Leveraging Spatio-Temporal Dependency for Skeleton-Based Action Recognition

Jungho Lee[1] , Minhyeok Lee[1] , Suhwan Cho[1] , Sungmin Woo[1] , Sungjun Jang[1] , Sangyoun Lee[1,2]

[1]School of Electrical and Electronic Engineering, Yonsei University,
[2]Korea Institute of Science and Technology (KIST)
{2015142131, hydragon516, chosuhwan, smw3250, jeu2250, syleee}@yonsei.ac.kr

Abstract

Skeleton-based action recognition has attracted considerable attention due to its compact representation of the human body's skeletal sructure. Many recent methods have achieved remarkable performance using graph convolutional networks (GCNs) and convolutional neural networks (CNNs), which extract spatial and temporal features, respectively. Although spatial and temporal dependencies in the human skeleton have been explored separately, spatio-temporal dependency is rarely considered. In this paper, we propose the Spatio-Temporal Curve Network (STC-Net) to effectively leverage the spatio-temporal dependency of the human skeleton. Our proposed network consists of two novel elements: 1) The Spatio-Temporal Curve (STC) module; and 2) Dilated Kernels for Graph Convolution (DK-GC). The STC module dynamically adjusts the receptive field by identifying meaningful node connections between every adjacent frame and generating spatio-temporal curves based on the identified node connections, providing an adaptive spatio-temporal coverage. In addition, we propose DK-GC to consider long-range dependencies, which results in a large receptive field without any additional parameters by applying an extended kernel to the given adjacency matrices of the graph. Our STC-Net combines these two modules and achieves state-of-the-art performance on four skeleton-based action recognition benchmarks. Code is available at https://github.com/Jho-Yonsei/STC-Net.

1. Introduction

Action recognition is one of the most important video understanding tasks used in various applications such as virtual reality and human–computer interaction. Recent studies on action recognition are divided into two methods, RGB-based [33, 31] and skeleton-based methods [36, 26, 27, 2, 4, 16]. Action recognition using the skeleton modality receives a video sequence with the three-dimensional coordinates of major human joints as its input. Skeleton-based action recognition has the advantage of being able to create a lightweight model with low computational complexity by compactly compressing the structure of the human body. In addition, it has the benefit of robustness in that it is not affected by background noise, weather, and lighting conditions unlike RGB-based methods.

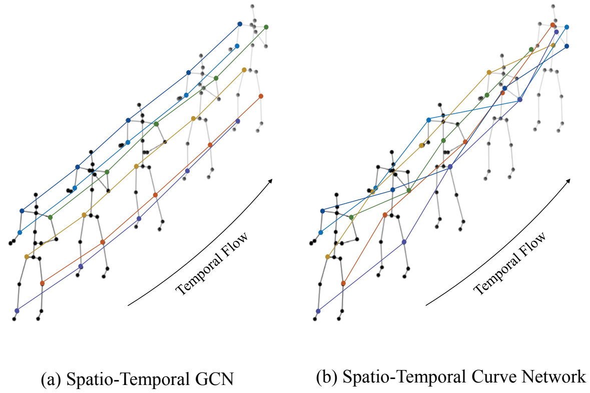


Figure 1. Comparison of temporal flows of spatio-temporal GCN (a) and STC module of our model (b). (b)'s curves have adaptive spatio-temporal receptive field by aggregating different nodes for different frames, whereas (a) treats the temporal features of each node independently.

Earlier approaches [7, 8, 38, 14, 23] extract features by dealing with every joint independently, which means that they do not consider information between structurally correlated human joints. However, the connections between human joints are identified as a graph structure after Yan *et al.* [36] has proposed spatio-temporal graph convolutional networks (GCNs) for the skeleton modality. Recent approaches [27, 4, 2, 6] adopt the GCNs as their baseline and attempt to enlarge the receptive field on the spatial domain.

However, methods based on Yan *et al.*'s GCNs have several limitations. (1) When a person performs an action, the movement of their body parts occurs in both space and time, and these two aspects of the movement are inherently interconnected. While incorporating both spatial and temporal components can provide a more complete and accurate representation of human actions, it is not feasible to directly utilize this spatio-temporal interconnectivity as the spatial and temporal modules exist independently of each other. (2) As they use graphs that include only the connectivity of physically adjacent joints, their networks with such graphs have small spatial receptive fields. Although several self-attention methodologies [27, 2] have been proposed to increase the spatial receptive field, they still rely on using physically adjacent graphs, which can lead to biased results towards those physically adjacent graphs and highlight a potential limitation in their effectiveness. To handle this problem, Liu *et al*. [21] has proposed a multi-scale graph that identifies the relationship between structurally distant nodes. However, as stated by Yan *et al*. [36], although it is crucial to differentiate human motion into concentric and eccentric patterns, Liu *et al*.'s method does not account for such patterns. Additionally, Liu *et al*.'s model suffers from the limitation of having high model complexity, as there are too many operations parallelly existing in a single layer.

To solve limitation (1), we propose a Spatio-Temporal Curve (STC) module to reflect direct spatio-temporal dependencies in a skeleton sequence. In addition to applying temporal convolution to aggregate node-wise sequential features, we construct curves that consider the sequential spatio-temporal features for every node and aggregate them with input feature map. To create the curves, we choose the most highly correlated nodes in feature space between all adjacent frames and connect them. Therefore, a more semantically effective graph structure can be adaptively generated by giving autonomy to the temporal connections between nodes. Fig. 1 compares the temporal flows of the spatio-temporal GCN for existing methods [36, 27, 2] and our proposed method. Fig. 1 (a) shows that the model reflects only the features of the same nodes in every frame, while Fig. 1 (b) shows that the model considers the spatio-temporal correlations through the generated curves that take account of features of different nodes in adjacent frames. Inspired by [35], we use an aggregation module to effectively combine all the curve features and apply them to the input feature map.

To handle limitation (2), we propose Dilated Kernels for Graph Convolution (DK-GC) to have large spatial receptive field for skeletal modality without any additional parameters. The GCNs for the human skeleton aggregate inward-facing (centripetal), identity, and outward-facing (centrifugal) features, unlike convolutional neural networks (CNNs), which aggregate left, identity, and right pixels features. To apply the dilated kernel to such GCNs, we create adjacency matrices to identify structurally distant relationships by modifying centripetal and centrifugal matrices. To incorporate spatial receptive fields from low-level to high-level, we divide the spatial module into several branch operations with different dilated windows. Meanwhile, dilated graph convolution has already been introduced by Li *et al*. [17] for 3D point clouds analysis task. However, Li *et al*.'s dilated graph convolution is completely different from what we propose and is not suitable for human skeletal modality. Firstly, this method does not utilize the given adjacency matrices, but instead uses dynamic graph via k-nearest neighborhood (k-NN) algorithm. The inability to utilize the given adjacency matrices reduces the robustness for the action recognition model as Shi *et al*. [27] experimentally has proven that not using those matrices leads to inferior performance. Moreover, the k-NN alone cannot identify all the physically adjacent nodes. The second reason is that Li *et al*.'s method requires a lot of GPU resources. It causes very high GPU memory consumption and low inference speed since the dynamic graphs are constructed by computing all pairwise distances between all the nodes for every GCN layer.

To verify the superiority of our STC-Net, extensive experiments are conducted on four skeleton-based action recognition benchmark datasets: NTU-RGB+D 60 [25], NTU-RGB+D 120 [18], Kinetics-Skeleton [12], and Northwestern-UCLA [32].

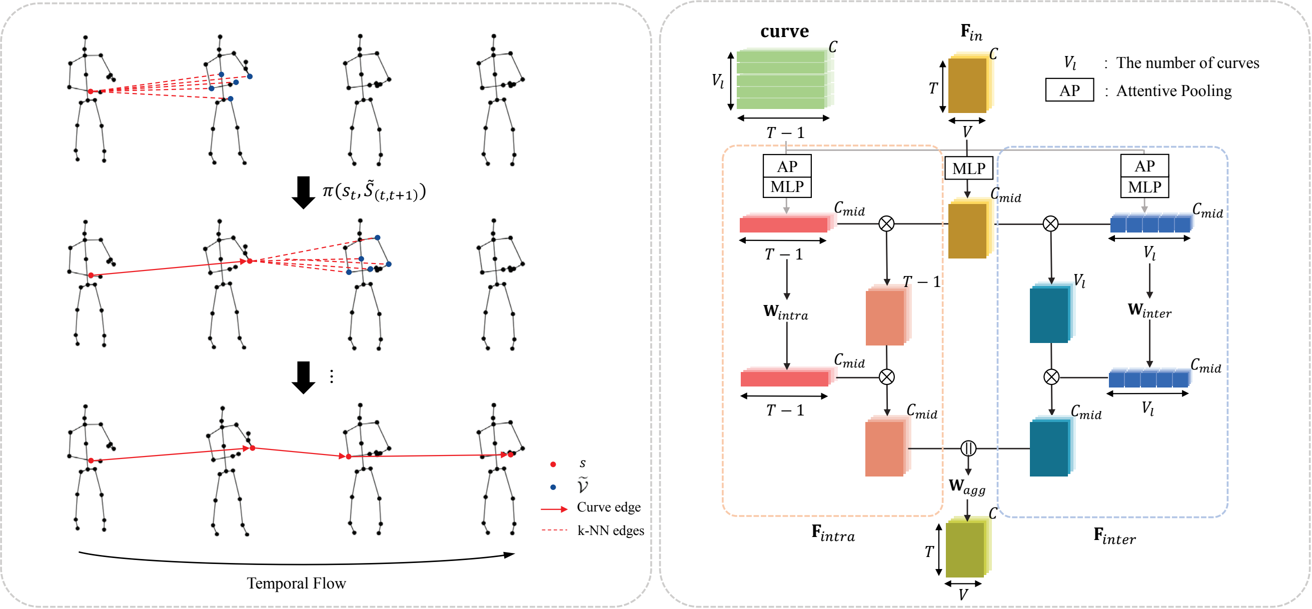
Our main contributions are summarized as follows:

- We propose the Spatio-Temporal Curve (STC) module to leverage the direct spatio-temporal correlation between different nodes of different frames.
- We propose the Dilated Kernels for Graph Convolution (DK-GC) that makes the model have a large spatial receptive field without any additional parameters by modifying the given skeletal adjacency matrices.
- Our proposed STC-Net outperforms existing state-of-the-arts methods on four benchmarks for skeleton-based action recognition.

2. Related Work

2.1. Skeleton-Based Action Recognition

Previous skeleton-based action recognition methods [7, 14, 24] do not consider the relationships between joint nodes for the human skeleton but treat all the joint nodes independently. However, Yan *et al*. [36] treats this modality as a graph structure. Most recent methods [27, 21, 28, 2] tend to rely on GCNs to deal with spatial correlation. They reflect the subordinate correlations of nodes by identifying the relationships between adjacent joint nodes. In particular, models with graph structures and a self-attention mechanism [26, 2] show remarkable performance. In addition, many RNN-based and CNN-based temporal modules have



(a) Spatio-Temporal Curve Generation

(b) Curve Aggregation

Figure 2. (a) Process of constructing an spatio-temporal curve and (b) the curve aggregation module that applies the curves to the input feature map $\mathbf{F}_{\text{in}}$. The dotted line in (a) is the relationship between the query node and the key nodes extracted by the inter-frame k-NN. The node selection policy π adopts the node of the highest correlation score with the query node as the next point in the curve. $||$ and $\times$ denote concatenation and matrix multiplication, respectively.

been proposed to deal with action recognition sequence data defined as time series.

However, there are drawbacks to these methods, and the first is that structurally distant nodes are treated almost independently because of their small spatial receptive fields, because the given graph considers only the adjacent relationships of nodes. To treat this limitation, learnable graph structures with self-attention mechanisms [27, 2, 6] are proposed. However, these structures use not only the graphs with self-attention mechanisms, but also given graph with adjacent connections. In other words, even if the receptive fields of their models are enlarged with self-attention mechanisms, they do not highlight the appropriate nodes much because they are highly biased to the given graph. Second, although several methods for handling skeleton sequences [19, 20] have been proposed, most of them do not take account of spatio-temporal dependencies because they consider only node-wise sequences. In other words, most recent methods are vulnerable to this dependency because of their low spatio-temporal receptive fields.

2.2. Curves for 3D Point Clouds

3D point clouds are unstructured representations of 3D coordinates, tasks for them are to analyze the information for 3D points that exist without any adjacency matrices given. In order to effectively analyze these point clouds, Xiang *et al.* [35] propose CurveNet, which identifies the relationships between every 3D points by aggregating both local and non-local features. In one scene, CurveNet initializes n starting points and constructs n curves with length l by finding the most correlated points. Applying the curves to the feature space, both the local and non-local point connections are identified. We propose an spatio-temporal curve module that makes the curves applicable to skeletal video data, and effectively increases the spatio-temporal receptive field.

3. Methodology

3.1. Spatio-Temporal Curve Module

In this subsection, we describe the spatio-temporal curve (STC) module to consider the direct spatio-temporal correlations of skeleton sequences.

Spatio-Temporal Curve Generation. In order to construct a curve between frame t and $(t+1)$, it is essential to find and connect the key node $\widetilde{v}_{t+1}$ that is most semantically close to the query node v_t, where v_t is a specific node on frame t. To specify the node $\widetilde{v}_{t+1}$, we select k nodes in frame $(t+1)$ that are semantically close to the query node on the feature space. The nodes are selected through inter-frame k-NN, which applies Euclidean distance-based k-NN algorithms between all adjacent frames to obtain the semantically closest k nodes in the feature space. However, if the query node v_t and obtained key node $\widetilde{v}_{t+1}$ refer to structurally identical locations, the model may identify only the same nodes between adjacent frames. Therefore, it hinders the model's ability to capture diverse curves. To prevent this problem, we apply the k-NN algorithm while excluding the node of frame $(t+1)$ that is located in the same structural position as query node v_t. The proposed inter-frame k-NN is as follows:

$$\widetilde{\mathcal{V}}_{(t,t+1)} = \sum_{v \in V} \text{k-NN}(v_t, \mathcal{V}_{t+1} - \{v_{t+1}\}), \tag{1}$$

where $\widetilde{\mathcal{V}}_{(t,t+1)}$ denotes a set of k nodes in frame $(t+1)$ that are the semantically closest nodes to the query node v_t, and $(\mathcal{V}_{(t+1)} - \{v_{t+1}\})$ refers to the node set in frame $(t+1)$ except the node that is structurally the same as v_t.

To construct effective curves, the key node most highly correlated with the query node v_t should be extracted using the node set $\widetilde{\mathcal{V}}_{(t,t+1)}$ created by inter-frame k-NN. For this extraction process, we propose an extended method for the node selection policy π in [35] to choose the key node $\widetilde{v}_{t+1}$. The policy in [35] reflects only nodes in a single frame without considering the temporal feature space. To consider the time domain as well, we apply a new node selection policy π_t to choose the key node by reflecting the key node features and semantically adjacent node set $\widetilde{\mathcal{V}}_{(t,t+1)}$.

$$s_{t+1} = \pi_t \left(s_t, \widetilde{\mathcal{S}}_{(t,t+1)} \right), \; 1 \leq t \in \mathbb{Z}^+ \leq T-1, \tag{2}$$

where s_t and s_{t+1} refer to the embedded spaces of the query and the key nodes, $\widetilde{\mathcal{S}}_{(t,t+1)}$ denotes the embedded features of $\widetilde{\mathcal{V}}_{(t,t+1)}$, and T is the length of the skeleton sequence, which is equal to $(curve\ length + 1)$. We build a learnable policy π_t to consider both the features of query node v_t and the features of extracted node set $\widetilde{\mathcal{V}}_{(t,t+1)}$. We obtain a new agent feature map $\mathbf{M}_{\text{agent}}$ by passing those features through an agent MLP layer. Then, we extract the node with the highest correlation score in adjacent frame $(t+1)$ via $\mathbf{M}_{\text{agent}}$. We obtain s_{t+1} from the input feature map $\mathbf{F}_{\text{in}}$ via $\widetilde{v}_{t+1}$, and it becomes the next waypoint of the curve. The process to choose the key node is as follows:

$$\text{M}_{\text{agent}} = \text{MLP}_{\text{agent}} \left(s_t \parallel \widetilde{\mathcal{S}}_{(t,t+1)} \right), \tag{3}$$

$$\pi \left(s_t, \widetilde{\mathcal{S}}_{(t,t+1)} \right) = \mathbf{F}_{\text{in}} \left[\arg\max(\mathbf{M}_{\text{agent}}) \right], \tag{4}$$

where $\parallel$ denotes the concatenation operation and $\mathbf{F}_{\text{in}}$ refers to the input feature map.

However, the weights of $\text{MLP}_{\text{agent}}$ cannot be updated smoothly during backpropagation due to the undifferentiable $\arg\max$ function. To solve this problem, we use a Gumbel Softmax function [11, 37], which is computed as a one-hot vector for forward operation, and updates the weight using the results of the softmax function for backward propagation. With these methods, our curves are represented as follows:

$$\mathbf{curve} = [s_1 \rightarrow s_2 \rightarrow \cdots \rightarrow s_T] \in \mathbb{R}^{C \times (T-1)}. \tag{5}$$

We set all joint nodes in the first frame to be the starting points of the curves, so the shape of the integrated curve is $\mathbf{curves} \in \mathbb{R}^{C \times (T-1) \times V}$, where V stands for the number of joint nodes.

Curve Aggregation. Inspired by [35], we use an aggregation module to effectively apply the curves to the input feature map. The process of the curve aggregation module is shown in Fig. 2 (b). By applying the curve aggregation, the model can consider both the relationship between nodes existing in one curve (intra-curve features $\mathbf{F}_{intra}$) and the relationship between every curves (inter-curve features $\mathbf{F}_{inter}$). To construct $\mathbf{F}_{intra}$, we first obtain $\mathbf{curve}_{intra} \in \mathbb{R}^{C_{mid} \times (T-1)}$ through the attentive pooling layer [10] and a simple MLP layer that reduces the number of channels. The $\mathbf{F}_{intra}$ is computed as follows:

$$\widetilde{\mathbf{F}}_{intra} = \texttt{softmax}\left(\mathbf{F}_{\text{in}} \times \mathbf{curve}_{intra}\right), \tag{6}$$

$$\mathbf{F}_{intra} = \mathbf{curve}_{intra} \mathbf{W}_{intra} \times \widetilde{\mathbf{F}}_{intra}, \tag{7}$$

where $\mathbf{W}_{intra} \in \mathbb{R}^{C_{mid} \times C_{mid}}$ is an MLP layer that linearly transforms the curve features. The $\mathbf{curve}_{intra}$ is applied to the input feature map $\mathbf{F}_{\text{in}}$ while the existing feature map shape $\in \mathbb{R}^{C_{mid} \times T \times V}$ is preserved. The $\mathbf{F}_{inter}$ is obtained in a same way to Eq. (6) and Eq. (7). Our curve aggregation module is shown in Fig. 2 (b). To aggregate the two different features, we use the following method:

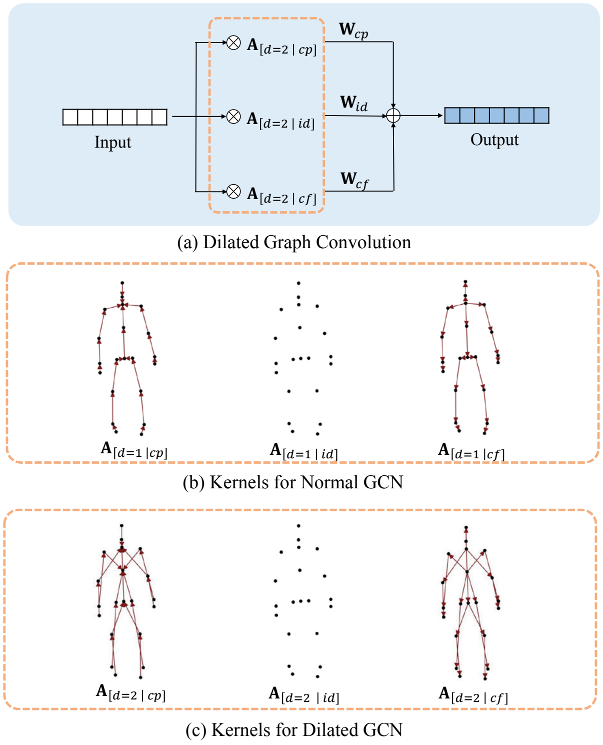
$$\mathbf{F}_{\text{out}} = \left(\mathbf{F}_{intra} \parallel \mathbf{F}_{inter}\right) \mathbf{W}_{agg} \in \mathbb{R}^{C \times T \times V}, \tag{8}$$

where $\mathbf{W}_{agg} \in \mathbb{R}^{2C_{mid} \times C}$ integrates $\mathbf{F}_{intra}$ and $\mathbf{F}_{inter}$. Finally, a new feature map $\mathbf{F}_{\text{out}}$ is generated, where the curve features are applied to the input feature map $\mathbf{F}_{\text{in}}$, and it enables the model to consider the spatio-temporal dependencies.

3.2. Dilated Kernels for Graph Convolution

In this subsection, we propose dilated kernels for graph convolution on the skeletal graph structure, which increase spatial receptive field without any additional parameters.

Kernel Analogy from CNNs to GCNs. The kernels for CNNs are applied to networks to aggregate local features in pixel units. In particular, for convolution to a single axis, local features are largely divided into ["Left", "Identity", "Right"] when the size of the kernel is 3. Those features are integrated into a representative information that identifies adjacent pixels by weighted summation. For example, if the "Left", "Identity", and "Right" features are symbolized as -1, 0, and 1, respectively, a kernel with a dilation of 2 can be expressed as [-2, 0, 2]. Therefore, the CNN operation



$\mathbf{W}_{cp}$
$\mathbf{A}_{[d=2\,|\,cp]}$
Input
$\mathbf{W}_{id}$
$\mathbf{A}_{[d=2\,|\,id]}$
Output
$\mathbf{W}_{cf}$
$\mathbf{A}_{[d=2\,|\,cf]}$

(a) Dilated Graph Convolution

$\mathbf{A}_{[d=1\,|cp]}$
$\mathbf{A}_{[d=1\,|\,id]}$
$\mathbf{A}_{[d=1\,|cf]}$

(b) Kernels for Normal GCN

$\mathbf{A}_{[d=2\,|\,cp]}$
$\mathbf{A}_{[d=2\;\,|\,id]}$
$\mathbf{A}_{[d=2\,|\,cf]}$

(c) Kernels for Dilated GCN

Figure 3. (a) Structure of dilated graph convolution, and comparison of (b) kernels for normal GCN and (c) kernels for dilated GCN. The arrows in (b) and (c) include information about the direction of the edges and the start and end points. The dilation value for kernels of (c) is fixed to 2.

with a dilation of d is as follows:

$$\mathbf{F}_{\text{out}} = \sum_{k\in[-1,0,1]} \mathbf{F}_{\text{in}\ [\ p\ +\ k\ *\ d\ ]}\,\mathbf{W}_k, \tag{9}$$

where p denotes location of the pixel and $\mathbf{W}_k$ denotes the weights of the kernel-wise MLP layer.

In non-Euclidean geometry, the concept of "Right" and "Left" for CNNs cannot be used because those directions cannot be defined, especially for the graph structures. Yan *et al*. [36] first propose a method for selecting the root node and dividing the kernel into ["Centripetal" (cp), "Identity" (id), "Centrifugal" (cf)] to aggregate the local features for skeleton-based action recognition. According to this policy, the dilated GCN operation is as follows:

$$\Phi\left(\mathbf{F}_{\text{in}}, d\right) = \sum_{k\in[cp,id,cf]} \mathbf{A}_{[d|k]}\mathbf{F}_{\text{in}}\mathbf{W}_k, \tag{10}$$

where $\mathbf{A}_{[d|k]}$ denotes a normalized adjacency matrix according to the dilation window given the direction of the kernel, and Φ refers to the DK-GC operation. For example, the GCNs with a dilation of 2 aggregate node features by skipping one adjacent node. Our DK-GC operation is systematically similar to that of the CNN (e.g., number of parameters, floating point operations) as shown in Fig. 3 (a). In addition, because the $[id]$ kernel itself denotes an identity matrix, $\mathbf{A}_{[d=n|id]}$ is the same matrix as $\mathbf{A}_{[d=1|id]}$. In other words, adjacency matrices for a kernel size of 3 and a dilation of 2 are divided into $[\mathbf{A}_{[d=2|cp]}, \mathbf{A}_{[d|id]}, \mathbf{A}_{[d=2|cf]}]$. The kernels for GCNs with adjacent connectivity and DK-GCs are shown in Fig. 3 (b) and (c).

Graph Convolution with Dilated Kernels. To construct adjacency matrices with the dilated kernels, we simply extend the approach of Liu *et al*. [21]'s method. The k power of an adjacency matrix includes edges that are structurally k steps away from the query node. However, if only the power of the adjacency matrix is used to reflect nodes that are k steps away, the output matrix includes the paths back to the query node. To exclude these paths, we use the difference between the d power and the $(d-1)$ power of the adjacency matrix:

$$\begin{aligned}\tilde{\mathbf{A}}_{[\mathrm{d}|k]} = \lambda&\left((\tilde{\mathbf{A}}_{[d=1|k]} + \mathbf{I})(\tilde{\mathbf{A}}_{[d=1|cf,cp]} + \mathbf{I})^{\mathrm{d}-1}\right)\\ -\ \lambda&\left((\tilde{\mathbf{A}}_{[d=1|k]} + \mathbf{I})(\tilde{\mathbf{A}}_{[d=1|cf,cp]} + \mathbf{I})^{\mathrm{d}-2}\right),\end{aligned} \tag{11}$$

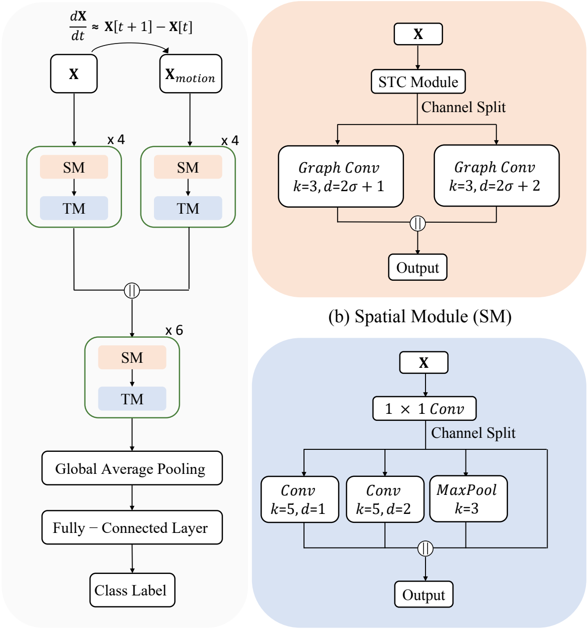
$$\mathbf{A}_{[\mathrm{d}|k]} = \mathbf{D}^{-\frac{1}{2}}_{[\mathrm{d}|k]}\tilde{\mathbf{A}}_{[\mathrm{d}|k]}\mathbf{D}^{-\frac{1}{2}}_{[\mathrm{d}|k]}, \tag{12}$$

where $\tilde{\mathbf{A}}$ and $\mathbf{A}$ stand for the unnormalized and normalized adjacency matrices, respectively, and $\mathbf{D}$ denotes the degree matrix for normalization. The λ function receives input as matrix and replaces all the elements greater than 1 with 1 and all values less than 1 with 0. Without the λ function, it hinders the optimization and convergence of the model since the elements for the overlapping paths become largely biased values. The λ function allows the model to converge stably while avoiding those biases on the edges for overlapping paths.

3.3. Network Architecture

Overall Architecture. We adopt the architecture of [36] as our baseline, which includes 10 spatio-temporal blocks. The output channels of each block are 64, 64, 64, 64, 128, 128, 128, 256, 256, and 256, and each block includes independent spatial and temporal module, and residual connections [9] for stable learning. For data composed of the time-series x_t, y_t, and z_t coordinates, we utilize the motion vector $\dfrac{d\mathbf{X}}{dt} \approx [\ x_{t+1} - x_t,\ y_{t+1} - y_t,\ z_{t+1} - z_t\ ]$ differently from existing methods [28, 2, 4]. Coordinate and motion data pass through four spatio-temporal blocks independently, and then they are concatenated to take account of those two modalities in a single network. Our overall architecture is shown in Fig. 4 (a).

Spatial Module. We construct a multi-branch spatial module based on DK-GC. Our spatial module has 2 layers



(b) Spatial Module (SM)

(a) Overall Architecture

(c) Temporal Module (TM)

Figure 4. Overall architecture of the model (a) and the spatial (b) and temporal modules (c). Graph convolutions of the spatial module and convolutions of the temporal module are the operations for the node axis and the frame axis, respectively.

in series and 2 operations in parallel, as shown in Fig. 4 (b). This module first passes the input feature map $\mathbf{F}_{\text{in}}$ through a STC module, which generates the curves and contains curve aggregation module. For efficient computational complexity, we place the STC modules only on specific layers of the model, and place point-wise convolution on the remaining layers. After the block, we split the transformed features into two branch operations along the channel axis, and feed them into two branch operations with an dilation scaling factor σ and fixed kernel size 3:

$$\mathbf{F}_{\text{out}} = \Big\Vert_{\tilde{d} \in [1,2]} \Phi\left(\mathbf{F}_{\text{in}},\ d = 2\sigma + \tilde{d}\right), \qquad (13)$$

where Φ denotes the DK-GC operation in Eq. (10) and d denotes the dilation of each GCN operation. Then, the two branch operation results are concatenated to create a new output feature map $\mathbf{F}_{\text{out}}$. Here, we train the model separately for $\sigma \in \{0, 1, 2\}$. The trained models are then used for ensemble, which enables us to combine the strengths of each individual model and improve overall performance.

Temporal Module. We adopt the multi-scale temporal convolution of [2] as the temporal module shown in Fig. 4 (c). This module includes one feature transformation block and four branch operations, similar to the spatial module. Two of their operations are temporal convolutions with dilations of 1 and 2, which have a kernel size of 5. The remaining operations are a max pooling layer with a kernel size of 3 and a identity layer. After all four operations are completed, the output feature map is constructed by concatenating all the resulting feature maps.

4. Experiments

4.1. Datasets

NTU-RGB+D 60. NTU-RGB+D 60 [25] is a large skeleton-based action recognition dataset that contains 56,880 action sequences. All the sequences are classified into a total of 60 classes. At most two subjects exist in an action sample. We follow two benchmarks suggested by the authors of the dataset. (1) Cross-Subject (X-Sub): The actions of 20 out of 40 subjects are used for training, and the actions of the remaining 20 are used for validation. (2) Cross-View (X-View): Two of the three camera views are used for training, and the other one is used for validation.

NTU-RGB+D 120. NTU-RGB+D 120 [18] is a dataset in which 57,367 action sequences are added from the NTU-RGB+D 60 dataset. All action sequences are classified into a total of 120 classes. We use two benchmarks proposed by the authors of this dataset. (1) Cross-Subject (X-Sub): The actions of 53 objects out of 106 objects are used for training, and the rest are used for validation. (2) Cross-Setup (X-Set): Among 32 numbered settings, even-numbered settings are used for training, and the others are used for validation.

Kinetics-Skeleton. The Kinetics-Skeleton dataset is derived from the Kinetics 400 video dataset [12], utilizing the OpenPose pose estimation [1] to extract 240,436 training and 19,796 testing skeleton sequences across 400 classes. Each skeleton graph includes 18 body joints with their 2D coordinates. We report Top-1 and Top-5 accuracies, following the convention.

Northwestern-UCLA. Northwestern-UCLA [32] is an action recognition dataset containing 1494 skeleton sequences. All the action samples are classified into 10 classes that are captured by three cameras at different angles. We use the protocol proposed by the authors: two of the three camera views are used for training and the other is for validation.

4.2. Experimental Settings

In our experiments, we set the number of epochs to 90, and we apply a warm-up strategy [9] to the first five epochs for more stable learning. We adopt an SGD optimizer with a Nesterov momentum of 0.9 and a weight decay of 0.0004.

Methods	Publication	NTU-RGB+D 60		NTU-RGB+D 120		Kinetics-Skeleton		Northwestern
		X-Sub (%)	X-View (%)	X-Sub (%)	X-Set (%)	Top-1 (%)	Top-5 (%)	UCLA
ST-GCN [36]	AAAI 2018	81.5	88.3	82.5	84.2	30.7	52.8	-
2s-AGCN [27]	CVPR 2019	88.5	95.1	88.5	95.1	36.1	58.7	-
DGNN [26]	CVPR 2019	89.9	96.1	-	-	36.9	59.6	-
AGC-LSTM [29]	CVPR 2019	89.2	95.0	-	-	-	-	93.3
Shift-GCN [5]	CVPR 2020	90.7	96.5	85.9	87.6	-	-	94.6
DC-GCN+ADG [4]	ECCV 2020	90.8	96.6	86.5	88.1	-	-	95.3
MS-G3D [21]	CVPR 2020	91.5	96.2	86.9	88.4	38.0	60.9	-
MST-GCN [3]	AAAI 2021	91.5	96.6	87.5	88.8	38.1	60.8	-
DDGCN [15]	ECCV 2020	91.1	97.1	-	-	38.1	60.8	-
CTR-GCN [2]	ICCV 2021	92.4	96.8	88.9	90.6	-	-	96.5
EfficientGCN-B4 [30]	TPAMI 2022	91.7	95.7	88.3	89.1	-	-	-
STF [13]	AAAI 2022	92.5	96.9	88.9	89.9	39.9	-	-
InfoGCN (4-ensemble) [6]	CVPR 2022	92.7	96.9	89.4	90.7	-	-	96.6
InfoGCN (4-ensemble) [6]	CVPR 2022	93.0	97.1	89.8	91.2	-	-	97.0
STC-Net (2-ensemble)		92.5	96.7	89.3	90.7	40.0	62.6	96.8
STC-Net (4-ensemble)		93.0	97.1	89.9	91.3	40.7	63.6	97.2
STC-Net (6-ensemble)		93.3	97.3	90.2	91.7	41.2	64.2	97.4

Table 1. **Comparison of the top-1 (or 5) accuracy (%) with the state-of-the-arts on NTU-RGB+D 60, NTU-RGB+D 120, Kinetics-Skeleton, and Northwestern-UCLA datasets.** The orange and yellow cells respectively indicate the highest and second-highest value.

The initial learning rate is set to 0.1, and we reduce the learning rate to 0.0001 through the cosine annealing scheduler [22]. We use Zhang *et al.* [39]'s data preprocessing method for NTU-RGB+D 60 and 120 datasets, and we set the batch size to 64. For the Northwestern-UCLA dataset, we use the data preprocessing method by Cheng *et al.* [5] and set the batch size to 16. For the Kinetics-Skeleton, we set the batch size to 64. In addition, the STC module is applied to the 3-rd, 6-th, and 9-th blocks of the 10 spatio-temporal blocks for memory efficiency, and pointwise convolution is applied to the remaining blocks. Our experiments are conducted on a single RTX 3090 GPU.

4.3. Comparison with the State-of-the-Arts

Many recent state-of-the-art models [28, 2, 4] use the ensemble method by training four data streams, i.e., joint, bone, joint motion, and bone motion. However, as networks with only the joint (bone) motion stream show inferior performances than networks with joint (bone) stream, learning two streams independently is inefficient. Unlike them, we train joint and joint motion streams on one network, as shown in Fig. 4 (a). We also train bone and bone motion streams on the network. We adopt the ensemble method of models trained with the dilation scaling factor σ as 0, 1, and 2 without considering the motion stream separately, where σ is explained in Sec. 3.3. In other words, we use the six models with the joint ($\sigma \in \{0, 1, 2\}$), bone ($\sigma \in \{0, 1, 2\}$) streams for our ensemble method.

We evaluate performance on four skeleton-based action recognition benchmarks. The performance comparisons for those datasets are shown in Tab. 1. We systematically evaluate the performance of our STC-Net with respect to the number of ensemble streams, and show that the 6-stream ensemble outperforms state-of-the-art performance on all datasets. For the all datasets, even with 4 ensembles, the STC-Net shows slightly better or about the same as [6], a state-of-the-art model with 6-stream ensemble.

Methods	M	Curve		CA	DK-GC	X-Sub (%)	X-Set (%)
		Intra	Inter				
A						83.5	85.4
B	✓					84.8 (↑ 1.3)	86.6 (↑ 1.2)
C	✓	✓				85.2 (↑ 1.7)	86.9 (↑ 1.5)
D	✓		✓			85.1 (↑ 1.6)	86.9 (↑ 1.5)
E	✓				✓	85.8 (↑ 2.3)	87.3 (↑ 1.9)
F	✓	✓	✓			85.4 (↑ 1.9)	87.2 (↑ 1.8)
G	✓	✓	✓		✓	86.0 (↑ 2.5)	87.7 (↑ 2.3)
H	✓	✓	✓	✓		85.9 (↑ 2.4)	87.7 (↑ 2.3)
I	✓	✓	✓	✓	✓	**86.2** (↑ 2.7)	**88.0** (↑ 2.6)

Table 2. **Performance comparison of variants of STC-Net.** M: motion, Intra: $\mathbf{curve}_{intra}$, Inter: $\mathbf{curve}_{inter}$, CA: Curve Aggregation module

4.4. Ablation Study

In this section, we conduct several experiments to prove the superiority of our proposed modules. The experiments are performed by dividing the components of our STC-Net to $\mathbf{curve}_{intra}$, $\mathbf{curve}_{inter}$, curve aggregation module, and DK-GC. The performance described in this section refers to cross-subject and cross-setup accuracy on the NTU-RGB+D 120 joint stream.

Spatio-Temporal Curve Module. In order to prove the effectiveness of the STC module, we specify a baseline model in which the temporal module of [2] is applied to the architecture in [36]. According to Tab. 2, the perfor-

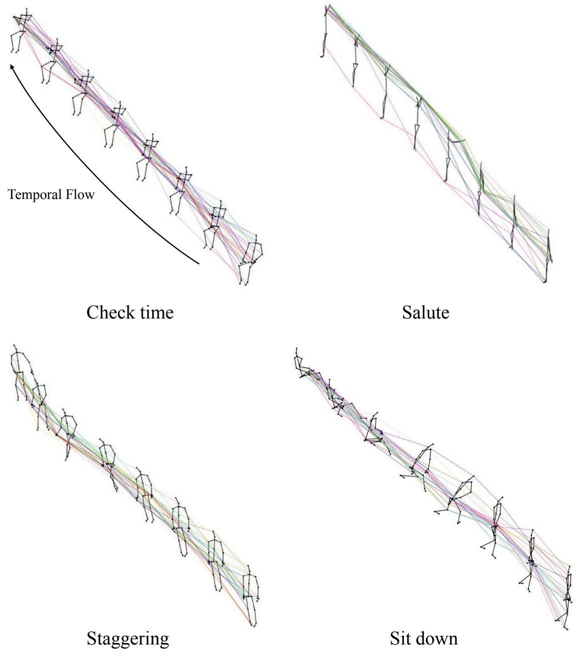


Figure 5. Curve visualizations for several samples. To make a clear distinction, each curve is represented by a unique color.

mance increases by 1.25% only with motion data. If we use either $\mathbf{curve}_{intra}$ or $\mathbf{curve}_{inter}$ via simple non-local block structure [34] instead of curve aggregation module, the performance improves slightly. It is due to the failure to properly utilize the two curve features designed for attention to the input feature map. Thus, we observe that the effect is more pronounced when using the curve aggregation module. The model H in Tab. 2 includes all the components of the spatio-temporal curve and shows a performance that is 2.35% higher than the baseline model A.

Methods	Kernel Size	X-Sub (%)	X-Set (%)	Params	FLOPs
STC-Net	3	86.2	88.0	1.46 M	1.88 G
STC-Net	5	86.4	88.1	1.78 M	2.32 G
STC-Net	7	86.2	87.9	2.09 M	2.76 G

Table 3. Comparison of different DK-GC based on kernel size.

Dilated Kernels for Graph Convolution. According to Tab. 2, the model E with only DK-GC shows 2.1% higher performance than the baseline model A. Furthermore, its effectiveness is more pronounced when used with the STC module (model I), which is 2.65% higher than the baseline model A. To find the optimal kernel size of the DK-GC, we conduct additional experiments by setting the kernel size to 3, 5, and 7 with the dilation scaling factor $\sigma = 0$. Tab. 3 shows that performance of our model does not change according to the size of the kernel. It is due to the relatively small number of joint nodes for action recognition. Therefore, considering that there is little difference in performance, selecting the smallest kernel size of 3 is most efficient in term of parameters and computation.

Methods	# Ensembles	NTU-RGB+D 120		Params	FLOPs
		X-Sub (%)	X-Set (%)		
MS-G3D [21]	2	86.9	88.4	6.44 M	24.50 G
InfoGCN [6]	2	88.5	89.7	3.14 M	**3.36 G**
STC-Net	2	**89.3**	**90.7**	**2.92 M**	3.70 G
DC-GCN [4]	4	86.5	88.1	13.80 M	51.52 G
MST-GCN [3]	4	87.5	88.8	11.68 M	67.12 G
CTR-GCN [2]	4	88.9	90.6	5.84 M	7.88 G
InfoGCN [6]	4	89.4	90.7	6.28 M	**6.72 G**
STC-Net	4	**89.9**	**91.3**	**5.84 M**	7.40 G
InfoGCN [6]	6	89.8	91.2	9.42 M	**10.08 G**
STC-Net	6	**90.2**	**91.7**	**8.76 M**	11.10 G

Table 4. Comparison of multi-stream complexity of the state-of-the-arts according to the number of ensemble streams.

4.5. Curve Visualization

To qualitatively evaluate whether the curves are well-generated, we visualize the curves for several samples as shown in Fig. 5. For the "Check time" and "Salute" classes, the curves start from every node in the first frame, and those curves tend to proceed toward hand or arm nodes. Inspired by human visual recognition, it is reasonable that the hand gestures should be highlighted for those classes. Similarly, the curves of "Staggering" and "Sit down" classes tend to directed toward lower body, which is also reasonable in that the leg gestures are important for those classes.

4.6. Analysis of Computational Complexity

A comparison of the complexity between our model and state-of-the-arts is shown in Tab. 4. Our 4-stream STC-Net shows slightly better performance than 6-stream InfoGCN [6] while having the fewer GFLOPs (×0.62) and fewer parameters (×0.73). With 6-stream ensemble, our model outperforms [6] by a larger margin.

5. Conclusions

In this paper, we propose a novel Spatio-Temporal Curve Network (STC-Net) for skeleton-based action recognition, which consists of spatial modules with an spatio-temporal curve (STC) module and graph convolution with dilated kernels (DK-GC). Our STC module constructs spatio-temporal curves by connecting the most highly correlated nodes in successive frames, which significantly increases the spatio-temporal receptive field. Our DK-GC is carefully designed for skeleton-based action recognition to give the model a large spatial receptive field through dilated graph kernels. By combining these two methods, we implement STC-Net and demonstrate its superiority through extensive experiments, and our proposed model outperforms existing methods on four benchmarks.

Acknowledgements. This research was supported by Multi-Ministry Collaborative R&D Program (R&D program for complex cognitive technology) through the National Research Foundation of Korea(NRF) funded by MSIT, MOTIE, KNPA(NRF-2018M3E3A1057289) and the KIST Institutional Program (Project No.2E32283-23-064).

References

[1] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 6

[2] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13359–13368, 2021. 1, 2, 3, 5, 6, 7, 8

[3] Zhan Chen, Sicheng Li, Bing Yang, Qinghan Li, and Hong Liu. Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1113–1122, 2021. 7, 8

[4] Ke Cheng, Yifan Zhang, Congqi Cao, Lei Shi, Jian Cheng, and Hanqing Lu. Decoupling gcn with dropgraph module for skeleton-based action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 5, 7, 8

[5] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 183–192, 2020. 7

[6] Hyung-gun Chi, Myoung Hoon Ha, Seunggeun Chi, Sang Wan Lee, Qixing Huang, and Karthik Ramani. Infogcn: Representation learning for human skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20186–20196, 2022. 1, 3, 7, 8

[7] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28, 2015. 1, 2

[8] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 1

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 5, 6

[10] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Learning semantic segmentation of large-scale point clouds with random sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 4

[11] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 4

[12] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 6

[13] Lipeng Ke, Kuan-Chuan Peng, and Siwei Lyu. Towards to-at spatio-temporal focus for skeleton-based action recognition. *arXiv preprint arXiv:2202.02314*, 2022. 7

[14] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pages 2688–2697. PMLR, 2018. 1, 2

[15] Matthew Korban and Xin Li. Ddgcn: A dynamic directed graph convolutional network for action recognition. In *European Conference on Computer Vision*, pages 761–776. Springer, 2020. 7

[16] Jungho Lee, Minhyeok Lee, Dogyoon Lee, and Sangyoon Lee. Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:2208.10741*, 2022. 1

[17] Guohao Li, Matthias Müller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[18] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2684–2701, 2019. 2, 6

[19] Jun Liu, Amir Shahroudy, Dong Xu, Alex C Kot, and Gang Wang. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):3007–3021, 2017. 3

[20] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C Kot. Global context-aware attention lstm networks for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1647–1656, 2017. 3

[21] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 143–152, 2020. 2, 5, 7, 8

[22] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 7

[23] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124, 2017. 1

[24] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs.

In *International conference on machine learning*, pages 2014–2023. PMLR, 2016. 2

[25] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1010–1019, 2016. 2, 6

[26] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7912–7921, 2019. 1, 2, 7

[27] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12026–12035, 2019. 1, 2, 3, 7

[28] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *IEEE Transactions on Image Processing*, 29:9532–9545, 2020. 2, 5, 7

[29] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1227–1236, 2019. 7

[30] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Constructing stronger and faster baselines for skeleton-based action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 7

[31] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. Differential recurrent neural networks for action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 4041–4049, 2015. 1

[32] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2649–2656, 2014. 2, 6

[33] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 1

[34] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 8

[35] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 915–924, October 2021. 2, 3, 4

[36] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI Conference on Artificial Intelligence*, 2018. 1, 2, 5, 7

[37] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3323–3332, 2019. 4

[38] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31, 2018. 1

[39] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jianru Xue, and Nanning Zheng. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1112–1121, 2020. 7