Mix Dimension in Poincaré Geometry for 3D Skeleton-based Action Recognition

Wei Peng, Jingga Shi†, Zhaoqiang Xia, Guoying Zhao

1Center for Machine Vision and Signal Analysis, University of Oulu, Finland
2School of Software Engineering, Xi’an Jiaotong University, Xi’an, China
3Northwestern Polytechnical University, Xi’an, China; 4School of Information and Technology, Northwest University, Xi’an, China

Abstract

Graph Convolutional Networks (GCNs) have already demonstrated their powerful ability to model the irregular data, e.g., skeletal data in human action recognition, providing an exciting new way to fuse rich structural information for nodes residing in different parts of a graph. In human action recognition, current works introduce a dynamic graph generation mechanism to better capture the underlying semantic skeleton connections and thus improves the performance. In this paper, we provide an orthogonal way to explore the underlying connections. Instead of introducing an expensive dynamic graph generation paradigm, we build a more efficient GCN on a Riemann manifold, which we think is a more suitable space to model the graph data, to make the extracted representations fit the embedding matrix. Specifically, we present a novel spatial-temporal GCN (ST-GCN) architecture which is defined via the Poincaré geometry such that it is able to better model the latent anatomy of the structure data. To further explore the optimal projection dimension in the Riemann space, we mix different dimensions on the manifold and provide an efficient way to explore the dimension for each ST-GCN layer. With the final resulted architecture, we evaluate our method on two current largest scale 3D datasets, i.e., NTU RGB+D and NTU RGB+D 120. The comparison results show that the model could achieve a superior performance under any given evaluation metrics with only 40% model size when compared with the previous best GCN method, which proves the effectiveness of our model.

Introduction

Human action recognition is one of the most important topics in computer vision studies. It could contribute to many potential applications such as human behavior analysis, video understanding, and virtual reality. In general, several different kinds of modalities, e.g., appearance, depth, optical flow, and skeletal data, are utilized in the action recognition tasks. Recently, skeleton-based human action recognition has attracted considerable attention, since the compact skeleton data make the models more efficient and more robust to the variations of viewpoints and environments.

In this paper, we focus on the problem of 3D skeleton-based action recognition and expect to provide a more robust neural network for this task. Recently, graph convolutional networks (GCNs) (Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2016), which have been designed well to preserve the natural topology structure of the skeleton, followed by a temporal convolutional network, i.e., ST-GCNs, have been successfully adopted in skeleton-based action recognition (Yan, Xiong, and Lin 2018; Peng et al. 2020; Shi et al. 2019). Yan et al. first proposed spatial-temporal graph convolutional network (Yan, Xiong, and Lin 2018) for this task, which decouples the neural architecture into a GCN to capture the spatial information and an 1D convolutional filter to model the dynamic information. At the GCN part, most current ST-GCN approaches provide a pre-defined graph embedding matrix to encode the skeleton topology. This matrix and the skeleton sequence data together are fed into the ST-GCNs to extract high-level representations. However, as mentioned in work (Peng et al. 2020), a fixed graph embedding matrix will introduce the constraints into the feature learning progress, which may be harmful for the representation at higher layers, leading to a negative influence to final classification. Therefore, works in (Shi et al. 2019; Peng et al. 2020) present either a global or a layer-wise dynamic graph generation paradigm to break the learning constraint. Experiments (Shi et al. 2019; Peng et al. 2020) prove that the the dynamic graph generation mechanism could further improve the performance for this task.

This paper aims to deal with the skeleton-based human action recognition tasks from another perspective. Instead of providing a dynamic graph embedding, we turn to explore a better modelling space for the skeleton graph sequences. Despite the success of current feature representation with deep neural networks in Euclidean space, graph data is proved to exhibit a highly non-Euclidean latent anatomy (Bronstein et al. 2017). However, as far as we know, all the previous ST-GCNs (Yan, Xiong, and Lin 2018; Shi et al. 2019; Peng et al. 2020) are defined in the Euclidean space, which may not be the optimal choice for modelling the hierarchical graph data. We argue that neural network operations directly defined in a data-related space, e.g., Hyperbolic manifold...
The approach learns a multidimensional structural summarization as follows: We present a novel spatial temporal graph convolutional neural network layer. Finally, with the resulted ST-GCN, we provide a more distinguished representation, an efficient way to explore the dimension for each graph convolutional operations will be conducted on the tangent space. Since there is a bijection between them, the convolutional operations will be conducted on the tangent spaces and then the extracted features will be projected back as a trajectory on the manifold. In this way, we can get the graph embedding on the hyperbolic space by projecting the feature back to the manifold. To further explore the optimal projection dimension in this non-Euclidean space, we mix different dimensions in the hyperbolic space and provide an efficient way to explore the dimension for each graph neural network layer. Finally, with the resulted ST-GCN, we evaluate our method on two currently most challenging 3D skeleton-based datasets, i.e., NTU RGB+D (Shahroudy et al. 2016) and NTU RGB+D 120 (Liu et al. 2019a). Compared with various state-of-the-art approaches, our method gets the best results under any given evaluation protocols. Moreover, in terms of the model size, current best model (Peng et al. 2018) is even 2.5 times bigger than our model, which proves the effectiveness of our method. Our contribution can be summarized as follows:

- We present a novel spatial temporal graph convolutional network via Poincaré geometry, which gives a brand-new ST-GCN to model the graph sequences on the Riemann manifold.
- The approach learns a multidimensional structural embedding for each graph based on a Poincaré model. To provide a more distinguished representation, an efficient way is provided to explore a better projection space by mixing the dimensions on the Poincaré model.
- To evaluate the effectiveness of our method, comprehensive experiments are conducted on two current most challenging 3D skeleton-based action recognition tasks. Results shows that our model obtains the best classification accuracy under any given metrics with an efficient fashion.

The rest of this paper is organized as follows. Section reviews related approaches and discusses their relationships to the present work. Section gives a detailed description about the methodology and the corresponding neural architecture. We conduct the experiments in Section and report the results obtained on the various datasets. In this section, experiments include an ablation experiment performed on the NTU RGB+D dataset and the performance comparisons with the state-of-the-art approaches. Finally, a conclusion is drawn in Section.

**Related Works**

The work presented here is to deal with 3D skeleton-based action recognition with a new graph convolutional network (GCN). Therefore, in this section, we will give a briefly review about the related topics, including GCN and human action recognition. Since we will define the neural network on the Riemannian space, here we will also give a short introduction about the current progress on the neural network on Riemann Manifold.

**Graph Convolutional Networks (GCNs)**

Generalizing convolutional neural networks from regular data, e.g., images, to irregular data, e.g., graph data, has been an active topic in recent years. As one of the most successful representative, Graph Convolutional Networks (GCNs), including spatial-temporal graph convolutional networks (ST-GCNs), increasingly attract attention and get promising results in many research fields.

Mainly, according to how the graph convolutions are defined, the GCN architectures can be divided into two categories, i.e., the spectral-domain and the spatial-domain methods. The spectral one converts graph data into its spectrum and applies a filter in spectral domain. There are many representative works in this stream (Kipf and Welling 2016; Defferrard, Bresson, and Vandergheynst 2016), and it could deal with the whole graph at the one time. Meanwhile, it is time-consuming especially for large graphs. Another limitation of the spectral method is that the spectral construction is limited to a single domain since the spectral filter coefficients are basis dependent. If one could construct compatible orthogonal bases across different domains, this problem can be solved. However, such construction requires expert knowledge of the correspondence between domains, which is extremely difficult in most cases. Instead of defining an operation in another domain, spatial based methods directly design convolution operation in spatial domain (Monti et al. 2017; Velicković et al. 2018), which resembles the traditional convolutional filter as it is for images. The spatial-domain methods are more scalable to large graphs as they directly perform the convolution in the graph domain via information aggregation. However, the disadvantage is that it is difficult to model the global structure. To
model the dynamic information for graph sequences, e.g., skeletal video for action recognition, many ST-GCNs (Yan, Xiong, and Lin 2018; Shi et al. 2019; Peng et al. 2020) are proposed. However, as far as our knowledge, all of them are defined in the Euclidean space. In this paper, we propose a brand-new GCN model, which is applied to human action recognition. It is totally different from previous GCNs as it is defined on a Non-Euclidean space. We also notice that there are also works (Nickel and Kiela 2017; Ganea, Bécigneul, and Hofmann 2018; Liu, Nickel, and Kiela 2019) define neural networks in Riemannian space, but they are either based on conventional forward networks or just design transforming and aggregating functions for a network with only two or three layers. Instead, we provide a deep spatial temporal graph convolutional network to deal with dynamic graph sequences.

**Human Action Recognition**

Action recognition (Yan, Xiong, and Lin 2018; Peng, Hong, and Zhao 2019; Peng et al. 2020) is one of the important areas in both industry and academia. We can find numbers of previous action recognition works based on RGB images or videos, which are available everywhere in real life. With the great advancements of deep learning, various approaches employ CNNs and RNNs to capture the spatial-temporal information and thus to recognize human actions. These CNNs and RNNs methods, which automatically learn useful representations, achieve outstanding performances than the previous methods based on hand-crafted features. However, one of the main disadvantages of these approaches is that the learnt representations are prone to being distracted since entire areas of video frames are exploited to learn the representations.

Currently, the skeleton data is more easy to access and become more popular for tasks like human action recognition. Compared with RGB based action recognition, skeleton based ones increasingly attract attention since skeleton data is more compact and more robust to the complicated and changing background. The works on this topic mainly follows three streams: 1) Hand-crafted features, in which works leverage the dynamics of joint motion by using hand-crafted features, including utilizing LOP feature to overcome intra-class variance issue (Wang et al. 2012), building histograms of 3D joint locations (Xia, Chen, and Aggarwal 2012), and modelling 3D geometric relationships in a Lie group (Vemulapalli, Arrate, and Chellappa 2014). These methods require much expert knowledge to design hand-crafted features. 2) Conventional deep learning approaches. With the development of deep learning, data-driven methods have become the mainstream methods. In total, previous deep learning methods process skeleton data directly by CNNs and RNNs. Early deep-learning based approaches using skeleton data manually create a sequence of joint-coordinate vectors or as a pseudo-image, and apply the data to RNNs or CNNs to infer corresponding action classes. Work (Kim and Reiter 2017) rearranges the graph-structure skeleton data and models it as a pseudo-image based on the manually designed transformation rules, such that the constructed grid data could directly benefited from CNNs. Since the input is time series data, there are also many attempts to utilize RNN and LSTM to model the dynamic information. Representative works include (Du, Wang, and Wang 2015; Shahroudy et al. 2016; Song et al. 2017; Zhang et al. 2017; Si et al. 2018), where they model the skeleton sequences by either extending RNN to spatio-temporal domains, dividing human skeleton into parts or providing a fully connected deep LSTM for this task. Nevertheless, graph-structure data does not lie in the Euclidean space, thus the traditional neural network may be not the optimal choice to model this data. 3) Methods based on spatio-temporal graph convolutional networks, which are more popular and suitable for this task. Intuitively, skeleton-features can be represented as a graph structure since their components are homeomorphic. Therefore, joints and bones in skeletons can be defined as the vertices and connections of the graph. Yan et al. developed a spatio-temporal graph convolutional network (ST-GCN) (Yan, Xiong, and Lin 2018) to model the skeleton data as the graph structure, which leverages the powerful ability to model the irregular data and thus achieves better performance than previous methods. ST-GCN becomes a general framework to deal with the skeleton-based action recognition task. Based on this, work in (Shi et al. 2019) explores a global adaptive embedding matrix generation method, which further improves the performance. Peng et al. introduced the neural architecture search and automatically designed a GCN (Peng et al. 2020) architecture for this task. This work (Peng et al. 2020) gets the current best result. Our work is also based on ST-GCN, but we provide a much more efficient way to model the graph sequences thus even no need to construct a dynamic graph.

**Neural Networks on Riemannian Manifold**

Popular deep learning methods typically embed data into a low dimensional Euclidean vector space using a strategy such as functional similarity to capture semantic representation. This is straightforward since our intuition for the real-life world is highly related to the Euclidean space. However, in many fields, e.g., genomics, social networks, and skeleton-based action recognition, the latent anatomy of the data is well defined by non-Euclidean spaces such as Riemannian manifolds. Most of previous deep neural networks are directly applied to such data and also get promising performance due to the powerful ability of deep learning to capture patterns from data.

However, the capability of an optimal modelling space should not only reduce the computational burden but also further improve the task performances. Recently, embedding hierarchical data in Riemannian space has achieved attractive results and gained popularity in deep learning (Huang et al. 2017; Mathieu et al. 2019; Nickel and Kiela 2017). For instance, by constructing in Riemannian space, Mathieu et al. proposed a Poincaré variational Auto-Encoders (Mathieu et al. 2019) and shows a better generalisation for the hierarchical structures. Cho et al. provided a Riemannian approach to batch normalization(Cho and Lee 2017) and also achieved superior performance. Here we focus on Hyperbolic geometry, which is a non-Euclidean geometry with a constant negative Gaussian curvature. One significant intrinsic property
in this geometry is the exponential growth. There have already been some attempts on designing neural networks in this space. Specifically, Nickel and Kiela (Nickel and Kiela 2017) reported pioneering research on learning representation in hyperbolic spaces. Then, work in (Ganea, Bécigneul, and Hofmann 2018) introduced Hyperbolic Neural Networks, connecting hyperbolic geometry with deep learning. After that, works also provide hyperbolic analogues of conventional operations, in which several other algorithms have been developed, such as Poincaré GloVe (Tifrea, Bécigneul, and Ganea 2018) and Hyperbolic Attention Networks (Gulcehre et al. 2018). We also find that works (Liu, Nickel, and Kiela 2019; Chami et al. 2019) construct graph neural networks using hyperbolic geometry, which are similar with our work. However, our model is different from these works since we are dealing with the dynamic graph sequences while they just focus on static graph. Besides, we provide an efficient way to explore the influence of the projection dimensions for the network while none of them touch this problem.

**METHODOLOGY**

In this section, we describe our ST-GCN defined on Poincaré model. The framework is illustrated in Figure 1. The basic module of our network consists of a GCN followed by a temporal convolutional filter. Such block is stacked for multiple layers to capture the high-level representations for the graph sequences. The spatial temporal model is defined by the Poincaré geometry. Finally, the learned features are projected back to Euclidean space for the prediction. We will detail the important components of our framework in the following parts.

**Spatial-Temporal Graph Convolutional Networks**

Graph data is a very useful data structure to model complex structures such as social network, and bio-information, while it is not easy to model this kind of irregular data using neural networks. Define $\mathcal{G}$ as one frame of the skeleton sequence. Assume the skeleton is composed of $N$ nodes and the node connections are encoded in the adjacency matrix $A \in \mathbb{R}^{N \times N}$. $X \in \mathbb{R}^{N \times \delta}$ denotes the input representations for the skeleton with $N$ nodes. To extract the feature representation of $\mathcal{G}$, a Fourier transform is conducted on it such that the transformed signal could then be processed with formulation of fundamental operations such as filtering. As a result, a normalized graph Laplacian $L = I_n - D^{-1/2}AD^{-1/2}$ is used for Fourier transform. Here, the diagonal degree matrix $D$ is constructed with elements $D_{ii} = \sum_j A_{ij}$. Then a graph filtered by operator $g_{\theta}$, parameterized by $\theta$, can be formulated as

$$Y = g_{\theta}(L)X = U g_{\theta}(\Lambda)U^T X,$$  

(1)

where $Y$ is the extracted graph feature and $U$ is the Fourier basis subjected to $L = U \Lambda U^T$ with the $\Lambda$ as its corresponding eigenvalues. Suggested by (Hammond, Vandergheynst, and Gribonval 2011), the filter $g_{\theta}$ can be further approximated by a Chebyshev polynomials with $K$-th order such that the computational burden is significantly reduced. That is

$$Y = \sum_{k=0}^{K} \theta^\prime_k T_k(\hat{L})X,$$  

(2)

where $\theta^\prime_k$ denotes Chebyshev coefficients. For $k > 1$, Chebyshev polynomial $T_k(\hat{L})$ is recursively defined as

$$T_k(\hat{L}) = 2\hat{L}T_{k-1}(\hat{L}) - T_{k-2}(\hat{L})$$  

(3)

with $T_0 = 1$ and $T_1 = \hat{L}$. Here $\hat{L} = 2L/\lambda_{\text{max}} - I_n$ is normalized to [-1,1]. For Eq. (2), work in (Kipf and Welling 2016) sets $K = 1$, $\lambda_{\text{max}} = 2$. In this way, a first-order approximation of spectral graph convolutions is formed. Therefore,

$$Y = \theta_0X + \theta_1(L - I_n)X = \theta_0X - \theta_1(D^{-1/2}AD^{-1/2})X,$$

(4)

This equation can be further simplified by using an unified parameter $\theta$, which means setting $\theta = \theta_0 = -\theta_1$ and making the training process adapt the approximation error, then

$$Y = \theta(I_n + D^{-1/2}AD^{-1/2})X.$$  

(5)

We set $L = I_n + D^{-1/2}AD^{-1/2}$ in the following parts for simplicity. An ST-GCN is then built by succeeding a temporal filter after each GCN. One can stack multiple GCN layers to get high-level graph features. Generally, at middle GCN layers, $X \in \mathbb{R}^{n \times \delta}$ is with multi-channels. Thus, we get

$$Y = LX\theta.$$  

(6)

To further improve the robustness, Works in (Shi et al. 2019; Peng et al. 2020) present an ST-GCN block which generates dynamic embedding matrix based on the node correlations. Instead of providing a dynamic one, we design our ST-GCN block in Poincaré model, making the node representations fit with original graph structure. In this way, we capture the graph features with much less parameters.

**Poincaré Manifold**

In this section, we will then discuss how to define an ST-GCN on a Riemannian manifold. Here, we focus on the Poincaré model in the Hyperbolic space (Reynolds 1993), which is a maximally symmetric, simply connected Riemannian manifold with constant negative sectional curvature. Hyperbolic space is analogous to the $n$-dimensional sphere, which has constant positive curvature. As a special case, the Poincaré model can be derived using a stereoscopic projection of the hyperboloid model onto the unit circle of the $z=0$ plane. It is hard to visualize as it is hard to imagine in a curved space. We can embed models of 2D hyperbolic geometry into a pseudo-Euclidean space called Minkowski space (Tataru 2001). Here, a $n$-dimensional Minkowski space is a real vector space of real dimension $n$, in which there is a constant Minkowski metric. As illustrated in the Figure 3, the Poincaré disk i.e., a 2D Poincaré model, is constructed as a projection from the upper half hyperboloid onto the unit disk at $z=0$. Poincaré disk breaks the rule in Euclidean space. For instance, like in the Figure 4, given a
There are mainly three stages in our framework, including (a) Graph feature embedding, (b) Extracting graph in Poincaré model, and (c) Classification in the Euclidean space. At the first stage, we utilize GCN filters to capture the graph representation for each frame and temporal filters are then used to capture the dynamic information. With this output from the first stage, we mix different dimensions and then project them to the tangent space, where an ST-GCN is used to extract higher-level graph representations. The feature then is mapped back to the manifold. In this way, we model the graph on the Riemannian manifold. Note here, the ST-GCN is based on the same module in stage (a) and the manifold space here is on Poincaré model. We stack several modules from stage (b) to capture higher-level semantic representations. After that, as shown in stage (c), the graph feature is projected back to Euclidean space such that a Euclidean loss function can be used to optimize this process.

Figure 1: Illustration of our framework. There are mainly three stages in our framework, including (a) Graph feature embedding, (b) Extracting graph in Poincaré model, and (c) Classification in the Euclidean space. At the first stage, we utilize GCN filters to capture the graph representation for each frame and temporal filters are then used to capture the dynamic information. With this output from the first stage, we mix different dimensions and then project them to the tangent space, where an ST-GCN is used to extract higher-level graph representations. The feature then is mapped back to the manifold. In this way, we model the graph on the Riemannian manifold. Note here, the ST-GCN is based on the same module in stage (a) and the manifold space here is on Poincaré model. We stack several modules from stage (b) to capture higher-level semantic representations. After that, as shown in stage (c), the graph feature is projected back to Euclidean space such that a Euclidean loss function can be used to optimize this process.

Figure 2: Illustration of ST-GCN block. There are two inputs, including the L and the X in Eq. (6). Here, GCN denotes the spatial graph convolutional network, and Conv-T represents the temporal filter. Both of them are followed by batch normalization (BN) layer and an activation layer (ReLU). Moreover, a residual connection is added for each block. The output and the original L are fed into next block.
A tangent space \( T_x \mathcal{M} \) at point \( x \) is defined as the first order linear approximation of \( \mathcal{M} \) around \( x \). A Riemannian metric \( g \) is a collection of inner-products \( g_x : T_x \times T_x \to \mathbb{R} \) varying smoothly with \( x \). For the Poincaré model, the open unit ball equipped with the Riemannian metric tensor, which is defined with

\[
g_x = \left( \frac{2}{1 - \|x\|^2} \right)^2 g^E
\]

where \( g^E = \mathbf{I}_n \) denotes the Euclidean metric tensor, which is conformal to the Euclidean one. Then, a Riemannian manifold \( (\mathcal{M}, g) \) is a manifold \( \mathcal{M} \) with a group Riemannian metric \( g \).

Now, we build our spatial-temporal graph convolutional networks on the Poincaré geometry to provide a more robustness representation for the temporal graph sequences. One advantage of the hyperbolic space is that it provides a bijection between the hyperbolic space and the tangent space at a point such that the operations for points on the hyperboloid manifold can be performed in tangent space and then mapped back and vice-versa. The bijection is done by the exponential map, which maps the points on the tangent space to the manifold and is defined as \( \exp_x : T_x \mathcal{M} \to \mathcal{M} \). The logarithmic map, which, as the inverse step, maps points on the tangent space back to the manifold, is defined as \( \log_x : \mathcal{M} \to T_x \mathcal{M} \). Mathematically,

\[
\exp_x(v) = x \oplus \left( \frac{\tanh(\frac{\lambda_x \|v\|}{2})}{\|v\|} \right) \cdot \frac{v}{\|v\|}
\]

\[
\log_x(y) = \frac{2}{\lambda_x} \text{arctanh}(\|x \oplus y\|) \frac{-x \oplus y}{\|x \oplus y\|}
\]

where \( v \) is a tangent vector, and \( \lambda_x = \frac{2}{1 - \|x\|^2} \) is a conformal factor. \( \oplus \) is the Mbius addition for any \( x, y \in \mathcal{M} \):

\[
x \oplus y = \frac{(1 + 2 \langle x, y \rangle + \|y\|^2)x + (1 - \|x\|^2)y}{1 + 2 \langle x, y \rangle + 2\|x\|^2\|y\|^2}
\]

With the aforementioned projection functions, we perform the GCN operations on the Poincaré models. Since there is no definition of vector space in the Riemannian networks on the Poincaré geometry to provide a more robustness representation for the temporal graph sequences. Formally, the distance for \( x, y \in \mathcal{M} \) are defined as:

\[
d(x, y) = \text{arcosh}\left(1 + 2 \frac{\|x - y\|^2}{1 - \|x\|^2(1 - \|y\|^2)}\right)
\]

A tangent space \( T_x \mathcal{M} \) at point \( x \) is defined as the first order linear approximation of \( \mathcal{M} \) around \( x \). A Riemannian metric \( g \) is a collection of inner-products \( g_x : T_x \times T_x \to \mathbb{R} \) varying smoothly with \( x \). For the Poincaré model, the open unit ball equipped with the Riemannian metric tensor, which is defined with

\[
g_x = \left( \frac{2}{1 - \|x\|^2} \right)^2 g^E
\]

where \( g^E = \mathbf{I}_n \) denotes the Euclidean metric tensor, which is conformal to the Euclidean one. Then, a Riemannian manifold \( (\mathcal{M}, g) \) is a manifold \( \mathcal{M} \) with a group Riemannian metric \( g \).

Now, we build our spatial-temporal graph convolutional networks on the Poincaré geometry to provide a more robustness representation for the temporal graph sequences. One advantage of the hyperbolic space is that it provides a bijection between the hyperbolic space and the tangent space at a point such that the operations for points on the hyperboloid manifold can be performed in tangent space and then mapped back and vice-versa. The bijection is done by the exponential map, which maps the points on the tangent space to the manifold and is defined as \( \exp_x : T_x \mathcal{M} \to \mathcal{M} \). The logarithmic map, which, as the inverse step, maps points on the tangent space back to the manifold, is defined as \( \log_x : \mathcal{M} \to T_x \mathcal{M} \). Mathematically,

\[
\exp_x(v) = x \oplus \left( \frac{\tanh(\frac{\lambda_x \|v\|}{2})}{\|v\|} \right) \cdot \frac{v}{\|v\|}
\]

\[
\log_x(y) = \frac{2}{\lambda_x} \text{arctanh}(\|x \oplus y\|) \frac{-x \oplus y}{\|x \oplus y\|}
\]

where \( v \) is a tangent vector, and \( \lambda_x = \frac{2}{1 - \|x\|^2} \) is a conformal factor. \( \oplus \) is the Mbius addition for any \( x, y \in \mathcal{M} \):

\[
x \oplus y = \frac{(1 + 2 \langle x, y \rangle + \|y\|^2)x + (1 - \|x\|^2)y}{1 + 2 \langle x, y \rangle + 2\|x\|^2\|y\|^2}
\]

With the aforementioned projection functions, we perform the GCN operations on the Poincaré models. Since there is no definition of vector space in the Riemannian space, inspired by (Ganea, Bécigneul, and Hofmann 2018), we conduct the feature extraction on the tangent space by projecting the graph embeddings with the logarithmic map. In this way, the Euclidean transformation conducted by the neural operations can be utilized for the feature in hyperbolic space. Specifically, as shown in Fig. 1, there are three stages in our framework. The input raw data is first encoded by a feature embedding network. Then, the captured feature is projected into a tangent space and filtered by a GCN and a temporal filter, which are

\[
Y_i = L \log_x(X_i) \Theta,
\]

![Figure 3: Illustration of the 2D Poincaré model. For any point on the hyperboloid, we form a line by extending it out to a focal point (0,0,-1). Then, the intersection point on the z=0 plane will be its projection point in the Poincaré model.](image-url)
Experiments

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

Datasets

In the experiments, we evaluate our method on two current largest scale 3D skeleton-based action recognition datasets, i.e., NTU RGB+D (Shahroudy et al. 2016) and NTU RGB+D 120 (Liu et al. 2019a).

**NTU RGB+D** NTU RGB+D (Shahroudy et al. 2016) is currently one of the most widely used action recognition datasets. There are four modalities, including RGB videos, depth sequences, infrared videos and 3D skeleton data. Here, we only conduct experiments on the skeleton data. There are totally 56,880 video clips involving 60 human action classes and the max number of frames in each sample is 300. They are captured from 3 cameras with different settings. In this dataset, each actor skeleton contains 25 3D joints coordinates and there are at most two actors in each video clip. The original benchmark provided two evaluations, which are Cross-Subject (X-subject) and the Cross-View (X-view) evaluations. In the X-subject evaluation, the training set contains 40,320 videos from 20 subjects, and the rest 16,560 video clips are used for testing. In the X-view evaluation, 37,920 videos captured from cameras No.2 and No.3 are used in the training and the rest 18,960 videos from camera No.1 are used for testing. We follow the original two benchmarks and report the Top-1 accuracy. For inputs with more than one stream, e.g., bones, a score-level fusion result is reported.

**NTU RGB+D 120** NTU RGB+D 120 (Liu et al. 2019a) is the extended version of the aforementioned NTU RGB+D dataset. It is a more challenging dataset with high variability since it involves more subjects and action categories. There are totally 114,480 samples which cover 120 classes. They are captured from 106 distinct subjects in a wide range of age distribution. There are two different evaluation protocols in this new released dataset: Cross-Subject (X-subject), like in the previous dataset, which splits subjects in half to training and testing parts; Cross-Setup (X-setup), which divides samples by 32 camera setup IDs. The even setup IDs is utilized for training (16 setups) and odd setup IDs for testing (16 setups). Here, 32 IDs are based on different camera setups, e.g., different heights with different horizontal angles. In consistent with the original benchmark method, Top-1 accuracy is reported in the two benchmarks.

Before the data was fed into the networks, there should be some pre-processing such that the data structure for each video clip is unified. We keep it consistent with the previous best methods (Shi et al. 2019; Peng et al. 2020) for fair comparison. All the skeleton sequences are unified to 300 frames with two actors. For the single-actor data, the second body will be padded with all zeros. Each dimension of the 3D coordinates is put into three channels as inputs. Like (Shi et al. 2019; Peng et al. 2020), we also compute the second-order information of joint, i.e., bones.

Architecture and Training

The basic framework is as illustrated in Figure 1. Firstly, we build an embedding model with the original ST-GCN module (like in Fig. 2), in which the graph convolutional layer is followed by a temporal convolution with a kernel size 9 × 1 to capture the dynamic information. The three dimensional node coordinates are projected into a space with 64D. Then, as shown in the second stage of the framework, we build networks on manifold. A group of manifold projection, [32, 48, 64, 80, 96], is constructed. The corresponding ST-GCN layers are also built on Poincaré model. We empirically stack six layers on this manifold space and divide them into three levels. For each level, we insert two ST-GCN layers and double the projection group setting based on the previous level, which means the mixed dimensions at second and third level are [64, 96, 128, 160, 192], and [128, 192, 256, 320, 384], respectively. Finally, as shown in the third stages of Fig. 1, the graph representations modeled from the manifold space is projected back to the Euclidean space. The resulted node embeddings are averaged to a feature vector. Then a fully connected layer followed by a softmax function is utilized to predict a class prediction.
Ablation Study

In this section, we evaluate the effectiveness of our method on the NTU RGB+D dataset under the given two evaluation metrics. Here, current state-of-the-art ST-GCN (Yan, Xiong, and Lin 2018) works as a baseline since we share the same ST-GCN module. To further evaluate how much we can gain from the Poincaré geometry, we implement a smaller ST-GCN module. To further evaluate that how much we can prove the effectiveness of our model.

Comparison with the State-Of-The-Art methods

First, we conduct experiments on NTU RGB+D dataset, and like (Shi et al. 2019; Peng et al. 2020), we report the best results after performing the score-level fusion on joints and bone data. Here, we compared with 14 State-Of-The-Art (SOTA) approaches under the given evaluation metrics. There is one hand-crafted method (Hu et al. 2015), but most of them are deep learning methods, including CNN-based method (Kim and Reiter 2017), RNN-based methods (Shahroudy et al. 2016; Song et al. 2017; Zhang et al. 2017; Si et al. 2018) and reinforcement learning based method (Tang et al. 2018). Besides, we also compare with various ST-GCNs (Li et al. 2018; Yan, Xiong, and Lin 2018; Li et al. 2019a; Gao et al. 2019; Li et al. 2019b; Shi et al. 2019; Peng et al. 2020), which get very promising results on this task. It is also worth mentioning that Peng et al. (Peng et al. 2020) introduced a neural architecture search (NAS) method into this task and automatically design an ST-GCN, which obtains the current best results. We also compare with this method. All the comparison results are listed in Table 2. It can be seen from Table 2 that under both X-subject and X-view evaluations, our model achieves the current best classification accuracy. Specifically, compared with current best method NAS-GCN (Peng et al. 2020), our model can achieve better results and also decrease the model size by 2.5 times.

On NTU RGB+D 120 dataset, we compared to 14 skeleton-based action recognition approaches under X-subject and X-setup evaluation metrics. The methods includes hand-crafted method (Hu et al. 2015), conventional deep learning methods, including Part-Aware LSTM (Shahroudy et al. 2016), Soft RNN (Hu et al. 2018), Spatio-Temporal LSTM (Liu et al. 2016), Internal Feature Fusion (Liu et al. 2017a), GCA-LSTM (Liu et al. 2017c), Multi-Task Learning Network (Ke et al. 2017), FSNet (Liu et al. 2019b), Skeleton Visualization (Liu, Liu, and Chen 2017), Two-Stream Attention LSTM (Liu et al. 2017b),
Multi-Task CNN with RotClips (Ke et al. 2018), and Body Pose Evolution Map (Liu and Yuan 2018). We also compare with the promising GCN-based models (Yan, Xiong, and Lin 2018; Li et al. 2019b). This task is more challenging than the above one, thus all the baselines are lower than 80%. Here, we report the best result on joint data. All the comparison results are listed in Table 3.

We can see from Table 3 that graph convolutional networks are much suitable for this task when comparing the last two methods to previous hand-crafted method and conventional deep learning methods. Under the X-subject protocol, our model improves previous best CNN-based method (Liu and Yuan 2018) by more than 15%. Compared to the GCN methods, we can still get 4.3% and 2.8% improvements under the X-setup and X-subject evaluation metrics, respectively.

Table 3: Performance comparison on NTU RGB+D 120 with 13 state-of-the-art methods. * means implemented by us.

<table>
<thead>
<tr>
<th>Methods</th>
<th>X-setup</th>
<th>X-subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part-Aware LSTM (Shahroudy et al. 2016)</td>
<td>25.5%</td>
<td>26.3%</td>
</tr>
<tr>
<td>Soft RNN (Hu et al. 2018)</td>
<td>36.3%</td>
<td>44.9%</td>
</tr>
<tr>
<td>Dynamic Skeleton (Hu et al. 2015)</td>
<td>50.8%</td>
<td>54.7%</td>
</tr>
<tr>
<td>Spatio-Temporal LSTM (Liu et al. 2016)</td>
<td>55.7%</td>
<td>57.9%</td>
</tr>
<tr>
<td>Internal Feature Fusion (Liu et al. 2017a)</td>
<td>58.2%</td>
<td>60.9%</td>
</tr>
<tr>
<td>GCA-LSTM (Liu et al. 2017c)</td>
<td>58.3%</td>
<td>59.2%</td>
</tr>
<tr>
<td>Multi-Task Learning Network (Ke et al. 2017)</td>
<td>58.4%</td>
<td>57.9%</td>
</tr>
<tr>
<td>FSNet (Liu et al. 2019b)</td>
<td>59.9%</td>
<td>62.4%</td>
</tr>
<tr>
<td>Skeleton Visualization (Liu, Liu, and Chen 2017)</td>
<td>60.3%</td>
<td>63.2%</td>
</tr>
<tr>
<td>Two-Stream Attention LSTM (Liu et al. 2017b)</td>
<td>61.2%</td>
<td>63.3%</td>
</tr>
<tr>
<td>Multi-Task CNN with RotClips (Ke et al. 2018)</td>
<td>62.2%</td>
<td>63.5%</td>
</tr>
<tr>
<td>Body Pose Evolution Map (Liu and Yuan 2018)</td>
<td>64.6%</td>
<td>66.9%</td>
</tr>
<tr>
<td>ST-GCN (Yan, Xiong, and Lin 2018)</td>
<td>71.3%</td>
<td>72.4%</td>
</tr>
<tr>
<td>AS-GCN (Li et al. 2019b)</td>
<td>78.9%</td>
<td>77.7%</td>
</tr>
<tr>
<td><strong>Ours-M</strong></td>
<td><strong>83.2%</strong></td>
<td><strong>80.5%</strong></td>
</tr>
</tbody>
</table>

Conclusions

In this paper, we address the task of video action recognition from skeleton sequences. To better modelling the graph-structure data, we propose a novel spatial-temporal graph convolutional network by applying Poincaré geometry. The graph embedding is projected into a Poincaré model where we further learn and extract a high-level of graph representation. To further explore the optimal projection dimension on the manifold space and provide a better setting for different layers, we provide an efficient exploration strategy by mixing a group of dimensions on the Poincaré model. With the resulted dimension setting for each layer, we construct our ST-GCN on this manifold space. Then, we conduct extensive comparison experiments to verify the effectiveness of our method. Comparison results on two current most challenging datasets show that the features modelled by our model are much more distinguishable than the model with the same setting in Euclidean space. Besides, our model can also get the best results for the given tasks with only 40% of the previous best model in term of the model size. All of these prove the effectiveness of our model.

Acknowledgements

This work was supported by the Academy of Finland ICT 2023 project (Grant No. 313600), Tekes Fidipro program (Grant No. 1849/31/2015) and Business Finland project (Grant No. 3116/31/2017), Infotech Oulu, and the National Natural Science Foundation of China (Grants No. 61772419). As well, the authors wish to acknowledge CSC-IT Center for Science, Finland, for computational resources.

References


Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio,


