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

Jianyang Xie, Yanda Meng, Yitian Zhao, Nguyen Anh, Xiaoyun Yang, Yalin Zheng*

***Abstract*—Human action recognition is an essential topic in computer vision and image processing. Graph convolutional networks (GCNs) have attracted significant attention and achieved noteworthy performance in skeleton-based human action recognition tasks. However, most of the previous graph-based works are designed to refine skeleton topology without considering the types of different joints and edges and the occurrence order of the frames. Such a limitation makes them insufficient to represent intrinsic semantic information. Differently, we proposed a dynamic semantic-based spatial-temporal graph convolution network (DS-STGCN) to address the challenge. DS-STGCN has two dynamic semantic modules for spatial and temporal contexts respectively. Specifically, the joints and edge types were encoded in the spatial module implicitly, and the occurrence order of frames was encoded in the temporal module implicitly. Extensive experiments on four datasets including NTU-RGB+D 60(120), Kinetics-400, and FineGYM show that our proposed two semantic modules can bring consistent recognition performance improvement with various backbones. Meanwhile, the proposed DS-STGCN notably surpassed state-of-the-art methods on these datasets. Notably, in the more challenging dataset, such as Kinetics-400, our model significantly outperformed other state-of-the-art GCN-based methods by a large margin. The code will be released upon acceptance.**

***Index Terms*—Human action recognition, Skeleton-based, Semantics encoding, Joints/edge type, Frames occurrence order, Graph convolution network**

I. Introduction

HUMAN action recognition (HAR) is an essential topic in computer vision and has a wide range of applications, such as intelligent surveillance system [1] and human-computer interaction [2]. Recently, skeleton-based action recognition has attracted much increased attention in the research community. Different from image-based recognition methods where RGB image sequence [3]–[6] or optical flows [7], [8] were utilized as the model input, they use the skeleton data extracted from images or videos [9]–[11], which represent the human body with joints and bones. Such topological representation explicitly provided body pose and movement information, making it more robust to the variations of camera viewpoint and video appearance. Meanwhile, low-cost depth sensors such as Microsoft Kinect [12] and increasing availability of powerful pose estimation algorithms [13] make the skeleton-based HAR extensively studied.

In the field of HAR, a skeleton represents human body as a set of coordinates of body joints. The motion patterns are extracted from a certain skeleton sequence for action classification. Traditional studies mainly focused on extracting handcrafted features from skeleton sequences [10], [14]–[16], such as features that capture the relative rotation [16] or translations [15] between various joints. However, the handcrafted features were not generalizable for various datasets and scenarios [17]. Recently, deep learning-based HAR has become the mainstream research due to its robust feature learning ability. For instance, To capture the temporal motion patterns, recurrent neural networks (RNNs) based methods [18]–[20] were proposed for HAR. Meanwhile, convolution neural networks (CNNs) have been adapted to represent the skeleton sequence as pseudo-images [21]–[23]. Apart from previous methods, spatial-temporal graph convolution networks (ST-GCNs) [9], [24]–[29] were proposed to capture the relationship among body joints and become the most popular pipeline for HAR since they can capture inherent interaction between body joints through vertices aggregation within a graph.

In the initial ST-GCN *et al.* [9], the human body was represented as a predefined skeleton graph with vertices and edges, and then the GCN was applied in spatial and temporal dimensions, respectively. However, the predefined fixed graph was inefficient for action recognition [29], and limited the changeable human movement representation for GCN. Thus some adaptive graph generation methods were proposed [25], [28]–[30] to boost the flexibility of the model for better recognition performance.

However, for all the aforementioned graph-based works, they assumed all joints/edges as the same type and did not consider the occurrence order of video frames. In other words, they did not exploit the underlying semantic relations of the human skeleton, which has been proven to be essential for HAR in this work [28]. On the one hand, human actions can be described as the relative movements of different body parts. For example, pointing to somewhere mainly depends on swinging the arms but kicking forward relies on swinging legs. In this case, the types of moving nodes will be useful information for action understanding. On the other hand, when capturing temporal information, existing methods update the features of the current frame by aggregating information from its previous and later frames. This leads to misclassifying the actions that share the same movement pattern but belong to the different occurrence order of frames, such as put on/off shoes.

Zhang *et al.* [28] first noticed this limitation and proposed a semantics-guided neural network to explore the semantic information in skeleton-based HAR. They enriched the input joints feature by explicitly adding one-hot vectors of different

node types and frame indexes. They achieved satisfying recognition performance but still faced several challenges: (1) the explicit encoding in the input is not flexible, and the semantic information might be over-smoothed when GCNs go deeper. In this way, the model cannot incorporate high-order semantic information, leading to a limited semantics representation. We have proved this in our experiments, please refer to Section. **IV.C** for details. The result shows that encoding the semantics in all ST-GCN layers can get better recognition performance than only encoding the semantics in the initial stage of ST-GCN. (2) The edge types between nodes were not considered. In detail, the connection in different types of joints might be various, even between the same type of joints but in different directions, so the connection weight value will be different. Taking legs and arms as an example, in a graph, 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 could be different and vice versa.

To address the aforementioned limitations, a dynamic semantic-based spatial-temporal graph neural convolution network (DS-STGCN) was proposed in this paper. Specifically, a dynamic semantic spatial encoding module (DS-SGCN) and a dynamic semantic temporal encoding module (DS-TGCN) were proposed for skeleton-based human action recognition. The main idea of the proposed method is to encode the dynamical semantic information in the GCNs aggregation process implicitly. Specifically, in the DS-SGCN, the individual transform function was learned for the joints/edges in different types. In this case, the joints/edges type can be encoded by projecting the joint/edge feature into their specific distribution. In the DS-TGCN, a causal convolution was designed, where only the previous frames were utilized for aggregation when updating the feature of the current frame. In this case, the order information of frames can be reserved in the temporal graph convolution progress.

Compared with the previous works, the advantages of the proposed DS-STGCN can be elaborated threefold. (1) Since the semantic information of joints/edges was learned from the sample itself, the dynamic nature of each skeleton can be maintained. (2) The joints/edges type, as well as the order information of frames, were encoded in each ST-GCN layer through the proposed semantic structure. Consequentially, the semantic information can be reserved without over-smoothing even if the model goes deeper. (3) Only the previous frames were aggregated when capturing the temporal feature, so our model can be utilized for live video with various lengths, making it more flexible.

The extensive experiments on NTU-RGBD [12], [31], Kinetics-400 [3], and FineGYM [32] show that: (1) the proposed DS-SGCN and DS-TGCN are generalized enough and can be plugged into various ST-GCNs structures to boost their performance. (2) the proposed DS-STGCN is efficient. It outperforms state-of-the-art methods notably on all four datasets but with the smallest model size.

The main contributions are summarized as follows:

- We proposed a dynamic semantic spatial convolution module (DS-SGCN) to encode the joint and edge type in the skeleton graph, leading to a more reasonable and generalizable skeleton graph modeling. Extensive experiments show that the proposed semantic spatial graph can be adapted to various state-of-the-art ST-GCN structures to boost their classification performance.
- We proposed a semantic temporal convolution module (DS-TGCN) to encode the occurrence order of the frames. Extensive experiences show that the proposed module can classify actions more accurately for the same movement pattern but different in the occurrence order of frames, such as taking on/off shoes. Meanwhile, it showed the potential to be applied to live videos.
- Based on the proposed semantic modules (DS-SGCN and DS-TGCN), a dynamic semantic-based spatial-temporal graph neural network (DS-STGCN) was developed. Extensive experiments highlight that the proposed DS-STGCN outperforms SOTA methods notably on NTU-RGB+D, Kinetics-400, and FineGYM.

The rest of this paper is organized as follows. The works related to skeleton-based human action recognition were reviewed in Section II. The formulation of the ST-GCN and its variants were discussed, and the proposed DS-STGCN was introduced then in Section III. Extensive experiments were done in Section IV. Finally, The conclusion of the paper was summarized in Section V.

II. RELATED WORK

In this section, graph neural network was briefly introduced first. Then the existing GCN-based methods for skeleton-based human action recognition were briefly reviewed. Finally, an overview of the semantic information exploration in the human skeleton was provided.

A. Graph neural networks

Graph neural networks (GNNs) have been widely explored in addressing graph-based data [33]–[38]. The main idea of these methods is message passing where the individual representation is obtained through aggregating the information from its *h*-hop neighbors. Taking the Graph Convolution Network (GCN) [34] as an example, the node representation is updated by averaging the information from the *one*-hop neighbor of each node in the graph. Then it is followed by a linear projection and non-linear activation operations to represent the node in a high-dimension feature space. In practice, to capture the long-distance correlation within a graph, multi-layer GCNs are connected. Most skeleton-based action recognition methods [9], [24]–[29], [39] adopted the same rule, where the initial embedding was set to the coordinates of joints; the graph topology was either pre-defined or adaptive; the output at the final layer corresponded to the representation of the human skeleton.

B. GCNs for skeleton-based action recognition

GCNs have attracted increasing attention in skeleton-based human action recognition [9], [24]–[29], [39]–[41]. Yan *et al.* [9] 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 [25], [26], [28], [29], [39]. Instead of only applying the fixed graph structure, these methods learned other adaptive graphs to boost the GCNs' representation ability. For instance, the 2S-AGCN [29] learned a data-driven adaptive graph for all feature channels, and CTR-GCN [25] learned an adaptive graph for each individual feature channel. Meanwhile, the multi-scale and shift GCN were proposed [26], [27] to address the over-smooth problem in graph long-distance transfer. In the temporal pattern, multi-scale temporal convolution was proposed to boost the information aggregation in temporal space [25], [39].

C. Movement semantic information exploration

Semantic information has been exploited in RNNs for skeleton-based human action recognition [18], [24], [42]. In these methods, the skeleton structure was manually partitioned into different functional parts, and processed by the individual RNN. As the network went deeper, the feature of different components was concatenated and progressed in a hierarchical way. Even though such semantic information was important, *i.e.* the joint types, was overlooked by most previous GCNs for skeleton-based human action recognition. To address this, Zhang *et al.* [28] proposed the SGN to encode the information of joint types in the initial features by explicitly adding one-hot vectors that represent different node types. However, this pre-defined semantics encoding in the input layer was not flexible and cannot represent such information in high-dimension space when networks went deeper.

To tackle the above limitations, we proposed a more elegant method to encode the semantics implicitly. In brief, the semantics were encoded during the GCN processing, so that it can be encoded in various layers of ST-GCN with more flexibility. For example, the joints/edge types were encoded in their corresponding distribution space by individual transformation functions, and the occurrence of frames order was encoded by a special temporal convolution design where only previous frames were considered during updating the feature in the current frame. In such cases, compared with adding pre-defined semantics in initial features, the proposed dynamic semantics encoding methods lead to a more flexible semantics representation and ensure that the semantics representation remains expressive without suffering from excessive smoothing effects even as the models become deeper.

III. METHOD

In this section, The notation of ST-GCN and its variants are formulated and discussed first, then the dynamic semantic spatial and temporal convolution modules will be introduced specifically. Finally, the proposed DS-STGCN will be described in detail.

A. Preliminaries

Notation. A skeleton data is denoted as a spatial-temporal graph $\mathcal{G} = (V, E_s, E_t, X)$ 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 $\mathcal{G}_s = (V, E_s, X)$, 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 $\mathcal{G}_t = (V, E_t, X)$, 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 to capture the relationship of joints within the same frame, and the temporal-GCN (T-GCN) to capture the joint movement along the temporal. The variants of S-GCN and T-GCN were formulated and discussed below.

1) Spatial Graph Convolution Networks.: The previous S-GCNs were categorized into three types in this paper: topology-fixed, topology-adaptive, and semantic-guided spatial graph convolution networks, which are formulated as follows:

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 [9], 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 the 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, such as:

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

where f is an updating function, which is a 2D convolution network with kernel size 1; 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 [9], [24]–[29], [39], an 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, 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.

Semantic-guided Graph Convolution Network. In explicit semantic encoding method [28], the input feature was refined by adding an 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, and the

topology-adaptive graph convolution network then works on this input.

2) Temporal Graph Convolution Network.: The key idea of T-GCN is to update the joints feature at the current frame by aggregating the feature from its *K-neighbor* frames. Here, the previous T-GCNs were categorized into two types, including the original T-GCN and the multi-scale T-GCN.

Original Temporal Graph Convolution Network. In ST-GCN [9], the original T-GCN was proposed, where the features in the current frame were represented as the combination of its *K-neighbor* frames. This process can be formulated as Eq.5.

$$X_t^{'} = \sum_{k=-l}^{l} w_k * X_{t+k}, \tag{5}$$

where l is the window size of temporal convolution; $w \in \mathbb{R}^{2l-1}$ is the learnable weight for feature aggregating.

Multi-scale Temporal Graph Convolution Network. The main idea of multi-scale temporal graph convolution networks [25], [39] was applying various window sizes to capture the movement with the different sequence lengths, then utilizing a transform function to combine the feature from all scales. This process can be formulated as Eq.6, such as:

$$X_t^{'} = f(Concat_{l \in L}(\sum_{k=-l}^{l} w_k * X_{t+k}), \theta), \tag{6}$$

where l is the window size of temporal convolution; L is the set of temporal length; $w_k \in \mathbb{R}^{2l-1}$ is the learnable weight for feature aggregating; f is the updating function which is a 2D convolution network with kernel size 1, and θ are the learnable parameters of the updating function.

B. Dynamic semantic spatial graph convolution network (DS-SGCN)

The general framework of the proposed DS-SGCN is adapted to the topology-adaptive GCN. Compared with the previous methods, the joint and edge type in the skeleton graph were encoded dynamically when calculating the adaptive graph. As shown in Figure. 1, the joints and edges were split into different types in advance. For the definition of the joint/edge type, the human body was decomposed into several parts (*i.e.* five parts in this paper, including left/right arms, left/right legs, and one trunk) according to the natural structure. The edge type can be determined based on the type pair of its end nodes. For instance, the link between the left arm and trunk differs from the link within the trunk.

Two semantic-aware modules were proposed to encode the joint/edge type, namely, the node type-aware adaptive graph module and the edge type-aware adaptive graph. as shown in Figure. 2. In the node type-aware module, the non-local mechanism was applied. But separate transform functions were designed for each body part to project the node representation in their specific type distributions. Thus, the adaptive graph can generate with consideration of the node type. Similar to the node type encoding, the edge type-specific transform functions were designed in the edge type-aware module, and were then applied to the adaptive skeleton

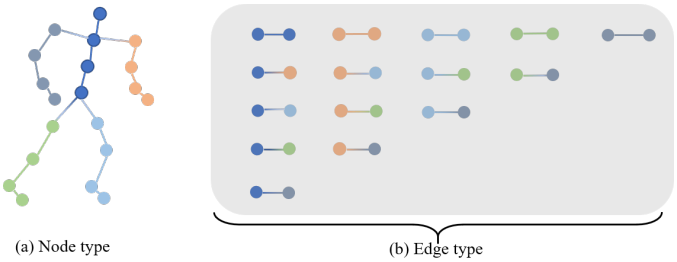

Fig. 1. Definition of joint/edge type. (a) A human body is split into five parts shown with different colors: left/right arms, left/right legs, and one trunk. (b) The edge type is represented as the type pair of its end nodes; the node type is represented by using different colors, and there are fifteen edge types.

graph to encode semantic information over each edge type. In this case, the spatial graph in our work can be defined as a directed graph $G = (V, E, A, R, X)$, where A denotes the joints type mapping function for each node and is represented as $V \rightarrow A : \tau(v) = \{\tau_1(v), \tau_2(v)\}$, and the R denote the edge type mapping function $E \rightarrow R : \phi(e)$ Supposing the input feature $X \in \mathbb{R}^{N \times d}$, The semantic-based adaptive graph is calculated as Eq. 7

$$\begin{aligned} A_D^n &= \sigma(\tau_1(X) - \tau_2(X)), \\ A_D^e &= \phi(A_D), \end{aligned} \tag{7}$$

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

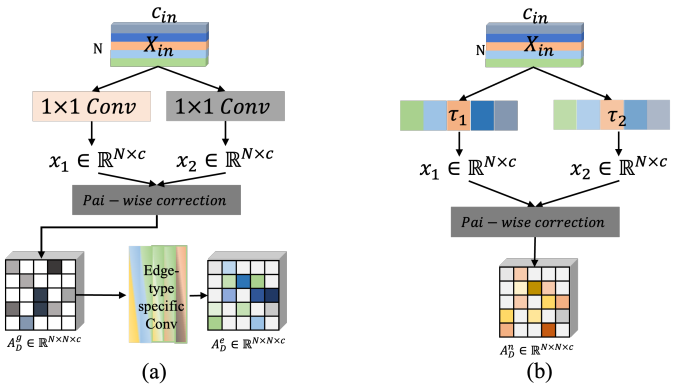

Fig. 2. Illustration of node and edge type-aware adaptive graph generation. (a) represents the edge type-aware adaptive graph generation. The general adaptive graph $A_D^g \in \mathbb{R}^{N \times c}$ is calculated first, then edge type-aware adaptive graph $A_D^e \in \mathbb{R}^{N \times c}$ can be calculated by the edge-type specific transform function. (b) represents the node type-aware adaptive generation. The input was first projected into corresponding feature space by utilizing the node-type specific transform function τ_1 and τ_2, then the $A_D^n \in \mathbb{R}^{N \times c}$ can be obtained according to the pair-wise correction manner.

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 be generated according to the non-local mechanism. Specifically, denoting s and t as two nodes of different types, $x_s \in \mathbb{R}^{1 \times d}$

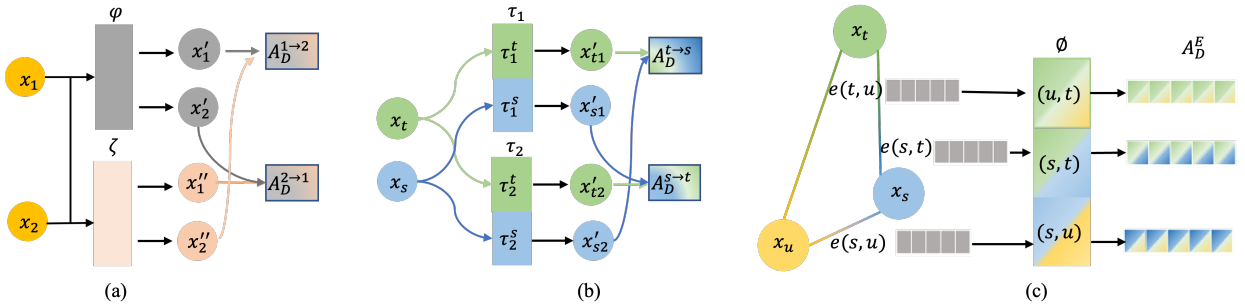

Fig. 3. Illustration of the adaptive graph generation. (a) represents the standard non-local mechanism. For each transform function $\varphi(\cdot)$ and $\xi(\cdot)$, the node features are updated by sharing the same parameters. (b) represents the node type-aware adaptive graph. In each transform function, the convolution kernels are divided into several parts, each of which corresponds to a specific node type. Then the node characteristics in different types were updated by their individual parameters set. In this case, the types of nodes can be represented dynamically. The colored circles denote different node types and the colored squares denote different convolution kernels. (c) illustrates the edge type-aware adaptive graph generation. For each type of edge, specific convolution kernels were designed and utilized for updating the edge feature. In this case, the types of edges can be represented in their individual feature space dynamically. The colored circles denote node types, and mix-colored squares denote edges with corresponding node pairs.

and $x_t \in \mathbb{R}^{1\times d}$ as the corresponding feature, then the node-aware feature representation can be formulated as Eq. 8

$$\begin{aligned} x^{'}_{s1} &= \tau^s_1(x_s), x^{'}_{s2} = \tau^s_2(x_s) \\ x^{'}_{t1} &= \tau^t_1(x_t), x^{'}_{t2} = \tau^t_2(x_t), \end{aligned} \tag{8}$$

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. 9:

$$A^{s\to t}_D = \sigma(x^{'}_{s1} - x^{'}_{t2}), A^{t\to s}_D = \sigma(x^{'}_{t1} - x^{'}_{s2}), \tag{9}$$

where σ is the activation function. $A^*_D \in \mathbb{R}^{1\times C}$. For the whole skeleton structure, the node aware-adaptive graph $A^n_D \in \mathbb{R}^{N\times N\times C}$ 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 features $e_{\langle s,t\rangle}$, $e_{\langle s,u\rangle}$ and $e_{\langle t,u\rangle}$. Thus, the edge type-aware adaptive correlation can be refined in Eq. 10:

$$\begin{aligned} A^{\langle s,t\rangle}_D &= \phi^{\langle s,t\rangle}(e_{\langle s,t\rangle}) \\ A^{\langle s,u\rangle}_D &= \phi^{\langle s,u\rangle}(e_{\langle s,u\rangle}) \\ A^{\langle t,u\rangle}_D &= \phi^{\langle t,u\rangle}(e_{\langle t,u\rangle}), \end{aligned} \tag{10}$$

where $\phi^{\langle *,*\rangle}(e)$ represents separate transform functions. In this work, the 2D convolution kernel with kernel size equal to 1 was applied. The edge type-aware topology can be represented as $A^e_D = \{A^{\langle s,t\rangle}_{D_{ij}} | i,j = 1,...,N, s,t = 1,...,M\}$, where s and t are the node type index respectively; M is the number of types.

Dynamic semantic spatial graph convolution: As shown in Figure.4. In DS-SGCN, the spatial graph convolution structure was decomposed into three branches, the node-type aware branch, the edge-type aware branch, and the general branch. This is different from the previous ST-GCNs which utilized the same spatial graph convolution structure on three pre-generated skeleton graphs. A branch-wise weight is set as learnable 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 branches. 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-SGCN can be formulated as Eq. 11.

$$\begin{aligned} X^{'} &= f(x,\theta) \\ x &= [x^n, x^e, x^g] \in \mathbb{R}^{N\times 3c} \\ x^n &= (A^1 + \lambda_1 A^n_D) f^1_{pre}(X) \\ x^e &= (A^2 + \lambda_2 A^e_D) f^2_{pre}(X) \\ x^g &= (A^3 + \lambda_3 A_D) f^3_{pre}(X), \end{aligned} \tag{11}$$

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}, and equals to C/K. K is set to 8 in this work. 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 a learnable weight to refine the effect of each semantic-based topology-adaptive graph, which is different between branches.

C. Semantic temporal graph convolution network

In this section, the proposed DS-TGCN is introduced in detail. The key idea of the proposed DS-TGCN is to encode the order of frames during the temporal aggregating, which is critical to distinguish actions that share the same movement pattern but occur in different orders of frames. Instead of encoding the frame index into the joint representation, a special temporal convolution was designed to encode the

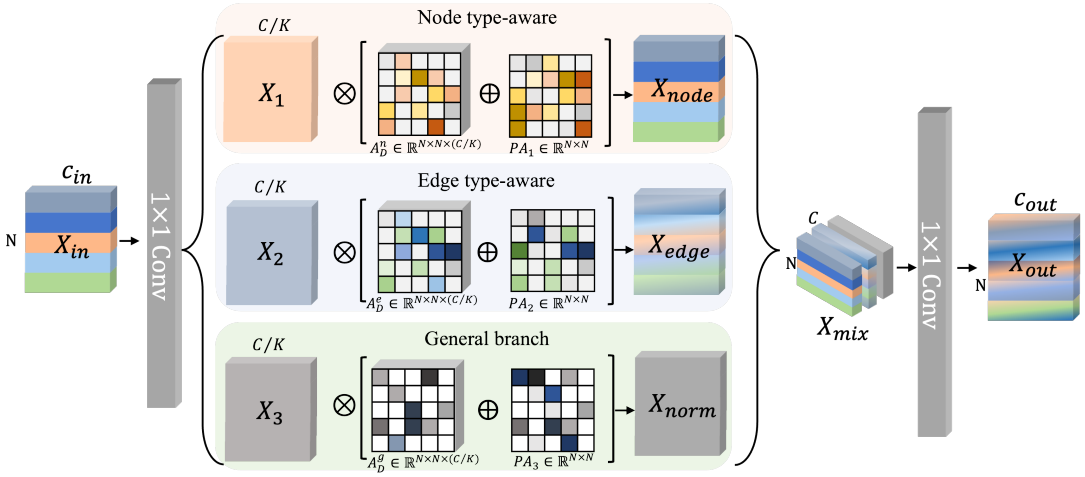

Fig. 4. The framework of the proposed DS-GCN. The spatial graph convolution structure was decomposed into three branches, the node-type aware branch, the edge-type aware branch, and the general branch. In each branch, the corresponding semantic self-adaptive graph and a shared correction matrix $PA_i \in \mathbb{R}^{N\times N}, i = 1, 2, 3$ were applied to represent the skeleton structure. Then the mix output X_{mix} can be obtained by contacting the three branches along the feature channel dimension, and the final output X_{out} can be calculated by a 1×1 convoluted X_{mix}.

temporal semantic information in an implicit way. As shown in Fig. 5, only the features in its previous frames were aggregated when updating the joint in the current frame. This process can be formulated as Eq. 12

$$X_t^{'} = \sum_{k=-l}^{0} w_k * X_{t+k}, \tag{12}$$

where l is the window size of temporal convolution; $w \in R^l$ is the learnable weight for feature aggregating.

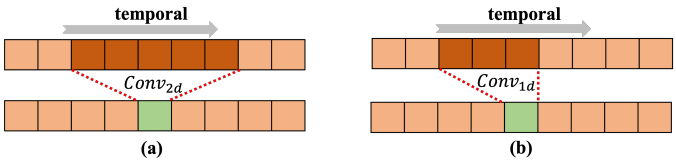

Fig. 5. Illustration temporal convolution. (a) is the original temporal convolution, where the Convolution 2D kernel was utilized to update the current frame by aggregating its previous and later frames. (b)is the dynamic semantic-based temporal convolution, where the convolution 1D kernel was adopted to update the current frame by only aggregating its previous frames.

The framework of DS-TGCN is illustrated in Fig. 6. It contains two branches: the first is the original temporal convolution to reserve the receptive field and the second is the semantic temporal convolution to highlight the order of the frames. These two branches were combined with a learnable λ_t to refine the effect of each other.

As for the semantic temporal convolution shown in Fig. 6, the 1D convolution kernel (*1D-Conv*) was utilized as the main aggregating function. In practice, denoting the input of DS-TGCN as $X \in \mathbb{R}^{B\times C_{in}\times\ T\times V}$, where the B, C_{in}, T, V is the batch size, feature channel, temporal length and the number of joints, respectively. To adopt the input to *1D-Conv*, the X was reshaped into $X^{'} \in \mathbb{R}^{BV\times C_{in}\times\ T}$ first, then the aggregation

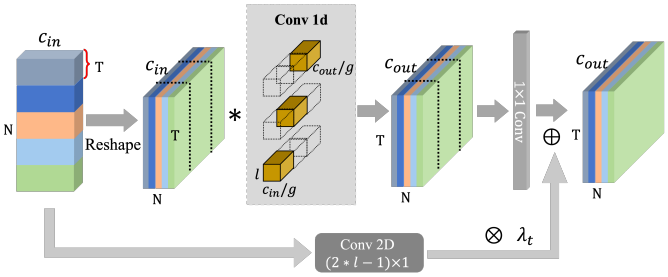

Fig. 6. The framework of DS-TGCN. The input was first reshaped to $X^{'} \in \mathbb{R}^{BV\times C\times\ T}$ and divided into several groups along the feature channel dimension. For each group, a specific Conv1d kernel was designed for feature aggregation. Thus the output can be obtained by concatenating the output of each group along the channel dimension. Meanwhile, an original temporal convolution kernel is applied as a res-connection to reserve the receptive field.

process was operated by a *1D-Conv* with kernel size equal to $[C_{in}, l, C_{out}]$, where C_{out} is the number of output channels.

However, this process makes the model become huge. In order to decrease the number of parameters, the input of the temporal module was divided into several groups g along the feature channel dimension. For each group, a specific *1D-Conv* with kernel size $[C_{in}/g, l, C_{out}/g]$ was designed for feature aggregation. In this case, the number of whole semantic temporal convolution kernels can be reduced to $[g, C_{in}/g, l, C_{out}/g]$, which is g times smaller than using the *1D-Conv* directly. The final output of the semantic temporal convolution can be obtained by stacking the output of all groups and followed by a 2D convolution network with kernel size 1 to combine the correlation within channels. Here, g is set equal to C_{in}; in this case, the temporal semantic feature was aggregated in a channel-wise manner. The formulation of the proposed DS-TGCN can be represented as Eq. 13

$$X_t^{'} = f(\sum_{k=-l}^{0} w_k * X_{t+k} + \lambda_t * \sum_{q=-l}^{l} w_q * X_{t+q}, \theta), \tag{13}$$

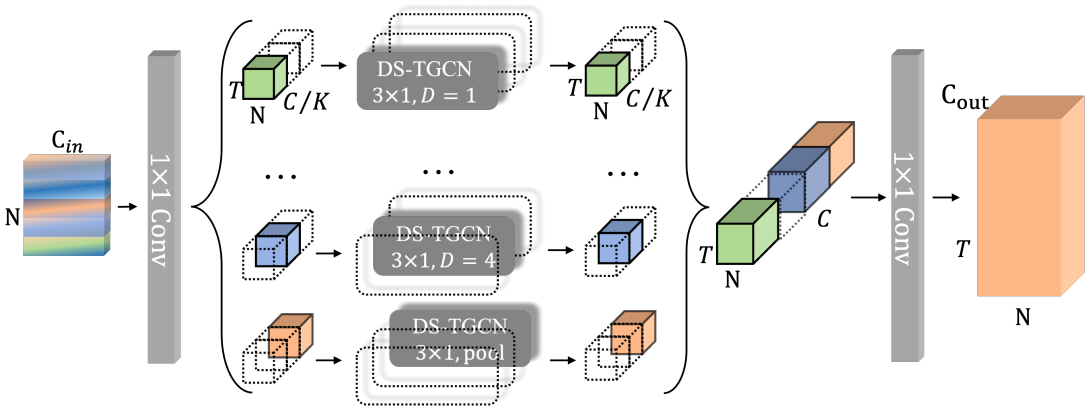

Fig. 7. The framework of multi-scale DS-TGCN. The input was first reshaped to $X^{'} \in \mathbb{R}^{BV\times C\times\ T}$ and divided into several groups along the feature channel dimension. For each group, a specific Conv1d kernel was designed for feature aggregation. Thus the output can be obtained by concatenating the outputs of each group along the channel dimension.

where $w_k \in R^l$ is the learnable weight for semantic temporal branch feature aggregating; $w_q \in R^{2l-1}$ is the learnable weight for original temporal branch feature aggregating. f is the updating function which is a 2D convolution network with kernel size 1, and θ is the learnable parameter.

Inspired by the multi-scale temporal module [25], [39], a multi-scale DS-TGCN was proposed and shown in Fig 7, where multiple branches with different temporal window sizes were contained. Each branch performs DS-TGCN for a temporal feature aggregation independently. To save amounts of computation, the input was first transferred with a 2D convolution network with kernel size 1 and divided into several groups for corresponding branches. The final output of multi-scale DS-TGCN is the representation of all branches followed by a transformation. In practice, there are four branches in our final framework. In each branch, a 1×1 convolution kernel was utilized to reduce channel dimension, then followed by semantic temporal convolutions with various dilation ([1,2,3,4] here) to model actions with different duration. The output of the temporal module can be obtained by concatenating the result of these four branches.

D. Dynamic semantic-based spatial-temporal GCN

Based on the proposed DS-SGCN in *Sec. B* and DS-TGCN *Sec. C*, a reasonable graph convolution network DS-STGCN was developed for skeleton-based human action recognition. Similar to ST-GCN [24], 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, a DS-SGCN and a multi-scale DS-TGCN were contained.

IV. Experiments

A. Datasets

To demonstrate the advantage of the proposed DS-STGCN, four datasets were utilized in this paper: NTU RGB+D 60 [31], NTU RGC+D 120 [12], Kinetics-400 [3], and FineGYM [32].

NTU RGB+D 60 [31].The action samples are performed by 40 volunteers and categorized into 60 classes. Each sample contains an action and is guaranteed to have at most 2 subjects, which are captured by three Microsoft Kinect v2 cameras from different views concurrently. The authors of this dataset recommend two benchmarks: (1) cross-subject (NTU60-Xsub): training data comes from 20 subjects, and testing data comes from the other 20 subjects. (2) cross-view (NTU60–view): training data comes from camera views 2 and 3, and testing data comes from camera view 1.

NTU RGB+D 120 [12]. NTU RGB+D 120 is currently the largest dataset with 3D joint annotations for HAR, which extends NTU RGB+D 60 with additional 57,367 skeleton sequences over 60 extra action classes. Totally 113,945 samples over 120 classes are performed by 106 volunteers, captured with three camera views. This dataset contains 32 setups, each denoting a specific location and background. The authors of this dataset recommend two benchmarks: (1) cross-subject (NTU120-Xsub): training data comes from 53 subjects, and testing data comes from the other 53 subjects. (2) cross-setup (NTU120-Xset): training data comes from samples with even setup IDs, and testing data comes from samples with odd setup IDs.

Kinetics-400 [3]. Kinetics-400 is a large-scale action recognition dataset with 400 actions. The skeletons utilized in this paper were provided by [43], where the Openose algorithm [44] was applied for joint estimation. The box threshold of human detection is set as 0.5. After the validation, there are a total of 236,489 skeleton sequences for training and 19,505 skeleton sequences for testing.

FineGYM [32]. FineGYM is a fine-grained action recognition dataset with 29000 videos of 99 fine-grained gymnastic action classes. In this paper, skeletons are extracted with ground-truth human bounding boxes as described in [23].

B. Implementations Details

All experiments are conducted on one A100 GPU with the PyTorch deep learning framework. All models are trained for 100 epochs with the Cosine Annealing learning rate scheduler by using SGD with momentum 0.9, weight decay $5e^{-4}$.

Method	NTU60-XSub	NTU120-XSet	GFLOPs	Params
ST-GCN [24]	87.8	85.0	3.27	3.10M
ST-GCN w/ T	**88.3**	**86.2**	3.55	3.37M
2s-GCN [29]	89.2	85.4	3.74	3.47M
2s-GCN w/ T	89.5	85.8	5.30	4.78M
2s-GCN w/ S	89.6	85.6	4.02	3.74M
2s-GCN w/ ST	**89.9**	**86.2**	5.59	5.05M
CTR-GCN [25]	89.7	85.8	1.69	1.45M
CTR-GCN w/ S	89.8	86.1	1.94	1.67M
CTR-GCN w/ T	90.0	86.2	1.75	1.50M
CTR-GCN w/ ST	**90.4**	**86.3**	2.00	1.72M
DS-STGCN w/o ST	90.0	86.4	1.11	1.31M
DS-STGCN w/o S	90.7	87.3	1.14	1.34M
DS-STGCN w/o T	90.8	87.2	1.19	1.38M
DS-STGCN	**90.8**	**87.6**	1.23	1.41M

TABLE I
GENERALIZATION OF DS-SGCN AND DS-TGCN, AND EFFECTIVENESS OF DS-STGCN. THE PROPOSED SEMANTICS ENCODING MODULES ARE GENERALIZED ENOUGH THAT CAN BE ADAPTED TO VARIOUS ST-GCNS. THE PROPOSED DS-STGCN CAN ACHIEVE THE BEST PERFORMANCE.

The initial learning rate was set to 0.1. 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 comparing the stare-of-the-arts. The pre-processing approach follows the setting in [43].

C. Ablation Study

In this section, two benchmarks (NTU60-Xsub and 120-Xset) were utilized for validation. At first, the effectiveness of the proposed DS-STGCN and its two semantic modules were assessed. Then the components for each semantic module were analyzed separately. The joint coordinates were utilized as the node feature in this part, and the initial adjacent matrix for the spatial graph was set to totally learnable.

1) Effectiveness of DS-STGCN: In order to validate the effectiveness of the proposed DS-STGCN, various ST-GCNs, vanilla ST-GCN [24], 2s-GCN [29] and CTR-GCN [25] were utilized as the backbones in this experiment. The results are shown in Table I, it can be observed that the topology-adaptive graph convolution network (2s-GCN, CTR-GCN, and DS-STGCN) achieves better performance than the topology-fixed graph convolution network (ST-GCN). Compared with the CTR-GCN [25], the accuracy of the proposed DS-STGCN observes a 1.1% and 1.8% increase in NTU60 Xsub and NTU120 Xset respectively. In terms of the model size, The DS-STGCN is the smallest compared with the others.

In order to analyze the classification performance in more detail, we performed some statistical analysis of the classification distribution in the whole dataset. For each sample, the max prediction probability but not the predicted label was utilized as the final result, then the violin map was generated to represent the distribution of classification results, the result is shown in Figure. 8. It is clear that, in the multi-class classification task, the greater the max prediction probability, the more accurate and confident the classification. When looking in Figure. 8 (a) and (d), we can observe that the distribution for the proposed DS-STGCN is more compact and the center of distribution is located near 1, which can explain the superiority of the proposed DS-STGCN. Meanwhile, to make the results more reasonable, we set 0.5 as the threshold to define the range of confidence, and the samples with a max prediction probability lower than 0.5 were utilized for another violin map generation. As shown in Figure. 8 (b) and (e). It can be found that the numbers of samples with a max prediction probability lower than 0.5 are significantly reduced in DS-STGCN when compared with STGCN and AAGCN, which indicates that the proposed algorithm can classify the indistinguishable samples more confidently.

2) Effectiveness and Generalization of DS-SGCN and DS-TGCN: In this part, we first verified the generalization of proposed semantic modules by introducing the DS-SGCN and DS-TGCN in several ST-GCNs (ST-GCN, 2s-GCN, and CTR-GCN). Then ablation experiments were done to explore the effectiveness of two semantic modules of the proposed DS-STGCN. The results are shown in Table I.

Generalization of DS-SGCN and DS-TGCN: In practice, the proposed DS-SGCN and DS-TGCN were adapted and utilized to replace the corresponding modules in these backbones. As shown in Table I, S and T represent the DS-SGCN and DS-TGCN respectively, and $w/$ and w/o represent $with$ and $without$ respectively. For instance, 2s-GCN $w/\ T$ means the original temporal convolution module in 2s-GCN was replaced by DS-TGCN. 2s-GCN $w/\ ST$ means the original temporal and spatial convolution module in 2s-GCN was replaced by DS-TGCN and DS-SGCN respectively. In detail, in ST-GCN, since there is no adaptive spatial graph generation, Only DS-TGCN was verified, meanwhile, in CTR-GCN, the mutiscale DS-TGCN was utilized. Noted that, because the spatial module in 2s-GCN and CTR-GCN shared the same structure in three branches. To keep this characteristic, the DS-SGCN was regenerated for a fair comparison as follows: The node/edge type-aware adaptive graph modules were combined in series. For each spatial branch, the node type-aware adaptive graph was calculated according to Eq 9, then the node type-aware adaptive module was applied on the A_D^n. The semantic-based adaptive graph A_D^{NE} can be formulated as Eq. 14.

$$A_D^{NE} = \phi(A_D^n) = \phi(\tau_1(X) - \tau_2(X)) \tag{14}$$

The results are presented in Table I. It can be observed that both proposed two semantic modules can introduce specific positive effects when utilized in three backbones respectively, and the combination of two semantic modules can achieve the best performance when compared with the original backbones.

In order to explain the performance of the proposed semantic modules, violin map was generated for each backbone. As shown in Figure. 8 (c) and (f), for each backbone, the left part denotes the result for the backbone with semantic modules and is represented as color orange, the right part denotes the result for the backbone without semantic modules and is represented as the color green. The samples with a max prediction probability lower than 0.5 were utilized for the result explanation. In each backbone, it can be observed that the area of the violin map for the model with semantic encoding shows a significant decrease when compared with the area for the model without semantic encoding, which means that the proposed semantic encoding algorithm can

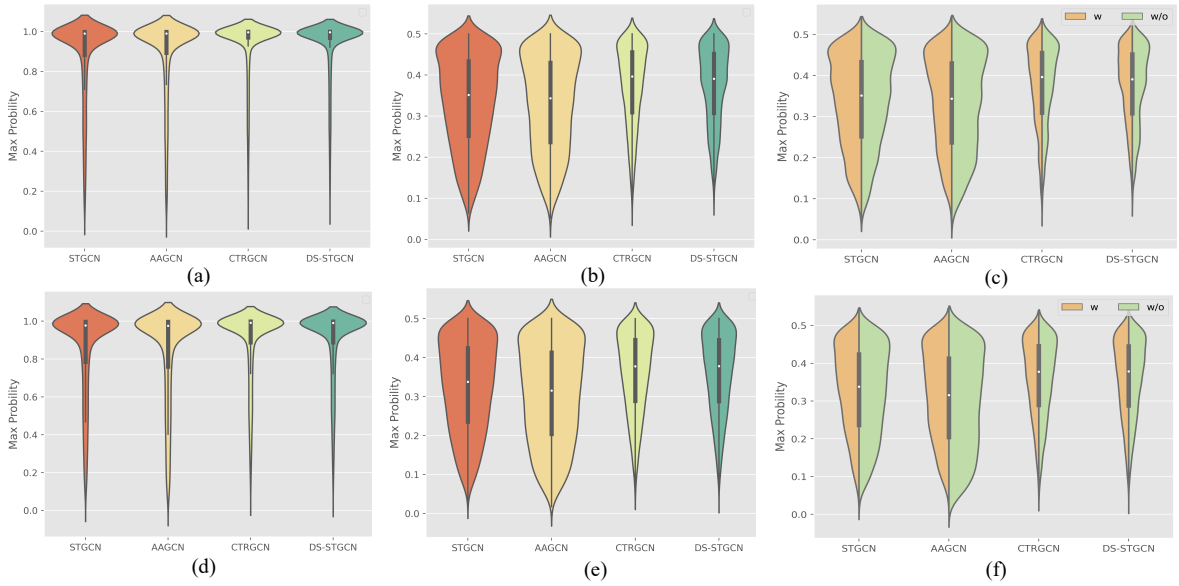

Fig. 8. Analysis of the classification distribution for each backbone. (a-c) represent the classification result distribution for all samples in NTU60 Xsub. (d-f) represent the classification result distribution for all samples in NTU120 Xset. The max probability of each sample was utilized as the final result. In (a)and (d), the samples with max probability ranging from 0 to 1 were analyzed. In (b-c) and (e-f), the samples with a max probability lower than 0.5 were analyzed. In (e) and (f), w means that backbone with semantic modules, and w/o means that backbone without semantic modules. The area for each violin map indicates the number of samples. Observing in (a) and (d), we can observe that the distribution generated by the proposed DS-STGCN is more compact, also in (b) and (e), it can be found that the numbers of samples with max probability lower than 0.5 are significantly reduced when comparing with STGCN and AAGCN. When looking at (e) and (f), we can see that for each backbone, the area for a model with semantic encoding is decreased significantly when compared with the model without semantic encoding, implying the proposed semantic modules can make the classification more accurate.

provide positive effects on classifying the indistinguishable samples without changing the model structure. Meanwhile, these results also proved that the proposed semantic modules are generalized and can be adjusted into various ST-GCNs.

Effectiveness of DS-SGCN and DS-TGCN: In this part, the effectiveness of each semantic module in DS-STGCN was verified. As shown in Table I, the DS-STGCN $w/o\ ST$ was utilized as the backbone, where all of the node and edge type-aware branches in spatial-GCN were replaced by the general branch, and all of the DS-TGCNs were replaced by the original temporal-GCN. To verify the effectiveness of DS-TGCN, only DS-TGCN was added in the backbone as a temporal module and represented as DS-STGCN $w/o\ S$. Similarly, adding the DS-SGCN to the backbone for spatial semantic module verification, and is represented as DS-STGCN $w/o\ T$. The result shows that all of the semantic modules can help to improve classification performance, the combination of two semantic modules can get the best, 0.8% and 1.2% increase in NTU60-Xsub and NTU120-Xset when compared with DS-STGCN $w/o\ ST$.

3) Configuration Exploration on DS-SGCN: In this part, the components of DS-SGCN were analyzed, and the original multi-scale temporal graph convolution was utilized.

Ablation on the edge/node type encoding: The effects of node and edge type-aware branches were studied separately, the results are shown in Table II. Specifically, the node type-aware adaptive branch was replaced with the general branch to justify the effect of edge-type encoding. In this case, there are two general adaptive branches and one edge-type adaptive graph in the model which is represented as DS-SGCN w/o N. Similarly, the edge-type adaptive branch was replaced by the general branch to validate the effect of node-type encoding, and the model is represented as DS-SGCN w/o E. The baseline is the model with three general branches and is represented as DS-SGCN w/o NE. It can be seen that after encoding the node or edge type in the graph separately, the performance of action recognition can have a stable increase, combining both semantic branches can achieve the best performance. It can be observed that the Top1-acc of the DS-SGCN has a 0.8% increase in both NTU60-Xsub and NTU1200-Xset when compared with the baseline.

Method	NTU60-XSub	NTU120-XSet
DS-SGCN w/o N&E	90.0	86.4
DS-SGCN w/o N	90.3	86.5
DS-SGCN w/o E	90.5	87.0
DS-SGCN	**90.8**	**87.2**

TABLE II
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.

Configuration Exploration. The spatial learnable weight λ^s is analyzed in this section and the result is shown in Table III. Different from other topology-adaptive structures where one shared λ was utilized in all branches, in DS-SGCN, the branch-wised λ^s was applied. Specifically, an individual refinement weight is learned for each branch. Thus the DS-

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

TABLE III
COMPARISON OF DS-SGCN WITH DIFFERENT LEARNABLE WEIGHT MANNERS. DS-SGCN$_{\text{shared}}$ REPRESENTS THE DS-SGCN WITH SHARED λ^s FOR ALL BRANCHED, DS-SGCN$_{\text{B-wise}}$ REPRESENT THE DS-SGCN WITH INDIVIDUAL λ^s FOR DIFFERENT BRANCHES.

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

TABLE IV
EXPLORATION ON THE SPATIAL SEMANTIC ENCODING STAGE. DS-SGCN W/O N&E REPRESENTS THAT NO SEMANTIC MODULE IS UTILIZED, DS-SGCN$_{ini}$ REPRESENTS JUST UTILIZED DS-SGCN IN LAYER [1-4], DS-SGCN$_{mid}$ REPRESENTS JUST UTILIZED DS-SGCN IN LAYER [5-7], DS-SGCN$_{end}$ REPRESENTS JUST UTILIZED DS-SGCN IN LAYER [8-10], DS-SGCN REPRESENTS DS-SGC IS UTILIZED IN ALL LAYER.

SGCN was trained in two ways: on the one hand, utilizing a shared λ^s and represented as DS-SGCN$_{shared}$, on the other hand, setting a specific λ^s for each branch and denoted as DS-SGCN$_{\text{B-wise}}$. The results in Table III show that DS-SGCN learned with branch-wised λ^s can achieve better performance.

Exploration in the spatial semantics encoding stage. There are ten layers in the implemented DS-STGCN. In order to study the effects of spatial semantics in different layers, the whole model was divided into three stages: the initial stage represented as DS-SGCN$_{ini}$ which contains the layers from 1_{st} to 4_{th}, the middle stage DS-SGCN$_{mid}$ with layers 5_{th}-7_{th}, and the end stage DS-SGCN$_{end}$ with 8_{th}-10_{th}. Then DS-SGCN was utilized in different stages alone for comparison. For instance, to justify the effects of semantic information in the initial stage, the DS-SGCN is only utilized in layers 1_{st} to 4_{th}, meanwhile, in the rest layers, all semantic-based modules are replaced with the general adaptive branch. The result shows in Table. IV, it can be observed that the spatial semantic encoding can introduce a positive effect no matter at which stages the DS-SGCN is utilized, but when utilizing DS-SGCN in all layers, the model shows the best performance.

When comparing the performance within different stages, It can be seen that the semantics encoded in the middle stage is the most important. This can be explained as over-smoothing problems, the semantic information encoded in the initial stage might be over-smoothed when the layer goes deeper, also in the case of encoding semantics in the end stage, the joints feature was already over-smoothed during the former stages, thus the correlation matrix plays weakly effect on feature updating, which limited the ability of the semantic encoding module.

4) Configuration Exploration on DS-TGCN: In this part, the components of DS-TGCN were analyzed, and the branches in the DS-SGCN were replaced by the general branch.

Ablation on the original/semantic temporal convolution. In DS-TGCN, there are two branches corresponding to original temporal convolution and semantic temporal convolution

Method	NTU60-XSub	NTU120-XSet
DS-TGCN w/o Sem	90.0	86.4
DS-TGCN w/o Ori	90.5	86.7
DS-TGCN	**90.7**	**87.3**

TABLE V
ABLATION ON THE ORIGINAL/SEMANTIC TEMPORAL CONVOLUTION, DS-TGCN $w/o\ Ori$ REPRESENTS JUST THE SEMANTIC TEMPORAL CONVOLUTION WAS APPLIED, DS-TGCN $w/o\ Sem$ REPRESENTS ONLY ORIGINAL TEMPORAL CONVOLUTION WAS CONTAINED. IT CAN BE OBSERVED THAT SEMANTIC TEMPORAL CONVOLUTION CAN IMPROVE THE ACCURACY OF ACTION RECOGNITION. MEANWHILE, THE RESULT OF THE COMBINATION OF THE TWO BRANCHES DEMONSTRATED THE BEST PERFORMANCE.

respectively. In this part, the effectiveness of each branch was verified, and the results are shown in Table. V. In practice, to justify the semantic temporal convolution, the res-connection was removed from DS-TGCN, and the model was represented as DS-TGCN $w/o\ Ori$. The model only containing the original temporal convolution was represented as DS-TGCN $w/o\ Sem$ and was utilized as a backbone. It can be observed that, compared with the original temporal convolution, the proposed semantic temporal convolution can improve the accuracy of action recognition. Meanwhile, the results of the combination of the two branches achieved the best performance, which means that the original temporal convolution has specific advantages in temporal information capturing, and can be included to obtain a more complete temporal feature description.

In terms of visualization, the classification confusion matrix for DS-TGCN $w/o\ Ori$ and DS-TGCN $w/o\ Sem$ in NTU60 XSub was generated and shown in Figure. 9. It can be seen that the proposed semantic temporal convolution is more powerful when distinguishing the actions with the same movement but different in order of occurrence. Taking put-on/off shoes for example, the semantic temporal modules can achieve a more than 10% improvement for each action.

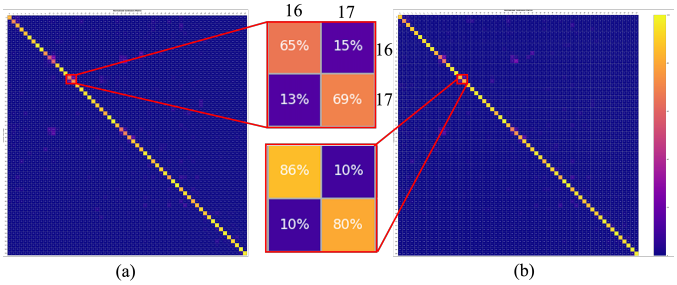

Fig. 9. Visualization of classification. The action index is as follows: taking on shoes (16), taking off shoes (17), (a) the confusion matrix for DS-TGCN $w/o\ Sem$, (b) the confusion matrix for DS-TGCN $w/o\ Ori$. It can be observed that semantic temporal modules can achieve a more than 10% improvement for each action.

Configuration Exploration. In this section, the learnable weight λ^t in DS-TGCN is analyzed. As shown in Table. VI. To justify the influence of λ^t, two different settings were applied, the one using the fixed λ^t equal to 1 and is represented as DS-TGCN$_{fix}$, the other one applying a learnable λ^t and is represented as DS-TGCN$_{adap}$. It can be observed that the

Method	NTU60-XSub	NTU120-XSet
DS-TGCN$_{fix}$	90.4	86.3
DS-TGCN$_{adap}$	**90.7**	**87.3**

TABLE VI
COMPARISON ON λ. DS-TGCN$_{fix}$ REPRESENTS THE FIX λ IN DS-TGCN, TGCN$_{adap}$ REPRESENTS THE ADAPTIVE λ IN DS-TGCN. THE RESULT SHOWS THAT TWO BRANCHES COMBINED BY A LEARNABLE λ ACHIEVED BETTER PERFORMANCE.

Module	60-Xsub	60-Xview	120-Xsub	120-Xset
ST-GCN [24]	81.5	88.3	70.7	73.2
SGN [28]	86.6	93.4	-	-
AS-GCN [46]	86.8	94.2	78.3	79.8
RA-GCN [47]	87.3	93.6	78.3	79.8
2s-GCN [29]	88.5	95.1	-	-
DGNN [48]	89.9	96.1	-	-
FGCN [49]	90.2	96.3	85.4	87.4
ShiftGCN [26]	90.7	96.5	85.9	87.6
DSTA-Net [50]	91.5	96.4	86.6	89.0
MS-G3D [27]	91.5	96.2	86.9	88.4
CTR-GCN [25]	92.4	96.8	88.9	90.6
ST-GCN++ [43]	92.6	97.4	88.6	90.8
PoseConv3D [23]*	**94.1**	97.1	86.9	90.3
infoGCN [11]	93.0	97.1	89.4	90.7
DS-STGCN	93.2	**97.5**	**89.4**	**91.2**

TABLE VII
COMPARISONS OF CLASSIFICATION ACCURACY AGAINST STATE-OF-THE-ART METHODS ON THE NTU RGB+D DATASET. $*$ REPRESENT THE CNN-BASED METHODS

Module	Kinetics-400	FineGYM
ST-GCN [24]	30.7	25.2*
AS-GCN [46]	34.8	-
RA-GCN [47]	34.8	-
2s-GCN [29]	36.1	-
DGNN [48]	36.9	-
MS-G3D [27]	38.0	92.6*
DG-STGCN [39]	40.3	-
PoseConv3D [23]*	47.7	94.3
DS-STGCN	**50.6**	**95.1**

TABLE VIII
CLASSIFICATION ACCURACY COMPARISONS AGAINST STATE-OF-THE-ART METHODS ON THE KINETICS-400 AND FINEGYM. MODEL WITH $*$ REPRESENT THE CNN-BASED METHODS, RESULT WITH * ARE REPORTED BY [23]

DS-SGCN with an adaptive combination manner can obtain a better performance in both NTU60 XSub and NTU120 XSet.

D. Comparisons with the State-of-the-Art

In order to make a fair comparison with the state-of-the-art (SOTA) methods, The input of temporal length was set to 100, and the multi-stream fusion proposed in [45] was utilized in this part. According to the multi-stream practice, the proposed DS-STGCN was trained on four input modalities which are joints (j), joints motion (jm), bone (b), and bone motion (bm), then the final result can be obtained by summering the prediction from all streams. The performance of the DS-STGCN was compared with state-of-the-art methods on NTURGB+D [12], [31] in Table VII, and Kinetics-400 [3], as well as FineGYM [32], in Table VIII. It can be observed that on most of the datasets, the proposed DS-STGCN outperforms all the compared existing methods. Compared with the SOTA GCN-based method, the proposed DS-STGCN can achieve a much more significant improvement on Kinetics-400, confirming its more powerful performance when modeling complex action movement.

V. CONCLUSION

In this work, a dynamic semantic-based spatial-temporal graph convolution network was proposed to encode the joints/edge types of the human skeleton for skeleton-based HAR, where two dynamic semantic modules were proposed to encode the semantic information implicitly in both spatial and temporal dimensions. In particular, in the dynamic semantic-based temporal graph convolution network, a causal convolution was designed, and the occurrence order of the frames was encoded during the feature aggregation. Furthermore, extensive ablation experiments have shown that the proposed two semantics encoding modules are generalized enough to be exploited in various backbones, and can introduce a positive influence on boosting HAR performance. Meanwhile, the proposed DS-STGCN outperforms the state-of-the-art methods on four challenging benchmarks, especially in modeling complex action movements, showing its impressive capability and effectiveness.

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]

REFERENCES

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

[2] L.-Y. Gui, K. Zhang, Y.-X. Wang, X. Liang, J. M. F. Moura, and M. Veloso, "Teaching robots to predict human motion," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 562–567.

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

[4] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi, "Action recognition with dynamic image networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2799–2813, 2017.

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

[6] D. Ahn, S. Kim, H. Hong, and B. C. Ko, "Star-transformer: A spatio-temporal cross attention transformer for human action recognition," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2023, pp. 3330–3339.

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

[8] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551–3558.

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

[10] R. Vemulapalli, F. Arrate, and R. Chellappa, "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*, 2014, pp. 588–595.

[11] H.-g. Chi, M. H. Ha, S. Chi, S. W. Lee, Q. Huang, and K. Ramani, "Infogcn: Representation learning for human skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 20 186–20 196.

[12] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2684–2701, 2019.

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

[14] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Learning actionlet ensemble for 3d human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 914–927, 2014.

[15] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognition*, vol. 68, pp. 346–362, 2017.

[16] R. Vemulapalli and R. Chellapa, "Rolling rotations for recognizing human actions from 3d skeletal data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4471–4479.

[17] L. Wang, D. Q. Huynh, and P. Koniusz, "A comparative review of recent kinect-based action recognition algorithms," *IEEE Transactions on Image Processing*, vol. 29, pp. 15–28, 2019.

[18] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1110–1118.

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

[20] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention lstm networks," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1586–1599, 2017.

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

[22] C. Caetano, J. Sena, F. Brémond, J. A. Dos Santos, and W. R. Schwartz, "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)*. IEEE, 2019, pp. 1–8.

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

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

[25] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel-wise topology refinement graph convolution for skeleton-based action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 359–13 368.

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

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

[28] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. 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*, 2020, pp. 1112–1121.

[29] L. Shi, Y. Zhang, J. Cheng, and H. 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*, 2019, pp. 12 026–12 035.

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

[31] A. Shahroudy, J. Liu, T.-T. Ng, and G. 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*, 2016, pp. 1010–1019.

[32] D. Shao, Y. Zhao, B. Dai, and D. Lin, "Finegym: A hierarchical video dataset for fine-grained action understanding," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2616–2625.

[33] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[34] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[35] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[36] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," *Advances in neural information processing systems*, vol. 28, 2015.

[37] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.

[38] J. Xie, Y. Meng, Y. Zhao, A. Nguyen, X. Yang, and Y. Zheng, "Dynamic semantic-based spatial graph convolution network for skeleton-based human action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 6, 2024, pp. 6225–6233.

[39] H. Duan, J. Wang, K. Chen, and D. Lin, "Dg-stgcn: Dynamic spatial-temporal modeling for skeleton-based action recognition," 2022. [Online]. Available: https://arxiv.org/abs/2210.05895

[40] Z. Qin, Y. Liu, P. Ji, D. Kim, L. Wang, R. McKay, S. Anwar, and T. Gedeon, "Fusing higher-order features in graph neural networks for skeleton-based action recognition," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[41] Z. Qin, P. Ji, D. Kim, Y. Liu, S. Anwar, and T. Gedeon, "Strengthening skeletal action recognizers via leveraging temporal patterns," in *European Conference on Computer Vision*. Springer, 2022, pp. 577–593.

[42] H. Wang and L. Wang, "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*, 2017, pp. 499–508.

[43] H. Duan, J. Wang, K. Chen, and D. Lin, "Pyskl: Towards good practices for skeleton action recognition," 2022. [Online]. Available: https://arxiv.org/abs/2205.09443

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

[45] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with multi-stream adaptive graph convolutional networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 9532–9545, 2020.

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

[47] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Richly activated graph convolutional network for robust skeleton-based action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1915–1925, 2020.

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

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

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