-based action recognition with spatial reasoning and temporal stack learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 103–118.

[17] Wei Peng, Xiaopeng Hong, Haoyu Chen, and Guoying Zhao, "Learning graph convolutional network for skeleton-based human action recognition by neural searching," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, pp. 2669–2676.

[18] Fanfan Ye, Shiliang Pu, Qiaoyong Zhong, Chao Li, Di Xie, and Huiming Tang, "Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 55–63.

[19] Ke Cheng, Yifan Zhang, Congqi Cao, Lei Shi, Jian Cheng, and Hanqing Lu, "Decoupling gcn with dropgraph module for skeleton-based action recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[20] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu, "Skeleton-based action recognition with shift graph convolutional network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[21] Maximillian Nickel and Douwe Kiela, "Learning continuous hierarchies in the lorentz model of hyperbolic geometry," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3779–3788.

[22] Qi Liu, Maximilian Nickel, and Douwe Kiela, "Hyperbolic graph neural networks," 2019.

[23] Wei Peng, Jingang Shi, Zhaoqiang Xia, and Guoying Zhao, "Mix dimension in poincaré geometry for 3d skeleton-based action recognition," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1432–1440.

[24] Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao, "Hyperbolic deep neural networks: A survey," 2021.

[25] Yi-Fan Song, Zhang Zhang, and Liang Wang, "Richly activated graph convolutional network for action recognition with incomplete skeletons," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1–5.

[26] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3595–3603.

[27] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan, "An attention enhanced graph convolutional lstm network for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1227–1236.

[28] Linjiang Huang, Yan Huang, Wanli Ouyang, and Liang Wang, "Part-level graph convolutional network for skeleton-based action recognition," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 11045–11052, Apr. 2020.