Multi-Scale Adaptive Large Kernel Graph Convolutional Network Based on Skeleton-Based Recognition

Yuqing Zhang[1], Chen Pang[1], Pei Geng[2], Xuequan Lu[3], Lei Lyu[1,*]
[1]School of Information Science and Engineering, Shandong Normal University
[2]School of Computer Science and Engineering, Nanjing University of Science and Technology
[3]Department of Computer Science and Software Engineering, University of Western Australia

Abstract

Graph convolutional networks (GCNs) have become a dominant approach for skeleton-based action recognition task. Although GCNs have made significant progress in modeling skeletons as spatial-temporal graphs, they often require the stacking of multiple graph convolution layers to effectively capture long-distance relationships among nodes. This stacking not only increases computational burdens but also raises the risk of oversmoothing, which can lead to the neglect of crucial local action features. To address this issue, we propose a novel multi-scale adaptive large-kernel attention graph convolutional network (MSLK-GCN) to effectively aggregate local and global spatio-temporal correlations while maintaining the computational efficiency. The core components of the network include multi-scale large kernel graph convolution module (MLKAGC), multi-channel adaptive graph convolution module (MSGC), and multi-scale temporal self-attention (MSTC) module. MLKAGC adaptively focuses on active motion regions by utilizing a large convolution kernel and a gating mechanism, effectively capturing long-distance dependencies within the skeleton sequence. Meanwhile, MSGC dynamically learns the relationships between different joints by adjusting the connection weights between nodes. To further enhance the model's ability to capture temporal dynamics, the MSTC module effectively aggregates the temporal information by integrating ECA with multi-scale convolution. In addition, we use a multi-stream fusion strategy to make full use of different modal skeleton data, including bone, joint, joint motion, and bone motion. Exhaustive experiments on three scale-varying datasets i.e., NTU-60, NTU-120, and NW-UCLA, demonstrate that our MSLK-GCN can achieves state-of-the-art performance with fewer parameters.

***Keywords:** Skeleton-based action recognition, Graph convolutional networks, Multi-scale, Large kernel attention.*

*Corresponding author: lvlei@sdnu.edu.cn

1. Introduction

Action recognition has emerged as a crucial task in computer vision, aiming to identify human actions from videos. It has been widely applied in content-based retrieval [1], video surveillance [7] and human-computer interaction [28]. Despite the progress made in RGB-based methods [26][2], these approaches often struggle with robustness, being particularly vulnerable to noise factors like variations in brightness, background, and clothing. Later, some skeleton based methods [4][11][18][41] have been developed. Unlike RGB-based approaches, the skeleton data represents the human body as a series of 2D or 3D key point coordinates. This representation not only simplifies the computational process but also enhances robustness against occlusion and background interference, making it a more reliable alternative in dynamic environments.

Deep learning methods have achieved performance improvements in skeleton action recognition, including recurrent neural networks (RNNs) [15][16], convolutional neural networks (CNNs) [37][17], graph convolutional networks (GCNs) [38][12] and transformer-based methods [20][21]. RNNs and their variants are adept at handling sequence data but fail to capture spatial relationships. CNNs, while effective at extracting spatial features, overlook temporal dynamics. Transformer-based methods, with their self-attention mechanism, can manage long-range dependencies but suffer from high computational complexity, particularly when processing long sequences or high-resolution inputs. GCNs, naturally align with the graph structure of skeleton data, allowing for efficient feature integration through message passing between skeleton joints, offering a flexible and effective solution for skeleton-based action recognition.

Despite the advancements in GCN-based methods, many existing GCN-based methods rely on a fixed adjacency matrix, which focuses primarily on capturing local relationships between neighboring nodes. To address this limitation, Shi et al. [24] introduce an adaptive graph

convolutional neural network that dynamically parameterizes the skeletal graph structure, allowing for joint learning and co-updating within the model framework. Similarly, Xia et al. [36] integrate an MD-AGCN module for adaptive graph topology and multidimensional spatial-temporal-channel analysis. To better capture global motion features, Liu et al. [5] present a multi-scale disentangled graph convolution approach and a G3D module for integrated spatial-temporal graph convolutions, employing dense cross-spacetime edges for efficient long-range modeling. However, when models stack multiple GCN layers to capture global dependencies, a challenge arises with over-smoothing. As the number of layers increases, node features begin to blend with those of neighboring nodes, causing a loss of feature distinctiveness. This blending reduces the models ability to capture crucial long-range dependencies, making it difficult to distinguish between subtle actions like "drinking water" and "putting on a hat," both of which involve hand-head interactions but require the perception of fine-grained variations in joint relationships. To cope with this defect, researchers introduced attention models to effectively capture the long-distance relations. Plizzari et al. [8] use the Transformer self-attention operator to model dependencies between joint points more effectively. Liu et al. [13] leverage a novel partition-aggregation temporal Transformer for effective long-range dependency. However, these attention-based methods often overlook the synergies between local joints and fail to fully integrate local and global information. To address this issue, Chen et al. [3] combine GCNs with a Transformer architecture, creating a Pyramid Spatial-Temporal Graph Transformer that alternates between local and global information processing. While this design improves performance, it often requires a large number of parameters to effectively model interactions across spatial and temporal domains.

To tackle the aforementioned questions, we propose a novel multi-scale adaptive large-kernel attention graph convolution network (MSLK-GCN), designed to effectively aggregate global and local features while maintaining the computational efficiency. The network is composed of three key modules: multi-scale large-kernel convolution (MLKAGC) module, adaptive graph convolution (MSGC) module, and multi-scale time self-attention (MSTC) module. Specifically, multi-scale large-kernel convolution (MLKAGC) module captures both local and global dependencies by leveraging a multi-scale large-kernel attention operator (MLKA) and a simplified gated spatial attention unit (GSAU). MLKA is responsible for extracting features across different scales and fusing them to reduce the risk of over-smoothing, ensuring that the model can still capture long-range dependencies effectively. In parallel, the GSAU reduces parameter complexity while adaptively enhancing the model's focus on key skeleton features, optimizing the balance between complexity and performance. Building on the local-global feature extraction from MLKAGC, adaptive graph convolution (MSGC) module further refines the models representation by adaptively learning the skeleton's topology for each GCN layer and sample. This dynamic adjustment ensures that the model can better emphasize critical behavioral features while also improving local feature aggregation. To further enhance the model's ability to capture temporal dynamics, the MTC module combines multi-scale time convolutions with an efficient channel attention. This allows it to integrate long-term temporal dependencies with the spatial features extracted by the MLKAGC and MSGC modules, achieving a more comprehensive representation of both spatial and temporal information. Additionally, our model employs a multi-stream framework to explicitly capture the second-order information of the skeleton data, effectively combining it with first-order information.

The primary contributions of this work are summarized as follows:

- We propose a multi-scale adaptive large kernel attention graph convolutional network (MSLK-GCN) that effectively aggregates both global and local motion features, improving the expression ability of the model while maintaining the computational efficiency of the model.
- We introduce a multi-scale large kernel graph convolution network (MLKAGC), which integrates a multi-scale large kernel attention module (MLKA) and gating unit (GSAU). This design mitigates the over-smoothing problem associated with excessive stacking of graph convolution layers while effectively aggregating global motion features through large kernel convolution blocks.
- We design an adaptive graph convolutional network (MSGC) and a multi-scale temporal self-attention (MSTC) module. The MSGC adaptively learns the topological structure of the graph to emphasize key local action features, while the MTC module effectively aggregates the temporal information by integrating an efficient channel attention with multi-scale convolution.

2. Related Work

2.1. Skeleton-based action recognition

The representation of skeleton data significantly influences the performance of action recognition. Early works [29][30] typically relied on handcrafted features, which required expert knowledge for their design and might not capture the full complexity of the data. With the advance-

ment of deep learning, automatic feature extraction methods [4][12] have gradually become the mainstream.

Skeleton recognition methods based on graph convolutional networks (GCNs) have been widely applied. Notably, there are two main variants of GCN-based approaches: spectral-based GCNs and spatial-based GCNs. The spectral approach utilizes eigendecomposition of the graph Laplacian matrix, whereas the spatial approach operates directly on the adjacency structure of the graph. Yan et al. [38] have introduced an innovative framework known as Spatial Temporal Graph Convolutional Networks, which surpasses traditional approaches by autonomously discerning spatial and temporal dynamics within datasets. Shi et al. [24] proposed the adaptive graph convolutional neural network, which constructed a two-stream framework to explicitly use the second-order information of skeleton data, and parameterize the graph structure of skeleton data and embed it into the network for joint learning and updating with the model. Li et al. [12] introduced an encoder-decoder architecture that stacks action-structure graph convolution and temporal convolution as the basic building blocks, simultaneously learning spatial and temporal features for action recognition, capturing richer dependencies. Chen et al. [4] introduced the innovative CTR-GC model for skeleton-based action recognition, enabling dynamic topology adaptation and efficient feature integration across channels. Wen et al. [34] utilized sample-dependent latent relations and hierarchical structures in human skeletal data, along with efficient local and non-local temporal blocks, to enhance action recognition performance. Lee et al. [11] proposed an architecture of hierarchical decomposition graph convolutional network with a novel hierarchical decomposition graph , which can extract both primary structural adjacency and distant edges, and utilize them to construct an HD-Graph that includes these edges within the same semantic space of human skeleton data. Liu et al. [19] proposed a multi-scale aggregation scheme to distinguish the importance of nodes in different neighborhoods for effective long-range modeling, and proposed that the G3D module uses dense cross-temporal edges as skip connections for direct information propagation across time-space graphs. Song et al. [27] designed a composite scaling strategy to synchronously expand the model's width and depth, ultimately achieving a cluster of efficient GCN baselines with high accuracy and a small number of trainable parameters. While these studies have achieved satisfactory results in addressing the recognition of distant joints, the inherent limitations of Graph Convolutional Networks (GCNs) have led many models to rely on excessive stacking of convolutional layers to achieve recognition performance. This approach significantly increases the likelihood of feature over-smoothing in models following multiple layers of stacking.

Concurrently, methods based on the Transformer architecture have also gained increasing popularity. Plizzari et al. [20] used the Transformer self-attention operator to model the dependencies between joints, and combined the spatial and temporal self-attention modules in a two-stream network to further enhance the performance of the model. Liu et al. [23] leverage a novel partition-aggregation temporal Transformer for effective long-range dependency and subtle temporal structure capture, along with a topology-aware spatial Transformer for enhanced spatial correlation modeling. While these models have achieved satisfactory results, they primarily address the recognition of distant joints, somewhat neglecting the local coordination among joints and failing to effectively integrate the two aspects.

2.2. Multi-scale large kernel attention

The self-attention mechanism was originally designed for natural language processing tasks. Due to its excellent feature capture ability, it has gradually emerged in the field of computer vision. However, the complexity of image data brings great challenges to the application of self-attention mechanism, especially the multi-dimensional characteristics of skeleton data. Compared with the use of multiple small convolution kernels to gradually increase the receptive field, the direct use of large convolution kernels can achieve the same effect in fewer layers, thereby reducing the number of parameters and calculations of the model. Guo et al. [8] proposed a new linear attention mechanism, called large kernel attention (LKA), to achieve adaptive and long-distance association of the model while avoiding the defects of self-attention. In order to solve the problem of secondary increase in computation and memory occupation caused by the increase of convolution kernel size in the deep convolution layer of LKA module, Lau et al. [10] proposed a new large separable kernel attention module for visual attention network. By decomposing the two-dimensional convolution kernel of the deep convolution layer into cascaded horizontal and vertical one-dimensional convolution kernels, the computational and memory requirements are reduced. Wang et al. [33] proposed a multi-scale attention network, which improves the large-core attention through multi-scale and gating schemes to obtain rich attention maps of different granularities to improve the performance of convolutional neural networks in super-resolution tasks.

Recent work has achieved satisfactory results by combining large-core attention with skeleton behavior recognition. Liu et al. [18] introduced large kernel attention into skeleton-based action recognition tasks to model long-distance dependencies. However, it is worth noting that LKAGCN may focus more on the capture of global features and ignore the aggregation of local features. In action recognition, local features (such as small movements of hands or feet) are also very important, especially in complex action sequences.

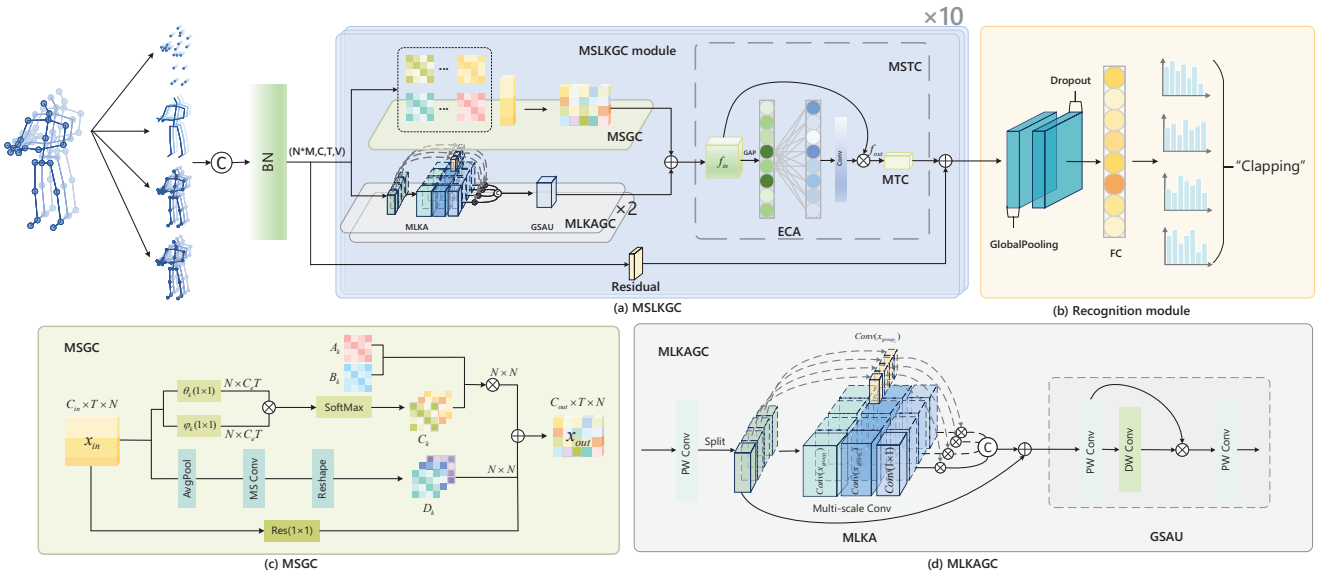

Fig.1. Framework of the proposed multi-scale adaptive large kernel attention graph convolutional network. (a): Pipeline of the MSLK-GCN, comprising of 10 MSLKGC module, and each module includes MSGC, MLKAGC, and MSTC. (b): The prediction part of the recognition module. (c): The architecture of the MSGC. (d): The structure of the MLKAGC.

3. Method

In this section, we first describe the overview of MSLKGC in Section 3.1. Then we describe the overall architecture of MLKAGC in Section 3.2. Following that, the detailed structure of MSGC and MSTC modules are introduced in Section 3.3 and Section 3.4, respectively. Finally, we introduce the multi-stream input structure in Section 3.5.

3.1. Overview

The overall architecture of the MSLKGC model is shown in Fig.1. Our model comprises of 10 spatio-temporal modules, organized into three stages with 4, 3, and 3 modules, respectively. Each module includes a spatial and a temporal component. The spatial module is composed of two MLKAGC and one MSGC, which are used to extract the global and local information of the skeleton respectively. In the MLKAGC module, there are two submodules, MLKA and GSAU. Input features are fed into each sub-module, where convolution kernels of different sizes capture multi-scale features. The spatial attention mechanism enables the model to focus on the most important area for skeleton recognition, filtering out irrelevant noise. In the MSGC module, the adaptive graph convolution block dynamically adjusts the key joint weight information from the skeleton data. By combining MLKAGC and MSGC, the module generates global and local features, enabling effective feature fusion into a new feature representation. This new feature is then passed into the multi-scale time self-attention convolution module (MSTC), which leverages an efficient channel attention mechanism and a multi-scale temporal module for time feature aggregation.

3.2. Multi-scale adaptive large kernel attention graph convolution

Traditional GCNs improve their ability to capture global features through multi-layer stacking. However, as the number of layers increases, convolutions progressively blend neighboring node features, which reduces feature distinctiveness and leads to over-smoothing. This can obscure important long-range dependencies. To address this and efficiently capture global skeletal information without encountering the over-smoothing problem, we introduce the MLKAGC module. This module combines multi-scale large kernel attention (MLKA) with a gating spatial attention unit (GSAU), using large receptive fields and gating mechanisms to consolidate indirect dependencies, as shown in Fig.1(d).

Multi-scale Large Kernel Attention (MLKA). In the MLKA module, we effectively aggregate long-distance joint points by combining large kernel decomposition and multi-scale learning. Firstly, we transform the input feature into a high-level feature $x \in R^{C\times T\times V}$, and use 1×1 convolution to change the number of channels, which is denoted as $\mathrm{Conv}_{1\times1}$, as shown in Eq.1:

$$x' = \mathrm{Conv}_{1\times1}(x) \tag{1}$$

Then, x' is divided into G groups according to the number of channels, and the number of channels in each group is $\frac{C}{G}$. Among them, in order to better adapt to the skeleton data, we set the G of the first layer to 3, and the G of the 2-10 layers to 4. Specifically, in the first layer of graph convolution, x' is divided into three groups, each processed using different sizes of large kernel convolution $(3\times3, 5\times5, 1\times1)$,$(5\times5, 7\times7, 1\times1)$, and $(7\times7, 9\times9, 1\times1)$.

Additionally, varying dilation rates of 2, 3, and 4 are applied to capture features at different scales effectively. Different from the former, the fourth group uses different sizes of large kernel convolution $(9 \times 9, 11 \times 11, 1 \times 1)$ and different expansion rates 8. After multi-scale convolution calculation, a_{g_i,k_i} is obtained. The process is shown in Eq.2 and Eq.3:

$$x_g \in \{x_{g_1}, x_{g_2}, x_{g_3}, x_{g_4}\}, x_{g_i} \in R^{C \times T \times V} \tag{2}$$

$$a_{g_i,k_i} = \mathrm{Conv}_{k_i \times k_i, d_i}(x_{g_i}) \tag{3}$$

where x_{g_i} is the data of group i, and $a_{(g_i,k_i)}$ represents the data after the convolution operation, $Conv_{(k_i \times k_i, d_i)}$ represents the convolution operation, where k_i is the convolution kernel size used in group i, and d_i is the corresponding expansion rate.

At the same time, deep separable convolution (DWConv) is used for each set of features x_{g_i}. Deep separable convolution first performs spatial convolution independently on each input channel and then merges the results, effectively capturing spatial features. After that, each group a_{g_i} employs the gating mechanism that uses the gating weight γ_{g_i} to adjust the original packet data x_{g_i}:

$$\gamma_{g_i} = \sigma\,(\,DWConv\,(\,a_{g_i}\,)\,) \tag{4}$$

$$x^{'}_{g_i} = x_{g_i} \odot \gamma_{g_i} \tag{5}$$

where $\odot$ represents the element-by-element multiplication, and σ is the Sigmoid activation function.

All the modulated feature groups $x^{'}_{g_i}$ are stitched together to form a complete feature tensor x_{mod} (Eq.6).

$$x_{mod} = Concat\,(\,x^{'}_0, x^{'}_1, \ldots, x^{'}_{G-1}\,) \tag{6}$$

By multiplying the spliced feature tensor x_{mod} with the initial aggregated high-level feature $x^{'}$ element by element, $x^{'}_{mod}$ is obtained. The process is shown in Eq.7:

$$x^{'}_{mod} = x_{mod} \odot x^{'} \tag{7}$$

Finally, another 1×1 convolution layer is used to map the modulated feature $x^{'}_{mod}$ back to the original channel number C, and the final output Eq.8 is obtained:

$$MLKA\,(\,x\,) = Conv_{1\times1}\,(\,x^{'}_{mod}\,) \tag{8}$$

Gated Spatial Attention Unit (GSAU). GSAU uses simple spatial attention and gated linear units to reduce parameters and calculations by using simpler structures, allowing the model to adaptively adjust the response of the feature map to better capture spatial information. We assume that the output data of $MLKA(x)$ is y, and y is introduced into GSAU for gating unit calculation. Firstly, for the input data $y \in R^{(C \times T \times V)}$, the normalized feature is mapped to a new feature space by using the 1×1 convolution layer, and the output is divided into two parts $y^{'}_1$ and $y^{'}_2$ according to the number of channels, as shown in the formula:

$$(\,y_1, y_2\,) \in Conv_{1\times1}\,(\,y\,) \tag{9}$$

where y_2 is normalized to obtain $y^{'}_2$, and $y^{'}_2$ is processed by depthwise separable convolution DWConv, such as formula:

$$y^{''}_2 = DWConv\,(\,y^{'}_2\,) \tag{10}$$

where $y^{''}_2$ is multiplied with y_1 element by element to obtain $y^{'}$, as shown in the formula:

$$y^{'} = y^{''}_2 \odot y_1 \tag{11}$$

After 1×1 convolution of the obtained $y^{'}$, Y is obtained by adjusting the layer normalization and scaling parameters, as shown in the formula:

$$Y = LayerNormal\,(\,Conv_{1\times1}\,(\,y^{'}\,)\,) \tag{12}$$

Finally, the adjusted feature Y is added to the original input y to obtain the final output, as shown in the formula:

$$GSAU\,(y) = Y + y \tag{13}$$

3.3. Multi-channel adaptive graph convolution

In skeletal data, joint positions and relationships are action-dependent, with temporal variations in skeleton data exposing action sequences and dynamics. Fixed topologies may inadequately capture these dynamics, constraining model adaptability to changes. Thus, we introduce the MSGC module, integrating spatial relationship modeling via parallel structures to enhance multi-scale feature extraction of topological structures, thereby bolstering the model's capacity for precise recognition and interpretation of complex spatial-temporal relationships in action recognition tasks.

The previous method based on STGCN uses a predefined graph to perform graph convolution on the skeleton data, and uses the formula to implement:

$$f_{out} = \sum_k^{K_v} W_k f_{in} A_k \tag{14}$$

where, f represents the feature map, $f_{in} \in R^{(C_{in} \times T \times N)}$ is the input feature, $f_{out} \in R^{(C_{out} \times T \times N)}$ is the output feature, N and T represent the number of joints and frames in the skeleton data, C_{in} and C_{out} represent the input channel and output channel respectively. $A_k = \wedge_k^{-\frac{1}{2}} \left(\bar{A}\right)_k \wedge_k^{-\frac{1}{2}}$, where $\bar{A}_k$ is equivalent to the $N \times N$ adjacency matrix, and the element $\bar{A}_k^{ij}$ represents whether the vertex is in the subset S_{ik}

of v_i. However, the fixed skeleton data graph is not suitable for identifying skeleton actions. Therefore, as shown in the Fig.1(c), this paper proposes a multi-channel adaptive graph convolution module (MSGC). By introducing the parameterized adjacency matrix B_k, the adaptive graph C_k and the dynamic adjacency matrix D_k, the adaptive graph structure is generated by modifying the formula as show in Eq.15:

$$f_{out} = \sum_{k}^{K_v} W_k f_{in} \left(A_k + B_k + C_k + D_k \right) \tag{15}$$

The key difference between Eq.14 and Eq.15 is reflected in the adjacency matrix. A_k, B_k, C_k, and D_k are variables used to describe different parts of the graph topology in the adaptive graph convolutional network. Specifically, A_k represents the original adjacency matrix, which represents the physical structure of the human skeleton, is set manually and is fixed in all layers and input samples. This adjacency matrix reflects the natural connection between human joints, for example, it may represent the connection between the arm joint and the shoulder joint. B_k is a learnable adjacency matrix whose elements are parameterized and optimized with other parameters during the training process. Unlike A_k, B_k has no fixed value limit, which means that the structure of the graph can be fully learned from the training data. B_k can represent any value, which not only indicates whether there is a connection between two joints, but also indicates the strength of the connection. C_k is a data-dependent graph that learns a unique graph for each sample. It is calculated by the SoftMax function based on the input feature graph, which represents the similarity between vertices, so as to learn the data-specific graph structure:

$$C_k = Softmax \left(f_{in}^T W_{\theta k}^T W_{\emptyset k} f_{in} \right) \tag{16}$$

where W_θ and W_ϕ are the parameters of θ and in the embedded function.

D_k is a dynamic adjacency matrix, and its elements obtain learning parameters through multi-scale feature transformation during training, which enhances the flexibility of the graph. The multi-scale feature transform is calculated by convolution of $(5 \times 5, 3 \times 3, 1 \times 1)$, which is denoted as MSConv, and expressed as:

$$D_k = MSConv \left(f_{in} \right) \tag{17}$$

3.4. Multi-scale temporal self-attention convolution

In this section, we introduce the self-attention time convolution module of the model in detail. As shown in Fig.2, we embed the efficient channel attention (ECA) module between the spatial convolution module and the temporal convolution module. This layout enables an efficient channel attention to more effectively aggregate information from

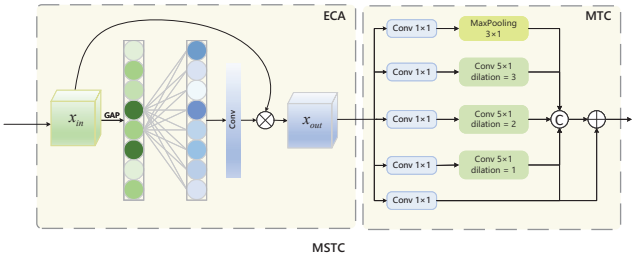

Fig.2. The basic block of our MSTC, which includes ECA and MTC modules.

spatial and temporal dimensions, thereby enhancing the feature extraction ability of the model.

Efficient Channel Attention (ECA). Inspired by [32], we consider that the model's ability to extract features can be enhanced by adaptively emphasizing the salient channels in the input feature map. By learning the importance weight of each channel, ECA can highlight those feature channels that are more discriminative for identifying skeleton actions while suppressing less important channels. This helps the model to focus more on useful information in local features, thereby improving the accuracy of recognition.

Firstly, according to the number of input channels C, the kernel size K is calculated, as shown in the formula:

$$K = \left[\frac{log_2 C + b}{\gamma} \right] + 1 \tag{18}$$

If K is even, then add K to 1 to ensure that it is an odd kernel.

After that, we use average pooling to perform channel compression on the input data x_{in} to generate a feature map with a shape of $(N, 1, T, V)$. After that, the channel weight y is calculated by the convolution kernel activation function. Finally, the channel weight y is extended back to the original channel number and multiplied by the input feature map x_{in} to achieve channel weighting. The formula can be expressed as:

$$x_{out} = x_{in} \odot (\sigma \left(Conv \left(AdaptiveAvgPool2d \left(x_{in} \right) \right) \right))_{exp} \tag{19}$$

where exp is the channel weight y, which is obtained through the expansion operation expand to match the channel dimension of x_{in}. $\odot$ represents the multiplication by elements.

Multi-scale temporal convolution (MTC). Since the ability of GCN to extract information relies on effective feature decomposition, a complex network structure with a large number of parameters in skeleton graph input can result in feature redundancy, significantly increasing the models computational and storage costs. Additionally, most current temporal modules use single-scale aggregation, which limits their ability to aggregate complex skeleton data. To address this, as shown in Fig.2, we propose a more streamlined multi-scale time module that reduces model complexity without compromising performance.

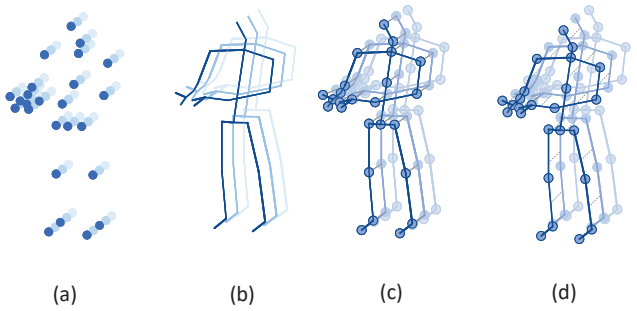

Fig.3. Multi-stream fusion skeleton diagram. Visualization of multi-stream fusion strategy, which includes bone, joint, joint motion, and bone motion. We take the "clap" action as an example. (a) The point represents the human joint. (b) The line segment represents the human skeleton. (c) The solid line connecting the joint points between adjacent frames represents the joint motion. (d) The dashed line connecting the bone points between adjacent frames represents the bone motion.

3.5. Multi-stream fusion strategy

In order to realize the fusion of multi-level information to further improve the performance of our MSLK-GCN in skeleton recognition tasks. We adopt a multi-stream fusion strategy to model joint and bone information and their motion information simultaneously in a multi-stream framework. As shown in the Fig.3, we adopt a multi-stream fusion strategy and perform a series of processing on the body joints in human skeleton data. Specifically, Fig.3 (a) and (b) represent human joint and bone data, we assume that the source joint is $j_{t,i} = (x_{t,i}, y_{t,i}, z_{t,i})$, and the target joint is $j_{t,j} = (x_{t,j}, y_{t,j}, z_{t,j})$, which are defined as the joint near the center of gravity of the skeleton and the joint far from the center of gravity, respectively. Where, from the source joint to the target joint $b_{t,i,j} = (x_{t,j} - x_{t,i}, y_{t,j} - y_{t,i}, z_{t,j} - z_{t,i})$ represents the skeletal coordinates. Fig.3 (c) and (d) represent joint motion and skeletal motion data, respectively. For motion data, it is defined as the coordinate difference of the same joint or bone in a continuous frame. For example, given the joint point $j_{t,i}$ and the joint point $j_{(t+1),i}$ of adjacent frames, the joint motion of adjacent frames can be expressed as $m_{t,t+1,i} = (x_{t+1,i} - x_{t,i}, y_{t+1,i} - y_{t,i}, z_{t+1,i} - z_{t,i})$. Similarly, the skeleton coordinates of adjacent frames can be expressed as $m_{t,t+1,i,j} = b_{t+1,i,j} - b_{t,i,j}$. Finally, we integrate the bones, joints and their movements into the four streams, and use the weighted method to fuse the scores of the four streams to obtain the final prediction results.

4. Experiments

In this section, in order to comprehensively evaluate the effectiveness of our MSLK-GCN, we conduct extensive experiments on the NTU-RGB+D-60, NTU-RGB+D-120, and Northwestern-UCLA datasets.

4.1. Datasets

NTU-RGB+D-60(NTU-60) [22]. The NTU-60 [22] comprises an extensive collection of 3D human action samples, captured utilizing Kinect sensors, and is frequently utilized for action recognition tasks. The complete NTU-RGB+D dataset comprises 56,880 skeletal motion sequences across 60 action categories. The collection encompasses depth data, 3D skeletal structures, color frames, and thermal imagery sequences. It comprises the spatial coordinates of 25 principal skeletal joints for each frame, with a maximum of two individuals per frame. Two distinct evaluation benchmarks exist: the Cross-subject (X-Sub) and the Cross-setup (X-Set) configurations.

NTU-RGB+D-120(NTU-120) [14]. NTU-120 [14] extends NTU-60, adding another 60 classes and another 57,600 video samples, that is, NTU-120 has a total of 120 classes and 114,480 samples. The authors of this dataset recommend two benchmarks: (1) Cross-Subject (X-Sub): Similar to NTU-60, 53 subjects were used as training data and the remaining 53 subjects were used as validation data. (2) Cross-Setup (X-Set): Use the setting of even id as training data, and use the setting of odd id as validation data.

Northwestern-UCLA(NW-UCLA) [31]. The dataset is a multi-view 3D action recognition dataset, which focuses on capturing RGB, depth information and human skeleton data of human actions from multiple perspectives. It uses three Kinect cameras to capture RGB, depth and human skeleton data simultaneously. It contains 1494 video clips covering 10 categories, each performed by 10 actors. We use the same evaluation protocol in [31], using samples from the first two cameras as training data, and samples from the other camera as test data.

4.2. Implementation Details

Our experiments are implemented on the PyTorch deep learning framework. The stochastic gradient descent (SGD) momentum is set to 0.9. The cross-entropy function is employed to calculate the loss.

In addition, all experiments are carried out on NVIDIA GeForce RTX 3090 GPU. All skeleton sequences are normalized to 64 frames. For samples with less than 64 frames, we fill them by repeating the existing frames., and for samples with more than 64 frames, we cut them.

For NTU-60 and NTU-120, the batch size is set to 64, the learning rate starts at 0.1, and then divides by 10 at the 35^{th}, 55^{th}, and 65^{th} rounds. The training process ends at the 300th round, the weight attenuation is set to 0.0004, and the batch size is set to 64. For Northwestern-UCLA, the weight decay is set to 0.0001. Other parameter settings are the same as the NTU-60 dataset.

Methods	Year	NTU-60		NTU-120		#Param.(M)
		X-Sub(%)	X-View(%)	X-Sub(%)	X-Set(%)	
ST-LSTM [15]	2016	69.2	77.7	58.2	60.9	-
GCA-LSTM [16]	2017	74.4	82.8	-	-	-
AGC-LSTM [25]	2019	95.0	89.2	-	-	-
Synthesized CNN [17]	2017	80.0	87.2	60.3	63.2	-
Ta-CNN [37]	2022	90.4	94.8	85.4	86.8	0.53†
ST-GCN [38]	2018	81.5	88.3	70.7	73.2	3.10∗
DGNN [23]	2019	89.9	96.1	-	-	26.24†
AS-GCN [12]	2019	86.8	94.2	77.9	78.5	9.50∗
2s-AGCN [24]	2019	88.5	95.1	82.9	84.9	6.94∗
MS-G3D [19]	2020	91.5	96.2	86.9	88.4	6.40∗
Shift-GCN [5]	2020	90.7	96.5	85.9	87.6	10.00 †
SGN [39]	2020	89.0	94.5	79.2	81.5	0.69†
CTRGCN [4]	2021	92.4	96.8	88.9	90.6	1.46†
EfficientGCN-B4 [27]	2022	92.1	96.1	88.7	88.9	1.10†
SMotif-GCN+TBs [34]	2022	90.5	96.1	87.1	87.7	-
LKAGCN(2s) [18]	2023	90.7	96.1	86.3	87.8	-
MS-TEGCN [9]	2023	91.4	96.6	86.5	88.0	15.80†
GSTLN [6]	2023	91.9	96.6	88.1	89.3	1.50†
MCTM-Net [35]	2024	92.8	96.8	89.3	91.0	-
STTR [20]	2021	89.9	96.1	84.3	86.7	12.46∗
STTFormer [21]	2022	92.3	96.5	88.3	89.2	5.70∗
TranSkeleton [13]	2023	92.8	97.0	89.4	90.5	2.20†
STD-Transformer [40]	2024	92.6	96.4	88.9	90.8	-
MSLK-GCN (Ours)		**93.5**	**97.2**	**89.8**	**91.8**	4.75

Table 1. Accuracy (%) comparison of classification accuracy with existing methods on NTU-60 and NTU-120 dataset. ∗: These results are implemented based on the released codes. †: These results are provided by their authors.

4.3. Comparison with state-of-the-art methods

To verify the superiority of our proposed model, we compare it with the state-of-the-art methods on NTU-60, NTU-120 datasets and NW-UCLA.

Results on the NTU-60 dataset: As shown in the Table 1, we compare methods based on LSTM, CNN, GCN, and Transformer. Among them, the LSTM-based method has [15][16][25], the CNN-based method has [37][17], the GCN-based method has [4][18][38][12][34] and the Transformer-based methods have [20][21][13][40]. In the LSTM-based methods, our model is 24.3% and 19.5% higher than the ST-LSTM [15] method on the X-Sub and X-View benchmarks, respectively. In the CNN-based methods, our model is 3.1% higher than the Ta-CNN [37] on the X-Sub benchmark. In the Transformer-based method, STTR [20] uses the Transformer self-attention operator to model the dependencies between joints, and uses the spatial and temporal self-attention modules to model the skeleton information. Compared with ST-TR, our model improves 3.6% on the X-Sub benchmark test set, and the parameter quantity decreases by 7.71M. This is due to the multi-scale large kernel graph convolution module fused in our method, which better captures the long-range dependence of long-distance joints with fewer parameters. In the GCN-based methods, SMotif-GCN [34] employs a multi-scale enhanced graph convolution module to comprehensively capture the feature representations of joints and bones. Our model is 3.0% and 1.1% higher than it on the X-Sub and X-View benchmarks, respectively, because our multi-scale large kernel graph convolution module overcomes the over-smoothing problem caused by over-stacked graph convolution by extracting features of different scales and fusing them. LKAGCN [18] effectively models the long-range dependency of the skeleton by introducing the large kernel attention module into the skeleton-based action recognition task. On the X-Sub benchmark, it is 2.8% lower than our method. This is because while our model introduces a multi-scale large kernel graph convolution module, the proposed adaptive adjacency matrix module enhances the model's ability to express the key behavior features in the skeleton by learning the graph topology of different GCN layers and skeleton samples, so as to better identify different skeleton actions.

Results on the NTU-120 dataset: In order to make the results more reliable, we compare the model with LSTM-based [15], CNN-based [37][17], GCN-based [4][18][38][12] and Transformer-based [20][21][13][40] approaches. As shown in the Table 1, the effectiveness of our model is proved. Our model achieves 89.8% and 91.8% on the X-Sub and X-Set benchmarks, respectively. At sim-

Methods	Year	NW-UCLA(%)
AGC-LSTM [25]	2019	93.3
Synthesized CNN [17]	2017	92.6
Ta-CNN [37]	2022	96.1
Shift-GCN [5]	2020	94.6
SGN [39]	2020	92.5
CTRGCN [4]	2021	96.5
HDGCN [11]	2023	97.2
GSTLN [6]	2023	94.8
MCTM-Net [35]	2024	97.2
MSLK-GCN (Ours)		**97.8**

Table 2. The classification accuracy is compared with the existing methods on the NW-UCLA dataset. The top-1 accuracy is listed.

ilar accuracy, our model parameters are significantly lower than other SOTA models. We consider that this improvement is due to the fusion of the multi-scale large kernel graph convolution module in our method, and the concise multi-scale time self-attention convolution module. The concise and effective feature aggregation uses a lower parameter amount to ensure the performance of the overall model.

Results on the NW-UCLA dataset: As shown in the Table 2, the accuracy of our model on NW-UCLA dataset is 97.8%, which is 4.5% higher than Ensemble AGC-LSTM [25] and 5.2% higher than Synthesized CNN [17]. These methods capture the dynamic changes of actions by processing time series and extracting spatial features, respectively. CTR-GCN [4] uses channel-level topological refinement graph convolution to dynamically learn different topological structures and effectively aggregate joint features in different channels. Our method demonstrates a competitive performance, outperforming it by 1.3%. GSTLN [6] combines GSTL with the time modeling unit to generate the global spatio-temporal collaborative topology of the joint. Our method improves by 3.0% over it, benefiting from the integration of our MLKAGC and MSGC module, which aggregate the global features of the skeleton while ensuring the effective aggregation of local features. Therefore, comparison with the above methods on UCLA datasets, our method not only applies to large-scale datasets, but also achieves excellent performance on small-scale datasets.

4.4. Ablation Study

In this section, in order to demonstrate the effectiveness of our proposed MSLK-GCN, we conduct ablation experiments on three datasets, including NTU-60, NTU-120 and NW-UCLA datasets.

4.4.1 Discussion of configuration

Evaluation of the Modules. To verify the effectiveness of each module of our proposed model, we set up ablation experiments on the X-Sub benchmark dataset of NTU-60. We replace the adjacency matrix of ST-GCN [38] with a parametrizable adjacency matrix to establish the baseline model.

Model Configurations	Par.	Acc(%)
Baseline	2.09	85.41
Baseline+MSGC	3.14	90.80(↑5.39)
Baseline+MLKAGC	3.99	88.55(↑3.14)
Baseline+MSTC(ECA+MTC)	2.53	88.62(↑3.21)
+1MLKAGC+MSGC	4.31	92.61(↑7.20)
+2MLKAGC+MSGC	4.31	92.70(↑7.29)
+3MLKAGC+MSGC	4.31	90.90(↑5.49)
MSLK-GCN w/o res	4.75	92.94(↑7.53)
MSLK-GCN w/o ECA	4.75	92.71(↑7.30)
MSLK-GCN(Ours)	**4.75**	**93.52(↑8.11)**

Table 3. The number of parameters and prediction accuracy of the model under different module combinations. MLKAGC stands for Multi-scale Large Kernel Attention Graph Convolution, MSGC refers to the Multi-channel Adaptive Graph Convolution module. Par.: Indicates Params (M).

The results are shown in Table 3. First, we add the MSGC, MLKAGC, ECA, and MTC modules to the baseline model individually and observe significant improvements in accuracy, confirming the effectiveness of each module.

Then, to further explore the effectiveness of different model architectures, we replace the GCs module in the baseline model with various architectures. The results indicate that although the parameter counts of the different aggregation modules are similar, the accuracy of MSLK-GCN increases by 8.11%. Moreover, all model configurations outperform the baseline model, demonstrating that our model significantly enhances recognition performance without significantly increasing the number of parameters.

Discussion of parameters. When the GNN model captures the complex relationships between nodes in the graph, this may lead to the need for more computing resources for model training. Although the Transformer-based methods can effectively enhance the recognition of long-distance joints in the skeleton, it has high computational costs due to its complex self-attention mechanism when dealing with skeleton data with high complexity and diversity. The MLKAGC module reduces the computational complexity by decomposing the large kernel convolution into multiple small kernel convolutions. This decomposition method can reduce the number of parameters and improve the computational efficiency of the model. Therefore, we set up ablation experiments on the X-Sub benchmark dataset of NTU-60 and compared the model with parameters based on GNN, GCN, and Transformer methods, as shown in Fig.4. It can be seen that our model has obvious advantages in computing and storage overhead.

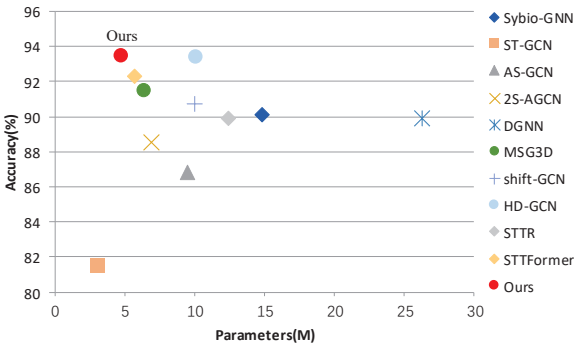

Fig.4. Comparison of model size between our method and state-of-the-art methods on the NTU-60 dataset under the xsub benchmark. Our model has obvious advantages in terms of computational and storage overhead.

4.4.2 Effectiveness of multi-scale adaptive large kernel attention

As shown in the Fig.5, we select 10 actions involving long-distance joint dependence in the NTU-60 dataset and visualize the recognition results on the X-Sub benchmark. Among them, it can be clearly seen that our MLKAGC module has obvious advantages in identifying such actions, which shows that the MLKAGC module can extract the global features of the image through multi-scale large kernel convolution, and effectively capture the large-scale information in the image.

4.4.3 Effectiveness of multi-stream fusion strategy

We design a multi-stream fusion structure, which inputs four types of data into the model after fusion. In order to verify the effectiveness of the long-term fusion strategy, we combined different input data and conducted four sets of experiments with MSLK-GCN. The Table 4 shows the specific performance comparison of different combinations. The strategy we employed achieves the highest accuracy on all three datasets. Obviously, the multi-stream fusion strategy is superior to the single-stream fusion strategy, and the fusion strategy we use by fusing the four data of bone, joint, bone action and joint action effectively improves the accuracy and efficiency of action recognition.

4.5. Visualization and discussion

4.5.1 Visualization of adaptive adjacency matrix

In order to more intuitively display the different features learned by the model, we visualize the trainable adjacency matrices corresponding to layers 1, 3, 5, 7, 9, and 10. Fig.6(a-c) represents the original predefined adjacency matrix, and Fig.6(d-i) represents the output adjacency matrix after each layer of training. We can clearly find that a richer motion relationship between joints is extended in the trained adjacency matrix. Therefore, our MSLKGC module effectively improves the recognition ability of these nuances by strengthening the capture of local features, thus showing its

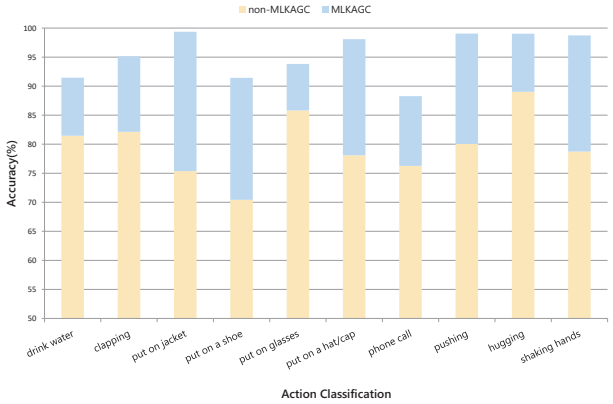

Fig.5. The effectiveness of MLKAGC module. We select 10 actions involving long-distance joint points. Yellow represents the result of removing the MLKAGC module, and blue represents the result of adding the MLKAGC module.

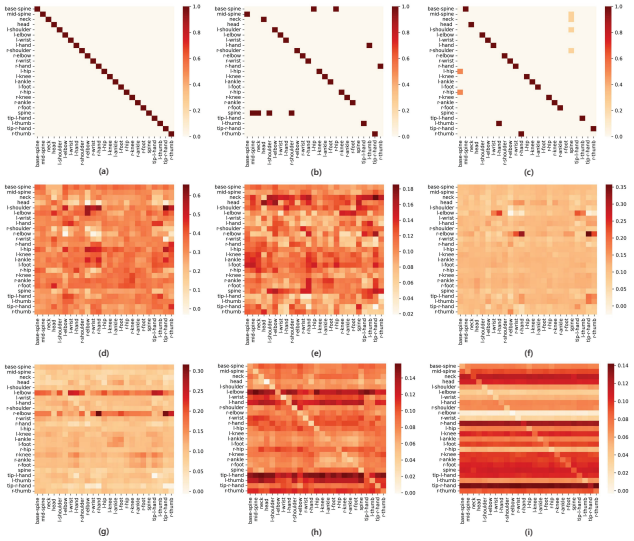

Fig.6. Visualization of the trainable adjacency matrixes corresponding to different layers of MSGC module. (a)-(c): The initial predefined adjacency matrix. (d)-(i): Corresponding to the adjacency matrix of 1, 3, 5, 7, 9, and 10 layers after training respectively.

unique advantages in behavior recognition tasks.

In addition, we visualize the "clapping" action to prove the recognition performance of our model in long-distance joints in Fig.7. It can be seen that although the left and right hands belong to the non-physically connected long-distance joint points, the weight of the two is significantly higher than that of the other joint points of the body in the heat map.

At the same time, we select two actions of "drink water" and "eat meal" and visualize the adjacency matrix heat map after their training Fig.8. In the "drink water" action, it can be seen that the "left hand" and "head" with far joint distance have higher weights, while in the "eating" action, "between the left and right hands" and "head" have higher weights, which indicates that our model does not ignore local fine-grained actions while identifying long-distance joint points.

Multi-stream	NTU-60		NTU-120		NW-UCLA
	X-Sub(%)	X-View(%)	X-Sub(%)	X-Set(%)	Top-1(%)
J	80.16	80.63	80.65	80.98	94.87
B	80.84	81.15	81.36	81.86	93.53
M	81.03	80.10	80.14	80.12	94.18
J+B	90.87	91.74	88.02	89.56	96.47
J+M	89.15	90.15	87.26	88.40	94.55
B+M	89.47	91.87	88.03	89.04	94.64
J+B+M	**93.52**	**97.20**	**89.81**	**91.82**	**97.88**

Table 4. Performance comparison of different fusion strategy combinations.

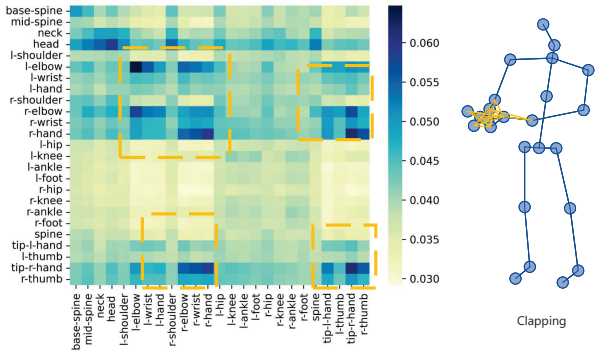

Fig.7. Visualize the adjacency matrix of the learned "clapping" action.

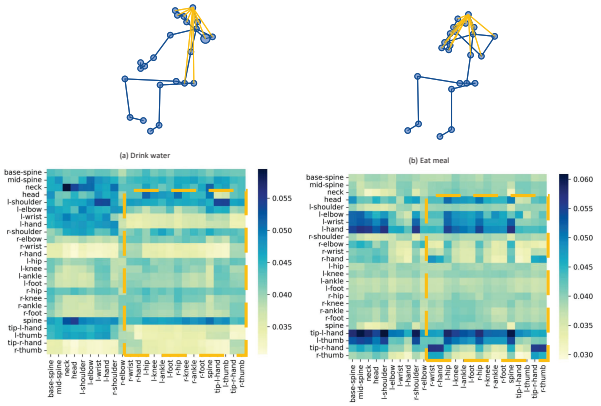

Fig.8. Visualization of adjacency matrix of similar actions. The left image is the "drink water" action, and the right image is the "eat meal" action, in which the yellow box is the gap between the two.

4.5.2 Visualization of confusion matrix

In this section, we verify the effectiveness of the module in effectively aggregating global and local action information by visualizing the confusion matrix. It can be seen from the Fig.9 that the four groups of actions of "drink water", "eat meal", "put on glasses", and "put on hat/cap" all involve the connection between the hand and the head. The boundaries of these two types are not clear, which increases the difficulty of classification. For instance, the action "drink water" is characterized by the left hand lifting a cup to the mouth, while "eat meal" involves both hands holding utensils to the mouth. Despite the commonality of hand and mouth interaction, the objectives and subtleties of execution differ significantly. Although both involve hand and

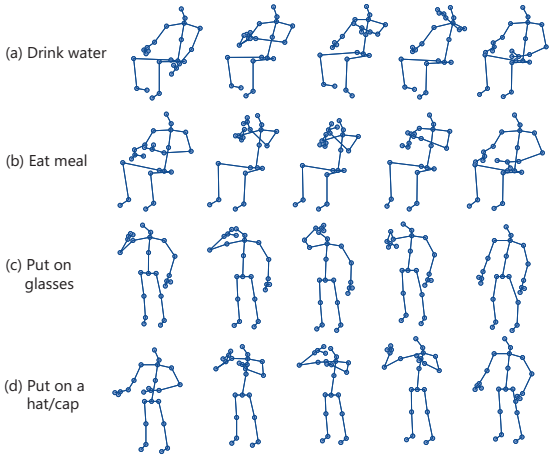

Fig.9. Visualization of skeleton actions for "drink water", "eat meal", "put on glasses", and "put on a hat/cap".

head interaction, their purpose and execution details are different. Recent studies frequently neglect the importance of local information extraction in the aggregation of global information. Our proposed MLKAGC and MSGC modules adeptly integrate global and local cues to more accurately discern features associated with specific actions.

As shown in Fig.10, we select ten similar actions containing the above four groups of actions to visualize the confusion matrix on the NTU-60, NTU-120 and NW-UCLA datasets. Among them, Fig.10(a-d) is the confusion result under three data sets. Under the NTU-60 and NTU-120 datasets, the accuracy rate is above 80%. Benefit from our adaptive adjacency matrix module, the adjacency matrix of similar actions is effectively distinguished, so that the overall accuracy rate is effectively improved. Under the NW-UCLA dataset, it can be clearly seen that our accuracy is above 90%. This shows that our model has also achieved good classification results on small data sets.

Among them, Fig.11(a) is the confusion result without using MLKAGC and MSGC modules, and Fig.11 (b) is the confusion result after using MLKAGC and MSGC modules. As shown in Fig.11(a), it is clear that four groups are very vague. In the confusion matrix Fig.11(b) after module aggregation, the accuracy of "drink water", "eat meal", "put on glasses", and "put on a hat/cap" increased by 10%, 10%, 7%, and 17% respectively, which effectively improved the accuracy of distinguishing similar actions.

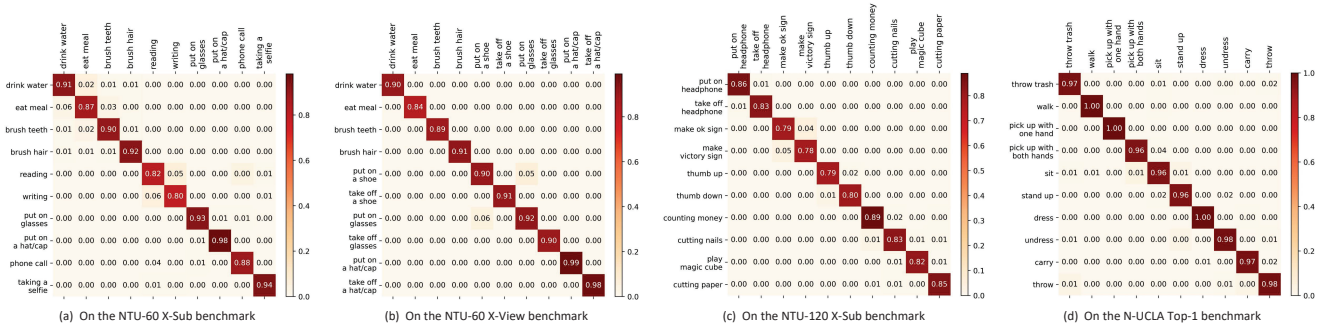

Fig.10. Confusion matrices for selected actions on the NTU-60 and NW-UCLA dataset. We selected 10 similar actions. Among them, (a-d) are the confusion matrices on X-Set and X-View benchmark of NTU-60, X-Sub benchmark of NTU-120, Top1% accuracy of NW-UCLA. The vertical axis label is an accurate label, and the horizontal axis label is a prediction label.

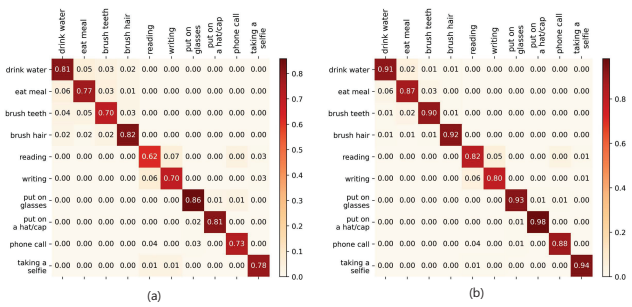

Fig.11. Confusion matrices for selected actions on the NTU-60 X-sub dataset. We selected 10 similar actions. (a) is the confusion matrix before adding MLKAGC and MSGC, (b) is the confusion matrix after adding MLKAGC and MSGC. The vertical axis label is an accurate label, and the horizontal axis label is a prediction label.

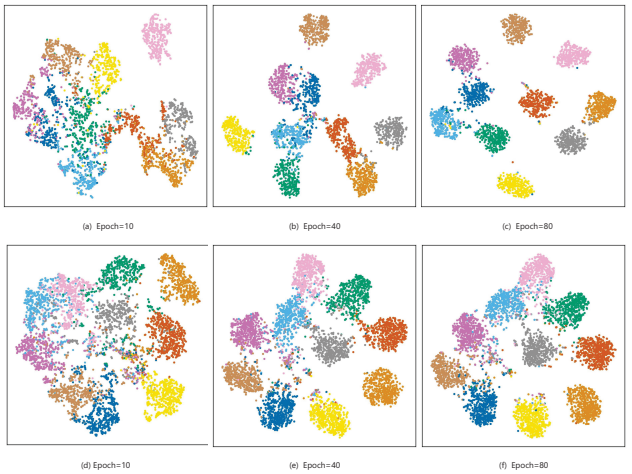

Fig.12. The t-SNE visualization of feature distribution in different periods. (a) - (c): Subgraphs obtained under the X-Sub benchmark of NTU-60. (d)-(f): The subgraph obtained under the X-View benchmark of NTU-120. Different colors represent different action classifications.

4.5.3 TSNE visualization

In order to better demonstrate the superiority of the model in suppressing the GCN over-smoothing problem. We visualize the data distribution under the X-Sub benchmark of the NTU-60 dataset in Fig.12. We select 10 classes in the data set. Through the distribution of feature points on the two-dimensional plane after t-SNE dimensionality reduction, we can clearly see that different types of actions have been effectively separated, which indicates that our MSLK-GCN model effectively alleviates the over-smoothing problem and has high discrimination and expression ability in feature extraction. We can observe that at the beginning of training, the data tends to be confused, but as our model training progresses, the learned features have gradually become clearly identifiable.

In particular, compared with NTU-60, NTU-120 contains 60 additional classes, we select 10 classes for testing and obtained similar experimental results, as shown in Fig.12.

5. Conclusion

In this paper, we propose a novel multi-scale adaptive large kernel graph convolutional network MSLK-GCN, which effectively aggregates local and global spatio-temporal correlations while maintaining the computational efficiency of the model. In this work, we design a multi-scale large-core attention network MLKAGC, which integrates multi-scale large-core attention and gating mechanism to adaptively adjust the attention map while capturing the long-distance dependencies in the skeleton data. At the same time, a multi-scale adaptive network MSGC is designed to adaptively learn the relationship between different joints by dynamically adjusting the connection weights between nodes in graph convolution. Finally, our module combines MLKAGC and MSGC to generate global and local features for effective feature fusion to obtain a new feature. The new features are input into the multi-scale time self-attention convolution module MSTC, which is composed of ECA attention mechanism and multi-scale time module, for time feature aggregation. We use a multi-stream fusion strategy to perform a series of processing on human joint points to achieve better recognition results. Our model outperforms the state-of-the-art models on four benchmark datasets from three interactive datasets: NTU-RGB + D 60, NTU-RGB + D 120, and Northwestern-UCLA. Both mathematical analysis and experimental re-

sults show that MSLK-GCN has stronger representation ability than other graph convolutions.

We find that the model also has good generalization ability for two-person skeleton behavior recognition. Therefore, this provides a new research idea for us to study the complex logical relationship in two-person behavior.

References

[1] A. Baisware, B. Sayankar, and S. Hood. Review on recent advances in human action recognition in video data. In *2019 9th International Conference on Emerging Trends in Engineering and Technology-Signal and Information Processing (ICETET-SIP-19)*, pages 1–5. IEEE, 2019. 1

[2] 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*, pages 6299–6308, 2017. 1

[3] S. Chen, K. Xu, X. Jiang, and T. Sun. Pyramid spatial-temporal graph transformer for skeleton-based action recognition. *Applied Sciences*, 12(18):9229, 2022. 2

[4] 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*, pages 13359–13368, 2021. 1, 3, 8, 9

[5] 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*, pages 183–192, 2020. 2, 8, 9

[6] M. Dai, Z. Sun, T. Wang, J. Feng, and K. Jia. Global spatio-temporal synergistic topology learning for skeleton-based action recognition. *Pattern Recognition*, 140:109540, 2023. 8, 9

[7] D. Gerónimo and H. Kjellström. Unsupervised surveillance video retrieval based on human action and appearance. In *2014 22nd international conference on pattern recognition*, pages 4630–4635. IEEE, 2014. 1

[8] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu. Visual attention network. *Computational Visual Media*, 9(4):733–752, 2023. 2, 3

[9] J. Kong, S. Wang, M. Jiang, and T. Liu. Multi-stream ternary enhanced graph convolutional network for skeleton-based action recognition. *Neural Computing and Applications*, 35(25):18487–18504, 2023. 8

[10] K. W. Lau, L.-M. Po, and Y. A. U. Rehman. Large separable kernel attention: Rethinking the large kernel attention design in cnn. *Expert Systems with Applications*, 236:121352, 2024. 3

[11] J. Lee, M. Lee, D. Lee, and S. Lee. Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10444–10453, 2023. 1, 3, 9

[12] 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*, pages 3595–3603, 2019. 1, 3, 8

[13] H. Liu, Y. Liu, Y. Chen, C. Yuan, B. Li, and W. Hu. Transkeleton: Hierarchical spatial–temporal transformer for skeleton-based action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(8):4137–4148, 2023. 2, 8

[14] 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*, 42(10):2684–2701, 2019. 7

[15] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):3007–3021, 2017. 1, 8

[16] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot. Global context-aware attention lstm networks for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1647–1656, 2017. 1, 8

[17] M. Liu, H. Liu, and C. Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017. 1, 8, 9

[18] Y. Liu, H. Zhang, Y. Li, K. He, and D. Xu. Skeleton-based human action recognition via large-kernel attention graph convolutional network. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2575–2585, 2023. 1, 3, 8

[19] 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*, pages 143–152, 2020. 3, 8

[20] C. Plizzari, M. Cannici, and M. Matteucci. Skeleton-based action recognition via spatial and temporal transformer networks. *Computer Vision and Image Understanding*, 208:103219, 2021. 1, 3, 8

[21] H. Qiu, B. Hou, B. Ren, and X. Zhang. Spatio-temporal tuples transformer for skeleton-based action recognition. arxiv 2022. *arXiv preprint arXiv:2201.02849*. 1, 8

[22] 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*, pages 1010–1019, 2016. 7

[23] 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*, pages 7912–7921, 2019. 3, 8

[24] 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*, pages 12026–12035, 2019. 1, 3, 8

[25] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proceedings of the IEEE/CVF*

conference on computer vision and pattern recognition, pages 1227–1236, 2019. 8, 9

[26] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014. 1

[27] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang. Constructing stronger and faster baselines for skeleton-based action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 45(2):1474–1488, 2022. 3, 8

[28] T. T. M. Tran, C. Parker, and M. Tomitsch. A review of virtual reality studies on autonomous vehicle–pedestrian interaction. *IEEE Transactions on Human-Machine Systems*, 51(6):641–652, 2021. 1

[29] 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*, pages 588–595, 2014. 2

[30] 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*, pages 4471–4479, 2016. 2

[31] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2649–2656, 2014. 7

[32] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11534–11542, 2020. 6

[33] Y. Wang, Y. Li, G. Wang, and X. Liu. Multi-scale attention network for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5950–5960, 2024. 3

[34] Y.-H. Wen, L. Gao, H. Fu, F.-L. Zhang, S. Xia, and Y.-J. Liu. Motif-gcns with local and non-local temporal blocks for skeleton-based action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2009–2023, 2022. 3, 8

[35] C. Wu, X.-J. Wu, T. Xu, Z. Shen, and J. Kittler. Motion complement and temporal multifocusing for skeleton-based action recognition. *IEEE transactions on circuits and systems for video technology*, 34(1):34–45, 2023. 8, 9

[36] Y. Xia, Q. Gao, W. Wu, and Y. Cao. Skeleton-based action recognition based on multidimensional adaptive dynamic temporal graph convolutional network. *Engineering Applications of Artificial Intelligence*, 127:107210, 2024. 2

[37] K. Xu, F. Ye, Q. Zhong, and D. Xie. Topology-aware convolutional neural network for efficient skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2866–2874, 2022. 1, 8, 9

[38] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 1, 3, 8, 9

[39] 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*, pages 1112–1121, 2020. 8, 9

[40] Z. Zhao, Z. Chen, J. Li, X. Xie, K. Chen, X. Wang, and G. Shi. Stdm-transformer: Space-time dual multi-scale transformer network for skeleton-based action recognition. *Neurocomputing*, 563:126903, 2024. 8

[41] Y. Zhou, X. Yan, Z.-Q. Cheng, Y. Yan, Q. Dai, and X.-S. Hua. Blockgcn: Redefine topology awareness for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2049–2058, 2024. 1