An Improved Graph Pooling Network for Skeleton-Based Action Recognition

Cong Wu
Jiangnan University
Wuxi, China
congwu@stu.jiangnan.edu.cn

Xiao-Jun Wu
Jiangnan University
Wuxi, China
wu_xiaojun@jiangnan.edu.cn

Tianyang Xu
Jiangnan University
Wuxi, China
tianyang_xu@163.com

Josef Kittler
University of Surrey
Guildford, United Kingdom
j.kittler@surrey.ac.uk

ABSTRACT

Pooling is a crucial operation in computer vision, yet the unique structure of skeletons hinders the application of existing pooling strategies to skeleton graph modelling. In this paper, we propose an Improved Graph Pooling Network, referred to as IGPN. The main innovations include: Our method incorporates a region-awareness pooling strategy based on structural partitioning. The correlation matrix of the original feature is used to adaptively adjust the weight of information in different regions of the newly generated features, resulting in more flexible and effective processing. To prevent the irreversible loss of discriminative information, we propose a cross fusion module and an information supplement module to provide block-level and input-level information respectively. As a plug-and-play structure, the proposed operation can be seamlessly combined with existing GCN-based models. We conducted extensive evaluations on several challenging benchmarks, and the experimental results indicate the effectiveness of our proposed solutions. For example, in the cross-subject evaluation of the NTU-RGB+D 60 dataset, IGPN achieves a significant improvement in accuracy compared to the baseline while reducing Flops by nearly 70%; a heavier version has also been introduced to further boost accuracy.

CCS CONCEPTS

• **Computing methodologies** → **Activity recognition and understanding**; • **Networks** → *Layering*.

KEYWORDS

Graph Pooling, Skeleton-based Action Recognition

1 INTRODUCTION

Recent studies have verified the inherent advantages of Graph Convolutional Networks (GCNs) in the representation of non-Euclidean structured data [59], with GCN-based methods achieving leading performance in various fields [19, 24, 36]. Yan *et al.* [61] first proposed to construct spatial-temporal graph to model skeleton sequences according to their natural arrangement. Refer to this paradigm, subsequent methods [9, 10, 29, 39, 49, 50, 65] have made significant improvements in performance. However, only global pooling is used before the final classifier in most methods, leading to the redundancy of homogeneous information in graph modelling [44]. Simultaneously, pooling is a conventional feature aggregation technique in Convolutional Neural Networks (CNNs), which typically integrates local information obtained after convolutional layers to produce more abstract feature representations. This operation reduces redundant information, lowers the computational cost, prevents over-fitting, and enhances the model's generalisation ability, while also increasing the integration of features and obtaining higher-level abstract representations through aggregation. The pooling operation is not only justified theoretically [4, 5], but also has been shown to significantly improve performance in many computer vision tasks [23, 47].

Skeleton graph pooling strategy has recently gained significant attention in research. [1] introduced novel pooling and unpooling operators within different homeomorphic skeletons. [8] divided the skeleton into regions based on physical arrangement and performed continuous pooling operations in each local region. Those approaches bring higher computational efficiency while achieving higher recognition accuracy. However, both methods rely on fixed manual pooling rules, which may remove useful information and limit their flexibility. [43] introduced triplet pooling loss to facilitate adaptive clustering of skeleton graph nodes and automatically generate new embedded matrix representations. This approach introduces great flexibility, making the pooling process a trainable method. However, it does not explicitly consider the structural-semantic information. In addition, existing pooling methods tend to force the network to discard essential but not the most critical nodes gradually. However, the accumulation of multiple pooling operations can significantly affect the final performance. Hence, it is necessary to preserve and strengthen the discriminative and heterogeneous features while discarding unnecessary homogeneous features.

To address the limitations of existing pooling networks that rely on either fixed manual rules or excessive flexibility while ignoring the inherent structure, we propose a trainable structure pooling operation that strikes a balance between flexibility and structural preservability. Specifically, we define the pooling area according to the physical structure of the skeleton, then calculate the contribution between each feature point in the pooling area and the entire graph representation according to the embedding similarity. After that, we adjust the proportion of the current feature point in the pooled features by constructing a trainable assign matrix.

Table 1: Performance comparison with different backbones on cross-subject of NTU-RGB+D 60. The inference speed and GPU memory is measured on a single GeForce RTX 2080 Ti.

Baseline	Pooling	Flops (G)	Infer Speed (Videos per Second)	GPU Mem (MiB)	Acc (%)
AGCN [49]	-	4.17	589	6988	88.9
	IGP-Light	1.16	915	4031	90.1
	IGP-Heavy	5.02	515	7195	90.5
Graph2Net [57]	-	1.16	824	9273	89.1
	IGP-Light	0.44	1268	4689	90.2
	IGP-Heavy	1.28	687	8463	91.2
CTR-GCN [9]	-	1.98	75	9613	89.9
	IGP-Light	0.70	115	4573	90.9
	IGP-Heavy	2.04	61	8229	91.1

With the accumulation of multi-layer pooling operations, the loss of information will only intensify without remediation, the network capacity also will be surplus. Here we propose block-based and network-based feature enhancement strategies, namely cross fusion block and information supplement module. In cross fusion block, based on the original pooling network, we design another parallel operation that directly performs graph operations and then fuses the features obtained from the two parallel structures after alignment. In this way, feature fusion of different granularities is performed to supplement and strengthen the information of features. In information supplement module, we decompose the skeleton sequence into position and vector-based features, then project them into common space through graph embedding for sufficient alignment, and combine them to obtain an efficient representation through feature fusion.

The conducted ablation experiments demonstrate our motivation and the effectiveness of the proposed method, and the results combined with various mainstream methods consistently show that our method can significantly reduce computational overhead while improving model performance. As shown in Table 1, the proposed improved graph pooling operation can can greatly reduce training overhead while ensuring improvement in accuracy performance.

Overall, our innovations can be summarised as follows:

- Aiming at the structural properties of the skeleton sequences, we propose a structure pooling strategy with region awareness, which increases the pooling flexibility while maintaining the skeleton structural characteristics.
- We use the cross fusion module and information supplement module to provide block-level and input-level information respectively, thus avoiding the loss of a large amount of information and the under-saturation of the network.
- We conduct comprehensive experiments on NTU-RGB+D (60&120) [34, 48], UWA3D Multiview Activity II [46] datasets respectively, the experimental findings provide full validation of our strategies.

2 RELATED WORK

2.1 Skeleton-based Action Recognition

Non-GCN Methods. Due to the particularity of the structure of skeleton features, existing strategies often focus on their essential physical characteristics to carry out targeted modelling. Traditional methods for skeleton sequence processing and recognition generally involve deforming the skeleton sequence by constructing suitable descriptors. [7] used the star skeleton as a representative descriptor of the human skeleton, then converted the skeleton sequence into a symbol sequence that can be processed by the Hidden Markov Model. [53] proposed to represent skeleton sequences as curves in the Lie group, and the classification was performed by combining classifiers. [20] proposed using Hierarchical Gaussian Process Latent Variable Model to learn a hierarchical manifold space of motion patterns. With the development of research in recent years, the effectiveness of deep learning frameworks has been widely verified in various vision tasks [21, 22, 40], which inspired people to leverage the discriminative feature extraction capabilities of those deep networks for skeleton modelling. Du *et al.* [16] proposed a method that converts the skeleton sequence into an image format, which can be processed using CNNs for feature extraction. [38] further used colourisation to implicitly encoding the spatiotemporal information of the skeleton sequences. [25] generated multiple sets of clip-based representations and processed this information in parallel with the help of the Multi-Task Learning Network. Inspired by the natural advantages of Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) in sequence modelling, methods such as [2, 17, 30, 51] used RNNs or LSTMs to capture long-term contextual information.

GCN-based Methods. As a typical message-passing network, the emergence of GCNs provides a new paradigm for skeleton sequence modelling. Yan *et al.* [61] extended the basic GCNs operations to the spatiotemporal dimension. Shi *et al.* [49] moved away from the fixed graph modelling paradigm and proposed an adaptive relational representation. [39] proposed a complex disentangling and unifying operation based on the previous works, which significantly improved the model's classification performance. [9] constructed a more comprehensive graph learning framework by strengthening the representation granularity of channels. Although the skeleton sequence contains relatively compact information and requires low computational overhead, researchers have recently conducted extensive research on recognition efficiency to avoid blindly stacking models. Combined with high-level semantic information, Zhang *et al.* [66] proposed a simple yet effective semantics-guided neural network. [12] used efficient shift operations to implement the interaction of spatiotemporal information, significantly reducing the number of trainable parameters and computational complexity. Taking into account both accuracy and efficiency, Wu *et al.* [57] divided

the channel dimensions and endowed them with diverse modelling capabilities, thereby improving the utilisation of network parameters. Qin *et al.* [45] proposed a novel approach for constructing high-dimensional feature representations using angular encoding. The study by Liu *et al.* [33] proposed a successful attack on existing skeletal graph models by exploiting spatiotemporal constraints.

2.2 Graph Pooling Operation

In contrast to digital images, skeletons are typical non-Euclidean structural data with feature points that are not rigidly arranged and exhibit strong physical structure characteristics. As a result, pooling methods designed for image tasks, which typically involve selecting a regular rectangular area and obtaining a new feature point by averaging or taking the maximum value of all points in that area, are not directly applicable. Instead, skeleton graph pooling often needs to consider both the graph's structure and the arrangement of its nodes. Specifically, it includes how to select the pooling area and how to generate a new feature representation. There are three main approaches for graph pooling: hard rule, graph coarsening, and node selection. Hard rule involves pre-setting pooling nodes and rules based on the known graph structure. Chen *et al.* [8] proposed a structure-based graph pooling scheme that considers the unique structure and prior topological knowledge of skeleton sequences. Graph coarsening is a trainable version of the hard rule that involves clustering nodes and synthesising a supernode. A differentiable graph pooling module constructed by [63] to map the nodes of each layer to a set of clusters. [3] introduced a graph-based clustering method for graph pooling using continuous relaxation of the normalised minimum segmentation problem and achieved cluster assignment by training Graph Neural Networks (GNNs). [43] proposed a hierarchical graph representation method using triplet pooling loss. Node selection is to select some critical nodes to construct a new graph representation, thereby replacing the original structure. [18] proposed graph pooling and unpooling operations by selecting important nodes and restoring global nodes. [28] used the attention mechanism in GCNs to generate graph topology masks and used this mask to complete the pooling operation. In most previous studies, people often directly apply the graph pooling strategy to the skeleton task, failing to consider the structural properties of skeleton features and the specificity of different features.

3 THE PROPOSED APPROACH

In this section, we will begin by constructing the basic graph modelling module. Subsequently, the proposed innovations will be introduced step by step. Finally, we will describe the overall architecture that integrates all the components.

3.1 Basic Graph Modelling Module

Referring to the standard graph definition, for any given graph $G = (V, E)$, V represents the set of nodes, E represents the set of corresponding edges. During the modelling process, each node aggregates information from its neighbouring nodes and updates its features accordingly. Given the graph G_l at l_th layer, according to the defined rules, the newly generated graph can be expressed as,

$$G_{l+1} = \mathrm{F}(G_l, W_l) = \mathrm{F}^{\mathrm{U}}((\mathrm{F}^{\mathrm{A}}(G_l, W_l^A)), W_l^U), \tag{1}$$

F represents the graph operation, W denotes the weight. The graph operation can be divided into two main components, namely aggregation F^{A} and updating F^{U}, where W^A and W^U represent the corresponding weights.

As described in [61], spatial graph convolution involves performing a standard 2D convolution with a 1×1 kernel, followed by multiplying the resulting feature with a normalised adjacency matrix. Meanwhile, temporal graph convolution can be represented as a well-defined convolution operation on the constructed temporal graphs. In this work, we adopt existing methods as the backbone network to fairly measure the effectiveness of the proposed method.

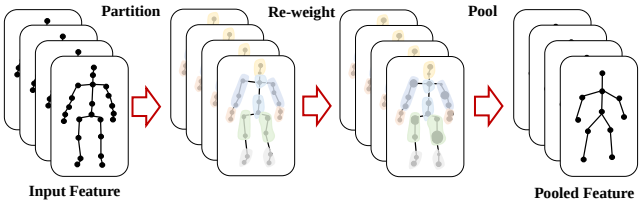


Figure 1: Structural Spatial Pooling. We divide the skeleton according to its spatial structure and assign different attentions to different nodes to achieve adaptive structured spatial pooling. The size of the dot indicates the corresponding intensity of attention.

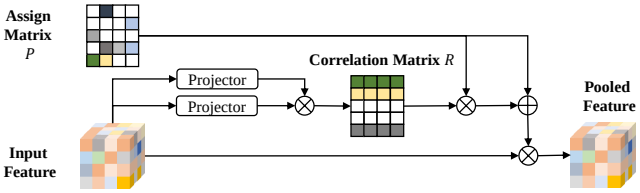


Figure 2: Structure pooling strategy with region awareness. We combine the characteristics of the current feature itself, design an adaptive feature pooling operation, and automatically calculate the relationship matrix of the current feature.

3.2 Structure Pooling Strategy with Region Awareness

In this section, we propose a novel structure pooling strategy (SSP) that incorporates region awareness to achieve more flexibility while preserving the underlying structure of the skeleton graph. The graph convolution operation applied to skeleton-based action recognition typically involves both the feature itself and its corresponding relationship representation (usually represented by its adjacency relationship). Therefore, to perform pooling on the skeleton sequence, it is essential to not only select the pooling area but also take into account the transformation between the old and new graphs. Additionally, it is necessary to generate a new relationship descriptor for the features that have undergone the pooling operation.

We begin by denoting the current features at t_th frame as $x_t = [x_{t0}, x_{t1}, ..., x_{ti}, ..., x_{tN}]$, where $x_{ti} \in \mathbf{R}^C$ corresponds to the feature

of node v_i at frame t. We then divide the skeleton graph, which consists of N nodes, into M regions based on its physical structure, and then perform feature aggregation separately on each region to obtain a new graph with M nodes. We generate the fundamental transformation matrix $P \in \mathbb{R}^{N\times M}$ between the old and new features. If joint i belongs to the $j_$th partitioned area (i.e., corresponds to the feature point j in the new graph), then $P_{ij} = 1$; otherwise, $P_{ij} = 0$. While we believe that the above assignment setting is conducive in preserving structural characteristics, it ignores the specificity responses of each point across the entire graph or even across different input sequences. For example, 'Rub Two Hands' is more focused on hand movement, while 'Jump Up' requires coordination of the entire body.

To tackle these challenges, we have imbued the pooling operation with the requisite flexibility without compromising its structural integrity. The entire calculation process is illustrated in Figure 1 and Figure 2. Initially, we compute the correlation matrix R, which captures the correlation between each feature point and other locations. The value associated with each feature point can be represented as the average of its connections with other feature points in the current graph, that is,

$$\forall i \in N, R_{ti} = \sigma(\sum_{j=1}^{N} \frac{\phi(h(x_{ti}))^T \psi(h(x_{tj}))}{N}), \tag{2}$$

where $\sigma(\cdot)$ is the normalisation operation, and ϕ and ψ are projection functions that can be implemented by convolution filters, the dimension of project space is set as C/r. The entry R_{ti} measures the mean of the connection between point v_i and all other nodes at $t_$th frame, and this can be computed based on their corresponding latent representations $h(\cdot)$. After that, the new feature after the spatial pooling operation can be expressed as,

$$\text{SPooling}(x_t) = x_t R_t P, \tag{3}$$

Residual connection is also introduced to achieve training stability. That is, for each newly generated x^{sp}_{tj},

$$x^{sp}_{tj} = \sum_{i=1}^{N} x_{ti}(P_{ij} + R_{ti}P_{ij}). \tag{4}$$

Combing with (2), we have

$$x^{sp}_{tj} = \sum_{i=1}^{N} x_{ti}(P_{ij} + \sigma(\sum_{j=1}^{N} \frac{\phi(h(x_{ti}))^T \psi(h(x_{tj}))}{N})P_{ij}). \tag{5}$$

By introducing a trainable allocation matrix, our proposed strategies provide greater flexibility in treating each feature differently during the pooling operation. Furthermore, we retain the basic transformation form, which ensures the preservation of the structural characteristics of the skeleton sequence. Finally, to obtain the relation representation for the newly pooled feature, we need to compute a new adjacency matrix that reflects the physical structure of the new graph. Since we maintain the basic structure in this pooling process, that is, the pooling is performed in a specific area, the feature representation and its physical structure associations can be conveniently obtained based on the pooling strategy. This enables the generation of the corresponding relationship description for each newly generated feature.

As for temporal pooling, due to the regular arrangement of the skeleton on the temporal dimension as discussed in Section 3.1, we can directly apply conventional pooling strategies. The final pooled feature is obtained by performing an average pooling with a kernel of 2 on the current feature, as shown in Figure 2, so the final formal can be written as,

$$\begin{aligned} x^p &= \text{STPooling}(x) = \text{TPooling}(\text{SPooling}(x)) \\ &= \text{TAP}(x^{sp}) = (x^{sp}_{[0::2]} + x^{sp}_{[1::2]})/2. \end{aligned} \tag{6}$$

TAP means the Average Pooling on the temporal dimension with a kernel size of 2, $[0 :: 2]$ and $[1 :: 2]$ represent to extract 1 frame every 2 frames starting from frame 0 and 1 respectively.

3.3 Cross Fusion Block

The pooling operation significantly enhances the gathering and transmission of features and has the potential to reduce redundancy. Bu tit may also have the opposite effect for skeletons due to the irreparable and significant loss of effective information. To alleviate the information loss during the pooling process, we maintain the basic pooling structure and construct a cross fusion block in a plug-and-play form. As depicted in Figure 3, the lower branch of the Cross Fusion Block represents the original graph pooling network module, while the newly introduced upper branch models obtain finer skeleton features that are cross-fused with the lower branch to enhance the representation ability of the original features.

As a fundamental component, we construct a graph pooling network module following the definitions in Section 3.1 and Section 3.2; Specifically, we first obtain a new graph using the predefined pooling strategy and then perform a graph convolution operation on the newly generated feature representation. Specially, for the current input feature of this block, based on the feature x_p obtained by pooling in Equation 6, the new representation can be expressed as,

$$h = \text{GraphConv}(x^p), \tag{7}$$

where GraphConv general represents the graph convolution operation. However, with this process, multiple nodes in the pooling area are replaced by a single new node representation before the graph operation. As a consequence, the subsequent graph convolution is executed on the compressed skeleton features with much lower dimension, resulting in the modelling of more abstract features. However, finer parts may not be fully captured in this process. Therefore, we need to emphasise the modelling of more refined features for our pooling network. We first use graph convolution to further model the current features x to obtain a more informative feature representation $\hat{x}$,

$$\hat{x} = \text{GraphConv}(x). \tag{8}$$

Then, the fusion of $\hat{x}$ with h is performed to improve the overall representation ability. However, as $\hat{x}$ and h have different resolutions, there is a misalignment issue between them. To address this, we align the dimensions of these two features by performing a pooling operation on $\hat{x}$ using h as a reference. Finally, the two aligned features are fused, as illustrated in Figure 3. According to the previous definition, we denote the upper branch feature after alignment as e,

$$e = \text{STPooling}(\hat{x}) = \text{TPooling}(\text{SPooling}(\hat{x})) \tag{9}$$

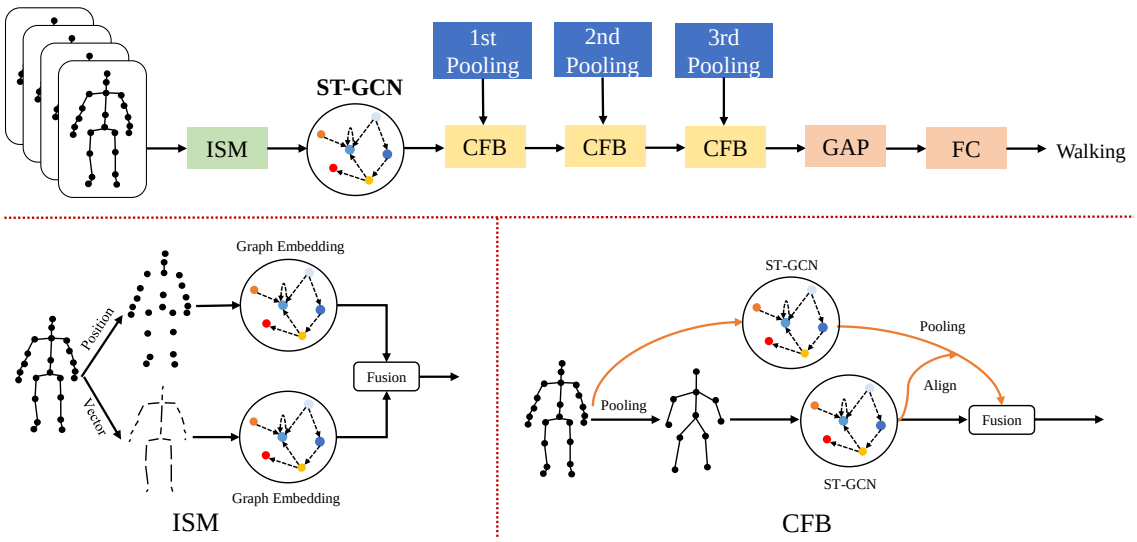


Figure 3: The Overall Structure. ISM stands for Information Supplement Module, CFB stands for Cross Fusion Block, and GCN, GAP, and FC correspond to graph convolution operations, global average pooling, and fully connected layers, respectively.

this formula represents the alignment operation, where $R(\hat{x})$ represents the relational representation calculated based on the current feature $\hat{x}$, and the calculation process can be referred to Equation 2. After that, the specific operations in the entire cross fusion block can be expressed as,

$$y = s \cdot h + (1 - s) \cdot e \tag{10}$$

where s means the corresponding weight, we will discuss it later in experimental section.

3.4 Information Supplement Module

Given that the introduction of the pooling network significantly eliminates most information, it is reasonable to believe that the current network structure has not fully utilised its performance potential. Rather than using conventional techniques such as distillation or pruning to compress the current network size, we opt to enhance the input features to fully exploit the modelling capabilities of the existing network structure. Previous related methods often adopted complex multi-stream structures [12, 39, 49] to exploit the multi-dimensional representation information of input features. However, this strategy usually results in limited performance improvement while significantly increasing the computational and parameter requirements, thereby limiting the advantages of using skeleton features in real-world application deployment.

Compared to traditional methods, our approach involves performing appropriate pre-processing on the input side of the network to construct efficient input features while maintaining the feature extraction network, as illustrated in Figure 3. We focus on two critical feature descriptors for the skeleton sequence: position-based and vector-based features, corresponding to joints and bones in the skeleton. To align and integrate the position-based and vector-based features of the skeleton sequence, we use graph convolution to project them into a high-dimensional vector space. The projection operation maps each feature into a common vector space, enabling direct alignment and fusion. This results in a more informative representation that can capture both the joint and bone information. These processes can be expressed as follows,

$$f_p(x) : x \underset{position}{\rightarrow} x^{pos},$$
$$f_v(x) : x \underset{vector}{\rightarrow} x^{vec}. \tag{11}$$

x^{pos} and x^{vec} correspond to the position-based and vector-based features obtained from the initial skeleton sequence, respectively, as mentioned in [49].

$$x_{in}^{vec} = \text{GrpahConv}(\text{Norm}(x^{vec})),$$
$$x_{in}^{pos} = \text{GrpahConv}(\text{Norm}(x^{pos})). \tag{12}$$

The normalisation operation can be expressed as,

$$\text{Norm}(x) = \frac{x - E(x)}{\sqrt{var(x) + \epsilon}} * \gamma + \beta, \tag{13}$$

where γ and β are trainable parameters, E and var are the mean and standard-deviation. The final input x_{in} is the fusion of position-based feature and vector-based feature.

$$x_{in} = \text{Concat}(x_{in}^{vec}, x_{in}^{pos}), \tag{14}$$

where Concat means feature concatenation along the channel dimension. The proposed input-strengthening strategy brings only a limited increase in the number of parameters and computational overhead compared to the original network structure but greatly enhances the representation of the new input features. The effectiveness of this strategy is verified through follow-up experiments.

3.5 The Overall Structure

Combining the above descriptions, we construct the improved pooling network model starts with a baseline backbone taken from [8], and replaces the basic graph convolution operation with the module proposed in [57]. We also make some optimisations to the data processing part following [9]. Our innovations are then added to the

baseline, resulting in the improved pooling network model whose final structure is shown in Figure 3. The improved pooling network model begins by processing the input feature using an information supplement module to obtain a new representation with richer information. The output of this module is then passed through several cross fusion blocks, where each block combines a graph pooling and a graph convolution operation. Finally, the classification results are obtained after global pooling and a classification layer. We also remove the cross fusion block to construct a lightweight version, named **IGPN-Light**. For distinction, the default model is denoted as **IGPN-Heavy**.

4 EXPERIMENTS

4.1 Datasets and Implementation Details

NTU-RGB+D. The NTU-RGB+D 60 dataset [48] comprises 56,880 videos captured from 80 camera views and 40 subjects, with 60 action categories. The NTU-RGB+D 120 dataset [34] extends the previous version to 120 action categories, with 114,480 videos captured from 155 views and 106 subjects. To assess the effectiveness of our approach, we utilise the commonly adopted evaluation methods: cross-subject and cross-view for NTU-RGB+D 60, cross-subject and cross-set for NTU-RGB+D 120. For cross-subject evaluation, we use videos performed by half of the subjects for training and the remaining half for testing in both datasets. For cross-view evaluation, we use the samples collected by cameras with ID 2 and 3 for training and the remaining for testing in NTU-RGB+D 60. For cross-set evaluation, we use samples with even camera IDs for training and the others for testing in NTU-RGB+D 120.

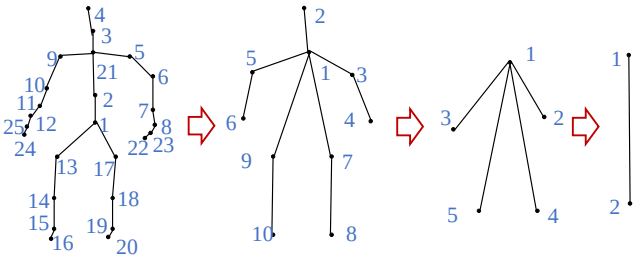
UWA3D Multiview Activity II. The UWA3D Multiview Activity II Dataset [46] comprises 30 different classes performed by 10 characters, with each video captured by four different views (front, left, right, and top). We randomly select two views as training data and test on the remaining two views separately. In our experiments, we report experimental results for all combinations of views, as well as the final average performance.

Table 2: The implementation details of the proposed method.

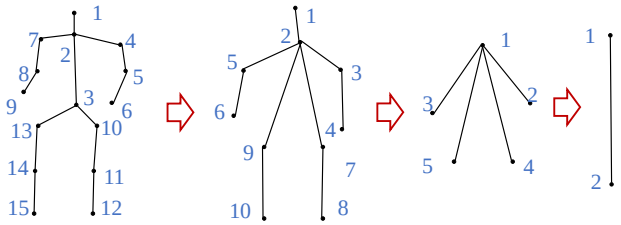
Dataset	NTU-RGB+D	UWA3D
Optimizer	SGD, Momentum=0.9, Nesterov=True	
Epoch	65	100
Warm up epoch	5	
Learning rate	1e-1	
Learning rate policy	Linear decay with 0.1	
Decay step	[35,55]	[50,80]
Weight decay	0.0004	0.01
Batchsize	64	32
Frame	64	60
Data augmentation	Random rotate	

4.2 Implementation Details

Training and evaluation. In general, our experiments are performed on the PyTorch platform, with cross-entropy loss and Stochastic Gradient Descent with Nesterov Momentum (0.9) employed as the training loss and optimisation algorithm, respectively. We use the processing steps from [66] for the NTU-RGB+D datasets. The initial learning rate is set to 0.1, with a batch size of 64 for NTU-RGB+D and 32 for UWA3D Multiview Activity II dataset. For NTU-RGB+D datasets, the learning rate is reduced to 1/10 of the previous epoch at the 35th and 55th epoch, and training stopped at the 65th epoch. For UWA3D Multiview Activity II dataset, the learning rate decay at 50th and 80th epochs, and the training stopped at the 100th epoch. We apply a warm-up strategy for the first five epochs across all datasets. Please find the details from Table 2.



(a) For NTU-RGB+D datasets.



(b) For UWA3D Multiview Activity II dataset.

Figure 4: The process of spatial pooling.

Predefined spatial pooling rules. As shown in the Figure 4, we give the spatial skeleton graph in the whole network. The left side of the arrow is the spatial graph before pooling, and the right side of the arrow represents the newly generated graph after the structure spatial pooling operation.

As for the pooling rules, for the given skeleton representations, structured region division is performed first; with all nodes in each region condensed, a new sparse and compact graph representation can be obtained. Refer to [43], the flow of several pooling operations is shown in the Table 3.

4.3 A Comparison with the state-of-the-art

The effectiveness of the proposed method is verified through comprehensive comparisons with state-of-the-art methods on multiple benchmarks, as presented in Figure 5, Table 4 and Table 5.

First, we conducted a performance comparison between IGPN and mainstream methods under single-stream evaluation. As shown in Figure 5, the proposed IGPN outperforms HD-GCN [29], InfoGCN [14], CTR-GCN [9], MS-G3D [39], DC-GCN [11] in accuracy while maintain fewer Flops and parameters. These results illustrate the efficiency and effectiveness of the proposed method, which destined its usability in real applications.

From Table 4, the results show demonstrate the proposed method is competitive with state-of-the-art methods. We observe that, compared with the baseline method Graph2Net, both version of IGPN achieve significant improvement in accuracy on NTU-RGB+D 60

Table 3: Predefined spatial pooling rules.

(a) For NTU-RGB+D datasets.

First Pooling		Second Pooling		Third Pooling	
Old IDs	New ID	Old IDs	New ID	Old IDs	New ID
(1,2,21)	1	(1,2)	1	(1,2,3)	1
(3,4,21)	2	(3,4)	2	(4,5)	2
(5,6,7)	3	(5,6)	3		
(8,22,23)	4	(7,8)	4		
(9,10,11)	5	(9,10)	5		
(12,24,25)	6				
(13,14)	7				
(15,16)	8				
(17,18)	9				
(19,20)	10				

(b) For UWA3D Multiview Activity II dataset.

First Pooling		Second Pooling		Third Pooling	
Old IDs	New ID	Old IDs	New ID	Old IDs	New ID
(1,2)	1	(1,2)	1	(1,2,3)	1
(2,3)	2	(3,4)	2	(4,5)	2
(4,5)	3	(5,6)	3		
(5,6)	4	(7,8)	4		
(7,8)	5	(9,10)	5		
(8,9)	6				
(10,11)	7				
(11,12)	8				
(13,14)	9				
(14,15)	10				

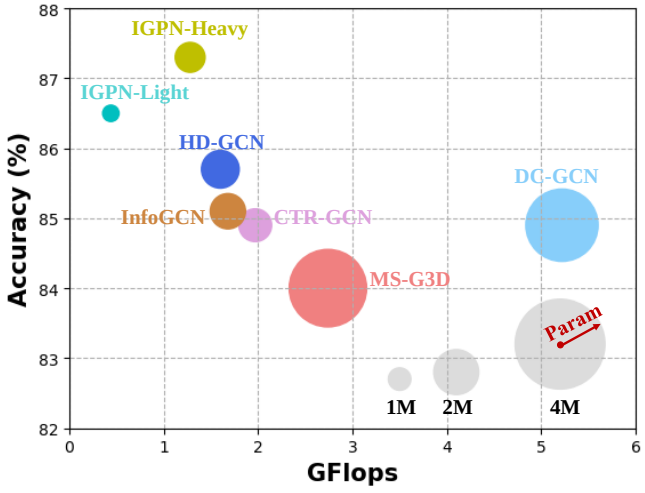


Figure 5: The performance of mainstream algorithms on Cross-Subject of the NTU-RGB+D 60 dataset. (All results are obtained from the joint-stream network.)

and NTU-RGB+D 120 datasets. CTR-GCN is one of the most representative methods published in recent years; compared with this

Table 4: Accuracy (%) comparison of our approach under multi-stream evaluation with the state-of-the-art methods on NTU-RGB+D 60&120.

Methods	NTU-RGB+D 60		NTU-RGB+D 120	
	X-Sub	X-View	X-Sub	X-Set
Lie Group [53]	50.1	52.8	-	-
H-RNN [17]	59.1	64.0	-	-
PA-LSTM [48]	62.9	70.3	25.5	26.3
SkeMotion[35]	-	-	67.7	66.9
STA-LSTM [51]	73.4	81.2	-	-
Visualize CNN [38]	76.0	82.6	-	-
Multi CNN+RotClips [26]	-	-	62.2	61.8
TSRJI [6]	-	-	67.9	62.8
Hybrid [15]	79.4	84.1	-	-
BPLHM [67]	85.4	91.1	-	-
GeomNet [41]	93.6	96.3	86.5	87.6
ST-GCN [61]	81.5	88.3	-	-
AGCN [49]	88.5	95.1	82.9	84.9
SGN [66]	89.0	94.5	79.2	81.5
AGC-LSTM [50]	89.2	95.0	-	-
SGCN [42]	89.4	95.7	-	-
MV-HPGNet [55]	-	-	83.5	85.4
MV-IGNet [55]	-	-	83.9	85.6
Shift-GCN [12]	90.7	96.5	85.9	87.6
MS-G3D [39]	91.5	96.2	86.9	88.4
Dynamic GCN [62]	91.5	96.0	87.3	88.6
GCN-HCRF [37]	90.0	95.5	-	-
CD-JBF-GCN [52]	89.0	95.7	-	-
Sym-GNN [31]	90.1	96.4	-	-
CTR-GCN [9]	92.4	96.8	88.9	90.6
InfoGCN [14]	93.0	**97.1**	89.8	91.2
FR-Head [68]	92.8	96.8	89.5	90.9
HD-GCN [29]	**93.0**	97.0	89.8	91.2
Koompan Pooling [56]	92.9	96.8	**90.0**	**91.3**
FRF-GCN [64]	91.3	96.5	87.1	88.4
Graph2Net [57] (Backbone)	90.1	96.0	86.0	87.6
IGPN-Light (Ours)	92.2	96.4	89.0	90.2
IGPN-Heavy (Ours)	92.9	96.7	89.6	91.1

method, IGPN-Heavy maintains a certain accuracy advantage. Following CTR-GCN, recent methods have shown limited improvements in accuracy. Compared with these methods, our method still achieves comparable performance. On the UWA3D dataset, From Table 5, IGPN-Heavy outperforms previous methods in 7 out of 12 possible combinations, with a higher average performance when compared with previous state-of-the-art.

Our method has achieved a clear lead in both accuracy and efficiency under single-stream evaluation. Under multi-stream evaluation, our method has also achieved performance similar to or better than the optimal method in terms of accuracy.

4.4 Ablation Studies

In this section, we present a comprehensive set of experiments and analyses for all of our proposed innovations on cross-subject of NTU-RGB+D 60.

Effects of different modules on performance. To demonstrate the effectiveness of our innovations, we have chosen Graph2Net [57] with same number of layers as the baseline for comparison. As

Table 5: Accuracy (%) comparison of our approach under multi-stream evaluation with the state of the art methods on UWA3D Multiview Activity II Dataset.

Training views	V_1&V_2		V_1&V_3		V_1&V_4		V_2&V_3		V_2&V_4		V_3&V_4		Mean
Test view	V_3	V_4	V_2	V_4	V_2	V_3	V_1	V_4	V_1	V_3	V_1	V_2	
HOJ3D[60]	15.3	28.2	17.3	27.0	14.6	13.4	15.0	12.9	22.1	13.5	20.3	12.7	17.7
Actionlet ensemble[54]	45.0	40.4	35.1	36.9	34.7	36.0	49.5	29.3	57.1	35.4	49.0	29.3	39.8
Lie Group [53]	49.4	42.8	34.6	39.7	38.1	44.8	53.3	33.5	53.6	41.2	56.7	32.6	43.4
Enhanced skeleton visualization[38]	66.4	68.1	56.8	66.1	58.8	66.2	74.2	67.0	76.9	64.8	72.2	54.0	66.0
Ensemble TS-LSTM v2[27]	72.1	79.1	74.0	77.6	75.6	70.1	79.6	79.9	83.9	66.1	79.2	69.7	75.6
ShiftGCN[12]	73.3	85.8	71.6	82.3	75.6	74.1	86.3	82.3	83.1	74.1	85.1	74.4	79.0
ShiftGCN++[13]	74.5	**86.6**	76.8	85.0	75.2	73.7	86.3	82.7	85.5	73.3	84.7	74.0	79.9
MCMT-Net [58]	**76.1**	86.0	**78.0**	84.6	76.8	77.3	86.3	81.9	85.1	**74.9**	87.1	**74.4**	80.7
IGPN-Heavy (Ours)	74.1	84.3	76.8	**86.6**	**78.7**	**78.9**	**86.7**	**84.6**	**87.1**	74.5	**88.6**	73.6	**81.2**

Table 6: The impact of different modelling strategies.

Method	Adaptive Pooling	CFB	ISM	Acc (%)
baseline	✗	✗	✗	88.8
with SGP [8]	✗	✗	✗	88.3
with IGPN	✓	✗	✗	89.1
	✓	✓	✗	89.9
	✓	✗	✓	90.2
	✓	✓	✓	91.2

Table 7: Configurations analysis in the pooling strategy.

Method	r	σ	Location	Acc (%)
ASGP	2	Tanh	$1-3$	88.4
	4	Tanh	$1-3$	89.1
	8	Tanh	$1-3$	88.6
	4	Sigmoid	$1-3$	88.3
	4	SoftMax	$1-3$	88.4
	4	Tanh	1	88.5
	4	Tanh	$1-2$	88.6
ASGP *wo*. Res	4	Tanh	$1-3$	88.6

Table 8: Configurations analysis in cross fusion block.

s	Location	Fusion	Acc (%)
0.25	$1-3$	Sum	89.5
0.5	$1-3$	Sum	89.9
0.75	$1-3$	Sum	89.6
0.5	1	Sum	89.0
0.5	$1-2$	Sum	89.5
0.5	$1-3$	Concat	89.7

Table 9: Configurations analysis in information supplement module.

BatchNorm	Embed	layer	channel	Acc (%)
✗	GraphConv	2	32	90.1
✓	GraphConv	2	32	91.2
✓	2D Conv	2	32	90.7
✓	GraphConv	1	32	90.9
✓	GraphConv	3	32	91.0
✓	GraphConv	2	16	90.9
✓	GraphConv	2	64	90.9

shown in Table 6, we conducted a thorough analysis of our proposed modules and found that the adaptive mechanism improved the model's accuracy to 89.1%. The fusion module further improved the accuracy by 0.8%. Finally, after supplementing the input features, the recognition performance reached 91.2%. The experimental results indicate a steady improvement in recognition accuracy by gradually adding our innovations. The final accuracy was significantly better than that of the original model and SGP.

Performance comparison on different backbones. In this section, we investigate the applicability of our pooling strategy to other widely used methods, including AGCN [49], Graph2Net [57] and CTR-GCN [9]. Specifically, we keep the basic framework unchanged and only replace specific graph convolution operations with these methods separately. The results in Table 1 consistently demonstrate that the proposed improved pooling strategy can enhance model accuracy while reducing computational overhead significantly. We choose the Graph2Net as the default graph convolution operation for all other experiments.

Analysis of specific configurations. In this section, we will conduct detailed experiments and analyses on the parameter selection and structural design.

- Structure pooling strategy with region awareness. As mentioned previously, we utilise SGP with the new implementation as the baseline to verify the impact of the normalisation function σ, the parameters of projection space r, as well as the location of the pooling operation. The experimental results are presented in Table 7. Firstly, we observe that the recognition accuracy improves after incorporating the adaptive mechanism. Moreover, our model performs best when $r = 4$ and Tanh is used as the normalisation function. Secondly, the model achieves an accuracy of 89.1% when the adaptive mechanism is introduced in all three pooling operations. Our also investigate the impact of retaining the initially assigned matrix through residual connection when introducing the adaptive mechanism. Results showed that removing the residual mechanism led to a drop in performance to 88.6%, highlighting the importance of preserving fundamental structural properties while introducing flexibility.

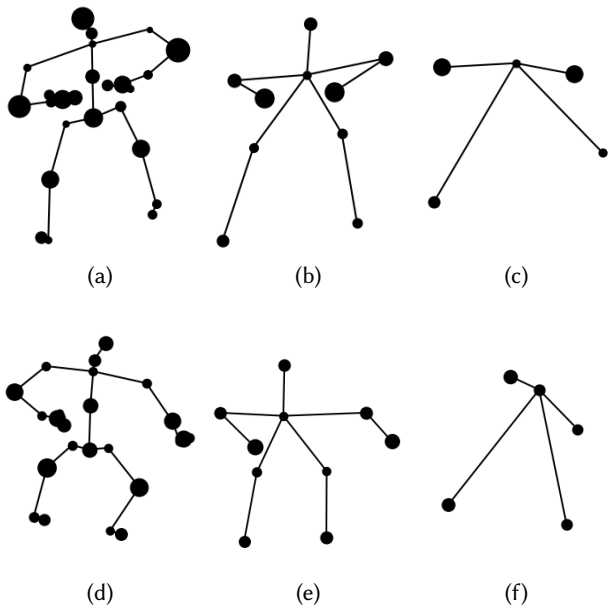


Figure 6: Visualisation of region awareness. We select two typical behaviours and visualise the associated responses of their nodes. In order to facilitate the display, we perform certain transformations on the specific values corresponding to the node size; that is, the larger the node, the more critical the node is. (a)(b)(c) represents 'rub two hands', (d)(e)(f) represents 'jump up'.

- Cross fusion block. In our experiments, we investigated the impact of feature augmentation on performance when incorporating the new pooling strategy. Firstly, we explored the effect of the weight parameter in Equation 10. The results in Table 8 show that the best performance is achieved when the weights of the two features being fused are both 0.5. After fixing the weight s, we further investigated the optimal location and fusion strategy for the fusion module. The experimental results show that the performance improves as more fusion modules are added, ultimately reaching 89.9%. We also found that summation performs better than concatenation in the channel dimension. Notably, we perform the fusion after the global pooling operation for the last fusion.
- Information supplement module. The experiments conducted in this section aimed to investigate the impact of parameter selection and structural design on the performance of the model. The results from Table 9 show that batch normalisation on both position-based and vector-based features is necessary for better performance. Also, changing the graph-based embedding operation to regular 2D convolution embedding resulted in a 0.5% decrease in performance. Furthermore, the optimal structure for graph embedding is found to be two layers of graph convolution with a final projected space dimension of 32.

Table 10: The experimental results obtained by the multi-stream fusion.

Methods	Spatial	Motion	Acc (%)
IGPN-Light	✓	✗	90.2
IGPN-Light	✓	✓	91.6
IGPN-Light$_{En}$	✓	✓	92.2
IGPN-Heavy	✓	✗	91.2
IGPN-Heavy	✓	✓	92.3
IGPN-Heavy$_{En}$	✓	✓	92.9

Effects of different modules on performance. We have presented the recognition accuracy of the models with the three proposed modules in each category, as shown in Figure 7. Based on the observations and analysis of the figure, several conclusions can be drawn: The incremental incorporation of innovative modules has a discernible impact on the recognition accuracy of different categories. For instance, the addition of the adaptive mechanism markedly enhances the recognition accuracy for behaviours such as drop and rub two hands, which necessitate more flexible pooling operations compared to the baseline. On the other hand, the fusion module performs exceptionally well in reading, wipe face, and other behaviours by providing more distinguishable multi-granularity feature representations. Moreover, the results reveal that the introduction of these two innovations had a certain degree of adverse effects on some categories, such as putting on a shoe and taking a selfie. Fortunately, the supplementary modules has successfully mitigated these adverse effects.

Visualisation of region awareness. To provide a more tangible illustration of our novel pooling method, we generated a visualisation of the region perception in Figure 6, where the size of each point represents the importance of its corresponding position, reflecting the region that needs to be preserved during the pooling operation. From (a), we can observe that the response at the hand joints is relatively large for the action 'rub two hands', and this trend is also evident in (b) and (c). For the action 'jump up' in (d) (e) (f), the response of the hands and legs is more evident due to the need for the cooperation of the entire body. The observed phenomena demonstrate that our model can adaptively enhance information acquisition in critical regions for actions that focus on local areas or require whole-body coordination, providing discriminative feature expression during pooling.

Multi-stream fusion. The effectiveness of multi-stream network fusion has been extensively demonstrated in previous research [9, 12, 39, 49, 57]. We also adopt this approach to improve the classification performance of our algorithm further for a fairer comparison. First, we separately verify the impact of spatial features and motion features on performance. The results in Table 10 show that the recognition effect can be improved to 91.6% and 92.3% by adopting the two-stream fusion strategy. We also explored the use of an ensemble approach where we trained another separate network using only half of the frames. This technique is commonly used in other works such as [32, 69]. We denoted this model as IGPN$_{En}$. The results in Table 10 show that this ensemble mechanism improves the final accuracy to 92.2% and 92.9% respectively.

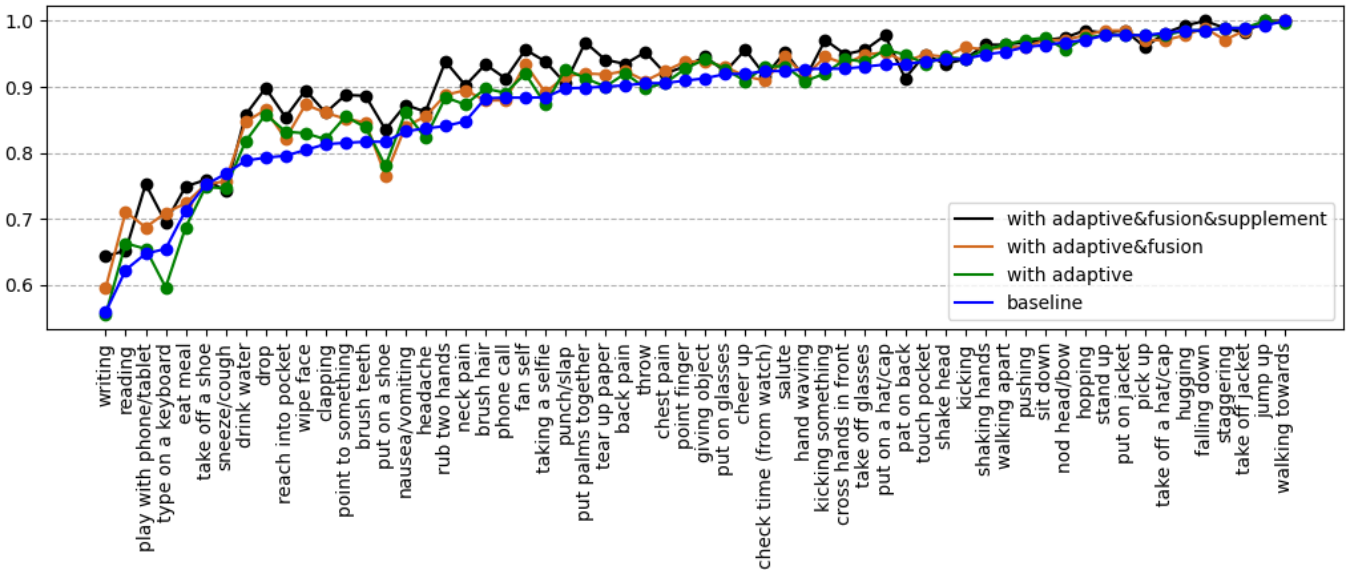


Figure 7: Comparison results for each category on the NTU-RGB+D Cross-Subject dataset.

Table 11: The experimental results obtained by different multi-stream fusion strategies.

Method	Stream	Flops (G)	Acc (%)
IGPN-Light (no ISM)	J&B&JM&BM	2.05	92.0
IGPN-Light	J&J$_{En}$&JM&JM$_{En}$	1.33	92.2
IGPN-Heavy (no ISM)	J&B&JM&BM	5.38	92.5
IGPN-Heavy	J&J$_{En}$&JM&JM$_{En}$	3.83	92.9

Effects of different multi-stream strategies Since the designed ISM module includes joint-based and bone-based features, there may be confusion about the difference between this method and training joint-based and bone-based networks separately. Indeed, mainstream methods[9, 12, 39, 49] generally choose to train separate joint-based, bone-based, and corresponding motion-based features, and fuse the corresponding classification features to obtain the final recognition result of the multi-stream recognition network. As shown in Table 11, we choose to delete the ISM module from the IGPN and train on different feature streams respectively, and finally fuse their results. The results show that compared with our default method, this strategy not only brings a significant increase in computational effort (2.05GFlops vs 1.33GFlops on IGPN-Light, 5.38GFlops vs 3.83GFlops on IGPN-Heavy), and also leads to a decrease in accuracy performance (92.0% vs 92.2% on IGPN-Light 92.5% vs 92.9% on IGPN-Heavy) .

5 CONCLUSION

Existing skeleton-based graph pooling research suffers from the failure to consider graph properties of the skeleton, resulting in inflexibility and loss of vital information. Here we have adopted several effective improvement strategies to enhance and supplement the pooling strategy from multiple perspectives. (1) We comprehensively analysed the strengths and limitations of various traditional graph pooling strategies and devised a pooling strategy that significantly enhances flexibility while preserving the structural characteristics based on the local-global association. (2) To address the issues of information loss and network under-saturation caused by pooling operations, we propose block-level multi-granularity representations and input-level discriminative information embedding. On the above innovations, we constructed IGPN. The lightweight version can greatly improve efficiency and effect at the same time; the heavyweight version can further improve classification accuracy by sacrificing certain computing efficiency.

REFERENCES

[1] Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. 2020. Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics* 39, 4 (2020), 62–1.

[2] Danilo Avola, Marco Cascio, Luigi Cinque, Gian Luca Foresti, Cristiano Massaroni, and Emanuele Rodola. 2019. 2-D skeleton-based action recognition via two-branch stacked LSTM-RNNs. *IEEE Transactions on Multimedia* 22, 10 (2019), 2481–2496.

[3] Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. 2020. Spectral clustering with graph neural networks for graph pooling. In *International Conference on Machine Learning*. PMLR, 874–883.

[4] Filippo Maria Bianchi and Veronica Lachi. 2024. The expressive power of pooling in graph neural networks. *Advances in Neural Information Processing Systems* 36 (2024).

[5] Y-Lan Boureau, Jean Ponce, and Yann LeCun. 2010. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th International Conference on Machine Learning*. 111–118.

[6] Carlos Caetano, François Brémond, and William Robson Schwartz. 2019. Skeleton image representation for 3D action recognition based on tree structure and reference joints. In *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE, 16–23.

[7] Hsuan-Sheng Chen, Hua-Tsung Chen, Yi-Wen Chen, and Suh-Yin Lee. 2006. Human action recognition using star skeleton. In *Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks*. 171–178.

[8] Yuxin Chen, Gaoqun Ma, Chunfeng Yuan, Bing Li, Hui Zhang, Fangshi Wang, and Weiming Hu. 2020. Graph convolutional network with structure pooling and joint-wise channel attention for action recognition. *Pattern Recognition* 103 (2020), 107321.

[9] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. 2021. Channel-wise topology refinement graph convolution for skeleton-based

action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13359–13368.

[10] Zhenjie Chen, Hongsong Wang, and Jie Gui. 2023. Occluded Skeleton-Based Human Action Recognition with Dual Inhibition Training. In *Proceedings of the 31st ACM International Conference on Multimedia*. 2625–2634.

[11] Ke Cheng, Yifan Zhang, Congqi Cao, Lei Shi, Jian Cheng, and Hanqing Lu. 2020. Decoupling gcn with dropgraph module for skeleton-based action recognition. In *European Conference on Computer Vision*. Springer, 536–553.

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

[13] Ke Cheng, Yifan Zhang, Xiangyu He, Jian Cheng, and Hanqing Lu. 2021. Extremely lightweight skeleton-based action recognition with shiftgcn++. *IEEE Transactions on Image Processing* 30 (2021), 7333–7348.

[14] Hyung-gun Chi, Myoung Hoon Ha, Seunggeun Chi, Sang Wan Lee, Qixing Huang, and Karthik Ramani. 2022. Infogcn: Representation learning for human skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 20186–20196.

[15] Chhavi Dhiman and Dinesh Kumar Vishwakarma. 2020. View-invariant deep architecture for human action recognition using two-stream motion and shape temporal dynamics. *IEEE Transactions on Image Processing* 29 (2020), 3835–3844.

[16] Yong Du, Yun Fu, and Liang Wang. 2015. Skeleton based action recognition with convolutional neural network. In *2015 3rd IAPR Asian Conference on Pattern Recognition*. IEEE, 579–583.

[17] Yong Du, Wei Wang, and Liang Wang. 2015. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1110–1118.

[18] Hongyang Gao and Shuiwang Ji. 2019. Graph U-Nets. In *International Conference on Machine Learning*. PMLR, 2083–2092.

[19] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. 2019. Graph convolutional tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4649–4659.

[20] Lei Han, Xinxiao Wu, Wei Liang, Guangming Hou, and Yunde Jia. 2010. Discriminative human action recognition in the learned hierarchical manifold space. *Image and Vision Computing* 28, 5 (2010), 836–849.

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

[22] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[23] Qibin Hou, Li Zhang, Ming-Ming Cheng, and Jiashi Feng. 2020. Strip pooling: Rethinking spatial pooling for scene parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4003–4012.

[24] Chang-Qin Huang, Fan Jiang, Qiong-Hao Huang, Xi-Zhe Wang, Zhong-Mei Han, and Wei-Yu Huang. 2022. Dual-graph attention convolution network for 3-D point cloud classification. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[25] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. 2017. A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3288–3297.

[26] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. 2018. Learning clip representations for skeleton-based 3d action recognition. *IEEE Transactions on Image Processing* 27, 6 (2018), 2842–2855.

[27] Inwoong Lee, Doyoung Kim, Seoungyoon Kang, and Sanghoon Lee. 2017. Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In *Proceedings of IEEE International Conference on Computer Vision*. 1012–1020.

[28] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. 2019. Self-attention graph pooling. In *International Conference on Machine Learning*. PMLR, 3734–3743.

[29] Jungho Lee, Minhyeok Lee, Dogyoon Lee, and Sangyoun Lee. 2023. Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10444–10453.

[30] Ce Li, Chunyu Xie, Baochang Zhang, Jungong Han, Xiantong Zhen, and Jie Chen. 2021. Memory attention networks for skeleton-based action recognition. *IEEE Transactions on Neural Networks and Learning Systems* 33, 9 (2021), 4800–4814.

[31] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. 2021. Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 6 (2021), 3316–3333.

[32] Ji Lin, Chuang Gan, and Song Han. 2019. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7083–7093.

[33] Jian Liu, Naveed Akhtar, and Ajmal Mian. 2020. Adversarial attack on skeleton-based human action recognition. *IEEE Transactions on Neural Networks and Learning Systems* 33, 4 (2020), 1609–1622.

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

[35] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C Kot. 2017. Global context-aware attention lstm networks for 3d action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1647–1656.

[36] Jiaying Liu, Feng Xia, Xu Feng, Jing Ren, and Huan Liu. 2022. Deep graph learning for anomalous citation detection. *IEEE Transactions on Neural Networks and Learning Systems* 33, 6 (2022), 2543–2557.

[37] Kai Liu, Lei Gao, Naimul Mefraz Khan, Lin Qi, and Ling Guan. 2020. A multi-stream graph convolutional networks-hidden conditional random field model for skeleton-based action recognition. *IEEE Transactions on Multimedia* 23 (2020), 64–76.

[38] Mengyuan Liu, Hong Liu, and Chen Chen. 2017. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition* 68 (2017), 346–362.

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

[40] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, Vol. 2. Makuhari, 1045–1048.

[41] Xuan Son Nguyen. 2021. GeomNet: A Neural Network Based on Riemannian Geometries of SPD Matrix Space and Cholesky Space for 3D Skeleton-Based Interaction Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13379–13389.

[42] Wei Peng, Xiaopeng Hong, Haoyu Chen, and Guoying Zhao. 2020. Learning graph convolutional network for skeleton-based human action recognition by neural searching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 2669–2676.

[43] Wei Peng, Xiaopeng Hong, and Guoying Zhao. 2021. Tripool: Graph triplet pooling for 3D skeleton-based action recognition. *Pattern Recognition* 115 (2021), 107921.

[44] Yongfeng Qi, Jinlin Hu, Liqiang Zhuang, and Xiaoxu Pei. 2022. Semantic-guided multi-scale human skeleton action recognition. *Applied Intelligence* (2022), 1–16.

[45] Zhenyue Qin, Yang Liu, Pan Ji, Dongwoo Kim, Lei Wang, RI McKay, Saeed Anwar, and Tom Gedeon. 2022. Fusing higher-order features in graph neural networks for skeleton-based action recognition. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[46] Hossein Rahmani, Arif Mahmood, Du Huynh, and Ajmal Mian. 2016. Histogram of oriented principal components for cross-view action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 12 (2016), 2430–2443.

[47] Dominik Scherer, Andreas Müller, and Sven Behnke. 2010. Evaluation of pooling operations in convolutional architectures for object recognition. In *International Conference on Artificial Neural Networks*. Springer, 92–101.

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

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

[50] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. 2019. 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*. 1227–1236.

[51] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. 2017. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[52] Zhigang Tu, Jiaxu Zhang, Hongyan Li, Yujin Chen, and Junsong Yuan. 2022. Joint-bone Fusion Graph Convolutional Network for Semi-supervised Skeleton Action Recognition. *IEEE Transactions on Multimedia* (2022), 1–1. https://doi.org/10.1109/TMM.2022.3168137

[53] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. 2014. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 588–595.

[54] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. 2013. Learning actionlet ensemble for 3D human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 5 (2013), 914–927.

[55] Minsi Wang, Bingbing Ni, and Xiaokang Yang. 2020. Learning Multi-View Interactional Skeleton Graph for Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), 1–1. https://doi.org/10.1109/TPAMI.2020.3032738

[56] Xinghan Wang, Xin Xu, and Yadong Mu. 2023. Neural Koopman pooling: Control-inspired temporal dynamics encoding for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10597–10607.

[57] Cong Wu, Xiao-Jun Wu, and Josef Kittler. 2021. Graph2Net: Perceptually-enriched graph learning for skeleton-based action recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 32, 4 (2021), 2120–2132.

[58] Cong Wu, Xiao-Jun Wu, Tianyang Xu, Zhongwei Shen, and Josef Kittler. 2024. Motion Complement and Temporal Multifocusing for Skeleton-Based Action Recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 34, 1 (2024), 34–45. https://doi.org/10.1109/TCSVT.2023.3236430

[59] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2021. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2021), 4–24.

[60] Lu Xia, Chia-Chih Chen, and Jake K Aggarwal. 2012. View invariant human action recognition using histograms of 3d joints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 20–27.

[61] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[62] Fanfan Ye, Shiliang Pu, Qiaoyong Zhong, Chao Li, Di Xie, and Huiming Tang. 2020. Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*. 55–63.

[63] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. *Advances in Neural Information Processing Systems* 31 (2018).

[64] Xiao Yun, Chenglong Xu, Kevin Riou, Kaiwen Dong, Yanjing Sun, Song Li, Kevin Subrin, and Patrick Le Callet. 2024. Behavioral Recognition of Skeletal Data Based on Targeted Dual Fusion Strategy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 6917–6925.

[65] Sophyani Banaamwini Yussif, Ning Xie, Yang Yang, and Heng Tao Shen. 2023. Self-Relational Graph Convolution Network for Skeleton-Based Action Recognition. In *Proceedings of the 31st ACM International Conference on Multimedia*. 27–36.

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

[67] Xikun Zhang, Chang Xu, Xinmei Tian, and Dacheng Tao. 2020. Graph edge convolutional neural networks for skeleton-based action recognition. *IEEE Transactions on Neural Networks and Learning Systems* 31, 8 (2020), 3047–3060.

[68] Huanyu Zhou, Qingjie Liu, and Yunhong Wang. 2023. Learning discriminative representations for skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10608–10617.

[69] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. 2018. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European Conference on Computer Vision*. 695–712.