Dynamic Semantic-Based Spatial Graph Convolution Network for Skeleton-Based Human Action Recognition

Jianyang Xie[1,2], **Yanda Meng**[2,6*], **Yitian Zhao**[4], **Anh Nguyen**[3], **Xiaoyun Yang**[5], **Yalin Zheng**[2,6 *]

[1]CDT in Distributed Algorithms, School of EEECS, University of Liverpool, UK
[2]Department of Eye and Vision Sciences, University of Liverpool, Liverpool, UK
[3]Department of Computer Sciences, University of Liverpool, Liverpool, UK
[4]Cixi Institute of Biomedical Engineering, Ningbo Institute of Materials Technology and Engineering, CAS, Cixi, China
[5]Remark AI UK Limited, London, UK
[6]Liverpool Centre for Cardiovascular Science, Liverpool, UK
yzheng@liverpool.ac.uk, myd@liverpool.ac.uk

Abstract

Graph convolutional networks (GCNs) have attracted great attention and achieved remarkable performance in skeleton-based action recognition. However, most of the previous works are designed to refine skeleton topology without considering the types of different joints and edges, making them infeasible to represent the semantic information. In this paper, we proposed a dynamic semantic-based graph convolution network (DS-GCN) for skeleton-based human action recognition, where the joints and edge types were encoded in the skeleton topology in an implicit way. Specifically, two semantic modules, the joints type-aware adaptive topology and the edge type-aware adaptive topology, were proposed. Combining proposed semantics modules with temporal convolution, a powerful framework named DS-GCN was developed for skeleton-based action recognition. Extensive experiments in two datasets, NTU-RGB+D and Kinetics-400 show that the proposed semantic modules were generalized enough to be utilized in various backbones for boosting recognition accuracy. Meanwhile, the proposed DS-GCN notably outperformed state-of-the-art methods. The code is released here https://github.com/davelailai/DS-GCN.

Introduction

Human action recognition (HAR) is an essential topic in computer vision and has a wide range of applications in video understanding (Gaur et al. 2011; Gui et al. 2018). Especially, skeleton-based action recognition has attracted much attention in the research community. Compared with RGB image squeeze (Carreira and Zisserman 2017; Bilen et al. 2017; Tran et al. 2015) or optical flows (Simonyan and Zisserman 2014; Wang and Schmid 2013), skeleton data (Yan, Xiong, and Lin 2018; Vemulapalli, Arrate, and Chellappa 2014) provided body pose and movement information directly, making it more robust to variations of camera viewpoint and video appearance. Meanwhile, low-cost depth sensors (Liu et al. 2019) (*e.g.*, Kinect) and availability of pose estimation algorithms (Sun et al. 2019) make the skeleton-based HAR can be extensively studied.

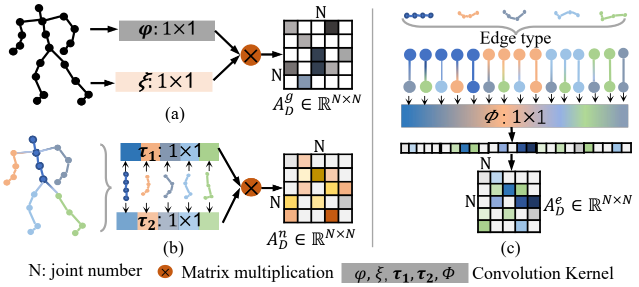

Figure 1: Illustration of node and edge type-aware adaptive graph generation. (a) represents the general adaptive graph generation, where the joint is considered as the same type, ϕ, and ξ is 1×1 convolution kernel. general adaptive graph $A_D^g \in \mathbb{R}^{N\times N}$ was generated based on the matrix multiplication. (b) represent node type-aware adaptive generation, human body is split into five parts with different colours, and node-type specific transform function τ_1 and τ_2 was designed, where the color is corresponding to the node type. For each part, the joints were projected into corresponding feature space, then the node type-aware adaptive graph $A_D^n \in \mathbb{R}^{N\times N}$ can be obtained based on the matrix multiplication. (c) represent the edge type-aware adaptive graph generation, the edge type is represented as the type pair of its end nodes. There are fifteen types of edges, and an edge-type specific transform function ϕ was designed and was utilized to transfer edge representations to their corresponding distribution space, thus the edge type-aware adaptive graph $A_D^e \in \mathbb{R}^{N\times N}$ can be obtained.

Early methods focus on extracting handcrafted features from skeleton sequences (Vemulapalli, Arrate, and Chellappa 2014; Wang et al. 2014). Recently, deep learning has become the mainstream research due to its strong feature learning ability, and various network structures have been investigated. For instance, recurrent neural networks (RNNs) (Du, Wang, and Wang 2015; Zhang, Liu, and Xiao 2017; Liu et al. 2017) have been applied to model the temporal information within the skeleton sequences, convolution neu-

[*] Corresponding Author

ral networks (CNNs) also have been adapted for HAR by representing the skeleton sequence as pseudo-images (Ke et al. 2017; Caetano et al. 2019; Duan et al. 2022c). Spatial-temporal graph convolution networks (ST-GCNs) have been proposed for working on the skeleton graph (Yan, Xiong, and Lin 2018; Si et al. 2018; Chen et al. 2021; Cheng et al. 2020b; Liu et al. 2020; Zhang et al. 2020; Shi et al. 2019b). Among these approaches, ST-GCNs have been the most popular one since they can capture inherent interaction between body joints through node aggregation scheme.

Yan (Yan, Xiong, and Lin 2018) first proposed the ST-GCN on predefined skeleton graphs. However, the fixed graph limited the representation of GCN and is inefficient in capturing the changeable human movement. In order to boost the flexibility of the model, some dynamic graph generation methods were proposed (Chen et al. 2021; Shi et al. 2019b; Cheng et al. 2020a) to learn an adaptive adjacent matrix. However, these works ignored the semantic information of the skeleton. They simply assumed all joints/edges as the same type, As shown in **Figure. 1 (a)**, making them insufficient to capture the semantic properties of actions. Intuitively, human actions involve movements of different body parts. For example, pointing to something mainly depend on swinging the arms but kicking forward indicates swinging legs. In this case, the types of moving nodes will be useful information for action recognition.

Zhang (Zhang et al. 2020) noticed this limitation and proposed a semantics-guided neural network to enrich the input joint feature by explicitly adding one-hot vectors of different node types. Although this method proved that the semantic information of joints type can boost performance, it faces several issues: Firstly, the explicit encoding in the input step is not flexible and cannot incorporate the high-order semantic information when GCNs go deeper. Secondly, the edge types were not considered. Because the connection in different types of joints might be various, even between the same type of joints but in different directions, the connection weight value might be different. Taking legs and arms as an example, the information passing from legs to arms should be different from that passing within arm joints. Meanwhile, within the arm, the information passing from elbow to wrist might be different and vice versa.

In light of these limitations, a dynamic semantic-based graph neural convolutions network (DS-GCN) was proposed in this paper. The main idea of the proposed work is to encode the dynamical semantic information of joints and edges in GCNs aggregation process implicitly. Specifically, a dynamic adaptive topology with semantic information on joints/edges types was generated. Instead of adding the predefined type encoding into the joint feature, the joint/edge type was encoded with different transform functions, each of which represents a specific distribution. Thus the feature of joint/edge in different types can be represented in their individual feature space. In other words, the types of joint/edge were encoded in an implicit way. Compared with the predefined encoding, there are two advantages of our proposed DS-GCN. On the one hand, since the semantic information of joints/edges was learned from the sample itself, the dynamic nature of each skeleton can be maintained. On the other hand, the joints/edges types were represented by the transform functions, and can be encoded in each ST-GCN layer. Thus, the semantic information can be reserved without over-smoothing even if the model goes deeper.

As shown in **Figure. 1 (b)**, the joints and edges were split into different types in advance. For the joint/edge type definition, the human body was decomposed into several parts (five parts in this paper, including left/right arms, left/right legs, and the trunk with the head) according to the natural human body structure, then the edge type can be obtained according to the type of its two end nodes, As shown in **Figure. 1 (c)**. For instance, the link between the left arm and trunk is different from the link within the trunk. Then two semantic-aware modules were proposed to encode the joint/edge types respectively, the node type-aware adaptive graph module and the edge type-aware adaptive graph module. In the node type-aware module, As shown in **Figure. 1 (b)**, a non-local mechanism was applied, but separate transform functions were designed for each body part to project the node representation in their specific type distribution, thus the adaptive graph can be generated with consideration of the node types. In the edge type-aware module, As shown in **Figure. 1 (c)**, similar to the node type encoding, the edge type-specific transform functions were designed, which were then applied to the adaptive skeleton graph to encode semantic information over each edge type.

Based on the two semantic modules and combining the temporal modeling modules proposed in (Duan et al. 2022a), the dynamic semantic-based graph neural networks (DS-GCN) was developed for skeleton-based human action recognition. The framework of the proposed method is as shown in **Figure. 2**. The extensive experiments on NTU-RGB+D (Shahroudy et al. 2016; Liu et al. 2019) and Kinetics-400 (Carreira and Zisserman 2017) show that: (1) the proposed two semantics modules are efficient and generalized to be adaptive to various ST-GCNs structure to boost the performance. (2) the generated DS-GCN outperforms state-of-the-art methods notably on all two datasets.

The main contributions are summarized as follows:

- We proposed to implicitly encode the joints and edge types for skeleton-based human action recognition. Two dynamical semantic-based adaptive graphs including a node type-aware adaptive graph, and an edge type-aware adaptive graph were generated. Extensive experiments show that the proposed semantic graph is very generalizable that can be easily adapted to various ST-GCNs.
- Generated a dynamic semantic-based graph neural network for skeleton-based human action recognition, and extensive experiments highlight that the proposed DS-GCN outperforms SOTA methods notably on both NTU-RGB+D and Kinetics-400.

Related Work

GCNs for Skeleton-Based Action Recognition

Graph convolution networks have attracted increasing attention in skeleton-based human action recognition (Yan, Xiong, and Lin 2018; Si et al. 2018; Chen et al. 2021; Cheng

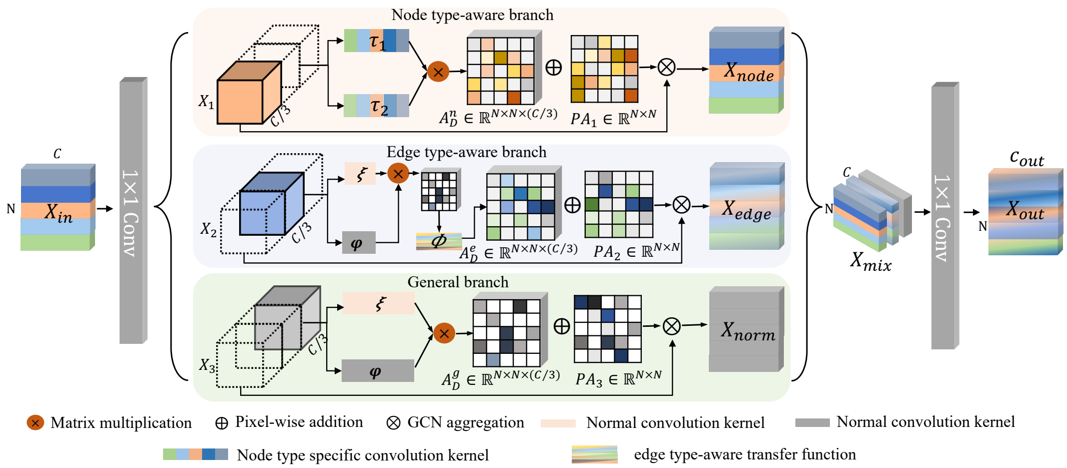
Node type-aware branch
Edge type-aware branch
General branch
C
N
X_{in}
1×1 Conv
X_1
X_2
X_3
C/3
τ_1
τ_2
ξ
φ
Φ
$A_D^n \in \mathbb{R}^{N\times N\times(C/3)}$
$PA_1 \in \mathbb{R}^{N\times N}$
$A_D^e \in \mathbb{R}^{N\times N\times(C/3)}$
$PA_2 \in \mathbb{R}^{N\times N}$
$A_D^g \in \mathbb{R}^{N\times N\times(C/3)}$
$PA_3 \in \mathbb{R}^{N\times N}$
X_{node}
X_{edge}
X_{norm}
X_{mix}
1×1 Conv
C_{out}
X_{out}
N
Matrix multiplication
Pixel-wise addition
GCN aggregation
Normal convolution kernel
Normal convolution kernel
Node type specific convolution kernel
edge type-aware transfer function

Figure 2: The framework of the proposed DS-GCN. The spatial graph convolution structure was decomposed into three branches, the node-type aware branch, the edge-type aware branch, and the general branch. C is the input channel, is the output channel. In each branch, the corresponding semantic self-adaptive graph and a shared correction matrix $PA_i \in \mathbb{R}^{N\times N}, i = 1, 2, 3$ were applied to represent the skeleton. The mixed output X_{mix} can then be obtained by concatenating the outputs of the three branches along the feature channel dimension. The final output X_{out} can be calculated by a 1×1 Conv function on X_{mix}.

et al. 2020b; Liu et al. 2020; Zhang et al. 2020; Shi et al. 2019b; Duan et al. 2022a). Yan . (Yan, Xiong, and Lin 2018) introduced a pre-defined skeleton graph according to the human body's natural link and proposed the ST-GCN to capture the spatial and temporal patterns from the graph structure. Upon this baseline, some spatial adaptive graph generation methods based on no-local mechanisms were proposed to increase the flexibility of the skeleton graph structure (Shi et al. 2019b; Chen et al. 2021; Cheng et al. 2020b; Zhang et al. 2020; Duan et al. 2022a). Instead of only applying the fixed graph structure, these methods learned another adaptive graph to boost the GCNs' representation ability. For instance, the 2S-AGCN (Shi et al. 2019b) learned a data-driven graph for all feature channels, and CTR-GCN (Chen et al. 2021) learned an adaptive graph for each individual feature channel. Meanwhile, the multi-scale and shift GCN were proposed (Cheng et al. 2020b; Liu et al. 2020) to release the over-smooth problem in graph long-distance transfer. In the temporal pattern, multi-scale temporal convolution was proposed in (Chen et al. 2021; Duan et al. 2022a) to boost the information aggregation in temporal space.

Semantic Information Exploration

The semantic information has been exploited in RNNs for skeleton-based human action recognition (Du, Wang, and Wang 2015; Wang and Wang 2017; Si et al. 2018). In these methods, the skeleton structure is manually partitioned into different functional parts, and processed by the individual RNN. As the network goes deeper, the feature of different components is concatenated and progressed in a hierarchical way. Even though such information is important, in most GCNs for skeleton-based human action recognition, the semantic information is overlooked. Inspired by this, Zhang (Zhang et al. 2020) proposed the SGN to encode the information of joint types in the initial feature by explicitly adding one-hot vectors for node types representation. However, this pre-defined semantics encoding in the input layer is not flexible and cannot represent such information in high-dimension space when networks go deeper. To tackle the above limitations, we proposed a more elegant method to encode the semantics implicitly.

Methods

In this section, The notation of ST-GCN will be introduced first, and then the ST-GCN with its variants are formulated and discussed. Finally, the proposed DS-GCN will be described in detail.

Preliminaries

Notation. A skeleton data is denoted as a spatial-temporal graph $G = (V, E_s, E_t, X)$ where $V = \{v_{ti}|t = 1, ..., T, i = 1, ..., N\}$ as the N body joints in T frames, E_s and E_t as the spatial and temporal link respectively. $X \in \mathbb{R}^{N\times T\times d}$ represents the joint coordinates as the node feature, where d is the feature dimension. For the spatial graph $G_s = (V, E_s, X)$, E_s is formulated as an adjacent matrix $A \in \mathbb{R}^{N\times N}$ to represent the intro-body connection. For the temporal graph $G_t = (V, E_t, X)$, E_t is constructed by connecting the same joints along consecutive frames. Then the ST-GCNs can be divided into two parts: the Spatial-GCN (S-GCN) with regular GCN and the Temporal-GCN (T-GCN) with $1D$ temporal convolution. The proposed method is adapted to the S-GCN.

Topology-Fixed Graph Convolution Network. The main operation of GCN is to update the node representation by aggregating information from its neighborhood. In ST-GCN (Yan, Xiong, and Lin 2018) A is defined as three partitions and represented as $A \in \mathbb{R}^{N\times N\times 3}$. Denoting $X = \{X_t \in \mathbb{R}^{N\times d}|t = 1, ...T\}$ as input feature, the output $X^{'} = \{X_t^{'} \in \mathbb{R}^{N\times C} | t = 1, ...T\}$ of S-GCN can be formulated as Eq. 1.

$$X^{'} = \sum_{i=1}^{3} f(A^i X, \theta), \tag{1}$$

where f is the updating function, which is a 2D convolution network with kernel size 1 normally, θ is the learnable parameters of the updating function, and C is the number of the output feature channel.

Topology-Adaptive Graph Convolution Network. In most ST-GCN variants (Yan, Xiong, and Lin 2018; Si et al. 2018; Chen et al. 2021; Cheng et al. 2020b; Liu et al. 2020; Zhang et al. 2020; Shi et al. 2019b; Duan et al. 2022a), the adaptive matrix A_D was dynamically learned with self-attention mechanism. As shown in **Figure 3 (a)**, supposing two transformation functions $\varphi(\cdot)$ and $\xi(\cdot)$, the correlation between two joints can be modeled as Eq. 2.

$$A_D = \sigma(\varphi(X) - \xi(X)), \tag{2}$$

where $\sigma(\cdot)$ represents the activate function in use, such as $Relu$. The adaptive S-GCN can be represented as Eq. 3.

$$X^{'} = \sum_{i=1}^{3} f((A^i + \lambda A_D^i)X, \theta), \tag{3}$$

where λ is the predefined or learnable weight to refine the effect of the adaptive graph. The adaptive graph has proved to be an advantageous topology for skeleton-based human action recognition (Shi et al. 2019b; Chen et al. 2021).

Semantic-Guided Graph Convolution Network. In explicit semantic encoding method (Zhang et al. 2020), the input feature was refined by adding a one-hot vector of joint types, which can be formulated as Eq. 4

$$X = \{[X_t, X_{t,k}] \in \mathbb{R}^{N\times c} | t = 1, ...T, k = 1, ..., m\} \tag{4}$$

where m is the joint type number, c is the modified feature channels, $X_{t,k}$ is the corresponding type encoding. The topology-adaptive graph convolution network is then worked on this input.

Dynamic Semantic-Based GCN

The general frame of the proposed DS-GCN origins from the topology-adaptive GCN, however, different to the above methods, the joint and edge types in the skeleton graph were introduced and encoded dynamically when calculating the adaptive graph. Specifically, the DS-GCN contains two modules: the node type-aware topology and the edge type-aware topology. As shown in **Figure 3**, when modeling the node type-aware adaptive graph, different conversion functions were defined for different types of nodes, and then the node type-aware topology can be obtained by capturing pair-wise joint correlations. In this paper, the non-local mechanism was applied in a channel-wise manner (Chen et al. 2021). In the edge type-aware correction graph, different update functions for edges of different types were applied to the adaptive graph. In this case, the graph in our work can be defined as a directed graph $G = (V, E, A, R, X)$, where A and R denote the type mapping function for each node: $\tau(v) = \{\tau_1(v), \tau_2(v)\} : V \to A$ and edge $\phi(e) : E \to R$ respectively. Supposing the input feature $X \in \mathbb{R}^{N\times d}$, the semantic-based adaptive graph is calculated as Eq. 5

$$\begin{aligned} A_D^n &= \sigma(\tau_1(X) - \tau_2(X)), \\ A_D^e &= \phi(A_D), \end{aligned} \tag{5}$$

where A_D^n represents the node type-aware graph and A_D^e represents the edge type-aware graph. The details of each of them are introduced as follows.

Node Type-Aware Adaptive Topology. As shown in **Figure. 3 (b)**, the node features were first projected into their individual feature space with a node type mapping function: $\tau(v)$, then the node type-aware adaptive graph can calculate according to the non-local mechanism. Specifically, denoting s and t as two nodes of different types, $x_s \in \mathbb{R}^{1\times d}$ and $x_t \in \mathbb{R}^{1\times d}$ as the corresponding feature, then the node-aware feature representation was formulated as Eq. 6

$$\begin{aligned} x_{s1}^{'} = \tau_1^s(x_s), x_{s2}^{'} = \tau_2^s(x_s) \\ x_{t1}^{'} = \tau_1^t(x_t), x_{t2}^{'} = \tau_2^t(x_t) \end{aligned} \tag{6}$$

where $x_*^{'} \in \mathbb{R}^{1\times C}$, C is the output feature channels. Supposing $\tau_1(v)$ as the source feature projection, $\tau_2(v)$ as the target feature projection, the directed correction between node s and t along channel dimension can be calculated as Eq. 7

$$A_D^{s\to t} = \sigma(x_{s1}^{'} - x_{t2}^{'}), A_D^{t\to s} = \sigma(x_{t1}^{'} - x_{s2}^{'}) \tag{7}$$

where σ is the activation function. $A_D^* \in \mathbb{R}^{1\times C}$. For the whole skeleton structure, the node aware-adaptive graph $A_D^n \in \mathbb{R}^{N\times N\times C}$ can be represented as the set of A_D^*.

Edge Type-Aware Adaptive Topology. As shown in **Figure. 3 (c)**, the edge type was encoded by applying separate convolution kernel $\phi(e)$ on the adaptive graph. Specifically, Given three nodes s, t and u of different types, the edge-type link between these nodes can be represented as $\langle s, t\rangle$, $\langle s, u\rangle$ and $\langle t, u\rangle$ with the feature $e_{\langle s,t\rangle}$, $e_{\langle s,u\rangle}$ and $e_{\langle t,u\rangle}$. Thus, the edge type-aware adaptive correlation can be refined as Eq. 8

$$\begin{aligned} A_D^{\langle s,t\rangle} &= \phi^{\langle s,t\rangle}(e_{\langle s,t\rangle}) \\ A_D^{\langle s,u\rangle} &= \phi^{\langle s,u\rangle}(e_{\langle s,u\rangle}) \\ A_D^{\langle t,u\rangle} &= \phi^{\langle t,u\rangle}(e_{\langle t,u\rangle}) \end{aligned} \tag{8}$$

where $\phi^{\langle *,*\rangle}(e)$ represent separate transform functions. Here the 2D convolution kernels with kernel size equal to 1 were applied. The edge type-aware topology can be represented as $A_D^e = \{A_{D_{ij}}^{\langle s,t\rangle} | i, j = 1, ..., N, s, t = 1, ..., M\}$, where s and t is the node type index respectively, M is the number of types.

Dynamic Semantic-Based GCN: As shown in **Figure.2**, Different from the previous ST-GCNs which utilized the same spatial graph convolution structure on three pre-generated skeleton graphs, in DS-GCN, the spatial graph convolution structure was decomposed into three branches, the node-type aware branch, the edge-type aware branch,

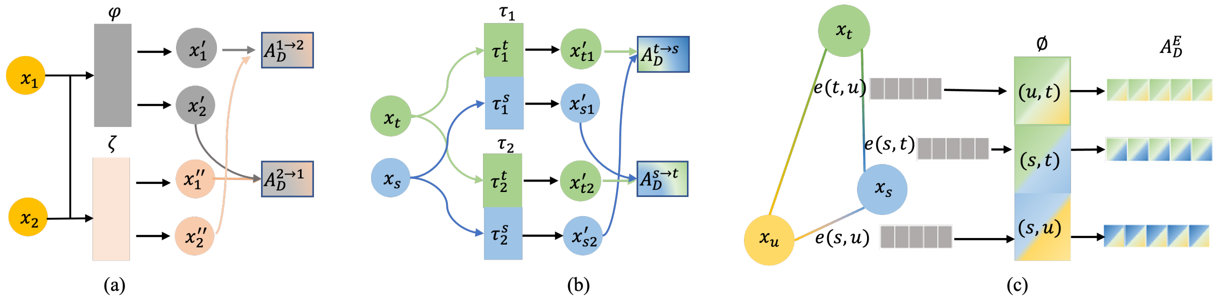

Figure 3: Illustration of the joint correlation calculation. (a) represents the standard non-local mechanism, for each transform function $\varphi(\cdot)$ and $\xi(\cdot)$, the node features are updated by sharing the same parameters. (b) represents the node type-aware correction. In each transform function, the convolution kernels are divided into several parts, each of which corresponds to a specific node type, and then the node characteristics in different types were updated by their individual parameters set. The colored circles denote different node types and the colored squares denote different convolution kernels. (c) illustrates the edge type-aware correlation. For each type of edge, specific convolution kernels were designed and utilized for updating the edge feature. The colored circles denote node types; mix-colored squares denote corresponding edges with node pairs.

and the general branch. A branch-wise weight is set as learnable and utilized for the combination of a shared correction matrix and the corresponding self-adaptive graph. Specifically, the input was first projected into a high dimension, which was split into three parts corresponding to different branched. For each branch, the combination of a shared correction matrix and a self-adaptive graph was utilized for spatial graph convolution operation. To balance the influence of the shared skeleton for each branch for action recognition, the pre-defined skeleton graph was replaced by a totally learnable correction matrix. Finally, the three branches were concatenated along the feature channel dimension and followed by a 1×1 convolution kernel, so that combines the information of the three branches and projects it into the output dimension. The process of the DS-GCN can be formulated as Eq. 9.

$$\begin{aligned} X^{'} &= f(x, \theta) \\ x &= [x^n, x^e, x^g] \in \mathbb{R}^{N \times 3c} \\ x^n &= (A^1 + \lambda_1 A_D^n) f_{pre}^1(X) \\ x^e &= (A^2 + \lambda_2 A_D^e) f_{pre}^2(X) \\ x^g &= (A^3 + \lambda_3 A_D) f_{pre}^3(X) \end{aligned} \tag{9}$$

where $X \in \mathbb{R}^{N \times C}$, f_{pre}^* is the projection function to reduce the feature channels; c is the output channels of the f_{pre}^*. x^n, x^e, x^g are the output of the node type-aware, edge type-aware, and general branch respectively. A^* is the learnable correlation matrix of each branch. λ_* is the learnable weight to refine the effect of each semantic-based topology-adaptive graph, which is different between branches.

Model Architecture

Based on the DS-GCN, a new spatial-temporal graph convolution network was introduced. Similar to ST-GCN (Si et al. 2018), ten basic blocks were connected in series, followed by a global average pooling and a softmax classifier for action classification. The number of basic feature channels is set as 64 and was doubled at 5_{th} and 8_{th} blocks. In each basic block, one DS-GCN and a multi-scale temporal modeling module proposed in (Chen et al. 2021) were contained.

Experiments

Datasets

To demonstrate the advantage of the proposed DS-GCN, two datasets were utilized in this paper: NTU RGB+D and Kinetics-400. The brief introduction is as follows and more details of these 2 datasets are in **Supplementary 1**.

NTU RGB+D. NTU RGB+D (Shahroudy et al. 2016; Liu et al. 2019) is a large-scale action recognition dataset. Here, four benchmarks recommended by the official are utilized: (1) NTU60 cross-subject (NTU60-Xsub), (2) NTU60 cross-view (NTU60-Xview), (3) NTU120 cross-subject (NTU120-Xsub), NTU120 cross-setup (NTU120-Xset).

Kinetics-400 (Carreira and Zisserman 2017). Kinetics-400 is a large-scale action recognition dataset with 400 actions. The skeletons utilized in this paper were provided by (Duan et al. 2022b), where the OpenPose algorithm (Cao et al. 2017) was applied for joint estimation.

Implementations Details

All experiments are conducted on one A100 GPU with the PyTorch deep learning framework. All models are trained with the SGD optimizer with momentum 0.9 and weight decay $5e^{-4}$. The initial learning rate was set to 0.1, and the model is trained for 100 epochs with the Cosine Annealing learning rate scheduler. The batch size was set to 128. To accelerate the training process, the input of temporal length was set to 64 in the ablation study. For a fair comparison, the input of temporal length was set to 100 when compared with SOTAs. The pre-processing approach follows the settings detailed in (Duan et al. 2022b).

Ablation Study

In this section, the proposed two semantic-based adaptive graph modules were analyzed on two benchmarks: NTU60-

Method	NTU60-XSub	NTU120-XSet
2s-GCN (Shi et al. 2019b)	89.5	86.0
2s-GCN+NE	**90.1**	**86.1**
CTR-GCN (Chen et al. 2021)	89.6	86.0
CTR-GCN+NE	**90.4**	**86.5**

Table 1: Generalization of the proposed semantic module. $+NE$ represents that the adaptive graph utilized in spatial-GCN is replaced by the semantic-based adaptive graph.

Method	NTU60-XSub	NTU120-XSet
ST-GCN (Si et al. 2018)	87.8	85.5
2s-GCN (Shi et al. 2019b)	89.5	86.0
CTR-GCN (Chen et al. 2021)	89.6	86.0
DS-GCN	**90.8**	**87.2**

Table 2: Effectiveness of DS-GCN. The proposed DS-GCN can achieve the best performance.

Method	NTU60-XSub	NTU120-XSet
DS-GCN$_{shared}$	90.1	86.8
DS-GCN$_{\text{B-wise}}$	**90.8**	**87.2**

Table 3: Comparison DS-GCN in different learnable weight manners. DS-GCN$_{\text{shared}}$ represents the DS-GCN with shared λ for all the branches, DS-GCN$_{\text{B-wise}}$ represent the DS-GCN with individual λ for different branches.

Method	NTU60-XSub
DS-GCN w/o N&E	90.0
DS-GCN w/o N	90.5
DS-GCN w/o E	90.4
DS-GCN	**90.8**

Table 4: Ablation On the edge/node type encoding. N represents the node type-aware encoding, and E represents the edge type-aware encoding. w/o means without, representing that the corresponding semantic encoding is replaced with the general branch.

Module	Encode stage	NTU60-XSub
DS-GCN w/o N&E	-	90.0
DS-GCN$_{ini}$	[1-4]	90.2
DS-GCN$_{mid}$	[5-7]	90.7
DS-GCN$_{end}$	[8-10]	90.5
DS-GCN	[1-10]	**90.8**

Table 5: Exploration on the semantic encoding stage. DS-GCN w/o N&E represents no semantic module is utilized, DS-GCN$_{ini}$ represents just utilized DS-GCN in layer [1-4], DS-GCN$_{mid}$ represents just utilized DS-GCN in layer [5-7], DS-GCN$_{end}$ represents just utilized DS-GCN in layer [8-10], DS-GCN represents DS-GCN is utilized in all the layers.

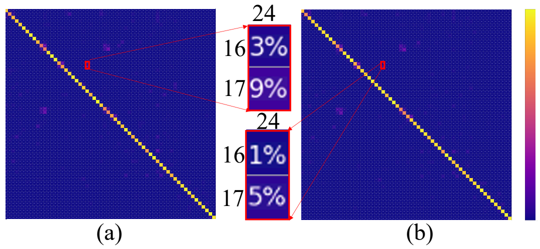

Figure 4: Visualization of classification. The action index is as follows: taking on shoes (16), taking off shoes (17), and kicking something (24). (a) the confusion matrix for CTR-GCN, (b) the confusion matrix for CTR-GCN+NE. It can be observed that, after encoding the semantic information, CTR-GCN+NE can distinguish kicking something from taking on/off shoes more accurately (e.g., errors reduce from 3% and 9% to 1% and 5% respectively).

XSub and NTU120-XSet. The joints coordinate were utilized as input and three learnable correlation matrices were randomly initialized for skeleton topology modeling.

Generalization Of The Semantic Encoding Modules: In order to justify the generalization and efficiency of proposed node/edge type-aware adaptive graph modules, several famous topology-adaptive topology ST-GCNs structures were utilized as the backbone, and the node/edge type-aware adaptive graph modules were adapted and utilized to replace the Spatial-GCN in these backbones. Here the 2s-GCN (Shi et al. 2019b), CTR-GCN (Chen et al. 2021) were utilized. For a fair comparison, the characteristic of the initial backbone was kept where three branches share the same structure. The node/edge type-aware adaptive graph modules were combined in series and then utilized as the Spatial-GCN in these backbones. The detail of structure is introduced in **Supplementary 2**. The results are shown in Table 1, It can be observed that, after encoding the node/edge types in these backbones, the accuracy of action recognition can have a stable increase.

To analyze the classification performance in more detail. The confusion metrics of CTR-GCN and CTR-GCN+NE on NTU60-XSub were generated as shown in **Figure 4**. Taking the action of kicking something (index 24 in the confusion matrix) and the action of taking on /off shoes (index 16/17 in the confusion matrix) for example, these actions can be described as the relative movement between two parts of the body, the Taking on/off shoes can be interpreted as the relative movement between arms and legs, but kicking something is the relative movement between two legs. It can be observed that after encoding the semantic information, CTR-GCN+NE can distinguish the action of kicking something from the action of taking on/off shoes more accurately.

Effectiveness of DS-GCN: In order to validate the effectiveness of the proposed dynamic semantic-based graph convolution, we compared the performance of the DS-GCN with several ST-GCN variants. Vanilla ST-GCN (Si et al. 2018), 2s-GCN (Shi et al. 2019b) and CTR-GCN (Chen et al. 2021) were utilized as the backbones in this experi-

Module	NTU60-Xsub	NTU60-Xview	NTU120-Xsub	NTU120-Xset	Kinetics-400
ST-GCN (Si et al. 2018)	81.5	88.3	70.7	73.2	30.7
SGN (Zhang et al. 2020)	86.6	93.4	-	-	-
AS-GCN (Li et al. 2019)	86.8	94.2	78.3	79.8	34.8
RA-GCN (Song et al. 2020)	87.3	93.6	78.3	79.8	34.8
2s-GCN (Shi et al. 2019b)	88.5	95.1	-	-	-
DGNN (Shi et al. 2019a)	89.9	96.1	-	-	-
FGCN (Yang et al. 2021)	90.2	96.3	85.4	87.4	-
ShiftGCN (Cheng et al. 2020b)	90.7	96.5	85.9	87.6	-
DSTA-Net (Shi et al. 2020a)	91.5	96.4	86.6	89.0	-
MS-G3D (Liu et al. 2020)	91.5	96.2	86.9	88.4	38.0
CTR-GCN (Chen et al. 2021)	92.4	96.8	88.9	90.6	-
ST-GCN++ (Duan et al. 2022b)	92.6	97.4	88.6	90.8	49.1
PoseConv3D (Duan et al. 2022c)*	**94.1**	97.1	86.9	90.3	47.7
DS-GCN	93.1	**97.5**	**89.2**	**91.1**	**50.6**

Table 6: Classification accuracy comparison against state-of-the-art methods. $*$ represents the SOTA CNN-based method.

ment. For all the models, the joints coordinate was set as the input, and the pre-defined graph was set as the totally trainable adjacency matrix. The results in Table 2 show that the topology-adaptive graph convolution network (2s-GCN, CTR-GCN, and DS-GCN) achieves better performance than the topology-fixed graph convolution network (ST-GCN). Compared with the CTR-GCN (Chen et al. 2021), the proposed DS-GCN has a smaller number of parameters but achieved a 1.2% $\mathrm{Top1\text{-}acc}$ increase in NTU60 Xsub and NTU120 Xset. The comparison of model size can be seen in **Supplementary 3**. This proves that the proposed DS-GCN is more effective in modeling the skeleton topology.

Configuration Exploration. In this section, the learnable weight λ is analyzed. Different to utilize one shared λ in other topology-adaptive graph learning, the individual refinement weight is learned for each branch in DS-GCN. To justify the branch-wise λ, we trained the DS-GCN in two ways: with the shared λ and with the individual λ in a branch-wise manner. The result is shown in Table 3, where it can be seen that the DS-GCN learned in a branch-wise weight manner has a stable improvement.

Ablation On The Edge/Node Type Encoding: In this section, the effectiveness of different configurations of DS-GCN was explored. In practice, to test the effects of node-type encoding, we replaced the node type-aware adaptive branch with the general branch, in this case, two branches were utilized to model the general adaptive graph, and one branch was utilized to model the edge type-aware adaptive graph. Similarly, the edge-type adaptive branch was replaced by the general branch to validate the effect of edge-type encoding. In Table 4, we can observe that the node/edge type-aware adaptive graph has a positive effect on the recognition performance, and combining both semantic branches can achieve the best performance. $\mathrm{Top1\text{-}acc}$ of the DS-GC outperforms the backbone with no semantic encoding by 0.8%.

Exploration Of The Semantic Encoding Stage. In practice, there are ten basic blocks in ST-GCN, as we described above that the proposed semantic encoding module is flexible that can be applied in different depths of the ST-GCN. Thus in order to explore the importance of semantic information encoding in various depths, the DS-GCN was utilized in different stages alone for comparison. Specifically, we split the whole DS-GCN into three stages: the initial stage represented as $\mathrm{DS\text{-}GCN}_{ini}$, which contains the layer from 1_{st} to 4_{th}, the middle stage $\mathrm{DS\text{-}GCN}_{mid}$ with layer 5_{th}-7_{th}, and the end stage $\mathrm{DS\text{-}GCN}_{end}$ with 8_{th}-10_{th}, then the DS-GCN was applied in each stage respectively. For instance, to justify the effect of semantic information on the initial stage, the DS-GCN is only utilized in layer 1_{st} to 4_{th}, in the rest block, all semantic-based modules are replaced by the general adaptive branch. The results in Table. 5 show that semantic encoding has a positive effect on human action recognition irrespective of the stage where the DS-GCN was used. When utilizing the DS-GCN in all layers, the model shows the best performance. Comparing within three stages, the middle stage outperforms the others, which can be explained as the over-smoothing problems. When the layer goes deeper, the semantic information encoded in the initial stage might be over-smoothed during the aggregation process. If only encoding the semantic information in the end stage, the feature of the node was already over-smoothed after the former stages' aggregation. Thus the correlation matrix plays weakly effect on feature updating, which limits the ability of the semantic encoding module.

Comparisons With the State-of-the-Art

Multi-stream fusion proposed in (Shi et al. 2020b) has been proven to be advanced for skeleton-based action recognition and has been adapted in many state-of-the-art methods (Chen et al. 2021; Duan et al. 2022a; Shi et al. 2019b). Thus, for a fair comparison, the DS-GCN was trained on four modalities respectively, the result for each modality was reported in **Supplementary 4**, and the final result was obtained by summering the probability from each stream. The performance of the DS-GCN was compared with SOTA methods on NTURGB+D 60 (120) and Kinetics 400 in Table 6. It can be observed that the proposed DS-GCN outperforms all existing methods.

Acknowledgments

This work was supported by the EPSRC (Engineering and Physical Sciences Research Council) Centre for Doctoral Training in Distributed Algorithms [Grant Ref: EP/S023445/1]. This work was supported in part by the National Natural Science Foundation of China (62272444); Zhejiang Provincial Natural Science Foundation (LR22F020008); and Youth Innovation Promotion Association CAS (2021298).

References

Bilen, H.; Fernando, B.; Gavves, E.; and Vedaldi, A. 2017. Action recognition with dynamic image networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(12): 2799–2813.

Caetano, C.; Sena, J.; Brémond, F.; Dos Santos, J. A.; and Schwartz, W. R. 2019. Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. In *2019 16th IEEE international conference on advanced video and signal based surveillance (AVSS)*, 1–8. IEEE.

Cao, Z.; Simon, T.; Wei, S.-E.; and Sheikh, Y. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7291–7299.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6299–6308.

Chen, Y.; Zhang, Z.; Yuan, C.; Li, B.; Deng, Y.; and Hu, W. 2021. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF ICCV*, 13359–13368.

Cheng, K.; Zhang, Y.; Cao, C.; Shi, L.; Cheng, J.; and Lu, H. 2020a. Decoupling gcn with drop graph module for skeleton-based action recognition. In *European Conference on Computer Vision*, 536–553. Springer.

Cheng, K.; Zhang, Y.; He, X.; Chen, W.; Cheng, J.; and Lu, H. 2020b. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 183–192.

Du, Y.; Wang, W.; and Wang, L. 2015. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1110–1118.

Duan, H.; Wang, J.; Chen, K.; and Lin, D. 2022a. DG-STGCN: Dynamic Spatial-Temporal Modeling for Skeleton-based Action Recognition.

Duan, H.; Wang, J.; Chen, K.; and Lin, D. 2022b. PYSKL: Towards Good Practices for Skeleton Action Recognition.

Duan, H.; Zhao, Y.; Chen, K.; Lin, D.; and Dai, B. 2022c. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2969–2978.

Gaur, U.; Zhu, Y.; Song, B.; and Roy-Chowdhury, A. 2011. A "string of feature graphs" model for recognition of complex activities in natural videos. In *2011 International Conference on Computer Vision*, 2595–2602.

Gui, L.-Y.; Zhang, K.; Wang, Y.-X.; Liang, X.; Moura, J. M. F.; and Veloso, M. 2018. Teaching Robots to Predict Human Motion. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 562–567.

Ke, Q.; Bennamoun, M.; An, S.; Sohel, F.; and Boussaid, F. 2017. A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3288–3297.

Li, M.; Chen, S.; Chen, X.; Zhang, Y.; Wang, Y.; and Tian, Q. 2019. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3595–3603.

Liu, J.; Shahroudy, A.; Perez, M.; Wang, G.; Duan, L.-Y.; and Kot, A. C. 2019. 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.

Liu, J.; Wang, G.; Duan, L.-Y.; Abdiyeva, K.; and Kot, A. C. 2017. Skeleton-based human action recognition with global context-aware attention LSTM networks. *IEEE Transactions on Image Processing*, 27(4): 1586–1599.

Liu, Z.; Zhang, H.; Chen, Z.; Wang, Z.; and Ouyang, W. 2020. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 143–152.

Shahroudy, A.; Liu, J.; Ng, T.-T.; and Wang, G. 2016. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1010–1019.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2019a. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7912–7921.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2019b. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12026–12035.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2020a. Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition. In *Proceedings of the Asian Conference on Computer Vision*.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2020b. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *IEEE Transactions on Image Processing*, 29: 9532–9545.

Si, C.; Jing, Y.; Wang, W.; Wang, L.; and Tan, T. 2018. Skeleton-based action recognition with spatial reasoning and temporal stack learning. In *Proceedings of the European conference on computer vision (ECCV)*, 103–118.

Simonyan, K.; and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27.

Song, Y.-F.; Zhang, Z.; Shan, C.; and Wang, L. 2020. Richly activated graph convolutional network for robust skeleton-based action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(5): 1915–1925.

Sun, K.; Xiao, B.; Liu, D.; and Wang, J. 2019. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5693–5703.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 4489–4497.

Vemulapalli, R.; Arrate, F.; and Chellappa, R. 2014. 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*, 588–595.

Wang, H.; and Schmid, C. 2013. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, 3551–3558.

Wang, H.; and Wang, L. 2017. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 499–508.

Wang, J.; Liu, Z.; Wu, Y.; and Yuan, J. 2014. Learning Actionlet Ensemble for 3D Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5): 914–927.

Yan, S.; Xiong, Y.; and Lin, D. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*.

Yang, H.; Yan, D.; Zhang, L.; Sun, Y.; Li, D.; and Maybank, S. J. 2021. Feedback graph convolutional network for skeleton-based action recognition. *IEEE Transactions on Image Processing*, 31: 164–175.

Zhang, P.; Lan, C.; Zeng, W.; Xing, J.; Xue, J.; and Zheng, N. 2020. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1112–1121.

Zhang, S.; Liu, X.; and Xiao, J. 2017. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 148–157. IEEE.