Spatial-Temporal Adaptive Graph Convolutional Network for Skeleton-Based Action Recognition

Rui Hang and MinXian Li(✉)

Nanjing University of Science and Technology, Nanjing, China
{hangrui,minxianli}@njust.edu.cn

Abstract. Skeleton-based action recognition approaches usually construct the skeleton sequence as spatial-temporal graphs and perform graph convolution on these graphs to extract discriminative features. However, due to the fixed topology shared among different poses and the lack of direct long-range temporal dependencies, it is not trivial to learn the robust spatial-temporal feature. Therefore, we present a spatial-temporal adaptive graph convolutional network (STA-GCN) to learn adaptive spatial and temporal topologies and effectively aggregate features for skeleton-based action recognition. The proposed network is composed of spatial adaptive graph convolution (SA-GC) and temporal adaptive graph convolution (TA-GC) with an adaptive topology encoder. The SA-GC can extract the spatial feature for each pose with the spatial adaptive topology, while the TA-GC can learn the temporal feature by modeling the direct long-range temporal dependencies adaptively. On three large-scale skeleton action recognition datasets: NTU RGB+D 60, NTU RGB+D 120, and Kinetics Skeleton, the STA-GCN outperforms the existing state-of-the-art methods. The code is available at https://github.com/hang-rui/STA-GCN.

Keywords: Action recognition · Adaptive topology · Graph convolution

1 Introduction

Action recognition is an essential task in human-centered computing and computer vision, which plays an increasingly crucial role in video surveillance, human-computer interaction, video analysis, and other applications [1,26,38]. In recent years, skeleton-based human action recognition has attracted much attention due to the development of depth sensors [46] and pose estimation algorithms [2,33]. Conventional deep learning methods adopt re-current neural networks (RNN) [8,20,43] and convolutional neural networks (CNN) [14–16] to analyze the skeleton sequence by representing it as vector sequence or pseudo-image. However, the skeleton sequence is naturally structured as a spatial-temporal

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-26316-3_11.

L. Wang et al. (Eds.): ACCV 2022, LNCS 13844, pp. 172–188, 2023.
https://doi.org/10.1007/978-3-031-26316-3_11

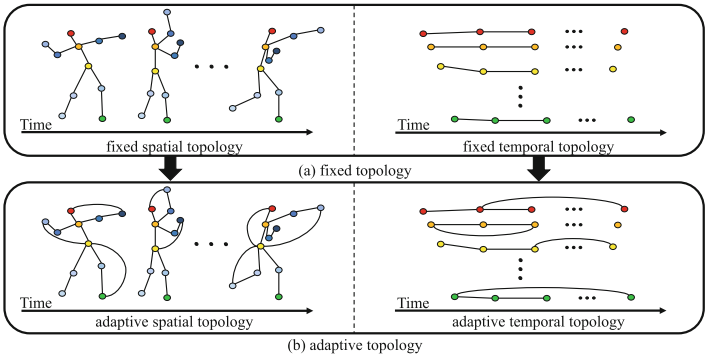

Fig. 1. Illustration of (a) the fixed spatial and temporal topology and (b) the adaptive spatial and temporal topology. Different color points indicate the different human joints, and the lines correspond to the spatial and temporal direct correlation between joints. Best viewed in color.

graph. For this reason, Yan *et al.* [40] firstly proposed the spatial-temporal graph convolutional network (ST-GCN) to model the motion patterns of action on a skeleton spatial-temporal graph. After that, a series of graph convolutional networks (GCN) methods based on spatial-temporal graphs have been proposed for skeleton-based action recognition [7,22,29].

However, there are still two disadvantages to the feature extraction operations on the spatial-temporal graph: (1) In the stage of learning spatial features, the fixed spatial topology is shared among all poses, which may not be optimal for the action with large changes in pose. For action such as "throw", before and after the throw, the human pose presents two forms of backward-leaning and forward-leaning, which represents different semantics. Using the fixed spatial topology may mistakenly enhance irrelevant connections or weaken critical ones, failing to accurately represent the spatial dependencies. This fact suggests that the spatial topology should be adaptive to each pose in the skeleton sequence. (2) In the stage of learning temporal features, existing methods apply temporal convolution with a fixed small kernel to extract the short-range temporal feature. It leads to the weak capacity to model temporal long-range joint dependencies vital for action recognition.

To learn the robust feature representation in the spatial and temporal dimensions, we propose a spatial-temporal Adaptive Graph Convolutional Network (STA-GCN) in this work. The proposed network is composed of two key modules: Spatial Adaptive Graph Convolution (SA-GC) and Temporal Adaptive Graph Convolution (TA-GC). Both SA-GC and TA-GC have a critical embedded component: Topology Adaptive Encoder (TAE). The SA-GC module is designed to extract spatial features by modeling the spatial adaptive joint dependencies. The TA-GC module is designed to learn temporal features by capturing the direct long-range joint dependencies in the temporal dimension. Combined with SA-GC and TA-GC modules, the proposed model can learn the discriminative features both in the spatial and temporal dimensions.

The TAE component is proposed to learn spatial adaptive topology and temporal adaptive topology. The existing GCN methods use fixed spatial and temporal topology (Fig. 1(a)). This fixed spatial topology forces each skeleton frame to adopt the same spatial topology, while the fixed temporal topology forces all trajectories to use the same temporal topology. We argue that this fixed topology is insufficient to represent the joint dependencies per pose or per trajectory. Therefore, we propose the TAE to solve this problem by learning the spatial adaptive topology and the temporal adaptive topology (Fig. 1(b)). The spatial adaptive topology can generate the pose-specific dependencies for each frame in the skeleton sequence to learn discriminative spatial features. The temporal adaptive topology can model the direct long-range dependencies between any two joints in the trajectory graph to extract robust temporal features.

The main contributions of this work are summarized as follows:

- A spatial adaptive graph convolution (SA-GC) module is proposed to extract the spatial feature for each pose with the spatial adaptive topology.
- A temporal adaptive graph convolution (TA-GC) module is proposed to learn the temporal feature by modeling the direct long-range temporal dependencies.
- A topology adaptive encoder (TAE) embedded into graph convolution is proposed to generate the adaptive spatial topology and the adaptive temporal topology.
- We propose a Spatial-Temporal Adaptive Graph Convolutional Network (STA-GCN) composed of SA-GC and TA-GC, which outperforms state-of-the-art approaches on three large-scale skeleton action recognition datasets: NTU RGB+D [27], NTU RGB+D 120 [19], and Kinetics-skeleton [13].

2 Related Work

2.1 Skeleton-Based Action Recognition

With the development of deep learning based video understanding technology, a series of video-based methods were proposed for action recognition. Specifically, 2D CNNs [18,36] efficiently recognize actions by modeling the relationships in the temporal dimension, while 3D CNNs [3,34] capture motion information in a unified network through a simple extension from the spatial domain to the spatial-temporal domain. Recently skeleton-based methods [7,8,14–16,20,22,24,29,39,40,42,43] have been developed extensively since skeleton data are more computationally efficient and exhibit stronger robustness. Skeleton data can eliminate the influences of variations of illumination, camera viewpoints, background changes, and clothing variance in real-world videos [21,35,37]. Therefore, we adopt the skeleton-based action recognition approach in this paper.

2.2 Graph Convolution Networks in Action Recognition

The type of skeleton-based action recognition approaches is divided into three categories: RNN-based, CNN-based, and GCN-based. RNN represents the skele-

ton sequence as a vector sequence [8,20,43], while CNN represents it as a pseudo-image [14–16]. However, both RNN and CNN methods ignore the information of skeleton topology among joints. GCN-based methods represent the skeleton sequence as a spatial-temporal graph and extract features in spatial and temporal dimensions by graph convolution and temporal convolution respectively. Yan [40] *et al.* firstly introduce graph convolution and temporal convolution into the skeleton-based action recognition to model the spatial configurations and temporal dynamics simultaneously. The following works improved the model by performing multi-hop methods to extract multi-scale features [12,17], applying additional mechanisms to adaptively capture the relations between distant joints [17,41,44], and adding additional edges between adjacent frames to extract features based on the extended graph [9,22,23]. However, all these methods only apply Graph Convolution Network (GCN) in the spatial dimension. In the temporal dimension, Temporal Convolution Network (TCN) is used to learn the temporal feature. It is not powerful due to the use of a fixed small temporal convolution kernel. In this work, we apply graph convolution both in spatial and temporal dimensions. Temporal graph convolution can effectively learn the temporal feature by modeling the direct long-range temporal dependencies.

2.3 Topology Adaptive-Based Methods

The topology-based adaptive methods [17,29,41,45] aim to generate the appropriate spatial topology based on the input data. This spatial topology indicates whether and how important a connection exists between any two joints in the graph. Existing methods focus only on the spatial dimension, and they mainly use parametric adjacency matrices to learn a spatial topology optimized for all data [29] or generate a specific spatial topology for each action sample based on the input data [17,41,45]. These methods alleviate the limitation caused by the fixed pre-defined spatial topology of GCN methods. However, existing methods force all frames to share the same spatial topology, which is unreasonable because each pose represents a different semantic and cannot use the same spatial topology to extract features. Moreover, these approaches do not consider topology in the temporal dimension, which hinders the long-range dependencies modeling in the temporal dimension. To address these problems, we propose the Topology Adaptive Encoder (TAE) to learn spatial adaptive topology for each frame in the spatial graph and temporal adaptive topology for each trajectory in the temporal graph. The spatial adaptive topology can efficiently help spatial graph convolution extract the features of different poses in action. On the other hand, the temporal adaptive topology is used in temporal graph convolution to extract long-range dependencies.

3 Methodology

3.1 Preliminaries

Notations. A human skeleton with N joints is presented as an undirected spatial graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_i | i = 1, \ldots, N\}$ is the set of N vertices

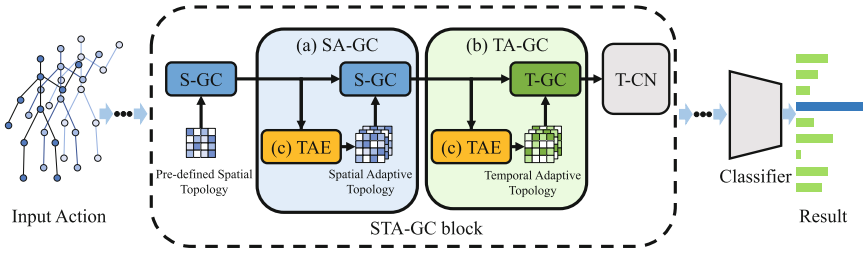

Fig. 2. The overview of the proposed Spatial-Temporal Adaptive Graph Convolutional Network (STA-GCN). (a) The spatial adaptive graph convolution (SA-GC) module is adopted to learn the spatial feature for each pose with the spatial adaptive topology. (b) The temporal adaptive graph convolution (TA-GC) module is subsequently adopted to learn the temporal feature for each trajectory with the temporal adaptive topology. (c) The Topology Adaptive Encoder (TAE) is used both in SA-GC and TA-GC modules to learn the spatial adaptive topology and the temporal adaptive topology.

representing joints and $\mathcal{E}$ is the edges set representing bones. The topology of $\mathcal{G}$ is formulated as an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ and its element $a_{ij} \in (0, 1)$ denoting weather an edge exists between vertex v_i and v_j.

Graph Convolution. According to Yan *et al.* [40], the spatial dependencies of the joints in each frame can be conveniently encoded with Graph Convolution. The operation of Graph Convolution is formulated as:

$$\mathbf{X}^{(l+1)} = \sigma\left(\sum_{k=0}^{K} \widetilde{\mathbf{A}}_{(k)} \mathbf{X}^{(l)} \mathbf{W}_{(k)}^{(l)}\right) \tag{1}$$

where $\mathbf{X}$ is the feature of each layer, $\sigma(\cdot)$is an activation function. K denotes the pre-defined maximum graphic distance. $\widetilde{\mathbf{A}}_{(k)} = \mathbf{\Lambda}_{(k)}^{-\frac{1}{2}}(\mathbf{A}_{(k)} + \mathbf{I}_{(k)})\mathbf{\Lambda}_{(k)}^{-\frac{1}{2}}$ is the k-th order normalized adjacency matrix, where $\mathbf{A} + \mathbf{I}$ is the skeleton graph with added self-loops to keep identity features, $\mathbf{\Lambda}$ is the diagonal degree matrix of $(\mathbf{A} + \mathbf{I})$. $\mathbf{W}_{(k)}$ is learnable parameters to implement the convolution operation.

3.2 Overview

To learn discriminative features on the spatial and temporal graph, we propose a Spatial-Temporal Adaptive Graph Convolutional Network (STA-GCN). An overview of our proposed method is illustrated in Fig. 2. After extracting local skeleton features using a pre-defined spatial topology (i.e. human skeleton structure), we adopt Spatial Adaptive Graph Convolution (SA-GC) and Temporal Adaptive Graph Convolution (TA-GC) to learn discriminative features. The SA-GC is proposed to extract the spatial feature for each pose with the spatial adaptive topology. The TA-GC is proposed to learn the temporal feature by modeling the direct long-range temporal dependencies. Both SA-GC and

TA-GC have an embedded Topology Adaptive Encoder (TAE), which generates a unique and appropriate topology for each pose and each trajectory. On the top of SA-GC and TA-GC, we apply a temporal convolutional convolution to aggregate the features in the temporal dimension. Based on the discriminative features extracted by several STA-GC blocks, we use a fully connected layer with softmax activation function to obtain the final class. The method will be discussed in detail in subsequent sections.

3.3 Spatial and Temporal Graph Convolution

Most existing works treat the human skeleton sequence as a spatial-temporal graph where features are extracted through spatial graph convolution and temporal convolution. However, the temporal convolution is not powerful to learn the temporal feature due to the use of a fixed small temporal convolution kernel. Therefore, in this section, we introduce a more robust feature extraction operator on the spatial and temporal graph. Let us first consider a graph convolution in spatial-temporal graph $\mathcal{G}^{st} = (\mathcal{V}^{st}, \mathcal{E}^{st})$ where $\mathcal{V}^{st} = \{v_{nt} | n = 1, \ldots, N, t = 1, \ldots, T, \}$ is the set of all nodes across T frames in the skeleton sequence and $\mathcal{E}^{st}$ is the spatial-temporal edge set.

We further deconstruct a spatial-temporal graph into T spatial graphs across time and N temporal graphs across joints. The spatial graphs are represented as $\mathcal{G}^{s} = (\mathcal{V}^{st}, \mathcal{E}^{s})$ where $\mathcal{E}^{s}$ is the spatial edge set and is formulated as a spatial adjacency matrix $\mathbf{A}^{s} \in \mathbb{R}^{T \times N \times N}$. Note that the spatial adjacency matrix is degraded to vanilla form $\mathbf{A} \in \mathbb{R}^{N \times N}$ when all spatial graphs have the same spatial correlations. Similarly, the temporal graphs are represented as $\mathcal{G}^{t} = (\mathcal{V}^{st}, \mathcal{E}^{t})$ and the temporal adjacency matrix $\mathbf{A}^{t} \in \mathbb{R}^{N \times T \times T}$ can be formulated. After the graph decomposition, we develop two graph convolutions: spatial graph convolution (S-GC) and temporal graph convolution (T-GC). S-GC and T-GC are respectively formulated as:

$$f_{out}\left(v_{nt}\right) = \sum_{p=1}^{N} a^{s}_{(pt)(nt)} f_{in}\left(v_{pt}\right) \mathbf{w}\left(v_{pt}\right) \tag{2}$$

$$f_{out}\left(v_{nt}\right) = \sum_{q=1}^{T} a^{t}_{(nq)(nt)} f_{in}\left(v_{nq}\right) \mathbf{w}\left(v_{nq}\right) \tag{3}$$

where $a^{s}_{(pt)(nt)}$ and $a^{t}_{(nq)(nt)}$ are elements of $\mathbf{A}^{s}$ and $\mathbf{A}^{t}$, respectively. Features on the spatial-temporal graph can be extracted by employing S-GC and T-GC. The whole process of feature extraction is formulated as:

$$f_{out}\left(v_{nt}\right) = \sum_{q=1}^{T} a^{t}_{(nq)(nt)} \left(\sum_{p=1}^{N} a^{s}_{(pq)(nq)} f_{in}\left(v_{pq}\right) \mathbf{w}_1\left(v_{pq}\right)\right) \mathbf{w}_2\left(v_{nq}\right) \tag{4}$$

3.4 Topology Adaptive Encoder

Previous works force each pose in the skeleton sequence to share a fixed spatial topology. However, different poses represent different semantics, and using the same topology will incorrectly extract spatial features for different poses, which leads to weak performance in recognizing action with large changes in pose. Moreover, existing methods do not apply graph convolution in the temporal dimension. If graph convolution is to be performed in the temporal dimension, a suitable temporal topology is needed to represent the relationships between joints in the temporal graph.

To generate a more detailed topology, we propose a unified topology generation module, Topology Adaptive Encoder (TAE), which applies an unshared topology generation strategy. Specifically, the TAE module generates the spatial topology for each pose in spatial graph convolution and the temporal topology for each trajectory in temporal graph convolution. Technically, the TAE applies self-attention operations to extract correlations between joints in the embedding space. The function is formulated as:

$$\mathrm{A} = \ell_2\_\mathrm{norm}(\mathrm{X}\mathrm{W}_\phi \mathrm{W}_\psi^T \mathrm{X}^T) \tag{5}$$

where X is the input feature, we first utilize two linear transformation functions ϕ and ψ to embed input feature into the embedding space, W_ϕ and W_ψ are the parameters of the embedding functions ϕ and ψ, respectively. Then, the two embedded feature matrices are multiplied to obtain a topology matrix A. Finally, the ℓ_2 normalization is applied to each row of the adjacency matrix, which eases the optimization and with the help of ℓ_2 normalization, the normalization of node degree is unnecessary.

3.5 Spatial and Temporal Adaptive Graph Convolution

We integrate the proposed TAE into S-GC and T-GC to obtain a pair of adaptive graph convolution operations on the spatial-temporal graph: Spatial Adaptive Graph Convolution (SA-GC) (Fig. 3 (a)) and Temporal Adaptive Graph Convolution (TA-GC) (Fig. 3 (b)). Since SA-GC on the spatial graph and TA-GC on the temporal graph are equivalent operations, for simplicity, we only introduce SA-GC in the rest part. Its counterpart TA-GC can be deduced naturally.

Specifically, our SA-GC contains three parts: (1) feature transformation with function $\mathcal{T}^s(\cdot)$, (2) topology adaptive Encoding with function $\mathcal{M}^s(\cdot)$, (3) feature aggregation with function $\mathcal{A}^s(\cdot)$. Given the input feature $\mathrm{X} \in \mathbb{R}^{T\times N\times C}$, the output $\mathrm{Y} \in \mathbb{R}^{T\times N\times C'}$ of SA-GC is formulated as:

$$\mathrm{Y} = \mathcal{A}^s(\mathcal{T}^s(\mathrm{X}), \mathcal{M}^s(\mathrm{X})) \tag{6}$$

Feature Transformation. The goal of feature transformation is to transform input features into high-level representations using function $\mathcal{T}^s(\cdot)$. Here we use the simple linear transformation function which is formulated as:

$$\mathrm{F}^s = \mathcal{T}^s(\mathrm{X}) = \mathrm{X}\mathrm{W}^s \tag{7}$$

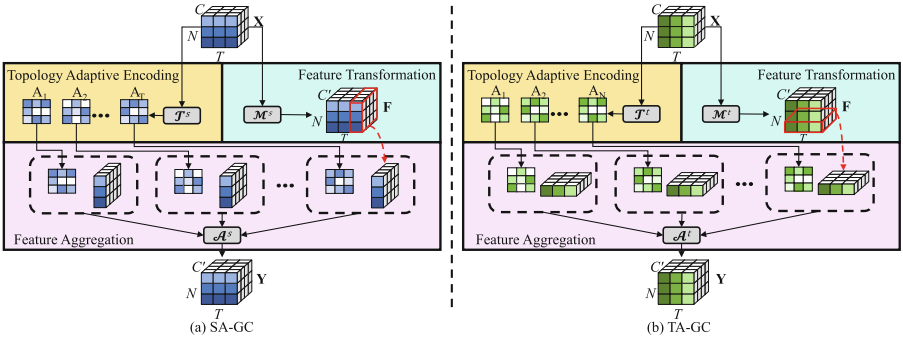

Fig. 3. (a) The Spatial Adaptive Graph Convolution (SA-GC) module. (b)The Temporal Adaptive Graph Convolution (TA-GC) module.

where $\mathrm{F}^s \in \mathbb{R}^{T\times N\times C'}$ is the transformed high-level features and $\mathrm{W}^s \in \mathbb{R}^{C\times C'}$ is the weight matrix.

Topology Adaptive Encoding. The topology adaptive encoding part in SA-GC is to use the proposed TAE module to generate a topology optimized for each skeleton individually. The function is formulated as:

$$\mathrm{A}^s = \mathcal{M}^s(\mathrm{X}) = \ell_2\_\mathrm{norm}(\mathrm{X}\mathrm{W}_\phi \mathrm{W}_\psi^T \mathrm{X}^T) \tag{8}$$

where X is the input feature, W_ϕ and W_ψ are the parameters of the embedding functions ϕ and ψ to embed the features. Then, the two feature maps are reshaped to matrix $\mathrm{M}_\phi \in \mathbb{R}^{T\times N\times C}$ and $\mathrm{M}_\psi \in \mathbb{R}^{T\times C\times N}$. By multiplying these two matrices to obtain a topology matrix $\mathrm{A}^s \in \mathbb{R}^{T\times N\times N}$, whose element represents the correlation between joints on a specific frame. Finally, the ℓ_2 normalization is applied to each row of the adjacency matrix.

Feature Aggregation. As indicated in Fig. 3, given the temporal-wise spatial adaptive adjacency matrix A^s from input samples, we aggregate high-level features F^s with temporal-wise feature aggregation function $\mathcal{A}^s$. The function is formulated as:

$$\mathrm{Y} = \mathcal{A}^s(\mathrm{A}^s, \mathrm{F}^s) = [\mathrm{A}_1^s\mathrm{F}_1^s \parallel_t \mathrm{A}_2^s\mathrm{F}_2^s \parallel_t \cdots \parallel_t \mathrm{A}_T^s\mathrm{F}_T^s] \tag{9}$$

where $\parallel_t$ is concatenation operation along the temporal dimension. $\mathrm{A}_t^s \in \mathbb{R}^{N\times N}$ and $\mathrm{F}_t^s \in \mathbb{R}^{N\times C}$ are respectively from t-th frame of A^s and F^s. During the whole process, the topology is optimized for each skeleton individually. Therefore, the proposed SA-GC can effectively distinguish actions, especially those with large changes in pose.

3.6 Model Details

The entire network consists of five STA-GC blocks, and the number of output channels of five blocks are 64-64-64-128-256. We apply a data BatchNorm layer at the start to normalize the input data. The temporal dimension is halved at the 4-th and 5-th blocks by strided temporal convolution. The temporal convolutional network used (TCN) in the STA-GC block is designed as multi-scale temporal convolutions following [22]. The main difference is that we reduce the number of channels and fuse six branches with a point-wise convolution. We also add extra residual connections to facilitate training. Due to the operations in SA-GC and TA-GC are equivalent, we only introduce a detailed implementation of SA-GC. We first utilize two linear transformation functions to transform input features into two neatly compact representations. Then, the two embedded features are multiplied to obtain spatial adaptive topology. We further apply ℓ_2 normalization to normalize the adjacency matrix, and the resulting adjacency matrix is used to apply the graph convolution.

4 Experiments

4.1 Datasets

NTU RGB+D 60 [27] is currently the most widely used indoor skeleton-based action recognition dataset, which contains $56,880$ skeleton action sequences with 60 action classes performed by 40 volunteers and captured by three Microsoft Kinect v2 cameras from different views concurrently. Each sample contains one action with two subjects at most, and each skeleton is composed of 25 joints. The dataset is separated into two benchmarks: (1) Cross-subject (X-Sub): $40,320$ samples performed by 20 subjects are separated into the training set, and the other $16,560$ samples performed by different 20 subjects belong to the test set. (2) Cross-view (X-View): the training set contains $37,920$ samples from camera views 2 and 3, and the test set contains $18,960$ samples from camera view 1.

NTU RGB+D 120 [19] is the largest indoor skeleton-based action recognition dataset, which extends NTU RGB+D 60 with additional 57,367 skeleton sequences over 60 extra action classes, totalling contains $114,480$ skeleton action sequences in 120 action classes performed by 106 volunteers, and has 32 different camera setups, each setup representing a specific location and background. Similarly, the dataset is separated into two benchmarks: (1) Cross-subject (X-Sub): $63,026$ samples performed by 53 subjects are separated into the training set, and the other $50,922$ samples performed by different 53 subjects belong to the test set. (2) Cross-setup (X-Set): the training set contains $54,471$ samples with even setup IDs, and the test set contains $59,477$ samples with odd setup IDs.

Kinetics Skeleton. Kinetics 400 [13] is a large-scale human action dataset that contains $300,000$ video clips of 400 classes collected from the Internet. After applying Openpose [2] pose-estimation algorithm on Kinetics 400, the Kinetics

Skeleton dataset obtain 240,436 training and 19,796 evaluation skeleton clips, where each skeleton graph contains 18 body joints, along with their 2D coordinates and confidence score.

4.2 Implementation Details

All experiments are implemented on two RTX 3090 GPUs with the PyTorch deep learning framework. The stochastic gradient descent (SGD) with the momentum of 0.9 and the weight decay of 0.0001 is used for optimization. The model is trained for 70 epochs in total. The initial learning rate is set to 0.1 and decays with a cosine schedule after the 10-th epoch. Moreover, a warm-up strategy [10] was applied over the first 10 epochs, gradually increasing the learning rate from 0 to the initial value in order to make the training procedure more stable. The batch size is set to 32. Input data are preprocessed following [32], and cross-entropy loss is employed.

4.3 Ablation Studies

We analyze the individual components and their configurations in the final architecture. The performance is reported as Top-1 and Top-5 classification accuracy on the Cross-Subject benchmark of NTU RGB+D 60 using only the joint data.

Table 1. Comparison of the accuracy when gradually adding STA-GC and only adding SA-GC or TA-GC on the X-Sub of NTU RGB+D 60.

Methods	Params	Top-1 (%)	Top-5 (%)
Baseline	1.44	88.1	98.2
+ 1 STA-GC	1.30	88.7	98.0
+ 2 STA-GC	1.36	89.1	98.4
+ 3 STA-GC	1.38	89.3	98.4
STA-GCN with SA-GC only	1.21	88.7	98.0
STA-GCN with TA-GC only	1.21	89.0	98.3
STA-GCN	1.40	**89.5**	**98.4**

Effectiveness of STA-GC. To verify the effectiveness of the proposed STA-GC block, we build up the model incrementally with its individual modules. We employ ST-GCN [40] as the baseline for controlled experiments. For a fair comparison, we add residual connections in ST-GCN and replace its temporal modeling module with temporal convolution described in Sect. 3.6. The experimental results are shown in Table 1. We first gradually add STA-GC into the baseline. For a fair comparison, we halved the original ten stages in the baseline to five after adding STA-GC to control for parameters, and this also alleviates the over-smoothing problem caused by adding the new graph convolution.

We observe that accuracies increase steadily, and the accuracy is substantially improved after each graph convolution has been added with the STA-GC module, which validates the effectiveness of STA-GC. Then we validate the effects of the SA-GC and the TA-GC respectively by adding either of them into the baseline. We observed performance raise of 1.4% and 0.4% respectively, indicating that our proposed SA-GC and TA-GC can effectively extract features in spatial and temporal dimension respectively. Moreover, the SA-GC and TA-GC are complementary and their combination can promote each other to achieve better performance for effective motion feature learning.

Table 2. Comparison of the accuracy when STA-GCN applies the fixed topology or the adaptive topology on the X-Sub of NTU RGB+D 60.

Spatial topology	Temporal topology	Top-1 (%)	Top-5 (%)
Fixed	Fixed	86.6	97.7
Adaptive	Fixed	88.0	97.8
Fixed	Adaptive	87.7	98.0
Adaptive	Adaptive	**89.5**	**98.4**

Effectiveness of TAE. To verify the effectiveness of our proposed TAE, we keep the backbone of the STA-GCN and apply the adaptive topologies generated by TAE or the fixed parameterized topologies in SA-GC and TA-GC respectively. As shown in Table 2, the models only using TAE in the spatial dimension or temporal dimension outperform the model using only the fixed topologies, and the model applying TAE both in spatial and temporal dimensions achieves the best results. It demonstrates that TAE can effectively generate appropriate spatial topology and temporal topology for robust feature learning.

Table 3. Comparison of the accuracy when STA-GCN applies different topology adaptive methods on the X-Sub of NTU RGB+D 60.

Methods	Top-1 (%)	Top-5 (%)
2s-AGCN [29]	88.9	97.9
Dynamic-GCN [41]	80.0	95.4
TAE	**89.5**	**98.4**

Comparison with Other Topology Adaptive Methods. To validate the effectiveness of our TAE, we also compare the performance of different topology adaptive methods in Table 3. Specifically, we keep the backbone of the STA-GCN and only replace the topology adaptive method in graph convolution for a fair comparison. From Table 3, we observe that TAE outperforms other topology adaptive methods from 2s-AGCN and Dynamic-GCN, proving that TAE is effective in generating adaptive typologies.

Table 4. Comparison of the accuracy when STA-GCN applies shared or unshared topology generation strategy on the X-Sub of NTU RGB+D 60.

Spatial topology	Temporal topology	Top-1 (%)	Top-5 (%)
Shared	Shared	88.6	98.1
Unshared	Shared	89.0	98.3
Shared	Unshared	88.8	98.2
Unshared	Unshared	**89.5**	**98.4**

Shared Topology vs Unshared Topology. We also verify the topology unshared strategy of TAE. Specifically, we keep the backbone of the STA-GCN and compare the model's performance when TAE uses shared or unshared topology generation strategies. The topology-unshared strategy means that TAE generates a specific topology for each skeleton or trajectory in each action sample, while the topology-shared strategy represents that TAE generates the same topology for all poses or trajectories in each action sample. As shown in Table 4, the topology-unshared strategy achieve better performance than the topology-shared strategy, indicating the importance of generating a specific topology for each skeleton or trajectory.

Table 5. Comparisons of the Top-1 accuracy (%) on actions with large changes in pose on the X-Sub of NTU RGB+D 60.

Methods	Actions				
	throw	stand up	hopping	pick up	falling down
ST-GCN [40]	88.3	96.8	97.1	92.8	97.1
STA-GCN	**91.0**	**98.5**	**98.6**	**95.2**	**99.6**

Performance for Recognizing Actions with Large Changes in Pose. To further verify that the proposed STA-GCN can recognize actions with large pose changes more effectively, we compare with ST-GCN on several actions. For a fair comparison, we added additional residual connections and applied the same temporal convolution module to the ST-GCN. The results are shown in Table 5, our method outperforms ST-GCN in Top1 accuracy on five poses (throw, stand up, hopping, pick up, and falling down), demonstrating that our approach can effectively identify actions with large changes in pose.

Table 6. Comparisons of the Top-1 accuracy with the state-of-the-art methods on the NTU RGB+D 60 and NTU RGB+D 120 datasets.

Methods	NTU RGB+D 60		NTU RGB+D 120	
	X-Sub (%)	X-View (%)	X-Sub (%)	X-Set (%)
ST-GCN [40]	81.5	88.3	–	–
AS-GCN [17]	86.8	94.2	–	–
2s-AGCN [29]	88.5	95.1	–	–
AGC-LSTM [30]	89.2	95.0	–	–
DGNN [28]	89.9	96.1	–	–
PL-GCN [11]	89.2	95.0	–	–
NAS-GCN [25]	89.4	95.7	–	–
SGN [44]	89.0	94.5	79.2	81.5
Shift-GCN [7]	90.7	96.5	85.9	87.6
MS-G3D [22]	91.5	96.2	86.9	88.4
DC-GCN+ADG [6]	90.8	96.6	86.5	88.1
PA-ResGCN-B19 [31]	90.9	96.0	87.3	88.3
Dynamic-GCN [41]	91.5	96.0	87.3	88.6
MST-GCN [5]	91.5	96.6	87.5	88.8
EfficientGCN-B4 [32]	92.1	96.1	88.7	88.9
CTR-GCN [4]	92.4	96.8	88.9	90.6
STA-GCN(J)	89.5	95.6	85.0	86.2
STA-GCN(B)	90.2	95.4	85.5	87.1
STA-GCN(JM)	88.3	94.3	82.9	84.0
STA-GCN(BM)	88.6	94.1	83.1	84.9
2s-STA-GCN(J, B)	91.6	96.2	88.1	89.6
3s-STA-GCN(J, B, JM)	92.7	96.9	89.2	90.6
4s-STA-GCN(J, B, JM, BM)	**92.8**	**97.0**	**89.4**	**90.8**

4.4 Comparisons with SOTA Methods

For fair comparisons, we follow the same multi-stream fusion strategy as [4,7,41]. Specifically, we use four modality streams, *i.e.*, joint stream (J), bone stream (B), joint motion stream (JM), and bone motion stream (BM). A simple score-level fusion strategy is adopted to obtain the fused score. The 1-stream model uses the individual stream of four modalities as input data. The 2-stream model fuses the joint and bone stream. The 3-stream model fuses the joint, bone, and joint motion stream. The 4-stream model fuses all four modality streams.

We compare our models with the state-of-the-art methods on NTU RGB+D 60, NTU RGB+D 120, and Kinetics Skeleton in Tables 6 and 7 respectively. On NTU RGB+D 60 and NTU RGB+D 120, our STA-GCN using three modality streams outperforms the previous state-of-the-art methods using four modality streams (*i.e.*, CTR-GCN [4] and MST-GCN [5]). Our final 4s-STA-GCN achieves

new state-of-the-art performance. On Kinetics Skeleton, our model with four streams fusion outperforms current state-of-the-art MST-GCN [5] by 2.1% and 2.7% on Top-1 and Top-5 accuracy respectively. All these experimental results demonstrate the superiority of the STA-GCN.

Table 7. Comparisons of the Top-1 and Top-5 accuracy with the state-of-the-art methods on the Kinetics dataset.

Methods	Kinetics skeleton	
	Top-1 (%)	Top-5 (%)
ST-GCN [40]	30.7	52.8
AS-GCN [17]	34.8	56.5
2s-AGCN [29]	36.1	58.7
DGNN [28]	36.9	59.6
NAS-GCN [25]	37.1	60.1
MS-G3D [22]	38.0	60.9
MST-GCN [5]	38.1	60.8
STA-GCN(J)	36.0	58.5
STA-GCN(B)	34.9	57.3
STA-GCN(JM)	33.1	56.4
STA-GCN(BM)	33.6	56.6
2s-STA-GCN(J, B)	38.5	61.5
3s-STA-GCN(J, B, JM)	40.0	63.0
4s-STA-GCN(J, B, JM, BM)	**40.2**	**63.5**

5 Conclusion

In this work, we present a Spatial-Temporal Adaptive Graph Convolutional Network (STA-GCN) to capture robust motion patterns for skeleton action recognition. The STA-GCN is composed of spatial adaptive graph convolution (SA-GC) and temporal adaptive graph convolution (TA-GC). The SA-GC module is designed to extract spatial features by modeling the spatial adaptive joint dependencies. The TA-GC module is designed to learn temporal features by capturing the direct long-range joint dependencies in the temporal dimension. Both SA-GC and TA-GC have a critical embedded component: Topology Adaptive Encoder (TAE). The TAE is adopted to generate spatial adaptive topology and temporal adaptive topology for learning the discriminative features. Extensive experimental results demonstrate the effectiveness of the proposed modules. On three large-scale skeleton action recognition datasets, the proposed STA-GCN achieves the state-of-the-art performance.

Acknowledgements. This work is supported by National Natural Science Foundation of China (Project No. 62076132) and Natural Science Foundation of Jiangsu (Project No. BK20211194).

References

1. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: a review. ACM Comput. Surv. (CSUR) **43**(3), 1–43 (2011)
2. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2D pose estimation using part affinity fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7291–7299 (2017)
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308 (2017)
4. Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., Hu, W.: Channel-wise topology refinement graph convolution for skeleton-based action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 13359–13368 (2021)
5. Chen, Z., Li, S., Yang, B., Li, Q., Liu, H.: Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 1113–1122 (2021)
6. Cheng, K., Zhang, Y., Cao, C., Shi, L., Cheng, J., Lu, H.: Decoupling GCN with DropGraph module for skeleton-based action recognition. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12369, pp. 536–553. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58586-0_32
7. Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., Lu, H.: Skeleton-based action recognition with shift graph convolutional network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 183–192 (2020)
8. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1110–1118 (2015)
9. Gao, X., Hu, W., Tang, J., Liu, J., Guo, Z.: Optimized skeleton-based action recognition via sparsified graph regression. In: Proceedings of the 27th ACM International Conference on Multimedia, pp. 601–610 (2019)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
11. Huang, L., Huang, Y., Ouyang, W., Wang, L.: Part-level graph convolutional network for skeleton-based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11045–11052 (2020)
12. Huang, Z., Shen, X., Tian, X., Li, H., Huang, J., Hua, X.S.: Spatio-temporal inception graph convolutional networks for skeleton-based action recognition. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 2122–2130 (2020)
13. Kay, W., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
14. Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F.: A new representation of skeleton sequences for 3d action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3288–3297 (2017)
15. Kim, T.S., Reiter, A.: Interpretable 3D human action analysis with temporal convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1623–1631. IEEE (2017)

16. Li, C., Zhong, Q., Xie, D., Pu, S.: Skeleton-based action recognition with convolutional neural networks. In: 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pp. 597–600. IEEE (2017)
17. Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., Tian, Q.: Actional-structural graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3595–3603 (2019)
18. Lin, J., Gan, C., Han, S.: TSM: temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7083–7093 (2019)
19. Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.Y., Kot, A.C.: NTU RGB+D 120: a large-scale benchmark for 3D human activity understanding. IEEE Trans. Pattern Anal. Mach. Intell. **42**(10), 2684–2701 (2019)
20. Liu, J., Wang, G., Hu, P., Duan, L.Y., Kot, A.C.: Global context-aware attention LSTM networks for 3D action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1647–1656 (2017)
21. Liu, M., Liu, H., Chen, C.: Enhanced skeleton visualization for view invariant human action recognition. Pattern Recogn. **68**, 346–362 (2017)
22. Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W.: Disentangling and unifying graph convolutions for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 143–152 (2020)
23. Obinata, Y., Yamamoto, T.: Temporal extension module for skeleton-based action recognition. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 534–540. IEEE (2021)
24. Ofli, F., Chaudhry, R., Kurillo, G., Vidal, R., Bajcsy, R.: Sequence of the most informative joints (SMIJ): a new representation for human skeletal action recognition. J. Visual Commun. Image Representation **25**(1), 24–38 (2014)
25. Peng, W., Hong, X., Chen, H., Zhao, G.: Learning graph convolutional network for skeleton-based human action recognition by neural searching. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 2669–2676 (2020)
26. Poppe, R.: A survey on vision-based human action recognition. Image Vision Comput. **28**(6), 976–990 (2010)
27. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: NTU RGB+D: a large scale dataset for 3D human activity analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1010–1019 (2016)
28. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Skeleton-based action recognition with directed graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7912–7921 (2019)
29. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12026–12035 (2019)
30. Si, C., Chen, W., Wang, W., Wang, L., Tan, T.: 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, pp. 1227–1236 (2019)
31. Song, Y.F., Zhang, Z., Shan, C., Wang, L.: Stronger, faster and more explainable: a graph convolutional baseline for skeleton-based action recognition. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 1625–1633 (2020)

32. Song, Y.F., Zhang, Z., Shan, C., Wang, L.: Constructing stronger and faster baselines for skeleton-based action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **45**(2), 1474–1488 (2022)
33. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5693–5703 (2019)
34. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4489–4497 (2015)
35. Vemulapalli, R., Arrate, F., Chellappa, R.: Human action recognition by representing 3D skeletons as points in a lie group. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 588–595 (2014)
36. Wang, L., et al.: Temporal segment networks for action recognition in videos. IEEE Trans. Pattern Anal. Mach. Intell. **41**(11), 2740–2755 (2018)
37. Wang, P., Li, W., Ogunbona, P., Wan, J., Escalera, S.: RGB-D-based human motion recognition with deep learning: a survey. Comput. Vis. Image Underst. **171**, 118–139 (2018)
38. Weinland, D., Ronfard, R., Boyer, E.: A survey of vision-based methods for action representation, segmentation and recognition. Comput. Vis. Image Underst. **115**(2), 224–241 (2011)
39. Xia, L., Chen, C.C., Aggarwal, J.K.: View invariant human action recognition using histograms of 3d joints. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 20–27. IEEE (2012)
40. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
41. Ye, F., Pu, S., Zhong, Q., Li, C., Xie, D., Tang, H.: Dynamic GCN: context-enriched topology learning for skeleton-based action recognition. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 55–63 (2020)
42. Zanfir, M., Leordeanu, M., Sminchisescu, C.: The moving pose: an efficient 3D kinematics descriptor for low-latency action recognition and detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2752–2759 (2013)
43. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2117–2126 (2017)
44. Zhang, P., Lan, C., Zeng, W., Xing, J., Xue, J., Zheng, N.: Semantics-guided neural networks for efficient skeleton-based human action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1112–1121 (2020)
45. Zhang, X., Xu, C., Tao, D.: Context aware graph convolution for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14333–14342 (2020)
46. Zhang, Z.: Microsoft kinect sensor and its effect. IEEE Multimedia **19**(2), 4–10 (2012)