Riemannian Graph Convolutional Network for Skeleton-Based Two-Person Interaction Recognition

Rui Wang[1] , **Zihao Bi**[1] , **Chen Hu**[1] , **Xiaoning Song**[1] , **Xiao-Jun Wu**[1] ,
Nicu Sebe [2] , **Ziheng Chen** [2*]
[1]School of Artificial Intelligence and Computer Science, Jiangnan University
[2]Department of Information Engineering and Computer Science, University of Trento
{cs_wr, wu_xiaojun, x.song}@jiangnan.edu.cn, 6233112017@stu.jiangnan.edu.cn,
niculae.sebe@unitn.it, {bzh_email, ziheng_ch}@163.com

Abstract

In the field of skeleton-based human action recognition, Graph Convolutional Networks (GCNs) have become a dominant framework. However, existing GCN-based approaches often treat the sequences of two-person interaction as separate entities, ignoring the inherent semantic dependencies and spatial correlations between interacting subjects. Furthermore, high-order skeleton representations naturally exhibit non-Euclidean structures, where Euclidean deep learning models are inherently limited in explicitly capturing and preserving such geometric information. As a countermeasure, we propose a Riemannian Graph Convolutional Network (RGCN) that operates on the Symmetric Positive Definite (SPD) manifolds. Specifically, we model high-order skeletal statistics via Gaussian embedding and propose a Riemannian network to capture inter-subject interactions and global correlations. The proposed RGCN is instantiated under three SPD geometries, and its effectiveness is validated through extensive experiments on three interaction benchmarks. Extensive experimental results show that RGCN provides a competitive and geometrically grounded alternative for skeleton-based interaction recognition.

1 Introduction

Skeleton-based recognition has gained increasing attention in the computer vision and pattern recognition fields in recent years, owing to its broad and impactful applications. Compared to the large number of pixels in RGB images, skeleton features simplify the representation of the human body into 3D keypoint coordinates. This drastically reduces the data volume and computational burden, and mitigates the challenges posed by environmental factors such as occlusion, low lighting, and deformation [Li *et al.*, 2023; Pang *et al.*, 2022].

Previous deep learning methods usually treat human joints as a set of independent features, organizing them as sequential

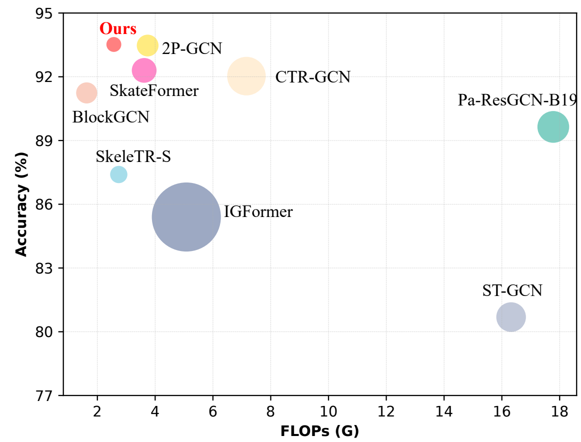

Figure 1: Teaser figure from Tab. 2 : Accuracy vs. FLOPs on the X-Sub120. The size of circles reflects the model parameters. Notably, our proposed method, highlighted in red, achieves SOTA results with 0.54M and 2.57G FLOPs.

representations or pseudo-image formats for action recognition [Zhang *et al.*, 2012; Du *et al.*, 2015]. However, these methods overlook inherent correlations among joints, which are essential for accurately understanding the motion of human skeletons. To address this issue, GCNs have become a dominant paradigm for skeleton-based action recognition by explicitly modeling the graph structure of human joints [Yan *et al.*, 2018; Li *et al.*, 2019; Li *et al.*, 2023]. Despite their effectiveness, most existing GCN-based methods still process two-person interactions by treating the two skeletons as separate or weakly coupled entities. As a result, the semantic dependencies and spatial correlations between interacting individuals are not sufficiently explored, which limits their ability to model complex interactive actions [Chen *et al.*, 2021; Zhou *et al.*, 2024; Wang *et al.*, 2025a].

On the other hand, deep neural networks on the Riemannian manifolds have achieved impressive performance across various applications [Huang and Van Gool, 2017; Wang *et al.*, 2024; Wang *et al.*, 2025b]. Among them, matrix manifolds, such as the SPD and Grassmannian manifolds, provide a convenient trade-off between structural richness and computational tractability [Cruceru *et al.*, 2021]. Given the effective-

*Corresponding author.

Metric	AIM	LEM	LCM
$g_{\boldsymbol{P}}(\boldsymbol{V},\boldsymbol{W})$	$\langle \boldsymbol{P}^{-1}\boldsymbol{V}, \boldsymbol{W}\boldsymbol{P}^{-1}\rangle$	$\langle D_{\boldsymbol{P}}\mathrm{mlog}(\boldsymbol{V}), D_{\boldsymbol{P}}\mathrm{mlog}(\boldsymbol{W})\rangle$	$\sum_{i>j}\boldsymbol{V}_{ij}\boldsymbol{W}_{ij}+\sum_{j=1}^{n}\boldsymbol{V}_{jj}\boldsymbol{W}_{jj}\boldsymbol{L}_{jj}^{-2}$
$\mathrm{dist}(\boldsymbol{P},\boldsymbol{Q})$	$\lVert\mathrm{mlog}(\boldsymbol{Q}^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{Q}^{-\frac{1}{2}})\rVert$	$\lVert\mathrm{mlog}(\boldsymbol{P})-\mathrm{mlog}(\boldsymbol{Q})\rVert$	$\left(\lVert\lfloor\boldsymbol{L}\rfloor-\lfloor\boldsymbol{K}\rfloor\rVert^2+\lVert\mathrm{mlog}(\mathbb{L})-\mathrm{mlog}(\mathbb{K})\rVert^2\right)^{\frac{1}{2}}$
$\mathrm{Log}_{\boldsymbol{P}}(\boldsymbol{Q})$	$\boldsymbol{P}^{\frac{1}{2}}\mathrm{mlog}(\boldsymbol{P}^{-\frac{1}{2}}\boldsymbol{Q}\boldsymbol{P}^{-\frac{1}{2}})\boldsymbol{P}^{\frac{1}{2}}$	$(D_{\boldsymbol{P}}\mathrm{mlog})^{-1}(\mathrm{mlog}(\boldsymbol{Q})-\mathrm{mlog}(\boldsymbol{P}))$	$D_{\boldsymbol{L}}(\mathrm{Chol}^{-1})\left(\lfloor\boldsymbol{K}\rfloor-\lfloor\boldsymbol{L}\rfloor+\mathbb{L}\mathrm{mlog}\left(\mathbb{L}^{-1}\mathbb{K}\right)\right)$
$\mathrm{Exp}_{\boldsymbol{P}}(\boldsymbol{V})$	$\boldsymbol{P}^{\frac{1}{2}}\mathrm{mexp}(\boldsymbol{P}^{-\frac{1}{2}}\boldsymbol{V}\boldsymbol{P}^{-\frac{1}{2}})\boldsymbol{P}^{\frac{1}{2}}$	$\mathrm{mexp}(\mathrm{mlog}(\boldsymbol{P})+D_{\boldsymbol{P}}\mathrm{mlog}(\boldsymbol{V}))$	$\mathrm{Chol}^{-1}\left(\lfloor\boldsymbol{Z}\rfloor+\lfloor\boldsymbol{L}\rfloor+\mathbb{L}\mathrm{mexp}\left(\mathbb{Z}\mathbb{L}^{-1}\right)\right)$
$\Gamma_{\boldsymbol{P}\to\boldsymbol{Q}}(\boldsymbol{V})$	$(\boldsymbol{Q}\boldsymbol{P}^{-1})^{\frac{1}{2}}\boldsymbol{V}(\boldsymbol{P}^{-1}\boldsymbol{Q})^{\frac{1}{2}}$	$(D_{\boldsymbol{Q}}\mathrm{mlog})^{-1}(D_{\boldsymbol{P}}\mathrm{mlog}(\boldsymbol{V}))$	$D_{\boldsymbol{K}}(\mathrm{Chol}^{-1})\left(\lfloor\boldsymbol{V}\rfloor+\mathbb{K}\mathbb{L}^{-1}\mathbb{V}\right)$

Table 1: Riemannian operators under the AIM, LEM, and LCM on the SPD manifolds. Here, $\lVert\cdot\rVert$ denotes the Frobenius norm. $\mathrm{mlog}(\cdot)$ and $\mathrm{mexp}(\cdot)$ denote the matrix logarithm and exponential, with differentials at $\boldsymbol{P}$ written as $D_{\boldsymbol{P}}(\mathrm{mlog}(\cdot))$ and $D_{\boldsymbol{P}}(\mathrm{mexp}(\cdot))$. $\mathrm{Chol}(\cdot)$ denotes the Cholesky decomposition and $\mathrm{Chol}^{-1}(\cdot)$ is the inverse operation. We set $\boldsymbol{L}=\mathrm{Chol}(\boldsymbol{P})$, $\boldsymbol{K}=\mathrm{Chol}(\boldsymbol{Q})$, and $\boldsymbol{Z}=\mathrm{Chol}(\boldsymbol{V})$. $\mathbb{V}$, $\mathbb{Z}$, $\mathbb{L}$, and $\mathbb{K}$ are diagonal matrices whose diagonals correspond to those of $\boldsymbol{V}$, $\boldsymbol{Z}$, $\boldsymbol{L}$, and $\boldsymbol{K}$, respectively.

ness of high-order statistics in capturing the geometric correlation between skeleton points, researchers have proposed several variants of covariance evolution [Hussein *et al.*, 2013; Zhang *et al.*, 2020; Nguyen, 2021]. Despite the success of these second-order statistical learning methods, they often disrupt the spatial topology of the skeleton, as covariance modeling typically aggregates multiple spatial joints into a holistic SPD representation. As a consequence, the dynamic inter-joint dependencies cannot be explicitly characterized.

To address these challenges, we propose a new framework for skeleton-based two-person interaction recognition. Specifically, the motion trajectories of each joint are extracted using a sliding time window, followed by the Gaussian embedding module for manifold modeling and geometric relationship characterization. The resulting SPD matrices contain both first-order and second-order statistics, enabling the network to be capable of characterizing complex motion patterns. These SPD data points are then used as node features in a two-person graph [Li *et al.*, 2023], which explicitly models both intra- and inter-individual dependencies via the adjacency matrix. Since the features lie in the SPD manifold, Euclidean GCNs can not be used directly. To this end, we extend the conventional GCN paradigm to the context of the SPD manifolds by designing a manifold-aware building module (SPD-GC), including neighborhood aggregation, feature transformation, and non-linear activation. The SPD-GC module manifested on three popular Riemannian metrics: Affine-Invariant Metric (AIM) [Pennec *et al.*, 2006], Log-Euclidean Metric (LEM) [Arsigny *et al.*, 2005], and Log-Cholesky Metric (LCM) [Lin, 2019]. Finally, we adopt a Riemannian pooling module to further extract global interaction features by computing the covariance of SPD matrices. As shown in Fig. 1, our approach achieves improved performance in two-person interactive skeleton action recognition.

In summary, our main contributions are given below:

- **Geometry-aware skeleton graph learning**: An SPD-GC building block is designed under three metric geometries, extending Euclidean GC mechanism to the context of SPD manifolds.
- **Information-aware high-order pooling**: A Riemannian pooling module is proposed to capture global dependencies in two-person interactions via higher-order covariance modeling.
- **Interaction-aware manifold learning**: Building on the proposed SPD-GC, a novel SPD manifold network is established, which geometrically encodes and learns spatial-temporal interaction features of two-person.
- **Experimental validity**: Extensive empirical evaluations on three benchmarking datasets for two-person interaction recognition demonstrate the effectiveness of our proposed RGCN.

2 Preliminary

This section briefly reviews the geometry of SPD manifolds. For in-depth discussions, please refer to [Do Carmo and Flaherty Francis, 1992].

Let Sym_n^+ and Sym_n be the spaces of $n\times n$ SPD matrices and $n\times n$ real symmetric matrices, respectively. As shown by [Pennec *et al.*, 2006], Sym_n^+ forms an open submanifold of Sym_n, referred to the SPD manifolds. There are three popular metrics defined on the SPD manifolds: AIM, LEM, and LCM. The Riemannian operator expressions common to all three metrics are detailed below.

Given a point $\boldsymbol{P}\in Sym_n^+$, let $\mathcal{T}_{\boldsymbol{P}}Sym_n^+$ denote its the tangent space. The Riemannian metric $g_{\boldsymbol{P}}(\cdot,\cdot)$ is a continuous family of inner products $\langle\cdot,\cdot\rangle$ defined on $\mathcal{T}_{\boldsymbol{P}}Sym_n^+$. The geodesic generalizes the straight line in Euclidean space. The geodesic distance $\mathrm{dist}(\cdot,\cdot)$ defines the minimum distance between two points on a manifold. Let $\boldsymbol{V},\boldsymbol{W}\in\mathcal{T}_{\boldsymbol{P}}Sym_n^+$ denote two tangent vectors. We define the exponential map at $\boldsymbol{P}$ as the function that maps $\boldsymbol{V}$ to Sym_n^+, $\mathrm{Exp}_{\boldsymbol{P}}(\boldsymbol{V}):\mathcal{T}_{\boldsymbol{P}}Sym_n^+\to Sym_n^+$. Its inverse is the logarithmic map, $\mathrm{Log}_{\boldsymbol{P}}(\cdot):Sym_n^+\to\mathcal{T}_{\boldsymbol{P}}Sym_n^+$. Additionally, the Parallel Transport (PT) $\Gamma_{\boldsymbol{P}\to\boldsymbol{Q}}(\boldsymbol{V})$ signifies the tangent vector $\boldsymbol{V}$ transported along the geodesic connecting $\boldsymbol{P}$ and $\boldsymbol{Q}$.

For the SPD manifolds, weighted Fréchet Mean (wFM) [Berger, 2003] is a crucial operator that computes the centre of several manifold-valued points. Given $\{\boldsymbol{P}_1,...,\boldsymbol{P}_N\}\in Sym_n^+$, the wFM is defined as:

$$\mathrm{wFM}(\{\boldsymbol{P}_i\}_{i=1}^N,\{w_i\}_{i=1}^N)=\underset{\boldsymbol{S}\in Sym_n^+}{\mathrm{argmin}}\sum_{i=1}^{N}w_i\mathrm{dist}\left(\boldsymbol{P}_i,\boldsymbol{S}\right)^2, \quad (1)$$

where the weights $\{w_1,w_2,...,w_N\}$ respect the convexity constraint, *i.e.*, $\forall i, w_i>0$ and $\sum_{i=1}^N w_i=1$, $\mathrm{dist}(\cdot,\cdot)$ denotes the geodesic distance. When $w_i=\frac{1}{N}$ for all i, Eq. (1) is reduced to the Fréchet Mean(FM), denoted by $\mathrm{FM}(\{\boldsymbol{P}_i\})$. Moreover, the AIM-based FM can be solved via the Karcher

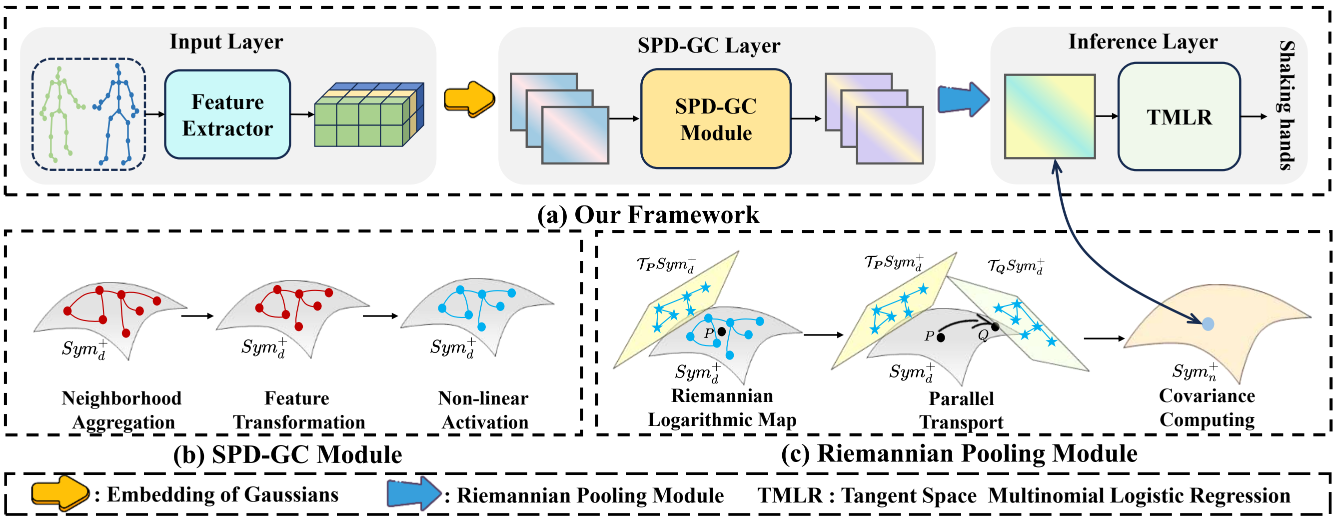

Figure 2: (a) An overall framework of our proposed RGCN, which contains an input layer, an SPD-GC layer, and an inference layer. (b) The SPD-GC module comprises three core components: neighborhood aggregation, feature transformation, and nonlinear activation. (c) The Riemannian pooling module employs Riemannian logarithmic map and Parallel Transport to project SPD matrices from point $\boldsymbol{P}$ to $\mathcal{T}_{\boldsymbol{Q}}Sym_d^+$, and finally computes SPD covariance representation.

flow algorithm [Karcher, 1977], while both the LEM- and LCM-based enjoy closed-form solutions [Chen *et al.*, 2024b]. For simplicity, we summarize the aforementioned geometric operators under the AIM, LEM, and LCM in Tab. 1.

3 Proposed Method

In this section, we introduce relevant notations and two-person graph (Sec. 3.1), then present our Gaussian embedding to map skeleton data onto the SPD manifold (Sec. 3.2). We next describe the SPD-GCN design (Sec. 3.3) and Riemannian pooling (Sec. 3.4), followed by the classifier (Sec. 3.5). The overall RGCN framework is shown in Fig. 2.

3.1 Notations

A human skeleton is represented by a graph with N joints as vertices and several bones as edges, signified as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$. Wherein, $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the set of N vertices representing joints, $\mathcal{E}$ represents the set of edges, and $\mathcal{X} \in \mathbb{R}^{C\times T\times N}$ is the C-channel feature map of the skeleton sequence across T frames. The relational adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ signifies all edges, with each element a_{ij} reflecting the correlation strength between joints v_i and v_j. The neighborhood of v_i is defined as $\mathcal{N}(v_i) = \{v_j | a_{ij} \neq 0\}$.

For skeleton sequences involving two-person interactions, we denote the data as $\mathcal{X} \in \mathbb{R}^{C\times T\times N\times M}$, where $M = 2$ represents the number of individuals. While most previous methods typically treat each individual independently and construct separate graphs, we adopt the two-person graph strategy [Li *et al.*, 2023], which models both individuals jointly. Under this representation, the input is reformulated as $\mathcal{X} \in \mathbb{R}^{C\times T\times 2N}$, allowing a single adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{2N\times 2N}$ to capture both inter- and intra-body connections.

3.2 Gaussian Embedding

The existing skeleton action recognition methods generally use stacked local temporal convolutions to model inter-frame dependencies [Chen *et al.*, 2021; Zhou *et al.*, 2024]. However, the convolution operation is essentially a local perception mechanism that mainly captures first-order statistics and is limited in modeling long-range temporal dependencies. To this end, we utilize Gaussian embeddings to capture both the first- and second-order statistics of joint trajectories, leading to more informative and geometry-aware action representations. Specifically, we first deploy a feature extractor to extract basic spatiotemporal features from skeleton sequences, denoted as $\boldsymbol{X} \in \mathbb{R}^{C'\times T\times 2N}$. Then, each sequence is divided into $L = \lceil T'/T_w \rceil$ subparts using a non-overlapping sliding window of size T_w along the temporal dimension, followed by reshaping the resulting tensor data into $\tilde{\boldsymbol{X}} \in \mathbb{R}^{L\times 2N\times C'\times T_w}$. Subsequently, the l-th mean vector $\boldsymbol{\mu}_l$ and covariance matrix $\boldsymbol{\Sigma}_l$ are computed by:

$$\boldsymbol{\mu}_l = \frac{1}{T_w}\sum_{i=1}^{T_w} \tilde{\boldsymbol{X}}_{li}, \tag{2}$$

$$\boldsymbol{\Sigma}_l = \frac{1}{T_w - 1}\sum_{i=1}^{T_w} (\tilde{\boldsymbol{X}}_{li} - \boldsymbol{\mu}_l)(\tilde{\boldsymbol{X}}_{li} - \boldsymbol{\mu}_l)^\top, \tag{3}$$

where $\tilde{\boldsymbol{X}}_{li}$ represents the i-th frame of the l-th subpart of $\tilde{\boldsymbol{X}}$. As demonstrated in [Nguyen, 2021], the single Gaussian model can be uniquely identified as an SPD matrix, shown below:

$$\boldsymbol{Y}_l = (\det \boldsymbol{\Sigma}_l)^{-\frac{1}{C'+k}} \begin{bmatrix} \boldsymbol{\Sigma}_l + k\boldsymbol{\mu}_l\boldsymbol{\mu}_l^\top & \boldsymbol{\mu}_l(k) \\ \boldsymbol{\mu}_l^\top(k) & \boldsymbol{I}_k \end{bmatrix}, \tag{4}$$

where $\boldsymbol{I}_k$ is a $k \times k$ identity matrix, $\boldsymbol{\mu}_l(k)$ is a matrix with k identical column vectors $\boldsymbol{\mu}$.

3.3 SPD-GC Module

In this subsection, we detail the primary components of the proposed graph convolution module on the SPD manifolds.

Relational adjacency matrix. A purely physical adjacency matrix considers only the fixed pairwise connections between joints in the human skeleton, representing a static and rigid topology. However, as demonstrated in several studies [Chen *et al.*, 2021; Zhou *et al.*, 2024], such a design fails to model correlations among physically non-connected joints, thus limiting the expressiveness of GCNs. To address this issue, we extract intra-body topological relations from individual entities, as well as the inter-body topological relations that connect joints of two different bodies, thereby modeling dynamic and non-shared dependencies. By employing a two-person graph structure, these relations can be computed concurrently as follows:

$$\boldsymbol{R}[t,i,j] = \exp\left(-\frac{|\mathrm{dist}(\tilde{\boldsymbol{X}}[t,i], \tilde{\boldsymbol{X}}[t,j])|}{C'}\right), \quad (5)$$

where $\boldsymbol{R}[t,i,j]$ denotes the correlation between the i-th and j-th joints in $\tilde{\boldsymbol{X}}$ at the t-th frame. The relational adjacency matrix $\tilde{\boldsymbol{A}} \in \mathbb{R}^{L\times 2N\times 2N}$ is constructed as:

$$\tilde{\boldsymbol{A}} = \mathcal{R}(\boldsymbol{A}, \boldsymbol{R}) = \boldsymbol{A} + \alpha \cdot \boldsymbol{R}, \quad (6)$$

where α is a learnable trade-off parameter, and $\boldsymbol{A}$ represents the purely physical adjacency matrix of the body.

Neighborhood aggregation. In Euclidean GCNs, neighborhood aggregation is employed to capture information from adjacent nodes. This process can be viewed as leveraging the normalized adjacency coefficients as weights to perform a weighted average of adjacent node features, thereby updating the representation of the target node. By stacking multiple layers, GCNs enable the propagation of higher-order neighborhood information and facilitate global semantic integration, allowing the model to encode both local connectivity and long-range dependencies within the graph.

The aforementioned neighborhood aggregation mechanism can be generalized to the context of SPD manifolds. At this point, feature aggregation is naturally implemented by wFM illustrated in Eq. 1, thus preserving the Riemannian geometry of the input data manifold. Given the relational adjacency matrix $\tilde{\boldsymbol{A}}$ and a set of SPD features $\boldsymbol{Y}$ of size $L \times 2N \times C \times C$, the wFM-based node feature aggregation can be expressed as:

$$\boldsymbol{D} = \mathrm{wFM}(\boldsymbol{Y}, \tilde{\boldsymbol{A}}). \quad (7)$$

where $\boldsymbol{D}$ denotes the aggregated SPD node features.

Feature transformation. In Euclidean GCNs, feature transformation is commonly implemented by a learnable linear mapping, expressed as $\boldsymbol{XW}$. For SPD-valued node features, however, such an unconstrained Euclidean transformation may distort pairwise manifold distances and the intrinsic geometric structure. We therefore adopt metric-consistent isometric mappings, which preserve geodesic distances under the corresponding Riemannian metric and provide a geometry-preserving counterpart to Euclidean feature transformation. In Thm. 1, we detail the specific expressions of isometric mapping based on the AIM, LEM, and LCM.

Theorem 1 (Isometries). *Let $\boldsymbol{D} \in Sym_n^+$ and $\boldsymbol{O} \in O(n)$ be an SPD matrix and an orthogonal matrix, respectively. The following transformations are Riemannian isometries under different Riemannian metrics:*

$$\textit{AIM \& LEM: } \boldsymbol{B} = \boldsymbol{ODO}^\top, \quad (8)$$

$$\textit{LCM: } \boldsymbol{B} = \mathrm{Chol}^{-1}\left(\lfloor L(\boldsymbol{P})\rfloor + \mathrm{mexp}\left(\mathbb{D}\left(L(\boldsymbol{P})\right)\right)\right), \quad (9)$$

where

$$L(\boldsymbol{P}) = \boldsymbol{O}\left(\mathrm{Sym}(\lfloor \mathrm{Chol}(\boldsymbol{P})\rfloor) + \mathrm{mlog}\left(\mathbb{D}(\mathrm{Chol}(\boldsymbol{P}))\right)\right)\boldsymbol{O}^\top. \quad (10)$$

with

$$\mathrm{Sym}(\lfloor \mathrm{Chol}(\boldsymbol{P})\rfloor) = \frac{1}{\sqrt{2}}\left(\lfloor \mathrm{Chol}(\boldsymbol{P})\rfloor + \lfloor \mathrm{Chol}(\boldsymbol{P})\rfloor^\top\right). \quad (11)$$

Proof. The proof is presented in App. Sec. G. □

Nonlinear activation. In deep neural networks, various nonlinear activation functions are designed to improve discriminative capabilities of models [Nair and Hinton, 2010; Chen *et al.*, 2021]. Substantial empirical evidences indicate that incorporating nonlinear activation layers significantly intensifies the performance of Riemannian neural networks [Huang and Van Gool, 2017; Chami *et al.*, 2019]. Most existing SPD neural networks perform nonlinear activation using the ReEig function [Huang and Van Gool, 2017], defined as:

$$\boldsymbol{P} = \mathrm{ReEig}(\boldsymbol{B}) = \boldsymbol{U}\max(\epsilon \boldsymbol{I}, \boldsymbol{\Sigma})\boldsymbol{U}^\top, \quad (12)$$

where $\boldsymbol{B} = \boldsymbol{U\Sigma U}^\top$ represents the Singular Value Decomposition (SVD), and ϵ is a small rectification threshold. Although the ReEig function introduces nonlinearity by regulating small eigenvalues to maintain the SPD properties, it inevitably discards potentially informative low-energy components. In contrast, the matrix power operation imposes a smooth, element-wise nonlinearity on the SPD matrix by applying a power function to its eigenvalues. This allows for flexible adjustment of the spectral distribution, preserves all eigenvalue information. The nonlinear activation function ϕ associated with the matrix power is formulated as:

$$\boldsymbol{P} = \phi(\boldsymbol{B}) = \boldsymbol{U}\boldsymbol{\Sigma}^{\theta_1}\boldsymbol{U}^\top, \quad (13)$$

where θ_1 ($\theta_1 < 1$ in this paper) is a positive real number. Since Eq. (13) can deform the latent metric geometries of the SPD manifold (reduce the matrix's condition number and smooth the curvature), it is qualified to generate more discriminative representations [Chen *et al.*, 2024a].

3.4 Riemannian Pooling

Traditional two-person interaction methods typically utilize mean pooling operation to realize skeleton-level feature fusion. However, it is not recommended to apply Euclidean pooling computations to the manifold-valued features produced by the SPD-GC module, as they cannot faithfully preserve the geometric relationships among skeleton joints inherent in two-person interactions. To address this limitation, we propose an interaction modeling method that computes the 'covariance of covariances'. Whereby, the covariance features of all joints within a skeleton are aggregated via their covariance matrix, resulting in a high-order representation. This strategy enables the capture of global dependencies across two-person interactions while preserving the intrinsic Riemannian geometry of the SPD data points. In other words, it can be regarded as a manifold-valued pooling mechanism. Specifically, let $\boldsymbol{P}^m = \mathrm{FM}(\{\boldsymbol{P}_i\}_{i=1}^L)$ signifies the Fréchet mean of a set of SPD matrices. Then, the empirical covariance matrix is computed with the aid of Parallel Transport:

$$\boldsymbol{\mathcal{P}} = \frac{1}{L-1}\sum_{i=1}^{L} f_v\left(\Gamma_{\boldsymbol{P}^m\to\boldsymbol{W}_{pt}}\left(\mathrm{Log}_{\boldsymbol{P}^m}\left(\boldsymbol{P}_i\right)\right)\right) f_v\left(\Gamma_{\boldsymbol{P}^m\to\boldsymbol{W}_{pt}}\left(\mathrm{Log}_{\boldsymbol{P}^m}\left(\boldsymbol{P}_i\right)\right)\right)^\top, \tag{14}$$

where $\Gamma_{\boldsymbol{P}^m\to\boldsymbol{W}_{pt}}(\cdot)$ represents the Parallel Transport along the geodesic connecting $\boldsymbol{P}^m$ and $\boldsymbol{W}_{pt}$, and $\boldsymbol{W}_{pt}$ is a learnable SPD matrix. The map $f_v(\cdot)$ vectorizes the symmetric matrix by taking its lower triangular part and applying $\sqrt{2}$ coefficients on its off-diagonal terms to maintain the norm [Nguyen, 2021]. To further obtain a compact and discriminative representation while preserving the intrinsic SPD geometry, we adopt a bilinear mapping on $\boldsymbol{\mathcal{P}}$ [Huang and Van Gool, 2017], defined as:

$$\boldsymbol{E} = \boldsymbol{W}_{dr}^\top \boldsymbol{\mathcal{P}} \boldsymbol{W}_{dr}, \tag{15}$$

where $\boldsymbol{W}_{dr} \in \mathbb{R}^{n\times d}$ $(d < n)$ is a column full-rank weight matrix with semi-orthogonality. This operation performs intrinsic dimensionality reduction on the SPD manifolds, effectively suppressing redundant correlation patterns and enhancing discriminatory power.

3.5 Classifier

It can be seen that the features produced by the Riemannian pooling module remain situated on the SPD manifold, preventing the direct application of traditional Euclidean classifiers. As an effective countermeasure, many existing SPD-based neural networks [Huang *et al.*, 2015; Nguyen *et al.*, 2019; Nguyen, 2021] utilize the Riemannian logarithmic map to project SPD data points onto the tangent space at the identity matrix, $\mathcal{T}_{\boldsymbol{I}}Sym_n^+$. Since tangent space conforms to the Euclidean geometry, such an embedding mapping enables the use of conventional Euclidean computations, *e.g.*, Multinomial Logistic Regression (referred to as Log-TMLR), for the subsequent feature analysis. The computation process of Log-TMLR is illustrated in Fig. 3, where $\mathrm{Log}_{\boldsymbol{I}}(\cdot)$ is equivalent to $\mathrm{mlog}(\cdot)$ at the identity matrix, defined as:

$$\boldsymbol{C}_{out} = \mathrm{Softmax}\left(\mathrm{FC}\left(f_{vec}\left(\mathrm{mlog}(\boldsymbol{E})\right)\right)\right). \tag{16}$$

However, Log-TMLR has an inherent limitation, *i.e.*, the matrix logarithm will alter the relative ordering of eigenvalues, which may diminish the discrimination of the resulting features [Li *et al.*, 2017]. Conversely, the matrix power effectively preserves the eigenvalue ordering. For a detailed analysis, please refer to App. Sec. F. As shown in [Chen *et al.*, 2025] [Tab. 2], the matrix logarithm can be expressed as $\mathrm{Log}_I(\boldsymbol{E}) = \frac{1}{\theta_2}(\boldsymbol{E}^{\theta_2} - \boldsymbol{I})$. Substituting this expression into Eq. (16) yields the resulting method, named Pow-TMLR:

$$\boldsymbol{C}_{out} = \mathrm{Softmax}\left(\mathrm{FC}\left(f_{vec}\left(\frac{1}{\theta_2}(\boldsymbol{E}^{\theta_2} - \boldsymbol{I})\right)\right)\right), \tag{17}$$

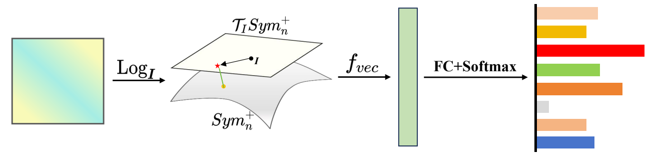

Figure 3: Illustration of Log-TMLR, with $f_{vec}(\cdot)$ vectorizes the lower triangular part of a matrix.

where $\boldsymbol{E}^{\theta_2} = \boldsymbol{U}\boldsymbol{\Sigma}^{\theta_2}\boldsymbol{U}^\top$ represents the matrix power and θ_2 ($\theta_2 < 1$ in this paper) is a positive real number.

For the optimization of SPD parameters and the RGCN algorithm flowchart, please refer to App. Sec. A and Sec. B.

4 Experiments

In this section, we conduct extensive experiments to assess the proposed RGCN on three benchmarks: SBU [Yun *et al.*, 2012], NTU-Interaction, and NTU120-Interaction. Note that NTU-Interaction and NTU120-Interaction are interaction subsets derived from NTU RGB+D [Shahroudy *et al.*, 2016] and NTU RGB+D 120 [Liu *et al.*, 2019], respectively. NTU-Interaction is evaluated under the standard cross-subject (X-Sub) and cross-view (X-View) protocols, while NTU120-Interaction follows the cross-subject (X-Sub120) and cross-setup (X-Set120) protocols. Further details about the datasets are included in App. Sec. C.

4.1 Implementation Details

Architecture. Within the RGCN framework, we consider three metric-consistent variants, namely RGCN-AIM, RGCN-LEM, and RGCN-LCM. They differ only in the Riemannian metric adopted, which is applied consistently to both the SPD-GC layer and the Riemannian pooling module. In addition, we propose a hybrid variant, RGCN-MIX, which decouples the metric choices by using LEM for SPD-GC and AIM for Riemannian pooling.

Training. For training, we adopt cross-entropy loss as the objective and employ the SGD optimizer with Nesterov momentum of 0.9, an initial learning rate of 0.1, and a weight decay of 0.0002. The model is trained for 65 epochs, with a warm-up phase during the first 5 epochs. The batch size for training and testing is set to 16. To capture local temporal patterns, a temporal windowing strategy is employed, where each sequence is split into overlapping sub-sequences of 15 frames with a stride of 10. As the feature extractor, we adopt 2P-GCN [Li *et al.*, 2023] as the base model but retain only its feature extraction components and a single output from the main branch. The hyperparameter choices for k, θ_1 and θ_2 are discussed in the experiments.

4.2 Results and Analysis

We compare the proposed RGCN with SOTA methods on the NTU-Interaction and NTU120-Interaction datasets. As shown in Tab. 2, RGCN achieves SOTA performance among manifold-based approaches and remains highly competitive with the strongest Euclidean baselines. Specifically, RGCN-MIX outperforms the relevant manifold deep learning method

Method	Conf./Jour.	NTU-Interaction X-Sub	NTU-Interaction X-View	NTU120-Interaction X-Sub120	NTU120-Interaction X-Set120	FLOPs (G)	#Param. (M)
ST-LSTM [Liu *et al.*, 2016]	ECCV 2016	83.00	87.30	63.00	66.60	N/A	N/A
SPDNet [Huang and Van Gool, 2017]*	AAAI 2017	74.85	76.07	60.72	62.08	N/A	N/A
ST-GCN [Yan *et al.*, 2018] *	AAAI 2018	89.31	93.72	80.69	80.27	16.32	3.10
2S-GCA-LSTM [Liu *et al.*, 2018]	TIP 2018	87.20	89.90	73.00	73.30	N/A	N/A
2S-AGCN [Shi *et al.*, 2019]*	CVPR 2019	93.36	96.67	87.83	89.21	37.32	6.94
SPDNetBN [Brooks *et al.*, 2019]*	NeurIPS 2019	75.24	76.31	61.11	62.36	N/A	N/A
Pa-ResGCN-B19 [Song *et al.*, 2020] *	ACM MM 2020	94.34	97.55	89.64	89.94	17.78	3.64
LSTM-IRN [Perez *et al.*, 2021]	TMM 2021	90.50	93.50	77.70	79.60	N/A	N/A
GeomNet [Nguyen, 2021]	ICCV 2021	93.62	96.32	86.49	87.58	N/A	N/A
CTR-GCN [Chen *et al.*, 2021] *	ICCV 2021	95.31	97.60	92.03	92.28	7.16	5.68
IGFormer [Pang *et al.*, 2022]	ECCV 2022	93.60	96.50	85.40	86.50	5.08	20.29
SkeleTR-S [Duan *et al.*, 2023]	ICCV 2023	94.60	97.10	87.40	88.00	2.74	0.81
SkeleTR-L [Duan *et al.*, 2023]	ICCV 2023	94.70	97.50	87.80	88.30	N/A	N/A
SkeleTR[C] [Duan *et al.*, 2023]	ICCV 2023	94.80	97.70	87.80	88.30	N/A	N/A
BlockGCN [Zhou *et al.*, 2024] ◇	CVPR 2024	95.57	98.12	91.24	92.54	1.63	1.36
SkateFormer [Do and Kim, 2024]	ECCV 2024	**97.10**	**99.30**	92.30	93.20	3.62	2.02
ProtoGCN [Liu *et al.*, 2025]◇	CVPR 2025	96.51	98.34	92.15	92.61	7.67	4.13
RGCN-AIM	N/A	**96.90**	**98.84**	**93.34**	93.27	2.57	0.54
RGCN-LEM	N/A	96.80	98.81	**93.41**	**93.47**	2.57	0.54
RGCN-LCM	N/A	96.83	**98.84**	93.30	**93.38**	2.57	0.54
RGCN-MIX	N/A	**96.99**	**98.96**	**93.52**	**93.41**	2.57	0.54

Table 2: Accuracy (%) comparison with SOTA methods on the NTU-Interaction and NTU120-Interaction datasets. The best three results are highlighted in **red**, **blue**, **cyan**. * indicates that the result has been reported in [Li *et al.*, 2023] or [Nguyen, 2021]. ◇ indicates that we produce the results based on the code published by the authors.

Method	Conf./Jour.	SBU
Co-LSTM [Zhu *et al.*, 2016]	AAAI 2016	90.40
ST-LSTM [Liu *et al.*, 2016]	ECCV 2016	93.30
SPDNet [Huang and Van Gool, 2017]*	AAAI 2017	92.38
2s-GCA-LSTM [Liu *et al.*, 2018]	TIP 2018	94.90
HCN [Li *et al.*, 2018]	IJCAI 2018	**98.60**
SPDNetBN [Brooks *et al.*, 2019]*	NeurIPS 2019	86.78
LSTM-IRN [Perez *et al.*, 2021]	TMM 2021	98.20
GeomNet [Nguyen, 2021]	ICCV 2021	96.33
IGFormer [Pang *et al.*, 2022]	ECCV 2022	**98.40**
ISTA-Net [Wen *et al.*, 2023]	IROS 2023	**98.51**
pruned GCN [Sahbi, 2024]	ICML 2024	**98.40**
RGCN-AIM	N/A	98.29
RGCN-LEM	N/A	98.29
RGCN-LCM	N/A	98.00
RGCN-MIX	N/A	**98.60**

Table 3: Accuracy (%) comparison with SOTA methods on the SBU dataset.

GeomNet by **3.38%**, **2.64%**, **7.03%**, and **5.83%** under the X-Sub, X-View, X-Sub120, and X-Set120 settings, respectively. Compared with SkateFormer, RGCN-MIX is slightly worse on NTU-Interaction but performs better on the more challenging NTU120-Interaction, improving by **0.32%** and **0.21%** on the X-Sub120 and X-Set120, while reducing parameters by **73.27%** and FLOPs by **29.01%**. Among the proposed variants, RGCN-MIX consistently delivers the best or near-best overall performance, while the metric-specific variants RGCN-AIM, RGCN-LEM, and RGCN-LCM remain highly competitive and outperform most existing methods across different evaluation protocols. Notably, RGCN-LEM achieves the highest accuracy of 93.47% on X-Set120, indicating that a metric tailored to the pooling stage can be particularly beneficial in this setting. Overall, these results demonstrate that RGCN effectively exploits both first- and second-order statistics to capture geometric dependencies among skeleton joints. They further suggest that decoupling the metrics used in the SPD-GC and Riemannian pooling modules provides additional flexibility to better adapt to the intrinsic geometry of SPD representations, which is consistent with the observations in [Nguyen *et al.*, 2024].

We further evaluate RGCN on the SBU dataset, which contains 282 samples. As shown in Tab. 3, all four variants achieve accuracy above 98.00%, and RGCN-MIX attains 98.60%. Despite the limited training data, RGCN remains highly competitive with prior methods, demonstrating robust performance in low-data regimes.

4.3 Ablation Study

Ablation of individual component. To quantify the contribution of each component, we conduct ablation experiments by removing either the SPD-GC or the Riemannian pooling. As shown in Tab. 4, omitting either component consistently degrades accuracy of the model on both NTU-Interaction and NTU120-Interaction. Removing SPD-GC reduces accuracy by 0.92% and 1.07% on X-Sub and X-View, respectively, and by 0.92% and 0.79% on X-Sub120 and X-Set120. This confirms the importance of SPD-GC in capturing the intrinsic geometric dependencies of skeletal joint trajectories. Moreover, removing Riemannian pooling leads to larger drops of 1.16%, 1.45%, 1.51%, and 1.36% across the four benchmarks, indicating that it strengthens discriminative interaction representations by capturing geometry-level semantic relationships

Method	SPD-GC	Pooling	NTU-Interaction X-Sub	NTU-Interaction X-View	NTU120-Interaction X-Sub120	NTU120-Interaction X-Set120
RGCN-AIM	✗	✓	95.94	98.09	92.24	92.11
	✓	✗	95.24	98.01	92.31	92.28
	✓	✓	**96.90**	**98.84**	**93.34**	**93.27**
RGCN-LEM	✗	✓	95.90	97.83	91.83	91.24
	✓	✗	96.00	98.09	91.51	91.58
	✓	✓	**96.80**	**98.81**	**93.41**	**93.47**
RGCN-LCM	✗	✓	96.08	97.77	92.65	92.02
	✓	✗	96.00	98.11	92.46	92.75
	✓	✓	**96.83**	**98.84**	**93.30**	**93.38**
RGCN-MIX	✗	✓	95.94	98.09	92.24	92.11
	✓	✗	96.00	98.09	91.51	91.58
	✓	✓	**96.99**	**98.96**	**93.52**	**93.41**

Table 4: Accuracy (%) comparison of different RGCN components on the NTU-Interaction and NTU120-Interaction datasets.

Method	θ_2	NTU-Interaction X-Sub	NTU-Interaction X-View	NTU120-Interaction X-Sub120	NTU120-Interaction X-Set120
RGCN-AIM	w/o	96.37	98.15	93.06	92.84
	0.25	**96.90**	**98.84**	**93.20**	**93.27**
	0.5	96.40	98.53	93.11	92.95
	0.75	95.74	97.77	92.68	92.23
RGCN-LEM	w/o	95.90	98.61	92.30	92.58
	0.25	96.80	**98.81**	**93.41**	**93.19**
	0.5	**96.86**	98.70	93.22	93.18
	0.75	96.73	98.29	92.49	92.72
RGCN-LCM	w/o	95.74	98.53	92.86	92.21
	0.25	**96.83**	**98.84**	**93.30**	**93.38**
	0.5	96.43	98.53	93.04	92.99
	0.75	96.37	98.47	92.90	92.81
RGCN-MIX	w/o	96.47	98.61	92.44	92.84
	0.25	**96.99**	**98.96**	**93.52**	93.27
	0.5	96.54	98.79	93.09	**93.41**
	0.75	96.60	98.66	92.61	93.03

Table 5: Accuracy (%) comparison of different RGCN variants under different settings of θ_2 on the NTU-Interaction and NTU120-Interaction datasets. w/o means using Log-TMLR (see Eq. (16)).

between the two individuals.

Ablation of classifier. We evaluate the impact of the proposed classifier under different θ_2 settings on the NTU-Interaction and NTU120-Interaction. As shown in Tab. 5, Pow-TMLR consistently outperforms Log-TMLR across all values of θ_2. This is mainly attributed to their different mechanisms for regulating the eigenvalue distribution. Specifically, the matrix power preserves the relative importance of eigenvalues by compressing those greater than 1 and stretching those less than 1. In contrast, the matrix logarithm substantially alters the scale of eigenvalues and may reverse their relative order of importance, potentially diminishing the expressive power of discriminative features in practice [Li *et al.*, 2017]. We also observe that Pow-TMLR is sensitive to θ_2. All variants of RGCN consistently achieve higher accuracy when $\theta_2 = 0.25$, compared to the cases where $\theta_2 = 0.5$ and $\theta_2 = 0.75$. The observed performance difference can be attributed to the characteristic of the power transformation in Pow-TMLR. When θ_2 is large, the transformation has limited effect in moderating extremely large or small eigenvalues. In contrast, smaller values of θ_2 more effectively compress the eigenvalue spectrum, resulting in a lower condition number. This reduction improves numerical stability and suppresses noise, enhancing the representational capacity of the model.

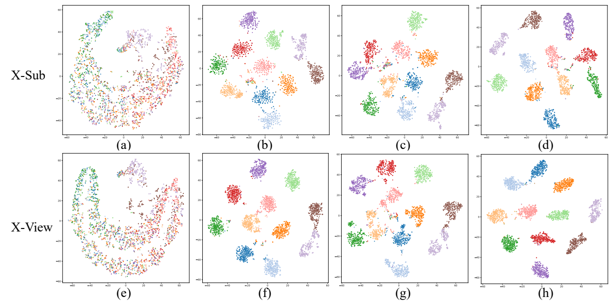

Figure 4: The t-SNE visualization of key components of RGCN-MIX on the NTU-Interaction. (a, e) Original data; (b, f) Without SPD-GC; (c, g) Without Riemannian pooling; (d, h) The full model. Noting that different colors denote different action categories.

4.4 Visualization

To intuitively verify the effectiveness of each major component in the proposed model, we conduct 2D visualization experiments using t-SNE [Maaten and Hinton, 2008], as shown in Fig. 4. In addition, to quantitatively assess the discriminability of the learned features, we adopt the Davies–Bouldin Index (DBI) [Davies and Bouldin, 2009] as the evaluation metric. Fig. 4 (a) and (e) depict the raw skeleton data without any preprocessing or representation learning, with DBI scores of 8.24 and 11.72, respectively. After removing the SPD-GC module, Fig. 4 (b) and (f) exhibit markedly improved cluster compactness, reflected by reduced DBI scores of 0.56 and 0.44. Similarly, Fig. 4 (c) and (g), which correspond to the model without the Riemannian pooling module, achieve DBI scores of 0.62 and 0.60, indicating enhanced class separability over the unprocessed inputs. As shown in Fig. 4 (d) and (h), the complete RGCN-MIX further improves intra-class compactness and inter-class separability, with DBI scores of 0.51 and 0.40 on the X-Sub and X-View. These findings collectively validate the effectiveness and necessity of each key module in our proposed framework. For other visualization results, please refer to App. Sec. E

4.5 Other Experiments

Due to space constraints, additional experiments are provided in App. Sec. D.

5 Conclusion

In this study, we propose a novel Riemannian neural network for skeleton-based two-person interaction action recognition. Leveraging isometric mapping, weighted Fréchet mean, and matrix power, we extend the Euclidean GCN into the SPD manifold. Additionally, we adopt the Riemannian pooling layer computes the covariance representation on the SPD manifold, further enhancing the ability of the network to capture deep interactive features. Extensive experiments and ablation studies demonstrate the effectiveness and flexibility of our network. For future work, we aim to extend RGCN to more complex scenarios, such as group activity recognition and human-machine interaction analysis.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62306127, 62332008), the Natural Science Foundation of Jiangsu Province (BK20231040), the Fundamental Research Funds for the Central Universities (JUSRP124015), the Postgraduate Research & Practice Innovation Program of Jiangsu Province (SJCX25_1319), and the National Key R&D Program of China (2023YFF1105102, 2023YFF1105105).

Contribution Statement

Zihao Bi contributes to the code implementation, experimental design and analysis, and the preparation of the initial manuscript draft.

References

[Arsigny *et al.*, 2005] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. *Fast and Simple Computations on Tensors with Log-Euclidean Metrics*. PhD thesis, Inria, 2005.

[Berger, 2003] Marcel Berger. *A panoramic view of Riemannian geometry*. Springer, 2003.

[Brooks *et al.*, 2019] Daniel Brooks, Olivier Schwander, Frédéric Barbaresco, Jean-Yves Schneider, and Matthieu Cord. Riemannian batch normalization for spd neural networks. In *NeurIPS*, pages 15463–15474, 2019.

[Chami *et al.*, 2019] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *NeurIPS*, pages 4869–4880, 2019.

[Chen *et al.*, 2021] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *ICCV*, pages 13339–13348, 2021.

[Chen *et al.*, 2024a] Ziheng Chen, Yue Song, Rui Wang, Xiaojun Wu, and Nicu Sebe. RMLR: Extending multinomial logistic regression into general geometries. In *NeurIPS*, pages 59716–59754, 2024.

[Chen *et al.*, 2024b] Ziheng Chen, Yue Song, Tianyang Xu, Zhiwu Huang, Xiao-Jun Wu, and Nicu Sebe. Adaptive Log-Euclidean metrics for SPD matrix learning. *IEEE Trans. Image Process.*, 33:5194–5205, 2024.

[Chen *et al.*, 2025] Ziheng Chen, Yue Song, Xiao-Jun Wu, Gaowen Liu, and Nicu Sebe. Understanding matrix function normalizations in covariance pooling through the lens of riemannian geometry. In *ICLR*, 2025.

[Cruceru *et al.*, 2021] Calin Cruceru, Gary Bécigneul, and Octavian-Eugen Ganea. Computationally tractable Riemannian manifolds for graph embeddings. In *AAAI*, pages 7133–7141, 2021.

[Davies and Bouldin, 2009] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.*, (2):224–227, 2009.

[Do and Kim, 2024] Jeonghyeok Do and Munchurl Kim. Skateformer: skeletal-temporal transformer for human action recognition. In *ECCV*, pages 401–420, 2024.

[Do Carmo and Flaherty Francis, 1992] Manfredo Perdigao Do Carmo and J Flaherty Francis. *Riemannian geometry*, volume 2. Springer, 1992.

[Du *et al.*, 2015] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, pages 1110–1118, 2015.

[Duan *et al.*, 2023] Haodong Duan, Mingze Xu, Bing Shuai, Davide Modolo, Zhuowen Tu, Joseph Tighe, and Alessandro Bergamo. Skeletr: Towards skeleton-based action recognition in the wild. In *ICCV*, pages 13634–13644, 2023.

[Huang and Van Gool, 2017] Zhiwu Huang and Luc Van Gool. A Riemannian network for SPD matrix learning. In *AAAI*, pages 2036–2042, 2017.

[Huang *et al.*, 2015] Zhiwu Huang, Ruiping Wang, S. Shan, Xianqiu Li, and Xilin Chen. Log-Euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *ICML*, pages 720–729, 2015.

[Hussein *et al.*, 2013] Mohamed E Hussein, Marwan Torki, Mohammad A Gowayyed, and Motaz El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *IJCAI*, pages 2466–2472, 2013.

[Karcher, 1977] Hermann Karcher. Riemannian center of mass and mollifier smoothing. *Commun. Pure Appl. Math.*, 30(5):509–541, 1977.

[Li *et al.*, 2017] Peihua Li, Jiangtao Xie, Qilong Wang, and Wangmeng Zuo. Is second-order information helpful for large-scale visual recognition? In *ICCV*, pages 2089–2097, 2017.

[Li *et al.*, 2018] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *IJCAI*, pages 786–792, 2018.

[Li *et al.*, 2019] Bin Li, Xi Li, Zhongfei Zhang, and Fei Wu. Spatio-temporal graph routing for skeleton-based action recognition. In *AAAI*, pages 8561–8568, 2019.

[Li *et al.*, 2023] Zhengcen Li, Yueran Li, Linlin Tang, Tong Zhang, and Jingyong Su. Two-person graph convolutional network for skeleton-based human interaction recognition. *IEEE Trans. Circuits Syst. Video Technol.*, 33(7):3333–3342, 2023.

[Lin, 2019] Zhenhua Lin. Riemannian geometry of symmetric positive definite matrices via Cholesky decomposition. *SIAM J. Matrix Anal. Appl.*, 40(4):1353–1370, 2019.

[Liu *et al.*, 2016] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal LSTM with trust gates for 3d human action recognition. In *ECCV*, pages 816–833, 2016.

[Liu *et al.*, 2018] Jun Liu, Gang Wang, Ling-Yu Duan, Kamila Abdiyeva, and Alex C Kot. Skeleton-based human action recognition with global context-aware attention LSTM networks. *IEEE Trans. Image Process.*, 27(4):1586–1599, 2018.

[Liu *et al.*, 2019] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. NTU RGB+D120: A large-scale benchmark for 3d human activity understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(10):2684–2701, 2019.

[Liu *et al.*, 2025] Hongda Liu, Yunfan Liu, Min Ren, Hao Wang, Yunlong Wang, and Zhenan Sun. Revealing key details to see differences: A novel prototypical perspective for skeleton-based action recognition. In *CVPR*, pages 29248–29257, 2025.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *J. Mach. Learn. Res.*, 9(86):2579–2605, 2008.

[Nair and Hinton, 2010] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[Nguyen *et al.*, 2019] Xuan Son Nguyen, Luc Brun, Olivier Lézoray, and Sébastien Bougleux. A neural network based on SPD manifold learning for skeleton-based hand gesture recognition. In *CVPR*, pages 12028–12037, 2019.

[Nguyen *et al.*, 2024] Xuan Son Nguyen, Shuo Yang, and Aymeric Histace. Matrix manifold neural networks++. In *ICLR*, pages 40448–40478, 2024.

[Nguyen, 2021] Xuan Son Nguyen. Geomnet: A neural network based on Riemannian geometries of SPD matrix space and Cholesky space for 3d skeleton-based interaction recognition. In *ICCV*, pages 13359–13369, 2021.

[Pang *et al.*, 2022] Yunsheng Pang, Qiuhong Ke, Hossein Rahmani, James Bailey, and Jun Liu. IGformer: Interaction graph transformer for skeleton-based human interaction recognition. In *ECCV*, pages 605–622, 2022.

[Pennec *et al.*, 2006] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A Riemannian framework for tensor computing. *Int. J. Comput. Vision*, 66(1):41–66, 2006.

[Perez *et al.*, 2021] Mauricio Perez, Jun Liu, and Alex C Kot. Interaction relational network for mutual action recognition. *IEEE Trans. Multimedia*, 24:366–376, 2021.

[Sahbi, 2024] Hichem Sahbi. Coarse-to-fine semi-structured pruning of graph convolutional networks for skeleton-based recognition. In *ICML*, 2024.

[Shahroudy *et al.*, 2016] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3d human activity analysis. In *CVPR*, pages 1010–1019, 2016.

[Shi *et al.*, 2019] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, pages 12018–12027, 2019.

[Song *et al.*, 2020] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In *ACM MM*, pages 1625–1633, 2020.

[Wang *et al.*, 2024] Rui Wang, Chen Hu, Ziheng Chen, Xiao-Jun Wu, and Xiaoning Song. A Grassmannian manifold self-attention network for signal classification. In *IJCAI*, pages 5099–5107, 2024.

[Wang *et al.*, 2025a] Rui Wang, Jiayao Jin, Ziheng Chen, Cong Wu, Xiao-Jun Wu, and Nicu Sebe. Structural topology refinement network for skeleton-based action recognition. *IEEE Trans. Instrum. Meas.*, 74:1–16, 2025.

[Wang *et al.*, 2025b] Rui Wang, Shaocheng Jin, Ziheng Chen, Xiaoqing Luo, and Xiao-Jun Wu. Learning to normalize on the spd manifold under bures-wasserstein geometry. In *CVPR*, pages 8289–8298, 2025.

[Wen *et al.*, 2023] Yuhang Wen, Zixuan Tang, Yunsheng Pang, Beichen Ding, and Mengyuan Liu. Interactive spatiotemporal token attention network for skeleton-based general interactive action recognition. In *IROS*, pages 7886–7892, 2023.

[Yan *et al.*, 2018] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, pages 7444–7452, 2018.

[Yun *et al.*, 2012] Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L Berg, and Dimitris Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *CVPRW*, pages 28–35, 2012.

[Zhang *et al.*, 2012] Yimeng Zhang, Xiaoming Liu, Ming-Ching Chang, Weina Ge, and Tsuhan Chen. Spatio-temporal phrases for activity recognition. In *ECCV*, pages 707–721, 2012.

[Zhang *et al.*, 2020] Tong Zhang, Wenming Zheng, Zhen Cui, Yuan Zong, Chaolong Li, Xiaoyan Zhou, and Jian Yang. Deep manifold-to-manifold transforming network for skeleton-based action recognition. *IEEE Trans. Multimedia*, 22(11):2926–2937, 2020.

[Zhou *et al.*, 2024] Yuxuan Zhou, Xudong Yan, Zhi-Qi Cheng, Yan Yan, Qi Dai, and Xian-Sheng Hua. Blockgcn: Redefine topology awareness for skeleton-based action recognition. In *CVPR*, pages 2049–2058, 2024.

[Zhu *et al.*, 2016] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *AAAI*, pages 3697–3704, 2016.